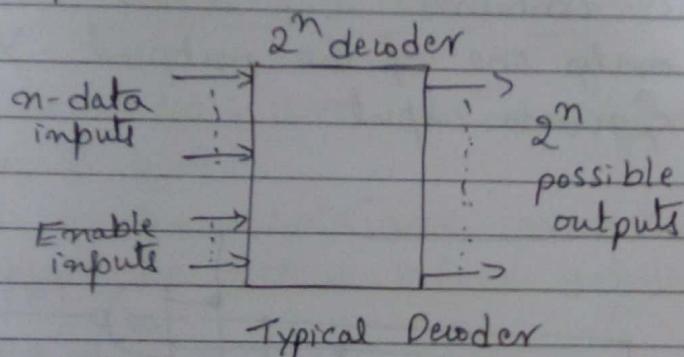


- Decoders :- Decoders are a class of combinational logic circuits that convert a set of input variables representing a code into a set of output variables representing a different code.
- Encoded information is presented as 'n' ip's producing  $2^n$  outputs.
  - The  $2^n$  output values can range from 0 to  $2^n - 1$



- 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.
- Decoder is a device which when activated selects one of its possible  $2^n$  outputs based on its m-bits.

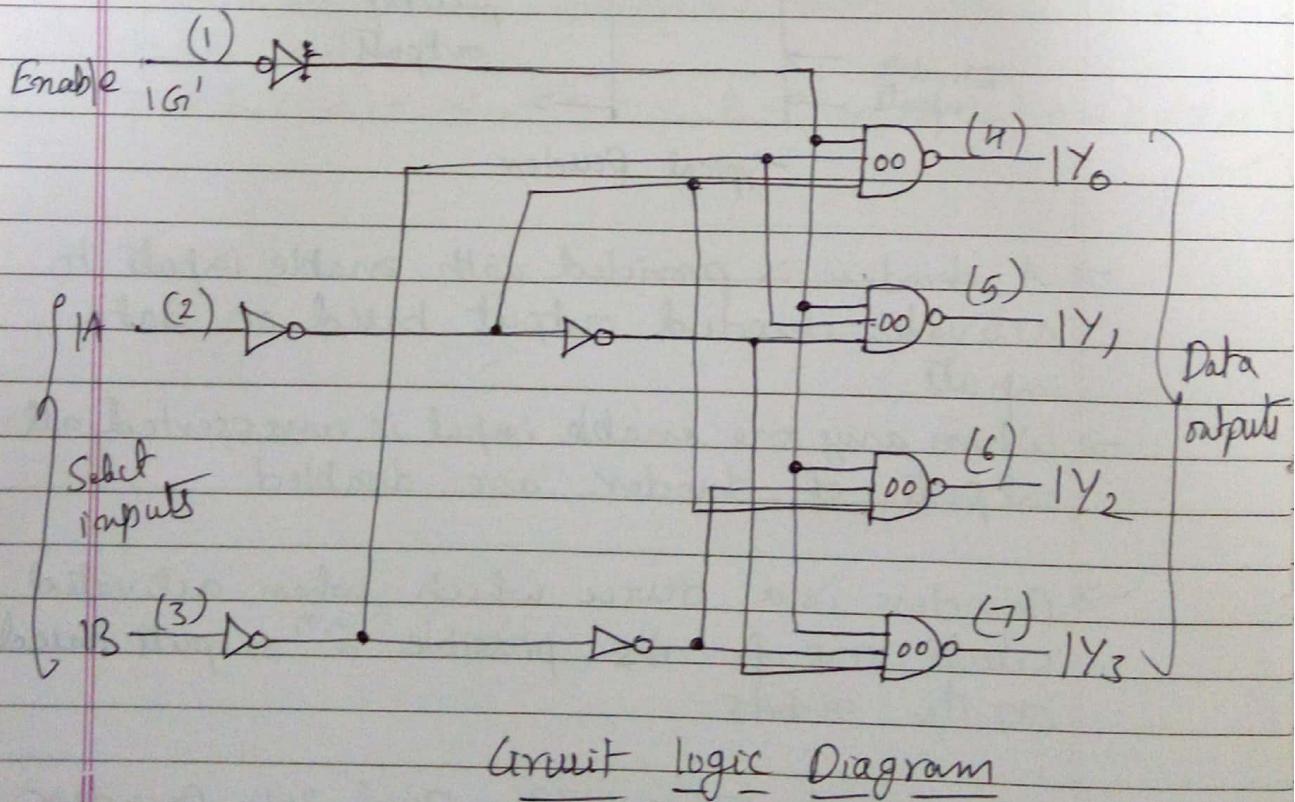
### The Dual 74XX139 Dual 2-1<sub>0</sub> Decoder



IEEE logic symbol

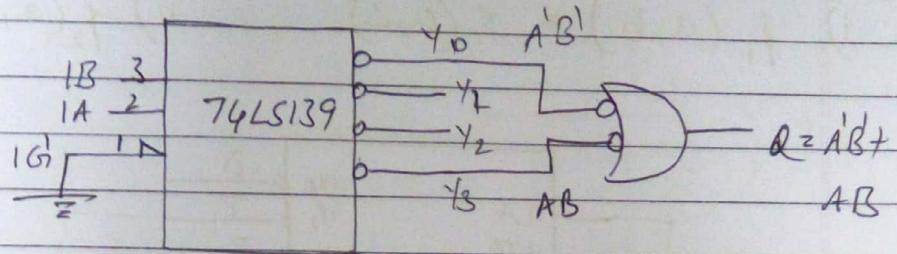
Inputs			Outputs				Function Table.
Enable	Select		$Y_0$	$Y_1$	$Y_2$	$Y_3$	
H	X	X	H	H	H	H	
L	L	L	L	H	H	H	
L	L	H	H	L	H	H	
L	H	L	H	H	L	H	
L	H	H	H	H	H	L	

