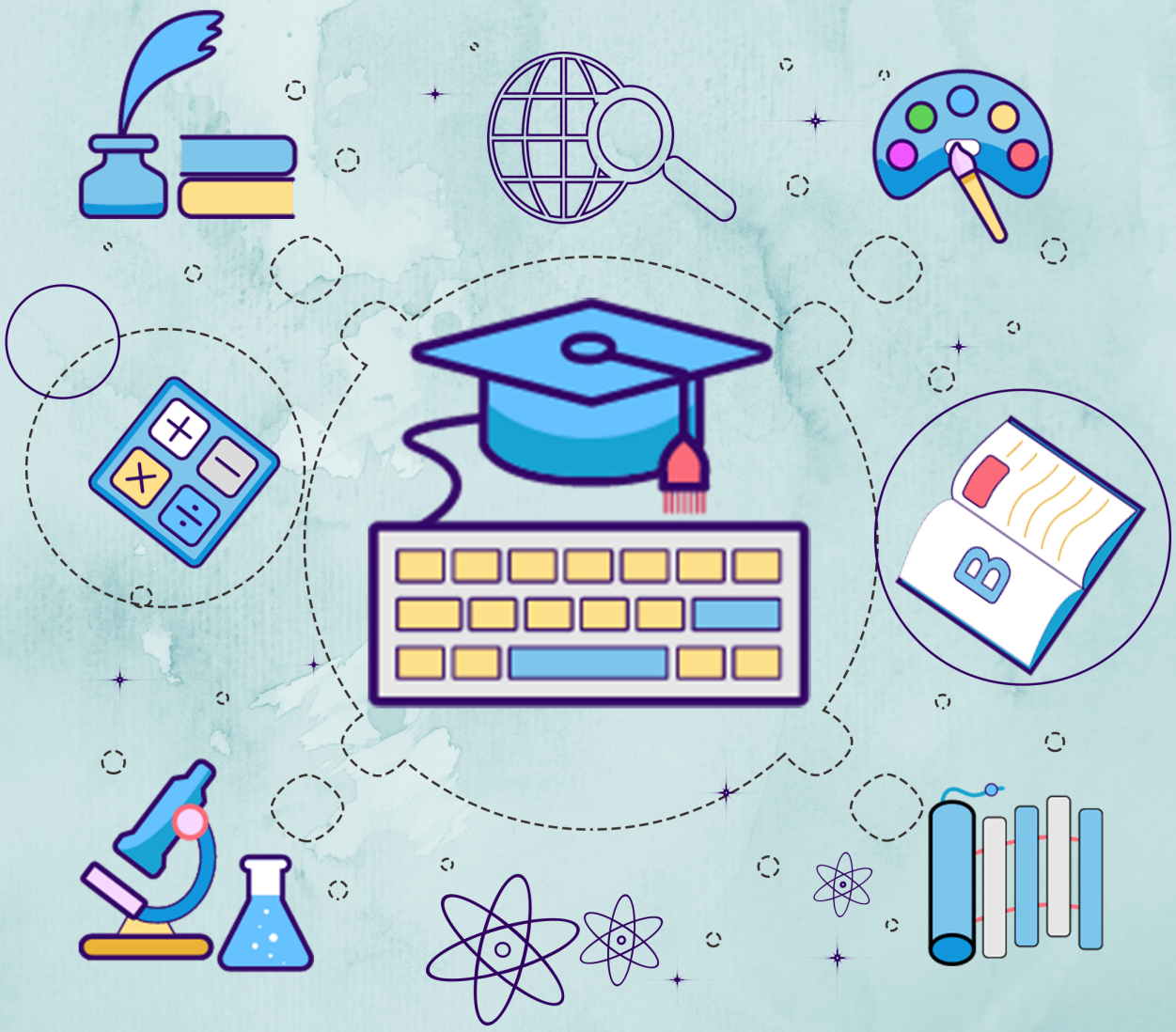
APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY

KeralaNotes



SYLLABUS | STUDY MATERIALS | TEXTBOOK PDF | SOLVED QUESTION PAPERS

www.keralanotes.com

KeralaNotes

KTU STUDY MATERIALS

LOGIC SYSTEM DESIGN

CST 203

Module 3

Related Link :

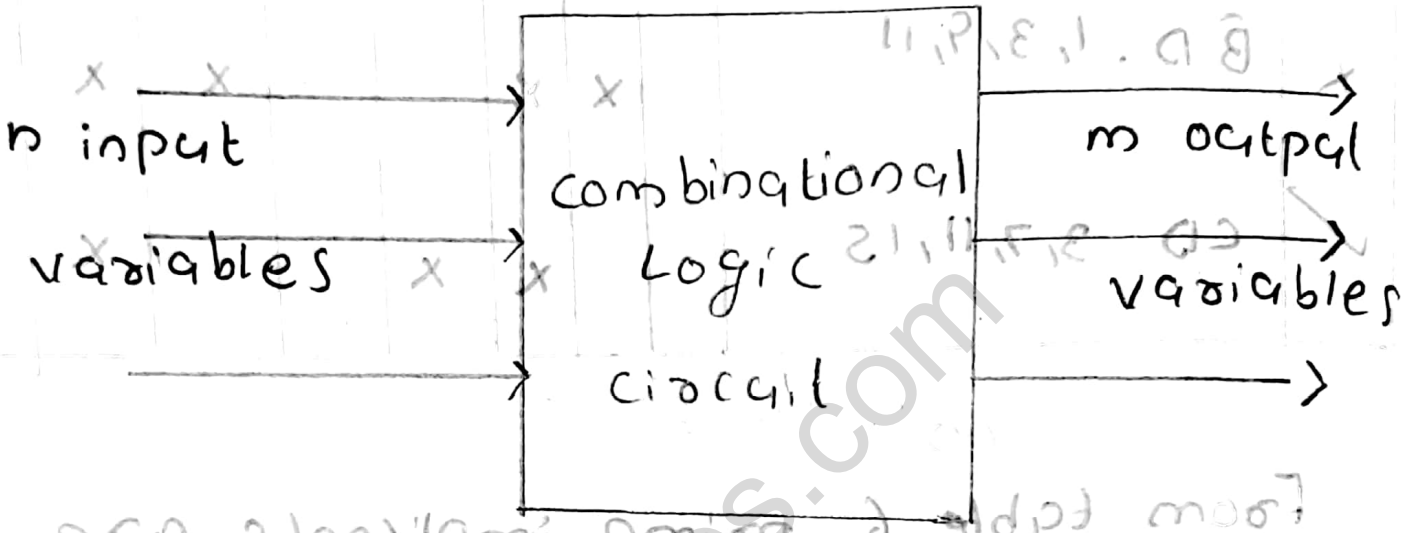
- KTU S3 STUDY MATERIALS
- KTU S3 NOTES
- KTU S3 SYLLABUS
- KTU S3 TEXTBOOK PDF
- KTU S3 PREVIOUS YEAR SOLVED QUESTION PAPER

www.keralanotes.com

KeralaNotes

Combinational Logic Circuit

Block Diagram



combinational logic circuit:

when Logic gates are connected together to produce a specified output for certain specific combination of input variables, with no storage involved

- output variables are depend on input variables
- consist of input variables, Logic gate and output variable
- Logic gate accept the signal from input variable and generate output variable

For More Study Materials : https://www.keralanotes.com/

KeralaNotes

Types of combinational logic

(i) Adder
- Half adder
- Full adder

(ii) Substractor
- Half substractor
- Full substractor

(iii) Multiplexer
MUX
- 2 : 1
- 4 : 1
- 8 : 1

(iv) Demultiplexer
DEMUX
- 1 : 2
- 1 : 4
- 1 : 8

(v) Decoder

(vi) Encoder

Design procedure

1. problem definition
2. Determination of input and output variables
3. Assigning letter symbol to input and output variable
4. Derivation of truth table
5. Obtain simplified Boolean Expression
6. obtain the logical design

keralanotes.com

For More Study Materials : https://www.keralanotes.com/

KeralaNotes

Examples:

1. Design a combinational logic circuit with three input variables that will produce a logical output when more than one input variables are logic 1

Ans:

Truth table

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

Boolean expression

A \ BC	$\bar{B}\bar{C}$ 00	$\bar{B}C$ 01	BC 11	$B\bar{C}$ 10
$\bar{A}$ 0	0	0	1	0
A 1	0	1	1	1

AB

AC

grouping only

8
4
2
1

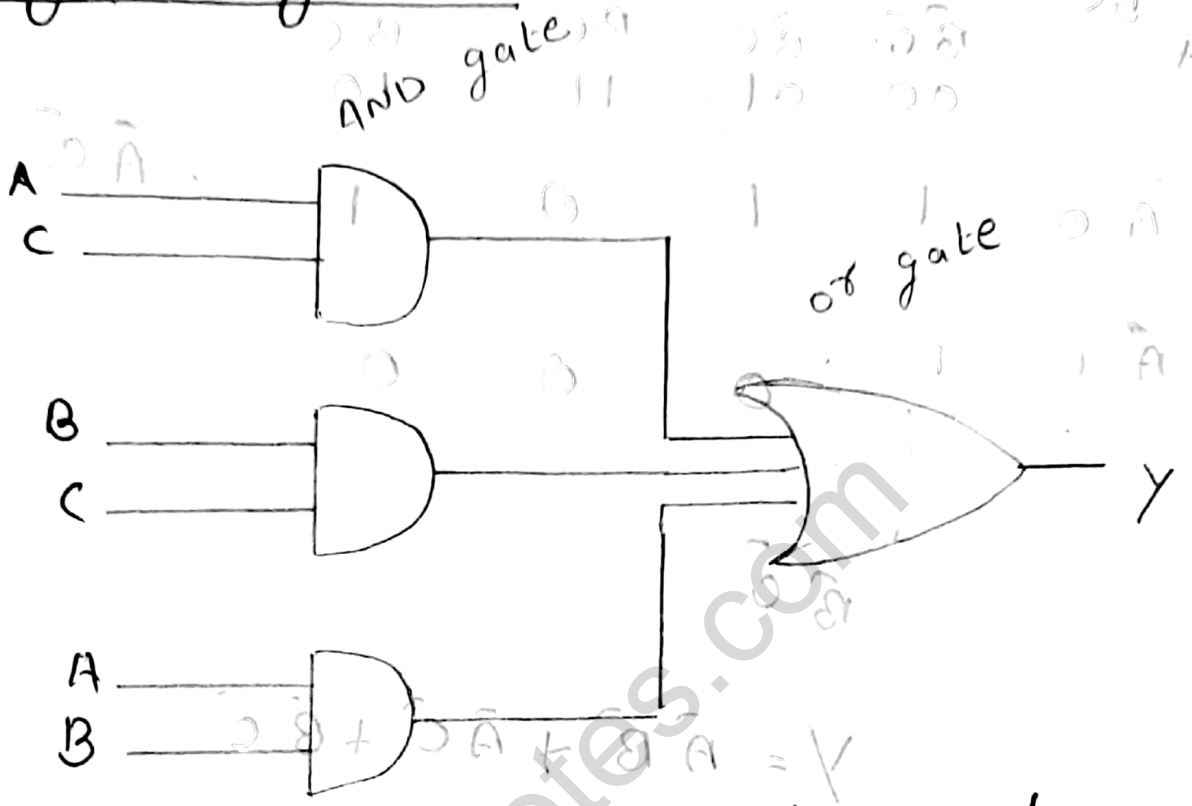
keralanotes.com

For More Study Materials : https://www.keralanotes.com/

KeralaNotes

$$Y = AC + BC + AB$$

Logic diagram



2. Majority gate is digital circuit whose input is equal to 0 if majority of the input are 1's. The output is 1 otherwise. Using TT find Boolean function and logic diagram.

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

For More Study Materials : https://www.keralanotes.com/

KeralaNotes

Boolean Expression

A \ BC	$\bar{B}\bar{C}$ 00	$\bar{B}C$ 01	BC 11	$B\bar{C}$ 10
$\bar{A}$ 0	1	1	0	1
A 1	1	0	0	0

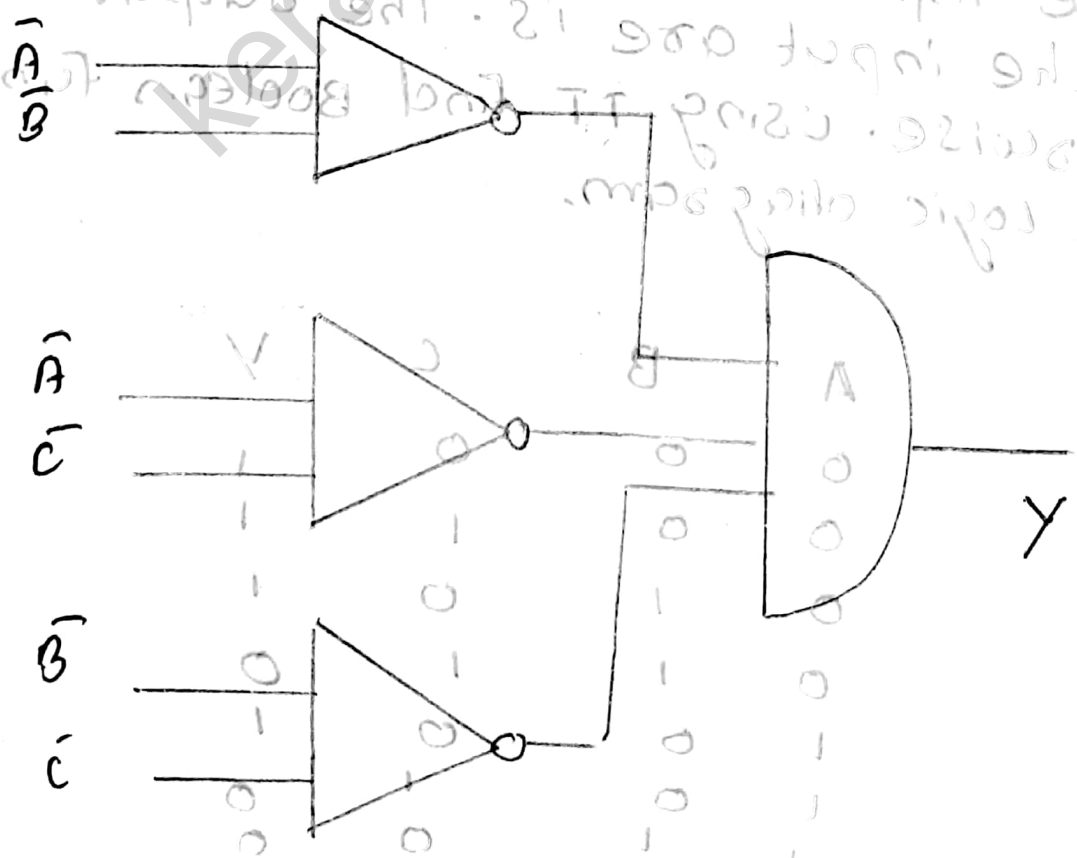
$\bar{A}\bar{B}$

$\bar{A}\bar{C}$

$\bar{B}\bar{C}$

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

Logic diagram



keralanotes.com

For More Study Materials : https://www.keralanotes.com/

Design of Adders

$0 + 0 = 0$

$0 + 1 = 1$

$1 + 0 = 1$

$1 + 1 = \underset{\text{carry}}{1}\ \underset{\text{sum}}{0} = 10$

- Simple addition consist of four possible elementary operations
- ~~First 3 operator~~
- First 3 operations produce sum whose length is one digit
- Last operation performed sum in two digits. Higher significant result is called carry and lower significant bit is called sum.

