Convert the following:-

(a) Decimal to binary → $(34.4375)_{10}$

Sol:-

$$\begin{array}{r|l}
2 & 34 \\
2 & 17 - 0 \\
2 & 8 - 1 \\
2 & 4 - 0 \\
2 & 2 - 0 \\
 & 1 - 0
\end{array}$$

4375

0x4375

$0.4375 \times 2 = 0.5$

$0.875 \times 2 = 1.7$

$34 = 100010 \quad (100010.01)_2$

(b) $(10110.0101)_2$ Binary fraction to decimal AND

Sol:- $10110 =$

$0\times2^0 + 1\times2^1 + 1\times2^2 + 0\times2^3 + 1\times2^4$

$10110 = 0+2+4+0+16$

$= 22$

$.0101 = 0\times2^{-1} + 1\times2^{-2} + 0\times2^{-3} + 1\times2^{-4}$

$= 0 + \frac{1}{4} + 0 + \frac{1}{16}$

$= \frac{4+1}{16}$

$= \cancel{\frac{5}{16}}$ 0.3125

$= 0.3125$

$(10110.0101)_2 \longrightarrow (22.3125)_{10}$

(c) $(26.24)_8$ octal to binary.

Sol:- Octal $= 2^3$;

Binary $= 2^1$;

2	6	.	2	4
010	110		010	100

$\therefore (26.24)_8 \rightarrow (010110.010100)_2$

(d) $(1110.10)_2 \rightarrow$ Binary to decimal.

Sol:- $= 0\times2^0 + 1\times2^1 + 1\times2^2 + 1\times2^3 + 1\times2^{-1} + 0\times2^{-2}$

$= 0+2+4+8+\frac{1}{2}+0$

$= 14 + \frac{1}{2}$ 0.5

$= 14.5$

$(1110.10)_2 \longrightarrow (14.5)_{10}$

(2) Add Binary number 1011 and 101

Sol:-

$$\begin{array}{r} \scriptstyle 1\,1\,1 \\ 1011 \\ +\ \ 101 \\ \hline 10000 \\ \hline \end{array}$$

(3) Find 9's and 10's Complement of the decimal

(i) 98127634.

Sol:- Let, n = 98127634

Substract 'n' from '9'. We get 9's Complement of the decimal.

$$\Rightarrow \begin{array}{r} 99999999 \\ -\ 98127634 \\ \hline 01872365 \\ \hline \end{array}$$

$\therefore$ 9's Complement = 1872365.

10's Complement = 9's Complement + 1

= add '1' to 9's Complement.

$$\Rightarrow \begin{array}{r} 1872365 \\ +1 \\ \hline 1872366 \\ \hline \end{array}$$

$\therefore$ 10's Complement = 1872366.

(ii) 72049900.

Sol: Let, $n = 72049900$

Substract 'n' from '9'. we get 9's Complement of the decimal.

$$\begin{array}{r} 99999999 \\ 72049900 \\ \hline 27950099 \\ \hline \end{array}$$

∴ 9's Complement = 27950099

10's Complement = 9's Complement + 1

$$\begin{array}{r} 27950099 \\ +1 \\ \hline 27950100 \\ \hline \end{array}$$

10's Complement = 27950100.

(4) Find 1's and 2's Complement

(a) 11101010

Sol: 1's Complement = convert 1's as 0's. and 0's as 1's.

11101010 ⇒ 00010101

1's Complement = 00010101.

⇒ 2's Complement = ~~Add~~ 1 to 1's Complement.

$$\begin{array}{r} 00010101 \\ +1 \\ \hline 00010110 \\ \hline \end{array}$$

(b) 0150 − 2100

Sol:- $N = 2100$; $r = 10$; $n = 4$

$= 10^4 - 2100$

$= 10,000 - 2100$

$= 7900 \Rightarrow$ 10's Complement

Step-2:- $M + r(N) = sum$

$M = 0150$

10's Complement $N = 7900$

$= 8050$

Step-3:- r's of sum

$r's = r^n - N$

$= 10^4 - 8050$

$= 10,000 - 8050$

$\boxed{M > N}$ $= 1,950$

(6) Perform substraction on the Unsigned no's Using 2's Complement.

(a) 11011 − 11001

Sol:-
$$\underset{M}{\overset{X}{11011}} - \underset{N}{\overset{Y}{11001}}$$

1's Complement 11001 − 00110 → 1's Complement

$$\begin{array}{r} 00110 \\ +1 \\ \hline 00111 \end{array} \rightarrow \text{2's Complement}$$

$$\begin{array}{rr} & 1\ 1\ 1\ 1 \\ Y & 0\,0\,1\,1\,1 \\ X & 1\,1\,0\,1\,1 \\ \hline & 1\,0\,0\,0\,1\,0 \end{array}$$

1,00,010 − 00010

(b) 101010 − 101011

Sol:- 101011 = 010100

$$\begin{array}{r} 010100 \\ +\;1 \\ \hline \end{array}$$

y $\quad$ 010101 → 2's Complement

y = 010101

x = 101010

$$\begin{array}{r} 010101 \\ 101010 \\ \hline 111111 \\ \hline \end{array}$$

Sum = 111111 → no end carry.

(1)

(i) $(x,y,z) = \Sigma(0,2,4,5,6)$

x \ yz	00	01	11	10
x' 0	m_0 1	m_1	m_3	m_2 1
x 1	m_4 1	m_5 1	m_7	m_6 1

$A = z'$

$B = xy'$

$f(x,y,z) = xy' + z'$

(ii) $F = A'C + A'B + AB'C + BC$ express in sum of m terms.

