Free Study Material from

All Lab Experiments

Electronics

for NET/Gate Physical Sciences

Multiplexer

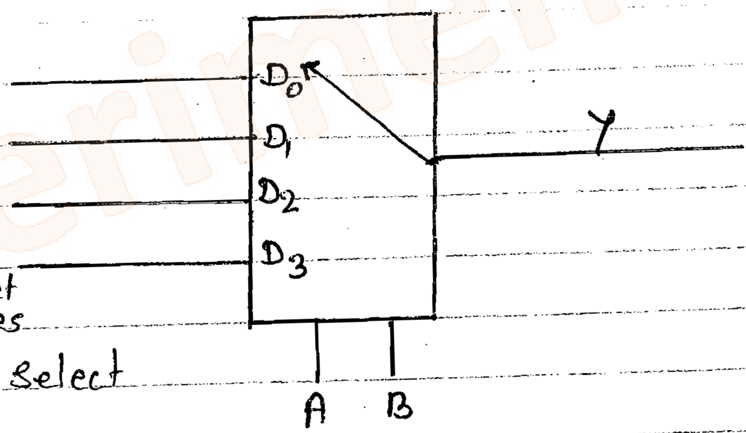
Multiplexer

A divice which select single output line from multiple number of input lines and display it at output.

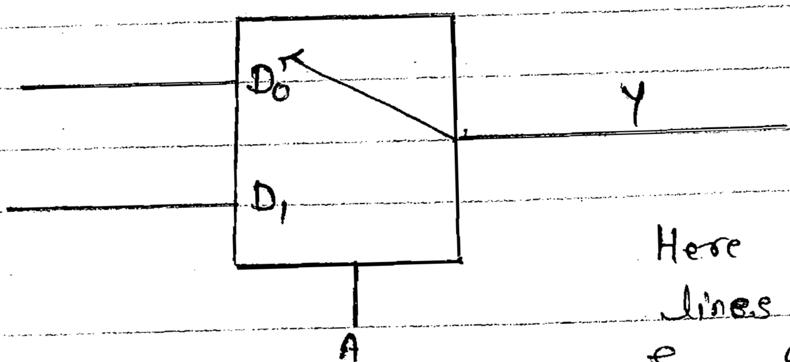
- The internal structure of Mux is a rotatory switch (Commutator).
- It is used for parallel to serial conversion.
- It is also called as universal combinational circuit.

Diagram :-

Since Here 4 input lines
So,
$4 = 2^{2}$ ← Select lines
So, there is 2 select lines (A, B)



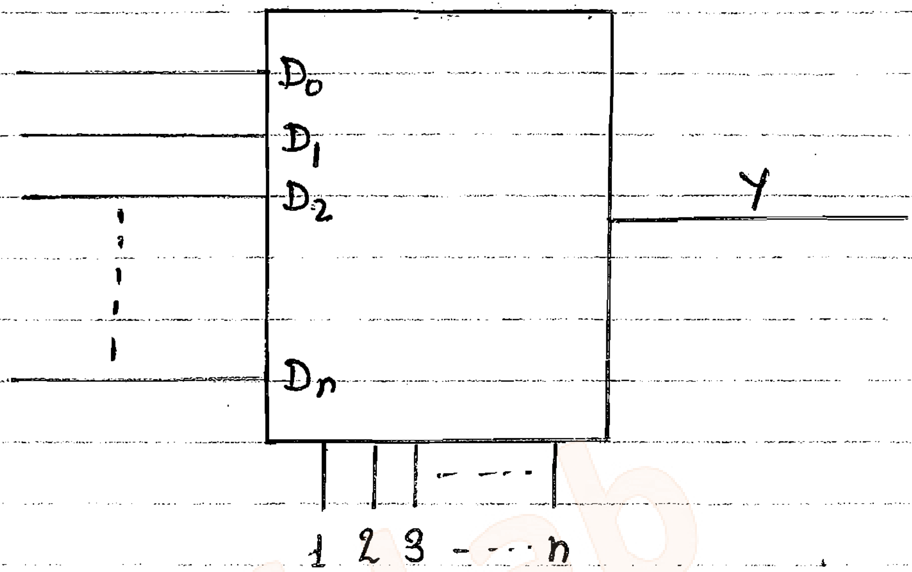
[4x1 Mux]



Here two Inpu
lines
So, 2^{1} ← select l

[2x1 Mux] So Here 1 select line.

Similarly for N inputs :-



Here numbers of input lines = N

So

$$N = 2^{n} \leftarrow \text{No. of select line}$$

No. of inputs.

A device having 2^n input lines with 'n' number of select line to select single input and display at the output:-

$$N = 2^n$$
$$\log N = \log 2^n$$
$$\log N = n \log 2$$

Since here binary numbers are used so base of logarithm is 2.

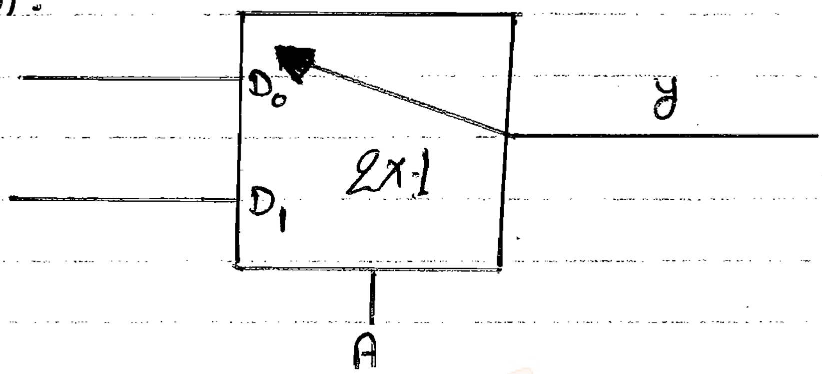
$$\boxed{\log_2 N = n \log_2 2}$$

or

No. of input lines $\boxed{\log_2 N = n}$ No. of select lines Since $\log_n n = 1$

* 2X1 Multiplexer :-

Diagram :-



SOP (Actual) :-

	A	B	Y
$\bar{A}\bar{B}$ ←	0	0	1
$\bar{A}B$ ←	0	1	0
$A\bar{B}$ ←	1	0	1
AB ←	1	1	1

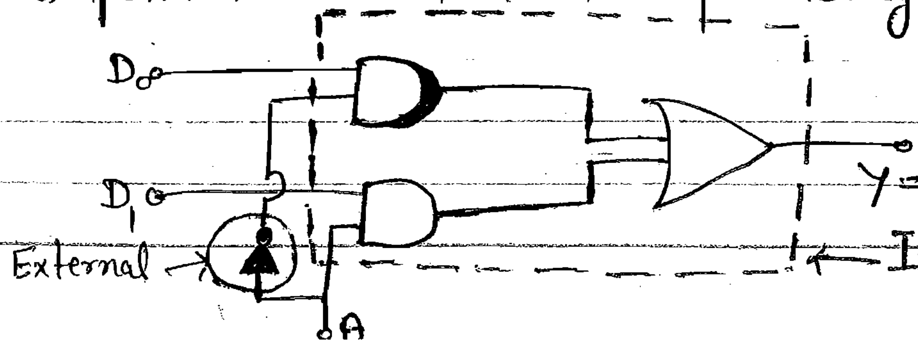
Due to this reson we can not take 0 o/p.

$y = \bar{A}\bar{B}\cdot 1 + \bar{A}B\cdot 0 + A\bar{B}\cdot 1 + AB\cdot 1 = \bar{A}\bar{B} + A\bar{B} + AB$

	A	Y
$\bar{A}D_0$ ←	0	D_0
AD_1 ←	1	D_1

$$\boxed{Y = \bar{A}D_0 + AD_1}$$

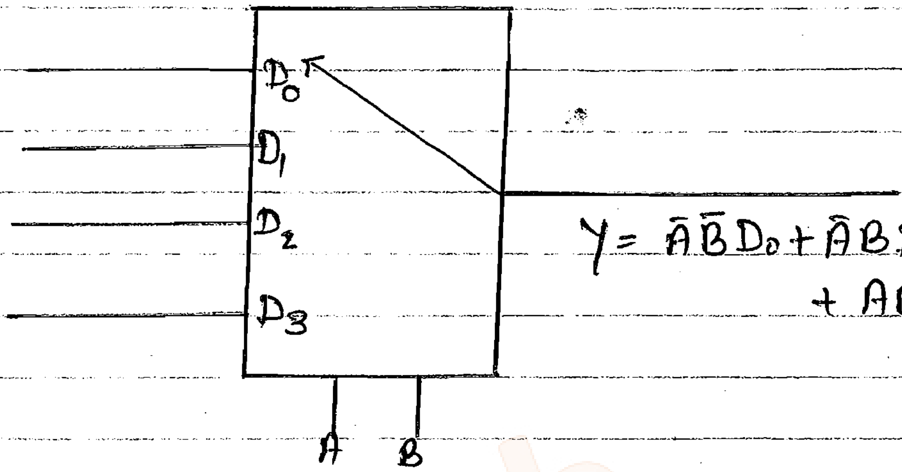
Qus Impliment 2x1 Mux by using logic gates :-



$Y = \bar{A}D_0 + AD_1$

← Internal Structure of Mux.