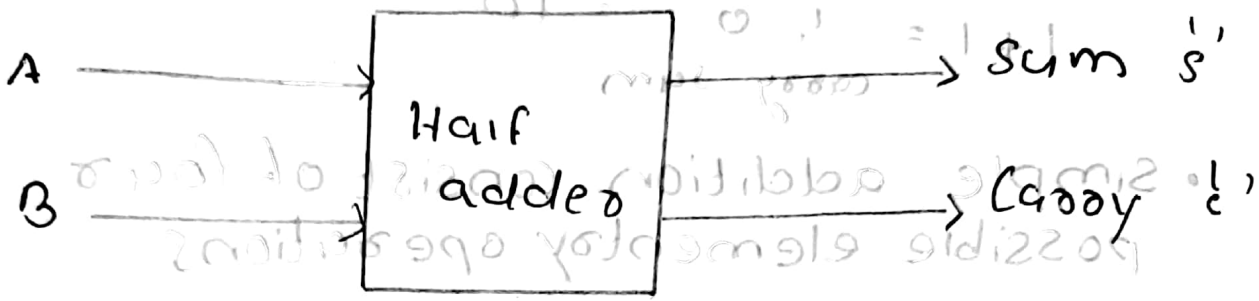
Half Adder

- Half adder is a combinational logic circuit with two inputs and outputs.
- The half adder circuit is designed to add two single bit binary number A and B
- It is the basic building block for addition of two single bit numbers

For More Study Materials : https://www.keralanotes.com/

KeralaNotes

This circuit has two outputs carry and sum



Truth table

Inputs		Outputs	
A	B	Carry	sum
0	0 [0+0]	0	0
0	1 [0+1]	0	1
1	0 [1+0]	0	1
1	1 [1+1]	1	0

Boolean Expression

keralanotes.com

For More Study Materials : https://www.keralanotes.com/

KeralaNotes

Carry

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

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

$Y = AB$

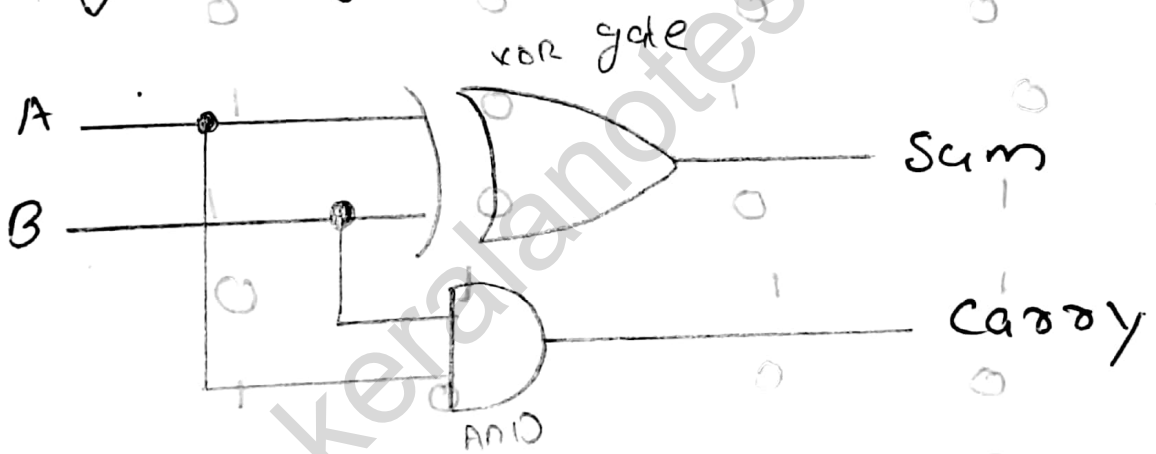
and

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

$= A \oplus B$

XOR

Logic diagram

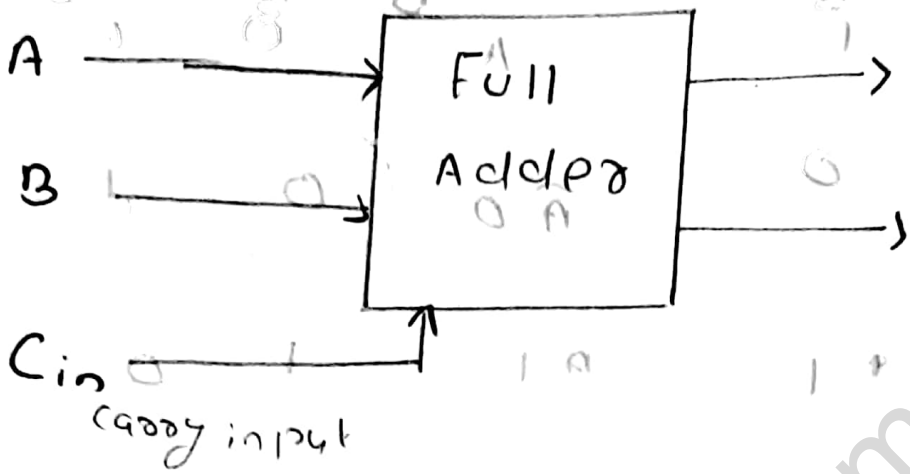


Full Adder

- Full Adder is developed to overcome the drawback of Half Adder
- It can add two one bit numbers A and B and C
- Full adder is a three input and two output combinational circuit

For More Study Materials : https://www.keralanotes.com/

KeralaNotes



Truth table

Inputs			Output	
A	B	Cin	Carry	Sum
0	0	0 [0+0+0]	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

keralanotes.com

For More Study Materials : https://www.keralanotes.com/

KeralaNotes

Carry

A \ $B C_{in}$	$\bar{B}\bar{C}$ 00	$\bar{B}C$ 01	BC 11	$B\bar{C}$ 10
$\bar{A}$ 0	0	0	1	0
A 1	0	1	1	1

Sum

A \ $B C_{in}$	$\bar{B}\bar{C}$ 00	$\bar{B}C$ 01	BC 11	$B\bar{C}$ 10
$\bar{A}$ 0	0	1	0	1
A 1	1	0	1	0

$C_{out} = AB + AC_{in} + BC_{in}$

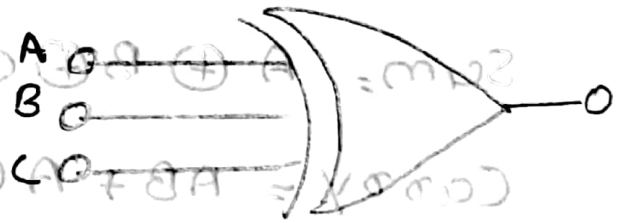
AND

$Sum = \bar{A}\bar{B}C_{in} + \bar{A}B\bar{C}_{in} + A\bar{B}\bar{C}_{in} + ABC_{in}$

XOR

Reference : 3 input XOR gate

Inputs			outputs
W	X	Y	$Q = A \oplus B \oplus C$
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1



we get bot truth table same

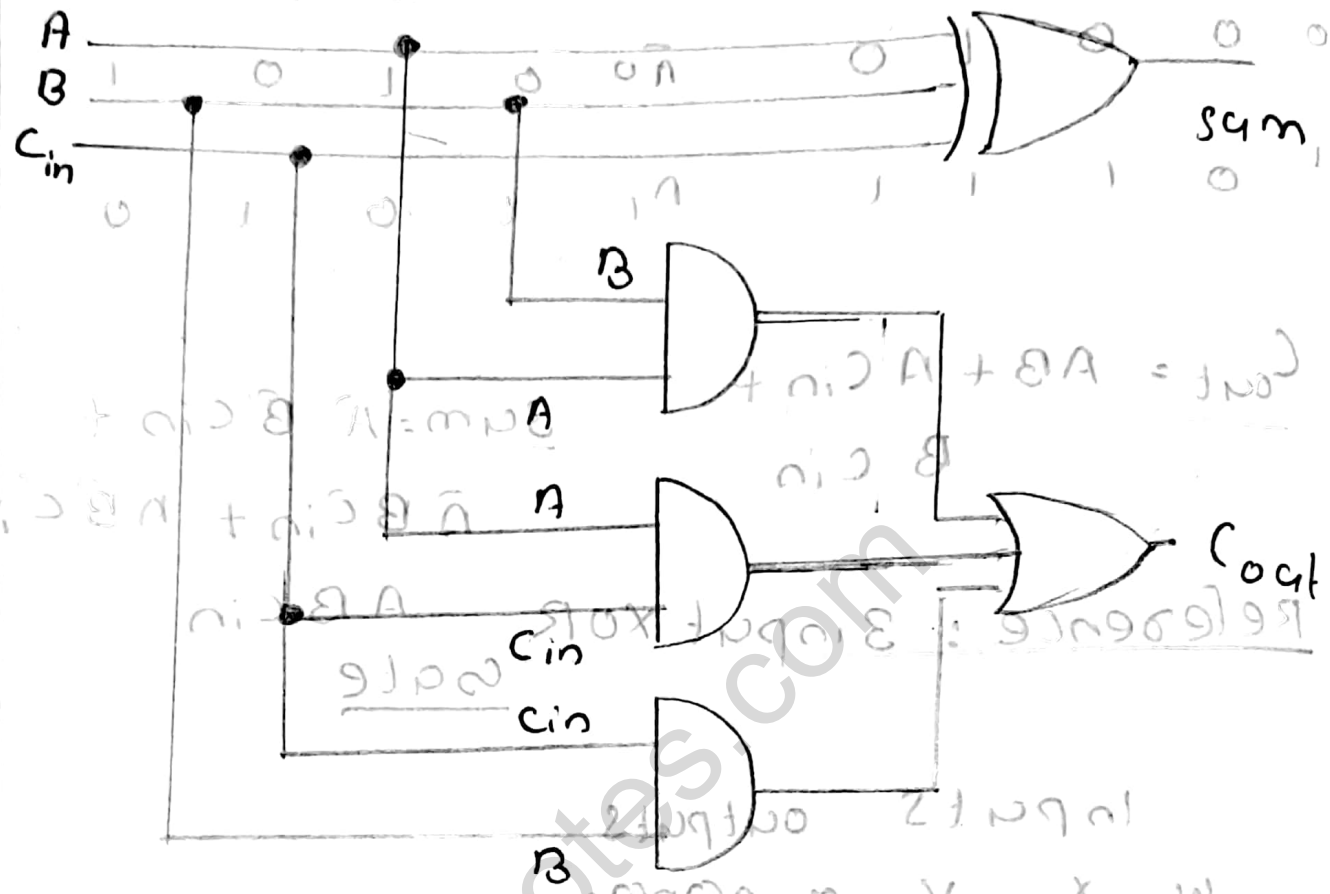
$$\boxed{Sum = A \oplus B \oplus C_{in}}$$