→ The 74xx139 contains 2, 2 to 4 decoders  
 → One, and only one of the output  $Y_0$  to  $Y_3$  is active for a given input



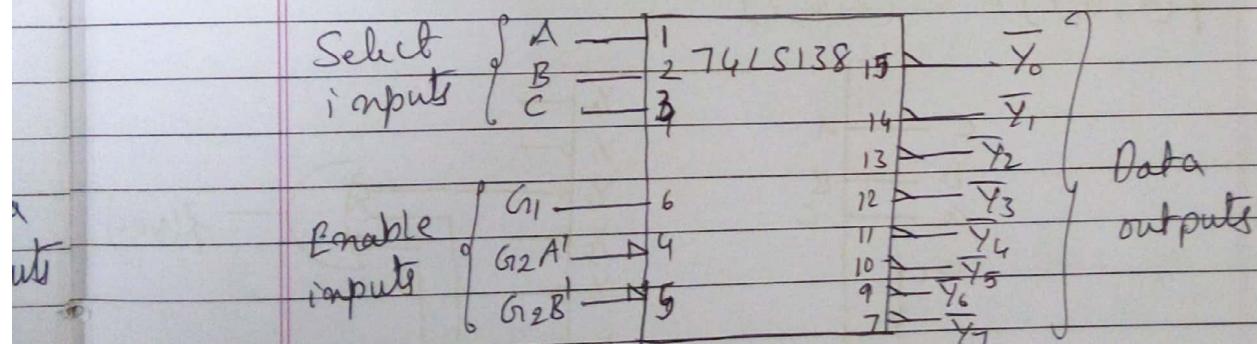
- Decoders can be used as a minterm or maxterm generators
- Eg:- 2 to 4 decoder can be used to generate minterms/maxterms (0, 1, 2, 3)

1. Design  $Q = P(a, b) = \Sigma(0, 3)$  using 74LS139



### The 74XX138 3-to-8 Decoder

- The 3 i/p's are decoded to produce one of the 8 o/p's.
- 3 enable i/p's are provided, all of which must be active before decoding can occur



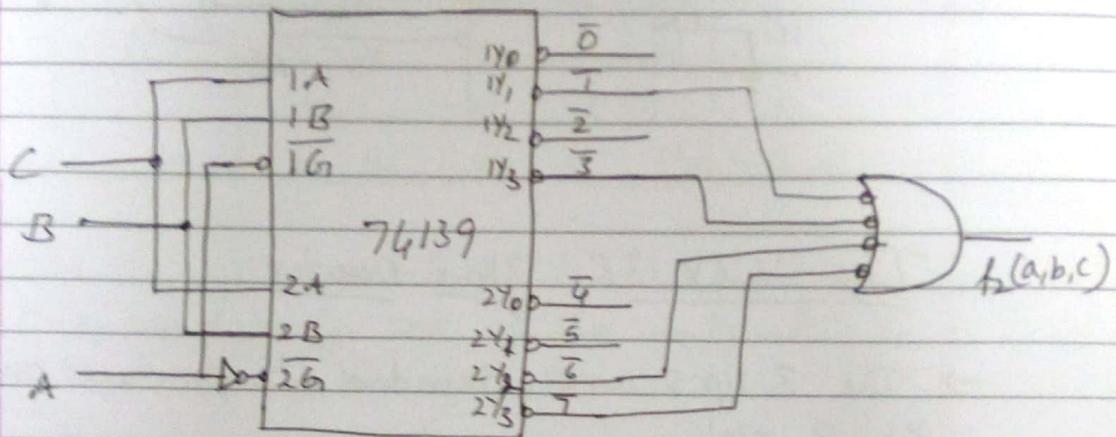
### Inputs

	$G_1$	$G_2 A'$	$G_2 B'$	$C$	$B$	$A$	$Y_0$	$Y_1$	$Y_2$	$Y_3$	$Y_4$	$Y_5$	$Y_6$	$Y_7$
0	X	X		X	X	X	1	1	1	1	1	1	1	1
1	1	X		X	X	X	1	1	1	1	1	1	1	1
1	X	1		X	X	X	1	1	1	1	1	1	1	1
1	0	0		0	0	0	0	1	1	1	1	1	1	1
1	0	0		0	0	1	0	1	1	0	1	1	1	1
1	0	0		0	1	0	1	1	1	0	1	1	1	1
1	0	0		1	0	0	1	1	1	1	0	1	1	1
1	0	0		1	0	1	1	1	1	1	1	0	1	1
1	0	0		1	1	0	1	1	1	1	1	1	1	0
1	0	0		1	1	1	1	1	1	1	1	1	1	0

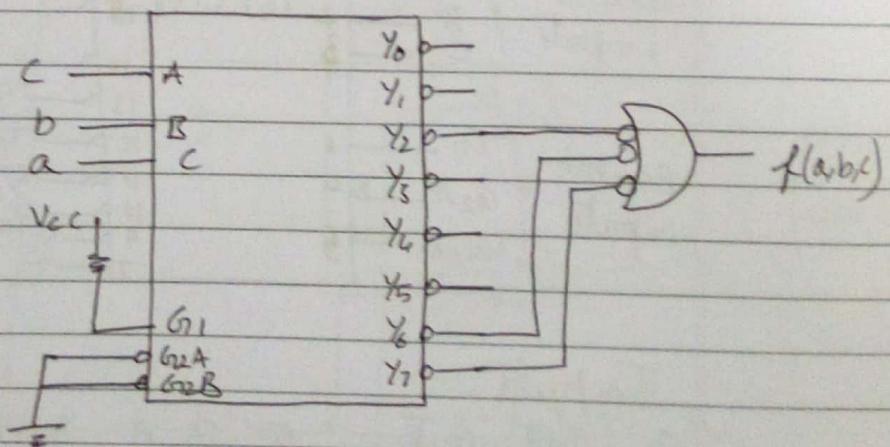
2 Realize the following function using 74139

