EC 3352 – DIGITAL SYSTEM DESIGN

UNIT – II : COMBINATIONAL LOGIC CIRCUITS

2.5 DECODER AND SEGMENT DISPLY DECODER

A decoder is a combinational circuit that converts binary information from ‗n‘ input lines to a maximum of ‗2^n‘ unique output lines. The general structure of decoder circuit is –

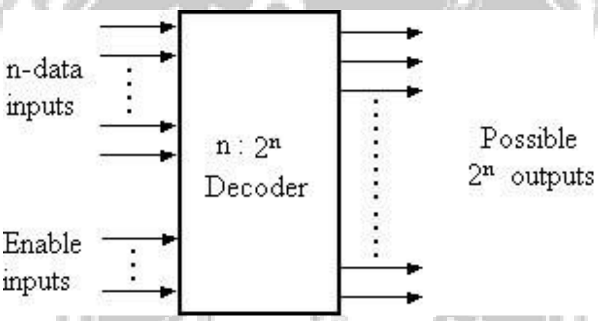


Fig : 2.24 - General structure of decoder

The encoded information is presented as ‗n‘ inputs producing ‗2^n‘ possible outputs. The 2^n output values are from 0 through 2^n-1. A decoder is provided with enable inputs to activate decoded output based on data inputs. When any one enable input is unasserted, all outputs of decoder are disabled.

Binary Decoder (2 to 4 decoder):

A binary decoder has ‗n‘ bit binary input and a one activated output out of 2^n outputs. A binary decoder is used when it is necessary to activate exactly one of 2^n outputs based on an n-bit input value.

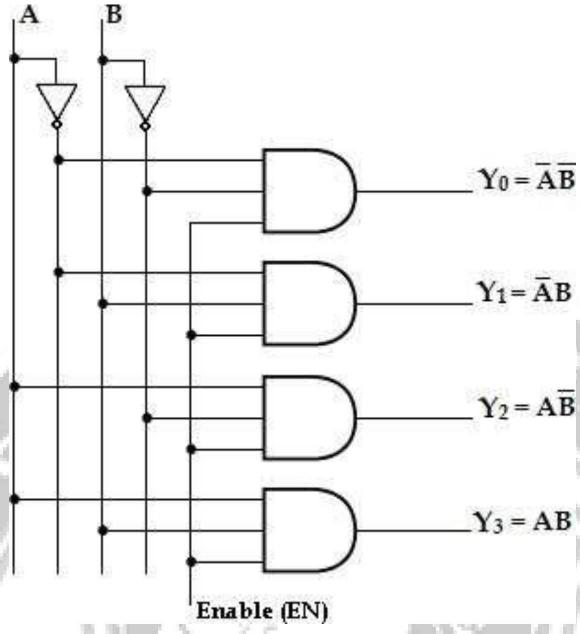


Fig : 2.25 - 2-to-4 Line decoder

Here the 2 inputs are decoded into 4 outputs, each output representing one of the minterms of the two input variables.

Input s			**Outputs**			
Enable	**A**	**B**	**Y3**	**Y2**	**Y1**	**Y0**
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

As shown in the truth table, if enable input is 1 (EN= 1) only one of the outputs (Y_0 – Y_3),is active for a given input.

The output Y_0 is active, ie., Y_0= 1 when inputs A= B= 0,

Y_1 is active when inputs, A= 0 and B= 1,
Y_2 isactive, when input A= 1 and B= 0, Y_3
is active, when inputs A= B= 1.

3-to-8 Line Decoder:

A 3-to-8 line decoder has three inputs (A, B, C) and eight outputs (Y_0- Y_7). Based on the3 inputs one of the eight outputs is selected.

The three inputs are decoded into eight outputs, each output representing one of the minterms of the 3-input variables. This decoder is used for binary-to-octal conversion. The input variables may represent a binary number and the outputs will represent the eight digits in the octal number system. The output variables are mutually exclusive because only one output can be equal to 1 at any one time. The output line whose value is equal to 1 represents the minterm equivalent of the binary number presently available in the input lines.

