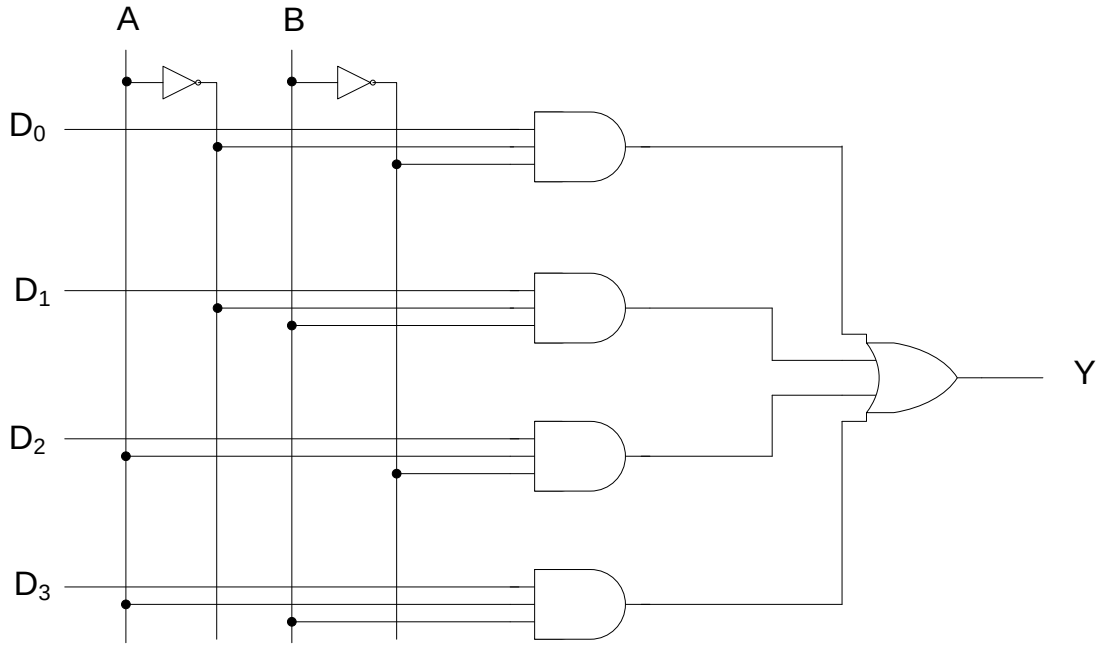
Q1. What is a multiplexer? Design a 4 to 1 multiplexer using logic gates. Write the truth table and explain its working principle.

Answer: Multiplexer is a circuit with many inputs but only one output.

Designing of 4 to 1 multiplexer shown below:



Truth table for 4 to 1 MUX:

A	B	Y
0	0	D_0
0	1	D_1
1	0	D_2
1	1	D_3

Working principle of 4 to 1 multiplexer:

From the above diagram, the Logic Equation for 4 to 1 multiplexer is

$Y = A'B'D_0 + A'BD_1 + AB'D_2 + ABD_3$

$If\ A=0,\ B=0\ then,\ Y = 0'0'D_0 + 0'0D_1 + 00'D_3 + 00D_3 = 1.1.D_0 + 1.0.D_1 + 0.1.D_2 + 0.0.D_3 = D_0$

Similarly, if A=0 and B=1 then $Y=D_1$, if A=1 and B=0 then $Y=D_2$ and , if A=0 and B=1 then $Y=D_3$

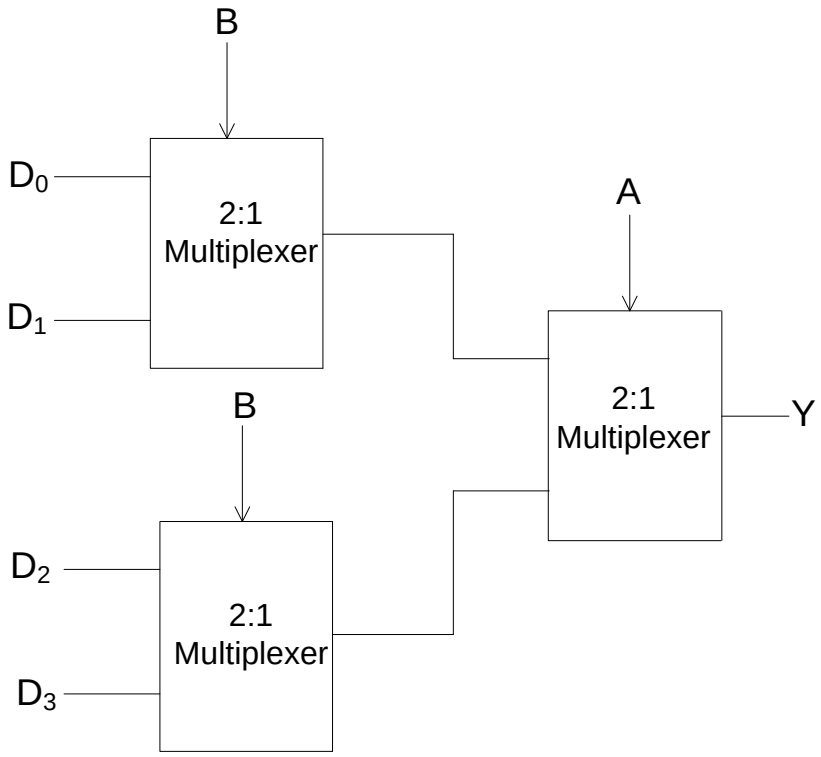
Q2. Construct 4:1 multiplexer using only 2:1 multiplexer.

Answer: *Logic Equation for* $2:1$ *Multiplexer is* $Y = A'D_0 + AD_1$

Logic equation for $4:1$ *Multplexer is* $Y = A'B'D_0 + A'BD_1 + AB'D_2 + ABD_3$

$\Rightarrow Y = A'(B'D_0 + BD_1) + A(B'D_2 + BD_3)$

We require three 2:1 Multiplexers and the connection is shown below.



Q3. Construct 8:1 multiplexer using only 2:1 multiplexer.

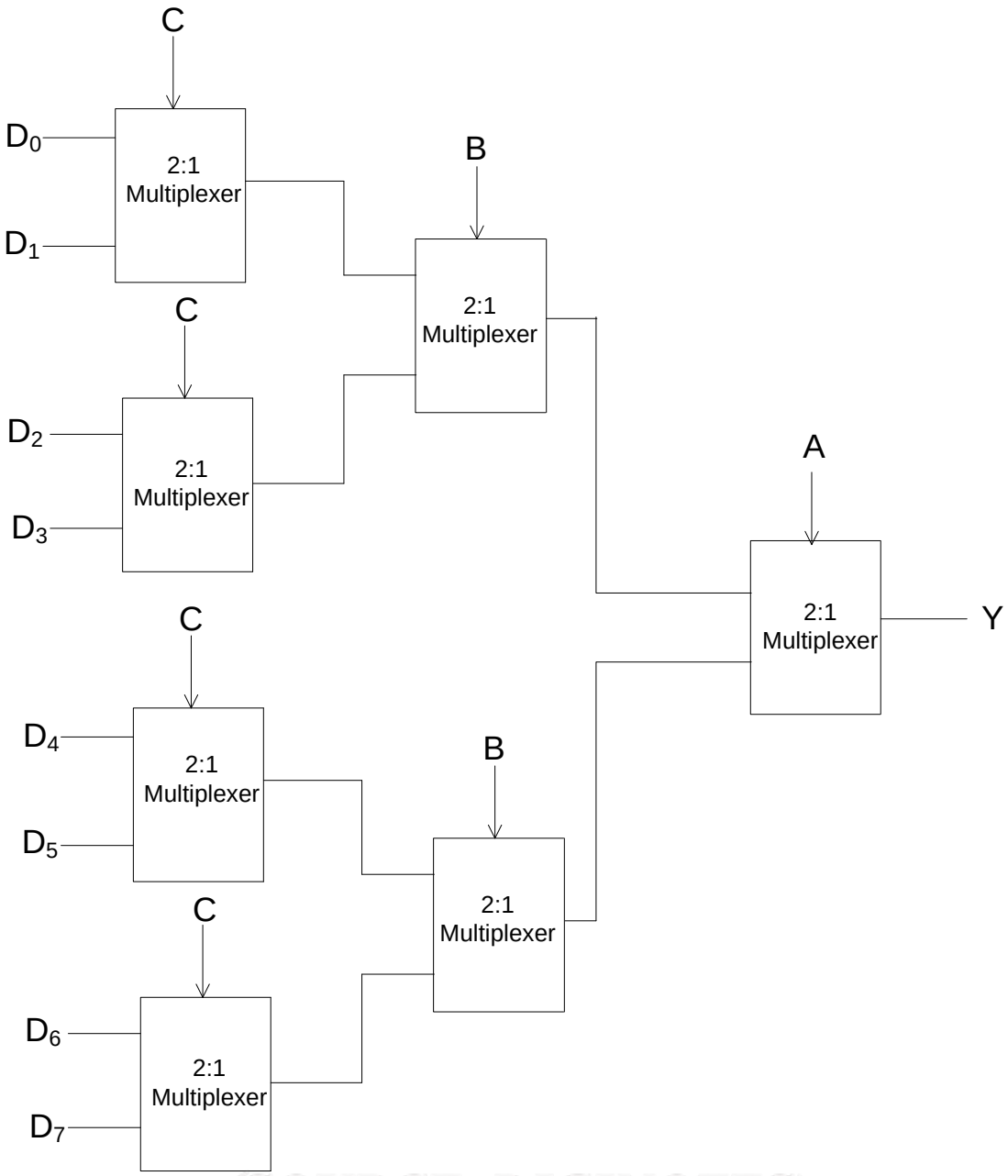
Answer: *Logic Equation for* $2:1$ *Multiplexer is* $Y = A'D_0 + AD_1$

Logic Equation for $8:1$ *Multiplexer is*

$Y = A'B'C'D_0 + A'B'CD_1 + A'BC'D_2 + A'BCD_3 + AB'C'D_4 + AB'CD_5 + ABC'D_6 + ABCD_7$

$\Rightarrow Y = A'(B'C'D_0 + B'CD_1 + BC'D_2 + BCD_3) + A(B'C'D_4 + B'CD_5 + BC'D_6 + BCD_7)$

$\Rightarrow Y = A'[B'(C'D_0 + CD_1) + B(C'D_2 + CD_3)] + A[B'(C'D_4 + CD_5) + B(C'D_6 + CD_7)]$



Q4. Design 16 to 1 multiplexer using 8 to 1 multiplexer and one 2 to 1 multiplexer.

Answer: *Logic Equation for* $2:1$ *Multiplexer is* $Y = A'D_0 + AD_1$

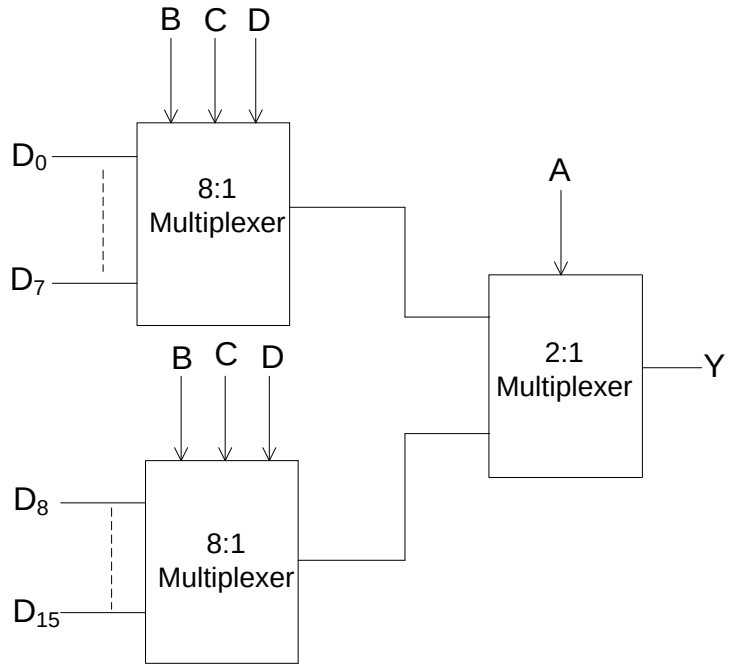
Logic Equation for $8:1$ *Multiplexer is*

$$Y = A'B'C'D_0 + A'B'CD_1 + A'BC'D_2 + A'BCD_3 + AB'C'D_4 + AB'CD_5 + ABC'D_6 + ABCD_7$$

Logic Equation for $16:1$ *Multiplexer is*

$$Y = A'B'C'D'D_0 + A'B'C'DD_1 + ... + A'BCD'D_6 + A'BCDD_7 + AB'C'D'_8 + AB'C'D_9 + ... + ABCD'D_{14} + ABCDD_{15}$$

$$\Rightarrow Y = A'(B'C'D'D_0 + B'C'DD_1 + ... + BCD'D_6 + BCDD_7) + A(B'C'D'_8 + B'C'D_9 + ... + BCD'D_{14} + BCDD_{15})$$



Q5. Design 32 to 1 multiplexer using 16 to 1 multiplexer and one 2 to 1 multiplexer.

Answer: *Logic Equation for* $2:1$ *Multiplexer is* $Y = A'D_0 + AD_1$

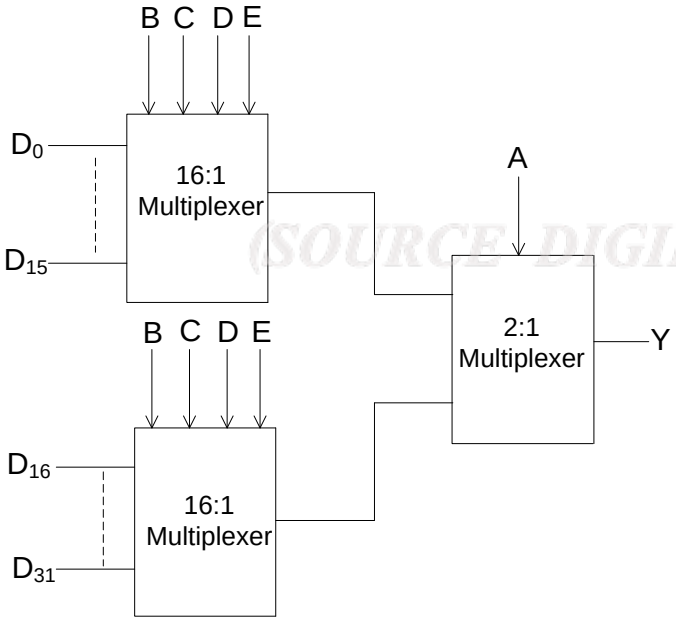
Logic Equation for $16:1$ *Multiplexer is*

$Y = A'B'C'D'D_0 + A'B'C'DD_1 + ... + ABCD'D_{14} + ABCDD_{15}$

Logic Equation for $32:1$ *Multiplexer is*

$Y = A'B'C'D'E'D_0 + ... + A'BCDE'D_{14} + A'BCDED_{15} + AB'C'D'E'D_{16} + ... + ABCDE'D_{30} + ABCDED_{31}$

$\Rightarrow Y = A'(B'C'D'E'D_0 + ... + BCDE'D_{14} + BCDED_{15}) + A(B'C'D'E'D_{16} + ... + BCDE'D_{30} + BCDED_{31})$



Q6. Mention the differences between decoder and demultiplexer.

Answer:

Demultiplexer	Decoder
There is one data input and multiple output. There are selects used as control bits.	There is no data input. The only inputs are the control bit.
The data input appears at one of the output as per the control inputs.	One of the output is high as per the control inputs.
Input appears at the output where subscription of the output is equal to the decimal equivalent to the inputs.	Output becomes high where subscription of the output is equal to the decimal equivalent to the inputs

Q7. (a) Realize $Y = A'B + B'C' + ABC$ using an 8 to 1 Multiplexer.

(b) Can it be realized with a 4 to 1 multiplxer?

