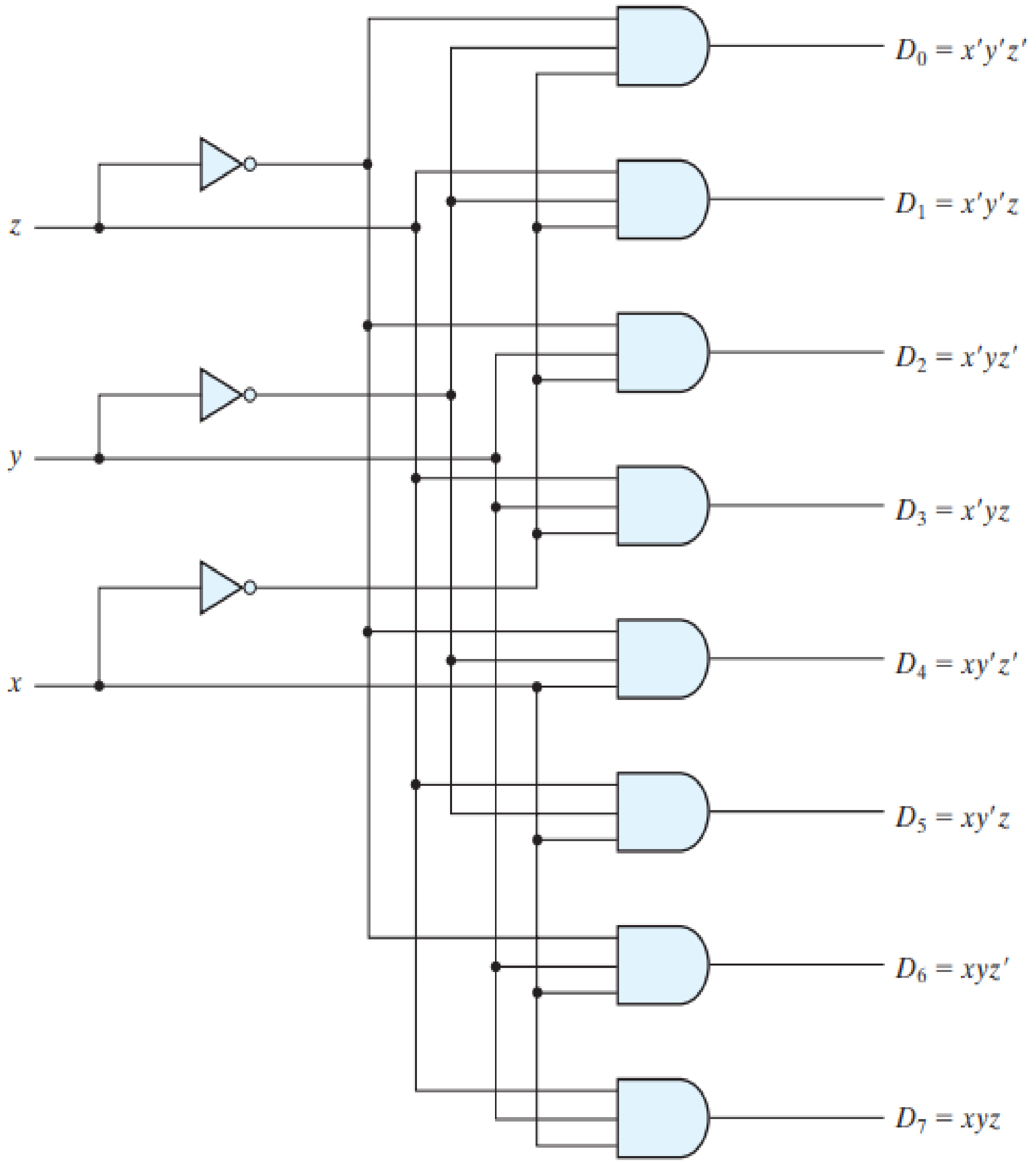
Encoder and Decoder

Decoder:

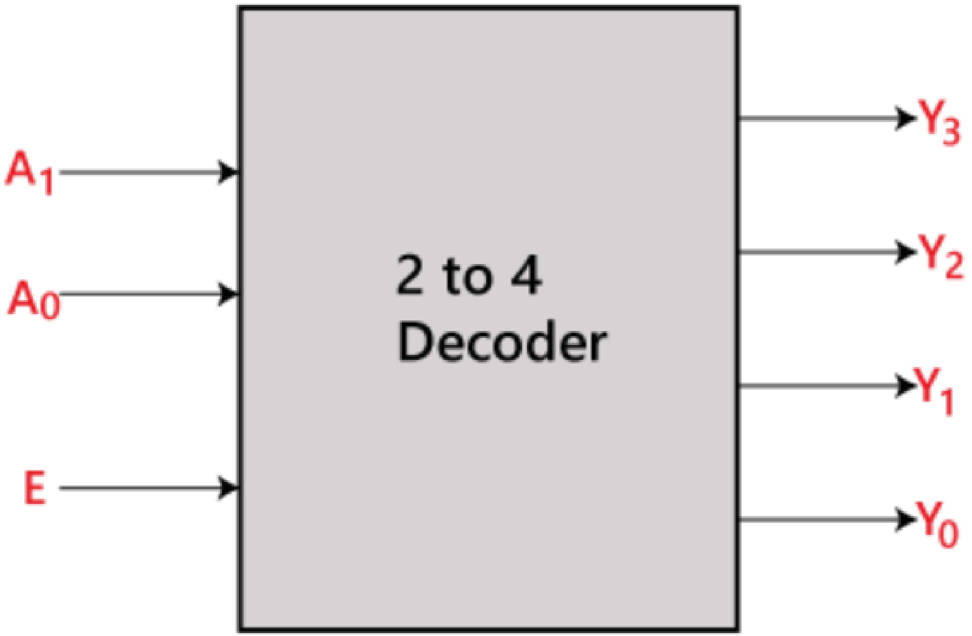
- A decoder is a combinational circuit that converts binary information from n input lines to a maximum of 2^n unique output lines.
- The decoder presented here is the n to m line decoders, where $m \leq 2^n$ .Their purpose is to generate the 2^n minterms of n input variables.
- The outputs of the decoder are nothing but the **minterms** of ‘n’ input variables lines, when it is enabled.