* 4x1 Multiplexer :-



$$Y = \bar{A}\bar{B}D_0 + \bar{A}BD_1 + A\bar{B}D_2 + ABD_3$$

		A	B
0	←	0	0
1	←	0	1
2	←	1	0
3	←	1	1

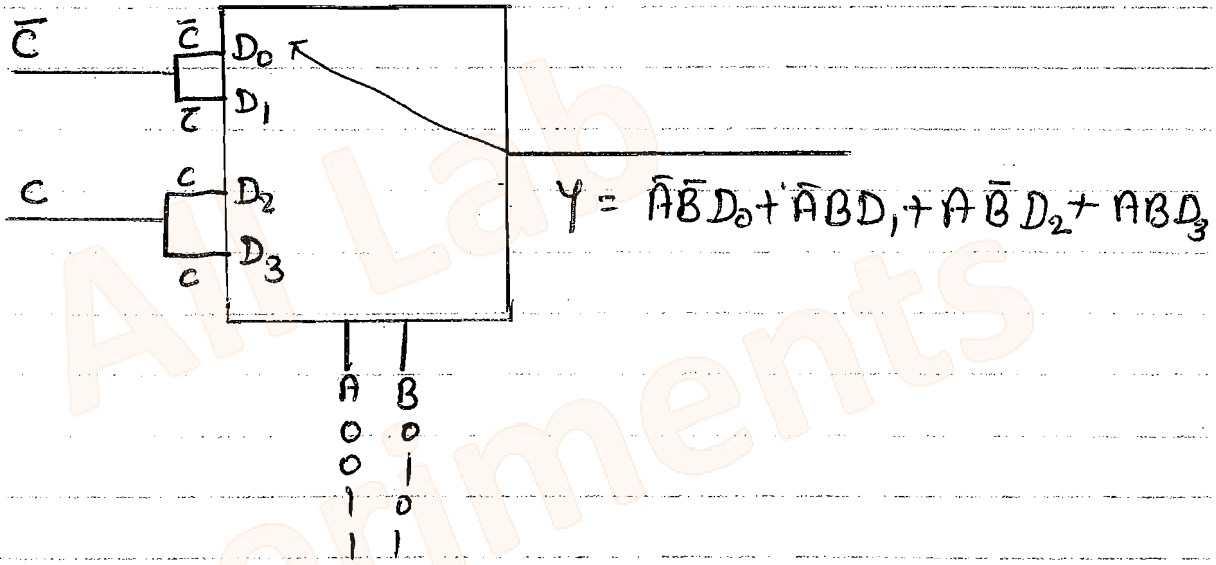
		A	B	Y
$\bar{A}\bar{B}$	←	0	0	D_0
$\bar{A}B$	←	0	1	D_1
$A\bar{B}$	←	1	0	D_2
AB	←	1	1	D_3

So $$\boxed{Y = \bar{A}\bar{B}D_0 + \bar{A}BD_1 + A\bar{B}D_2 + ABD_3}$$

* Minimisation of logic expression by using Mux :-

• Type I :-

Ques Find the output minimise expression for the given Mux circuit.



Sol^n

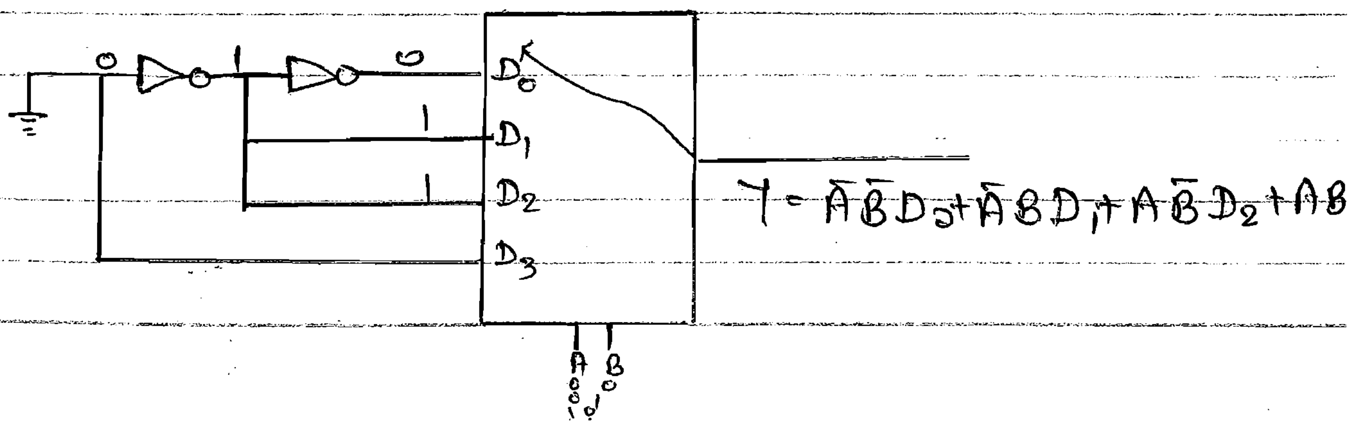
$y = \bar{A}\bar{B}\bar{C} + \bar{A}B\bar{C} + A\bar{B}C + ABC$

$y = \bar{A}\bar{C}(\bar{B}+B) + AC(\bar{B}+B)$

$y = \bar{A}\bar{C} + AC$

$\boxed{y = A \odot C}$

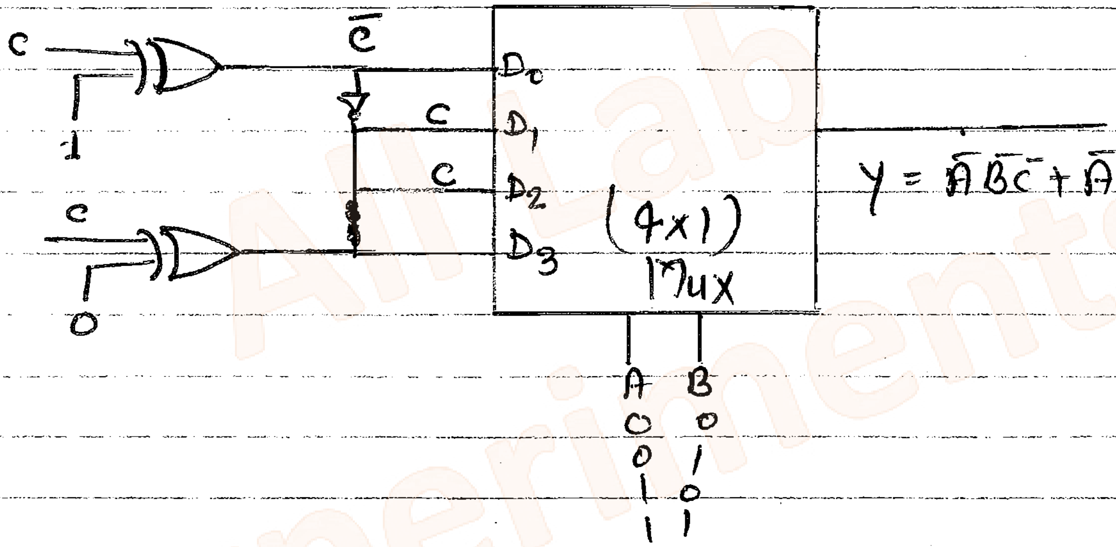
Ques For the given circuit diagram find the minimise expression.