$$i) f_1(a, b) = \Sigma(0, 2)$$

$$ii) f_2(a, b, c) = \Sigma(1, 3, 6, 7)$$



1 Implement the following function using 74138  
 $f(a, b, c) = \Sigma(2, 6, 7)$

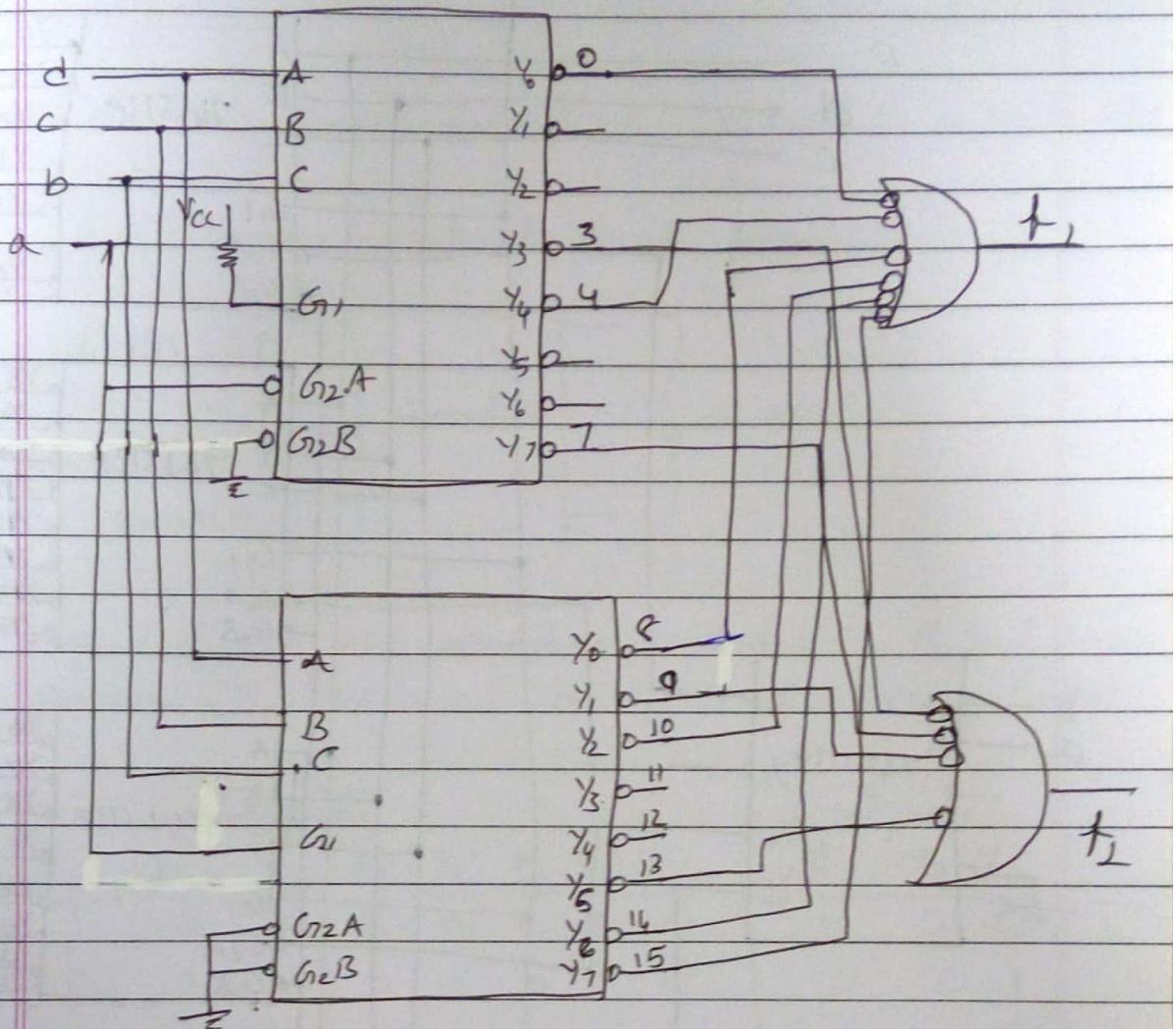


2 Implement the multiple functions !

$$f_1(a, b, c, d) = \Sigma(0, 4, 8, 10, 14, 15)$$

$$f_2(a, b, c, d) = \Sigma(3, 7, 9, 13)$$

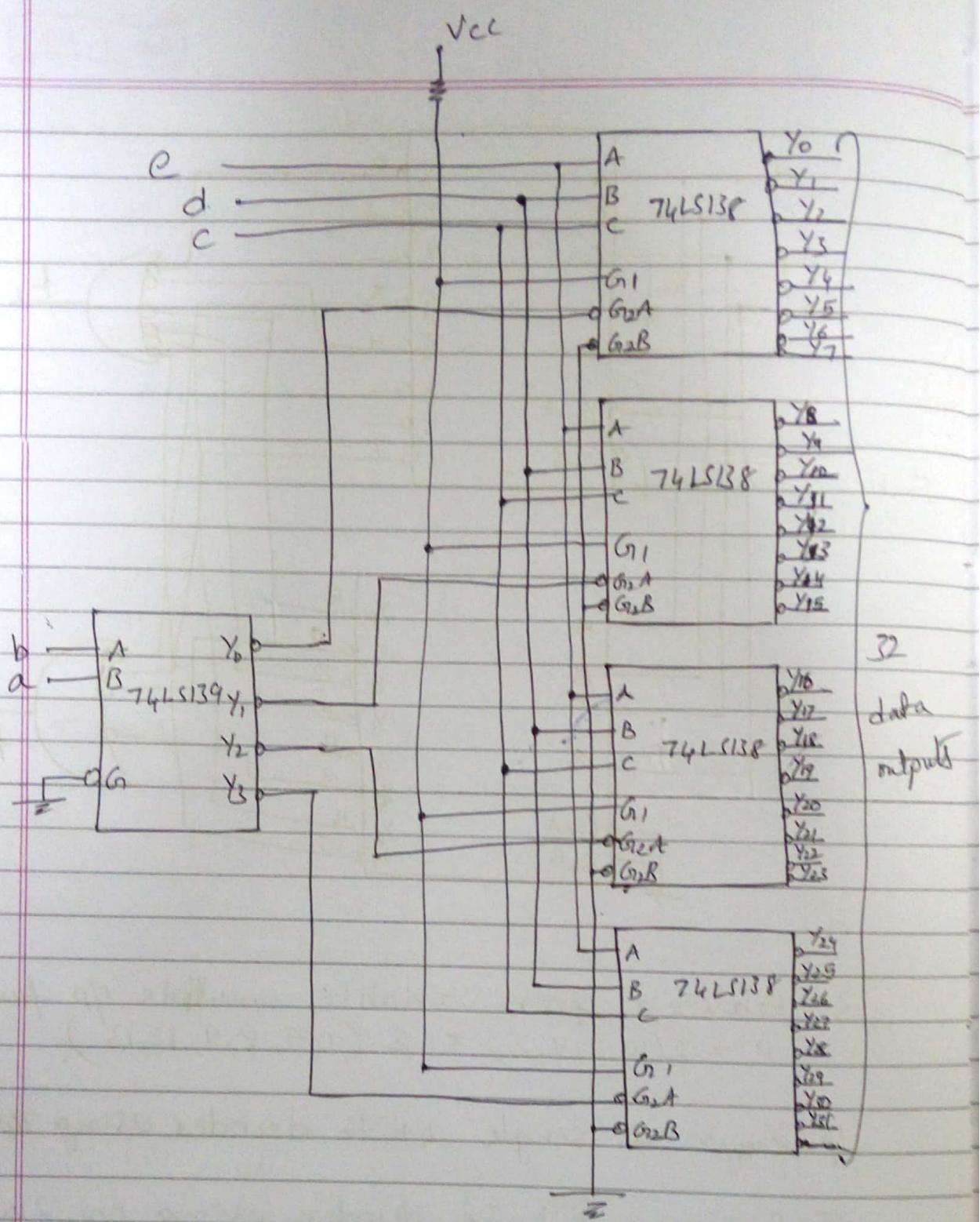
using two 3-to-8 decoders in 74138 IC's.



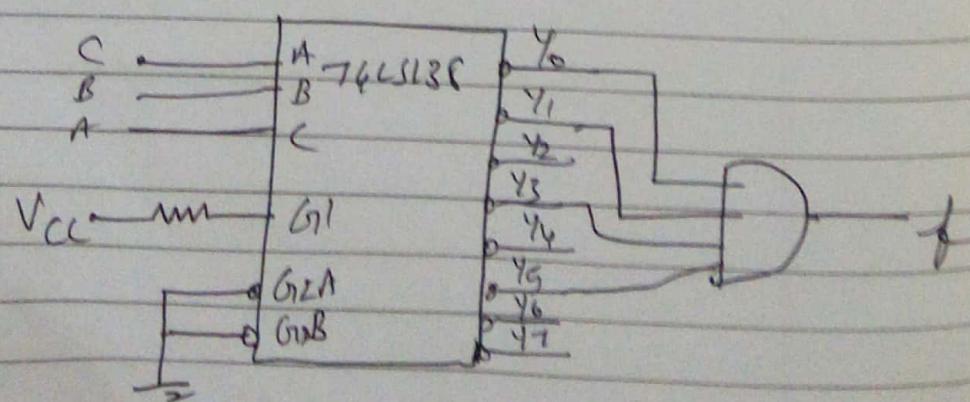
3. Realize a four-variable multiple o/p function  