Answer : (a) *Logic Equation for* $8:1$ *Multiplexer is*

$$Y = A'B'C'D_0 + A'B'CD_1 + A'BC'D_2 + A'BCD_3 + AB'C'D_4 + AB'CD_5 + ABC'D_6 + ABCD_7$$

We should express Y as a function of three variables i.e function of minterms.

$Y = A'B + B'C' + ABC$

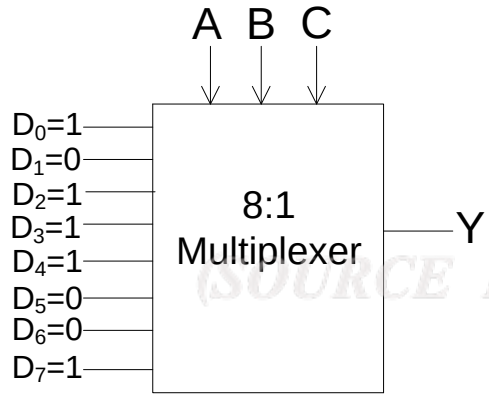
$\Rightarrow Y = A'B(C + C') + B'C'(A + A') + ABC$

$\Rightarrow Y = A'BC + A'BC' + AB'C' + A'B'C' + ABC$

$\Rightarrow Y = A'B'C' + A'BC' + A'BC + AB'C' + ABC$

Comparing with the Logic equation of 8:1 Multiplexer, we have

$D_0 = D_2 = D_3 = D_4 = D_7 = 1$ *and* $D_1 = D_5 = D_6 = 0$



$(b) Y = A'B + B'C' + ABC$

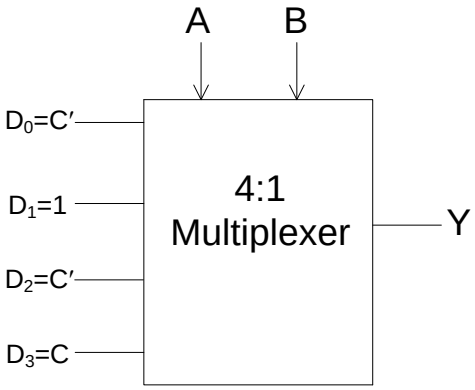
$\Rightarrow Y = A'B + B'C'(A + A') + ABC$

$\Rightarrow Y = A'B + AB'C' + A'B'C' + ABC$

$\Rightarrow Y = A'B'.C' + A'B.1 + AB'.C' + AB.C$

Logic equation for $4:1$ *Multplexer is* $Y = A'B'D_0 + A'BD_1 + AB'D_2 + ABD_3$

We have, $D_0 = C'$, $D_1 = 1$, $D_2 = C'$ *and* $D_3 = C$



Q8. Implement the following Boolean functions using 4:1 multiplexer (MUX):

$(i) Y = f(A,B,C,D) = \sum m(0,1,2,4,6,9,12,14)$

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

Answer: *Logic equation for* $4:1$ *Multplexer is* $Y = A'B'D_0 + A'BD_1 + AB'D_2 + ABD_3$

$Y = f(A,B,C,D) = \sum m(0,1,2,4,6,9,12,14)$

$\Rightarrow Y = A'B'C'D' + A'B'C'D + A'B'CD' + A'BC'D' + A'BCD' + AB'C'D + ABC'D' + ABCD'$

$\Rightarrow Y = A'B'(C'D' + C'D + CD') + A'B(C'D' + CD') + AB'C'D + AB(C'D' + CD')$

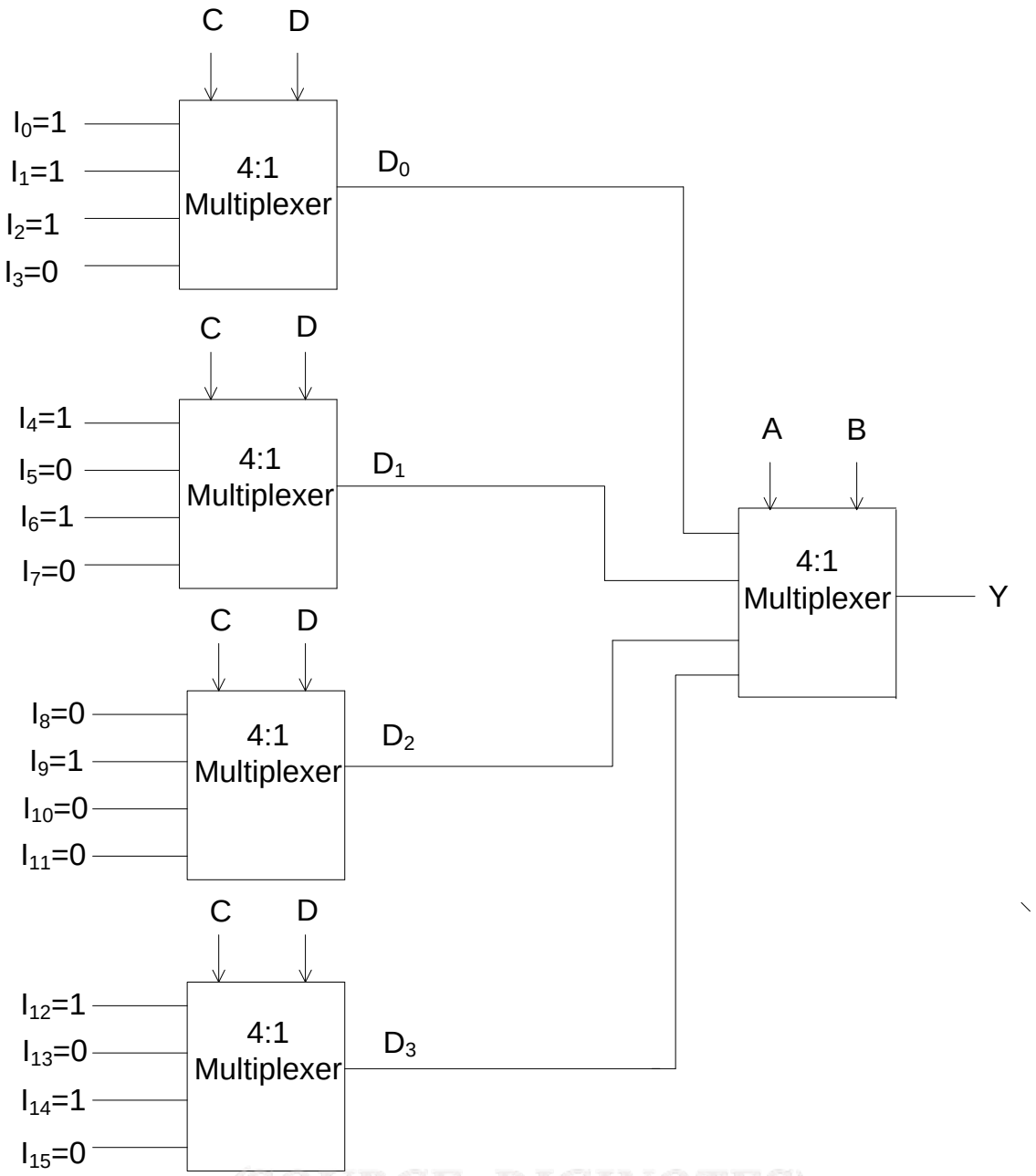
Comparing with Logic Equation of $4:1$ *Multiplexer, We have*

$D_0 = C'D' + C'D + CD' \Rightarrow D_0 = C'D'.1 + C'D.1 + CD'.1 + CD.0$

$D_1 = C'D' + CD' \Rightarrow D_1 = C'D'.1 + C'D.0 + CD'.1 + CD.0$

$D_2 = C'D \Rightarrow D_2 = C'D'.0 + C'D.1 + CD'.0 + CD.0$

$D_3 = C'D' + CD' \Rightarrow D_3 = C'D'.1 + C'D.0 + CD'.1 + CD.0$



(ii) *Logic equation for* $4:1$ *Multplexer is* $Y = A'B'D_0 + A'BD_1 + AB'D_2 + ABD_3$

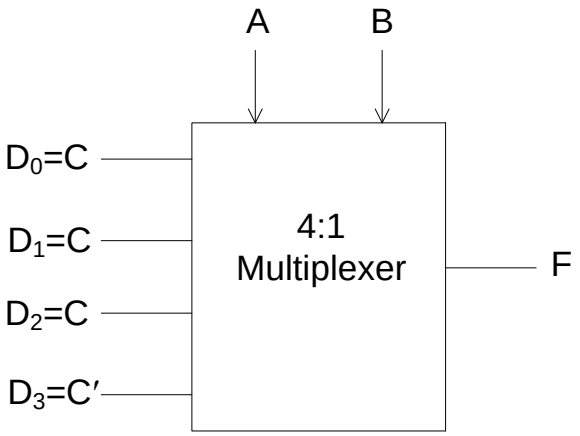
$F(A,B,C) = \sum m(1,3,5,6)$

$\Rightarrow F = A'B'C + A'BC + AB'C + ABC'$

$\Rightarrow F = A'B'.C + A'B.C + AB'.C + AB.C'$

Comparing with Logic Equation of $4:1$ *Multiplexer*, we *have*

$D_0 = C,\ D_1 = C,\ D_2 = C$ *and* $D_3 = C'$



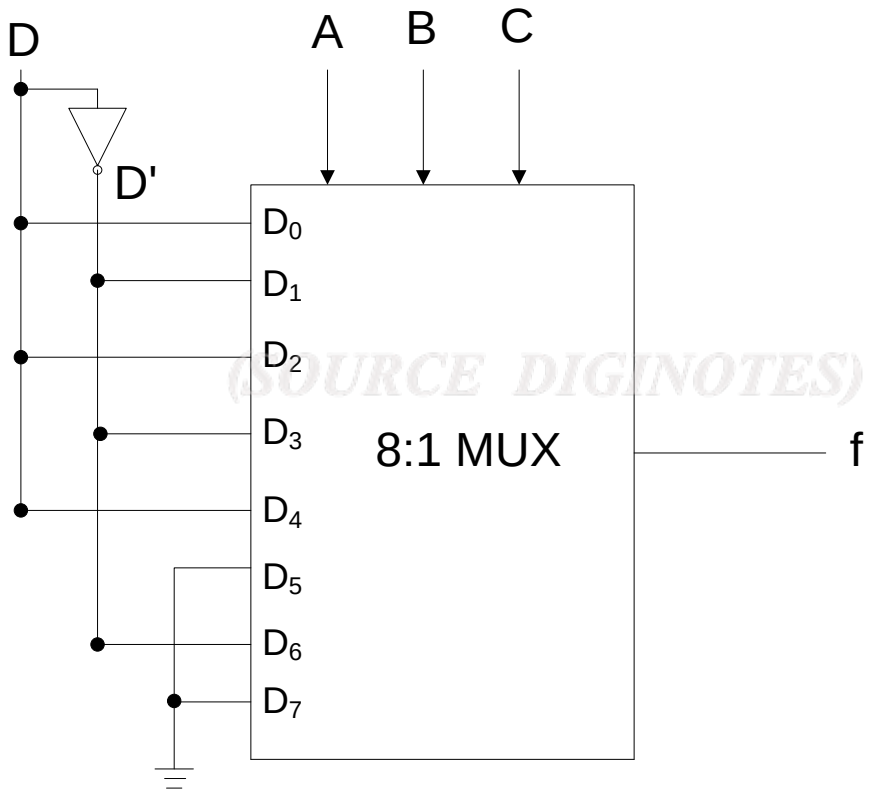
Q9. Implement the Boolean function expressed by SOP:

$f(A,B,C,D) = \sum m(1,2,5,6,9,12)$ using 8 to 1 MUX.

Answer:

ABC	000	001	010	011	100	101	110	111
D=0	0	1	0	1	0	0	1	0
D=1	1	0	1	0	1	0	0	0
f	1	0	d	d'	d'	d'	d'	d
8:1 MUX data input	D_0=D	D_1= D'	D_2=D	D_3=D'	D_4= D	D_5= 0	D_6= D'	D_7=0

Circuit diagram:



Q10. Implement the Boolean function:

$F(A,B,C,D) = \sum m(0,1,2,4,5,7,8,9)$ using 8 to 1 multiplexers. Draw the logic diagram and explain the operation. Additional gates can be used if required.

Answer : (*a*) *Logic Equation for* $8:1$ *Multiplexer is*

$Y = A'B'C'D_0 + A'B'CD_1 + A'BC'D_2 + A'BCD_3 + AB'C'D_4 + AB'CD_5 + ABC'D_6 + ABCD_7$

Logic Equation for $2:1$ *Multiplexer is* $Y = A'D_0 + AD_1$

$F(A,B,C,D) = \sum m(0,1,2,4,5,7,8,9)$

$\Rightarrow F = A'B'C'D' + A'B'C'D + A'B'CD' + A'BC'D' + A'BC'D + A'BCD + AB'C'D' + AB'C'D$

$\Rightarrow F = A'(B'C'D' + B'C'D + B'CD' + BC'D' + BC'D + BCD) + A(B'C'D' + B'C'D)$

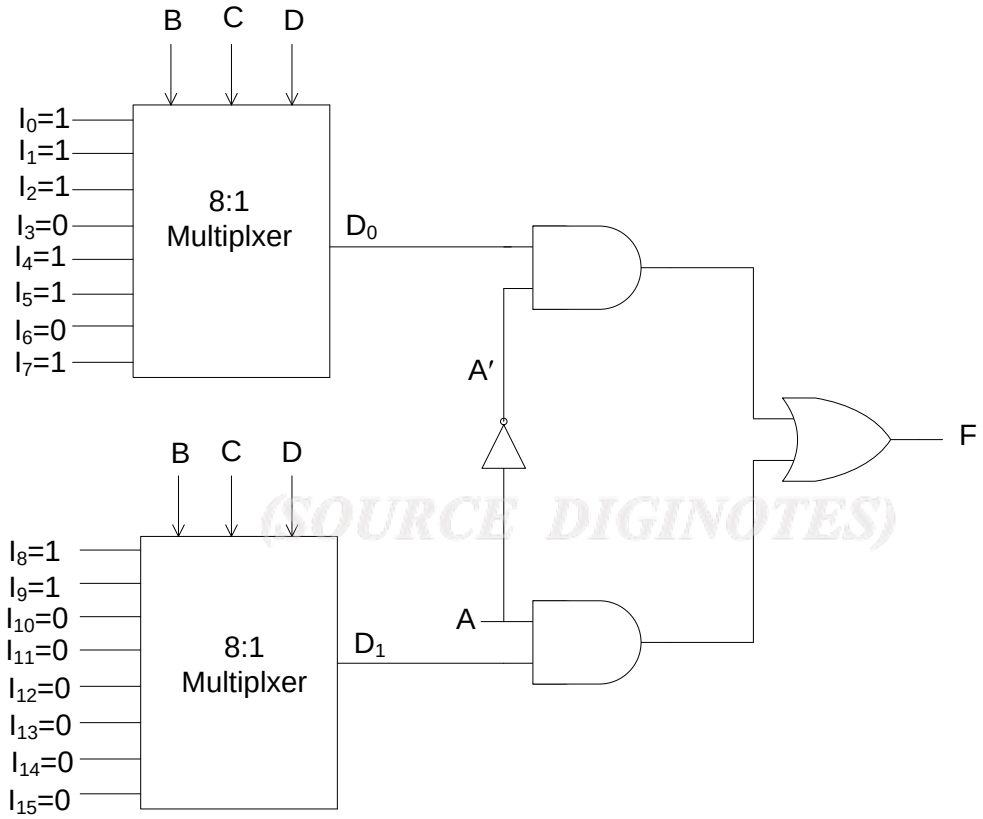
Comparing with $2:1$ *Logic Equation, we have*

$D_0 = B'C'D' + B'C'D + B'CD' + BC'D' + BC'D + BCD$

$\Rightarrow D_0 = B'C'D'.1 + B'C'D.1 + B'CD'.1 + B'CD.0 + BC'D'.1 + BC'D.0 + BCD'.0 + BCD.1$