Sol^n

$$y = \bar{A}\bar{B}D_0 + \bar{A}BD_1 + A\bar{B}D_2 + ABD_3$$
$$= \bar{A}\bar{B}\,0 + \bar{A}B.1 + A\bar{B}.1 + AB.0$$
$$= \bar{A}B + A\bar{B}$$
$$\boxed{y = A \oplus B}$$

Q. For the given circuit diagram find minimise expression.



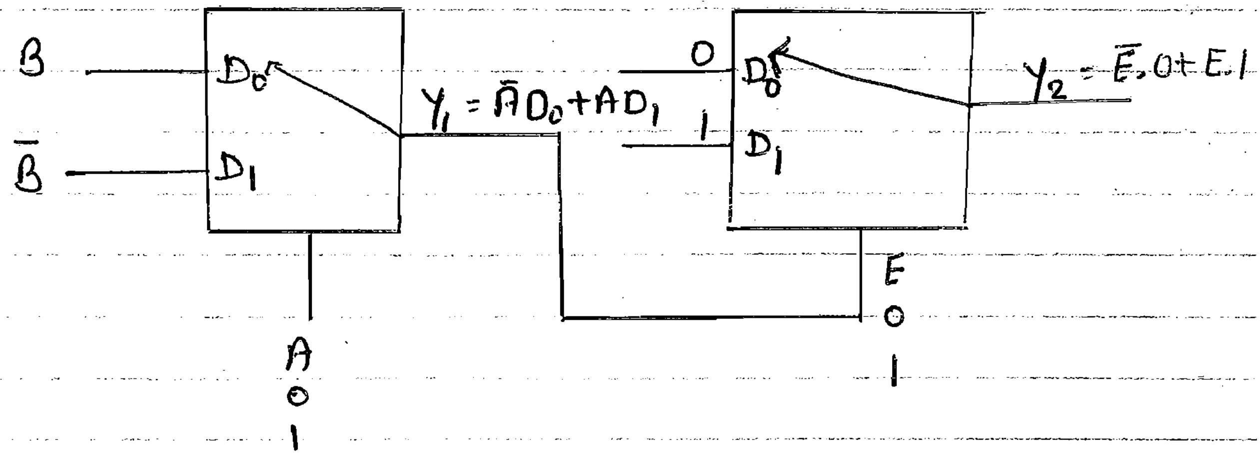
$y = \bar{A}\bar{B}\bar{C} + \bar{A}BC + A\bar{B}C + ABC$

Sol^n

$$y = \bar{A}\bar{B}\bar{C} + \bar{A}BC + A\bar{B}C + ABC$$
$$y = \bar{A}\bar{B}\bar{C} + \bar{A}BC + AC(\bar{B}+B)$$
$$y = \bar{A}\bar{B}\bar{C} + \bar{A}BC + AC$$
$$y = \bar{A}\bar{B}\bar{C} + (\bar{A}B + A)C$$
$$y = \bar{A}\bar{B}\bar{C} + C(A+B)(A+\bar{A})$$
$$\boxed{y = \bar{A}\bar{B}\bar{C} + AC + BC}$$

Ques Find the output expression y.

Sol^n



$$y_1 = \bar{A}D_0 + AD_1$$

$$y_1 = \bar{A}B + A\bar{B}$$

$$y_1 = A \oplus B = E \text{ (Say)}$$

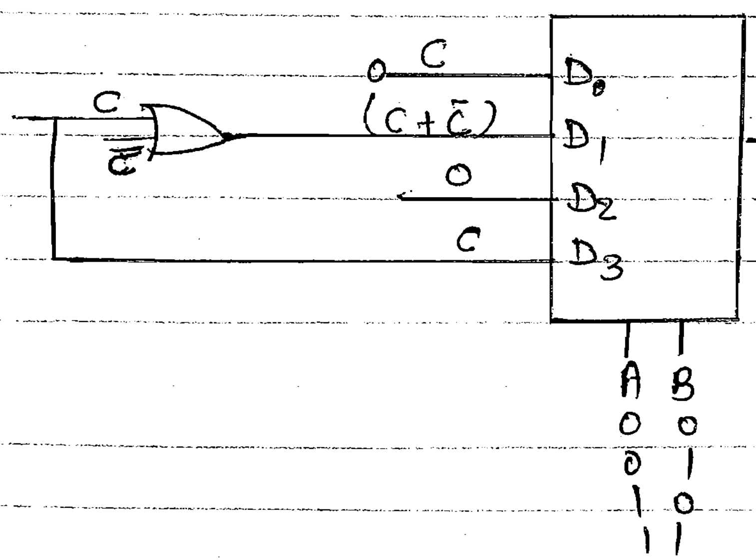
So $\quad y_2 = \bar{E}.0 + E.1$

$$y_2 = \overline{A \oplus B}.0 + A \oplus B.1$$

$$\boxed{y_2 = A \oplus B}$$

Ans

Que Find the output minimise expression.



$$Y = \bar{A}\bar{B}C + \bar{A}B.1 + A\bar{B}.0 + ABC$$

Sol^n

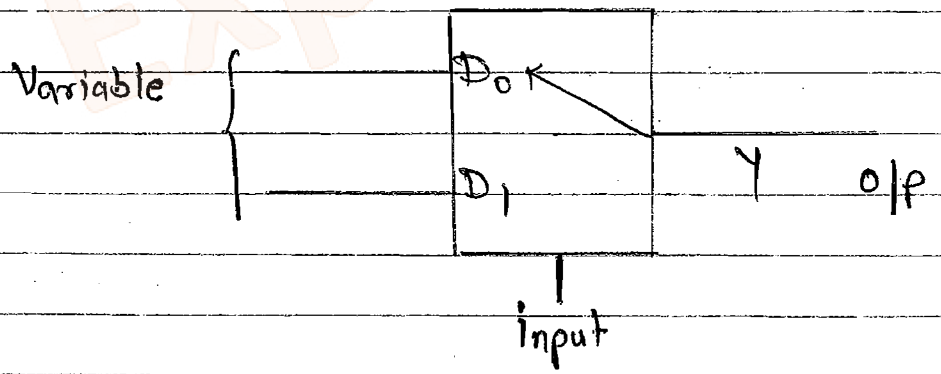
$$Y = \bar{A}\bar{B}C + \bar{A}B(C+\bar{C}) + A\bar{B}\cdot 0 + ABC$$
$$= \bar{A}\bar{B}C + \bar{A}B\cdot 1 + ABC$$
$$= \bar{A}\,(\underbrace{B + \bar{B}C}_{1\,+\,2\cdot 3}) + ABC$$
$$= \bar{A}\,(B+C) + ABC$$
$$= \bar{A}B + \bar{A}C + ABC$$
$$= \bar{A}B + C\,[\underbrace{\bar{A} + AB}_{1\,+\,2\cdot 3}]$$
$$= \bar{A}B + C\,[\bar{A}+B]$$
$$\boxed{Y = \bar{A}B + \bar{A}C + BC}$$

Ans

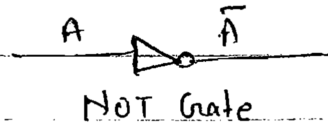
- **Type II :-**

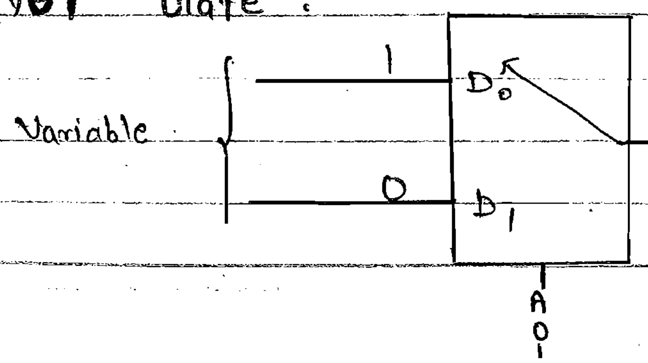
* Implementation of logic circuits by using Mux.

2x1 Mux :-



1. NOT Gate :-

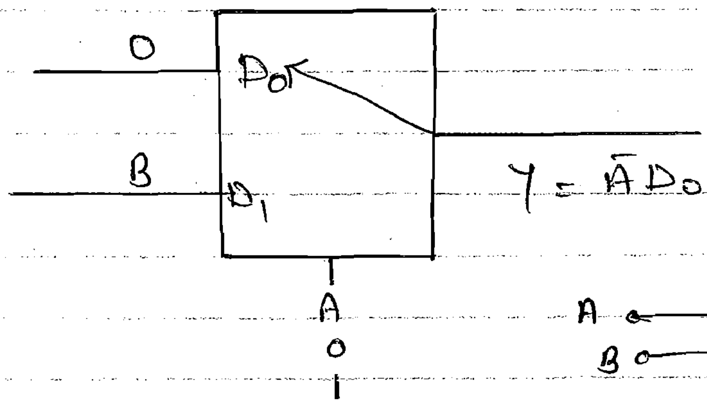




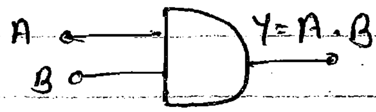
$$Y = \bar{A}D_0 + AD_1$$

2. AND Gate :-

Variable

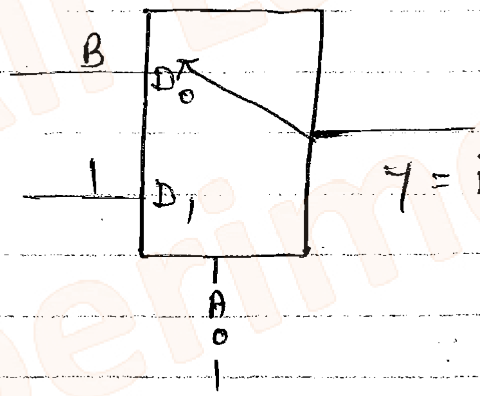


$Y = \bar{A}D_0 + AD_1$

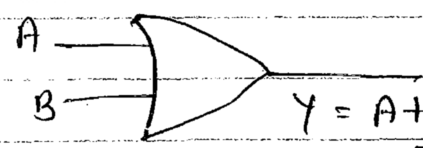


AND Gate.