keralanotes.com

For More Study Materials : https://www.keralanotes.com/

KeralaNotes

Logic Diagram



Sum = $A \oplus B \oplus C_{in}$

Carry = $AB + AC_{in} + BC_{in}$

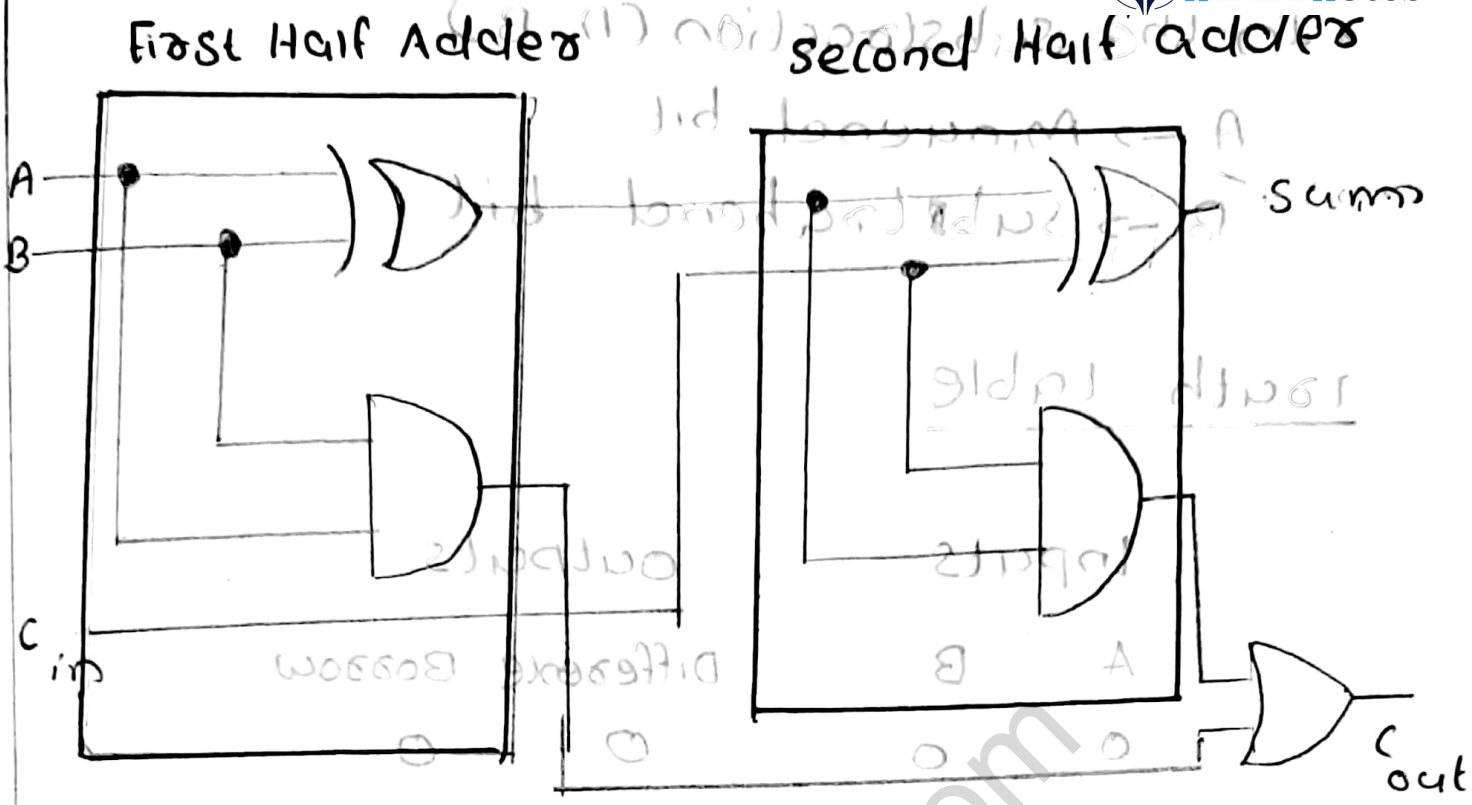
~~Implementation of Full Adder using~~

Implementation of Full Adder

Circuit using two Half Adders

keralanotes.com

KeralaNotes



Design of substractor

→ 0 - 0 = 0

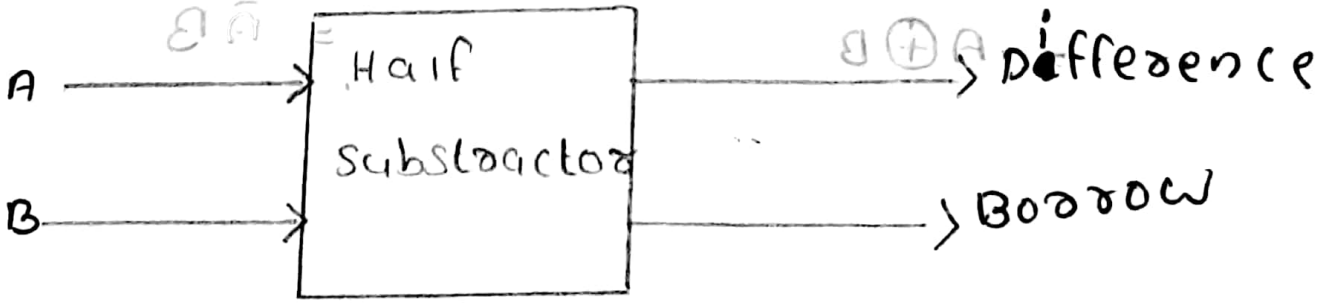
→ 0 - 1 = 1 with borrow 1

→ 1 - 0 = 1

→ 1 - 1 = 0

Half substractor

- Half substractor is a combinational circuit with two input and two outputs (difference and borrow)



For More Study Materials : https://www.keralanotes.com/

KeralaNotes

- In the substraction (A - B)

  A → Minuend bit

  B → subtrahend bit

Truth table

Inputs		Outputs	
A	B	Difference	Borrow
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

Boolean expression

For Difference

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

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

$= A \oplus B$

For Borrow

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

$\bar{A}B$

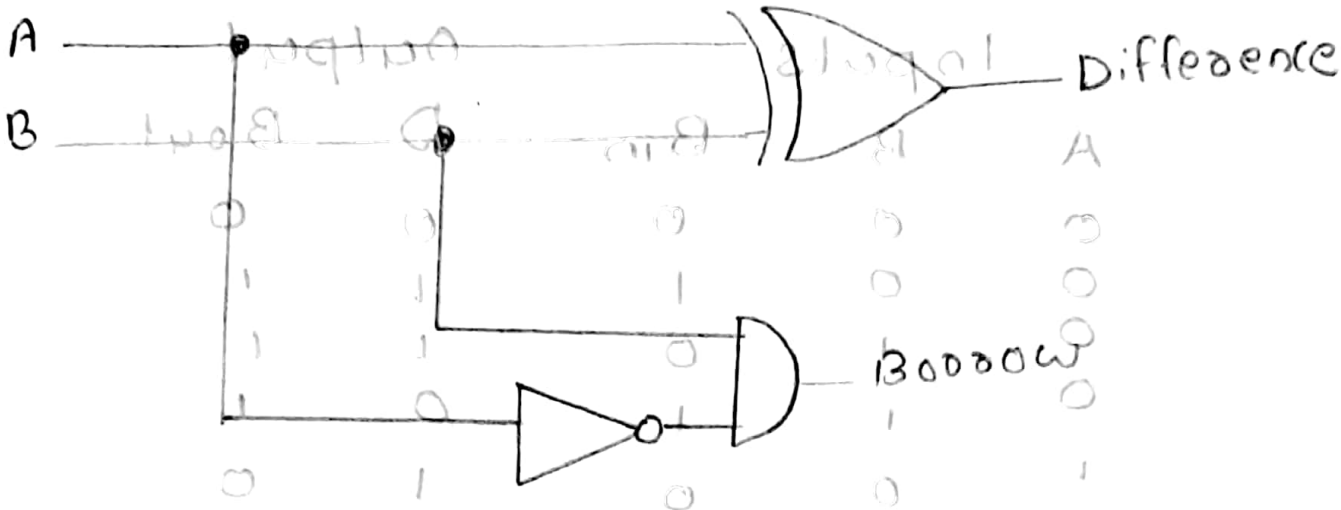
$= \bar{A}B$

keralanotes.com

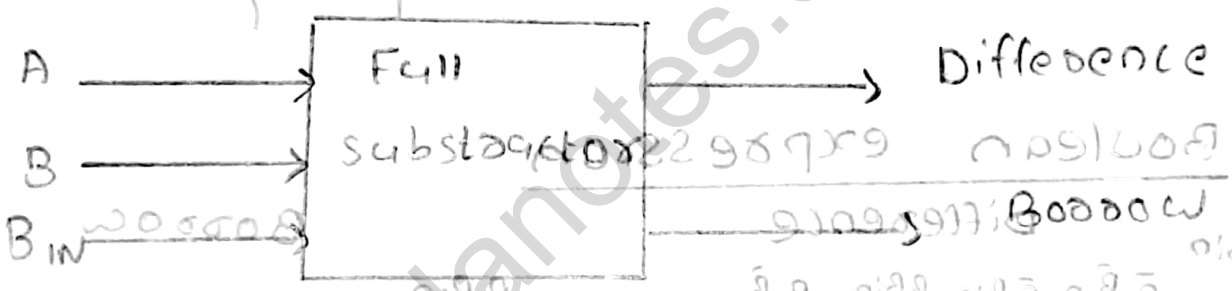
For More Study Materials : https://www.keralanotes.com/

KeralaNotes

Logic Diagram



Full substractor



- Disadvantages of half substractor is overcome by full substractor
- 3 input A, B, Bin and
  2 output D and Bout
- Input: A → Minuend
  B → Subtrahend
  B_{in} → previous Borrow

  output D → Difference
  B_{out} → output borrow

keralanotes.com

For More Study Materials : https://www.keralanotes.com/

KeralaNotes

Truth table

0 - 1 = 1 Bout = 1

1 - 0 = 1 = Bin

Inputs			Output	
A	B	Bin	D	Bout
0	0 (0-0-0)	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	1
1	0	0	1	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

Boolean expression

Difference

A \ B Bin	$\bar{B}\bar{B}in$ 00	$\bar{B}Bin$ 01	$BBin$ 11	$B\bar{B}in$ 10
$\bar{A}$ 0	0	1		1
A 1	1		1	