A \ BC	00	01	11	10
A' 0	m_0	m_1 1	m_3 1	m_2 1
A 1	m_4	m_5 1	m_7 1	m_6

$A = C$

$B = A'B$

$F(A,B,C) = \Sigma(1,2,3,5,7)$

$= A'B + C$

(iii) $F(x,y,z) = \Sigma(0,2,6,7)$

x \ yz	00	01	11	10
0	m_0 1	m_1	m_3	m_2 1
1	m_4	m_5	m_7 1	m_6 1

$A = xz'$

$B = xy$

$F(x,y,z) = xy + xz'$

(iv) $F(a,b,c) = \Sigma(0,1,2,3,7)$

$a \backslash bc$	00	01	11	10
0	1	1	1	1
1			1	

$A = a'$

$B = bc$

$F = a' + bc$

(v) $F(A,B,C) = \Sigma(0,2,3,4,6)$

$A \backslash BC$	00	01	11	10
0	1		1	1
1	1			1

$A = C'$

$B = A'B$

$F = A'B + C$

(vi) $xy' + x'y'z' + x'yz'$

$x \backslash yz$	00	01	11	10
0	1			1
1			1	1

$A = x'z'$

$B = xy$

$F = xy + x'z'$

(vii) $x'y' + yz + x'yz'$

$x \backslash yz$	00	01	11	10
0	1	1	1	1
1			1	

$A = x'$

$B = yz$

$F = x' + yz$

(viii) $A'B + BC' + B'C'$

$A \backslash BC$	00	01	11	10
0	1		1	1
1				

$A = A'C'$

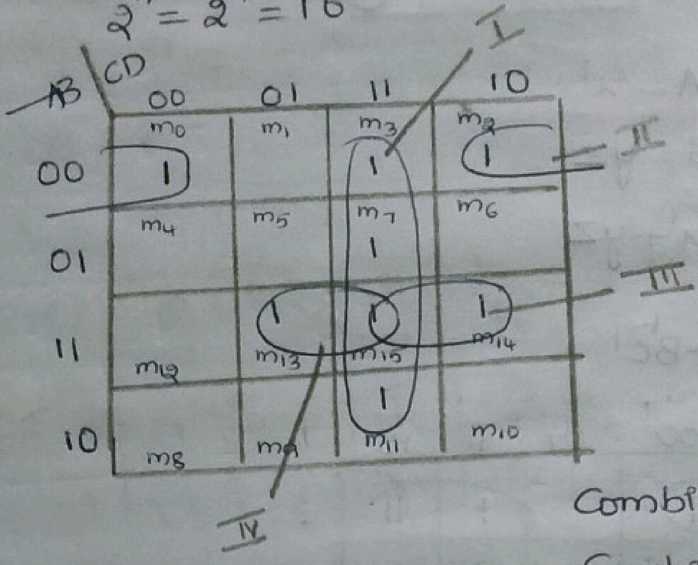
$B = A'B$

$F = A'B + A'C'$

(1) Explain 4 variable and 3 variable K-map with explain.

Ex :- $F(A,B,C,D) = \Sigma m(0,2,3,7,11,13,14,15)$

$2^n = 2^4 = 16$



Combine 2's → literal is …

Combine 4's → 2 litera…

Combine 8's → 3 literal…

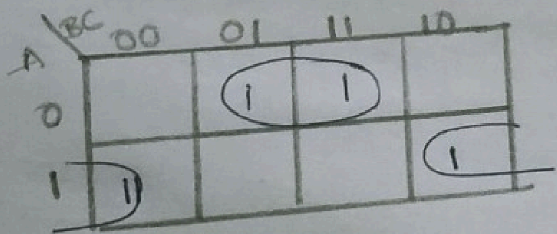
Combine 16's → 4 liter…

$f = \text{I} + \text{II} + \text{III} + \text{IV}$

$= CD + A'B'D' + ABC + ABD$

3 variable :-

$A'B'C + A'BC + AB'C' + ABC'$



$F = \text{I} + \text{II}$

$= A'C + AC'$

(2) Explain Don't care Condition with example:-

Sol:- $F(w,x,y,z) = \Sigma(1,3,7,11,15)$

$d(wxyz) = \Sigma(0,2,5) = d$

d (or) x symbol

→ Sum of product $x = 1$

→ product of sum $x \neq 0$

→ Large group

→ Reduce taking don't care in a group

wx \ yz	00	01	11	10
00	X	1	1	X
01		X	1	
11			1	
10			1	

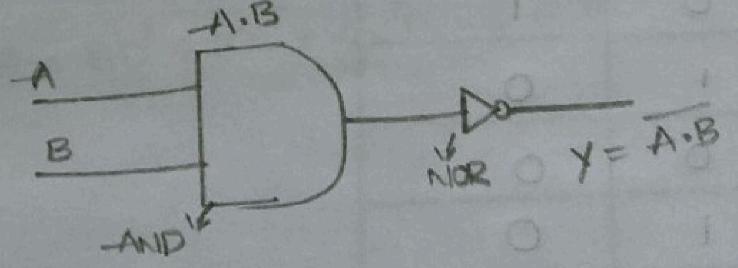
$Y = YZ + W'X' \rightarrow ZX$

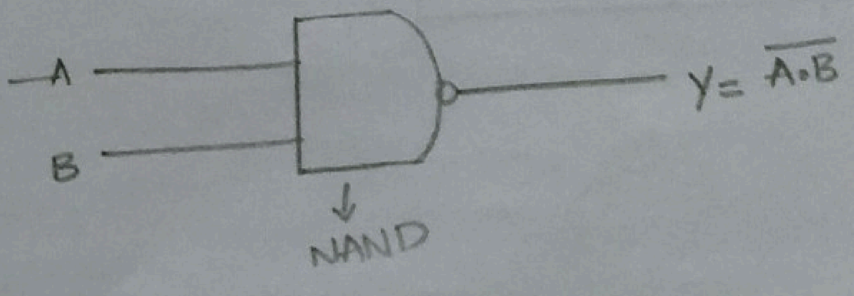
$Y = YZ + W'Z$

(3) Explain NAND and NOR circuits.

Sol:- NAND ~~Gate~~ :- NAND gate it is a Combination of AND & NOT gates. It has two or more input & only one output.

→ when all the input are 'HIGH' the output is 'LOW' if any one or both input are 'LOW' then the output is 'HIGH'.

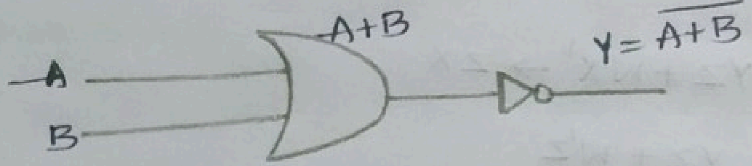


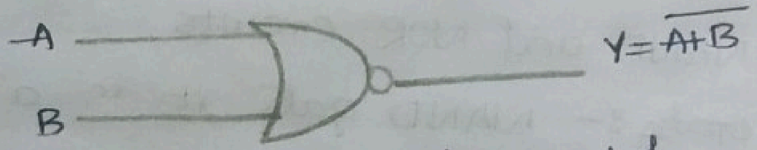


Truth table :-

input		output
A	B	$Y = \overline{A \cdot B}$
0	0	1
0	1	1
1	0	1
1	1	0

NOR Gate :- It is a Combination of NOT and OR Gate. It has two or more input only one output. The output is 'HIGH' only when all the input are "LOW". If one or both input are 'HIGH' then the output is "LOW".