$D_1 = B'C'D' + B'C'D$

$\Rightarrow D_1 = B'C'D'.1 + B'C'D.1 + B'CD'.0 + B'CD.0 + BC'D'.0 + BC'D.0 + BCD'.0 + BCD.0$



(Note: Q9 and Q10 are similar. But method for Q9 is preferable)

Q11. Realize the following Boolean function:

$P = f(w,x,y,z) = \sum(0,1,5,6,7,10,15)$ using (i) 16:1 MUX (ii) 8:1 MUX (iii) 4:1 MUX

Answer : (*i*)*Logic Equation for* 16:1 *Multiplexer is*

$$P = w'x'y'z'.D_0 + w'x'y'z.D_1 + w'x'yz'.D_2 + w'x'yz.D_3 + w'xy'z'.D_4 + w'xy'z.D_5 + w'xyz'.D_6 + w'xyz.D_7 + wx'y'z'.D_8$$
$$+wx'y'z.D_9 + wx'yz'.D_{10} + wx'yz.D_{11} + wxy'z'.D_{12} + wxy'z.D_{13} + wxyz'.D_{14} + wxyz.D_{15}$$

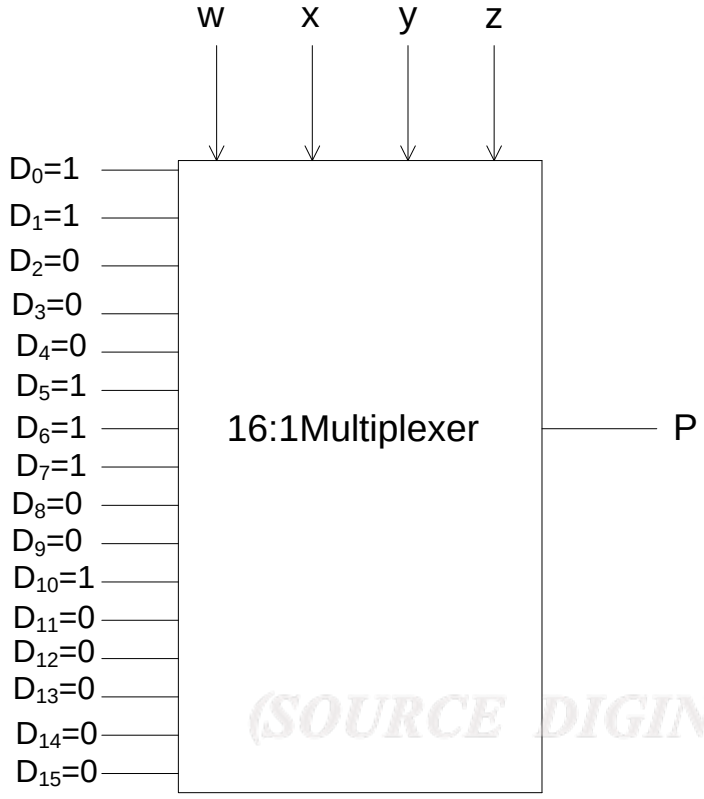
$P = f(w,x,y,z) = \sum m(0,1,5,6,7,10,15)$

$\Rightarrow P = w'x'y'z' + w'x'y'z + w'xy'z + w'xyz' + w'xyz + wx'yz' + wxyz$

$\Rightarrow P = w'x'y'z'.1 + w'x'y'z.1 + w'x'yz'.0 + w'x'yz.0 + w'xy'z'.0 + w'xy'z.1 + w'xyz'.1 + w'xyz.1 + wx'y'z'.0$
$+wx'y'z.0 + wx'yz'.1 + wx'yz.0 + wxy'z'.0 + wxy'z.0 + wxyz'.0 + wxyz.1$

Comparing with Logic Equation for 16:1 *Multiplexer, we have*

$D_0 = D_1 = D_5 = D_6 = D_7 = D_{10} = D_{15} = 1$ *and* $D_2 = D_3 = D_4 = D_8 = D_9 = D_{11} = D_{12} = D_{13} = D_{14} = 0$



(*ii*)*Logic Equation for* 8:1 *Multiplexer is*

$$Y = x'y'z'.I_0 + x'y'z.I_1 + x'yz'.I_2 + x'yz.I_3 + xy'z'.I_4 + xy'z.I_5 + xyz'.I_6 + xyz.I_7$$

Logic Equation for 2:1 *Multiplexer is* $Y = w'D_0 + wD_1$

$$P = f(w,x,y,z) = \sum m(0,1,5,6,7,10,15)$$

$$\Rightarrow P = w'x'y'z' + w'x'y'z + w'xy'z + w'xyz' + w'xyz + wx'yz' + wxyz$$

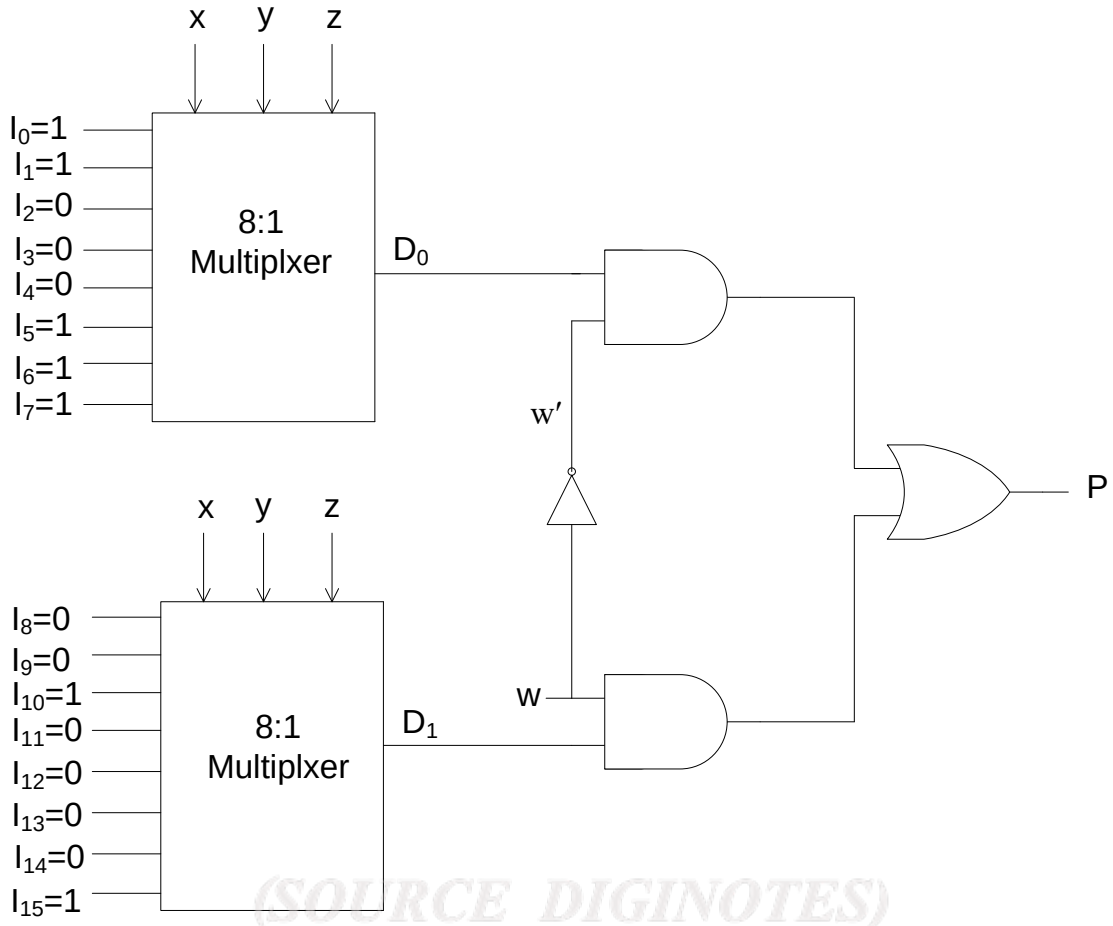
$$\Rightarrow P = w'(x'y'z' + x'y'z + xy'z + xyz' + xyz) + w(x'yz' + xyz)$$

$$\Rightarrow P = w'(x'y'z'.1 + x'y'z.1 + x'yz'.0 + x'yz.0 + xy'z'.0 + xy'z.1 + xyz'.1 + xyz.1)$$
$$+ w(x'y'z'.0 + x'y'z.0 + x'yz'.1 + x'yz.0 + xy'z'.0 + xy'z.0 + xyz'.0 + xyz.1)$$

Comparing with Logic Equation of 2:1 *Multiplexer, we have*

$$D_0 = x'y'z'.1 + x'y'z.1 + x'yz'.0 + x'yz.0 + xy'z'.0 + xy'z.1 + xyz'.1 + xyz.1$$

$$D_1 = x'y'z'.0 + x'y'z.0 + x'yz'.1 + x'yz.0 + xy'z'.0 + xy'z.0 + xyz'.0 + xyz.1$$



(iii) *Logic equation for* $4:1$ *Multplexer is* $Y = w'x'D_0 + w'xD_1 + wx'D_2 + wxD_3$

$P = f(w, x, y, z) = \sum m(0,1,5,6,7,10,15)$

$\Rightarrow P = w'x'y'z' + w'x'y'z + w'xy'z + w'xyz' + w'xyz + wx'yz' + wxyz$

$\Rightarrow P = w'x'(y'z' + y'z) + w'x(y'z + yz' + yz) + wx'yz' + wxyz$

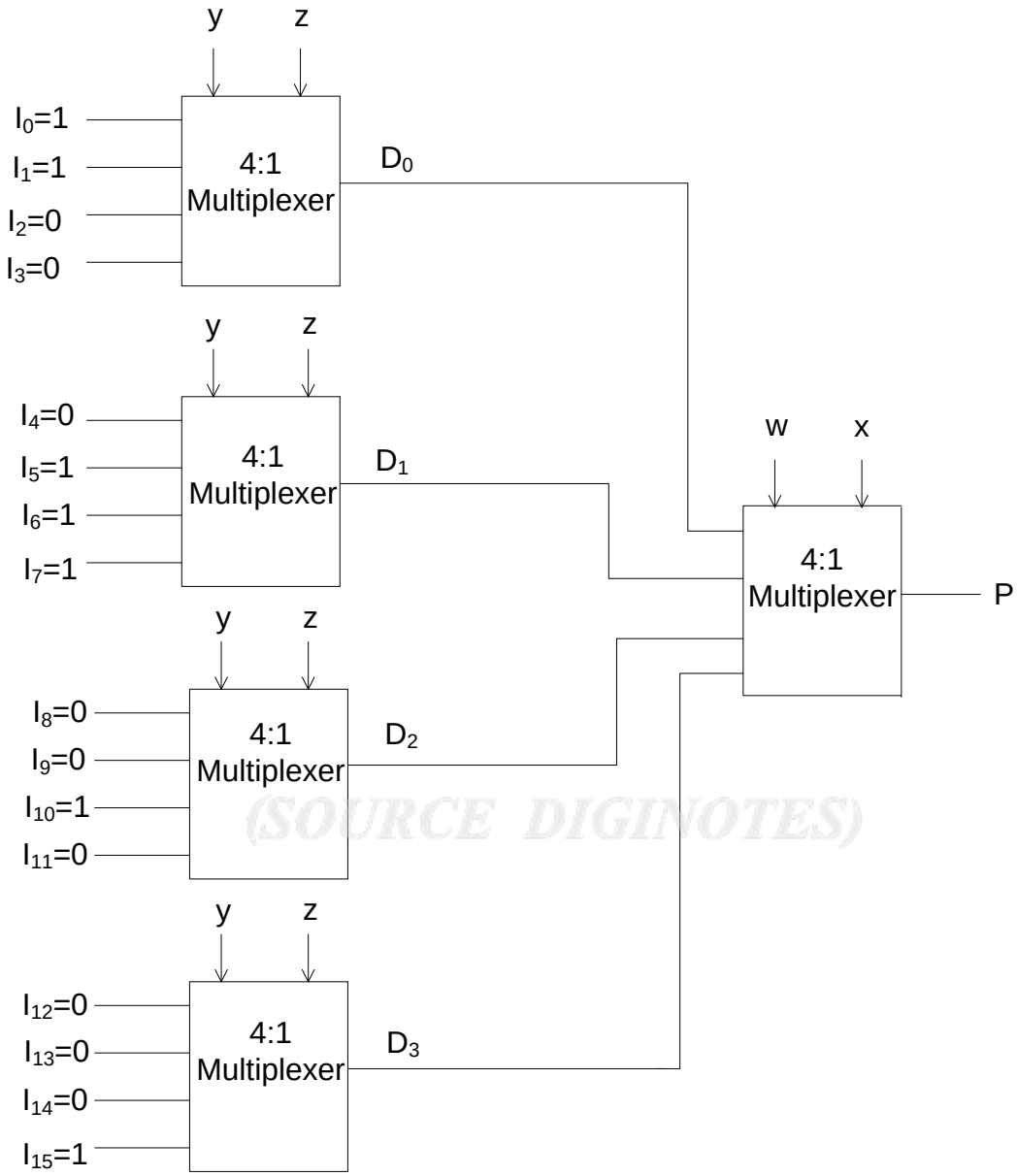
Comparing with Logic Equation for 4:1 *Multiplexer, we have*

$D_0 = y'z' + y'z \Rightarrow D_0 = y'z'.1 + y'z.1 + yz'.0 + yz.0$

$D_1 = y'z + yz' + yz \Rightarrow D_1 = y'z'.0 + y'z.1 + yz'.1 + yz.1$

$D_2 = yz' = y'z'.0 + y'z.0 + yz'.1 + yz.0$

$D_3 = yz = y'z'.0 + y'z.0 + yz'.0 + yz.1$



Q12. Design and implement BCD to excess-3 code converter using four 8:1 multiplexers. Take MSB 'A' as map entered variable(input variable) 'BCD' lines as select lines, assuming f(A,B,C,D) as BCD input.

