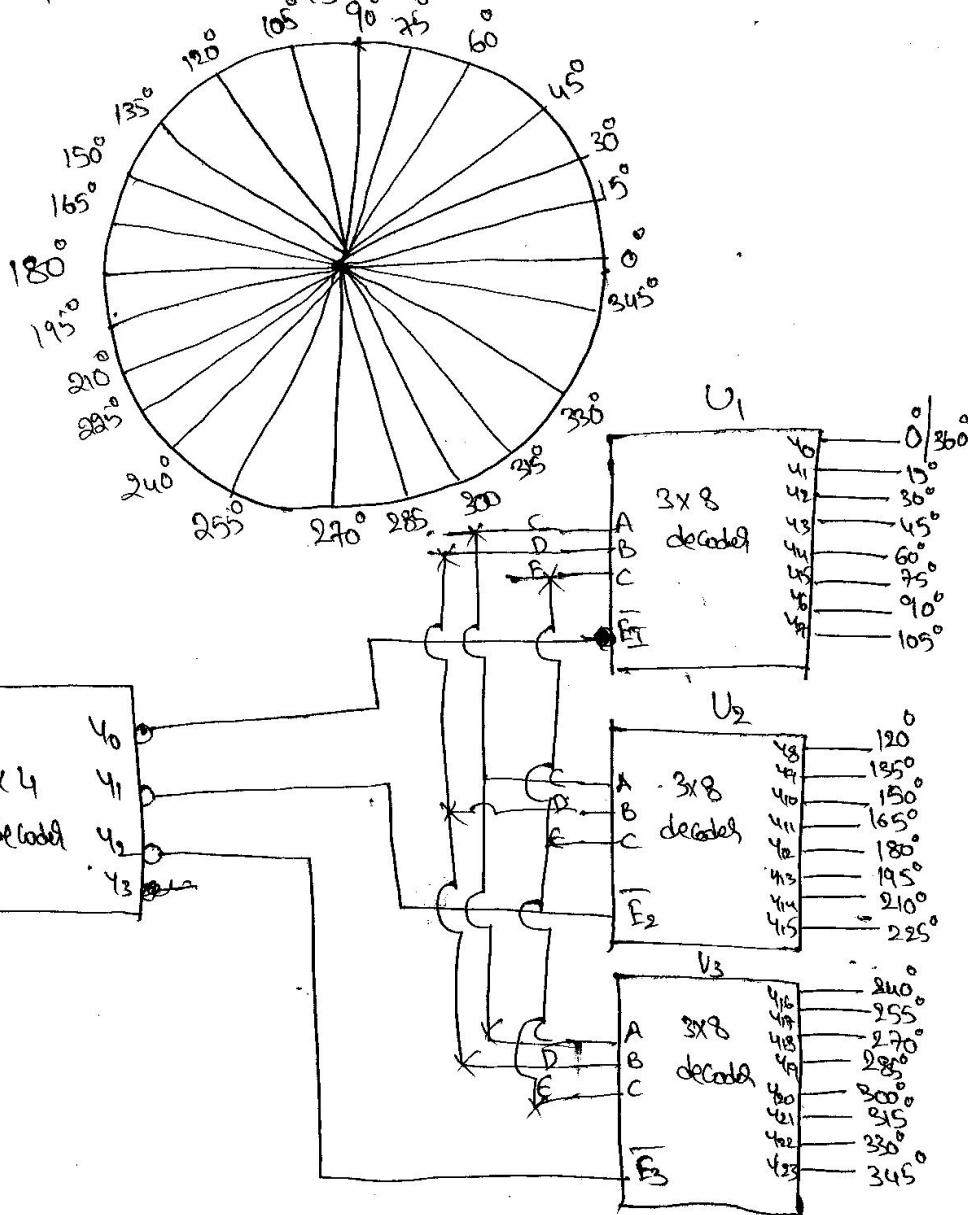


\* A mechanical disk rotates in a circle in different positions. (34)  
 two successive positions differ with an angle of  $15^\circ$ . Provide an encoding mechanism for every position of the disk. The disk in the mechanical system has this encoded information to detect the exact position. Design a decoder with an enable input to identify the position of the disk.

Sol: one disk rotation =  $360^\circ$   
 200 positions differ with an angle =  $15^\circ$   
 total position =  $\frac{360^\circ}{15^\circ} = 24$  positions.



<u>C</u>	<u>A</u>	<u>B</u>	<u>C</u>	<u>D</u>	<u>E</u>	<u>Shade(R)</u>	<u>Shade(P)</u>
0	0	0	0	0	0	0° / 360°	
0	0	0	0	0	1	15°	
0	0	0	0	1	0	30°	
0	0	0	0	1	1	45°	
0	0	0	0	1	0	60°	
0	0	0	0	1	1	75°	
0	0	0	0	1	0	90°	
0	0	0	0	1	1	105°	
0	0	0	0	1	0	120°	
0	0	0	0	1	1	135°	
0	0	0	0	1	0	150°	
0	0	0	0	1	1	165°	
0	0	0	0	1	0	180°	
0	0	0	0	1	1	195°	
0	0	0	0	1	0	210°	
0	0	0	0	1	1	225°	
0	1	0	0	0	0	240°	
0	1	0	0	0	1	255°	
0	1	0	0	0	0	270°	
0	1	0	0	1	1	285°	
0	1	0	1	0	0	300°	
0	1	0	1	1	0	315°	
0	1	0	1	1	1	330°	
1	X	X	X	X	X	+ 0 -	

(35)

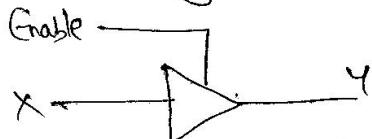
## \* Three State devices :- (8) Three State Buffers :-

(26)

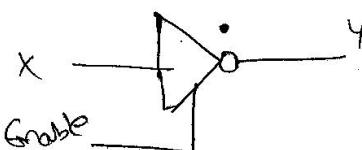
It is also known as 3-state device. These are designed to have greater op current & voltage capability than ordinary circuit.

Ordinary Circuit +

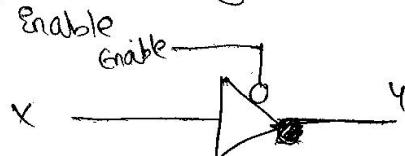
(a) non inverting active high enable



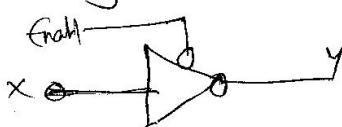
(c) Inverting active high enable



(b) non inverting active low enable



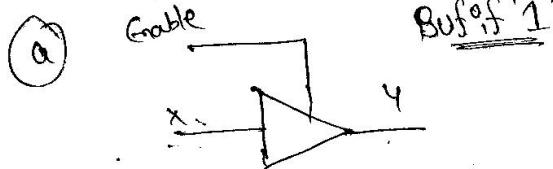
(d) Inverting active low enable



The 3-state device is a digital circuit exhibits 3-states

- (1) Logic 1
- (2) Logic '0'
- (3) High impedance state (l) floating

The high impedance state is o/p appears as disconnected like an open state.

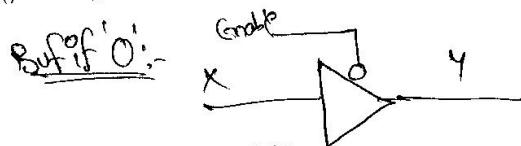


Enable	X	Y
0	X	Z
0	1	1
1	0	0

Entity Bufif'1' is

```
Port( x: in std_logic;
      En: in std_logic;
      y: out std_logic);
End Bufif'1';
```

```
Architecture
begin
  y := 'Z' when En = '0' else x;
End Bufif'1';
```

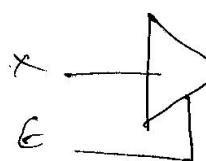


Enable	X	Y
0	0	0
0	1	1
1	X	Z

Architecture dataflow of Bufif'0' is  
 $y = 'Z'$  when  $En = 0$  else  $x$ ;  
 End Bufif'0';

High impedance :- means that a point in a circuit (a node) allows relatively small amount of current per unit of applied voltage at that point. High impedance circuits are low resistive, high voltage.  
Low impedance :- low voltage, high current.