NOR gate symbol

Truth table

Input		output
A	B	$Y = \overline{A+B}$
0	0	1
0	1	0
1	0	0
1	1	0

(4) Simplify sum of products and products of sum problem.

Sum of product (SOP) $\rightarrow$ sum = +

Product of sum (POS) $\rightarrow$ product = x

<u>3 Variable</u> :- A, B, C

$\rightarrow (AB) + (BC)$

<u>4 Variable</u> :- A, B, C, D

Eg:- $(ABC) + (BCD) + (CDA) \rightarrow$ standard form

$\underbrace{(A'BCD) + (AB'CD)}_{\text{Sum of product}} + (ABC'D) + (ABCD') \rightarrow$ Concinoal

All forms are involved

Product of Sum:- (POS)

Eg:- 3 variable (A, B, C)

$(A+B).(B+C) \rightarrow$ standard product of sum

$(A'+B+C)(A+B'+C)(A+B+C') =$ Canonical product of sum.

<u>Sum of product Example</u>:-

$F(A, B, C, D) = ?(0, 2, 5, 8, 9, 10)$.

AB \ CD	00	01	11	10
00	1 m_0	1 m_1	m_3	1 m_2
01	m_4	m_5	m_7	m_6
11	m_{12}	m_{13}	m_{15}	m_{14}
10	1 m_8	1 m_9	m_{11}	1 m_{10}

$F = B'D' + B'C' + A'C'D \rightarrow$ SOP product of sum:-

AB\CD	00	01	11	10
00			0	
01	0		0	0
11	0	0	0	0
10			0	

$P' = AB + BD + CD$

By using demorgan's law

$(A+B)' = A'B$

$P' = (AB + CD + BD)'$

$P = (AB)' + (CD)'(BD)'$

$= (A'+B')(C'+D').(B'+D')$

↓ product of sum

(1) Explain (or) Define Combination Circuit.

sol:- Combination Circuit

(1) Half adder

(2) Full adder

(3) Multiplexer

2:1
4:1
8:1
16:1

(4) Demultiplexer

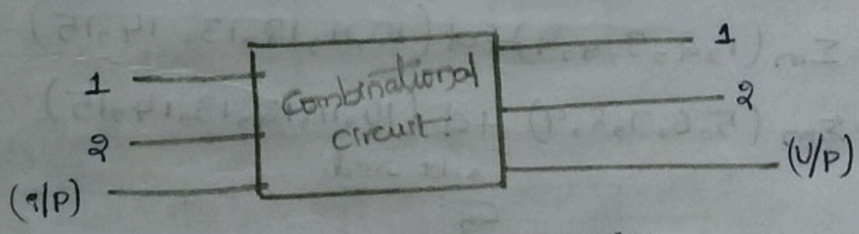
1:2
1:4
1:8
1:16

(5) Encoders

(6) decoders

<u>Combinational Circuit</u> :-

* The output of the circuit depends upon the Combination of input variables.
* It is a logic circuit having gates inbuilt (or) integrated in it.
* It doesn't use any memory.
* 'n' number of inputs, 'm' no. of outputs



It does not use any memory.

It depends on present input.

* It design with logic gates. It does not depend on provides inputs only depends on present inputs is called Combinational Circuit.

(2) Explain logic diagram for BCD to excess 3 Code Converter.

Designing implementation

Decimal	BCD Code					Excess-3 Code			
NO	B_4	B_3	B_2	B_1		X_4	X_3	X_2	X_1
0	0	0	0	0	3	0	0	1	1
1	0	0	0	1	4	0	1	0	0
2	0	0	1	0	5	0	1	0	1
3	0	0	1	1	6	0	1	1	0
4	0	1	0	0	7	0	1	1	1
5	0	1	0	1	8	1	0	0	0
6	0	1	1	0	9	1	0	0	1
7	0	1	1	1	10	1	0	1	0
8	1	0	0	0	11	1	0	1	1
9	1	0	0	1	12	1	1	0	0

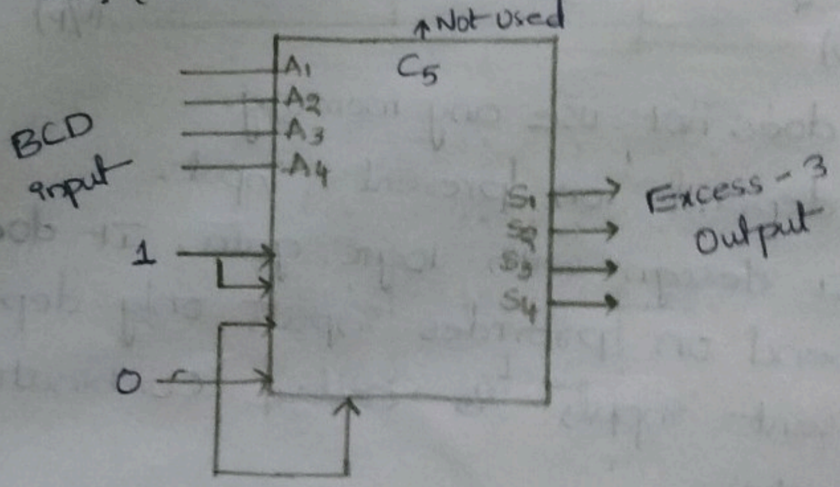
$X_1 = \Sigma m(0,2,4,6,8) +$ don't care $(10,11,12,13,14,15)$

min term

$X_2 = \Sigma m(0,3,4,7,8) + d\,(10,11,12,13,14,15)$

$X_3 = \Sigma m(1,2,3,4,9) + d\,(10,11,12,13,14,15)$

$X_4 = \Sigma m(5,6,7,8,9) + d\,(10,11,12,13,14,15)$



BCD-to excess-3 Code Converter

K-Map

$X_1 = \Sigma m(0, 2, 4, 6, 8) + d(10, 11, 12, 13, 14, 15)$

$B_2B_1 \backslash B_4B_3$	00	01	11	10
00	1	0	0	1
01	1	0	0	1
11	X	X	X	X
10	1	0	X	X