Answer: Truth table for converting BCD to Excess-3

BCD				Excess-3			
A	B	C	D	W	X	Y	Z
0	0	0	0	0	0	1	1
0	0	0	1	0	1	0	0
0	0	1	0	0	1	0	1
0	0	1	1	0	1	1	0
0	1	0	0	0	1	1	1
0	1	0	1	1	0	0	0
0	1	1	0	1	0	0	1
0	1	1	1	1	0	1	0
1	0	0	0	1	0	1	1
1	0	0	1	1	1	0	0

Designing of the multiplexer whose output is W

BCD	000	001	010	011	100	101	110	111
A=0	0	0	0	0	0	1	1	1
A=1	1	1	X	X	X	X	X	X
W	A	A	0	0	0	1	1	1
8:1 MUX Data Input	D_0=A	D_1=A	D_2=0	D_3=0	D_4=0	D_5=1	D_6=1	D_7=1

Designing of the multiplexer whose output is X

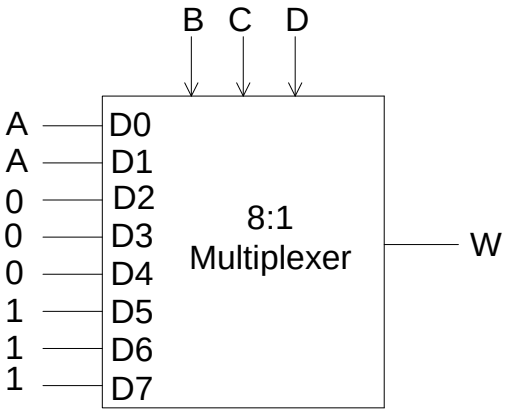
BCD	000	001	010	011	100	101	110	111
A=0	0	1	1	1	1	0	0	0
A=1	0	1	X	X	X	X	X	X
X	0	1	1	1	1	0	0	0
8:1 MUX Data Input	D_0=0	D_1=1	D_2=1	D_3=1	D_4=1	D_5=0	D_6=0	D_7=0

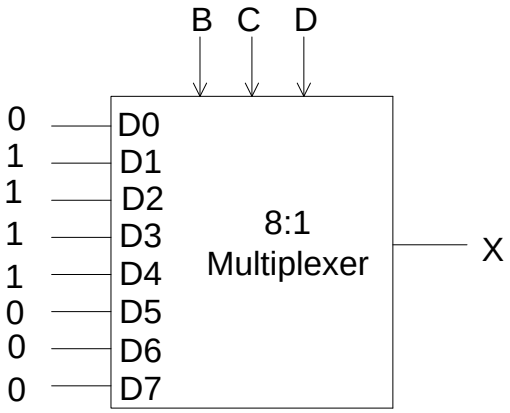
Designing of the multiplexer whose output is Y

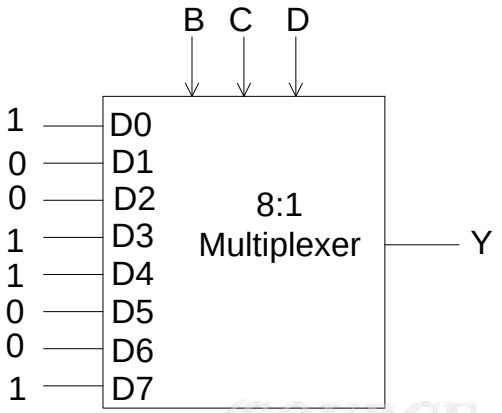
BCD	000	001	010	011	100	101	110	111
A=0	1	0	0	1	1	0	0	1
A=1	1	0	X	X	X	X	X	X
Y	1	0	0	1	1	0	0	1
8:1 MUX Data Input	D_0=1	D_1=0	D_2=0	D_3=1	D_4=1	D_5=0	D_6=0	D_7=1

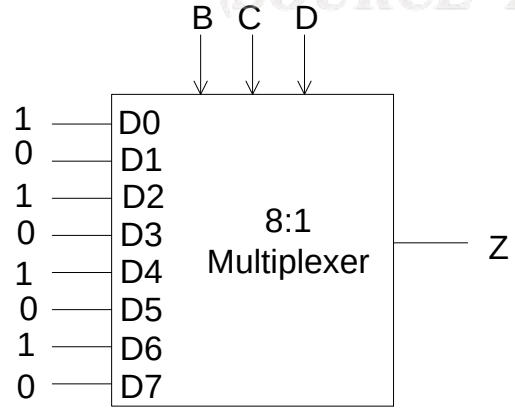
Designing of the multiplexer whose output is Z

BCD	000	001	010	011	100	101	110	111
A=0	1	0	1	0	1	0	1	0
A=1	1	0	X	X	X	X	X	X
Z	1	0	1	0	1	0	1	0
Data Input	D_0=1	D_1=0	D_2=1	D_3=0	D_4=1	D_5=0	D_6=1	D_7=0





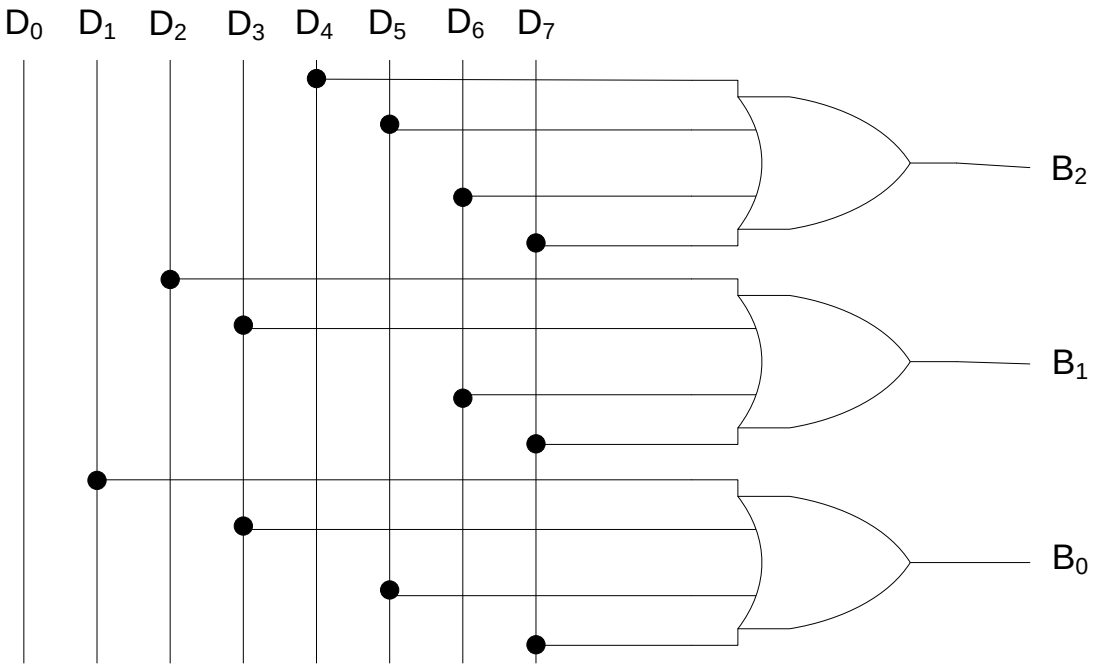




Q13. Realize a logic circuit for octal to binary encoder.

Answer: Truth table for octal to binary encoder

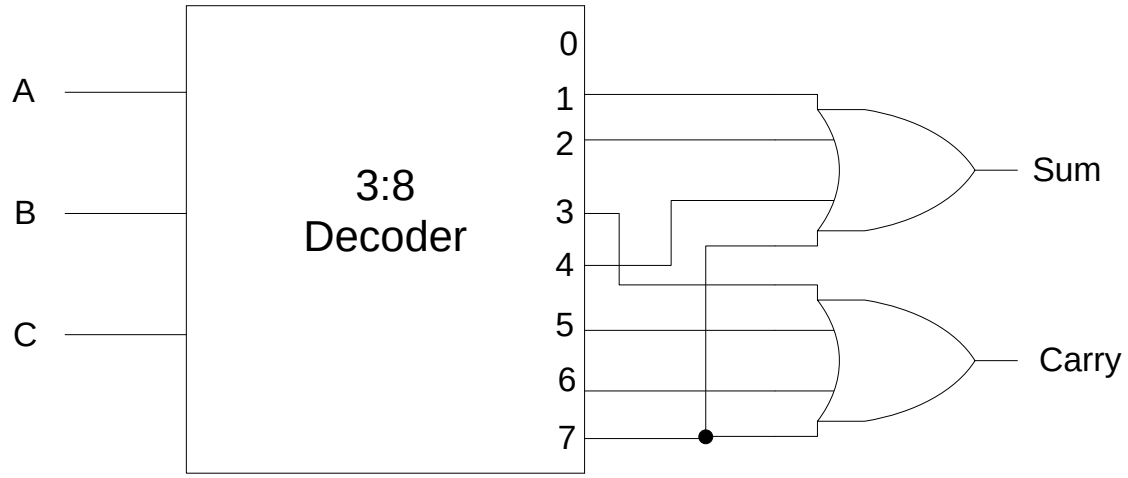
Input								Output		
D_0	D_1	D_2	D_3	D_4	D_5	D_6	D_7	B_2	B_1	B_0
1	0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0	1
0	0	1	0	0	0	0	0	0	1	0
0	0	0	1	0	0	0	0	0	1	1
0	0	0	0	1	0	0	0	1	0	0
0	0	0	0	0	1	0	0	1	0	1
0	0	0	0	0	0	1	0	1	1	0
0	0	0	0	0	0	0	1	1	1	1



Q14. Implement a full adder using a 3 to 8 decoder.

Answer: Truth table for full adder

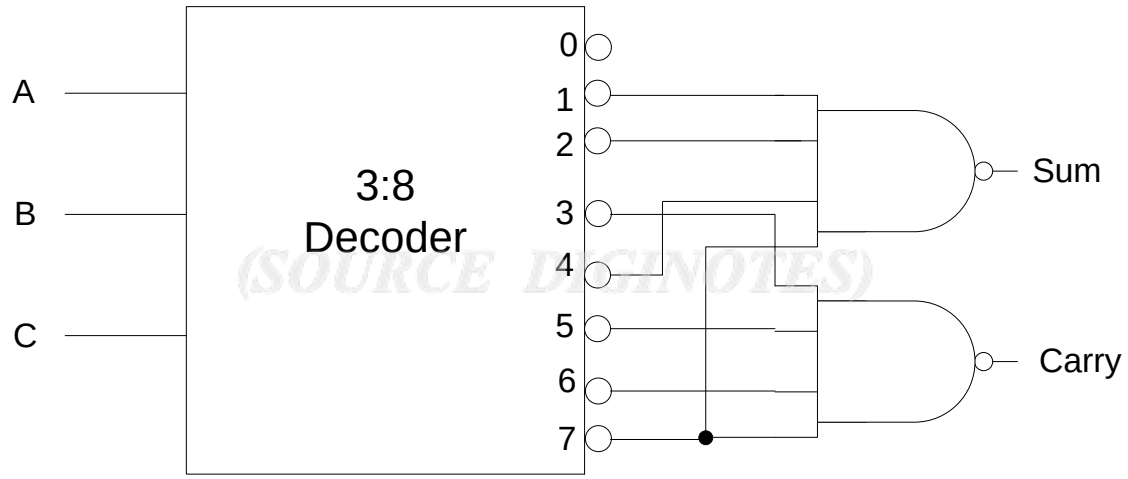
A	B	C	Sum	Carry
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1



Q15. Implement full adder using IC 74138

Answer: Truth table for full adder

A	B	C	Sum	Carry
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1



Q16. Implement 3 bit binary to gray code conversion by using IC 74139.

Answer: Truth table for converting 3 bit binary to gray code

Binary code			Gray code		
A	B	C	X	Y	Y
0	0	0	0	0	0
0	0	1	0	0	1
0	1	0	0	1	1
0	1	1	0	1	0
1	0	0	1	1	0
1	0	1	1	1	1
1	1	0	1	0	1
1	1	1	1	0	0

K-Map for X:

AB \ C	0	1
00	0	0
01	0	0
11	1	1
10	1	1

K-Map for Y:

AB \ C	0	1
00	0	0
01	1	1
11	0	0
10	1	1

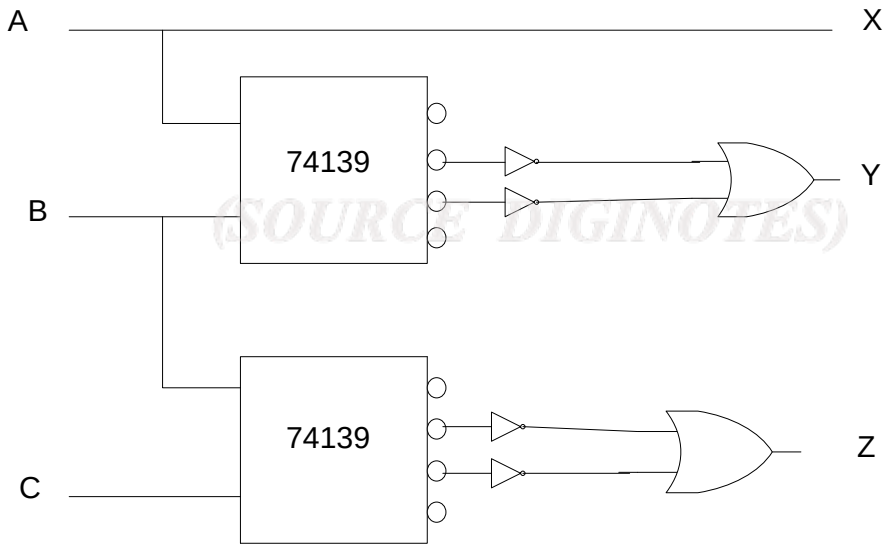
K-Map for Z:

AB \ C	0	1
00	0	1
01	1	0
11	1	0
10	0	1

X=A

$Y=\bar{A}B+A\bar{B}$

$Z=B\bar{C}+\bar{B}C$

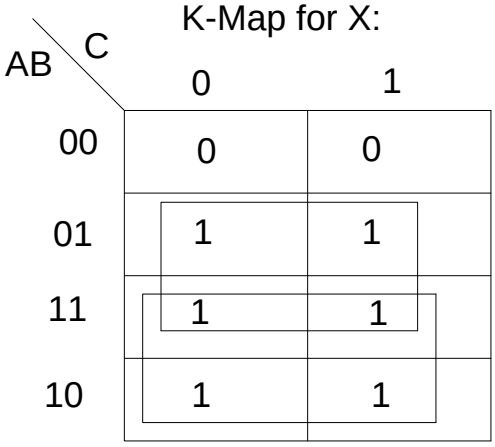


Q17. Design a priority encoder for a system with a 3 inputs, the middle bit with highest priority encoding to 10, the MSB with the next priority encoding to 11, while the LSB with least priority encoding to 01.

Answer: Truth table of the priority encoder

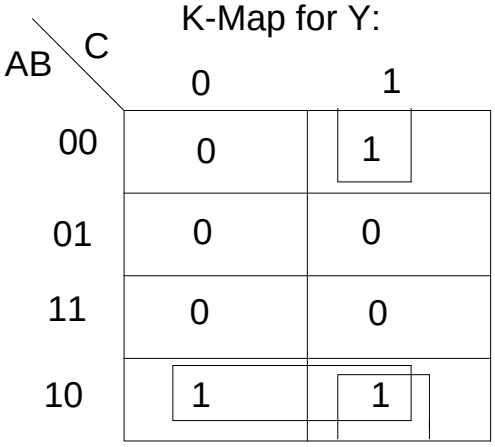
Input			Output	
A	B	C	X	Y
0	0	0	0	0
0	0	1	0	1
0	1	0	1	0
0	1	1	1	0
1	0	0	1	1
1	0	1	1	1
1	1	0	1	0
1	1	1	1	0

K-Map for X:

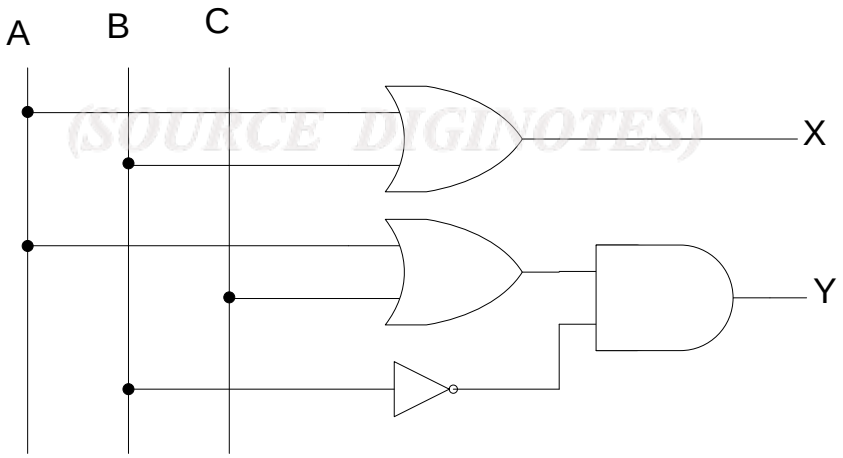


$X=A+B$

K-Map for Y:

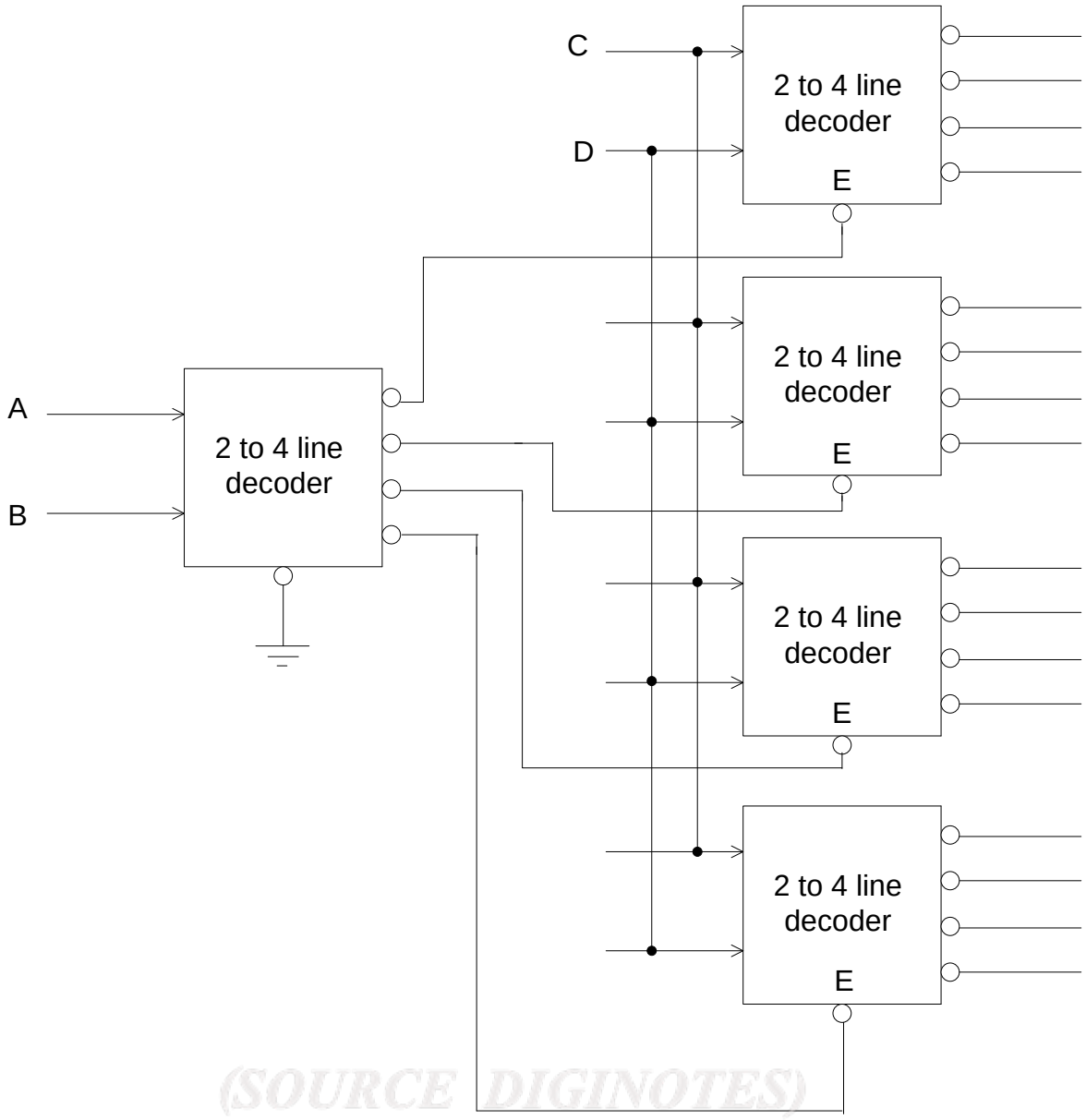


$Y=A\overline{B}+\overline{B}C=\overline{B}(A+C)$



Q18. Design a 4 to 16 line decoder using 2 to 4 line decoder which has the active low outputs as active low enable input. Explain its operation.

Answer:



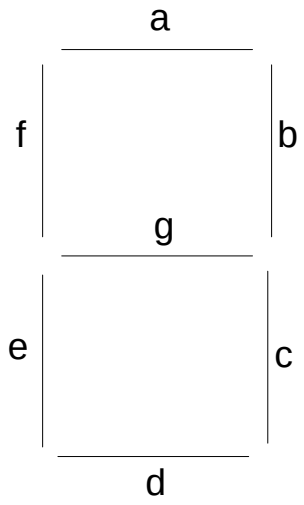
Q19. Write the comparisons between PLA and PAL.

Answer:

PAL	PLA
The output OR-gate array is fixed while the input AND gate array is fusible linked and thus programmable.	Both output OR-gate array and input AND gate array are fusible linked.
PAL is easier to program.	PLA is more complicated since the number fusible links are more compared to PAL
PAL is less expensive.	PLA is more expensive compared to PAL.

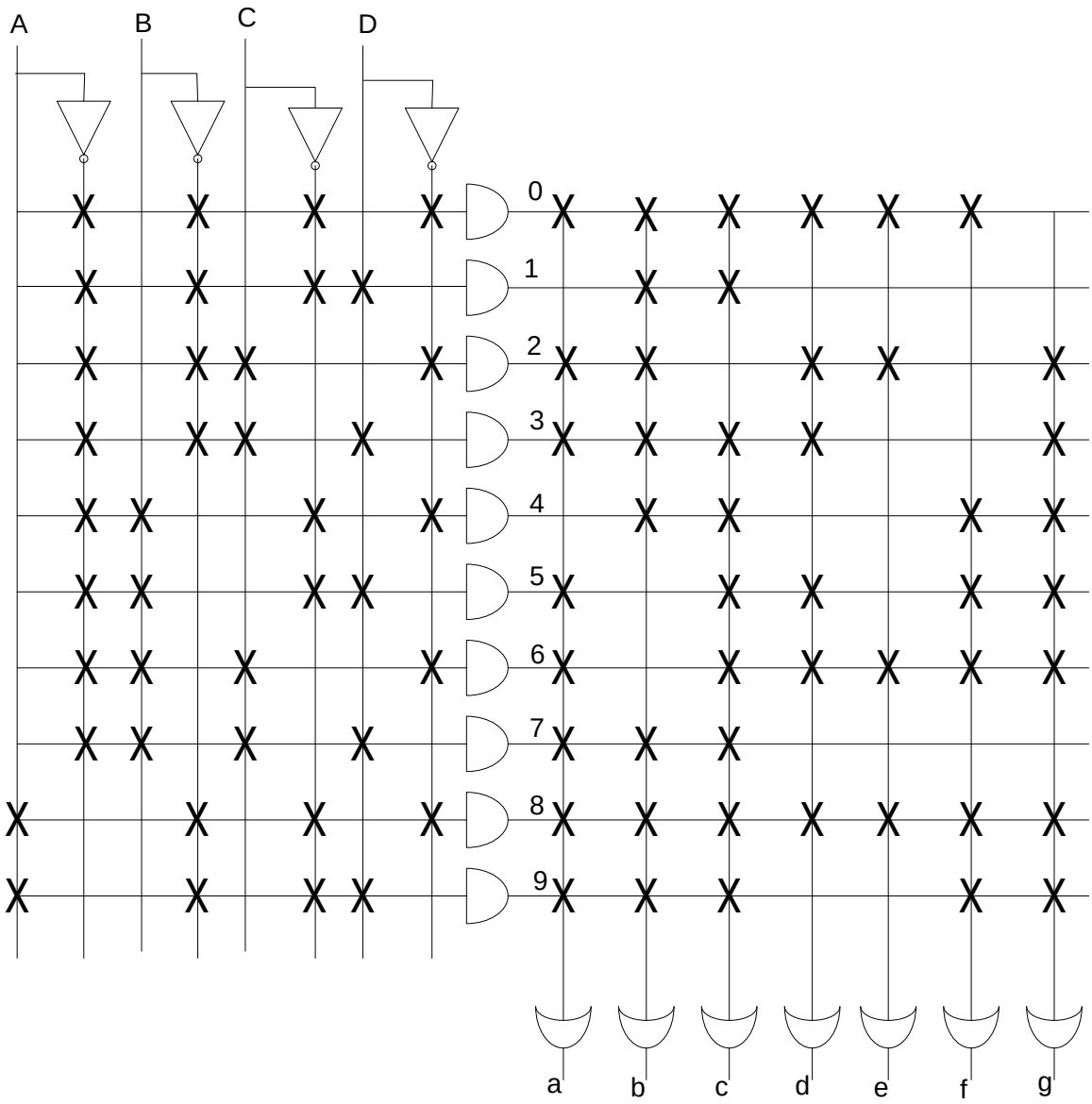
Q20. Design 7-segments decoder using PLA.

Answer: Seven segment indicator:



Following table shows the segments should light up to display a number.

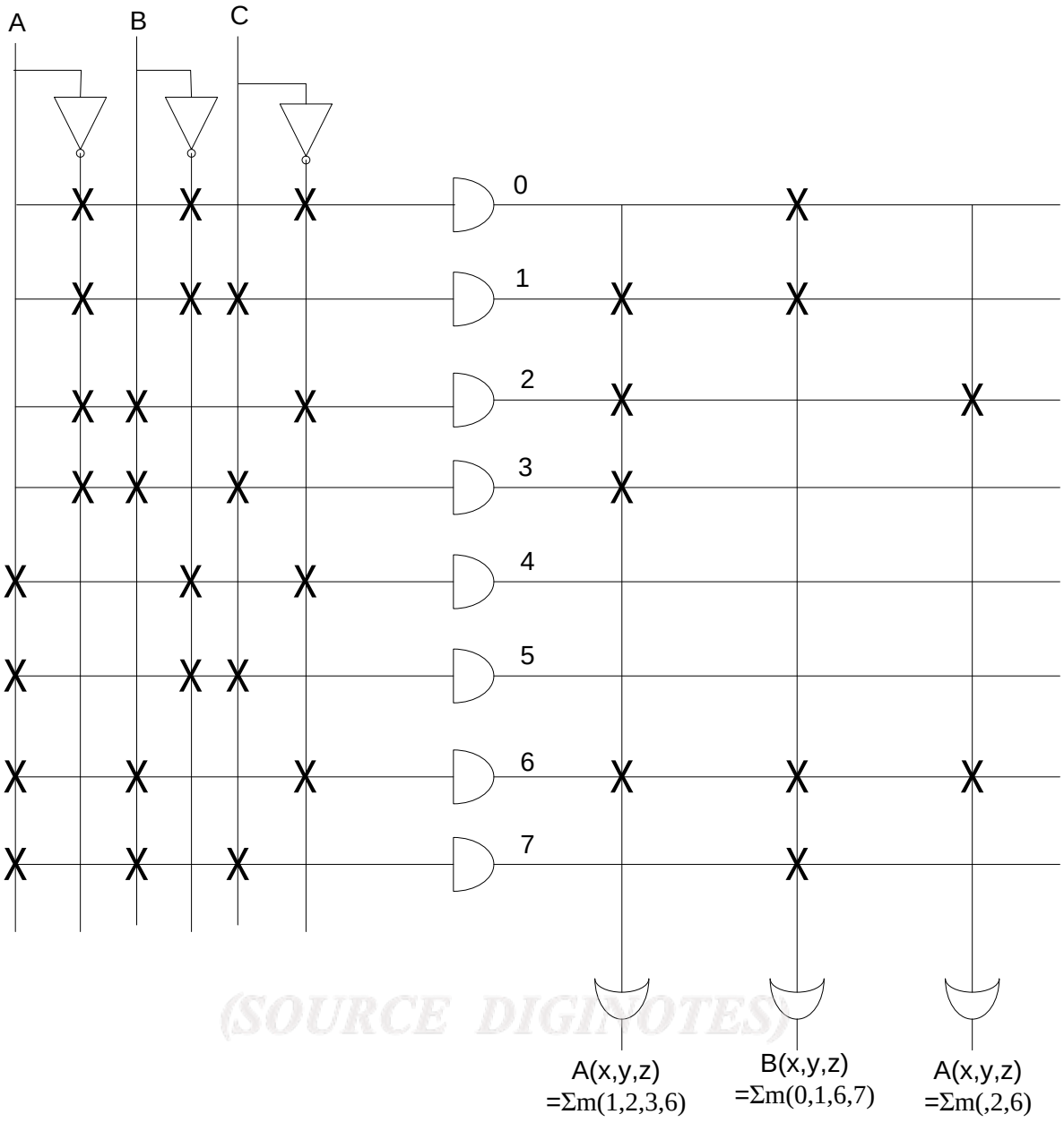
Number to display	Segments to light up
0	a,b,c,d,e,f
1	b,c
2	a,b,d,e,g
3	a,b,c.d,g
4	b,c,f,g
5	a,c,d,f
6	a,c,d,e,f,g
7	a,b,c
8	a,b,c,d,e,f,g
9	a,b,c,f,g



Q21. Implement the following function using PLA:

A(x,y,z)=Σm(1,2,3,6); B(x,y,z)=Σm(0,1,6,7); C(x,y,z)=Σm(2,6)

Answer:



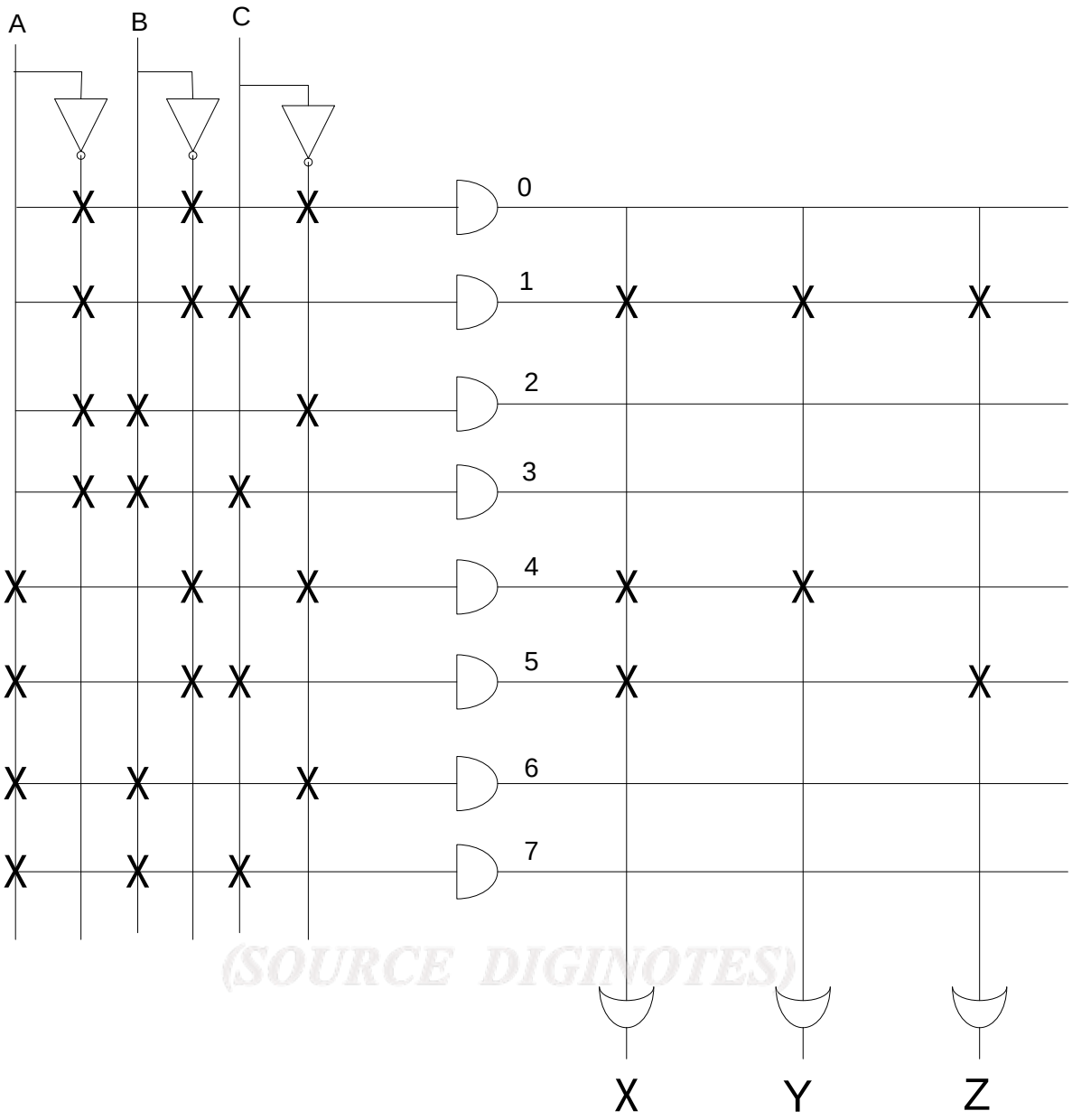
Q22. Draw the PLA circuit and realize the Boolean functions:

$X = A'B'C + AB'C' + B'C,\ Y = A'B'C + AB'C', Z = B'C$

$Answer: X = A'B'C + AB'C' + B'C = A'B'C + AB'C' + B'C(A + A') = A'B'C + AB'C' + AB'C$

$Y = A'B'C + AB'C'$

$Z = B'C = B'C(A + A') = AB'C + A'B'C$

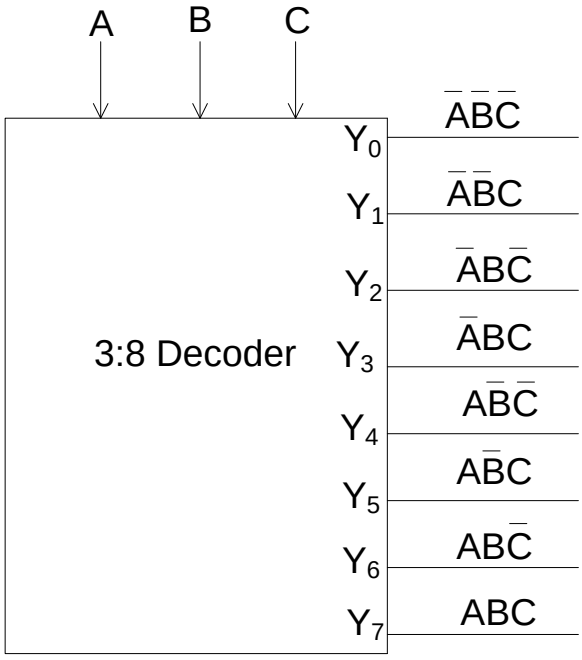


Q23. Describe the working principle of 3:8 decoder. Design a circuit that realizes the following functions using a 3:8 decoder and multi input OR gates.