Not of '1':-



Enable	X	Y
1	0	1
1	1	0
0	X	Z

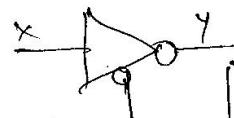
Archer

begin

$Y_k = 'Z'$  when  $E_n = '0'$  else  $\text{not } X_j$

end Inbuff;

Not of '0':-



C	X	Y
0	0	1
0	1	0
1	X	Z

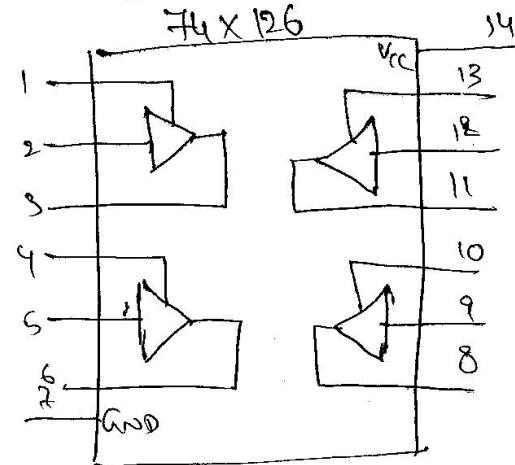
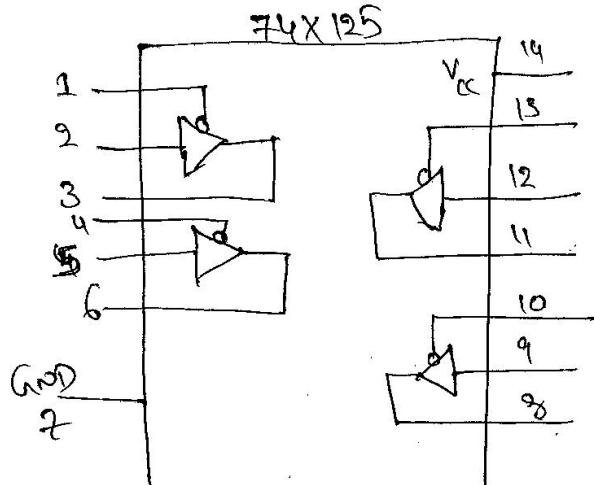
(37)

$Y_k = 'Z'$  when  $E_n = '1'$  else  $\text{not } X_j$

End

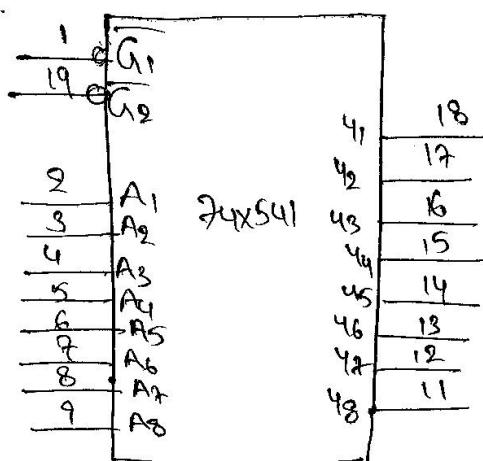
### \* 8-to State Buffer Ic's:- (74X125) & (74X126)

$74X125 \rightarrow$  IC tri-state enable op's are low } Contains non-inverting  
 $74X126 \rightarrow$  IC " " " op's are high. } tri-state buffer in a  
 14-pin package

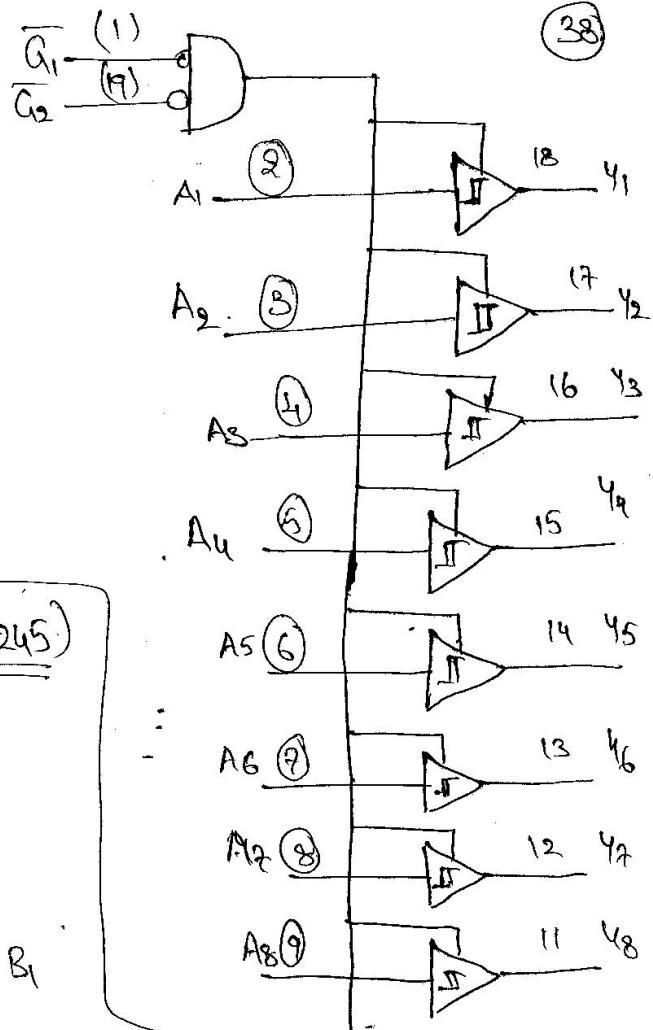


### \* Octal tri-state Buffer:- (74X541)ic

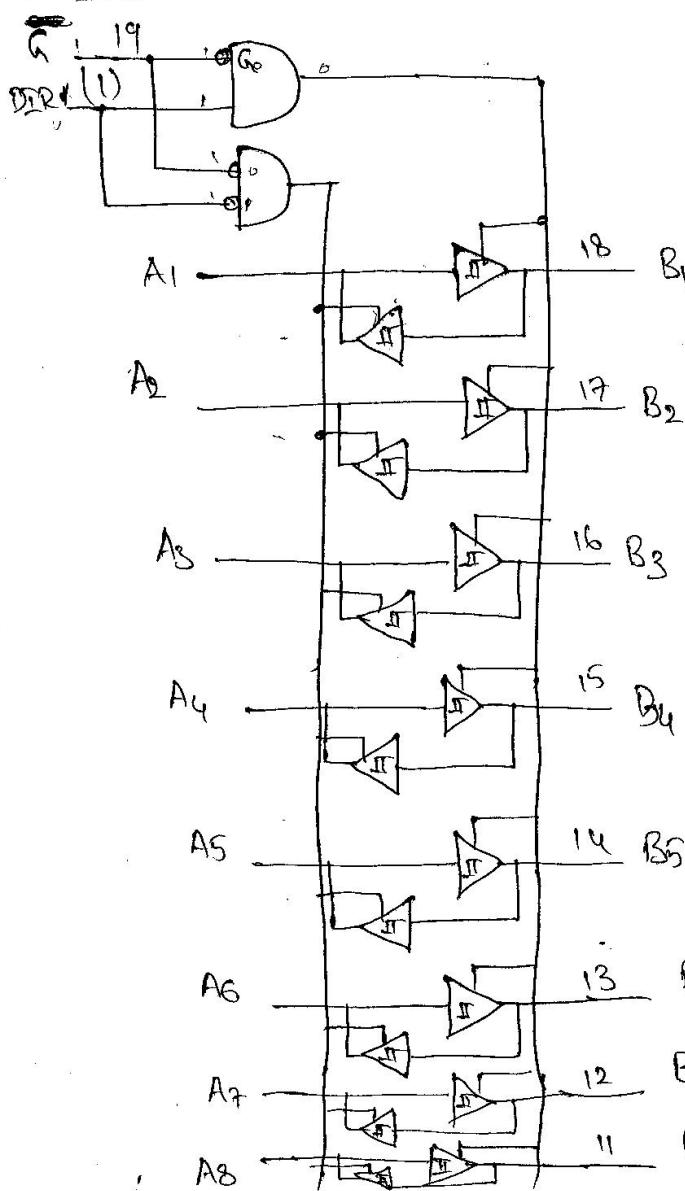
It is an non-inverting tri-state buffer. The  $G_1$  and  $G_2$  are enable op's, which must be asserted to enable the tri-state op's. The small rectangular symbols inside the buffer symbols indicates "hysteresis", an electrical characteristics of the op that improve noise immunity.