Borrow

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

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

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

$AB\,Bin +$

$\bar{A}B\bar{B}in$

XOR gate

Referento

$= \{A \oplus B \oplus Bin\}$

$= \bar{A}Bin + \bar{A}B + BBin$

$\Rightarrow \bar{A}\bar{B}Bin + \bar{A}B\bar{B}in +$

$A\bar{B}\bar{B}in + ABBin$

For More Study Materials : https://www.keralanotes.com/

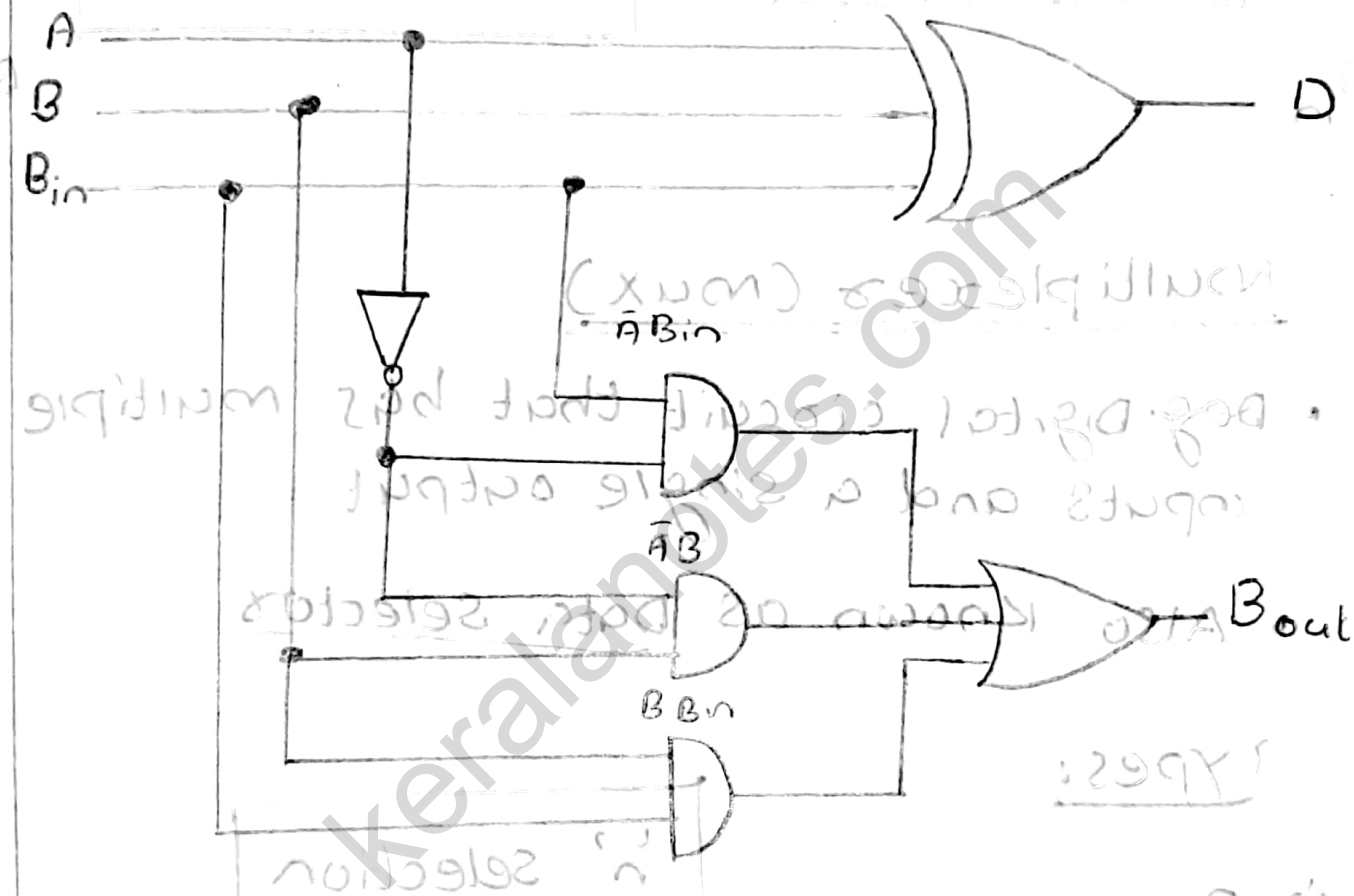
Difference

$$B_{in} \oplus (A \oplus B)$$

Borrow

$$B_{out} = \bar{A} B_{in} + \bar{A} B + B B_{in}$$

Logic Diagram

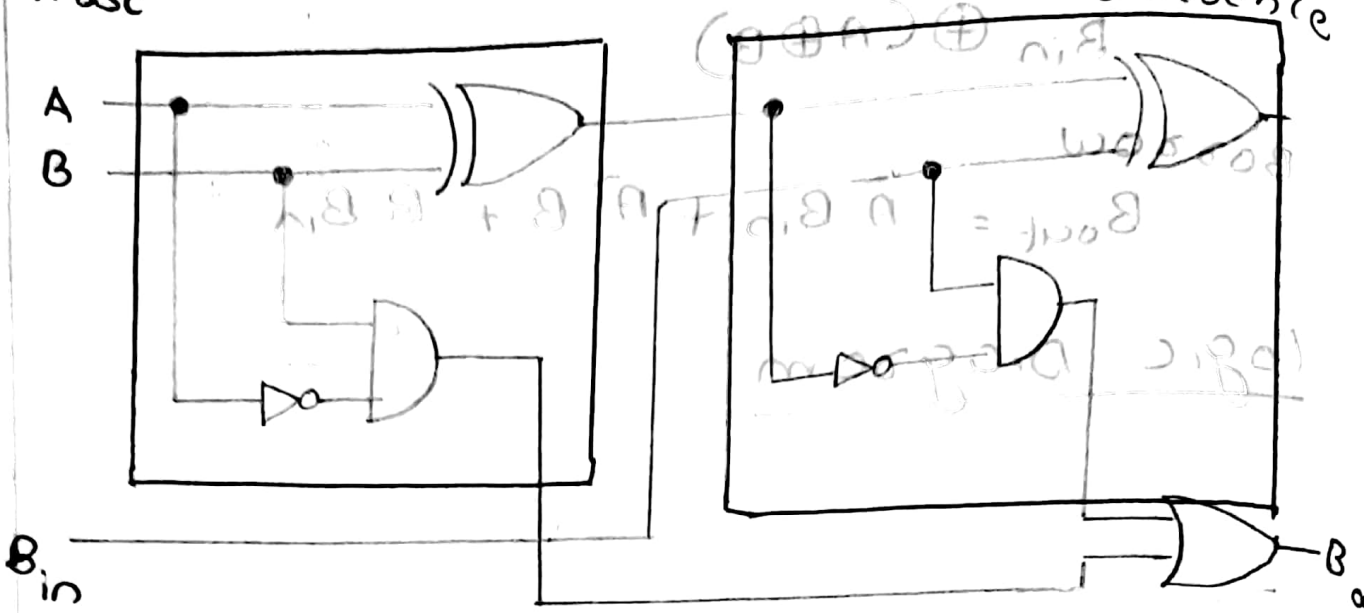


Implementation of full substractor circuit using two Half substractor

KeralaNotes
keralanotes.com

KeralaNotes

First - Half subtractor second half subtractor
Difference



Multiplexer (MUX)

- ~~Def~~ Digital circuit that has multiple inputs and a single output
- Also known as Data selector

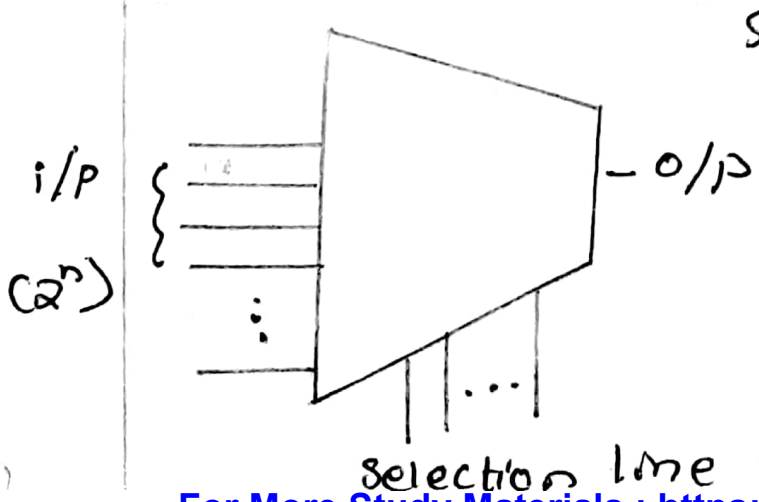
Types:

(i) 2:1 MUX
(ii) 4:1 MUX
(iii) 8:1 MUX

'n' selection
2^n input

Selection Line 3,
inp = 2^3 = 8:

S.L = 4, 2^4 = 16:1

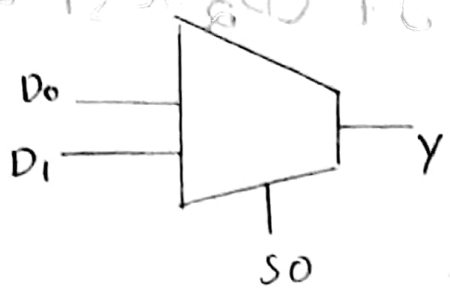


keralanotes.com

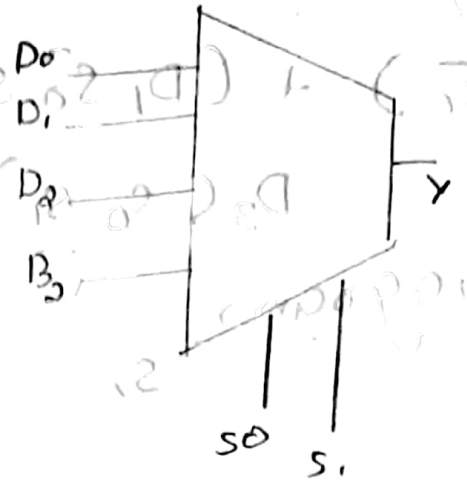
For More Study Materials : https://www.keralanotes.com/

KeralaNotes

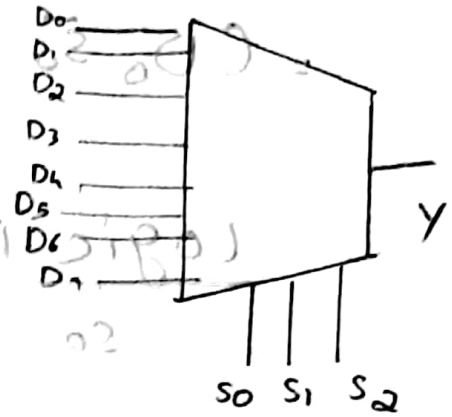
2:1 MUX



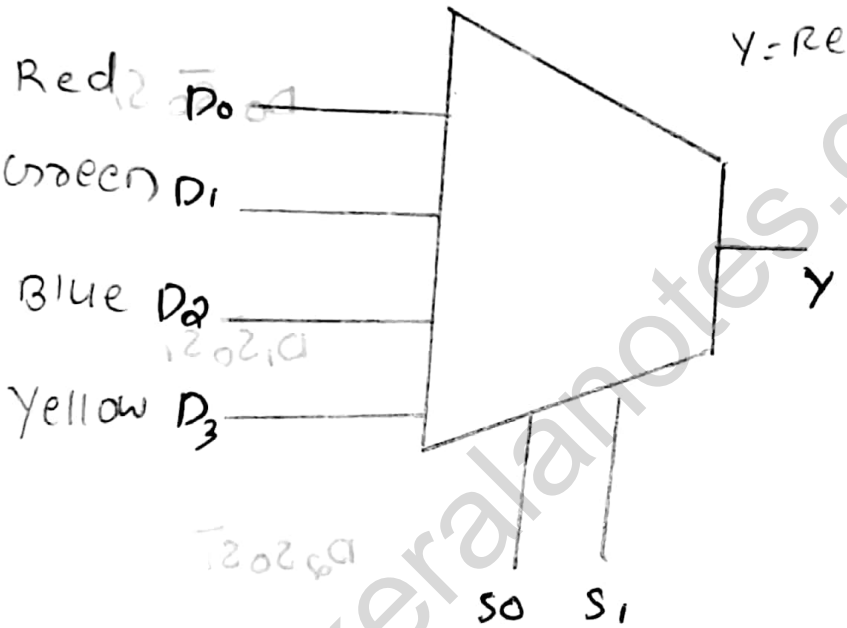
4:1 MUX



8:1 MUX



(i) 4:1 MUX or 4 TO 1 MUX



Y = Red [0 0] - off, OFF

00 → D0
01 → D1
10 → D2
11 → D3

Truth table:

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

Boolean Expression

SOP

$$Y = D_0 \cdot (\bar{S_0} \cdot \bar{S_1}) + D_1 \cdot (\bar{S_0} \cdot S_1) + D_2 \cdot (S_0 \cdot \bar{S_1}) + D_3 \cdot (S_0 S_1)$$

keralanotes.com

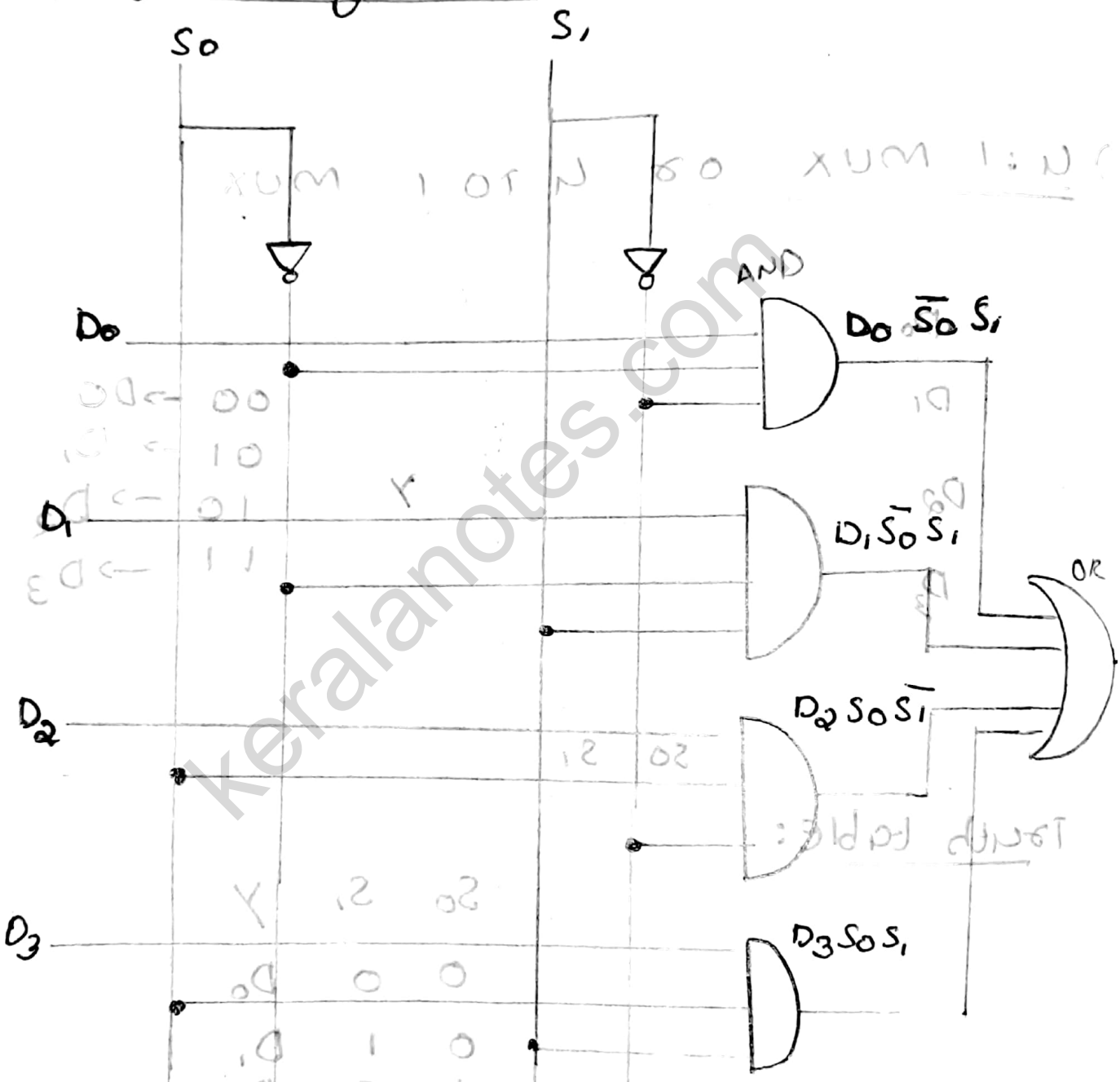
For More Study Materials : https://www.keralanotes.com/

KeralaNotes

Logic Diagram

$$= (D_0 \bar{S_0} \bar{S_1}) + (D_1 \bar{S_0} S_1) + (D_2 S_0 \bar{S_1}) + D_3 (S_0 S_1)$$

Logic Diagram

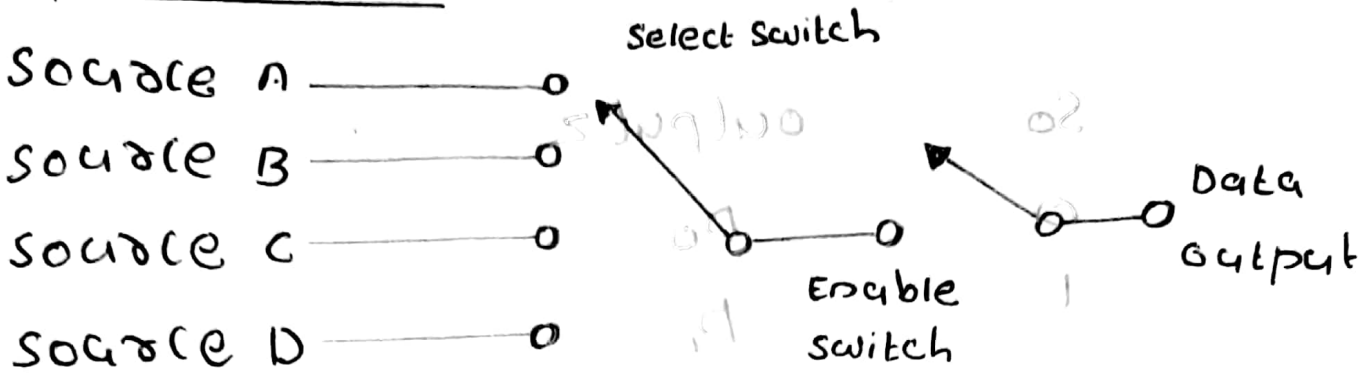


keralanotes.com

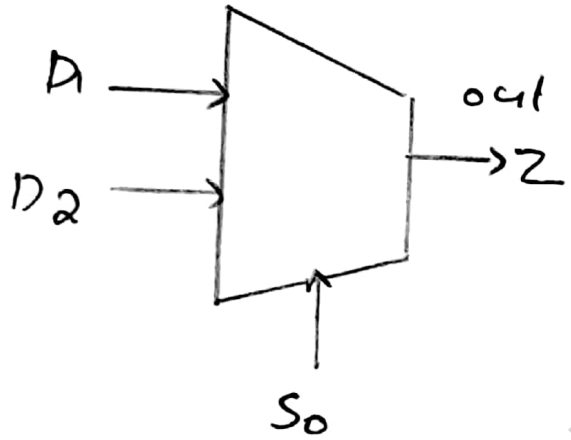
For More Study Materials : https://www.keralanotes.com/