$X_1' = \overline{B_2}$

Σmap for X_2

$X_2 = \Sigma m(0, 3, 4, 7, 8) + d(10, 11, 12, 13, 14, 15)$

$B_2B_1 \backslash B_4B_3$	00	01	11	10
00	1	1	X	1
01	0	0	X	0
11	1	1	X	X
10	0	0	X	X

$\overline{B_2}\,\overline{B_1} + B_2B_1$

$X_3 = \Sigma m(1, 2, 3, 4, 9) + d(10, 11, 12, 13, 14, 15)$

$B_2B_1 \backslash B_4B_3$	00	01	11	10
00	0	1	X	0
01	1	0	X	1
11	1	0	X	X
10	1	0	X	X

$\overline{B_3}\,\overline{B_2}\,\overline{B_1} + \overline{B_2}B_1 + \overline{B_3}B_2$

$X_4 = \Sigma m(5, 6, 7, 8, 9) + d(10, 11, 12, 13, 14, 15)$

$B_2B_1 \backslash B_4B_3$	00	01	11	10
00	0	0	X	1
01	0	1	X	1
11	0	1	X	X
10	0	1	X	X

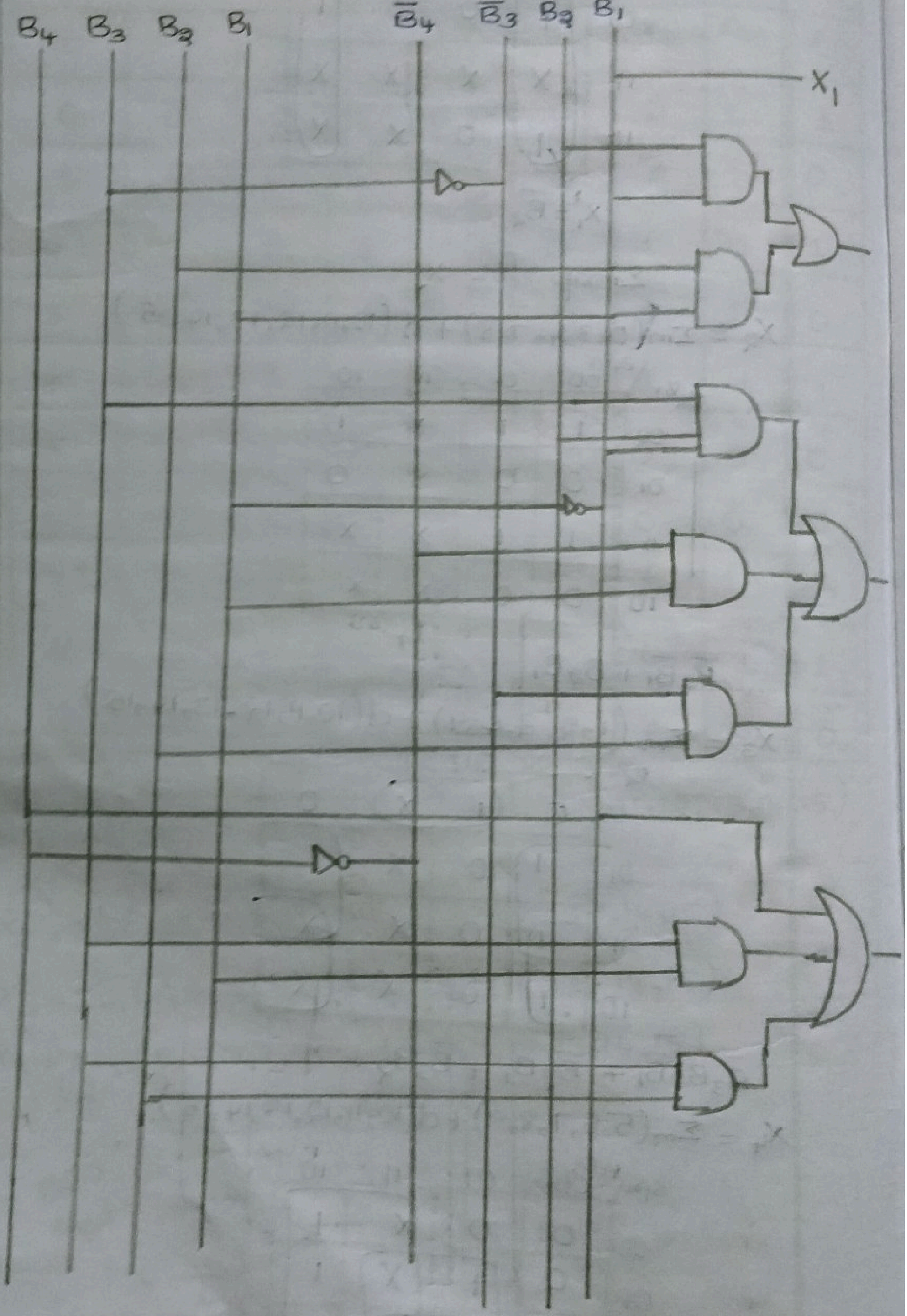
$B_4 + B_3B_2 + B_3B_2$

$X_1 = \overline{B_1}$

$X_2 = \overline{B_2}\,\overline{B_1} + B_2 B_1$

$X_3 = \overline{B_3}\,\overline{B_2}\,\overline{B_1} + \overline{B_3} B_1 + \overline{B_3} B_2$

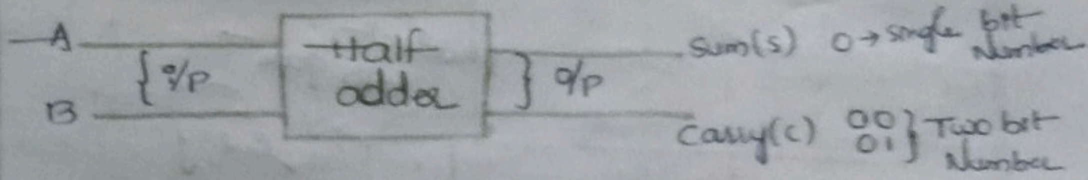
$X_4 = B_4 + B_3 B_1 + B_3 B_2$



(3) Explain half-adder and full-adder with an example.