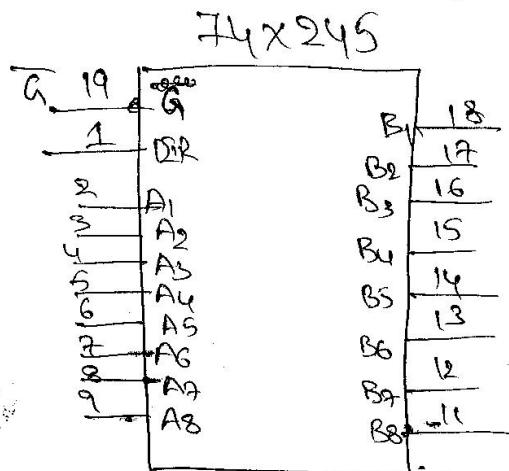
(a) Logic symbol



(b) Octal to state transceiver :- (74x245)

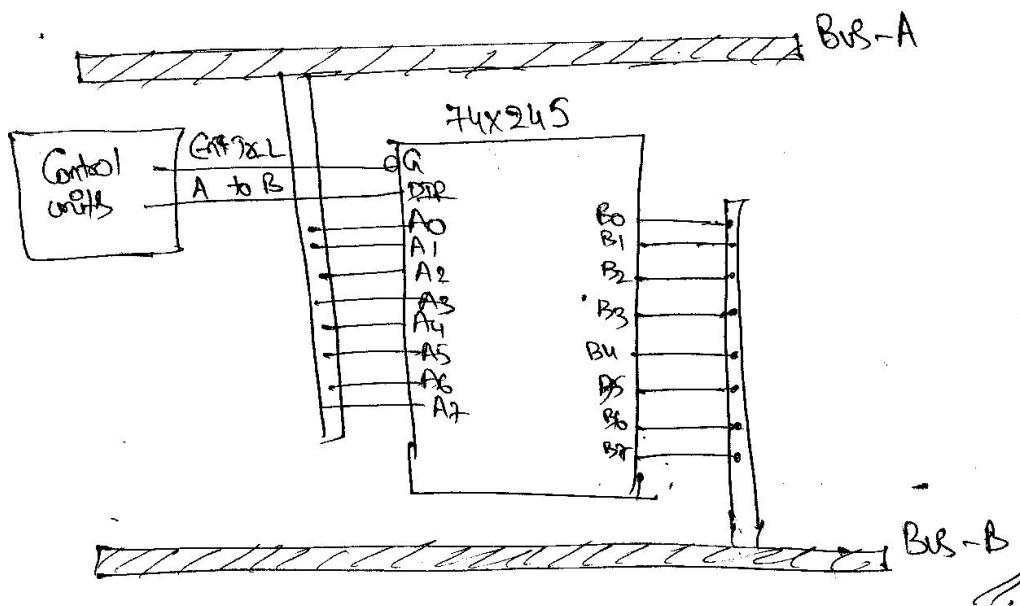


DIR  $\rightarrow$  Portaflow (d) Direction of data transfer



In oral 740 state transceiver (74x248) it has DIR pin which (39)  
determine the direction of transfer from A to B (DIR = 1) (8)  
from B to A (DIR = 0). The tri state buffer for selected direction is  
enabled only if G is asserted.

Three state buffer for the selected direction is enabled only  
if G is set 'low' (0). Three different modes of operations are  
possible based on the state of G and DIR. 11.



\* → In three state devices a single line is shared by a large no. of  
sources, but it is limited to only one source at a time.  
This is understood by using 3 to 8 decoder.

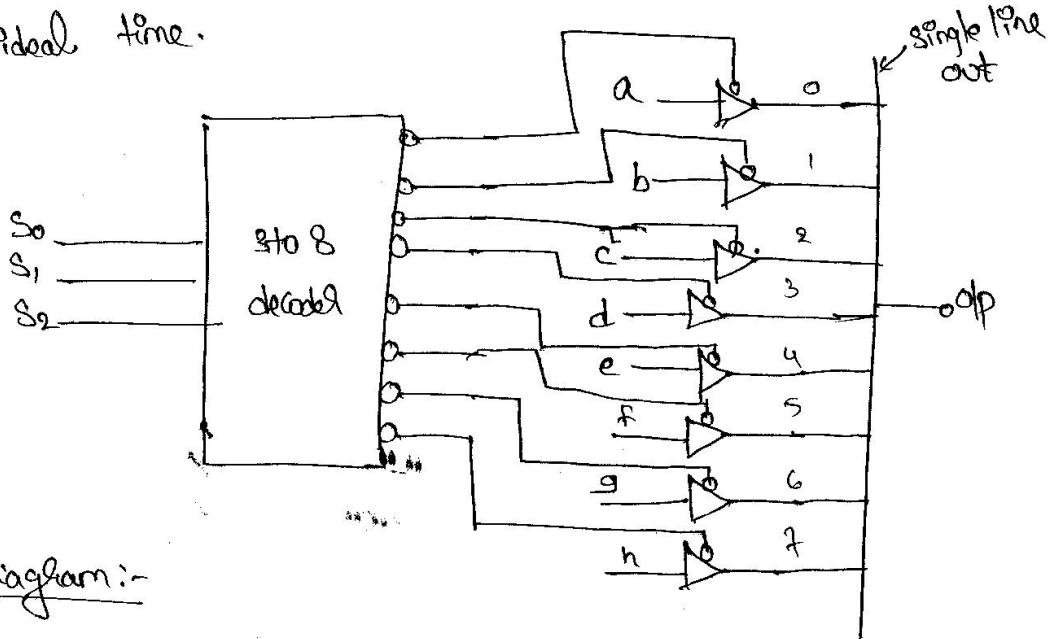
\* → In this it is having  $S_0, S_1$  &  $S_2$  as pins which are used  
to drive a single op at a time.

\* → The time the device takes to come out of high impedance state is  
more than the time it takes to go into high impedance state.

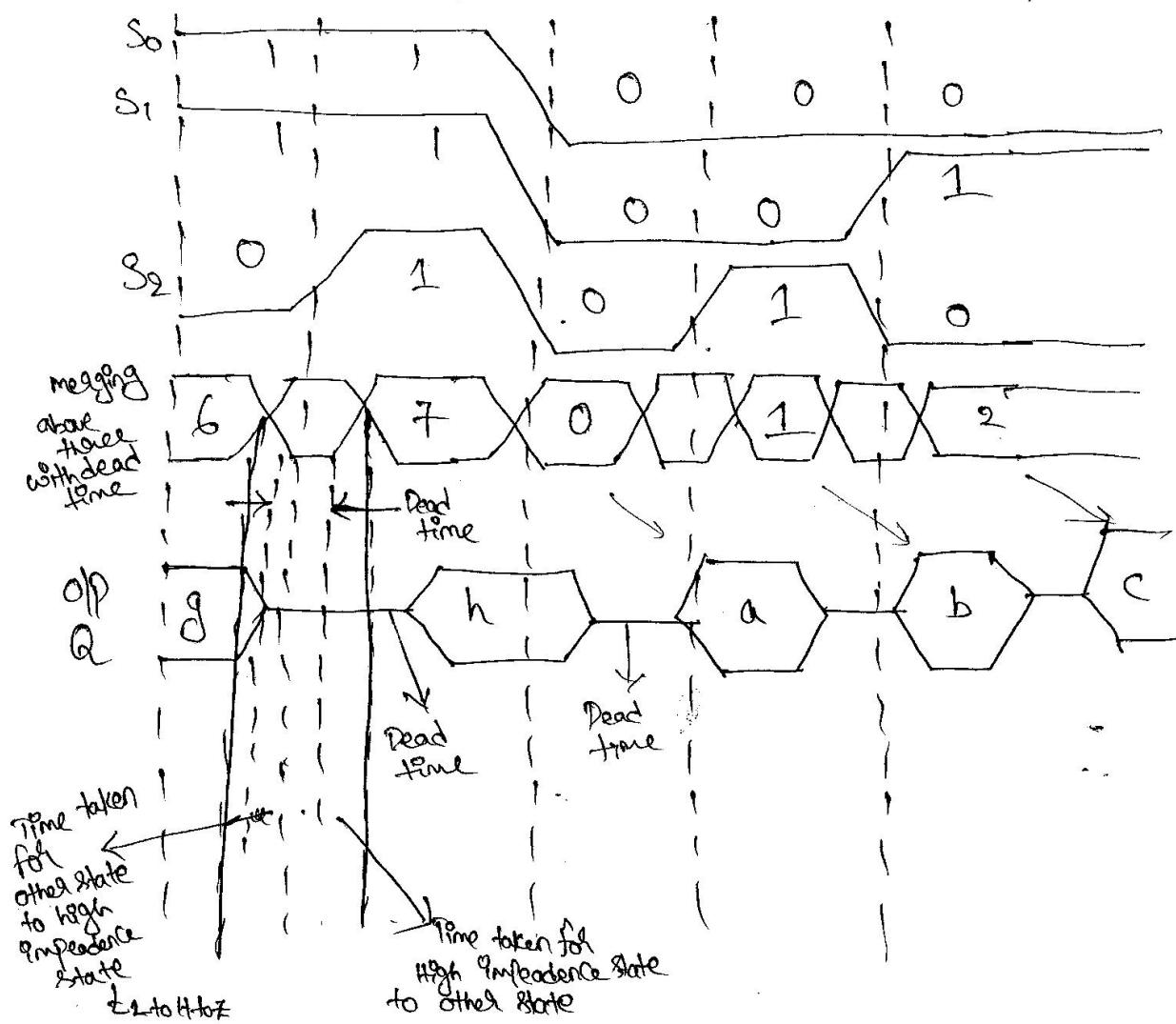
If 3 state devices are connected having different turn-on & turn-off  
time it takes to go into high problem called "fighting" arises. To avoid

(10)

In this problem, the logic circuit has to be designed in such a way that it has a dead time, that means it must have some ideal time.