Input s			Output s							
A	**B**	**C**	**Y0**	**Y1**	**Y2**	**Y3**	**Y4**	**Y5**	**Y6**	**Y7**
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

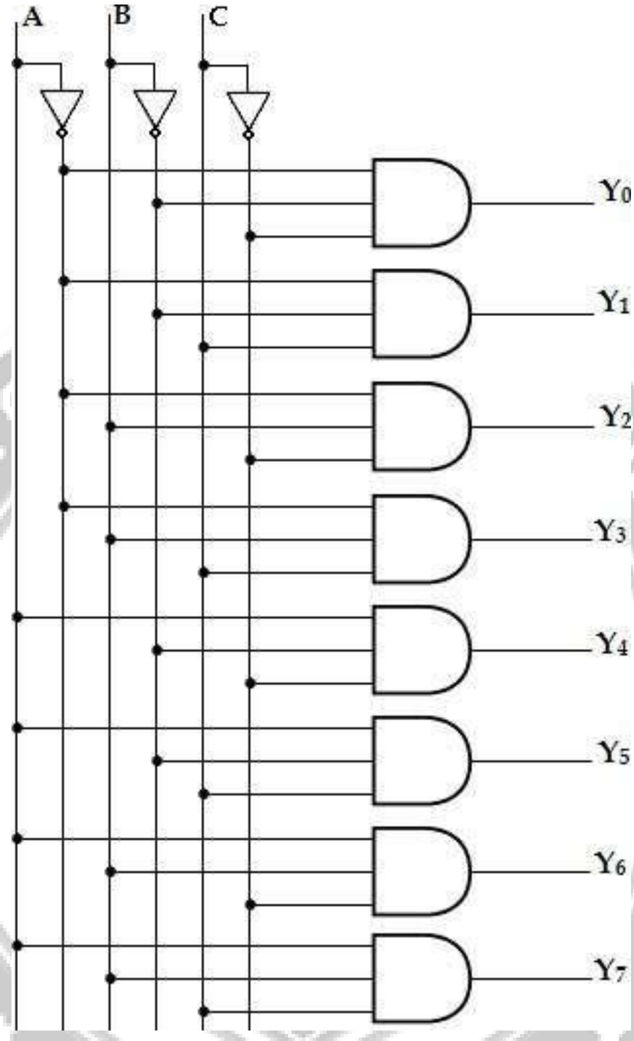
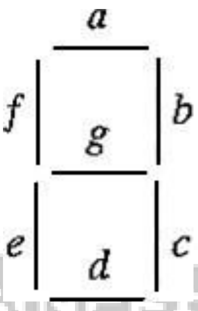


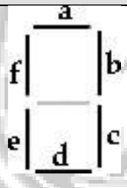
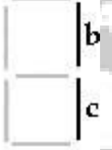
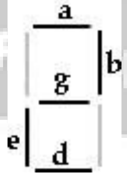
Fig : 2.26 - 3-to-8 line decoder

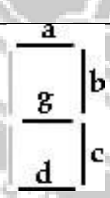
BCD to 7-Segment Display Decoder:

A seven-segment display is normally used for displaying any one of the decimal digits, 0 through 9. A BCD-to-seven segment decoder accepts a decimal digit in BCD and generates the corresponding seven-segment code.

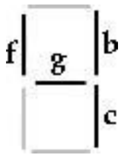
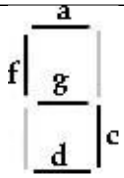
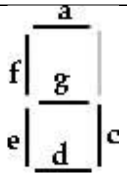
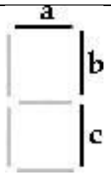
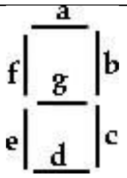
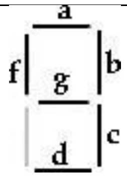


Each segment is made up of a material that emits light when current is passed through it. The segments activated during each digit display are tabulated as—

Digit	Display	Segments Activated
0	 	a, b, c, d, e, f
1		b, c
2	  	a, b, d, e, g
3		a, b, c, d, g