Half - adder

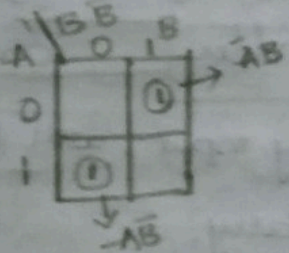
Designed to add single bit number
It has two inputs and two outputs



0+0 (OR) GATE

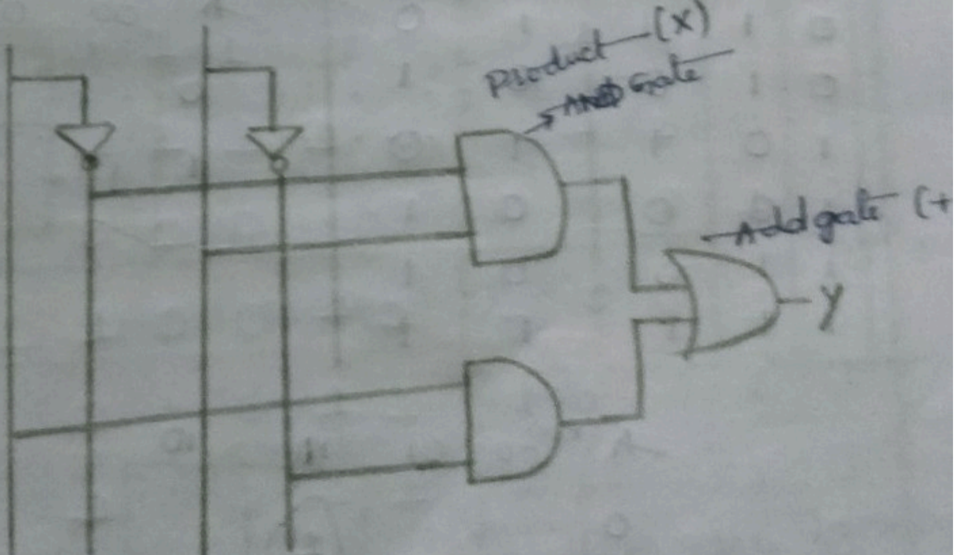
Truth table

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



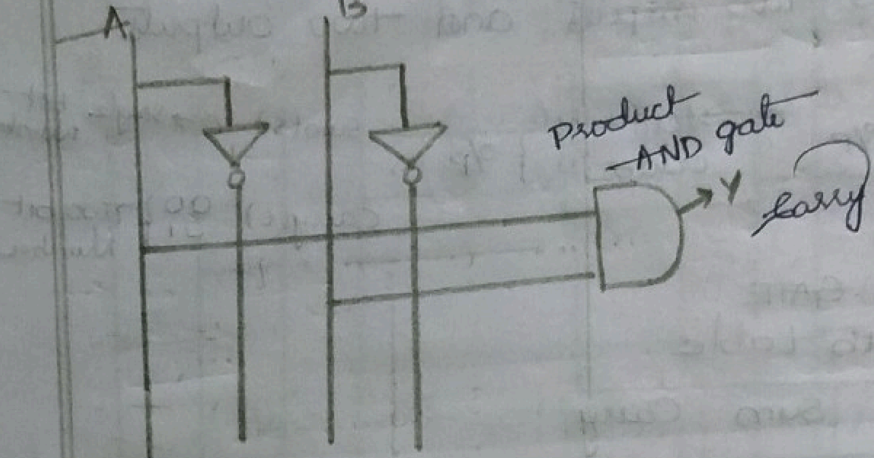
$A\bar{B} + \bar{A}B$.

Sum:-



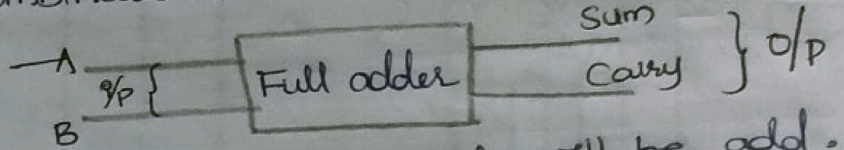
* Carry

A \ B	0	1
0		
1		1 → AB



* **Full adder** :-

→ It is designed to add ~~two~~ single bit numbers with a carry (or) Arithmetic logical Combinational



cin → Carry input will be add.

A	B	Cin	Sum	Carry
0	0	0	0	0
0	0	1	1	0
0	1	1	1	0
0	1	0	0	1
1	0	1	1	0
1	0	0	0	1
1	1	0	0	1
1	1	1	1	1