Pining diagram:-



(41)

\* ) VHDL Program for 74x245 :-

library IEEE;

use IEEE. STD-LOGIC-1164.ALL;

entity 74x245 is

Port ( G, DIR : in STD-LOGIC ;  
A, B : out STD-LOGIC-VECTOR (8 down to 1) );

end 74x245;

Architectural behavioral of 74x245 is //

begin process (G, DIR)

begin  
if G = '0' , DIR = '0'  
A <= B ;

else B <= A ;

end if ;

end Process ;

end behavioral ;

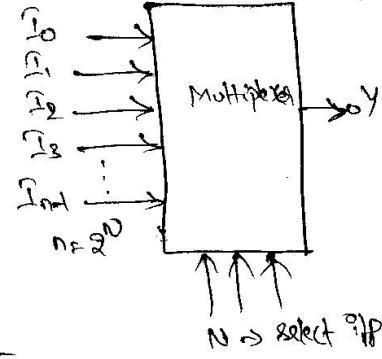
42

### Multiplexer :- (Data Selector)

It is known as a digital switch. It is used to switch the data from multiple sources destination. A multiplexer is generally represented "MUX".

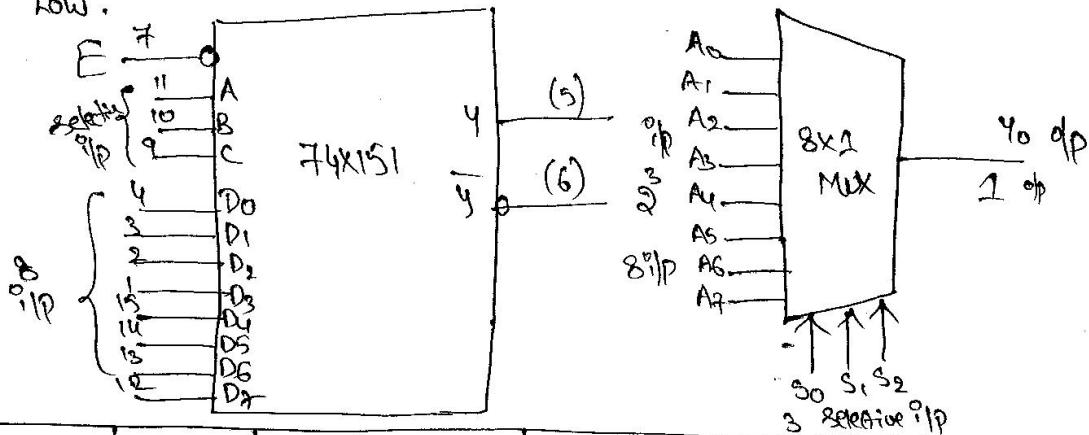
$$Y = \sum_{j=0}^{n-1} E_j \cdot M_j \cdot D_j$$

where  $\Sigma$  stand for logical sum of Product (SOP) term  
 $M_j$  = minterms       $j$  = select  $^{\text{of}} \text{IP}$   
 $D_j$  =  $^{\text{of}} \text{IP}$  Bit.



\* The 74XX151 (8 to 1 Multiplexed) :-

Has  $^{\text{of}} \text{IP}$  and it provides 2-ops one is Active high and other is active low.



Enable	A	B	C	Y - op	Y <sub>0</sub>
0	0	0	0	D(0)	D <sub>0</sub>
0	0	0	1	D(1)	D <sub>1</sub>
0	0	1	0	D(2)	D <sub>2</sub>
0	0	1	1	D(3)	D <sub>3</sub>
0	1	0	0	D(4)	D <sub>4</sub>
0	1	0	1	D(5)	D <sub>5</sub>
0	1	1	0	D(6)	D <sub>6</sub>
0	1	1	1	D(7)	D <sub>7</sub>

Program:- (behavioral if):-

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

entity MX8X1 is

Port ( D : in STD\_LOGIC (7 downto 0);  
 S : in STD\_LOGIC (2 downto 0);  
 E\_n : in STD\_LOGIC;  
 Y : out STD\_LOGIC );

end MX8X1;

Architecture behavioral of MX8X1 is

begin

Process (I, S, E\_n)

begin

if  $E_n = 0$  then

if  $S = "000"$  then  $Y = D_0$ ;

elsif  $S = "001"$  then  $Y = D_1$ ;

elsif  $S = "010"$  then  $Y = D_2$ ;

elsif  $S = "011"$  then  $Y = D_3$ ;

elsif  $S = "100"$  then  $Y = D_4$ ;

elsif  $S = "101"$  then  $Y = D_5$ ;

elsif  $S = "110"$  then  $Y = D_6$ ;

elsif  $S = "111"$  then  $Y = D_7$ ;

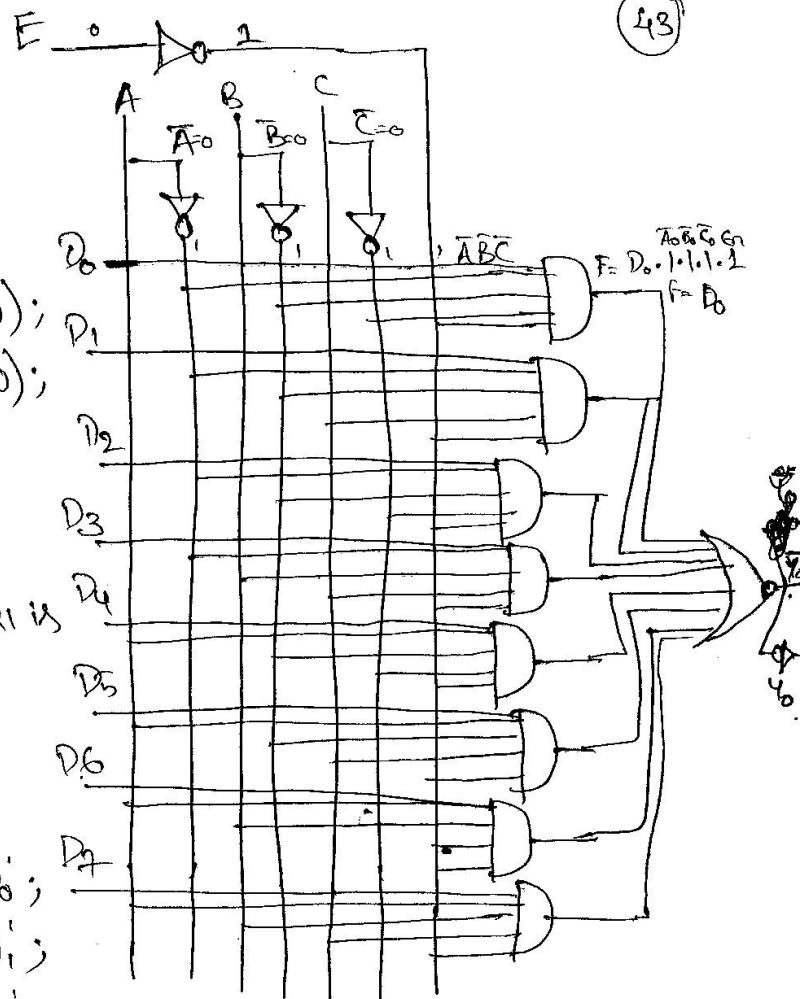
end if;

else  $E_n = 1$  then "others";

end if;

end Process;

end behavioral;



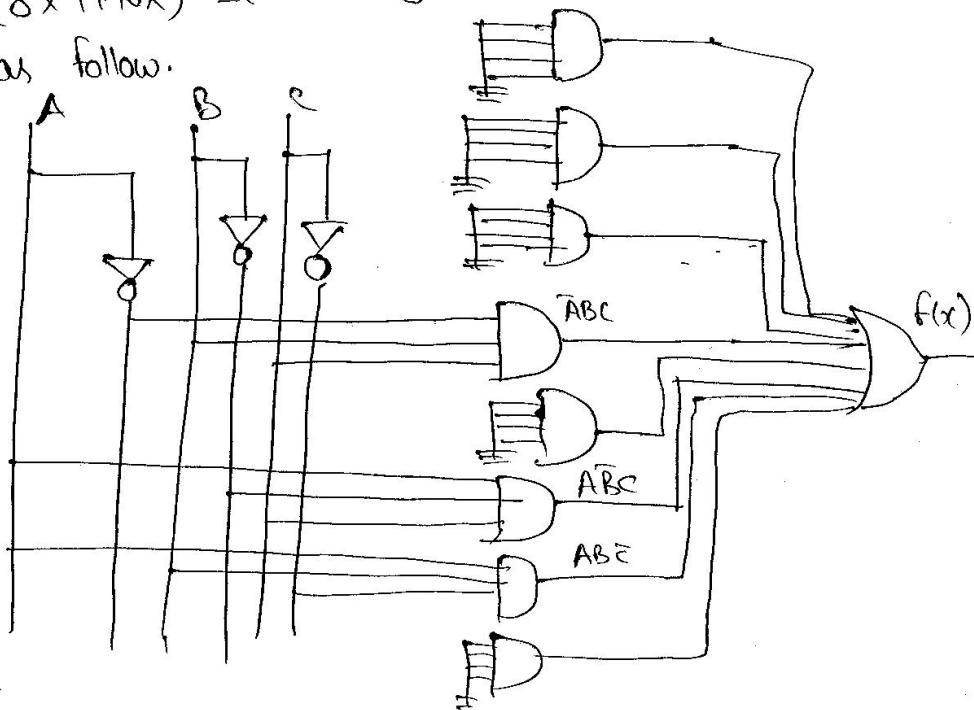
43

\* Realize the following function using 74x151

(44)

$$f(x) = \underbrace{\bar{ABC}}_3 + \underbrace{\bar{A}\bar{B}C}_5 + \underbrace{AB\bar{C}}_6$$

Sol:- from given  $f(x) = \bar{ABC} + \bar{A}\bar{B}C + AB\bar{C}$  from the internal structure of 74x151 (8x1 Mux) in the logical diagram to extract this function is as follow.