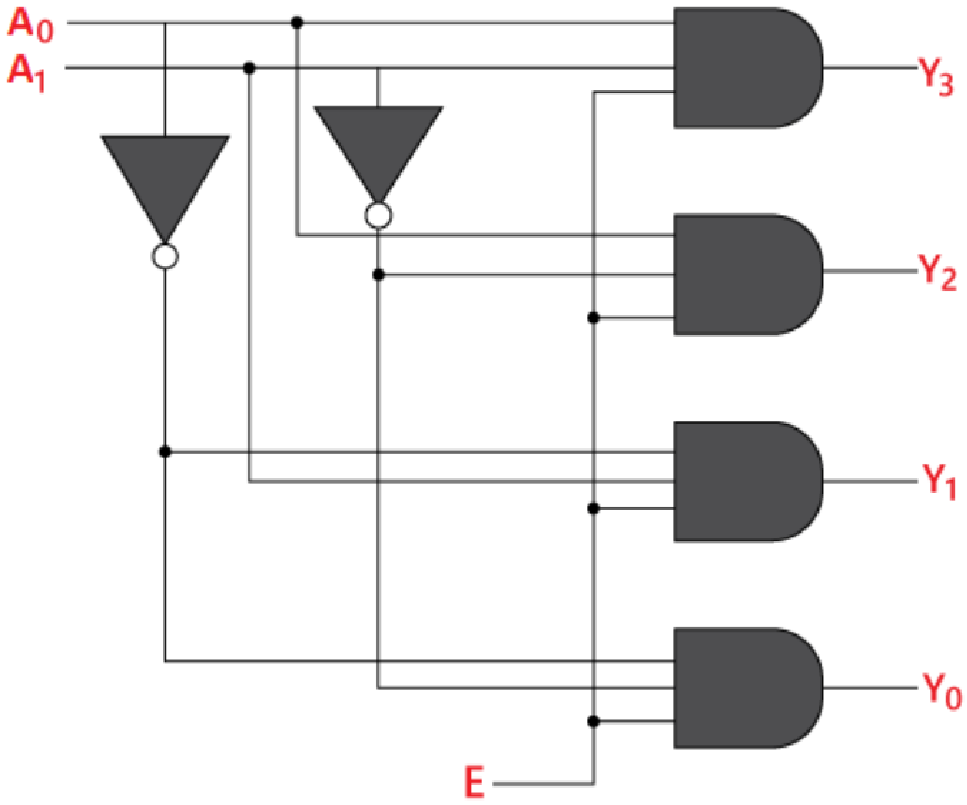


Truth Table of a Three-to-Eight-Line Decoder

Inputs			**Outputs**							
x	y	z	D_0	D_1	D_2	D_3	D_4	D_5	D_6	D_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

2 to 4 line Decoder





E	A0	A1	Y0	Y1	Y2	Y3
1	0	0	1	0	0	0
1	0	1	0	1	0	0
1	1	0	0	0	1	0
1	1	1	0	0	0	1
0	x	x	0	0	0	0

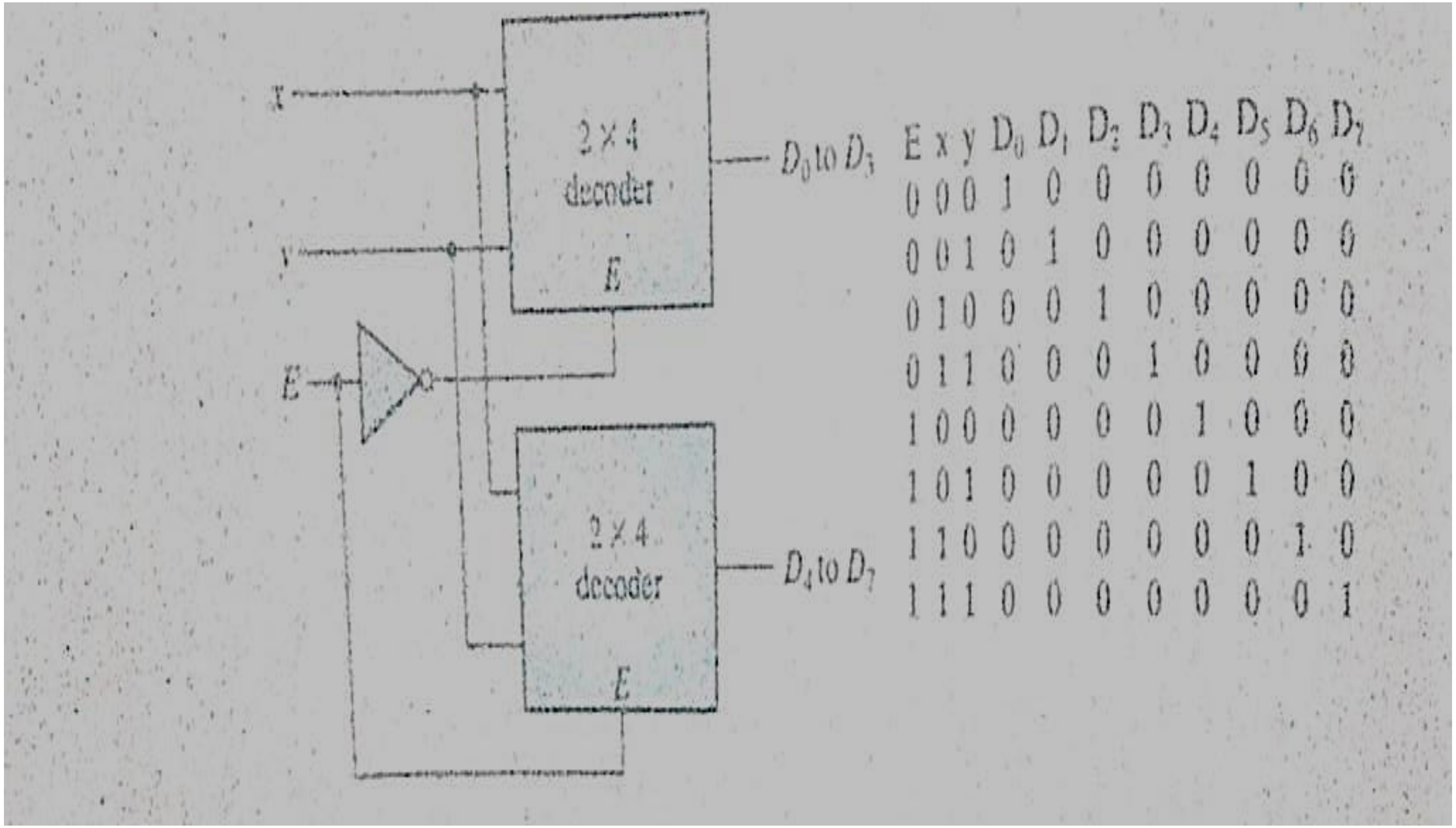
$Y_3=E.A_1.A_0$

$Y_2=E.A_1.A_0'$

$Y_1=E.A_1'.A_0$

$Y0=E.A_1'.A_0'$

3 to 8 decoder using two 2x4 decoder:

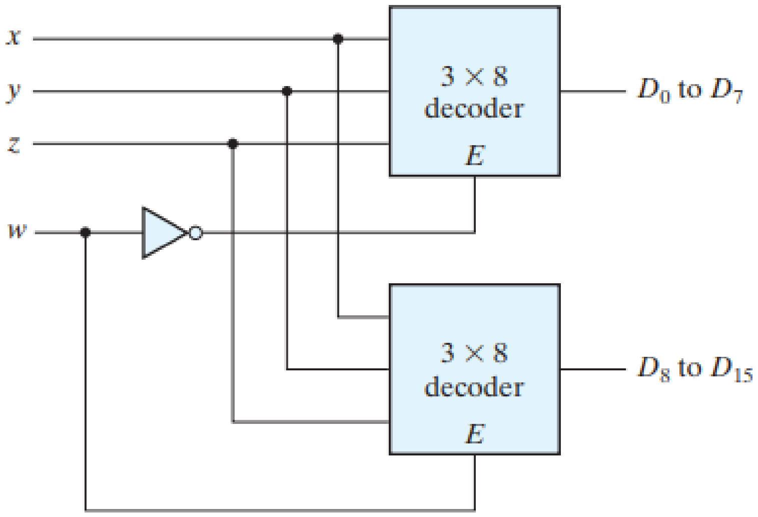


E	x	y	D_0	D_1	D_2	D_3	D_4	D_5	D_6	D_7
0	0	0	1	0	0	0	0	0	0	0
0	0	1	0	1	0	0	0	0	0	0
0	1	0	0	0	1	0	0	0	0	0
0	1	1	0	0	0	1	0	0	0	0
1	0	0	0	0	0	0	1	0	0	0
1	0	1	0	0	0	0	0	1	0	0
1	1	0	0	0	0	0	0	0	1	0
1	1	1	0	0	0	0	0	0	0	1