$(i) F_1(A,B,C) = \sum m(1,3,7)$ (ii) $F_2(A,B,C) = \sum m(2,3,5)$

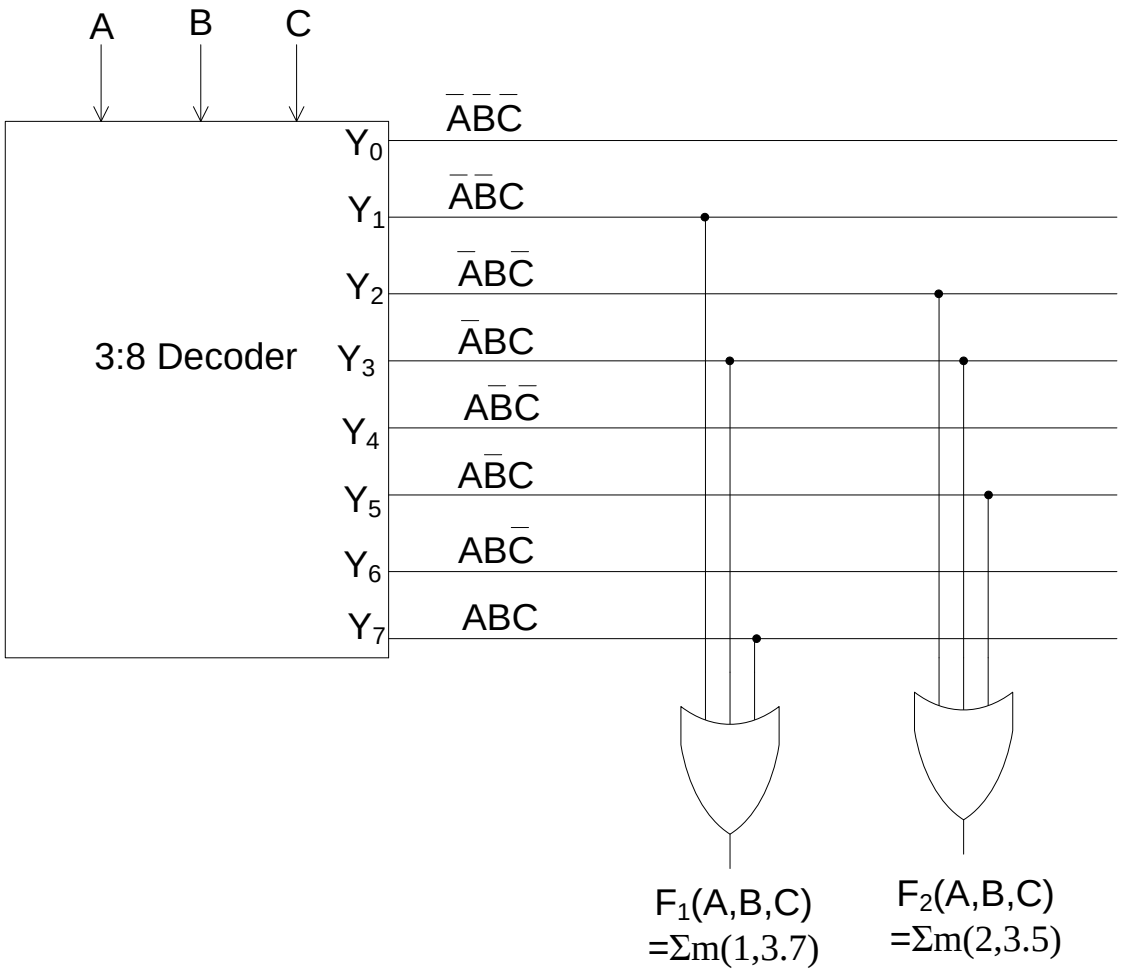
Answer: Working principle of 3:8 decoder:

There are 3 inputs and 8 outputs in a 3:8 decoder. One of the output is HIGH and remaining seven are LOW according to inputs. This is shown in truth table.



Truth table of 3:8 Decoder :

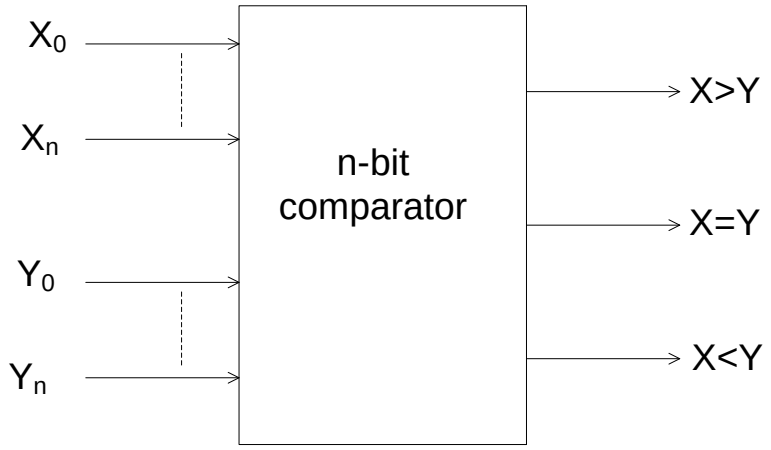
A	B	C	Y_0	Y_1	Y_2	Y_3	Y_4	Y_5	Y_6	Y_7
0	0	0	1	0	0	0	0	0	0	0
0	0	0	0	1	0	0	0	0	0	0
0	1	0	0	0	1	0	0	0	0	0
0	1	1	0	0	0	1	0	0	0	0
1	0	0	0	0	0	0	1	0	0	0
1	0	1	0	0	0	0	0	1	0	0
1	1	0	0	0	0	0	0	0	1	0
1	1	1	0	0	0	0	0	0	0	1



Q24. What is magnitude comparator? Design one bit comparator and write the truth table, logic circuit using basic gates.

Answer: A magnitude comparator compares two binary numbers and it produces an output showing the comparison of the two input numbers.

For example, two n-bit binary numbers $X=X_0X_1..X_n$ and $Y=Y_0Y_1\ldots Y_n$ are compared. There are three ouputs. The outputs are for X>Y, X=Y and X<Y as shown in the figure below.

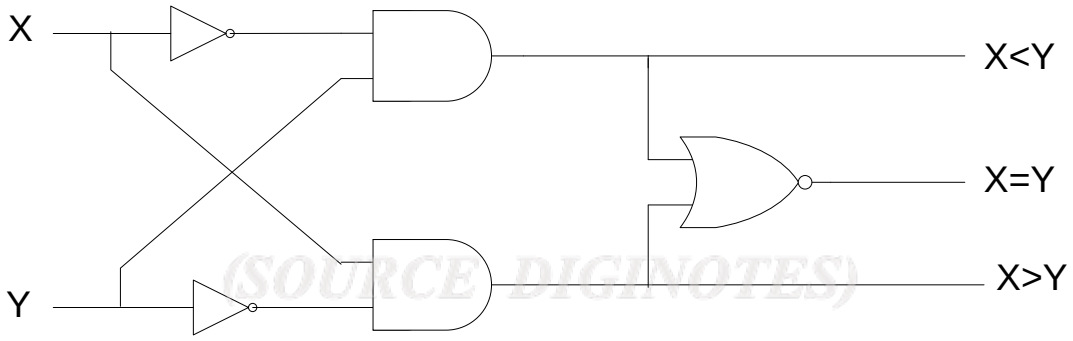


Designing of one bit comparator:

Truth table

Input		Output		
X	Y	X>Y	X=Y	X<Y
0	0	0	1	0
0	1	0	0	1
1	0	1	0	0
1	1	0	1	0

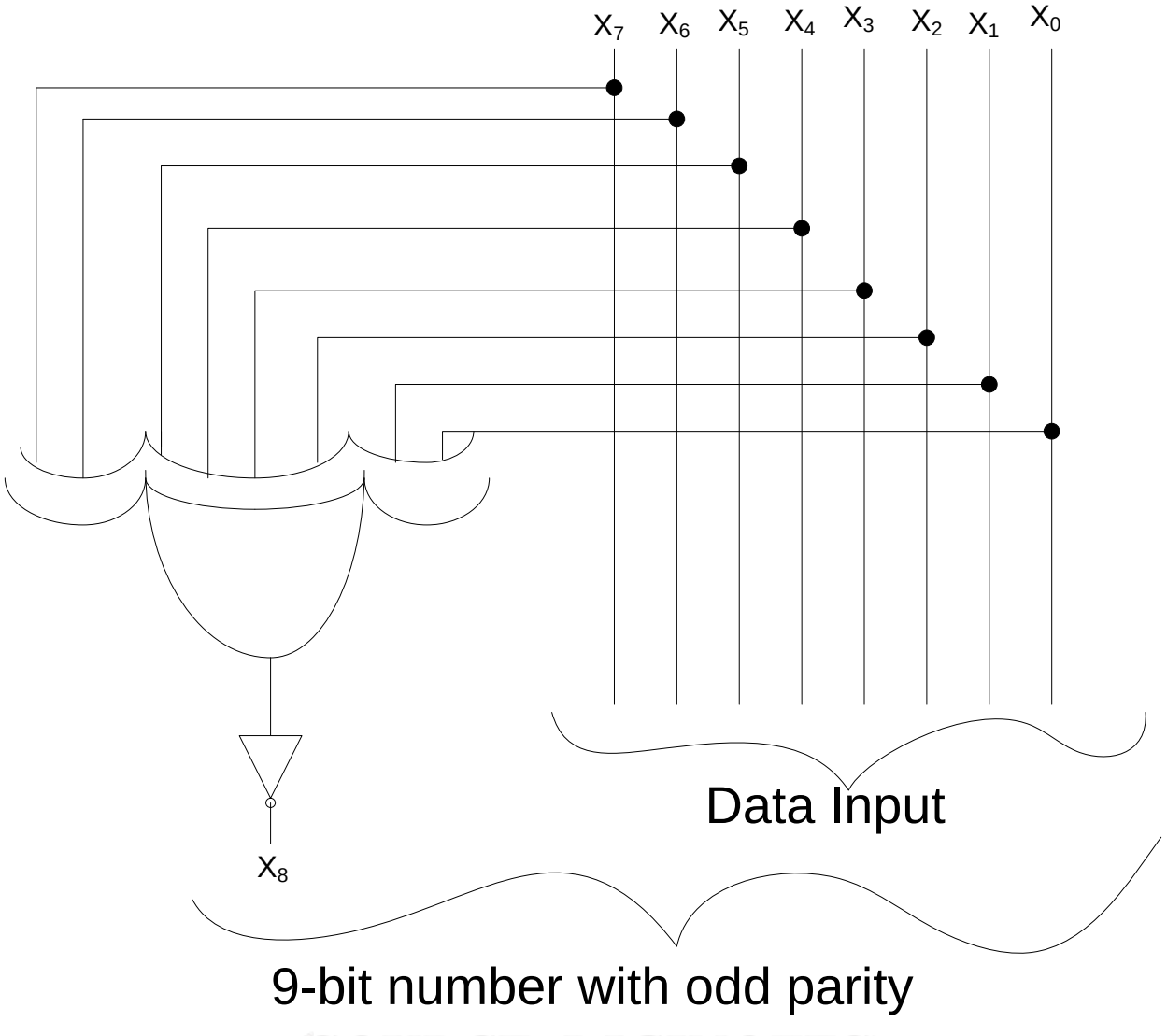
If G, L, E stand for greater than, less than and equal to respectively, then
(X>Y): G=XY′; (X<Y):L=X′Y; (X=Y): E=X′Y′+XY=(XY′+X′Y)′=(G+L)′



Q25. What is parity generator? Explain with an example.

Answer: A parity generator is a logic circuit which produces either even parity number or odd parity number as per requirement.

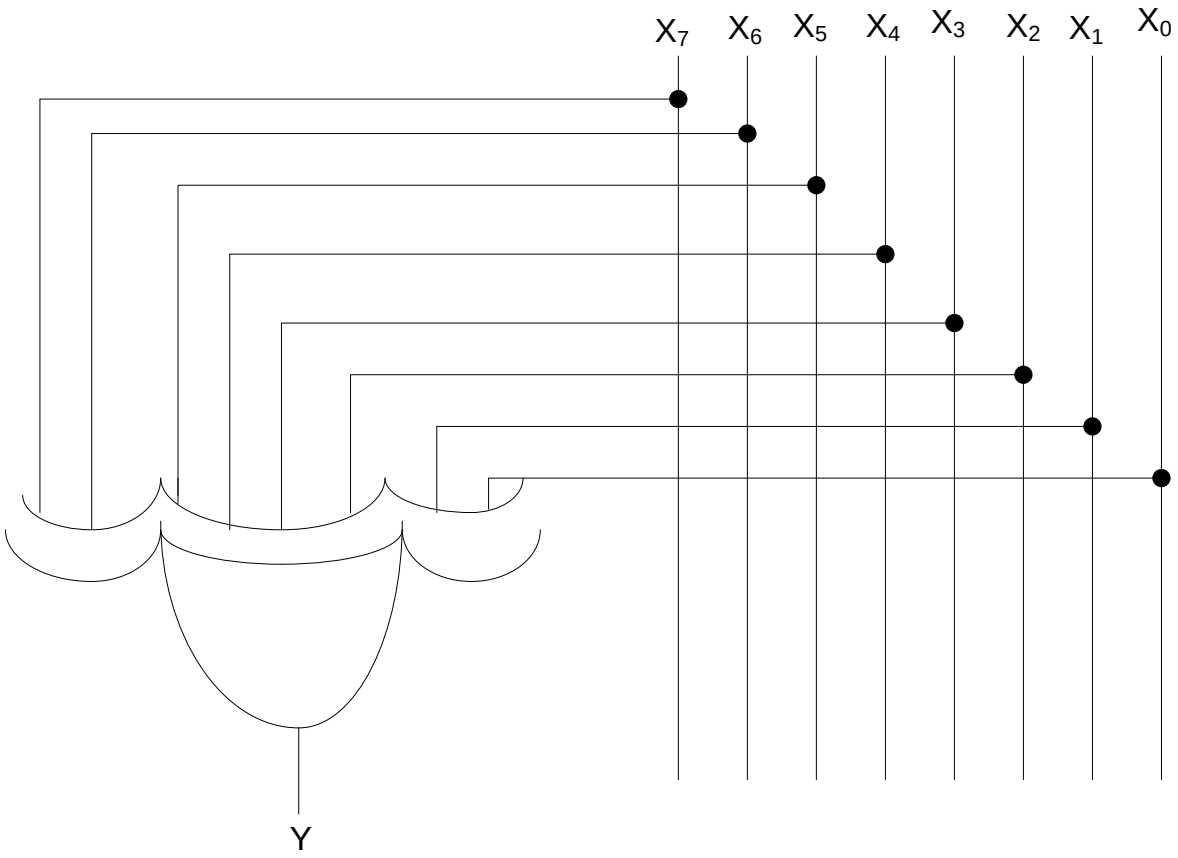
For example:



Q26. What is parity checker? Explain with example.