3. OR - Gate :-



$Y = \bar{A}D_0 + AD_1$



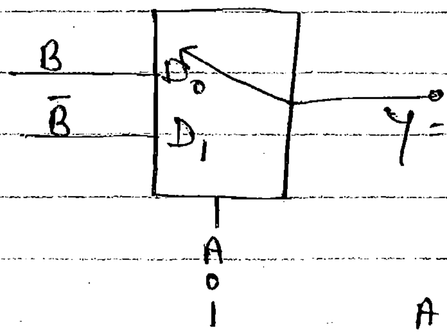
$Y = A+B$

$= A+\bar{A}B$

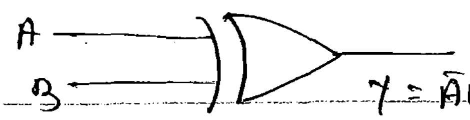
$= (A+\bar{A})(A+B)$

$= (A+B)$

4. Ex-OR Gate :-

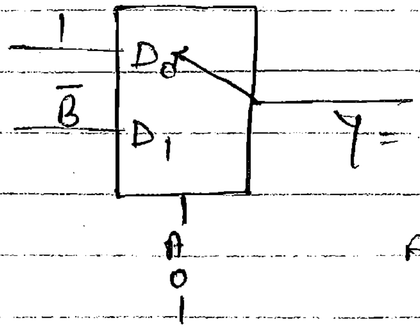


$Y = \bar{A}D_0 + AD_1$

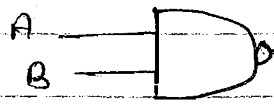


$Y = \bar{A}B + A\bar{B}$

5. NAND Gate :-



$Y = \bar{A}D_0 + AD_1$



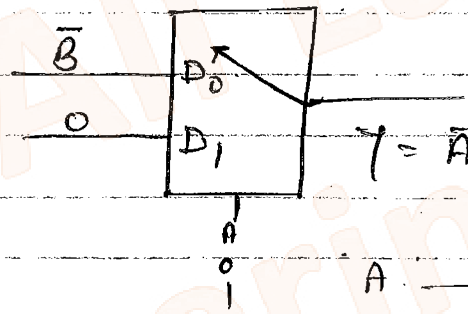
$Y = \overline{AB}$

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

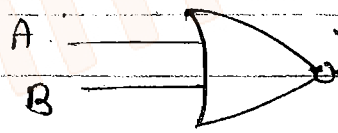
$Y = \bar{A} + A\bar{B}$

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

6. NOR Gate :-



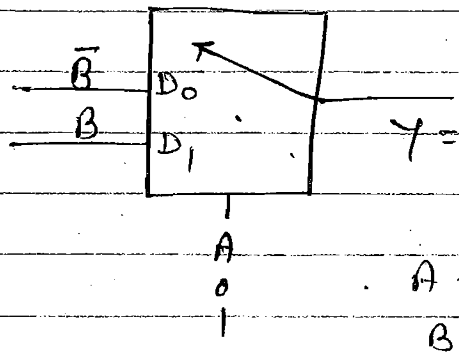
$Y = \bar{A}D_0 + AD_1$



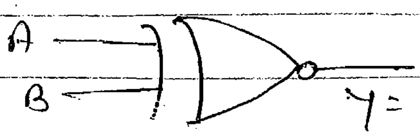
$Y = \overline{A+B}$

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

7. Ex-NOR Gate :-



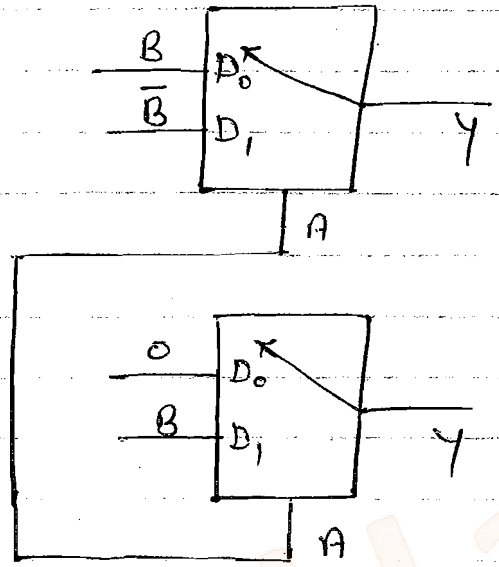
$Y = \bar{A}D_0 + AD_1$



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

Ques Impliment half adder by using 2x1 mux.

Sol^n

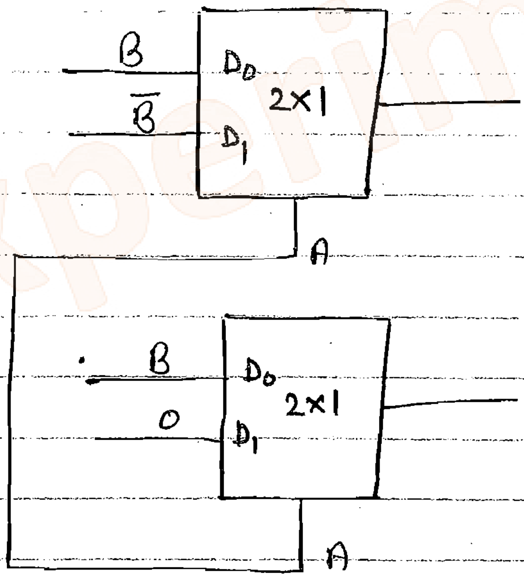


Sum = $A \oplus B$

Carry = $A \cdot B$

Ques Impliment half subtractor by using 2x1 Mux.

Ans



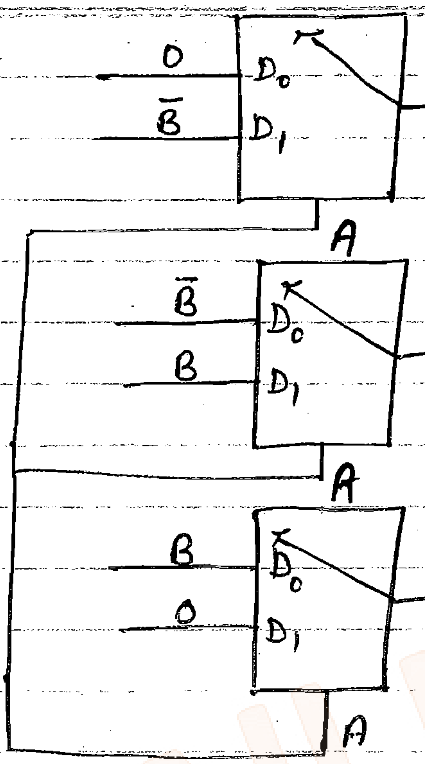
Diff. = $A \oplus B$

Borrow = $\bar{A}B$

Ques By Using 2x1 Mux impliment single bit comprator.

Sol^n

$\because$ Single bit comparator has 3 o/ps (for $A > B$, $A = B$ & $A < B$) So we use 3, 2x1 Mux.