4 to 16 Line decoder using Two 3x8 decoder:

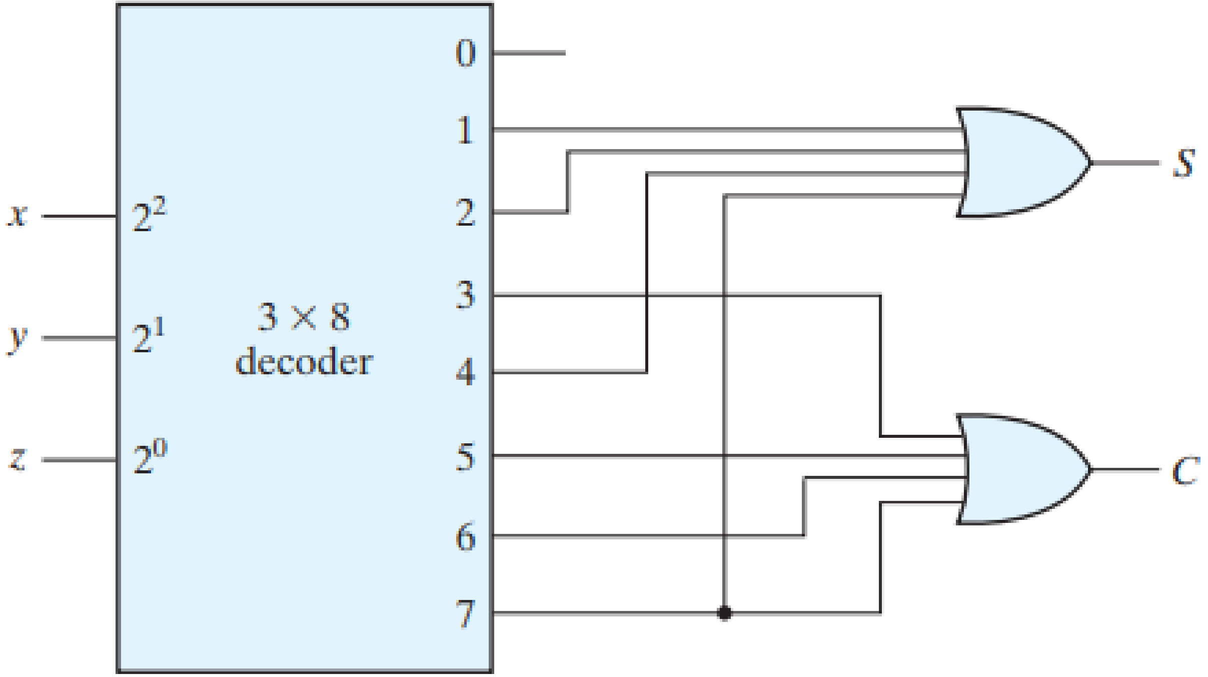
- Decoder with enable input can be connected together to form a larger decoder circuit. The following fig shows two 3 to 8 line decoder with enable input connected to form a 4-16 line decoder.
- When w=0,top decoder enabled and other is disabled.the bottom decoder output all zeros,and top 8 output generate the minterm 0000 to 0111.
- When w=1,bottom decoder generate the output miterms 1000 to 1111,while ouput of top decoder all zero.
- This shows that the usefulness of enable input in combinational circuits.In general enable inputs are the convenient feature for interconnecting two or more standard component.

4 to 16 line Decoder Using three 3x8 decoder:



E	x	y	z	D0	D1	D2	D3	D4	D5	D6	D7	D8	D9	D10	D11	D12	D13	D14	D15
0	0	0	0	1	0		0	0	0	0	0	0	0	0	0		0	0	0
0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	1	0	0	0	1	0	0	0	0	0	0		0	0	0	0	0	0
0	0	1	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
0	1	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
0	1	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
0	1	1	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0	1	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	0	0
1	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
1	0	1	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
1	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Full adder using 3x8 decoder:



$$S(x, y, z) = \Sigma(1, 2, 4, 7)$$

$$C(x, y, z) = \Sigma(3, 5, 6, 7)$$

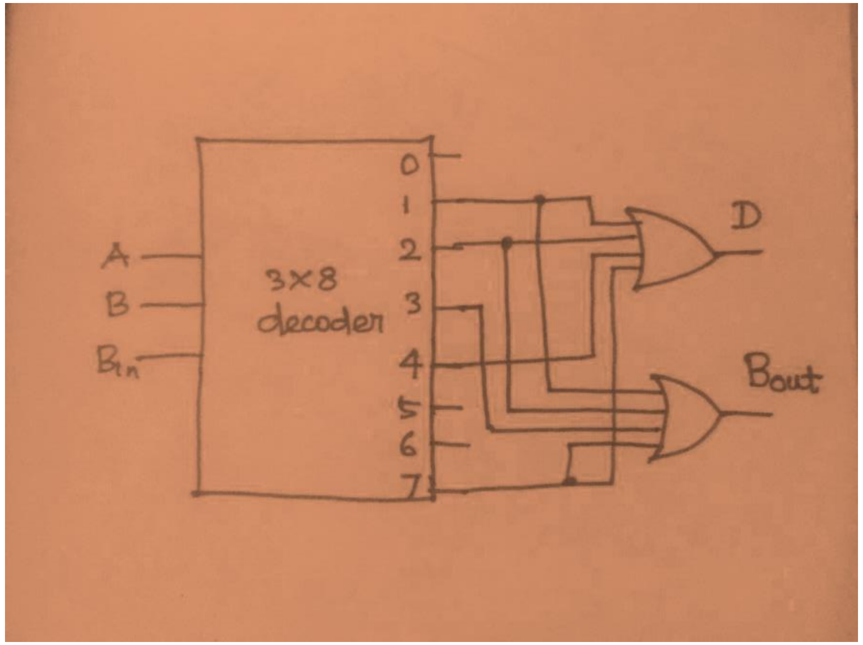
Implement the full subtractor using decoder:

A	B	Bin	D	Bout
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	1
1	0	0	1	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