A \ Bcin	00	01	11	10
0		1		1
1	1		1	

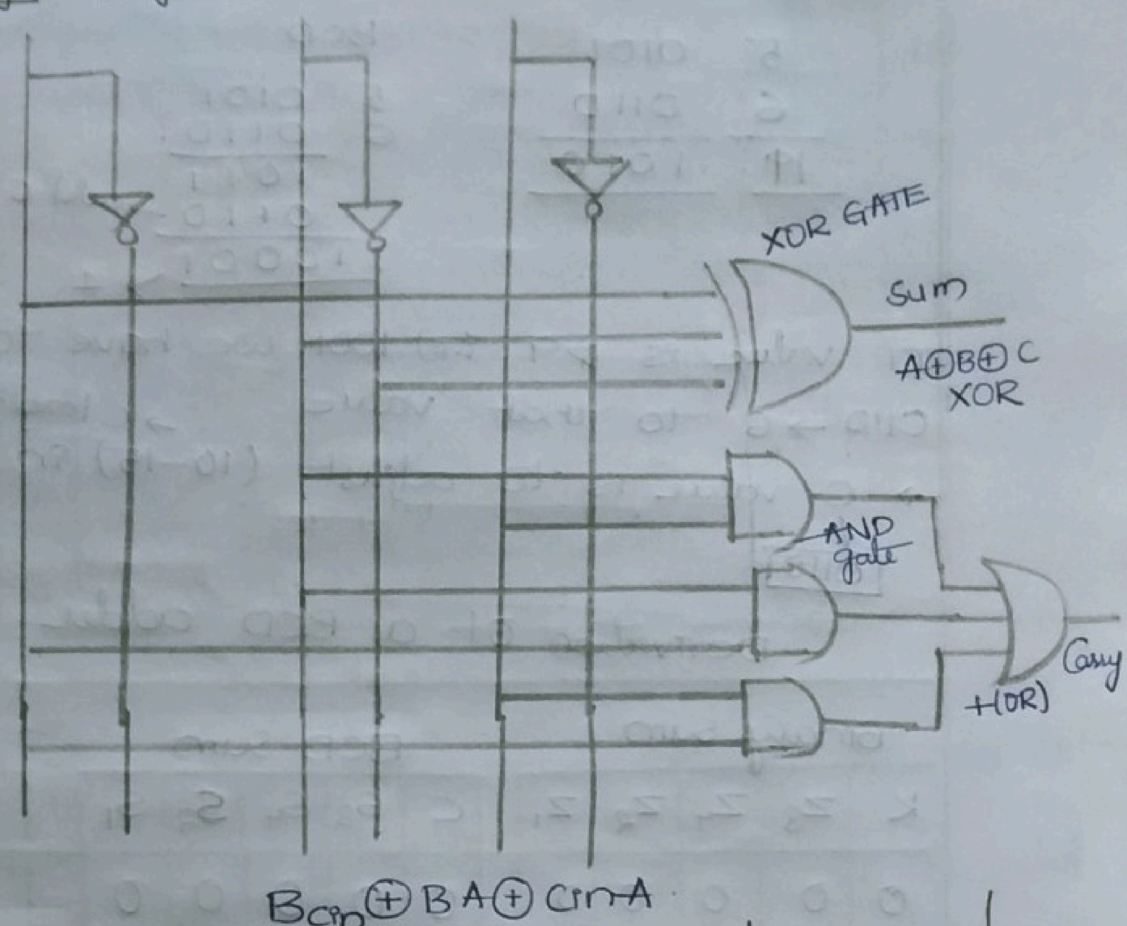
Carry

A \ Bcin	00	01	10	10
0			1	
1		1	1	1

↓ CinA   ↓ BA

$B_{cin} + BA + C_{in}A$.

* Logic diagram:-



(4) Explain BCD adder with an explain example diagram.

Sol:- Adding 4 bit BCD Code

* In 4 bit we have max. value upto $9 \rightarrow 1001$
* So, we can add. max value up to 9+9+1.
  1 is the input carry.

  $\boxed{9+9+1=19}$

  Addend Augend

* BCD Addition and binary addition are same up to 9 → from 0 to 1 being

Case-1:- Total is less than or equal to 9.

$$\begin{array}{r} 1 \\ 5 \\ \hline 6 \end{array} \quad \begin{array}{r} \rightarrow 0001 \\ \rightarrow 0101 \\ \hline 0110 \end{array}$$

Up to 9 → 1001 both BCD are binary are going to be same.

Case-2:- Total is >9

5	0101	
6	0110	
11	1010	

BCD

5	0101	
6	0110	
	1011	
	0110	→ add 6
	10001	→ 1

If value is >9 i.e) 1001 we have to add 0110 → 6 to that value

→ 6 length

→ 6 value is to adjust (10–15) in Binary

$\boxed{0110}$

Derivation of a BCD adder

Binary sum					BCD sum					Decimal
K	Z_8	Z_4	Z_2	Z_1	C	S_8	S_4	S_2	S_1	
0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	1	0	0	0	0	1	1
0	0	0	1	0	0	0	0	1	0	2
0	0	0	1	1	0	0	0	1	1	3
0	0	1	0	0	0	0	1	0	0	4
0	0	1	0	1	0	0	1	0	1	5
0	0	1	1	0	0	0	1	1	0	6
0	0	1	1	1	0	0	1	1	1	7
0	1	0	0	0	0	1	0	0	0	8
0	1	0	0	1	0	1	0	0	1	9
0	1	0	1	0	1	1	0	0	0	10
0	1	0	1	1	1	0	0	0	1	11
0	1	1	0	0	1	0	0	1	0	12

K	Z_8	Z_4	Z_2	Z_1
0	1	1	0	1
0	1	1	1	0
0	1	1	1	1
1	0	0	0	0
1	0	0	0	1
1	0	0	1	0
1	0	0	1	1

C	S_8	S_4	S_2	S_1	
1	0	0	1	1	13
1	0	1	0	0	14
1	0	1	0	1	15
1	0	1	1	0	16
1	0	1	1	1	17
1	1	0	0	0	18
1	1	0	0	1	19

→ From truth table

If C=0 Comes Under case 1 so no change is required.

If C=1 Comes Under case 2 we have to add 0110 to the Output.

To find the relation we have to find Boolean function.