For $A > B$

$Y = A\bar{B}$

for $A = B$

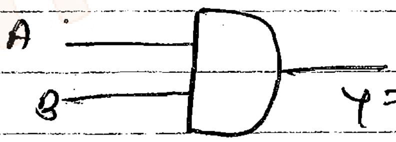
$Y = A \odot B = \bar{A}\bar{B} + AB$

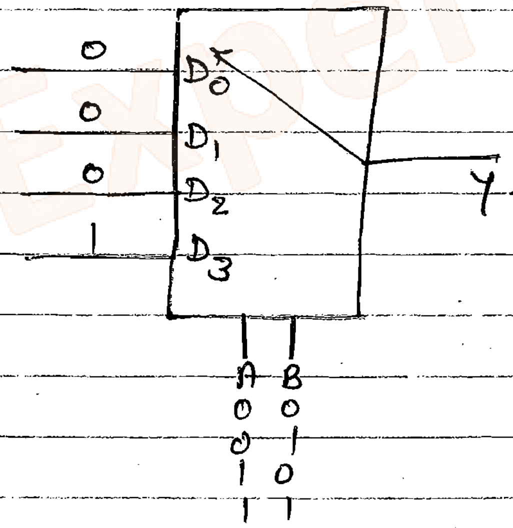
$A < B$

$Y = \bar{A}B.$

2. Construct gate by using 4x1 Muxs.

1 AND Gate :-

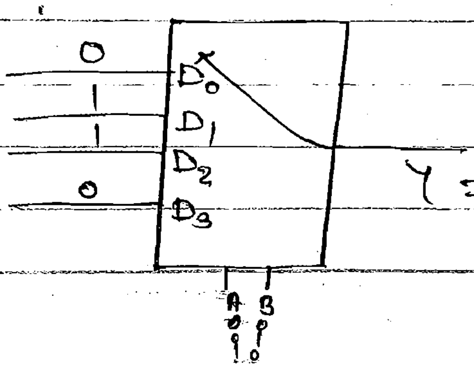




$Y = AB$

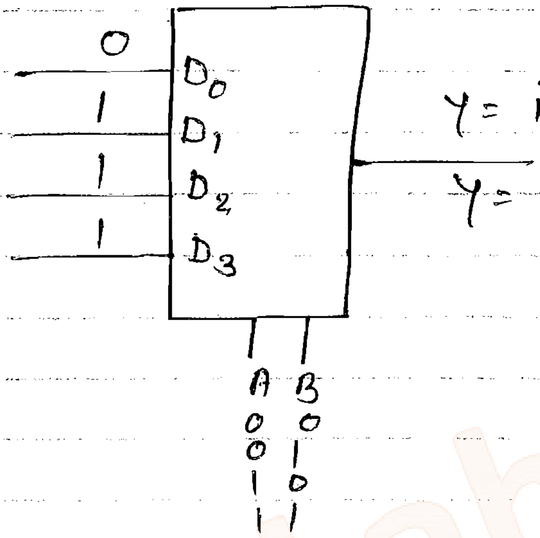
A	B	Y
0	0	0
0	1	0
1	0	0
1	1	1

2. Ex-OR Gate :-



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

3 OR Gate :-

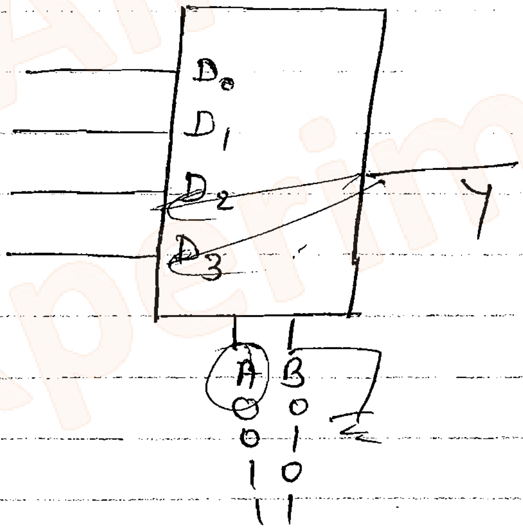


$Y = \bar{A}B + A\bar{B} + AB = \bar{A}B + A(\bar{B}+B)$

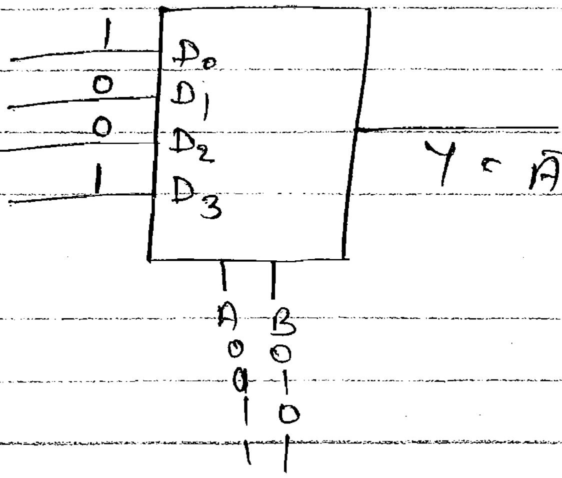
$= \bar{A}B + A = (\bar{A}+A)(A+B)$

$Y = A+B$

4. NOT Gate :-

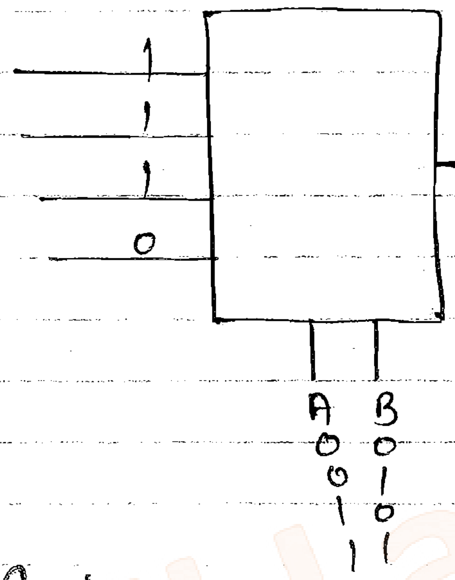


5. Ex-NOR Gate -



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

6. NAND Gate :-



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

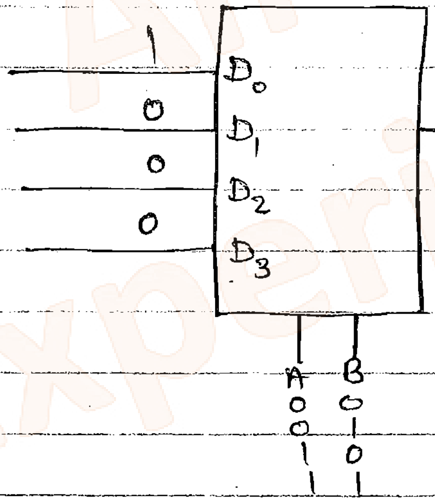
$Y = \bar{A} + A\bar{B} = (\bar{A}+A)(\bar{A}+\bar{B})$

$Y = \overline{AB}$

$\because \bar{A}+A = 1$

$\bar{A}+\bar{B} = \overline{AB}$

7. NOR Gate :-



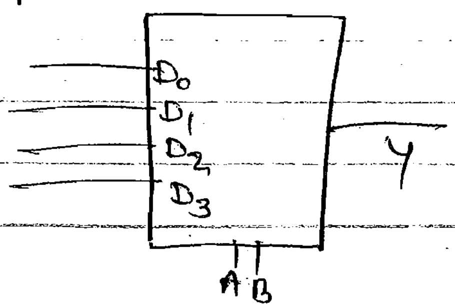
$Y = \bar{A}.\bar{B} = \overline{(A+B)}$

* Type III :-

Identification of number of Mux required for the implimentation of higher order mux by using lower order Mux.

Ques Impliment 4x1 Mux by using 2x1 Mux.

Sol^n

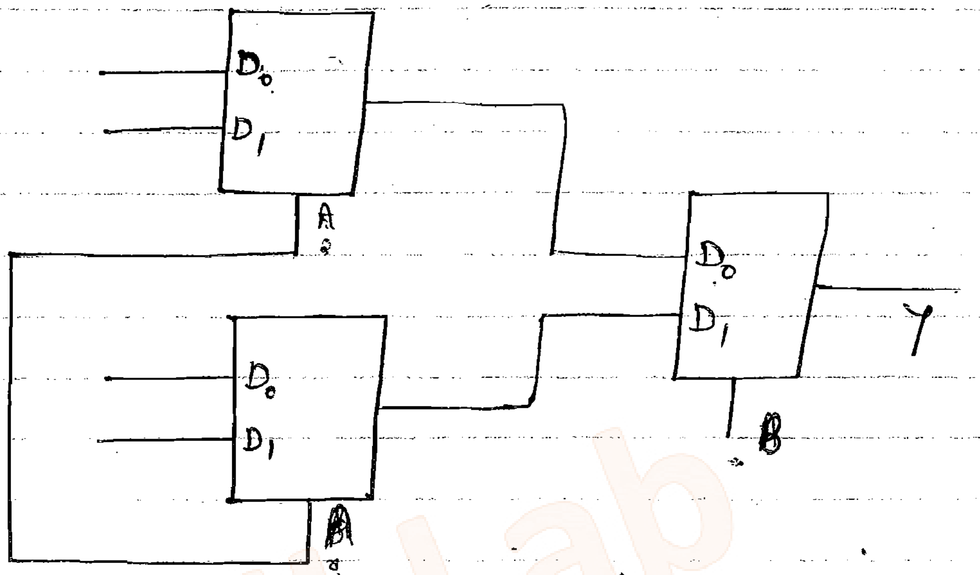


$4\times1 \xrightarrow[2+1=3]{} 2\times1$

$8\times1 \xrightarrow[4+2+1=7]{} 2\times1$

$16\times1 \xrightarrow[8+4+2+1=15]{} 2\times1$

Similarly, $2^n \times 1 \xrightarrow[2^n - 1]{} 2 \times 1$.