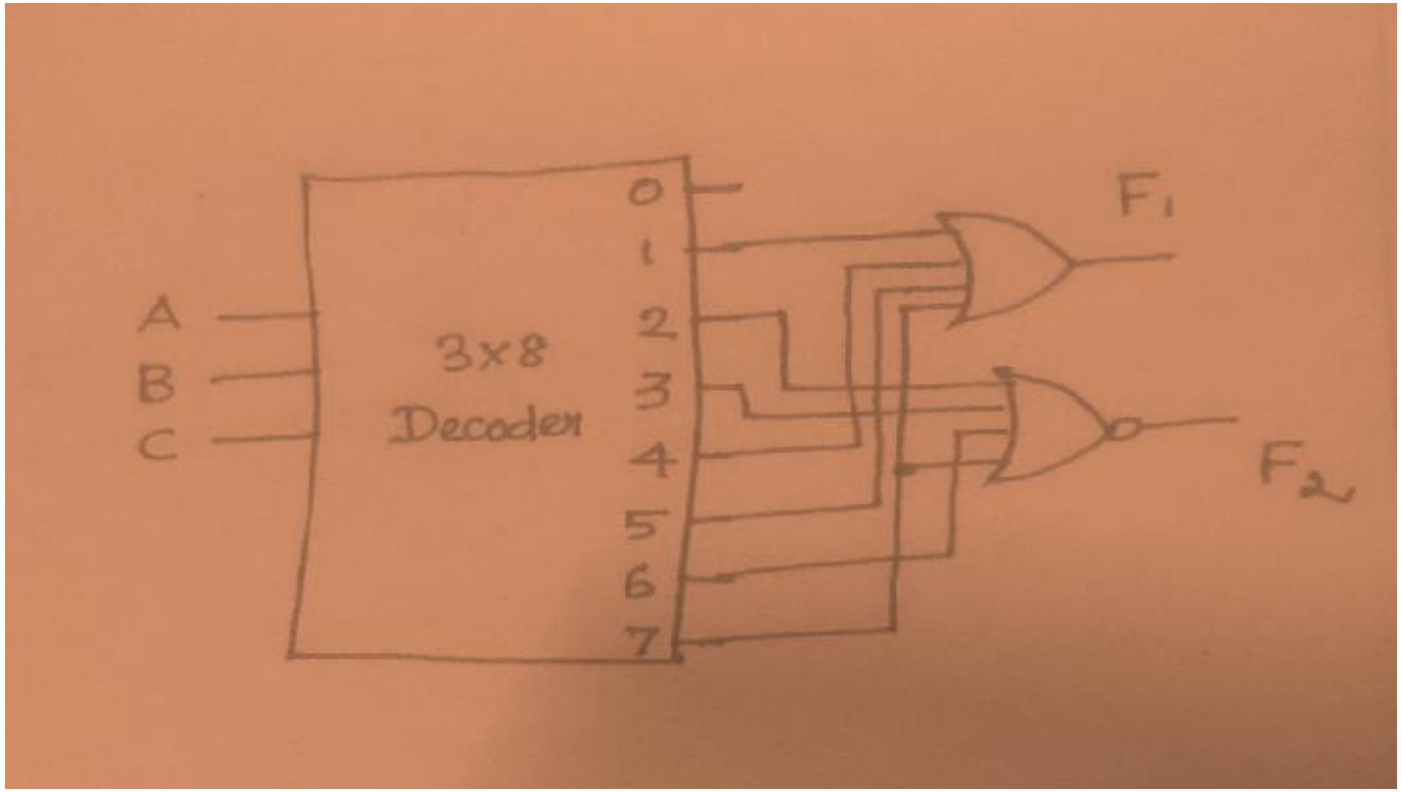
From the truth table

D=$\sum_m(1,2,4,7)$

Bout= $\sum_m(1,2,3,7)$

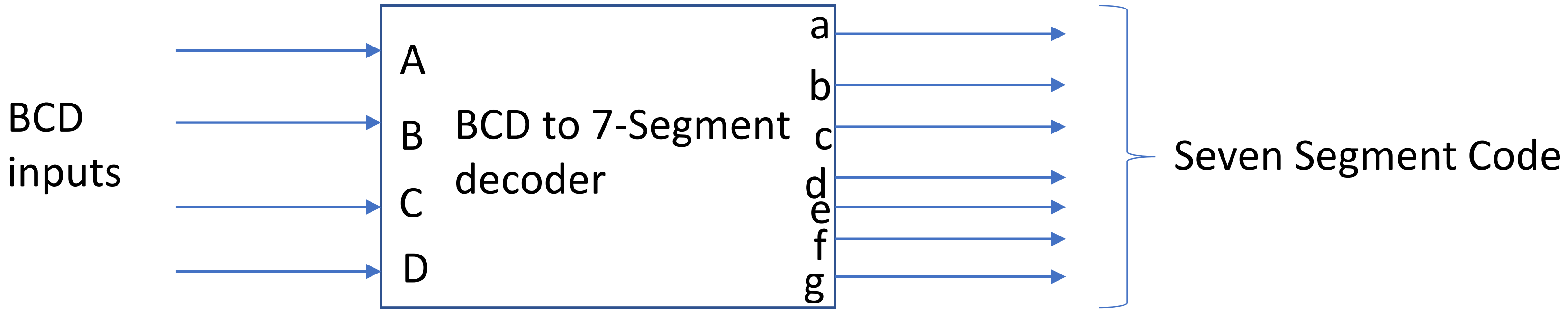


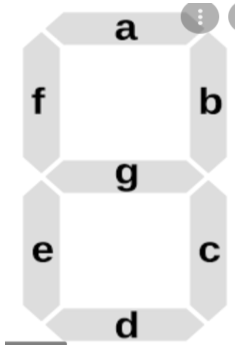
1.Implement the multiple output function using decoder ,F1(A,B,C)=$\sum_m(1,4,5,7)$,F2(A,B,C)=$\sum_{\prod_M}(2,3,6,7)$



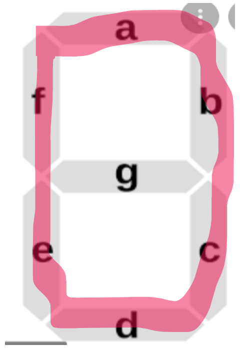
BCD to seven segment Decoder:

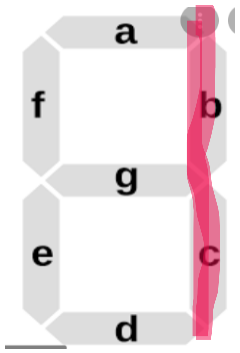
- A block diagram of BCD to seven segment decoder with four inputs (A,B,C,D) and seven outputs (a,b,c,d,e,f,g),corresponding seven segment of a display.