WHDL Program:-

```
library IEEE;
use IEEE.STD.LOGIC_1164.all;
```

Entity function 74x151 is

```
Port(A,B,C : in STD-LOGIC;
      F(x) : out STD-LOGIC);
```

End 74x151 function;

Architecture dataflow of func 74x151 is

Signal D : STD-LOGIC\_VECTOR(2 DOWNTO 0);

```
begin
```

D(2) <= (A and B and not C);

D(1) <= (A and not B and C);

D(0) <= (not A and B and C);

F(x) <= (D(2) OR D(1) OR D(0));

End dataflow;

\* Realize the following expression using 74x151 IC

(45)

$$F(x) = AB + BC + AC$$

Ques: Given  $F(x) = AB + BC + AC$  this can be written as

$$\begin{aligned} f(x) &= AB(C + \bar{C}) + BC(A + \bar{A}) + A(B + \bar{B})C \\ &= ABC + ABC\bar{C} + ABC + \bar{ABC} + ABC + \bar{ABC} \end{aligned}$$

$$f(x) = \underbrace{ABC}_{D_2} + \underbrace{\frac{ABC\bar{C}}{D_3}}_{\bar{B}} + \underbrace{\frac{ABC}{D_3}}_{B} + \underbrace{\frac{\bar{ABC}}{D_3}}_{\bar{B}}$$

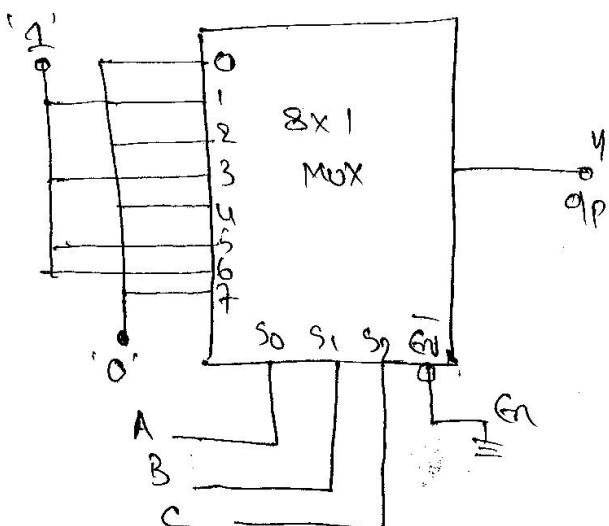
Draw logic diagram:-

\* Implementation of Combinational logic using Mux:-

Implement the following Boolean function using 8x1 Mux

$$F(A, B, C) = \sum m(1, 3, 5, 6)$$

All those variables A, B, C are applied as selective o/p. of Mux. The minterms (1, 3, 5, 6) are chosen by making their corresponding o/p line equal to '1' and minterms (0, 2, 4, 7) are not included by making them o/p to connect to '0' o/p line



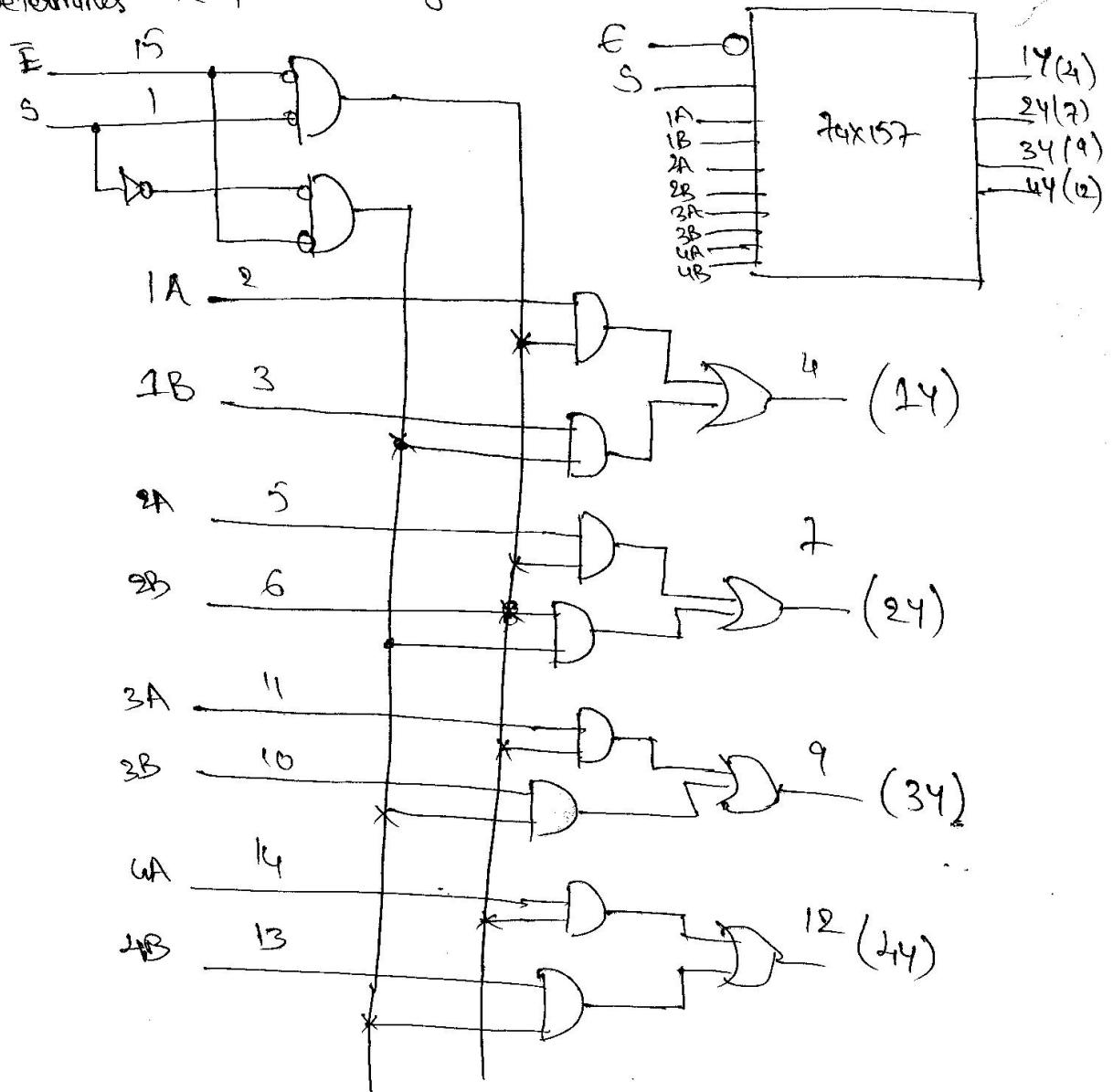
\* Implement

\* ) The 74x157 Quad 2<sup>0</sup>p Multiplexer:-

(46)

IC 74x157 which selects four bits of data from two sources under the control of a common Select 0<sup>1</sup>P (S). The Enable 0<sup>1</sup>P is active low. When E is high all of the 0<sup>1</sup>P (4) are forced low regardless of all other 0<sup>1</sup>P conditions.

Moving data from two groups of registered to four common 0<sup>1</sup>P buses in a common use of IC 74x157. The state of Select 0<sup>1</sup>P determines the particular register from which the data comes.