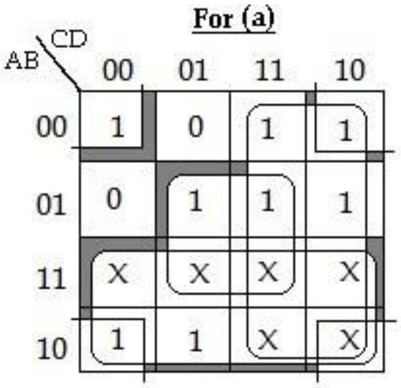
Truth table:

4		b, c, f, g
5		a, c, d, f, g
6		a, c, d, e, f, g
7		a, b, c
8		a, b, c, d, e, f, g
9		a, b, c, d, f, g

	BCD code				7-Segment code						
Digit	**A**	**B**	**C**	**D**	**a**	**b**	**c**	**d**	**e**	**f**	**g**
0	0	0	0	0	1	1	1	1	1	1	0
1	0	0	0	1	0	1	1	0	0	0	0
2	0	0	1	0	1	1	0	1	1	0	1
3	0	0	1	1	1	1	1	1	0	0	1
4	0	1	0	0	0	1	1	0	0	1	1
5	0	1	0	1	1	0	1	1	0	1	1
6	0	1	1	0	1	0	1	1	1	1	1
7	0	1	1	1	1	1	1	0	0	0	0
8	1	0	0	0	1	1	1	1	1	1	1
9	1	0	0	1	1	1	1	1	0	1	1

K-map Simplification:

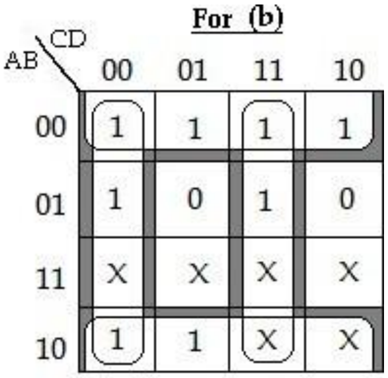
For (a)



AB\CD	00	01	11	10
00	1	0	1	1
01	0	1	1	1
11	X	X	X	X
10	1	1	X	X

a= A+ C+ BD+ B'D'

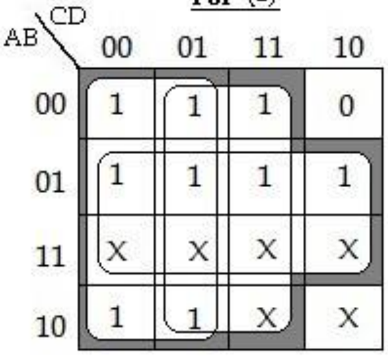
For (b)



AB\CD	00	01	11	10
00	1	1	1	1
01	1	0	1	0
11	X	X	X	X
10	1	1	X	X

b= B'+ C'D'+ CD

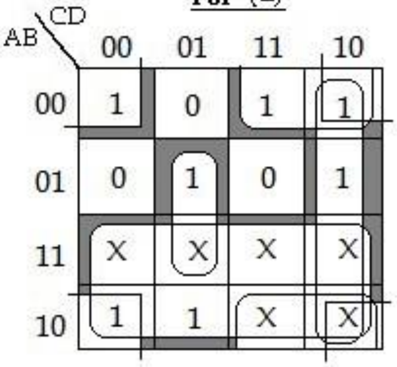
For (c)



AB\CD	00	01	11	10
00	1	1	1	0
01	1	1	1	1
11	X	X	X	X
10	1	1	X	X

c= B+ C'+ D

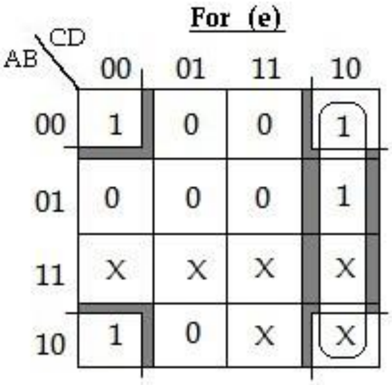
For (d)