KeralaNotes

2:1 MUX

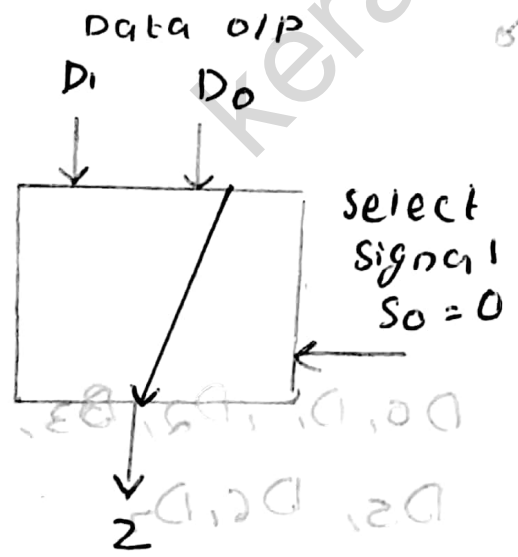


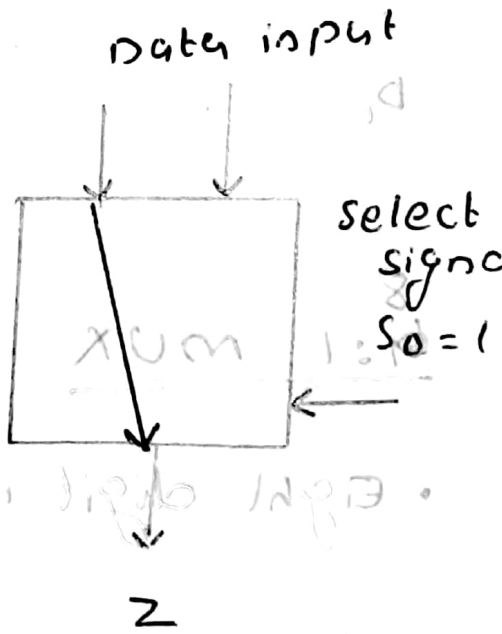
(ii) 2:1 MUX



- IF selection line $S_0 = 0$ then $o/p = D_0$
- If selection Line $S_0 = 1$ then $o/p = D_1$

operation





keralanotes.com

For More Study Materials : https://www.keralanotes.com/

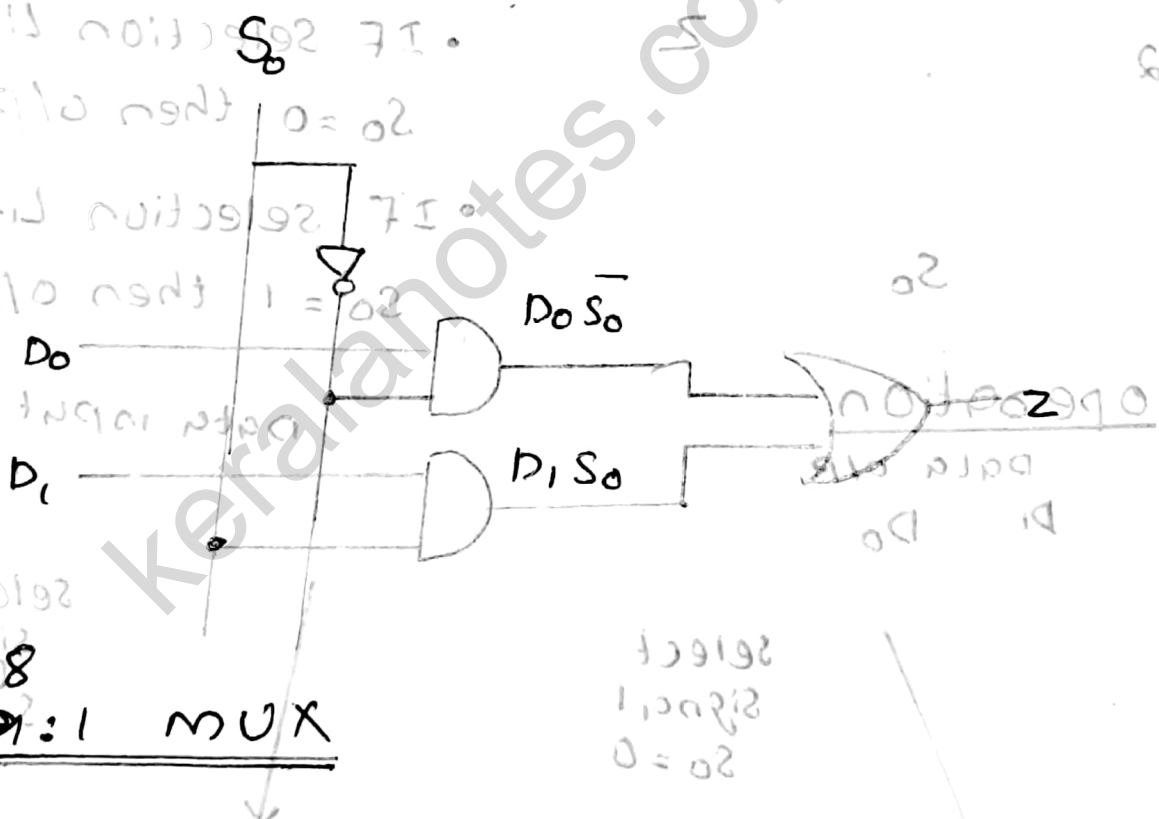
KeralaNotes

Truth table

S_0	output Z
0	D_0
1	D_1

$Z = D_0 . \overline{S_0} + D_1 . S_0$ => Boolean Express

logic Diagram



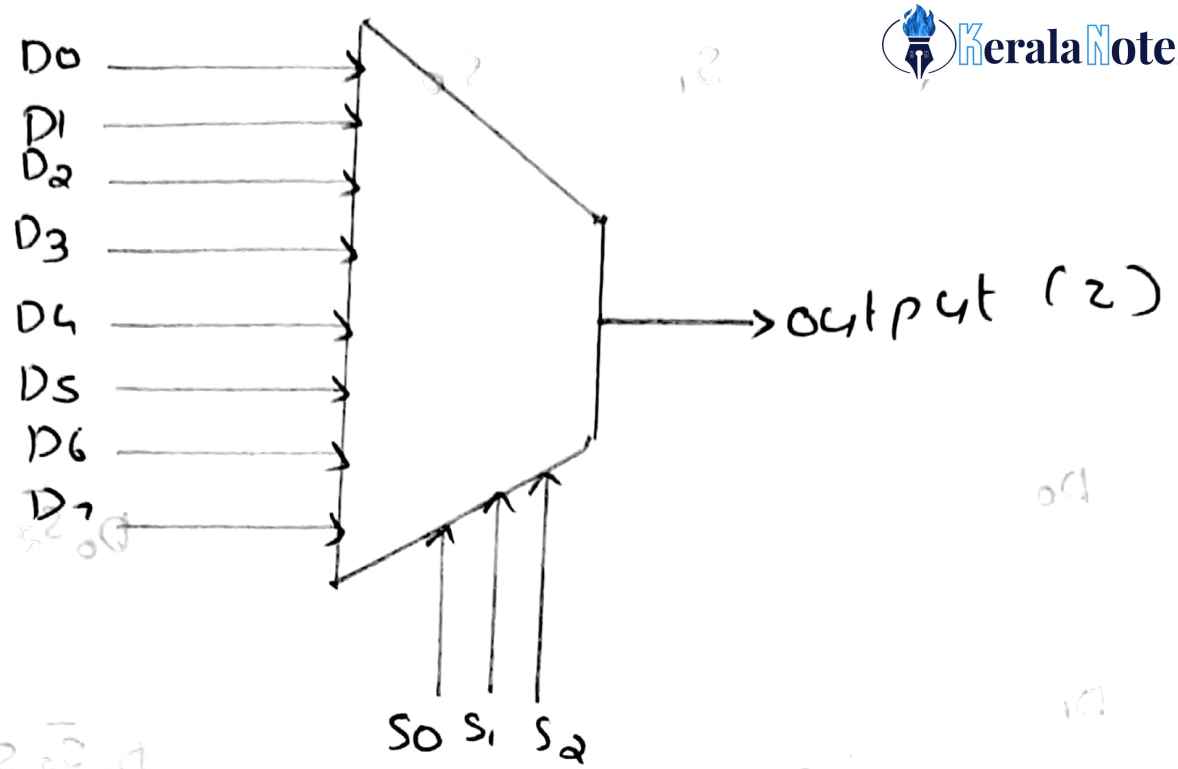
keralanotes.com

(ii) 8 : 1 MUX

- Eight digit inputs, $D_0, D_1, D_2, D_3, D_4, D_5, D_6, D_7$
- Three select lines S_0, S_1, S_2
- one output Z

For More Study Materials : https://www.keralanotes.com/

KeralaNotes



Truth table

S_2	S_1	S_0	output Z
0	0	0	D_0
0	0	1	D_1
0	1	0	D_2
0	1	1	D_3
1	0	0	D_4
1	0	1	D_5
1	1	0	D_6
1	1	1	D_7

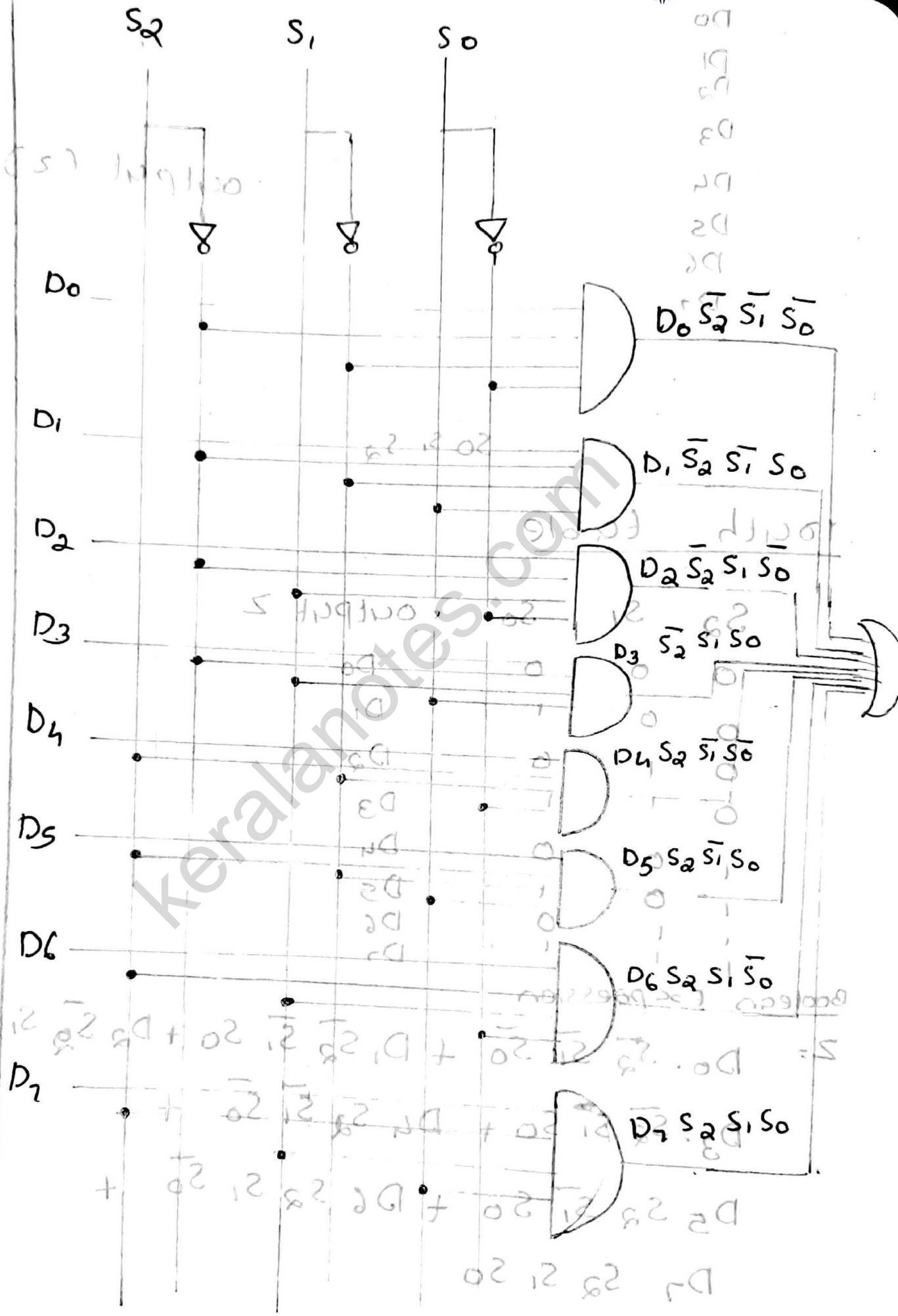
keralanotes.com

Boolean Expression

$$Z = D_0 . \bar{S_2}\,\bar{S_1}\,\bar{S_0} + D_1 \bar{S_2}\,\bar{S_1}\,S_0 + D_2 \bar{S_2}\,S_1\,\bar{S_0} + D_3 . \bar{S_2}\,S_1\,S_0 + D_4\,S_2\,\bar{S_1}\,\bar{S_0} + D_5\,S_2\,\bar{S_1}\,S_0 + D_6\,S_2\,S_1\,\bar{S_0} + D_7\,S_2\,S_1\,S_0$$