 $P = f(w, x, y, z) = \Sigma (1, 5, 8, 9, 12, 13)$ .

4 Design a single 4-to-16 decoder using 74x138

5 Design a 5-to-32 decoder using one 2-to-4 decoder & 4 3-to-8 decoder IC's.

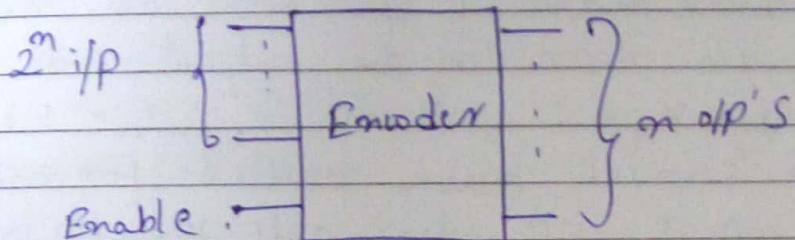


6. Realize POS  $\Pi(0, 1, 3, 5) = f(A, B, C)$  using  
74LS188 IC.



## Encoders

- Encoders have multiple inputs & outputs like decoders
- An encoder produces n-o/p from  $2^n$  i/p's



Eg: 8 to 2 line decoder encoders, 8 to 3 line encoders, decimal to BCD encoders & so on.