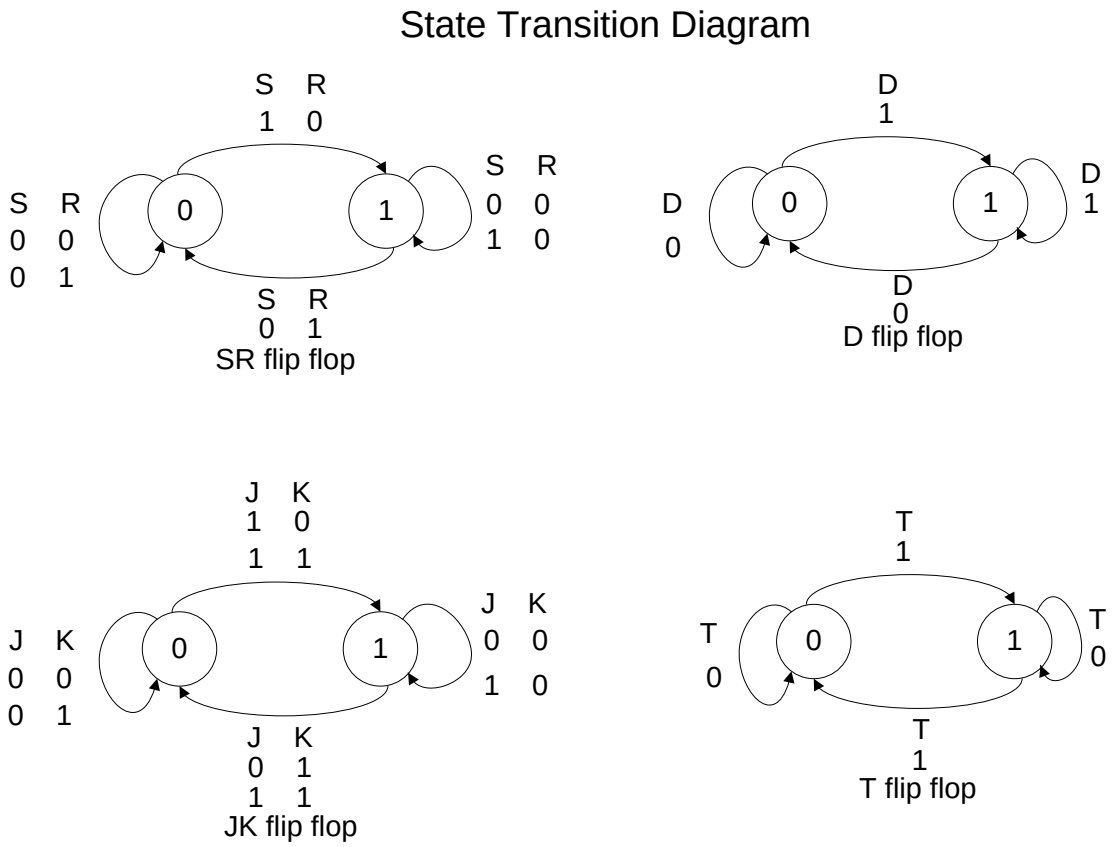
Answer: A parity checker check the parity of a number whether the number is of even parity or odd parity.

For example: The exclusive OR gate produces an output 1 when the input ($X_0 \ldots X_7$) is of odd parity and produces 0 when the input is of even parity.



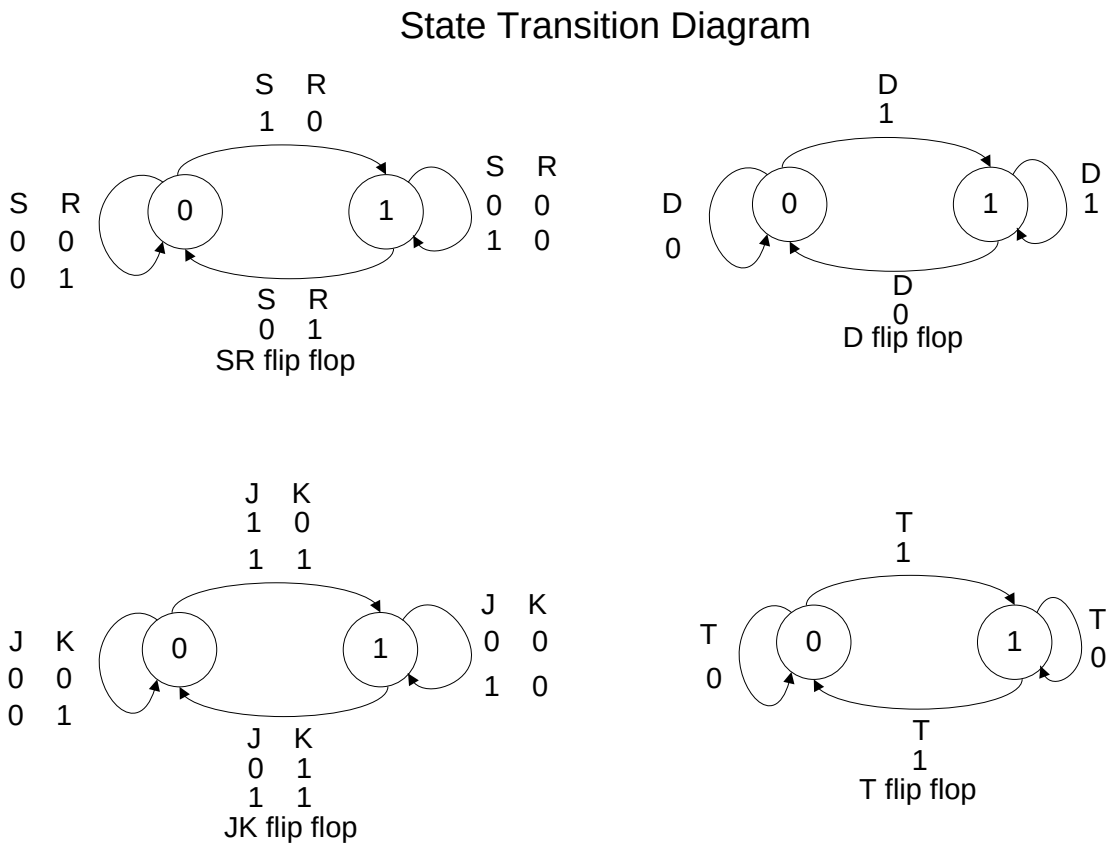
Q27. Give state transition diagram of SR, D, JK and T flip flops.

Answer:



Q28. Obtain the characteristic equation of SR, JK, D and T flip flops.

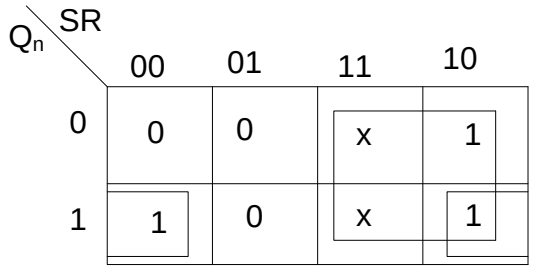
Answer:



Excitation Table for SR, JK, D and T flip flop is given below is prepared State Transition Diagram above

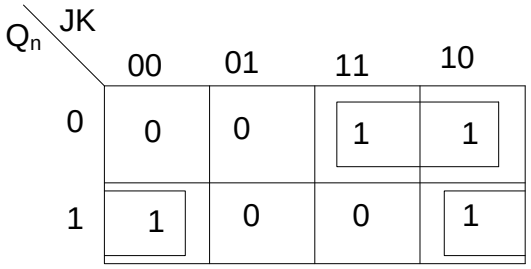
$Q_n \rightarrow Q_{n+1}$	S R	J K	D	T
0 0	0 X	0 X	0	0
0 1	1 0	1 X	1	1
1 0	0 1	X 1	0	1
1 1	X 0	X 0	1	0

From the Excitation Table, K-map is formed and then the characteristic equation is determined.



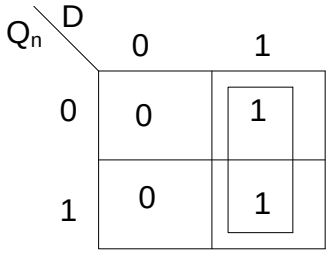
Characteristic Equation for SR flip flop is

$Q_{n+1}=S+\overline{R}Q_n$



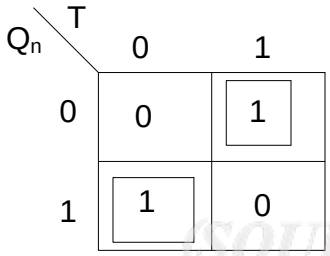
Characteristic Equation for JK flip flop is

$Q_{n+1}=J\overline{Q}_n+\overline{K}Q_n$



Characteristic Equation for D flip flop is

$Q_{n+1}=D$

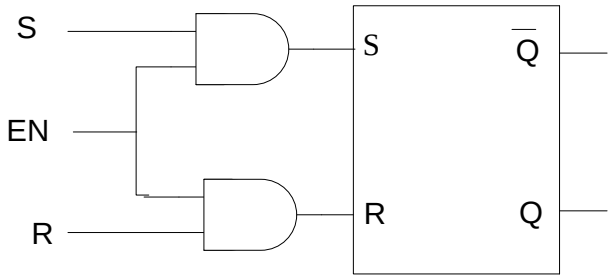


Characteristic Equation for T flip flop is

$Q_{n+1}=T\overline{Q}_n+\overline{T}Q_n$

Q29. Explain the operation of a gated SR latch with a logic diagram and truth table.

Logic diagram and truth table of gated SR flip flop is shown below:



Logic Diagram

EN	S	R	Q_{n+1}
1	0	0	Q_n (No Change)
1	0	1	0
1	1	0	1
1	1	1	Illegal
0	x	x	Q_n (No Change)

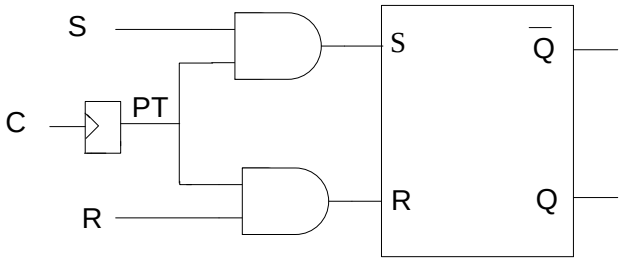
Truth table

When the Enable (EN) input is high, information at the R and S inputs will be transmitted directly to the outputs. The latch is said to be enabled. When the Enable (EN) input is low, the outputs of the AND gates are low and information at the R and S inputs will not be transmitted to the outputs. The latch is said to be disabled.

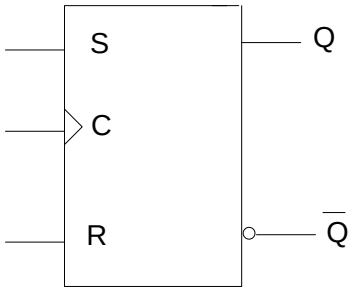
It is possible to strobe or clock the flip flop in order to store information at any time and then hold the stored information for any desired period of time. This flip flop is called a gated or clocked RS flip flop.

Q30. Explain the operation of edge triggered 'SR' flip flop with the help of a logic diagram and truth table. Also draw the relevant waveforms.

Answer: Positive edge triggered 'SR' flip flop



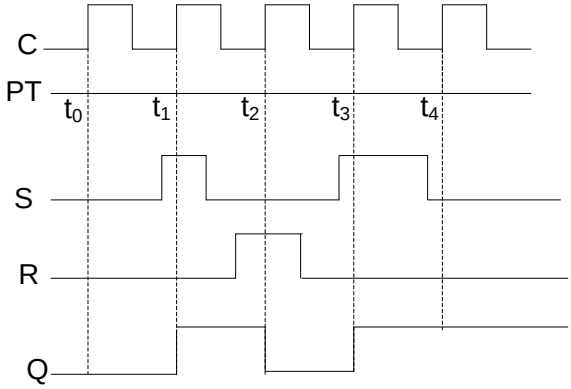
Logic Diagram



IEEE Symbol

C	S	R	Q_{n+1}
↑	0	0	Q_n (No Change)
↑	0	1	0 (Reset)
↑	1	0	1 (Set)
↑	1	1	Illegal

Truth table



Waveform of positive edge triggered RS flip flop

Positive edges occur at t_0,t_1,t_2,t_3 and t_4.

At t_0, S=0 and R=0, hence no change in the output and Q=0.

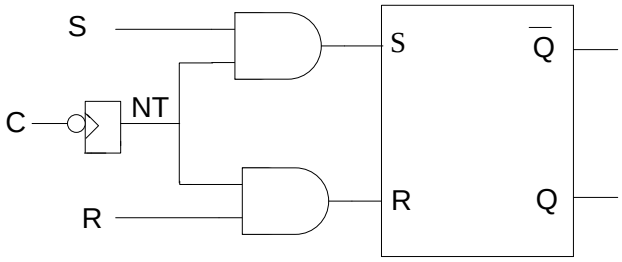
At t_1, S=1 and R=0, hence the output is set and Q=1.

At t_2, S=0 and R=1, hence the output is reset and Q=0.

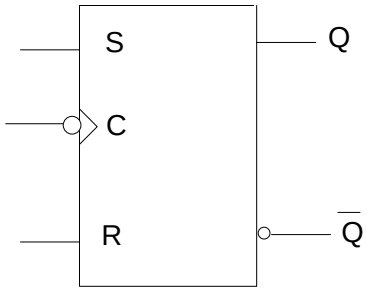
At t_3, S=1 and R=0, hence the output is set and Q=1

At t_4, S=0 and R=0, hence no change in the output and Q=1.

Negative edge triggered 'SR' flip flop



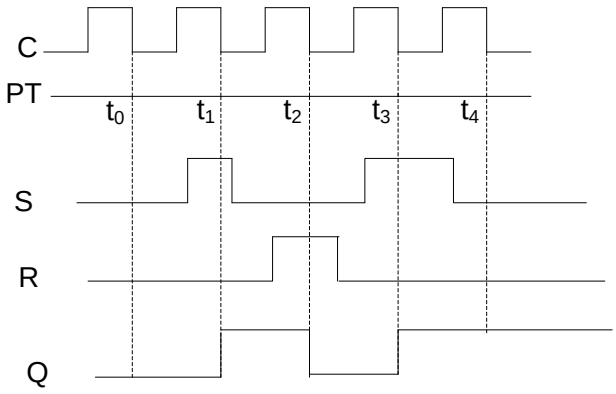
Logic Diagram



SR flip flop Symbol

C	S	R	Q_{n+1}
↓	0	0	Q_n (No Change)
↓	0	1	0 (Reset)
↓	1	0	1 (Set)
↓	1	1	Illegal

Truth table



Waveform of negative edge triggered SR flip flop

.

Negative edges occur at t_0,t_1,t_2,t_3 and t_4.

At t_0, S=0 and R=0, hence no change in the output and Q=0.

At t_1, S=1 and R=0, hence the output is set and Q=1.