AB\CD	00	01	11	10
00	1	0	1	1
01	0	1	0	1
11	X	X	X	X
10	1	1	X	X

d= B'D'+ CD'+ BC'D+ B'C+ A

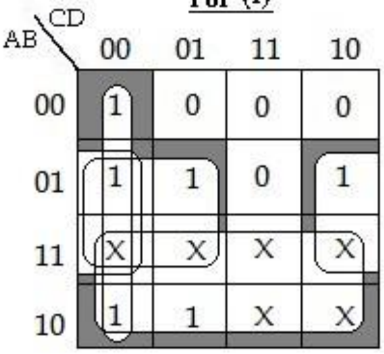
For (e)



AB\CD	00	01	11	10
00	1	0	0	1
01	0	0	0	1
11	X	X	X	X
10	1	0	X	X

e= B'D'+ CD'

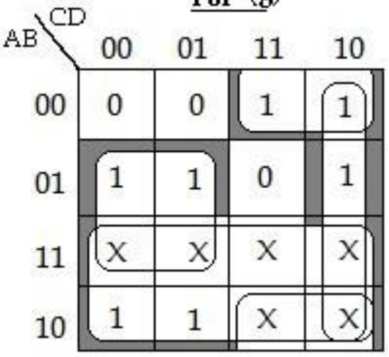
For (f)



AB\CD	00	01	11	10
00	1	0	0	0
01	1	1	0	1
11	X	X	X	X
10	1	1	X	X

f= A+ C'D'+ BC'+ BD'

For (g)



AB\CD	00	01	11	10
00	0	0	1	1
01	1	1	0	1
11	X	X	X	X
10	1	1	X	X

g= A+ BC'+ B'C+ CD'

Applications of decoders:

1. Decoders are used in counter system.
2. They are used in analog to digital converter.
3. Decoder outputs can be used to drive a display system.

ENCODERS AND PRIOPRITY ENCODER :

An encoder is a digital circuit that performs the inverse operation of a decoder. Hence, the opposite of the decoding process is called encoding. An encoder is a combinational circuit that converts binary information from 2^n input lines to a maximum of _n‘ unique output lines. The general structure of encoder circuit is –

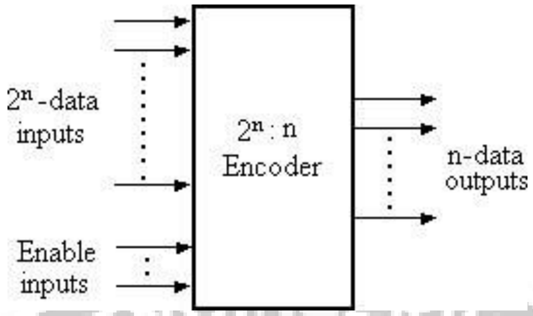


Fig : 2.27 - General structure of Encoder

It has 2^n input lines, only one which 1 is active at any time and _n‘ output lines. It encodes one of the active inputs to a coded binary output with _n‘ bits. In an encoder, the number of outputs is less than the number of inputs.

Octal-to-Binary Encoder:

It has eight inputs (one for each of the octal digits) and the three outputs that generate the corresponding binary number. It is assumed that only one input has a value of 1 at any given time.

Inputs								**Outputs**		
D0	**D1**	**D2**	**D3**	**D4**	**D5**	**D6**	**D7**	**A**	**B**	**C**
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

The encoder can be implemented with OR gates whose inputs are determined directly from the truth table. Output z is equal to 1, when the input octal digit is 1 or 3 or 5 or 7. Output y is 1 for octal digits 2, 3, 6, or 7 and the output is 1 for digits 4, 5, 6 or 7. These conditions can be expressed by the following output Boolean functions:

$\mathbf{z= D_1+ D_3+ D_5+ D_7}$

$\mathbf{y= D_2+ D_3+ D_6+ D_7\ x= D_4+}$

$\mathbf{D_5+D_6+ D_7}$

The encoder can be implemented with three OR gates. The encoder defined in the below table, has the limitation that only one input can be active at any given time. If two inputs are active simultaneously, the output produces an undefined combination.

For eg., if D_3 and D_6 are 1 simultaneously, the output of the encoder may be 111. This does not represent either D_6 or D_3. To resolve this problem, encoder circuits must establish an input priority to ensure that only one input is encoded. If we establish a higher priority for inputs with higher subscript numbers and if D_3 and D_6 are 1 at the same time, the output will be 110 because D_6 has higher priority than D_3.