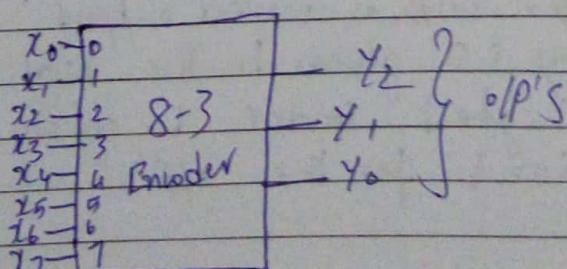
Inputs								Outputs		
$x_0$	$x_1$	$x_2$	$x_3$	$x_4$	$x_5$	$x_6$	$x_7$	$y_2$	$y_1$	$y_0$
1	0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0	1
0	0	1	0	0	0	0	0	0	1	0
0	0	0	1	0	0	0	0	0	1	1
0	0	0	0	1	0	0	0	1	0	0
0	0	0	0	0	1	0	0	1	0	1
0	0	0	0	0	0	1	0	1	1	0
0	0	0	0	0	0	0	1	1	1	1

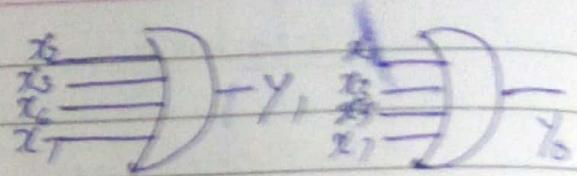
The boolean expression for the outputs are

$$y_2 = x_4 + x_5 + x_6 + x_7$$

$$y_1 = x_2 + x_3 + x_6 + x_7$$

$$y_0 = x_1 + x_3 + x_5 + x_7$$





Drawbacks:-

- When more than 1 i/p's are at logic 1 there may be an error in the output code.  
i.e., if  $x_3 = x_4 = 1$ , then  $Y_2, Y_1, Y_0 = 111$  whereas this op should have resulted for only  $x_7 = 1$ .
- The op is 000 when all the i/p are at logic 0 as well as when  $x_0 = 1$ .

The problem can be overcome by the priority encoder with a validity indicator.

Row No.	Inputs							Outputs		
	$x_0$	$x_1$	$x_2$	$x_3$	$x_4$	$x_5$	$x_6, x_7$	$Y_1$	$Y_0$	$V$
0	0	0	0	0	0	0	0	0	0	0
1	1	0	0	0	0	0	0	0	0	1
2	X	1	0	0	0	0	0	0	1	1
3	X	X	1	0	0	0	0	0	1	1
4	X	X	X	1	0	0	0	0	1	1
5	X	X	X	X	1	0	0	1	0	1
6	X	X	X	X	X	1	0	1	0	1
7	X	X	X	X	X	X	1	0	1	1
8	X	X	X	X	X	X	X	1	1	1

8 to 3 line priority encoder.

- Write the condensed truth table for a 4-to-2 line priority encoder with a valid output where the highest priority is given to the highest bit position or i/p with highest index & obtain the minimal sum expressions for the op's

All No.	$x_0$	$x_1$	$x_2$	$x_3$	$y_1$	$y_0$	Valid
0	0	0	0	0	0	0	0
8	1	0	0	0	0	0	1
4, 12	X	1	0	0	0	1	1
2, 6, 10, 14	X	X	1	0	1	0	1
1, 3, 5, 7, 15	X	X	X	1	1	1	1
9, 11, 13							

K-map for  $Y_1$

$x_0$	$x_1$	$x_2$	$x_3$
00	0	1	1
01	0	1	1
11	0	1	1
10	0	1	1

$$Y_1 = x_2 + x_3$$

K-map for  $Y_0$

$x_0$	$x_1$	$x_2$	$x_3$
00	.	1	1
01	1	1	1
11	1	1	1
10	1	1	1

$$Y_0 = x_3 + \overline{x_1}x_2$$

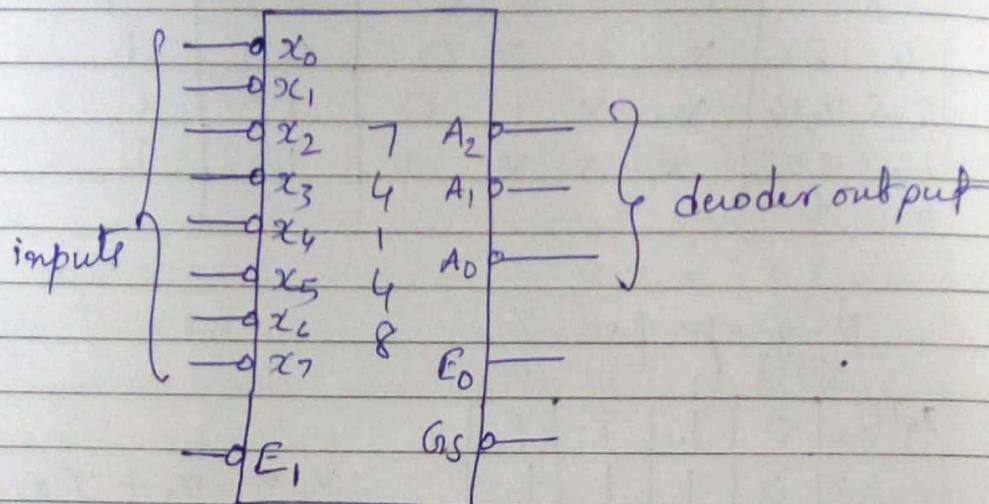
K-map for Valid

$x_0$	$x_1$	$x_2$	$x_3$
00	0	1	1
01	1	1	1
11	1	1	1
10	1	1	1

$$\text{Valid} = x_3 + x_2 + x_1 + x_0$$

2 Assign highest priority to the least significant input or input with lowest index.

74148 IC :- 8 line to 3 line priority encoder.



$E_1$	Input								Output				
	$x_0$	$x_1$	$x_2$	$x_3$	$x_4$	$x_5$	$x_6$	$x_7$	$A_2$	$A_1$	$A_0$	$GS$	$E_0$
1	x	x	x	x	x	x	x	x	1	1	1	1	1
0	1	1	1	1	1	1	1	1	1	1	1	1	0
0	0	1	1	1	1	1	1	1	1	1	1	0	1
0	x	0	1	1	1	1	1	1	1	1	0	0	1
0	x	x	0	1	1	1	1	1	1	0	1	0	1
0	x	x	x	0	1	1	1	1	1	0	0	0	1
0	x	x	x	x	0	1	1	1	0	1	1	0	1
0	x	x	x	x	x	0	1	1	0	0	1	0	1
0	x	x	x	x	x	x	0	1	0	0	0	0	1

$GS \rightarrow$  one of the ifp is activated.