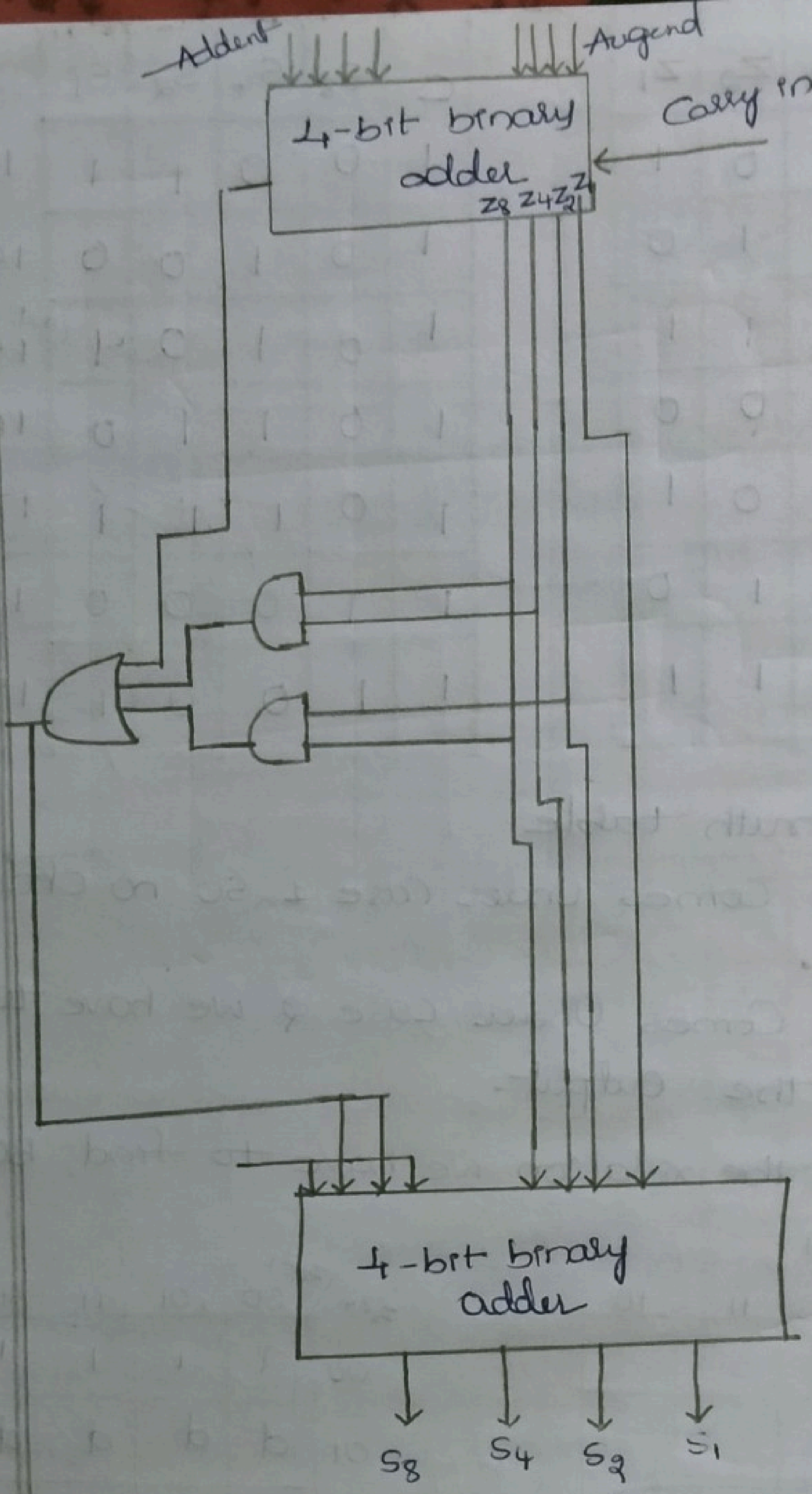
z_8z_4 \ z_2z_1	00	01	11	10
00				
01				
11	1	1	1	1
10			1	1

K=0

z_8z_4 \ z_2z_1	00	01	11	10
00	1	1	1	1
01	d	d	d	d
11	d	d	d	d
10	d	d	d	d

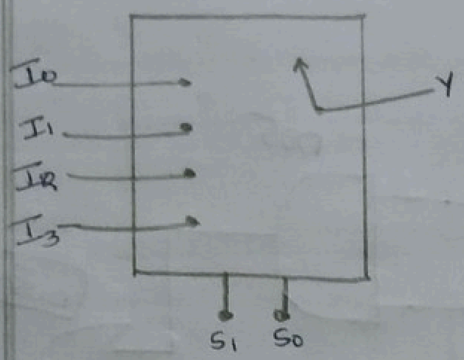
K=1

$$C = K + Z_8Z_4 + Z_8Z_2$$



→ BLOCK DIAGRAM OF A BCD-ADDER

(5) Explain binary multiplier with example



$m = 2^{n} \rightarrow$ select

i/p. $m = 2^{2} = 4$ 2 line select

$\boxed{m=4} \rightarrow$ inputs

$m = 2^{3} \rightarrow$ select line

$= 8$ inputs

$m = 2^{4} \rightarrow$ select line

$= 16$ inputs

Advantages:- Any input but 1 is output it depends upon selection lines(s)

* Reduces no. of wires.
* Reduces Complex (or) Cost.
* MUX design implement various design.
* Simplify the circuit it is not Complex.