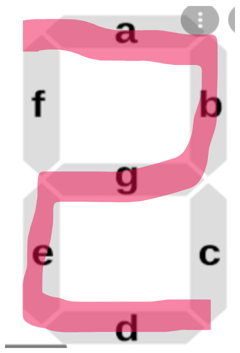


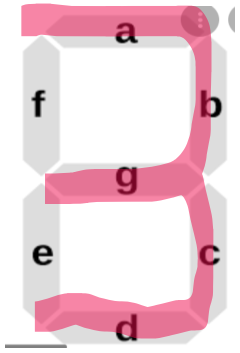


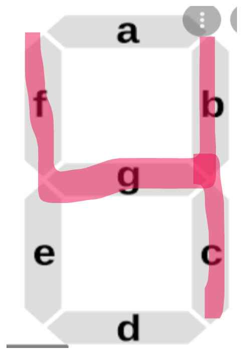
Seven Segment Display

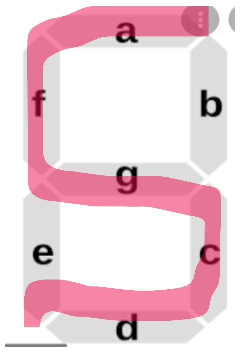

a
f
b
g
e
c
d

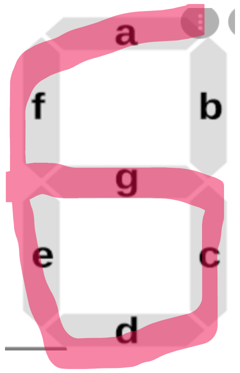

a
f
b
g
e
c
d

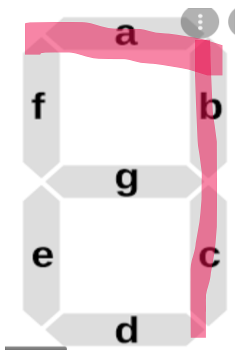

a
f
b
g
e
c
d

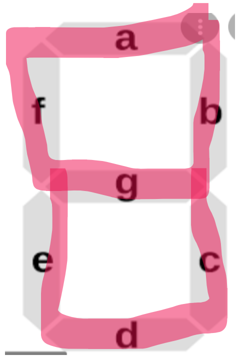

a
f
b
g
e
c
d

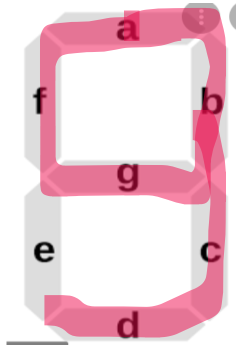

a
f
b
g
e
c
d


a
f
b
g
e
c
d


a
f
b
g
e
c
d


a
f
b
g
e
c
d


a
f
b
g
e
c
d


a
f
b
g
e
c
d

From the above truth table minterms of outputs are,

a $=\sum_{m}(0,2.3,5,6,7,8,9)+\sum_{d}(10,11,12,13,14,15)$

b$=\sum_{m}(0,1,2,3,4,7,8,9)+\sum_{d}(10,11,12,13,14,15)$

c$=\sum_{m}(0,1,3,4,5,6,7,8,9)+\sum_{d}(10,11,12,13,14,15)$

d$=\sum_{m}(0,2.3,5,6,8,9)+\sum_{d}(10,11,12,13,14,15)$

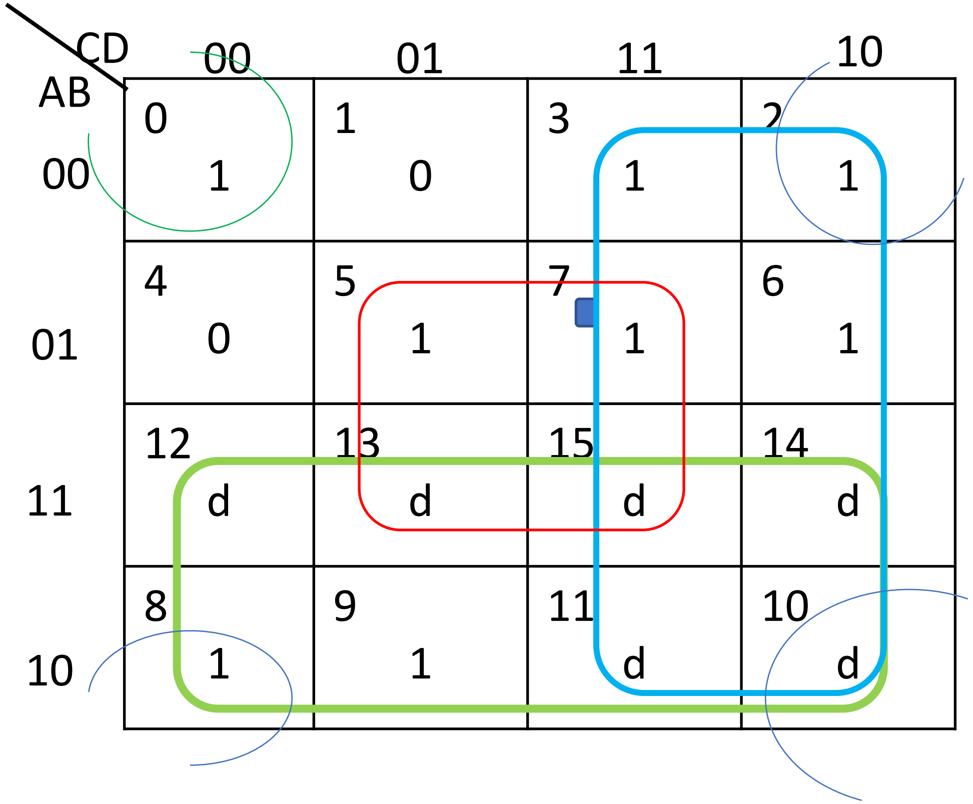
e$=\sum_{m}(0,2,6,8)+\sum_{d}(10,11,12,13,14,15)$

f $=\sum_{m}(0,4,5,6,8,9)+\sum_{d}(10,11,12,13,14,15)$

g$=\sum_{m}(2,3,4,5,6,8,9)+\sum_{d}(10,11,12,13,14,15)$

-	BCD INPUT				Seven Segment Output						
-	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 for a



AB \ CD	00	01	11	10
00	0 1	1 0	3 1	2 1
01	4 0	5 1	7 1	6 1
11	12 d	13 d	15 d	14 d
10	8 1	9 1	11 d	10 d

$a = C+A+BD+B'D' = A+C+(B\oplus D)'$

similarly,

$b = B'+CD+C'D' = B'+(C\oplus D)'$

$c = B+C'+D$

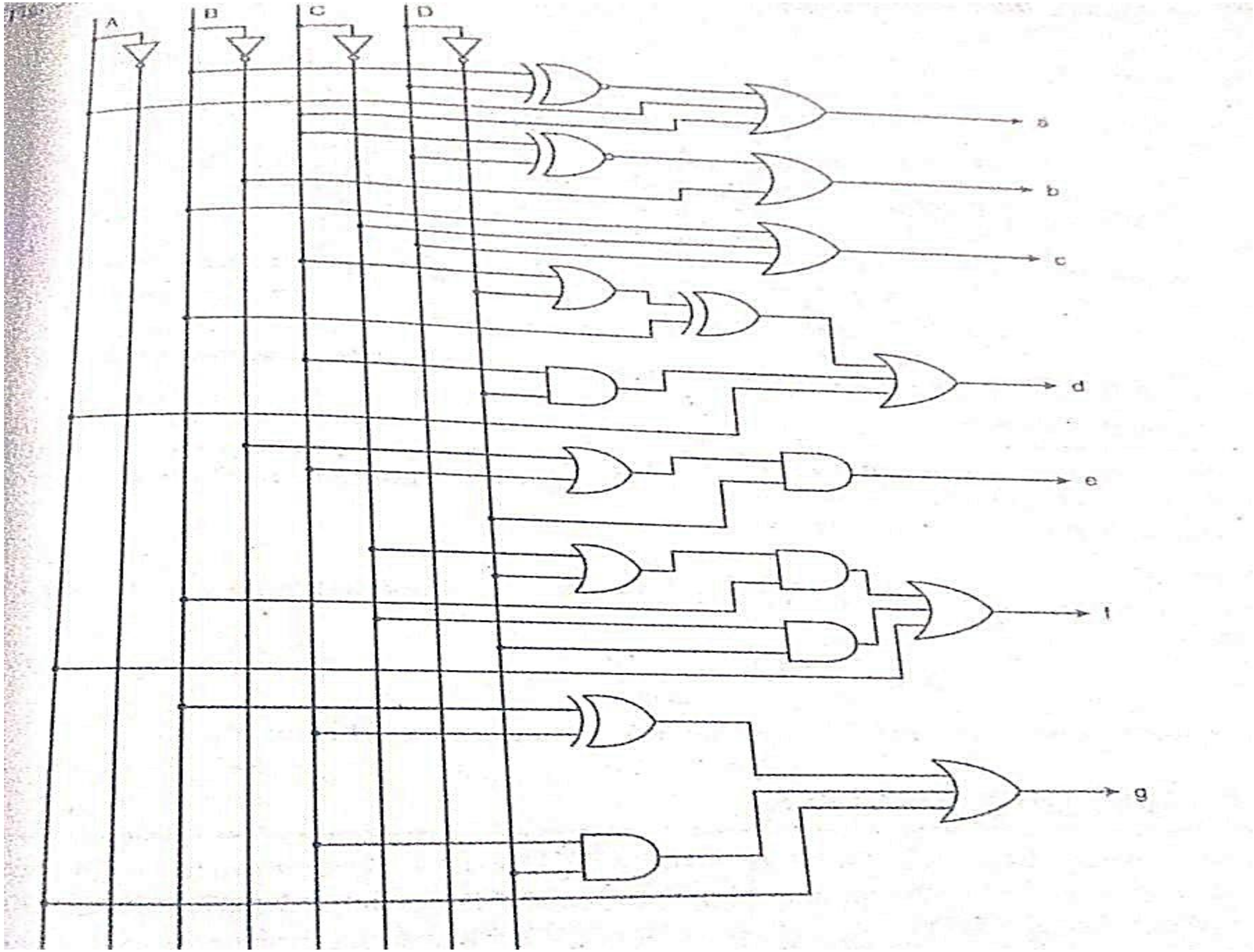
$d = A+B'D'+CD'+B'C+BC'D$

$e = B'D'+CD'$

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

$g = A+CD'+BC'+B'C = A+CD'+B\oplus C$

Logic Diagram of BCD to 7 segment Decoder:



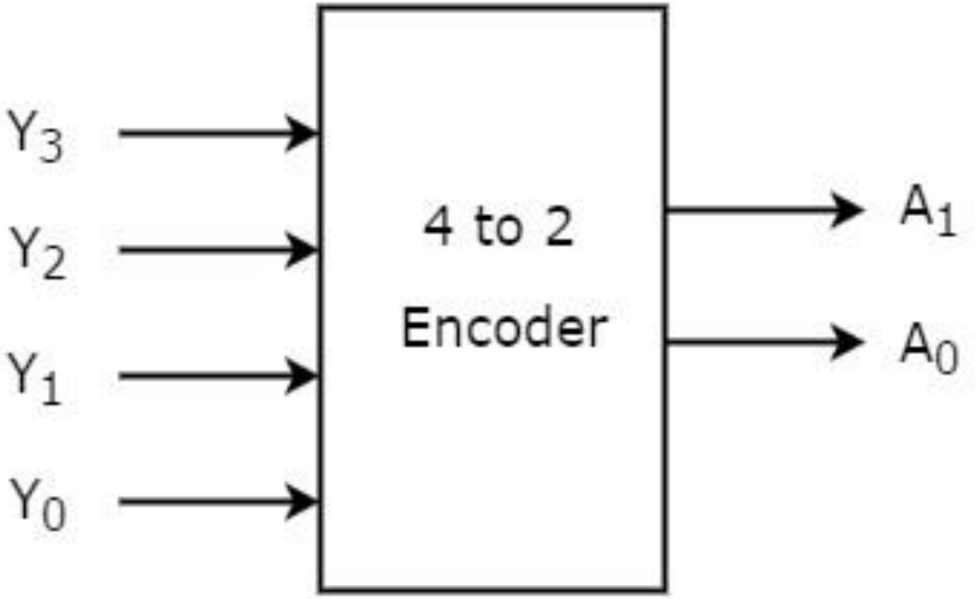
Applications of decoder:

- It is used in a counter systems
- They are used in a analog- to- digital converter
- Decoder output can be used to drive the display systems.

Encoder

- An **Encoder** is a combinational circuit that performs the reverse operation of Decoder. It has maximum of 2^n input lines and 'n' output lines. It will produce a binary code equivalent to the input, which is active High. Therefore, the encoder encodes 2^n input lines with 'n' bits.

4 x 2 Encoder



4 x 2 Encoder Truth table

- At any time, only one of these 4 inputs can be ‘1’ in order to get the respective binary code at the output.

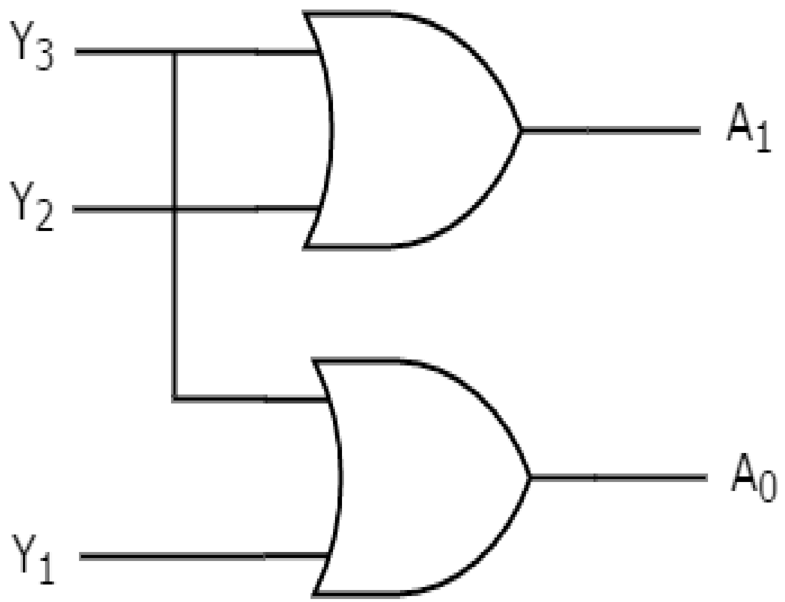
Inputs				Outputs	
Y3	**Y2**	**Y1**	**Y0**	**A1**	**A0**
0	0	0	1	0	0
0	0	1	0	0	1
0	1	0	0	1	0
1	0	0	0	1	1

A1=Y3+Y2

A0=Y3+Y1

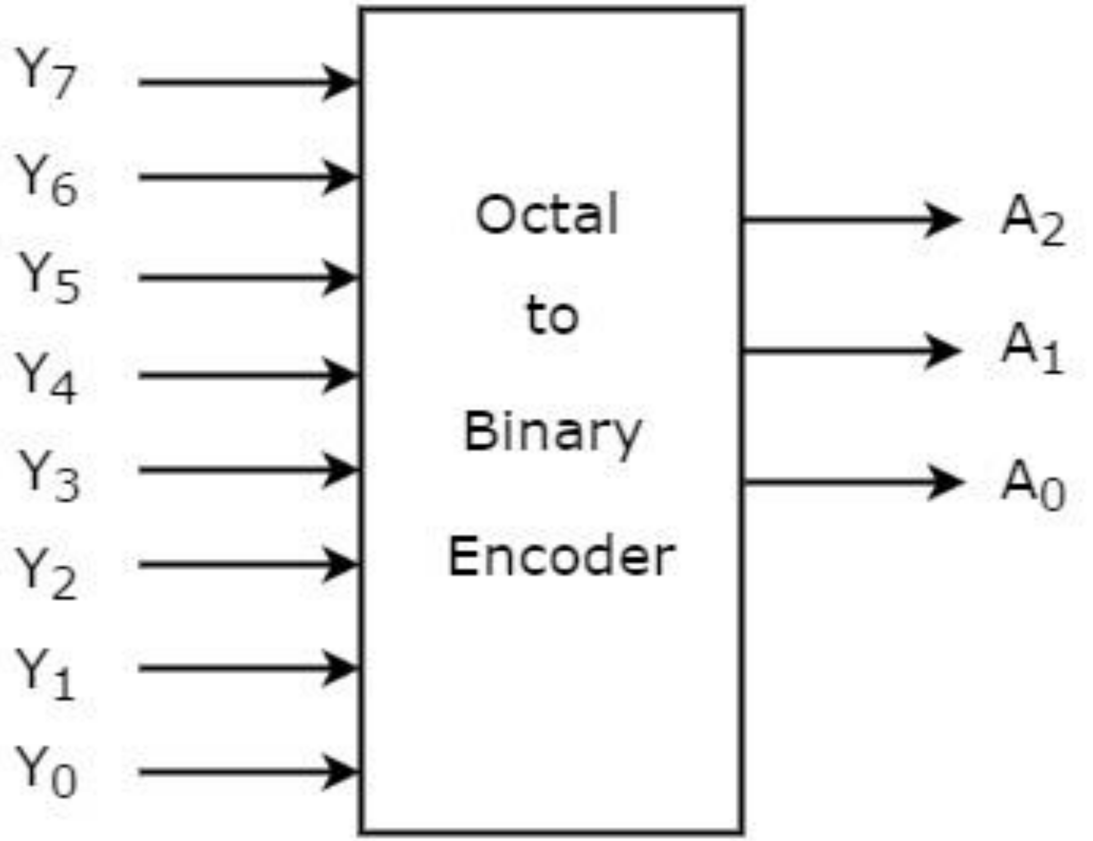
4 x 2 Encoder logic diagram

The **circuit diagram** of 4 to 2 encoder is



Octal to Binary Encoder(8 to 3 line encoder)

- Octal to binary Encoder has eight inputs, Y_7 to Y_0 and three outputs A_2, A_1 & A_0. Octal to binary encoder is nothing but 8 to 3 encoder.