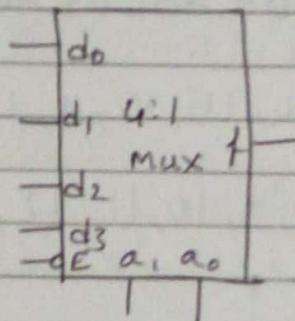
1. Design a priority encoder for a system with 3 inputs with the middle bit with highest priority encoding to 10, the MSB with the next priority to encoding to 11, while the LSB with the least priority encoding to 01.

A	B	C	Y	Z
0	0	0	00	
0	0	1	01	
0	1	0	10	$Y = A + B$
0	1	1	10	
1	0	0	11	$Z = A\bar{B} + \bar{B}C$
1	0	1	11	
1	1	0	10	
1	1	1	10	

## Multiplexer :-

- A digital multiplexer connects one of  $n$  inputs to a single output line.
- The one of  $n$  input selection is determined by  $m$  select inputs, where  $n = 2^m$ .

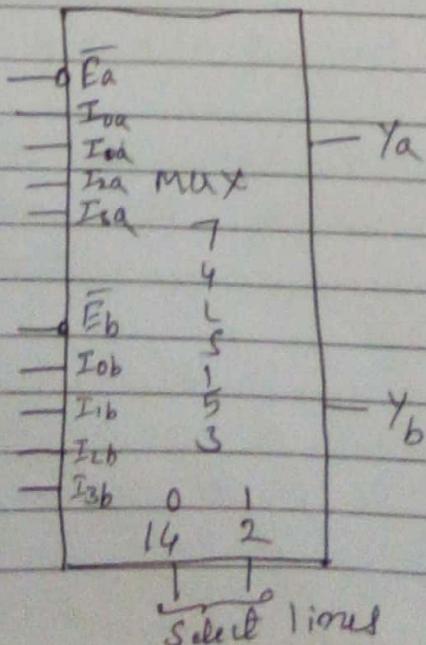
### 4:1 Mux



E	a <sub>1</sub>	a <sub>0</sub>	f
1	X	X	0
0	0	0	d <sub>0</sub>
0	0	1	d <sub>1</sub>
0	1	0	d <sub>2</sub>
0	1	1	d <sub>3</sub>

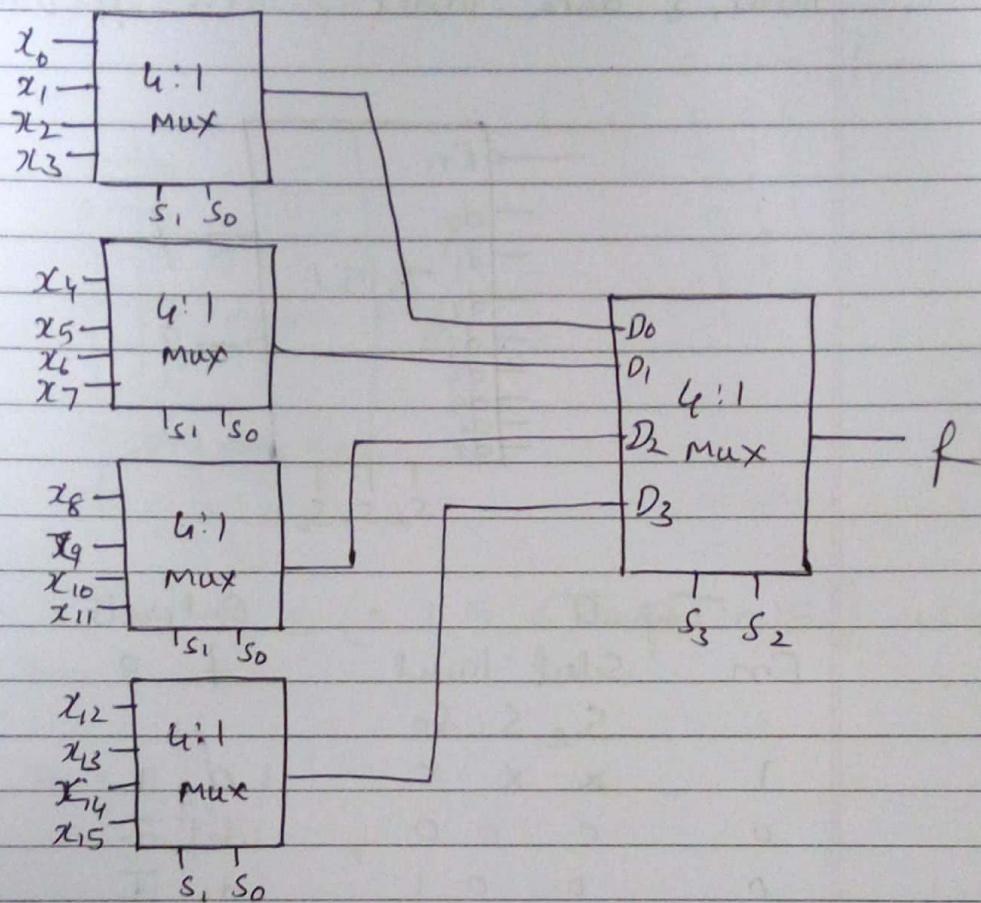
$$f = \bar{a}_1 \bar{a}_0 d_0 + \bar{a}_1 a_0 d_1 + a_1 \bar{a}_0 d_2 + a_1 a_0 d_3$$

### 74LS153



~~76153 IX~~

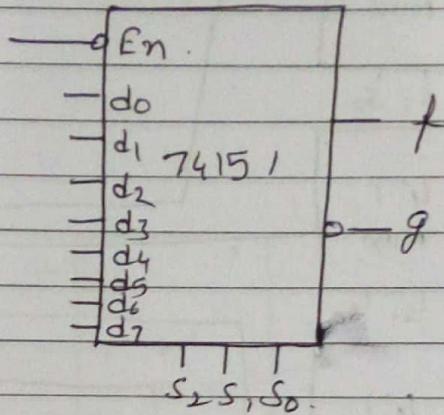
1 Configure 16:1 mux using 4:1 mux.