For More Study Materials : https://www.keralanotes.com/

KeralaNotes



keralanotes.com

For More Study Materials : https://www.keralanotes.com/

KeralaNotes

Demux / Demultiplexer

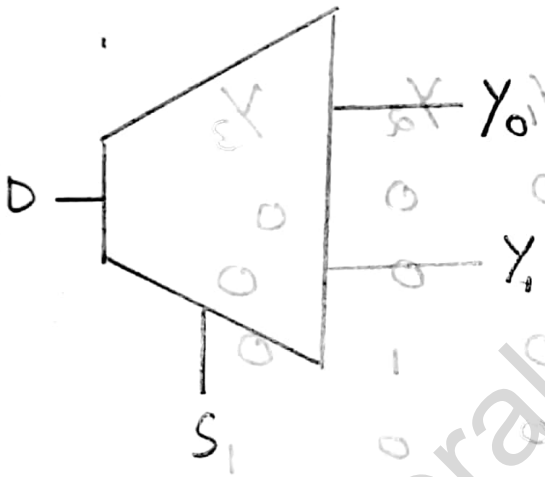
It is a circuit that recieve information on a single line and transmit this information on one of 2^n possible o/p lines

The selection of specific o/p line is controlled by values of selection line

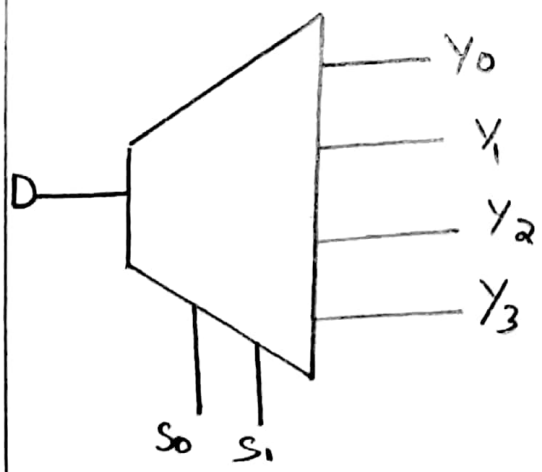
- Also known as Data Distributer

1 -> i/p
n -> S.L
2^n -> o/p

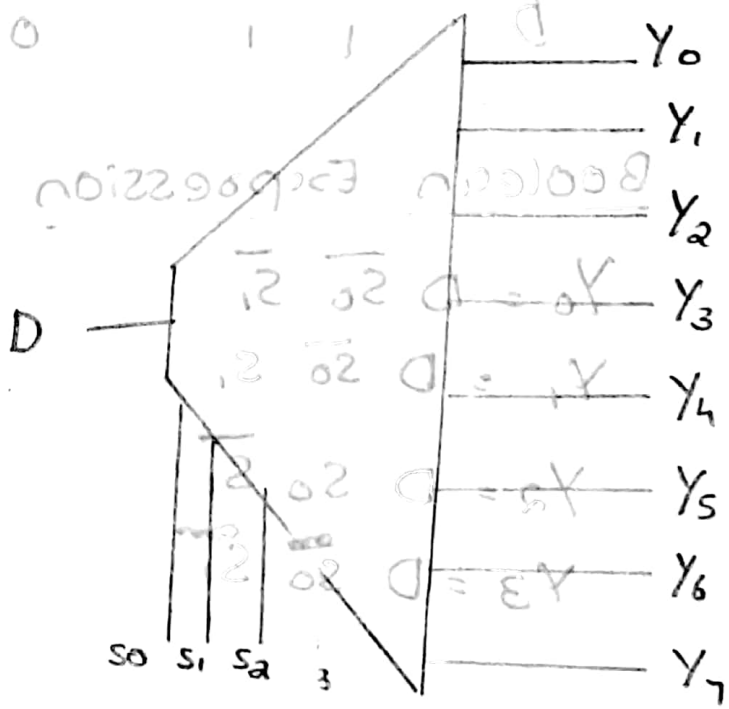
1:2



1:4



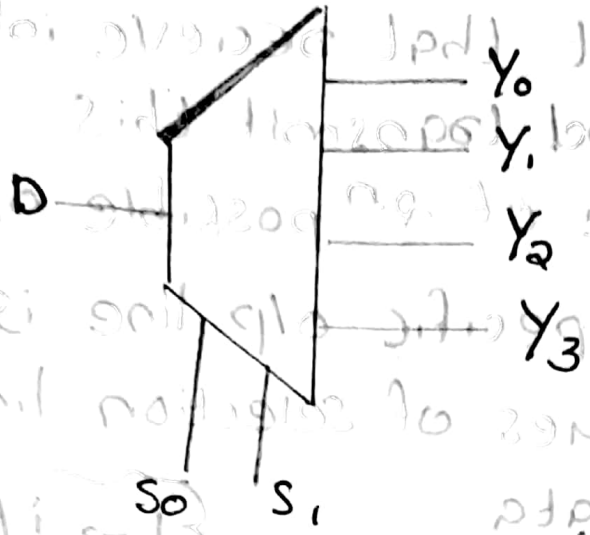
1:8



keralanotes.com

For More Study Materials : https://www.keralanotes.com/

1:4 DeMux



when 0 0 output will be Y_0 ($S_0 = 1$) other ($=0$)

Truth table

I/P	S_0	S_1	Y_0	Y_1	Y_2	Y_3
D	0	0	1	0	0	0
D	0	1	0	1	0	0
D	1	0	0	0	1	0
D	1	1	0	0	0	1

Boolean Expression

$$Y_0 = D\,\overline{S_0}\,\overline{S_1}$$

$$Y_1 = D\,\overline{S_0}\,S_1$$

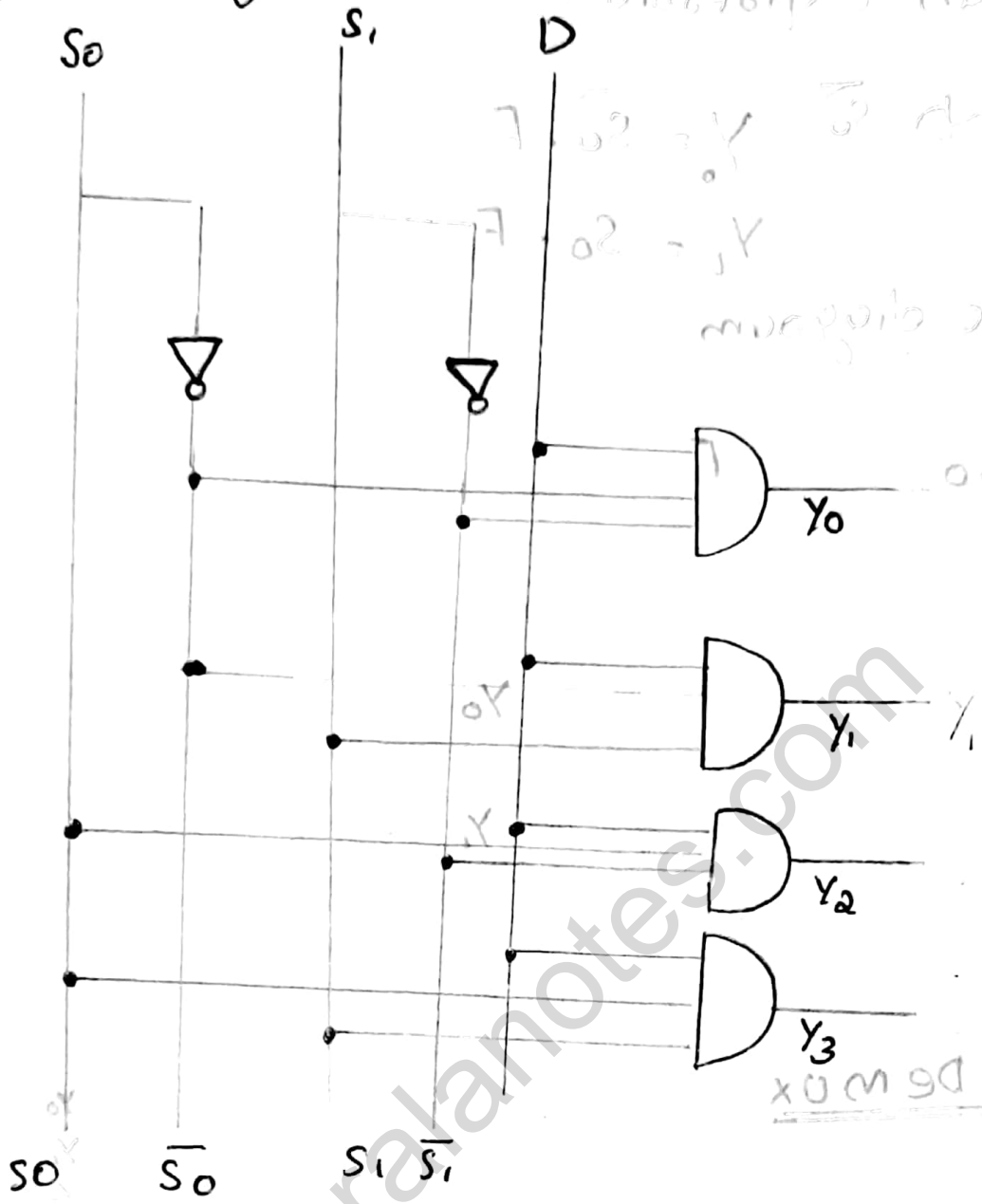
$$Y_2 = D\,S_0\,\overline{S_1}$$

$$Y_3 = D\,S_0\,S_1$$

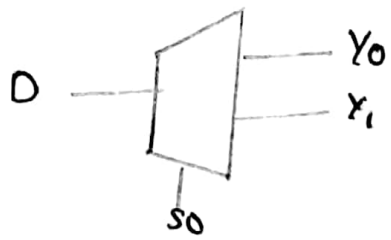
For More Study Materials : https://www.keralanotes.com/

KeralaNotes

Logical Diagram



1:2 De MUX



Truth table

S_0	Y_0	Y_1
0	F	0
1	0	F

keralanotes.com

For More Study Materials : https://www.keralanotes.com/

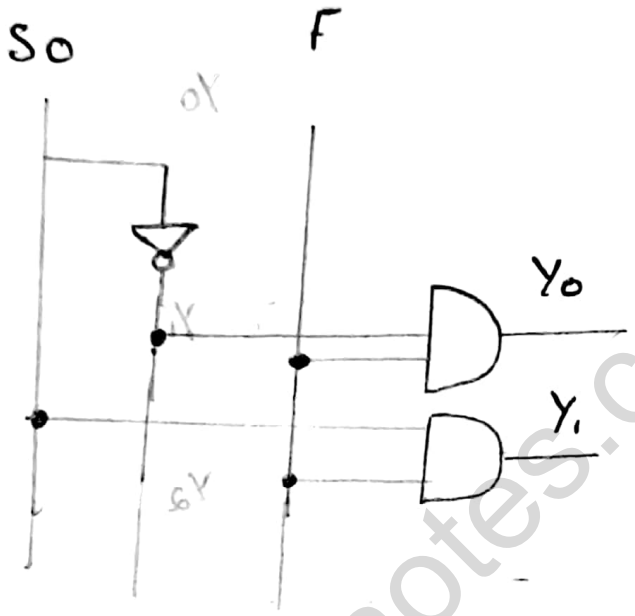
KeralaNotes

Boolean Expression

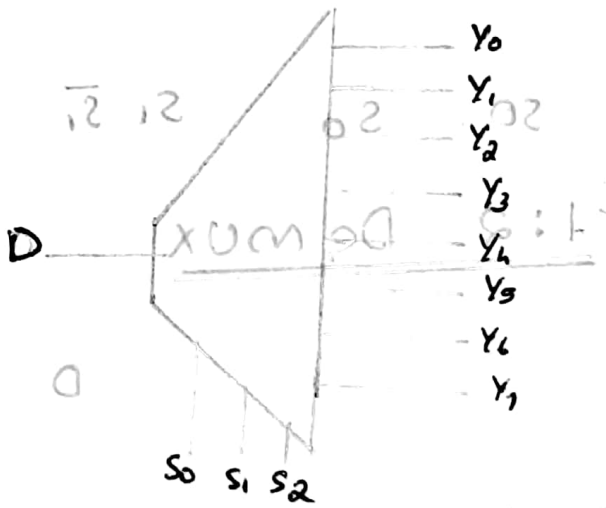
$\cancel{Y_0 = D\,\overline{S_0}}$ $\quad Y_0 = \overline{S_0} \cdot F$