Octal to Binary Encoder Truth table

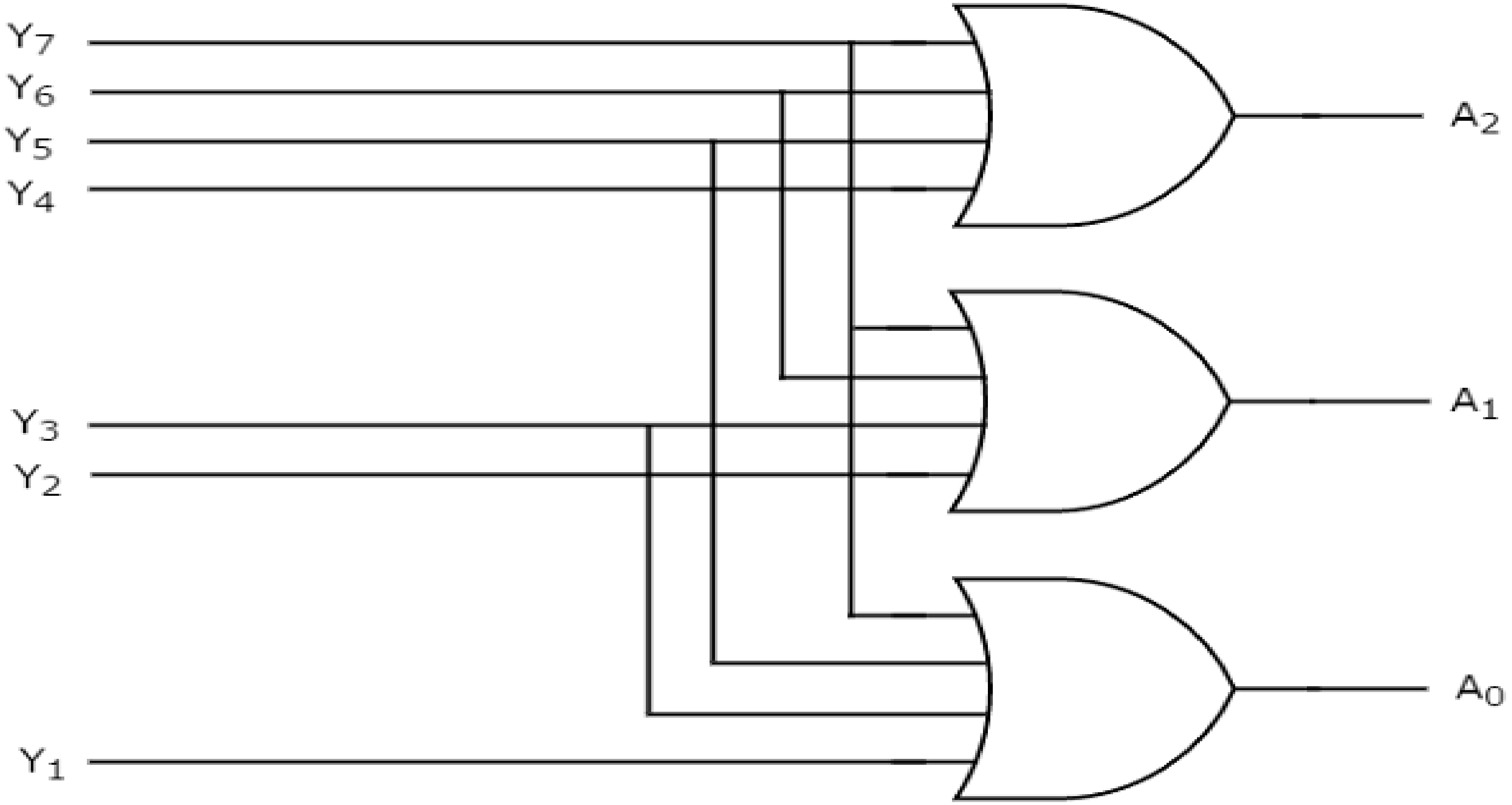
Inputs								Outputs		
Y_7	Y_6	Y_5	Y_4	Y_3	Y_2	Y_1	Y_0	A_2	A_1	A_0
0	0	0	0	0	0	0	1	0	0	0
0	0	0	0	0	0	1	0	0	0	1
0	0	0	0	0	1	0	0	0	1	0
0	0	0	0	1	0	0	0	0	1	1
0	0	0	1	0	0	0	0	1	0	0
0	0	1	0	0	0	0	0	1	0	1
0	1	0	0	0	0	0	0	1	1	0
1	0	0	0	0	0	0	0	1	1	1

Octal to Binary Encoder logic expression and diagram

$A_2=Y_7+Y_6+Y_5+Y_4$

$A_1=Y_7+Y_6+Y_3+Y_2$

$A_0=Y_7+Y_5+Y_3+Y_1$



Drawbacks of Encoder

- There is an ambiguity, when all outputs of encoder are equal to zero. Because, it could be the code corresponding to the inputs, when only least significant input is one or when all inputs are zero.
- If more than one input is active High, then the encoder produces an output, which may not be the correct code. For example, if both Y_3 and Y_6 are ‘1’, then the encoder produces 111 at the output. This is neither equivalent code corresponding to Y_3, when it is ‘1’ nor the equivalent code corresponding to Y_6, when it is ‘1’.
- So, to overcome these difficulties, we should assign priorities to each input of encoder. Then, the output of encoder will be the binary code corresponding to the active High inputs, which has higher priority. This encoder is called as **priority encoder.**