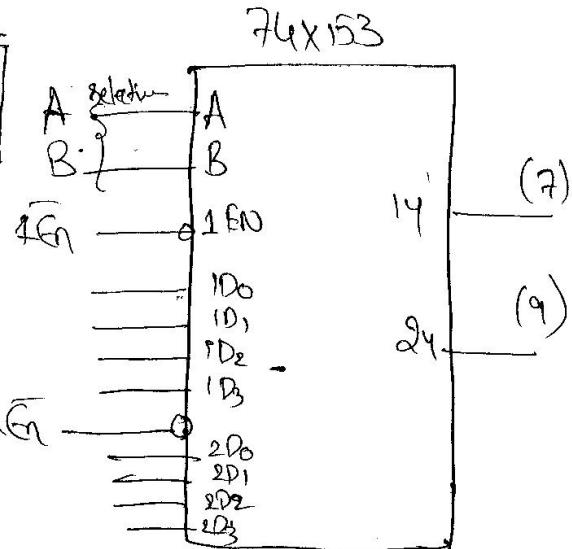
Truth table :- (74x157)

(47)

		Op			
$\bar{E}_N$	S	14	24	34	44
1	X	0	0	0	0
0	0	1A	2A	3A	4A
0	1	1B	2B	3B	4B

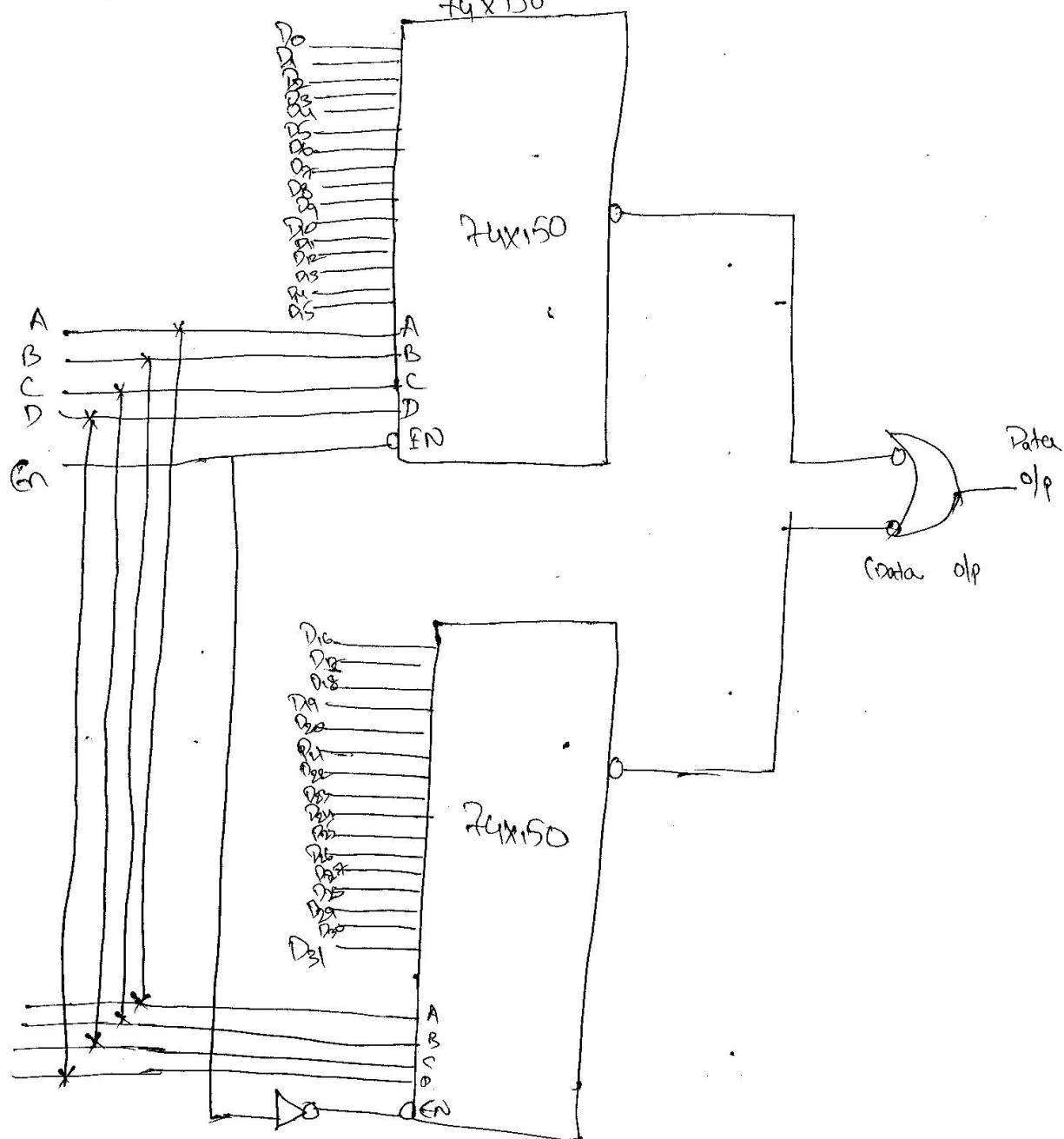
\*) 74x153 Dual 4 to 1 MUX :-

Enable Selection				Op's	
$\bar{E}_N$	$\bar{2}E_N$	B	A	14	24
0	0	0	0	$1D_0$	$2D_0$
0	0	0	1	$1D_1$	$2D_1$
0	0	1	0	$1D_2$	$2D_2$
0	0	1	1	$1D_3$	$2D_3$
-	-	-	-	-	-
0	1	0	0	$1D_0$	0
0	1	0	1	$1D_1$	0
0	1	1	0	$1D_2$	0
0	1	1	1	$1D_3$	0
-	-	-	-	-	-
1	0	0	0	0	$2D_0$
1	0	0	1	0	$2D_1$
1	0	1	0	0	$2D_2$
1	0	1	1	0	$2D_3$
-	-	-	-	-	-
1	1	X	X	0	0



\* Design 32 to 1 Mux using two 74LS150 IC's

(48)