$S_3$	$S_2$	$S_1$	$S_0$	$f$
0	0	0	0	$x_0$
0	0	0	1	$x_1$
0	0	1	0	$x_2$
0	0	1	1	$x_3$
0	1	0	0	$x_4$
0	1	0	1	$x_5$
0	1	1	0	$x_6$
0	1	1	1	$x_7$
1	0	0	0	$x_8$
1	0	0	1	$x_9$
1	0	1	0	$x_{10}$
1	0	1	1	$x_{11}$
1	1	0	0	$x_{12}$
1	1	0	1	$x_{13}$
1	1	1	0	$x_{14}$
1	1	1	1	$x_{15}$

## 8 to 1 Mux

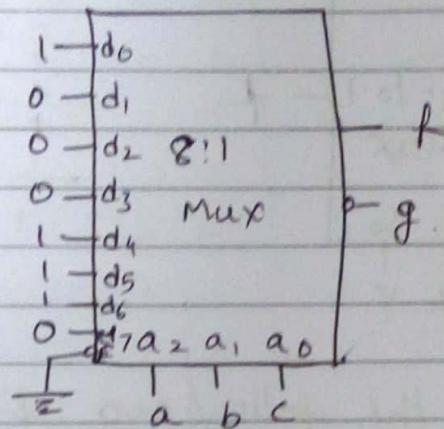
The 74151 is a 8:1 mux having 3 select lines, 8 data lines ; one off line & enable line



Inputs			Outputs	
En	Select lines		f	g
S <sub>2</sub> S <sub>1</sub> S <sub>0</sub>				
1	*	x	x	0 1
0	0	0	0	d <sub>0</sub> $\overline{d_0}$
0	0	0	1	d <sub>1</sub> $\overline{d_1}$
0	0	1	0	d <sub>2</sub> $\overline{d_2}$
0	0	1	1	d <sub>3</sub> $\overline{d_3}$
0	1	0	0	d <sub>4</sub> $\overline{d_4}$
0	1	0	1	d <sub>5</sub> $\overline{d_5}$
0	1	1	0	d <sub>6</sub> $\overline{d_6}$
0	1	1	1	d <sub>7</sub> $\overline{d_7}$

1 Implement the following function using a 8 to 1 mux.

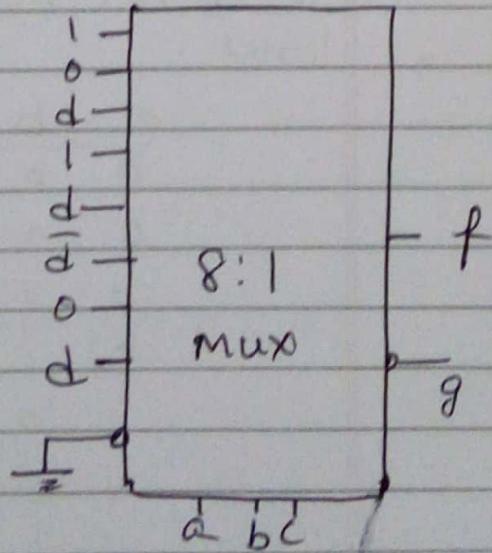
$$f(a, b, c) = \Sigma(0, 4, 5, 6)$$



$a$	$b$	$c$	$f$
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	0

2  $f(a, b, c, d) = \Sigma(0, 1, 5, 6, 7, 9, 10, 15)$ . using 8 to 1 mux.

$a$	$b$	$c$	$d$	$f$
0	0	0	0	1
0	0	0	1	1
0	0	1	0	0
0	0	1	1	0
0	1	0	0	0
0	1	0	1	1
0	1	1	0	1
0	1	1	1	1
1	0	0	0	0
1	0	0	1	0
1	0	1	0	1
1	0	1	1	0
1	1	0	0	0
1	1	0	1	0
1	1	1	0	1
1	1	1	1	0



3 Implement  $f(x,y,z) = \sum m(0,2,3,5)$  using 4 to 1 line multiplexer.

	$x$	$y$	$z$	$f$
0	0	0	1	1
1	0	0	0	0
2	0	1	0	1
3	0	1	1	1
4	1	0	0	0
5	1	0	1	1
6	1	1	0	0
7	1	1	1	0

	$\bar{z}$	$\bar{y}$	$\bar{x}$	$f$
0	1	1	1	1
1	1	1	0	0
2	1	0	1	1
3	1	0	0	0
4	0	1	1	1
5	0	1	0	0
6	0	0	1	1
7	0	0	0	0