$Y_1 = S_0 \cdot F$

Logic diagram



1:8 Demux



Truth table

S_0	S_1	S_2	
0	0	0	Y_0
0	0	1	Y_1
0	1	0	Y_2
0	1	1	Y_3
1	0	0	Y_4
1	0	1	Y_5
1	1	0	Y_6
1	1	1	Y_7

keralanotes.com

For More Study Materials : https://www.keralanotes.com/

KeralaNotes

Boolean expression

$Y_0 = D\,\overline{S_0}\,\overline{S_1}\,\overline{S_2}$

$Y_1 = D\,\overline{S_0}\,\overline{S_1}\,S_2$

$Y_2 = D\,\overline{S_0}\,S_1\,\overline{S_2}$

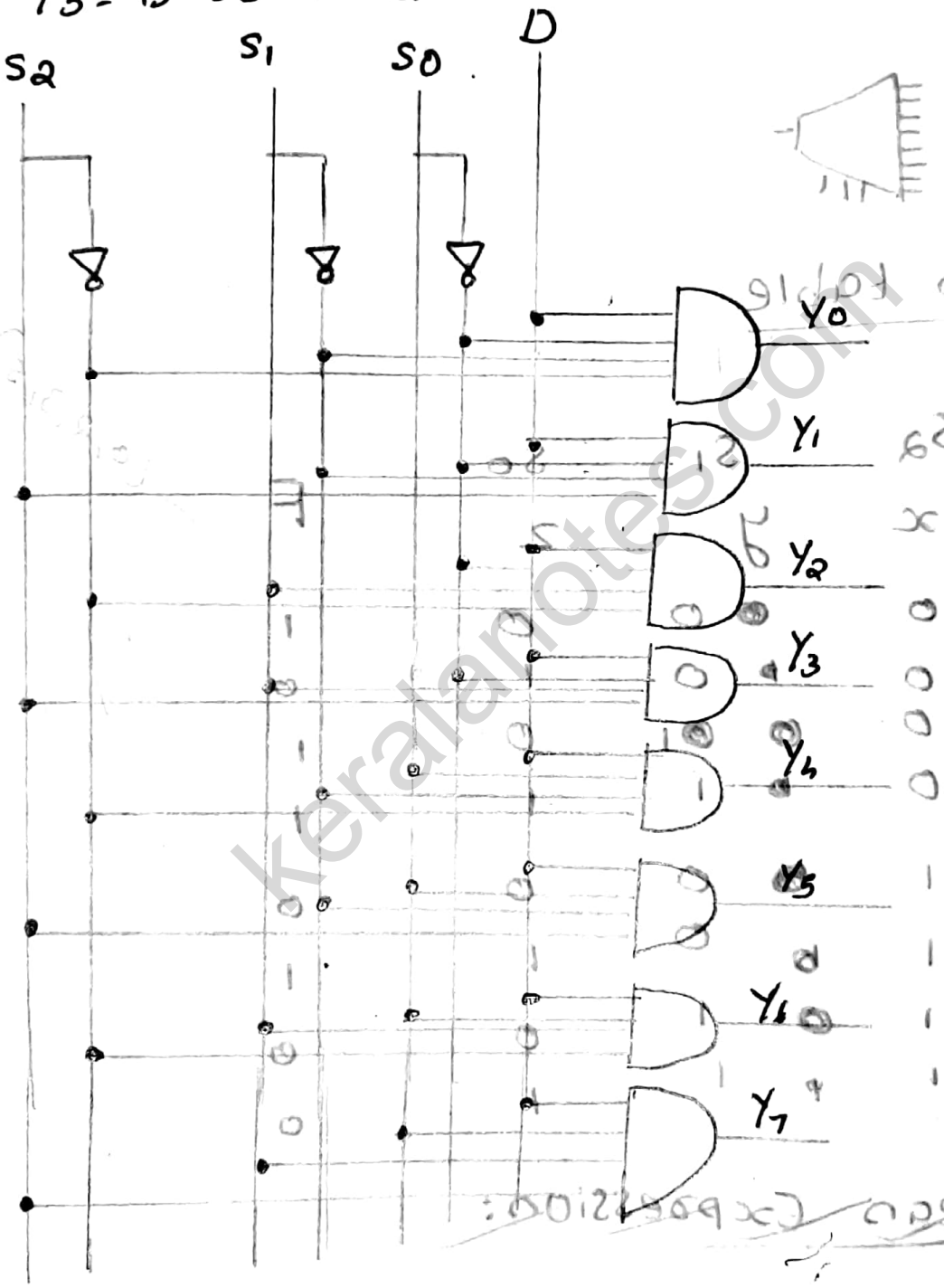
$Y_3 = D\,\overline{S_0}\,S_1\,S_2$

$Y_4 = D\,S_0\,\overline{S_1}\,\overline{S_2}\,D$

$Y_5 = S_0\,\overline{S_1}\,S_2\,D$

$Y_6 = S_0\,S_1\,\overline{S_2}\,D$

$Y_7 = S_0\,S_1\,S_2\,D$



keralanotes.com

For More Study Materials : https://www.keralanotes.com/

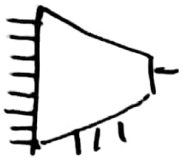
KeralaNotes

problems

1. Implement the following function using 8-to-1 MUX

$$F(x, y, z) = \Sigma m(0, 2, 3, 5)$$

ans:



Truth table

S_2	S_1	S_0	F	
x	y	z		
0	0	0	1	D_0
0	0	1	0	D_1
0	1	0	1	D_2
0	1	1	1	D_3
1	0	0	0	D_4
1	0	1	1	D_5
1	1	0	0	D_6
1	1	1	0	D_7

(0, 2, 3, 5)

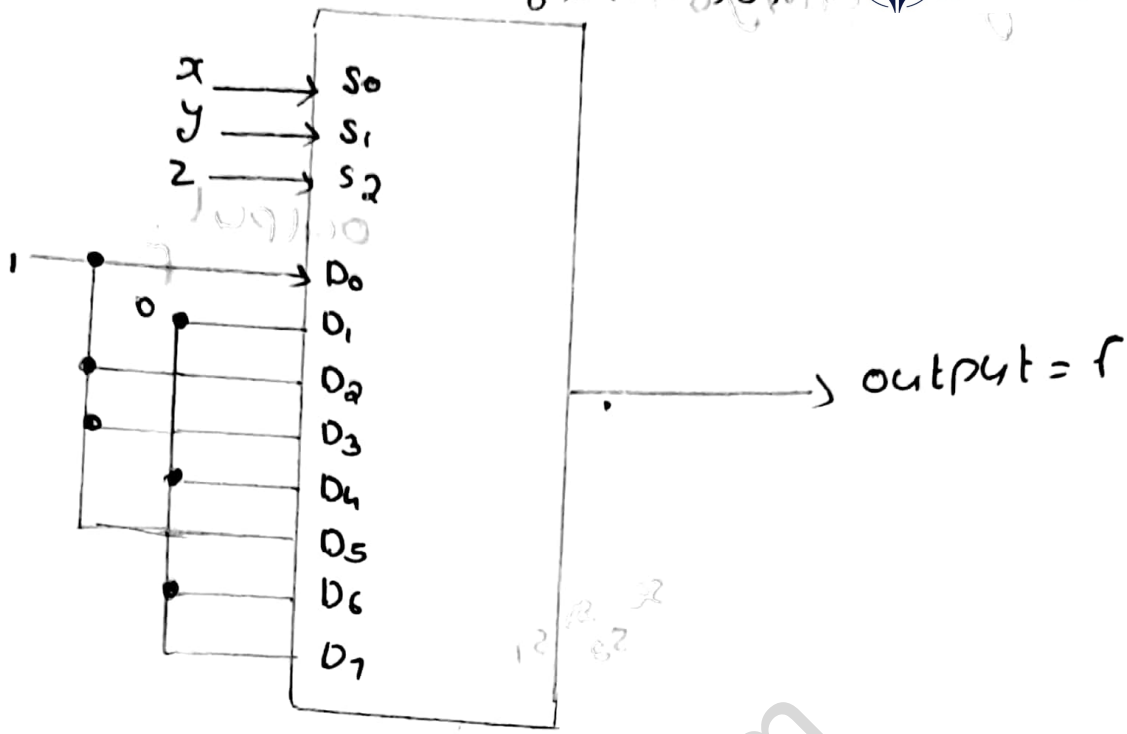
~~Boolean Expression:~~

keralanotes.com

For More Study Materials : https://www.keralanotes.com/

KeralaNotes

8 x 1 MUX



2. Use a 4:1 MUX to implement the logic function

$F(A,B,C) = \Sigma m(1,2,4,7)$

truth table

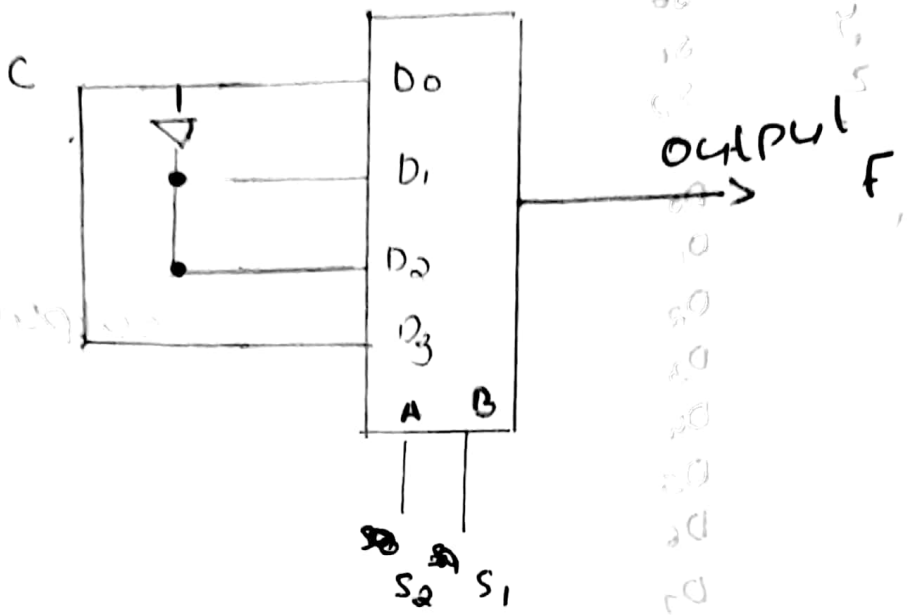
Minterm	S_2 A	S_1 B	C	F		
0	0	0	0	0	F = C	D_0
1	0	0	1	1		
2	0	1	0	1	$F = \bar{C}$	D_1
3	0	1	1	0		
4	1	0	0	1	$F = \bar{C}$	D_2
5	1	0	1	0		
6	1	1	0	0	F = C	D_3
7	1	1	1	1		

comparing

keralanotes.com

For More Study Materials : https://www.keralanotes.com/

logic diagram



3. use a multiplexer having four data select to implement the logic for the function given below. Also realize the same using 16:1 MUX and 8:1 MUX

$$f = \Sigma m(0,1,2,3,4,10,11,14,15)$$

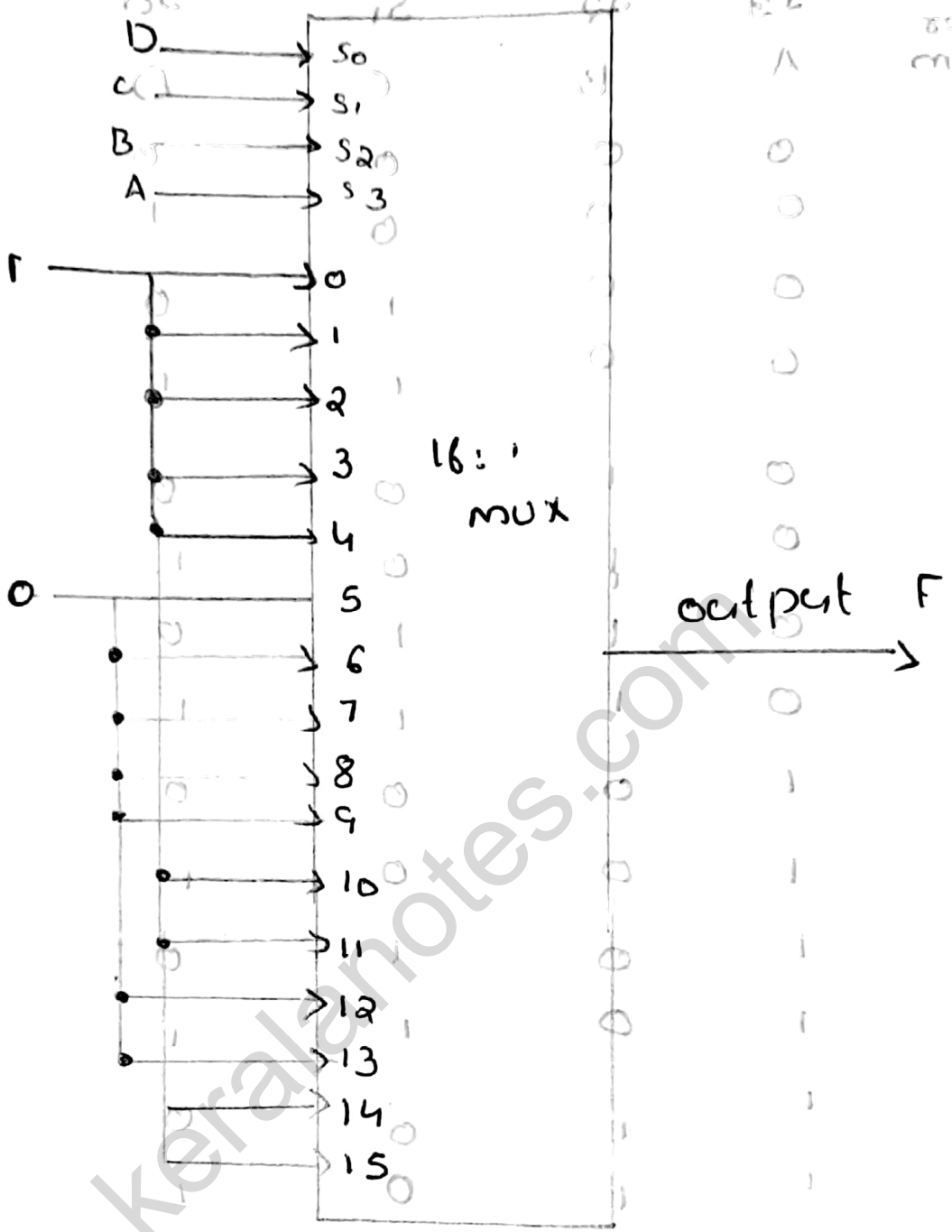
16:1 MUX

keralanotes.com

Minterm	S_3	S_2	S_1	S_0	F
	A	B	C	D	
0	0	0	0	0	1
1	0	0	0	1	1
2	0	0	1	0	1
3	0	0	1	1	1
4	0	1	0	0	1
5	0	1	0	1	0
6	0	1	1	0	0
7	0	1	1	1	0
8	1	0	0	0	0
9	1	0	0	1	0
10	1	0	1	0	1
11	1	0	1	1	1
12	1	1	0	0	0
13	1	1	0	1	0
14	1	1	1	0	1
15	1	1	1	1	1