Priority Encoder

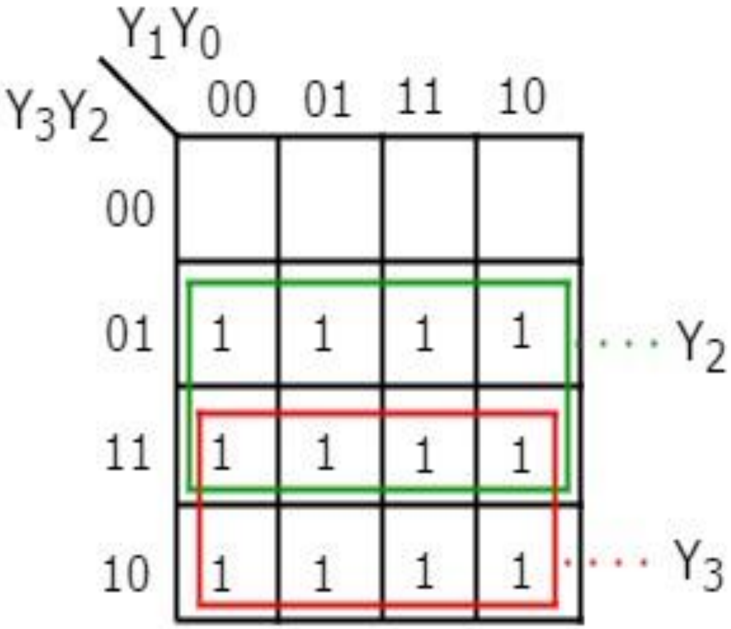
- A 4 to 2 priority encoder has four inputs Y3, Y2, Y1 & Y0 and two outputs A1 & A0. Here, the input, Y3 has the highest priority, whereas the input, Y0 has the lowest priority. In this case, even if more than one input is ‘1’ at the same time, the output will be the binary code corresponding to the input, which is having higher priority.
- We considered one more output, V in order to know, whether the code available at outputs is valid or not.
- If at least one input of the encoder is ‘1’, then the code available at outputs is a valid one. In this case, the output, V will be equal to 1.
- If all the inputs of encoder are ‘0’, then the code available at outputs is not a valid one. In this case, the output, V will be equal to 0.

INPUTS				OUTPUTS		
Y3(HIGH)	Y2	Y1	Y0(LOW)	A1	A0	V
0	0	0	0	x	x	0
0	0	0	1	0	0	1
0	0	1	0	0	1	1
0	0	1	1	0	1	1
0	1	0	0	1	0	1
0	1	0	1	1	0	1
0	1	1	0	1	0	1
0	1	1	1	1	0	1
1	0	0	0	1	1	1
1	0	0	1	1	1	1
1	0	1	0	1	1	1
1	0	1	1	1	1	1
1	1	0	0	1	1	1
1	1	0	1	1	1	1
1	1	1	0	1	1	1
1	1	1	1	1	1	1

For Ex..Y3=Y2=Y1=Y0=0 indicates that no priority is assigned .Hence V=0

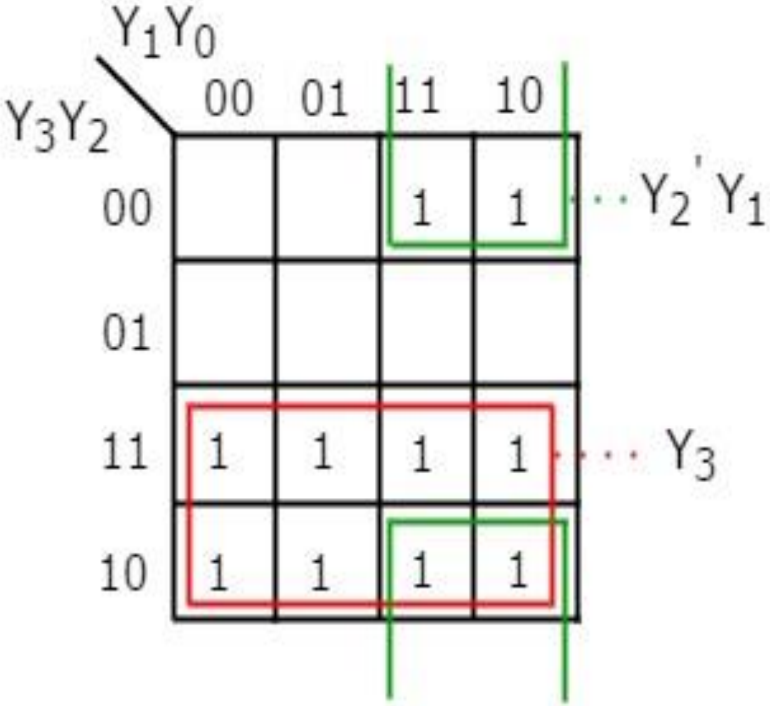
If V=1 Priority is assigned to any one of input for Ex.. Y3=Y2=Y0=0 and Y1=1, it indicates priority assigned to Y1 hence A1A0=01 and V=1

K-Map for A_1



Y_3Y_2 \ Y_1Y_0	00	01	11	10
00				
01	1	1	1	1
11	1	1	1	1
10	1	1	1	1

K-Map for A_0



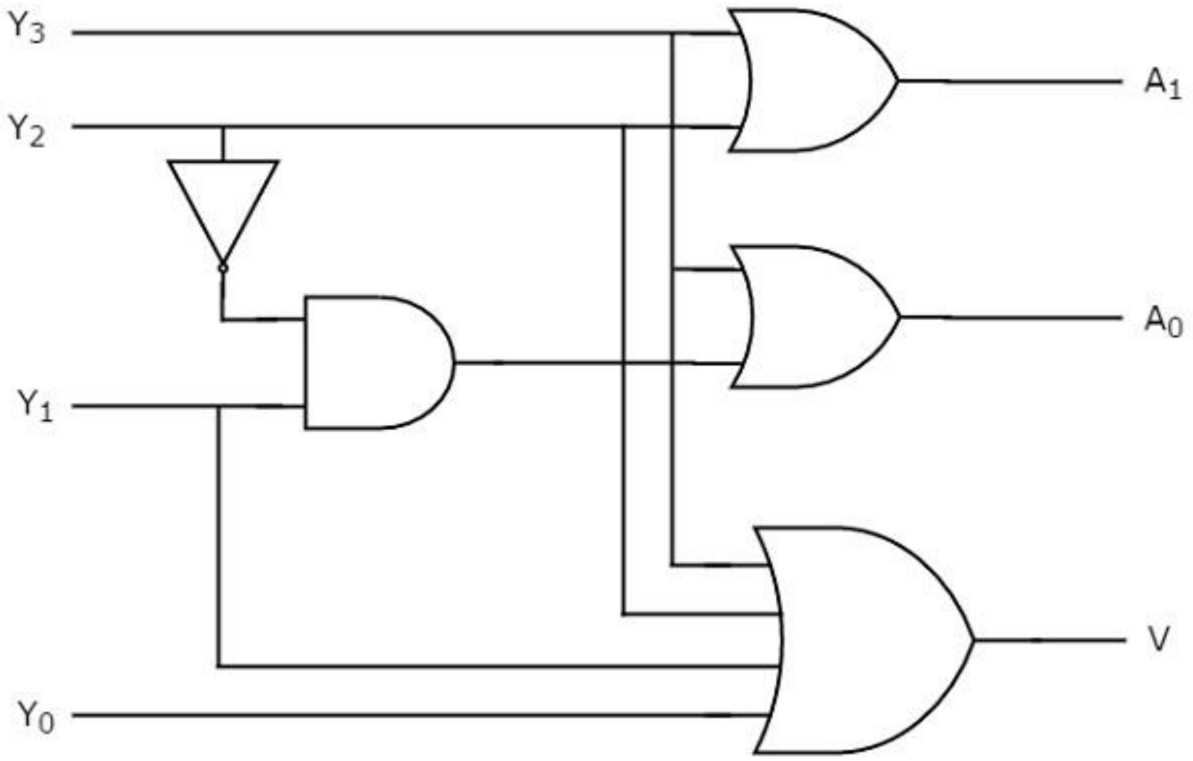
Y_3Y_2 \ Y_1Y_0	00	01	11	10
00			1	1
01				
11	1	1	1	1
10	1	1	1	1

$A_1=Y_3+Y_2$

$A_0=Y3+Y2'Y1$

$V=Y_3+Y_2+Y_1+Y_0$

Priority Encoder circuit diagram


Y_3
Y_2
Y_1
Y_0
A_1
A_0
V