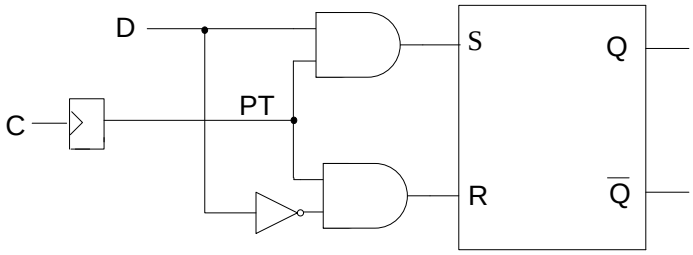
At t_2, S=0 and R=1, hence the output is reset and Q=0.

At t_3, S=1 and R=0, hence the output is set and Q=1

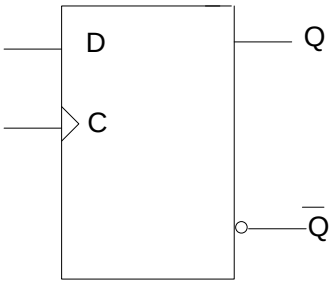
At t_4, S=0 and R=0, hence no change in the output and Q=1.

Q31. Explain the operation of edge triggered 'D' flip flop with the help of a logic diagram and truth table. Also draw the relevant waveforms.

Answer: Positive edge triggered 'D' flip flop



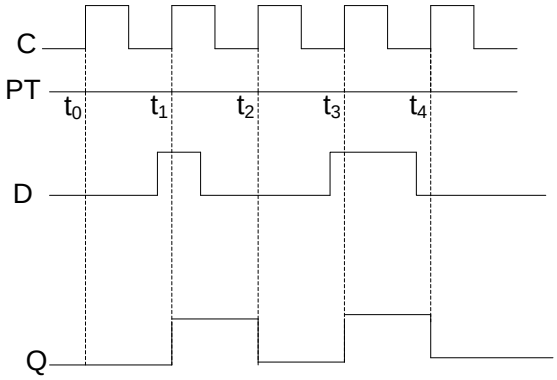
Logic Diagram



D flip flop Symbol

C	D	Q_{n+1}
0	x	Q_n (No Change)
↑	0	0
↑	1	1

Truth table



Waveform of positive edge triggered D flip flop

Positive edges occur at t_0,t_1,t_2,t_3 and t_4.

At t_0, D=0, hence the output is low and Q=0.

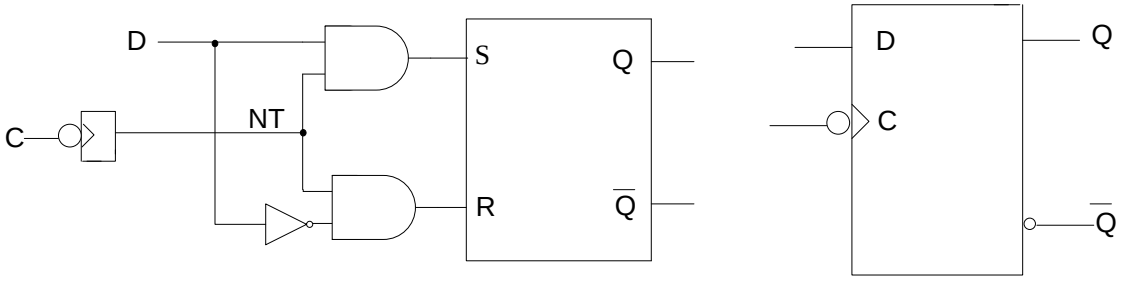
At t_1, D=1, hence the output is high and Q=1.

At t_2, D =0, hence the output is low and Q=0.

At t_3, D=1, hence the output is high and Q=1

At t_4, D=0, hence no change in the output and Q=1.

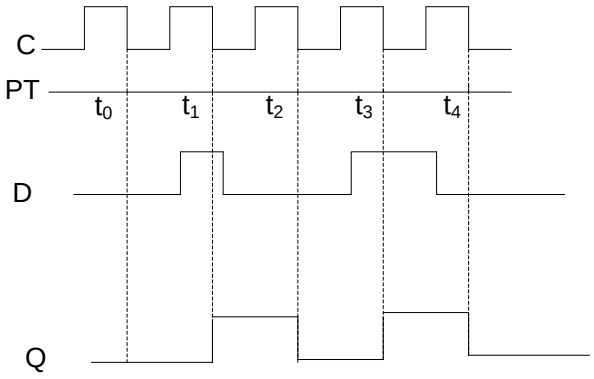
Negative triggered D flip flop



Logic Diagram D flip flop Symbol

C	D	Q_{n+1}
0	x	Q_n (No Change)
↓	0	0
↓	1	1

Truth table

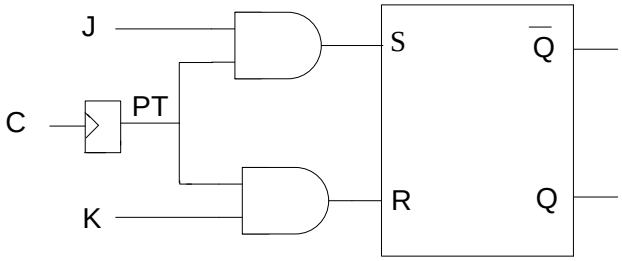


Waveform of positive edge triggered D flip flop

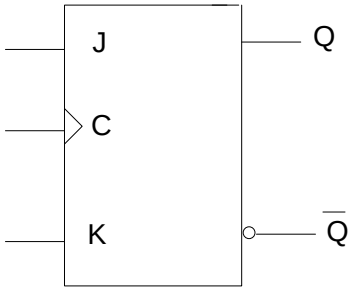
Negative edges occur at t_0,t_1,t_2,t_3 and t_4.
At t_0, D=0, hence the output is low and Q=0.
At t_1, D=1, hence the output is high and Q=1.
At t_2, D =0, hence the output is low and Q=0.
At t_3, D=1, hence the output is high and Q=1
At t_4, D=0, hence no change in the output and Q=1.

Q32. Explain the working of pulse triggered JK flip flop with typical JK flip flop waveform.

Answer: Positive Edge Triggered JK flip flop



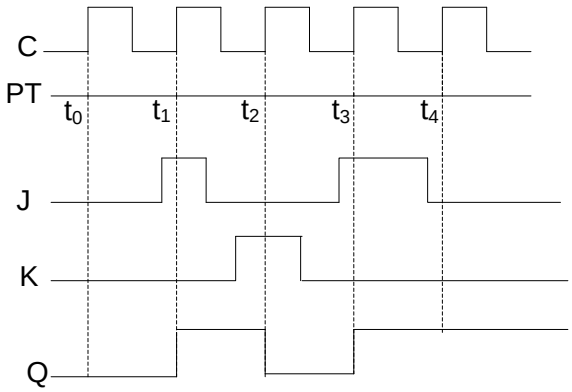
Logic Diagram



JK flip flop Symbol

C	J	K	Q_{n+1}
↑	0	0	Q_n (No Change)
↑	0	1	0
↑	1	0	1
↑	1	1	Toggle

Truth table



Waveform of positive edge triggered JK flip flop

Positive edges occur at t_0,t_1,t_2,t_3 and t_4.

At t_0, J=0 and K=0, hence no change in the output and Q=0.

At t_1, J=1 and K=0, hence the output is high and Q=1.

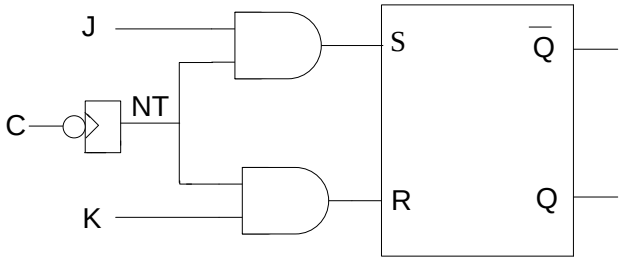
At t_2, J=0 and K=1, hence the output is low and Q=0.

At t_3, J=1 and K=0, hence the output is high and Q=1

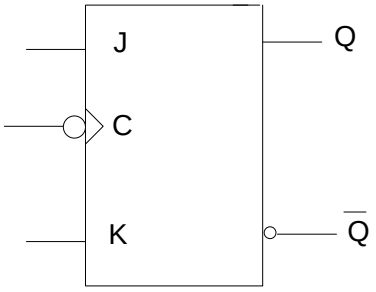
At t_4, J=0 and K=0, hence no change in the output and Q=1.

:

Negative Edge Triggered JK flip flop



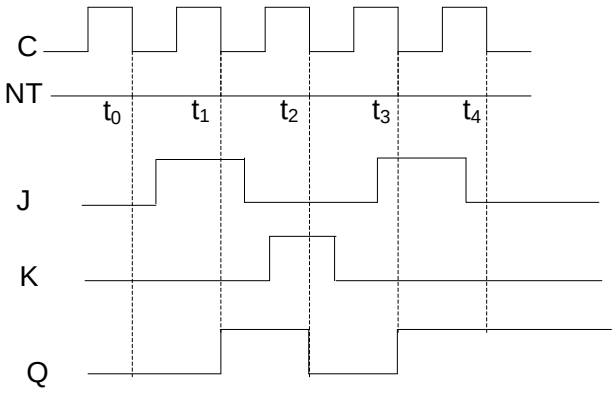
Logic Diagram



JK flip flop Symbol

C	J	K	Q_{n+1}
↓	0	0	Q_n (No Change)
↓	0	1	0
↓	1	0	1
↓	1	1	Toggle

Truth table

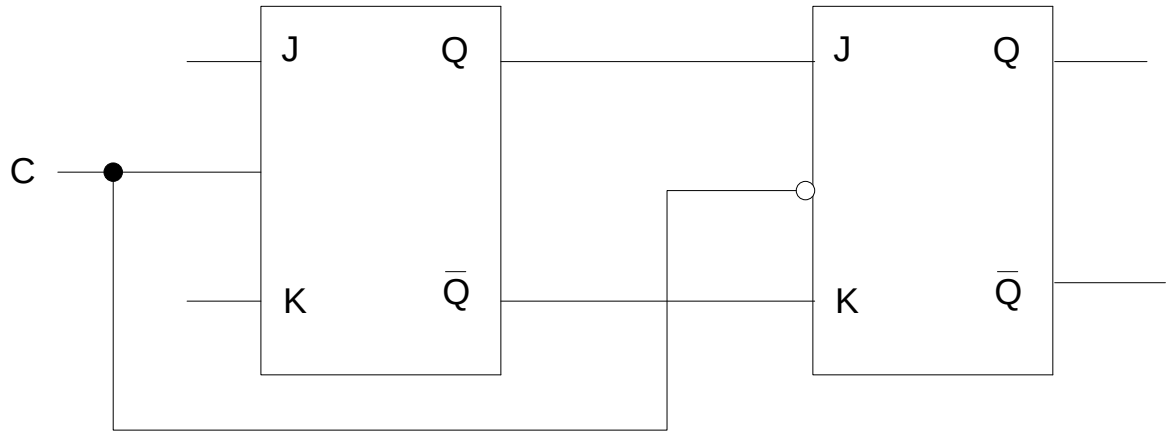


Waveform of positive edge triggered JK flip flop

Negative edges occur at t_0,t_1,t_2,t_3 and t_4.
At t_0, J=0 and K=0, hence no change in the output and Q=0.
At t_1, J=1 and K=0, hence the output is high and Q=1.
At t_2, J=0 and K=1, hence the output is low and Q=0.
At t_3, J=1 and K=0, hence the output is high and Q=1
At t_4, J=0 and K=0, hence no change in the output and Q=1.

Q33. Explain the working of Master Slave J K flip flops with logic diagram.

Answer:



Master Slave flip flop

Master is positive-level-triggered and the slave is negative-level-triggered. The master responds to its J and K inputs before the slave.

If J=1 and K=0, the master sets on the positive clock transition. The high Q output of the master drives the J input of the slave. So, on the negative clock transition, the slave sets, thus copying the action of the master.

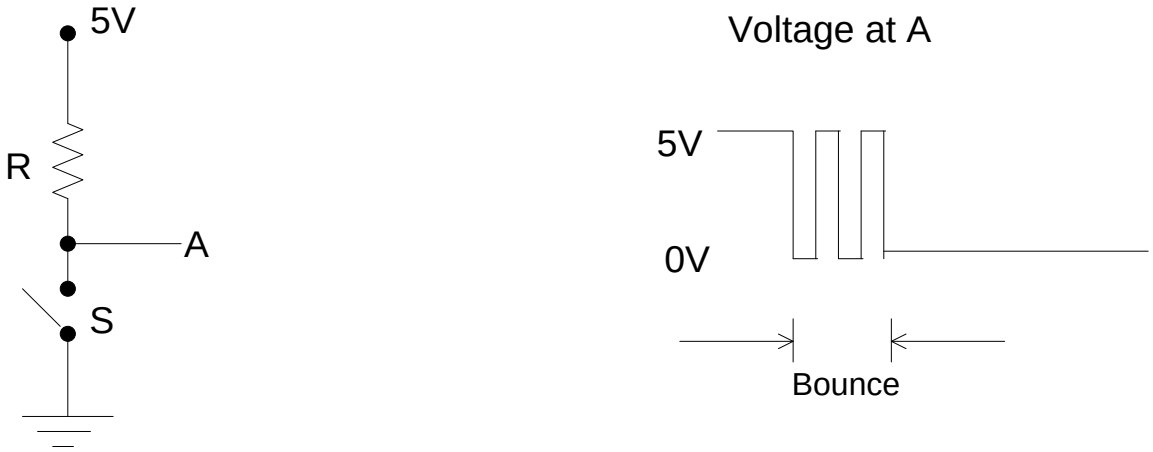
If J=0 and K=1, the master resets on the positive clock transition. The high $\overline{Q}$ output of the master drives the K input of the slave. So, on the negative clock transition, the slave resets, thus copying the action of the master.

If J=1 and K=1, the master toggle on the positive clock transition. The slave also toggle at the negative clock transition thus copying the action of the master.

If J=0 and K=0, the master and the slave both are disabled, thus copying the action of the master.

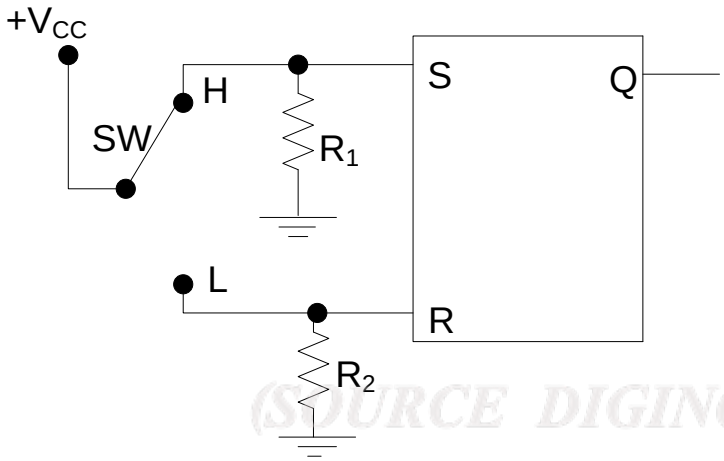
Q34. What is contact bounce? With neat diagram, explain the working principles of Switch De bounce circuit.

Answer:



Any mechanical switching device consists of a moving contact arm restrained by some spring system. As a result, when a mechanical switch is closed, the arm is moved from one stable position to other and the arm bounces much as a hard ball bounces when dropped on a hard surface. This phenomenon is known as contact bounce. When switch S is closed, due contact bounce the voltage at the A is shown in the above figure.

RS Latch Debounce Circuit



When switch(SW) is moved to the position H, R=0 and S=1. Bouncing occurs at S due to contact bounce of the switch. The flip flop treat as high and low inputs. The flip flop will be set with Q=1 at the firs high of the contact bounce. When the switch continue to bounce, losing contact, the input signals are R=S=0, thus the flip flop remains at Q=1. As a result, the flip flop responds only to the first high of the contact bounce.