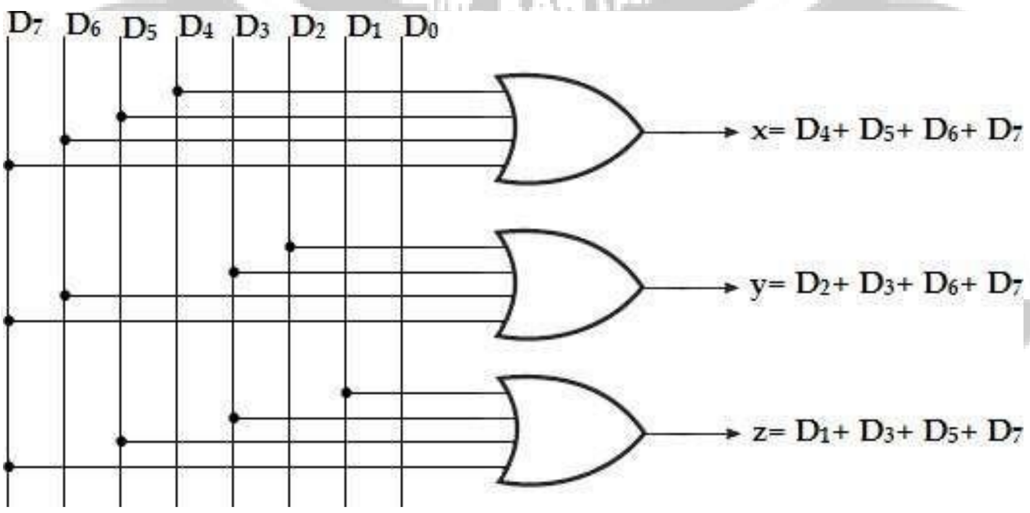


Fig : 2.28 - Octal-to-Binary Encoder

Another problem in the octal-to-binary encoder is that an output with all 0‘s is generated when all the inputs are 0; this output is same as when D_0 is equal to 1. The discrepancy can be resolved by providing one more output to indicate that atleast one input is equal to 1.

Priority Encoder:

A priority encoder is an encoder circuit that includes the priority function. In priority encoder, if two or more inputs are equal to 1 at the same time, the input having the highest priority will take precedence.

In addition to the two outputs x and y, the circuit has a third output, V (valid bit indicator). It is set to 1 when one or more inputs are equal to 1. If all inputs are 0, thereis no valid input and V is equal to 0.

The higher the subscript number, higher the priority of the input. Input D_3, has the highest priority. So, regardless of the values of the other inputs, when D_3 is 1, the output for xy is 11.

D_2 has the next priority level. The output is 10, if D_2= 1 provided D_3= 0. The output for D_1 is generated only if higher priority inputs are 0, and so on down the priority levels.

Truth table:

Inputs				**Outputs**		
D0	**D1**	**D2**	**D3**	**x**	**y**	**V**
0	0	0	0	x	x	0
1	0	0	0	0	0	1
x	1	0	0	0	1	1
x	x	1	0	1	0	1

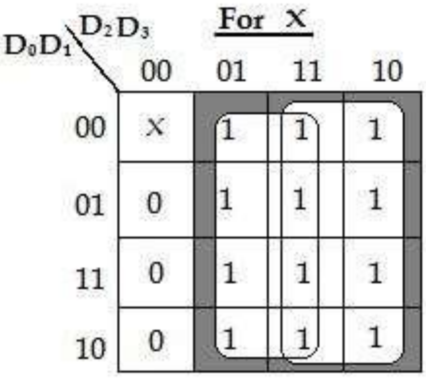
x	x	x	1	1	1	1

Although the above table has only five rows, when each don‘t care condition is replaced first by 0 and then by 1, we obtain all 16 possible input combinations. For example, the third row in the table with X100 represents minterms 0100 and 1100. The don‘t care condition is replaced by 0 and 1 as shown in the table below.