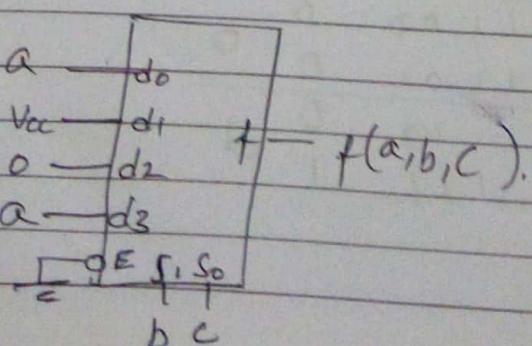
4-to-1 MUX

4 Implement the following fn using 4:1 mux.  $f(a,b,c) = \sum m(1,4,5,7)$

5 Implement the following fn using 8:1 mux  $f(w,x,y,z) = \sum m(0,1,2,4,5,7,8,9,12,13)$  (IE74151)

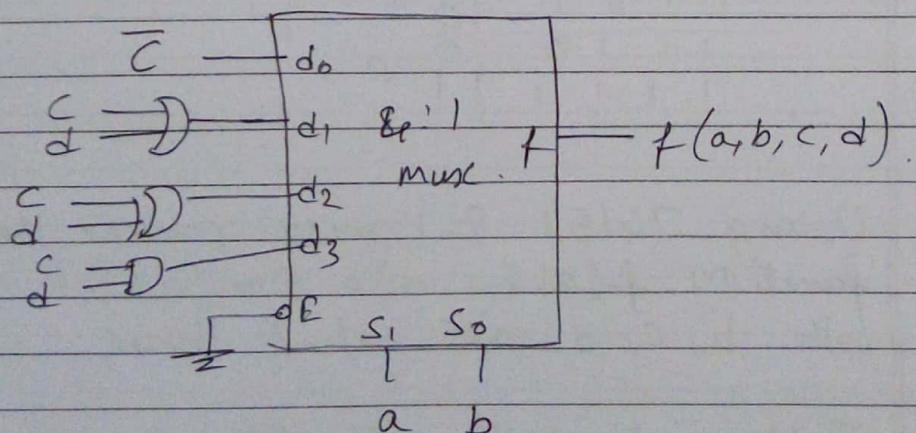
6 Implement  $f(a,b,c) = \sum m(1,4,5,7)$  using a 4 to 1 mux using b & c as the address lines.

$a$	$b\ c$	$f$	$b\ c$	$f$
0	00	0	00	a
0	01	1	01	1
0	10	0	10	0
0	11	0	11	a
1	00	1		
1	01	1		
1	10	0		
1	11	1		



7. Implement the fin using 4:1 mux with a & b as select lines.  $f(a, b, c, d) = S(0, 1, 5, 6, 7, 9, 10, 15)$

a . b	c	d	$f$
0 0 0 0			1
0 0 0 1			1
0 0 1 0			0
0 0 1 1			0
0 1 0 0			0
0 1 0 1			$c+d$
0 1 1 0			1
0 1 1 1			1
1 0 0 0			0
1 0 0 1			1
1 0 1 0			$c \oplus d$
1 0 1 1			0
1 1 0 0			0
1 1 0 1			$cd$
1 1 1 0			0
1 1 1 1			1



8. Implement the full adder circuit using 8:1 mux & 4:1 mux.

9. Implement the following Boolean fn using 8:1 multiplexer:

$$Y = f(A, B, C, D) = \bar{A}B\bar{D} + ACD + \bar{B}CD + \bar{A}\bar{C}D$$

$$\Rightarrow Y = \bar{A}B\bar{D}(C+\bar{C}) + A(B+\bar{B})CD + (A+\bar{A})\bar{B}CD + \bar{A}(B+\bar{B})$$

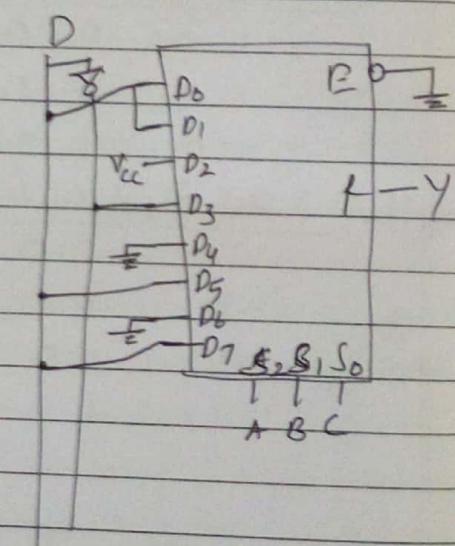
$$= \bar{A}BC\bar{D} + \bar{A}B\bar{C}\bar{D} + ABCD + A\bar{B}CD + A\bar{B}\bar{C}D + \bar{A}\bar{B}CD +$$

$$\bar{A}B\bar{C}D + \bar{A}\bar{B}\bar{C}D$$

$$= \bar{A}BC\bar{D} + \bar{A}B\bar{C}\bar{D} + ABCD + A\bar{B}CD + A\bar{B}\bar{C}D + \bar{A}\bar{B}CD + \bar{A}B\bar{C}D +$$

$$= \sum_m(1, 3, 4, 5, 6, 11, 15)$$

A	B	C	D	$f_i$
0	0	0	0	0
0	0	0	1	0
0	0	1	0	1
0	0	1	1	0
0	1	0	0	1
0	1	0	1	1
0	1	1	0	1
0	1	1	1	0
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	1
1	1	0	0	0
1	1	0	1	0
1	1	1	0	0
1	1	1	1	1



10 Using 74151 8:1 mux realize the Boolean function  $f(a, b, c, d) = \sum_m(0, 1, 5, 6, 7, 10, 15)$  with  $b, c, d$  as select lines.

11 Implement full subtractor using decoders

12 Implement the following function using IC 74138 & external gates. Also write the truth table.

$$P = f_1(x, y, z) = \sum(1, 2, 5, 6) \quad Q = f_2(x, y, z) = \sum(3, 5, 6, 7)$$

13 Write block diagram representation of a full adder using 3:8 decoders.

A B Cin S Cout

0 0 0 0 0

0 0 1 1 0

$$S = \Sigma m(1, 2, 4, 7)$$

0 1 0 1 0

0 1 1 0 1

$$\text{Cout} = \Sigma m(3, 5, 6, 7)$$

1 0 0 1 0

1 0 1 0 1

1 1 0 0 1

1 1 1 1 1