4 x 1 (Mux)

Ques Impliment 8x1 Mux by using 4x1 Mux.

Sol^n

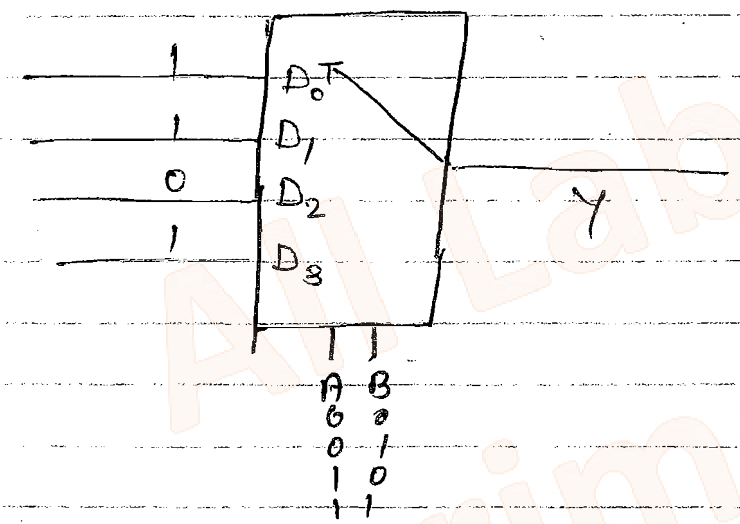
26/July/2014

Type - IV

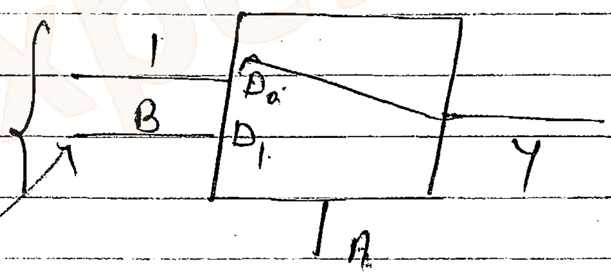
Implimentation of higher order function by using lower order Mux.

Ques Impliment $f(a,b) = \sum m(0,1,3)$ by using 4x1 mux. and 2x1 mux.

Sol^n



for 2x1 mux:-



	D_0	D_1
$\bar{B}$	0 1	2 0
B	1 1	3 1
	1	B

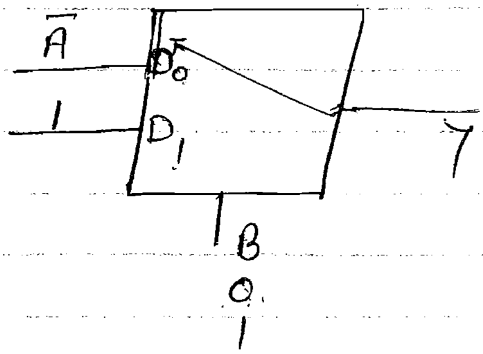
A	B
0	0
0	1
1	0
1	1

Ques Impliment $f(a,b) = \sum(0,1,3)$ by using 2x1 mux by select line B.

Here value of $A=0$ so it is 0 no. box in $\bar{A}$
Again when $B=1$ then rotatory switch goes to D_1 here value of $A=0$ so it is fill in $\bar{A}$ and it is 1 no. box

A	B
0	0
0	1
1	0
1	1

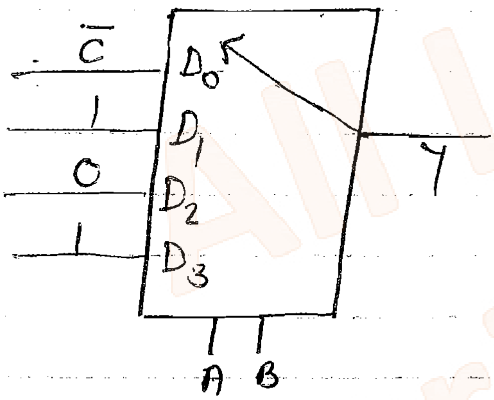
Sol^n



	D_0	D_1
$\bar{A}$	0_1	1_1
A	2_0	3_1
	$\bar{A}$	1

Que For the given function $f(a,b,c) = \Sigma m(0,2,3,6,7)$ impliment the function by using 4×1 Mux.

Sol^n



	A	B
0 ←	0	0
1 ←	0	1
2 ←	1	0
3 ←	1	1

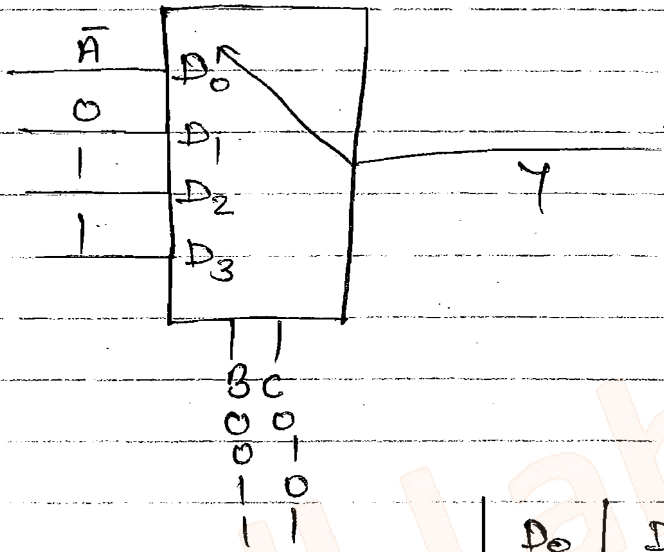
	A	B	C
0 ←	0	0	0
1 ←	0	0	1
2 ←	0	1	0
3 ←	0	1	1
4 ←	1	0	0
5 ←	1	0	1
6 ←	1	1	0
7 ←	1	1	1

	D_0	D_1	D_2	D_3
$\bar{C}$	0_1	2_1	4_0	6_1
C	1_0	3_1	5_0	7_1
	$\bar{C}$	1	0	1

- **Note :-**
Either substitute 1 in the required box or incircle the desired number is also 1.

Ques Repeat the previous ques use BC be the select line.

Sol^n

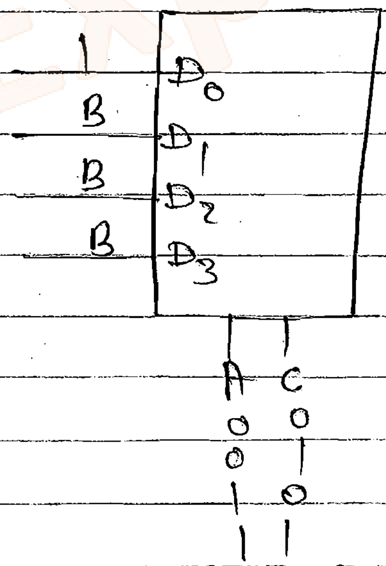


A	B	C	
0	0	0	$\rightarrow$ 0
0	0	1	$\rightarrow$ 1
0	1	0	$\rightarrow$ 2
0	1	1	$\rightarrow$ 3
1	0	0	$\rightarrow$ 4
1	0	1	$\rightarrow$ 5
1	1	0	$\rightarrow$ 6
1	1	1	$\rightarrow$ 7

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

Ques Repeat the same ques us AC be the select line.

Sol^n

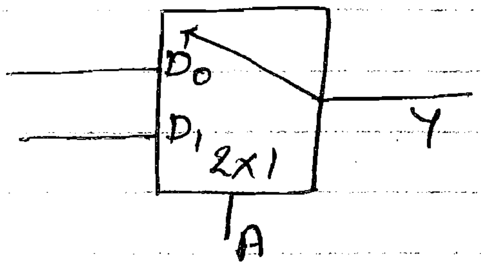


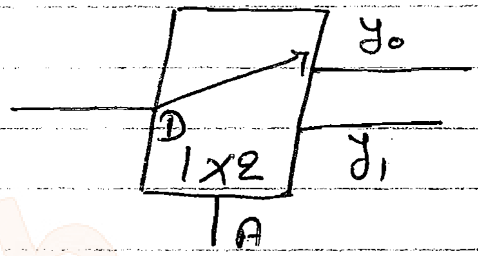
A	B	C	
0	0	0	$\rightarrow$ 0
0	0	1	$\rightarrow$ 1
0	1	0	$\rightarrow$ 2
0	1	1	$\rightarrow$ 3
1	0	0	$\rightarrow$ 4
1	0	1	$\rightarrow$ 5
1	1	0	$\rightarrow$ 6
1	1	1	$\rightarrow$ 7

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

* De-Mux :-

The outer surface of the De-Mux is the mirror image of Mux. The internal structure of De-Mux is equivalent to Decoder.