Modified Truth table:

	Inputs			Outputs		
D0	**D1**	**D2**	**D3**	**x**	**y**	**V**
0	0	0	0	x	x	0
1	0	0	0	0	0	1
0	1	0	0	0	1	1
1	1	0	0			
0	0	1	0	1	0	1
0	1	1	0			
1	0	1	0			
1	1	1	0			
0	0	0	1	1	1	1
0	0	1	1			
0	1	0	1			
0	1	1	1			
1	0	0	1			
1	0	1	1			
1	1	0	1			
1	1	1	1			

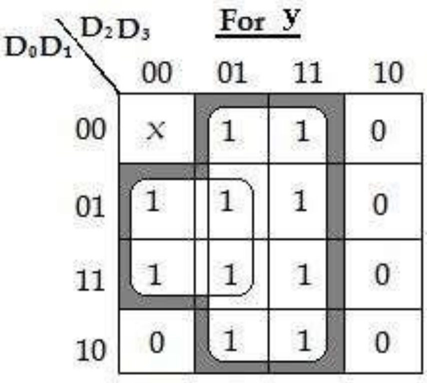
K-map Simplification:



For X

$D_0D_1 \backslash D_2D_3$	00	01	11	10
00	X	1	1	1
01	0	1	1	1
11	0	1	1	1
10	0	1	1	1

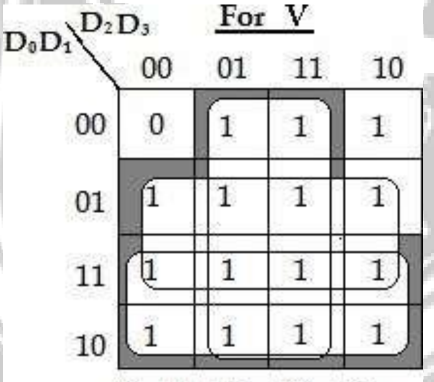
$x = D_2 + D_3$



For y

$D_0D_1 \backslash D_2D_3$	00	01	11	10
00	X	1	1	0
01	1	1	1	0
11	1	1	1	0
10	0	1	1	0

$y = D_3 + D_1D_2$



For V

$D_0D_1 \backslash D_2D_3$	00	01	11	10
00	0	1	1	1
01	1	1	1	1
11	1	1	1	1
10	1	1	1	1

$V = D_0 + D_1 + D_2 + D_3$

The priority encoder is implemented according to the above Boolean functions.

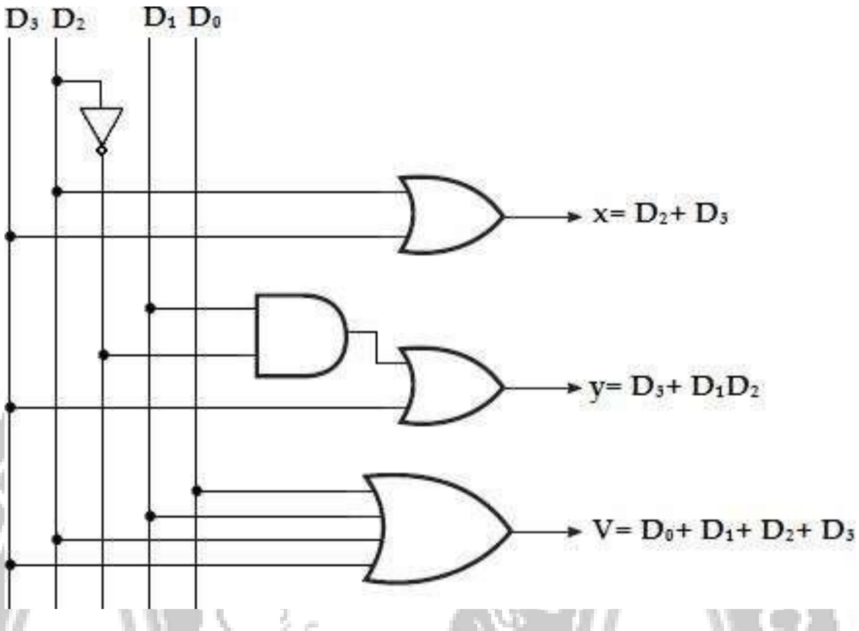


Fig : 2.29 - 4-Input Priority Encoder