Types of MUX :-

2:1 MUX

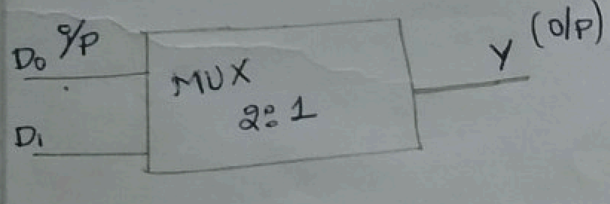
4:1 MUX

8:1 MUX

16:1 MUX

32:1 MUX

2:1 MUX

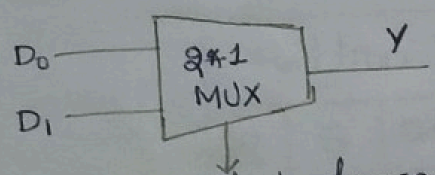


$m = 2^{n} \rightarrow$ select line

$m = 2^{1} = 2$

$m = 2$

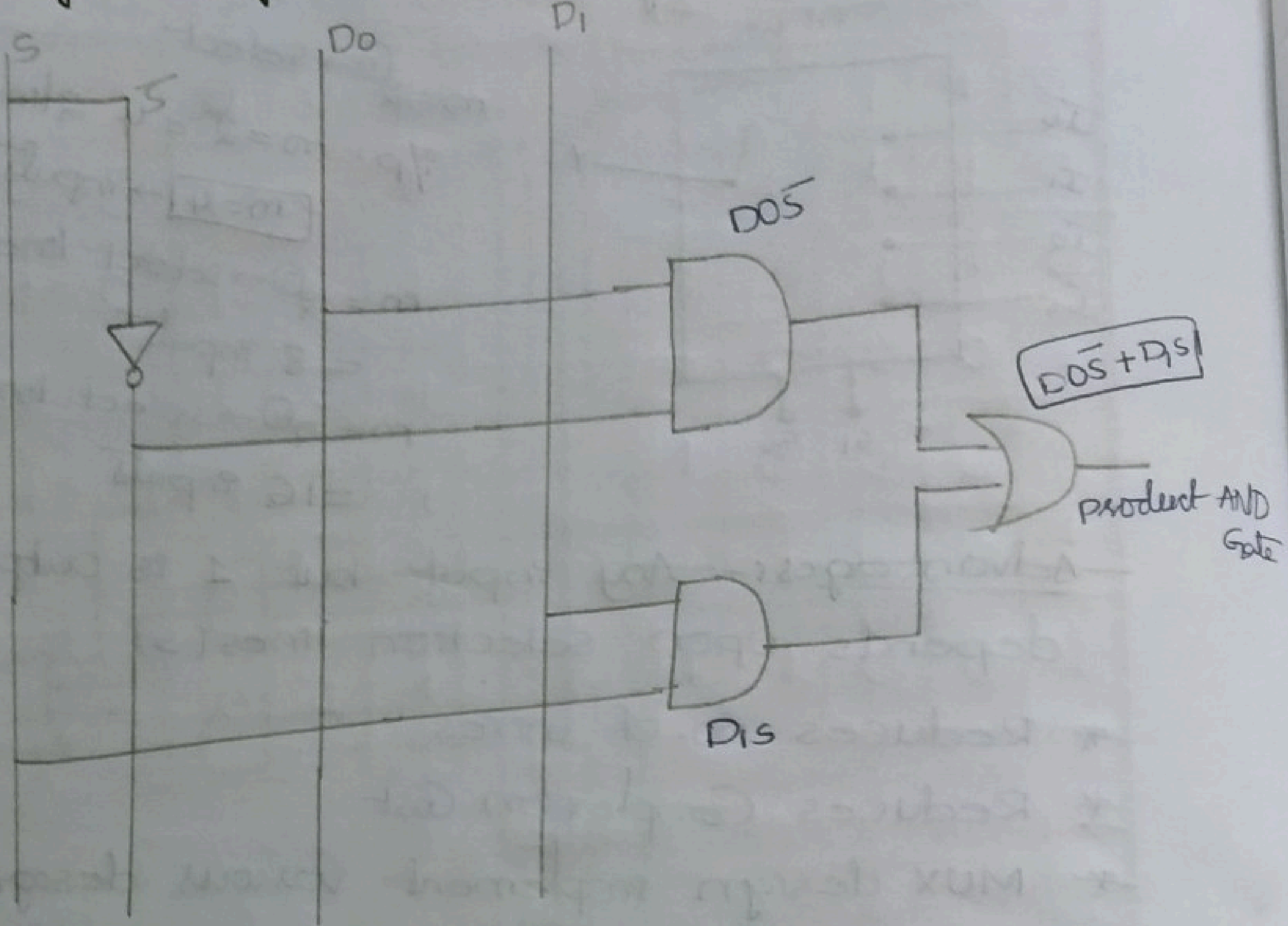
S	Y
0	O_0
1	O_1



Block diagram

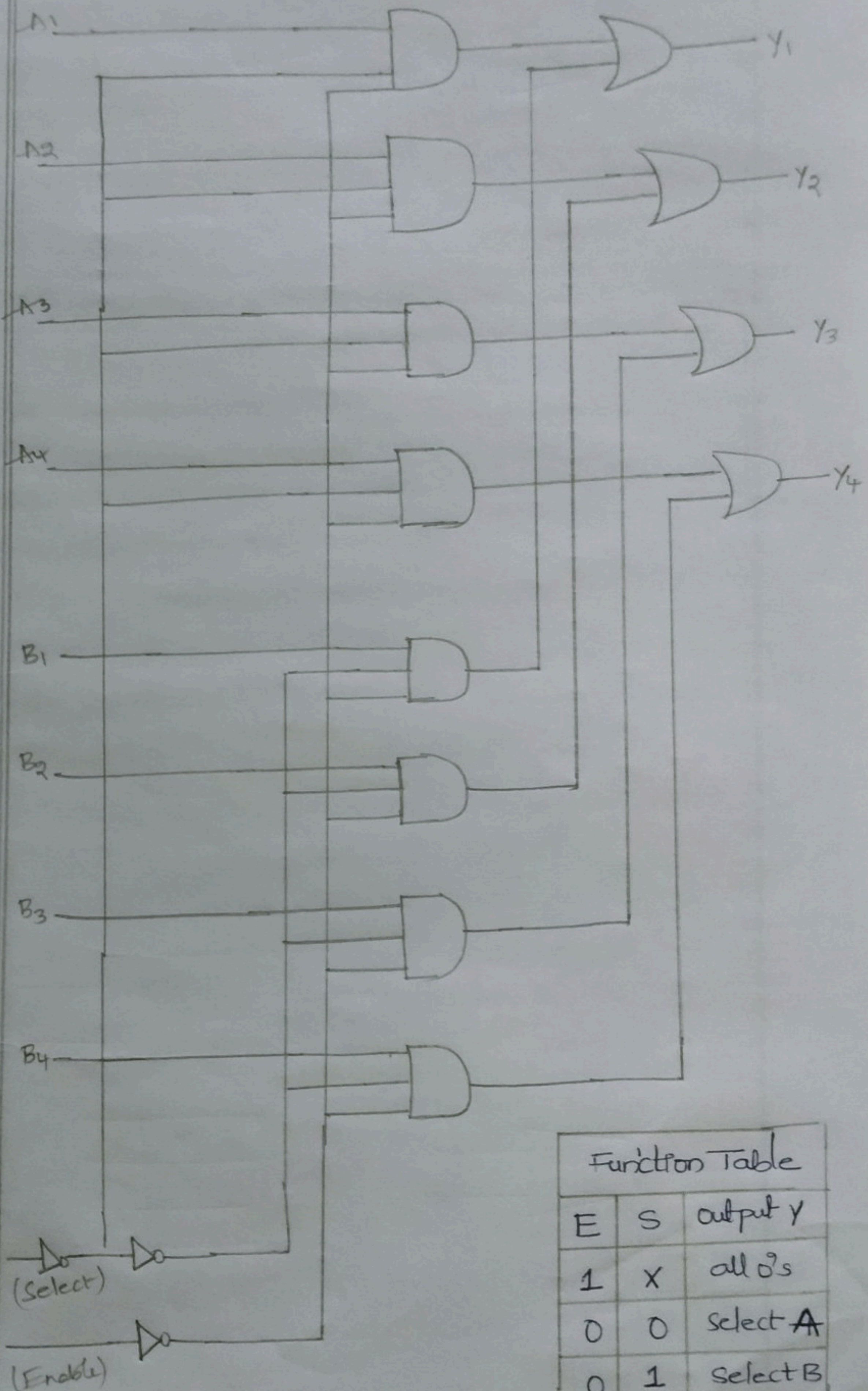
$Y = D_0\bar{S} \oplus D_1 S$

* Logic diagram



→ 2* 1 MUX diagram

(6) Explain 2 to 1 line multiplexers?



Function Table		
E	S	output Y
1	X	all 0's
0	0	Select A
0	1	Select B

→ Quadruple - 2 to - 1 line multiplexer