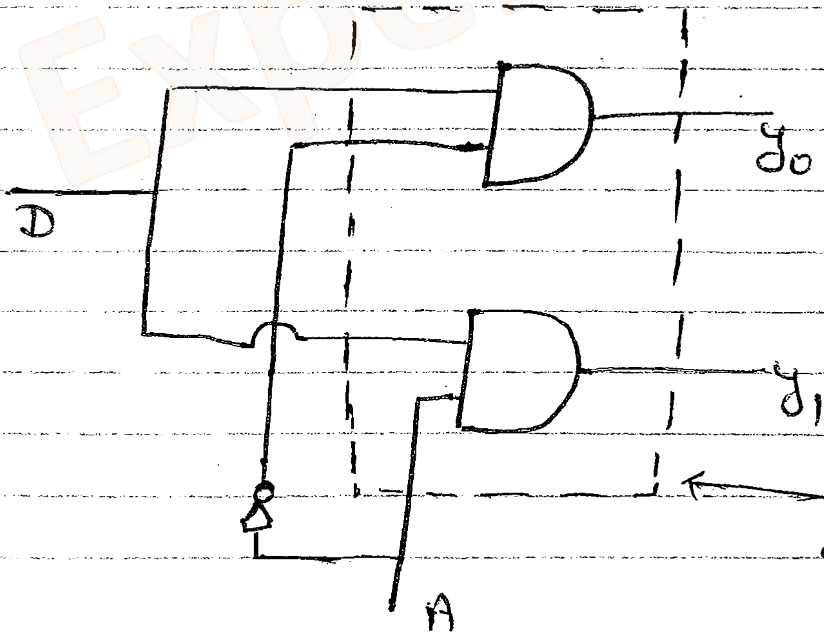
Mux

$$\boxed{(y_0)_{sop} = \bar{A} D}$$

$$\boxed{(y_1)_{sop} = A D}$$

De-Mux

D	y_0	y_1
0	D	0
1	0	D



Internal Structure of De-Mux contains only AND gate.

$1 \times 4 \xrightarrow[2+1=3]{} 1 \times 2$ Demux

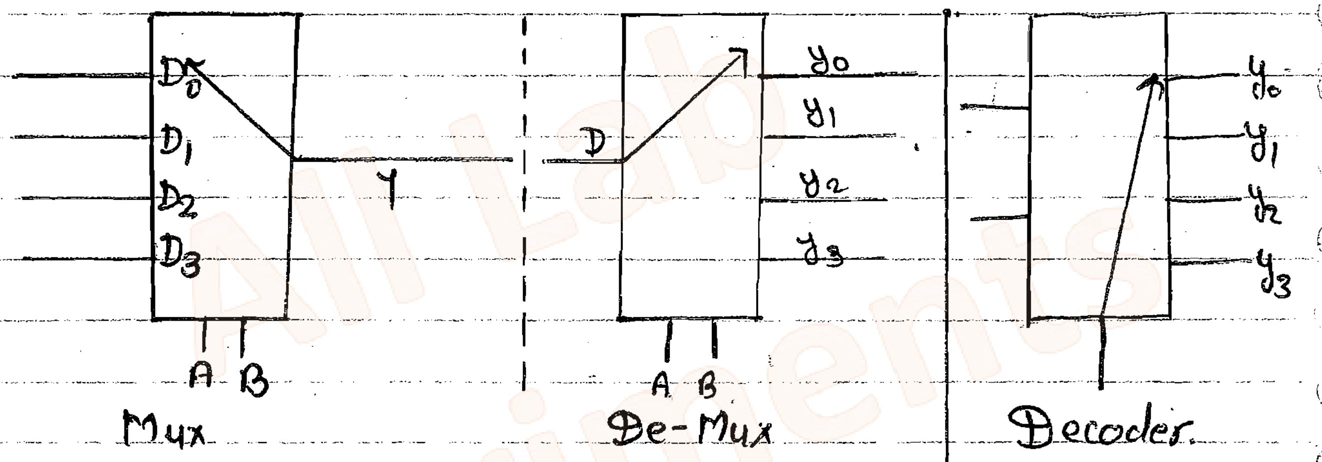
$1 \times 16 \xrightarrow[4+1=5]{} 1 \times 4$

Imp for Gate

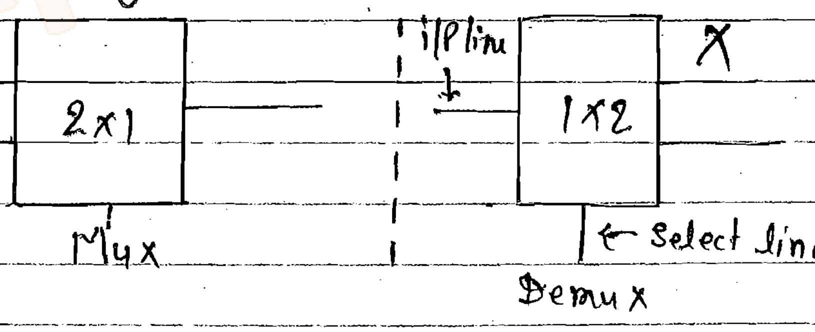
* Decoder :-

The I.C. developed for decoder can be used as De-Mux because internal structures are same.

A De-Mux can be exchanged into decoder by interchanging inputs lines and select line provided the number of input lines should not be equal to number of select line.



We can't construct 1x2 DeMux becouse in 1x2 Demux no. of input lines = no. of select line.

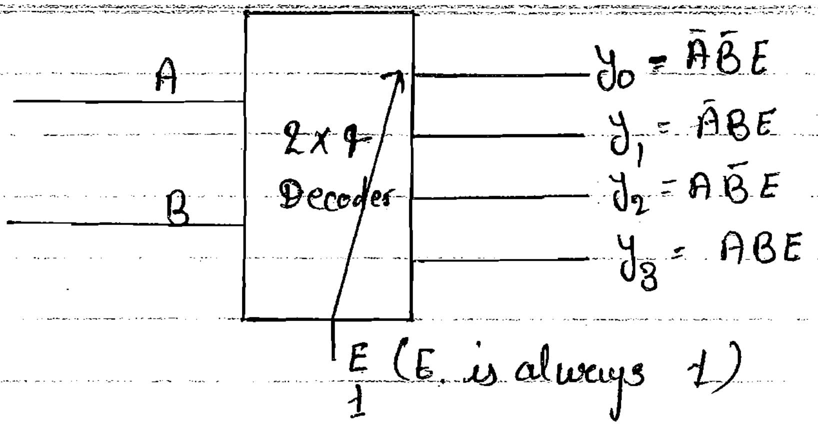


Here no. of i/p line = No. of select line

So it can't use.

• Therefore the lowest possible decoder is 2x4 decoder.