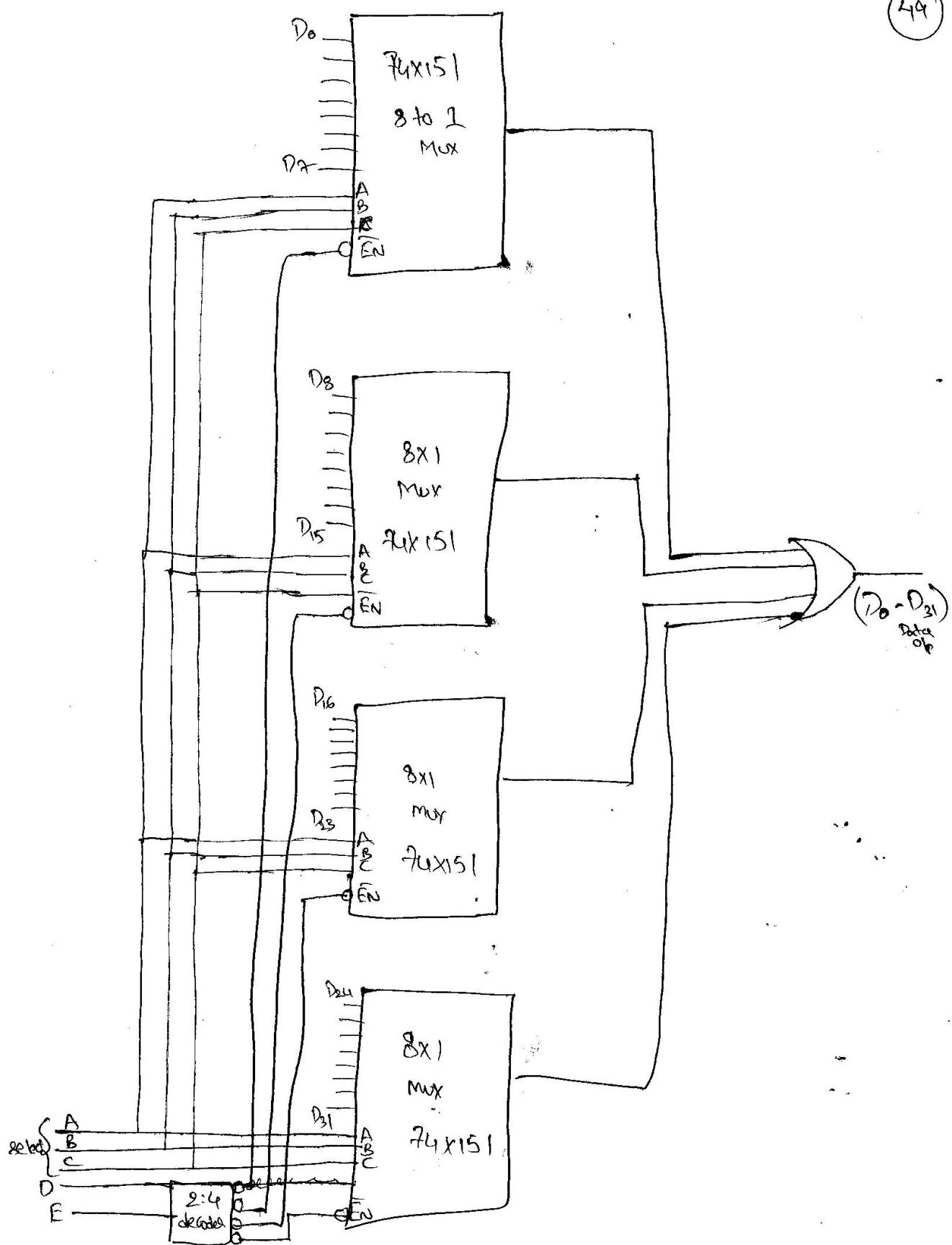


Write a VHDL program using dataflow style for above

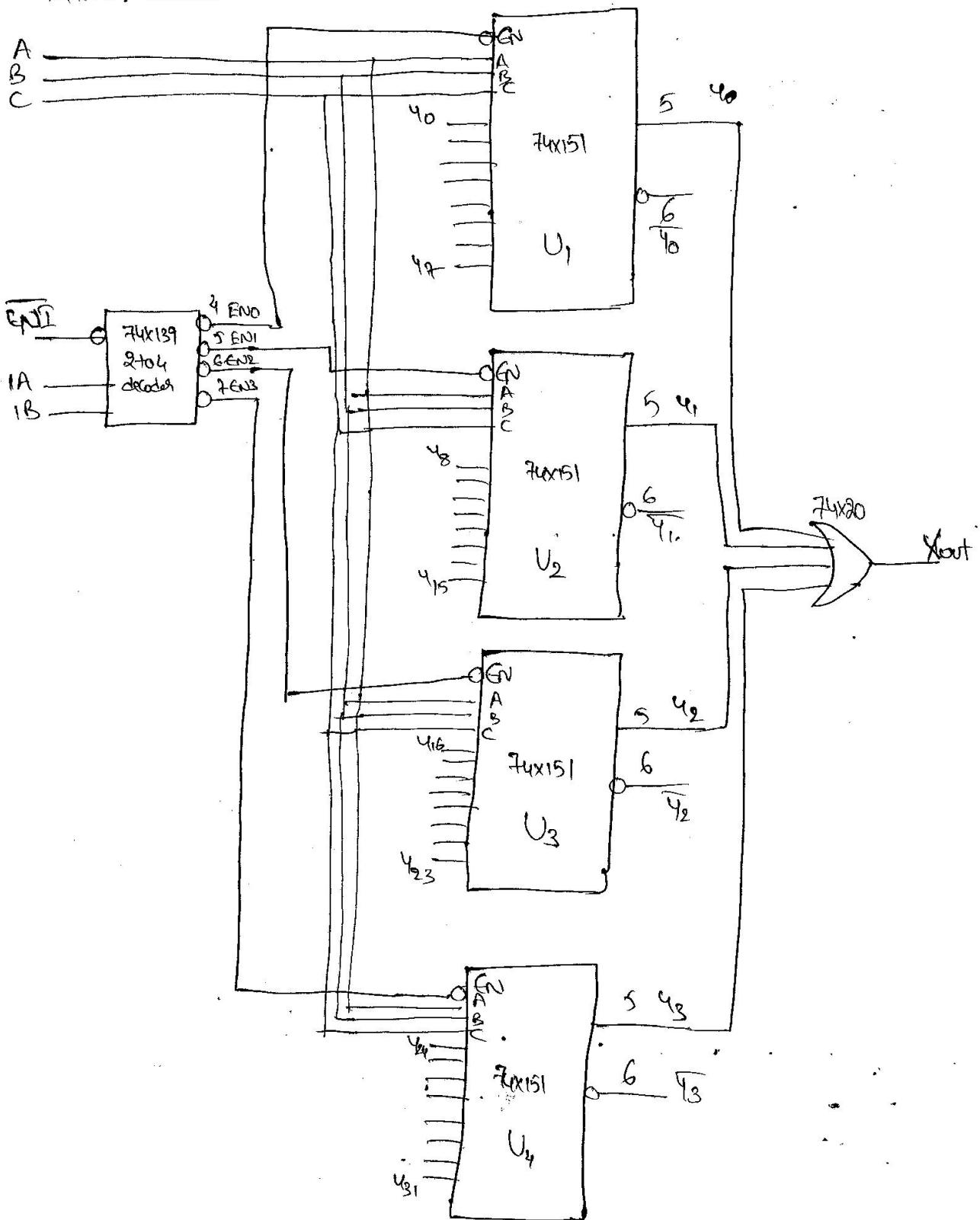
32 x 1 Mux?

\*) Design 32 to 1 Mux using four 8 to 1 Mux and 2 to 4 decoder?

(49)



Design a 32 to 1 mux using four 74x151 multiplexers and one 74x139 decoder? (50)

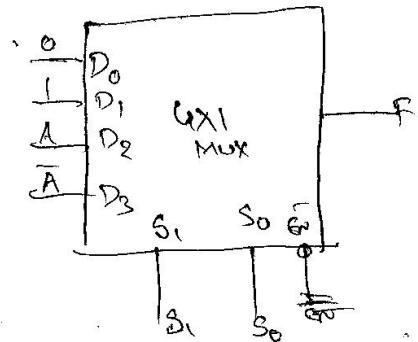


d) Implement the following Boolean function using 4:1 MUX.

$$f(A, B, C) = \sum m(1, 3, 5, 6)$$

here how many '1's are required '4' will be considered in rows

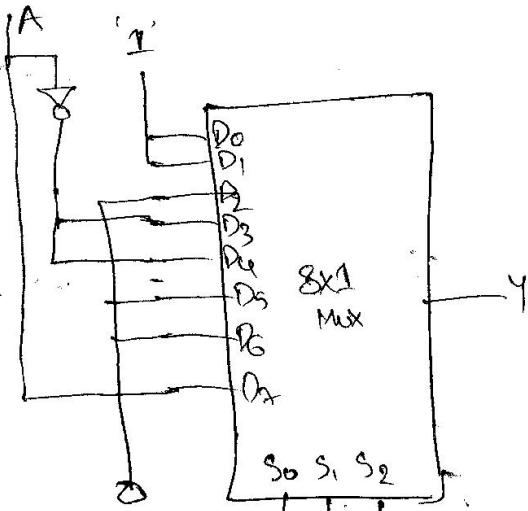
	$D_0$	$D_1$	$D_2$	$D_3$
$\bar{A}$	0	1	2	3
A	4	5	6	7
	0	1	A	$\bar{A}$



d) Implement the following boolean function using 8x1 MUX

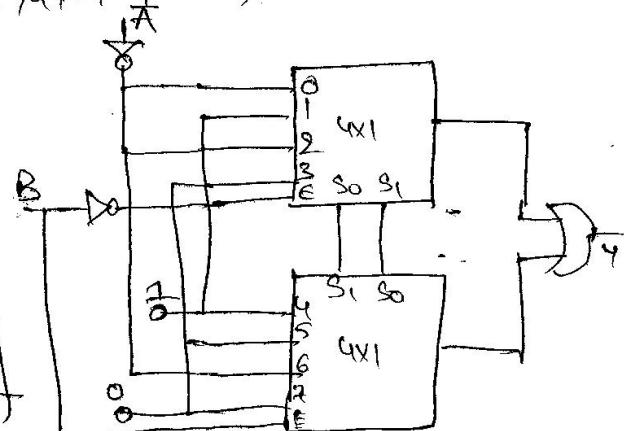
$$f(P, Q, R, S) = \sum m(0, 1, 3, 4, 8, 9, 15)$$

	$D_0$	$D_1$	$D_2$	$D_3$	$D_4$	$D_5$	$D_6$	$D_7$
$\bar{A}$	0	1	2	3	4	5	6	7
A	8	9	10	11	12	13	14	15
	1	1	0	$\bar{A}$	$\bar{A}$	0	0	A
	(X)	(X)						



d) Implement following function for 4x1 Mux? 0  
 $f(A, B, C, D) = \sum m(0, 1, 2, 4, 6, 9, 12, 14)$ .

	$D_0$	$D_1$	$D_2$	$D_3$	$D_4$	$D_5$	$D_6$	$D_7$
$\bar{A}$	0	1	2	3	4	5	6	7
A	8	9	10	11	12	13	14	15
	$\bar{A}$	1	$\bar{A}$	0	1	0	$\bar{A}$	0



Q) Implement following func using 8x1 mux

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

(52)

P) Implement the following Boolean func using 8x1 mux

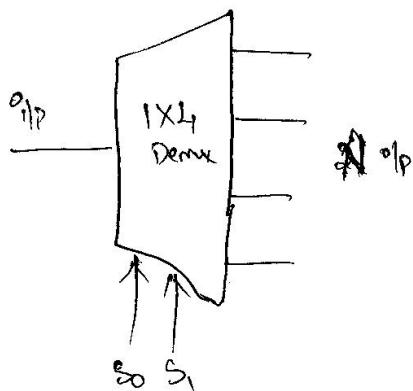
$$F(A, B, C, D) = \Sigma m(0, 3, 5, 8, 9, 10, 12, 14).$$

R) Implement the following Boolean func by using 8x1 mux

$$F(A, B, C, D) = \Sigma m(0, 2, 6, 10, 11, 12, 13) + d(3, 8, 14).$$

\* De Muxplexer :-

is a circuit that receives information on a single line, and transmits this information on one of  $2^n$  possible o/p lines.