	A	B
$\bar{A}\bar{B}$	0	0
$\bar{A}B$	0	1
$A\bar{B}$	1	0
AB	1	1

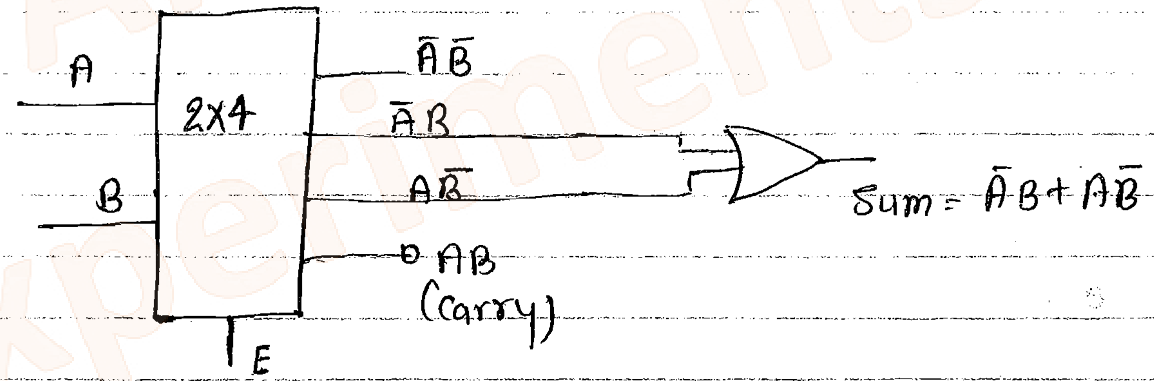


{ 2x4 Decoder }

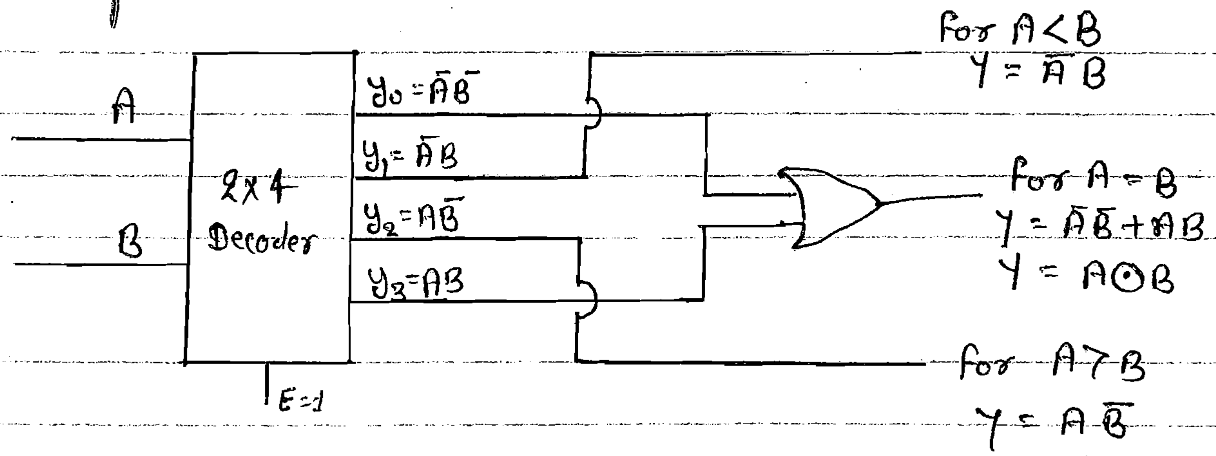
Ques Impliment half adder by using 2x4 Decoder

Sol^n

$$Sum = A \oplus B = \bar{A}B + A\bar{B}$$
$$Carry = A.B$$



Ques By using 2x4 decoder impliment single bit comprator.

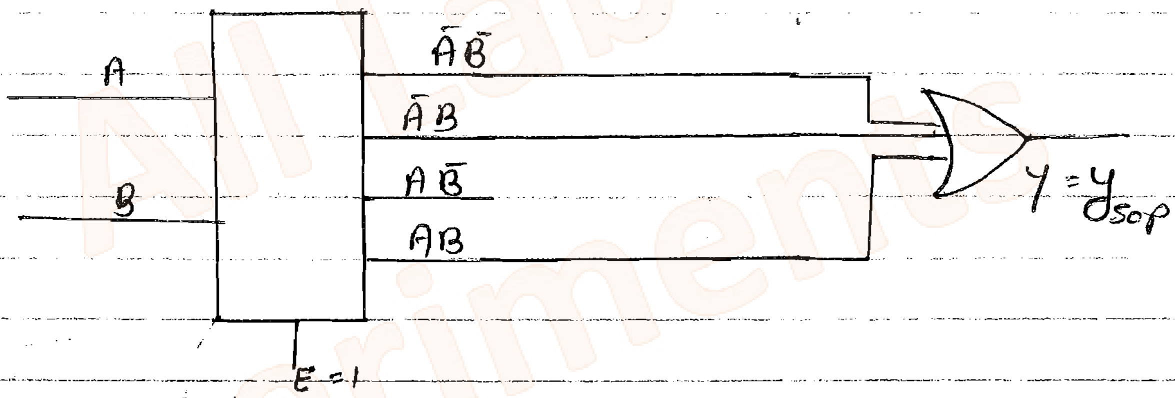


Ques By using 2x4 decoder impliment the function $f(A,B) = \Sigma m(0,1,3)$.

Sol^n

A	B	Y	
0	0	1	$\rightarrow \bar{A}\bar{B}$
0	1	1	$\rightarrow \bar{A}B$
1	0	0	
1	1	1	$\rightarrow AB$

$$Y_{sop} = \bar{A}\bar{B} + \bar{A}B + AB$$



- Note :-
  Decoder is non universal because external gates are used.

* Order of Decoder :-

Mux	Demux	Decoder
4 x 1	1 x 4	$2 \times 4 = 2 \times 2^2$
8 x 1	1 x 8	$3 \times 8 = 2 \times 2^3$.
n x 1	1 x n	$n \times 2^n$

order of decoder.

Full Adder: $y_{Sum} = A \oplus B \oplus C$

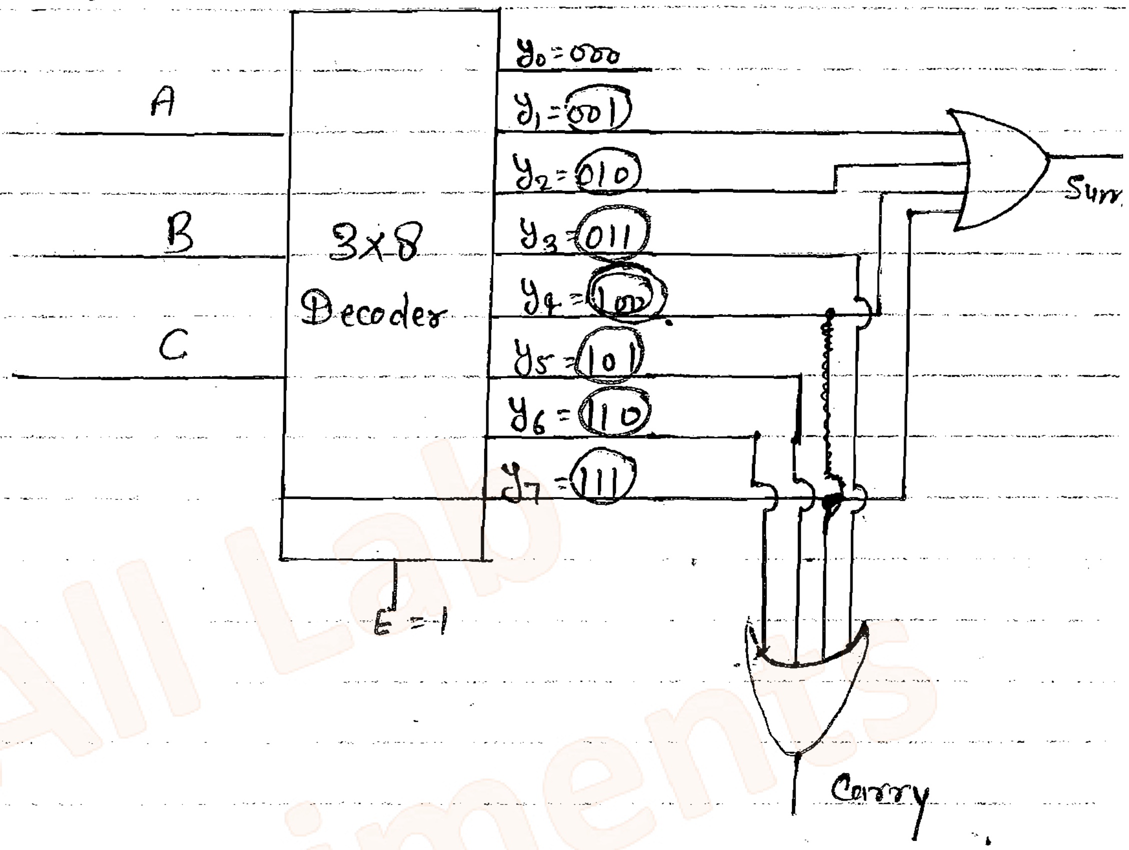
$y_{Carry} = AB + BC + AC = \bar{A}BC + AB\bar{C} + A\bar{B}C + ABC$

Full Subtractor: $y_{Diff} = A \oplus B \oplus C = \bar{A}\bar{B}C + \bar{A}B\bar{C} + A\bar{B}\bar{C} + ABC$

$y_{Borrow} = \bar{A}B + \bar{A}C + BC = \bar{A}\bar{B}C + \bar{A}B\bar{C} + \bar{A}BC + ABC$

Ques Impliment full adder by using 3x8 decoder

Sol^n



Ques Impliment full subtrator by using 3x8 decode

Sol^n:-