$\hookrightarrow$  Selective

$S_1$	$S_0$	$D_3$	$D_2$	$D_1$	$D_0$
0	0	0	0	0	1
0	1	0	0	1	0
1	0	0	1	0	0
1	1	1	0	0	0

IC No:-

Description

O/P

Inverted O/P

1) 74X138

1:8 Demux (3 line to 8 line decode)

" O/P

2) 74X139

Dual 1:4 Demux (2 line to 4 line decode)

one same as O/P and other inverted O/P

3) 74X155

Dual 1:4 Demux (2 line to 4 line decode)

do expect open collector

4) 74X156

Dual 1:4 Demux (2 line to 4 line decode)

same as O/P

5) 74X154

1:16 Demux (4 line to 16 line decode)

open collector

6) 74X159

1:16 demux (4 line to 16 line decode)

some as O/P

VHDL Program for 32x1 Mux using dataflow style

$2^5 \rightarrow 5$  Selections

library IEEE;

use IEEE.STD.LOGIC\_1164.ALL;

entity 32x1\_mux is

Port ( D : in STD.LOGIC\_VECTOR(31 downto 0);

A : in STD.LOGIC\_VECTOR(2 downto 0);

EN\_L : in STD.LOGIC;

EN : inout STD.LOGIC\_VECTOR(3 downto 0);

Y : inout STD.LOGIC\_VECTOR(3 downto 0);

Y\_L : out STD.LOGIC\_VECTOR(3 downto 0);

Q : out STD.LOGIC);

end 32x1\_mux;

Architecture dataflow of 32x1 mux is

signal ENbar : STD.LOGIC\_VECTOR(3 down to 0);

signal S : STD.LOGIC\_VECTOR(2 down to 0);

begin

ENbar <= not EN\_L;

if (EN\_L = '0') then Y <= S(4 down to 3);

with A select

Y <= "0001" when "00";

Y <= "0010" when "01";

Y <= "0100" when "10";

Y <= "1000" when "11";

end if;

ENbar <= not Y;

EN <= not ENbar; S <= D (2 down to 0);

if (EN(0) <= '1')

with S select

Y\_L(0) <= D(0) when "000";

(54)

$Y_L(0) = D(1)$  when '001';

$Y_L(0) = D(2)$  when '010';

$Y_L(0) = D(3)$  when '011';

$Y_L(0) \leftarrow D(4)$  when '100';

$Y_L(0) \leftarrow D(5)$  when '101';

$Y_L(0) \leftarrow D(6)$  when '110';

$Y_L(0) \leftarrow D(7)$  when '111';

else of ( $EN(1) \leq 1$ )

with S select

$Y_L(1) \leftarrow D(8)$  when '000';

$Y_L(1) \leftarrow D(9)$  when '001';

$Y_L(1) \leftarrow D(10)$  when '010';

⋮ ⋮

⋮ ⋮

$Y_L(1) \leftarrow D(15)$  when '111';

else of ( $EN(2) \leq 1$ )

with S select

$Y_L(2) \leftarrow D(16)$ , when '000';

⋮ ⋮

$Y_L(2) \leftarrow D(23)$  when '111';

else

with S select

$Y_L(3) \leftarrow D(24)$  when '000';

⋮ ⋮

$Y_L(3) \leftarrow D(31)$  when '111';

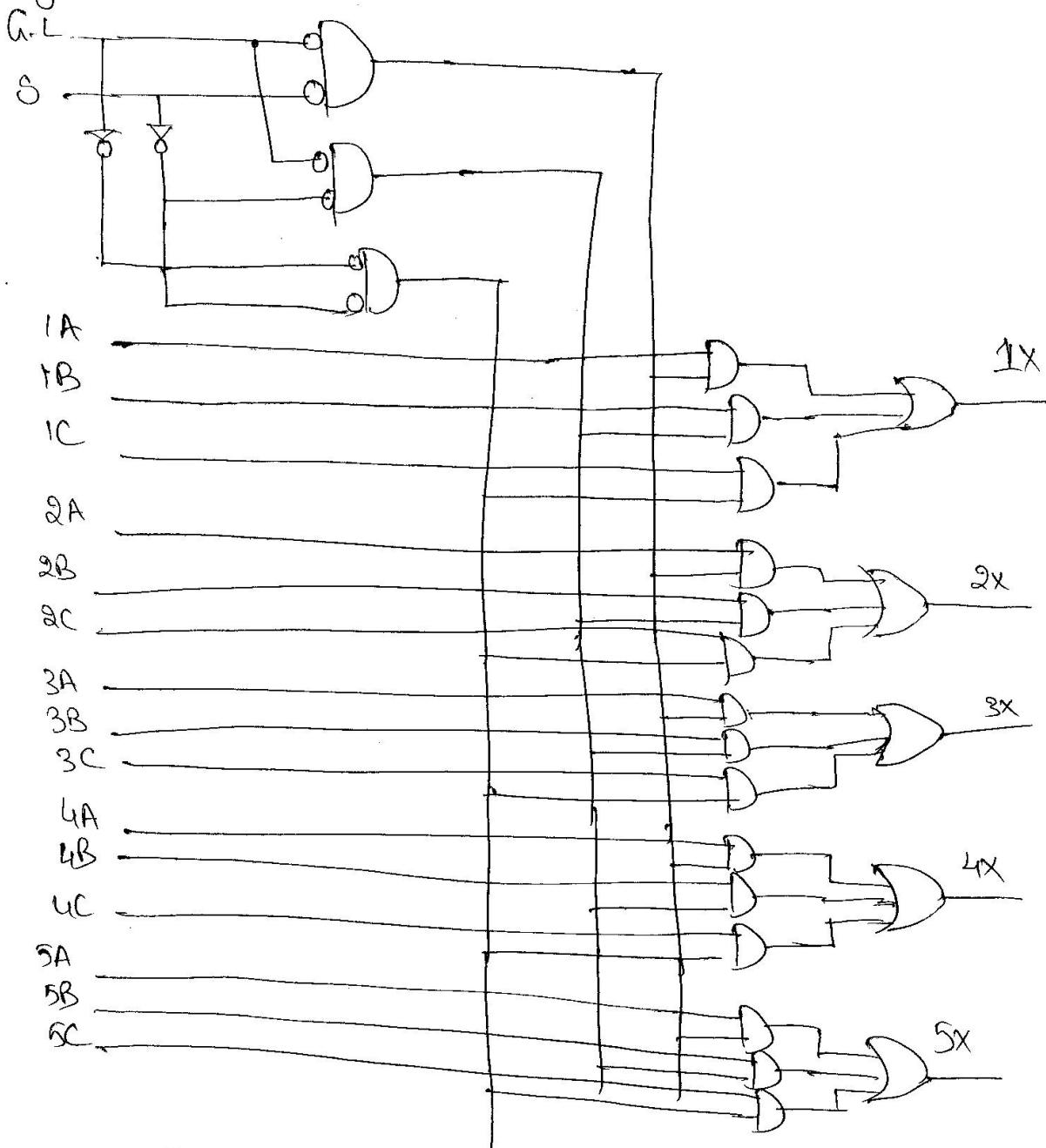
end of;

$Q \leftarrow (Y_L(3) \text{ or } Y_L(2) \text{ or } Y_L(1) \text{ or } Y_L(0));$

End Mux 32x1;

\* Design 3 input 5 bit Mux . write the truth table, and draw the logic diagram Provide the dataflow VHDL Program.

(55)



G-L	S	1x	2x	3x	4x	5x
0	0	IA	2A	3A	4A	5A
0	1	IB	2B	3B	4B	5B
1	0	0	0	0	0	0
1	1	IC	2C	3C	4C	5C

(56)

VHDL Program :-

library IEEE;

use IEEE.STD.LOGIC\_1164.ALL;

entity MUX is

port (S : in STD\_LOGIC\_VECTOR (2 downto 0);

A,B,C : in STD\_LOGIC\_VECTOR (1 to 5);

G,L : in STD\_LOGIC;

X : out STD\_LOGIC\_VECTOR (5 downto 1);

end mux;

Architectural dataflow of MUX is

$X \leftarrow A$  when "00";

$X \leftarrow B$  when "01";

$X \leftarrow C$  when "10";

$X \leftarrow D$  when "11";

end dataflow; //.

### \* Code Converters:-

A Code Converter is a logic circuit which changes data present in one type of binary code to another type of binary code. The 9/10 and 10/10 codes written in form of truth table.

K-map

1) BCD to Excess 3 Code Converter

2) Binary to Gray Code Converter & Gray to Binary Converter

3) Excess 3 to BCD Code Converter

4) BCD to Gray Code Converter