KeralaNotes
keralanotes.com
For More Study Materials : https://www.keralanotes.com/

logic diagram



8:1 MUX

KeralaNotes
keralanotes.com

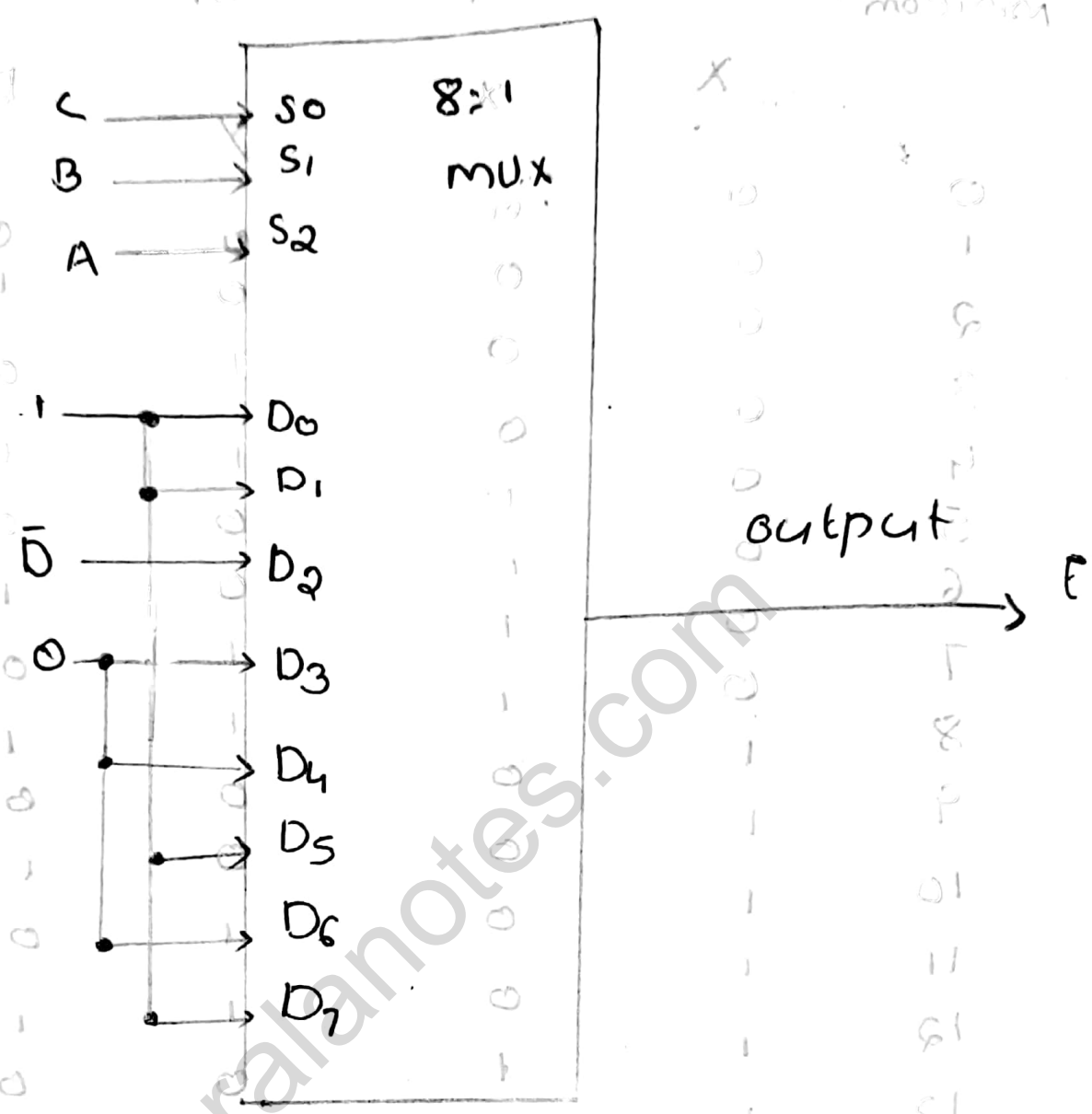
if F in both minterm 01 is =1 then F=1 we case

minterm	S_3	S_2	S_1			
	X	X	Y	D	F	
0	0	0	0	0	1	F=1
1	0	0	0	1	1	
2	0	0	1	0	1	F=1
3	0	0	1	1	1	
4	0	1	0	0	1	F=$\bar{D}$
5	0	1	0	1	0	
6	0	1	1	0	0	F=0
7	0	1	1	1	0	
8	1	0	0	0	0	F=0
9	1	0	0	1	0	
10	1	0	1	0	1	F=1
11	1	0	1	1	1	
12	1	1	0	0	0	F=0
13	1	1	0	1	0	
14	1	1	1	0	1	F=1
15	1	1	1	1	1	

KeralaNotes
keralanotes.com
For More Study Materials : https://www.keralanotes.com/

KeralaNotes

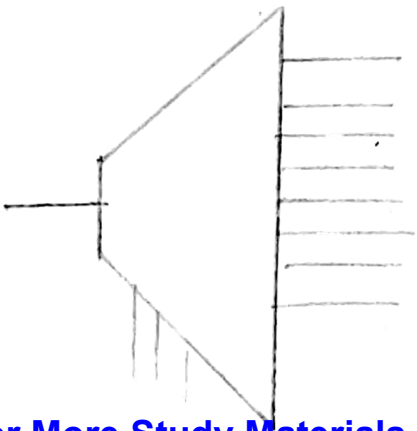
logic diagram



4. Use a Demultiplexer implement the functions $f_1 = \Sigma m(0,3,7)$

$f_2 = \Sigma m(1,2,5)$

Here (0,3) Number is 7 so it is 1:8 DE

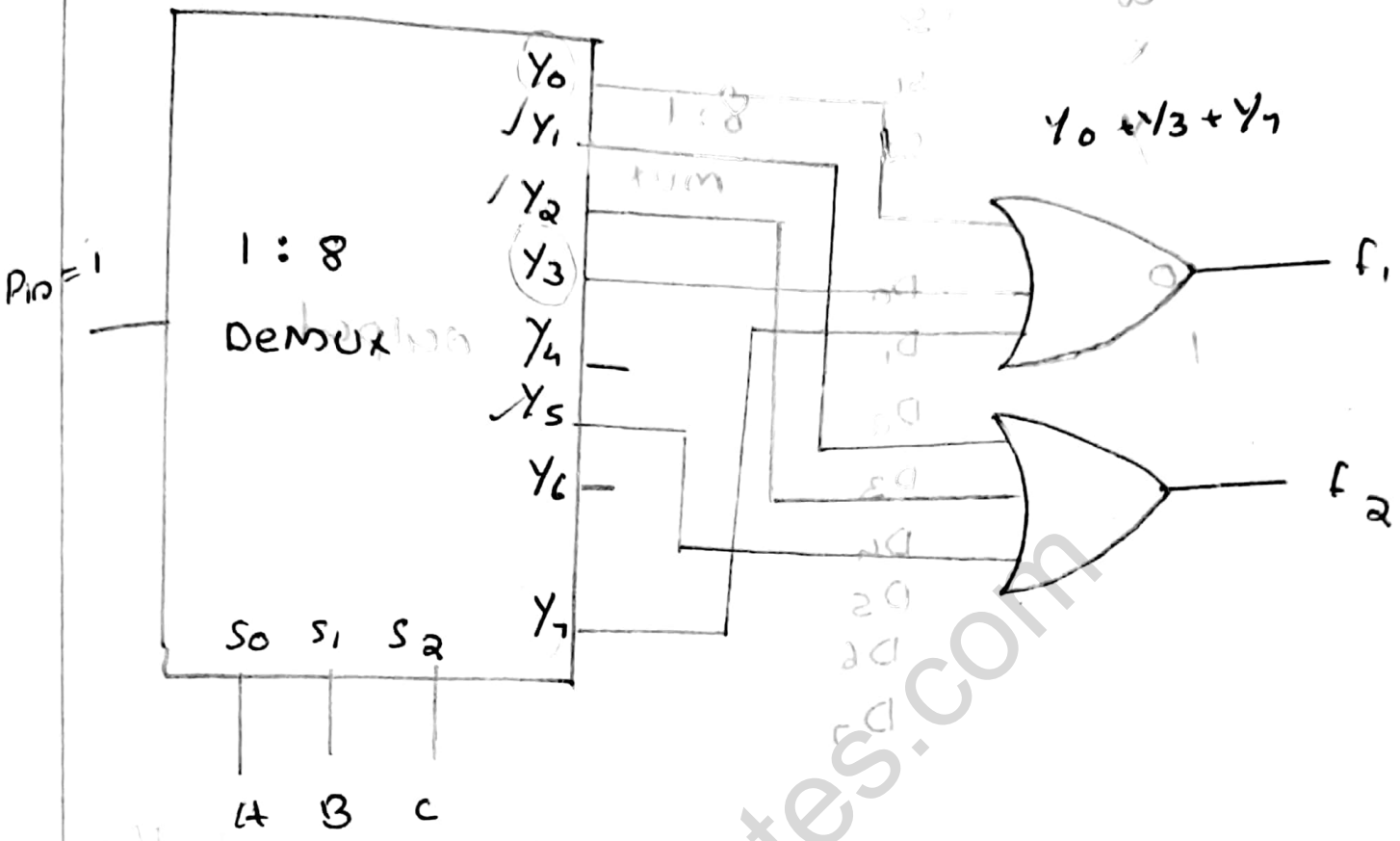


keralanotes.com

For More Study Materials : https://www.keralanotes.com/

KeralaNotes

Logical Diagram



5. Use mutiplexer implement the logic function $f = A \oplus B \oplus C$

imp

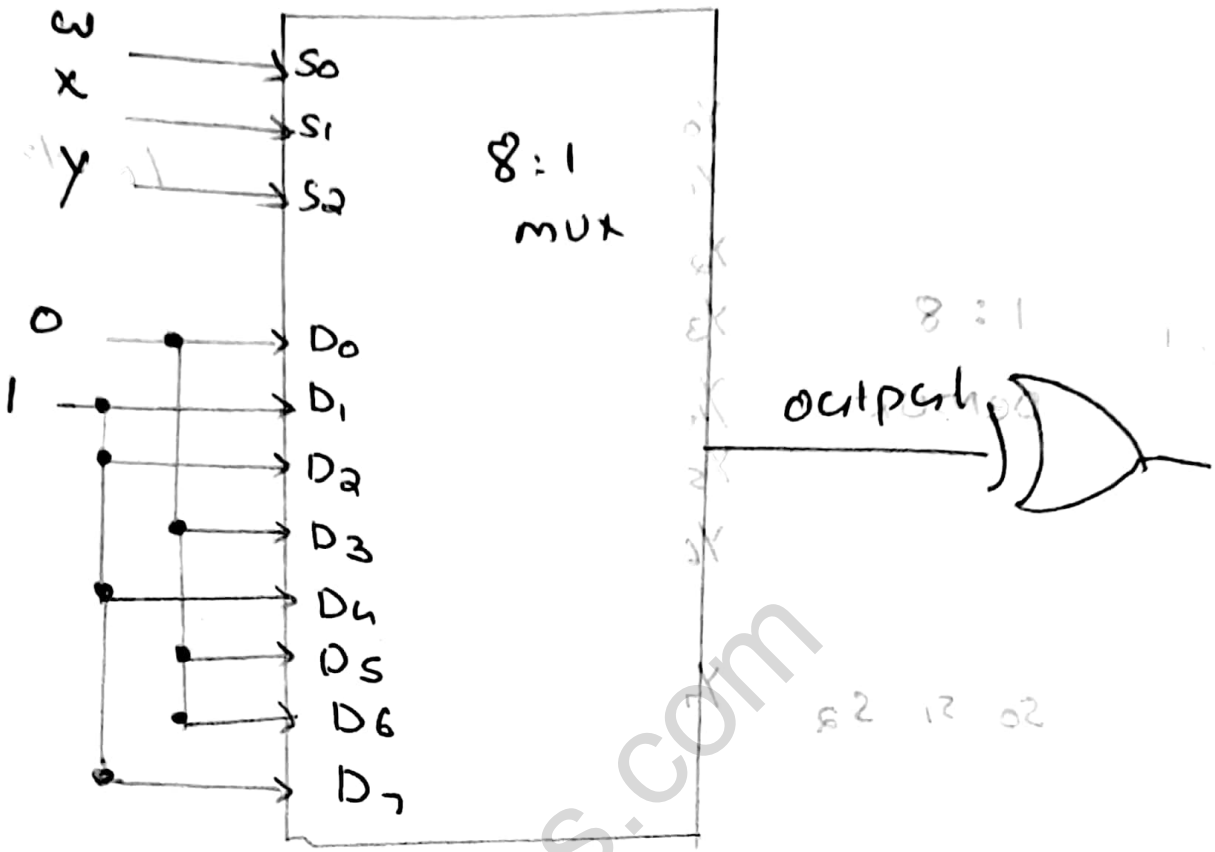
Inputs			outputs	
W	X	Y		
0	0	0	0	D_0
0	0	1	1	D_1
0	1	0	1	D_2
0	1	1	0	D_3
1	0	0	1	D_4
1	0	1	0	D_5
1	1	0	0	D_6
1	1	1	1	D_7

input odd no. of one's → output → 1

keralanotes.com

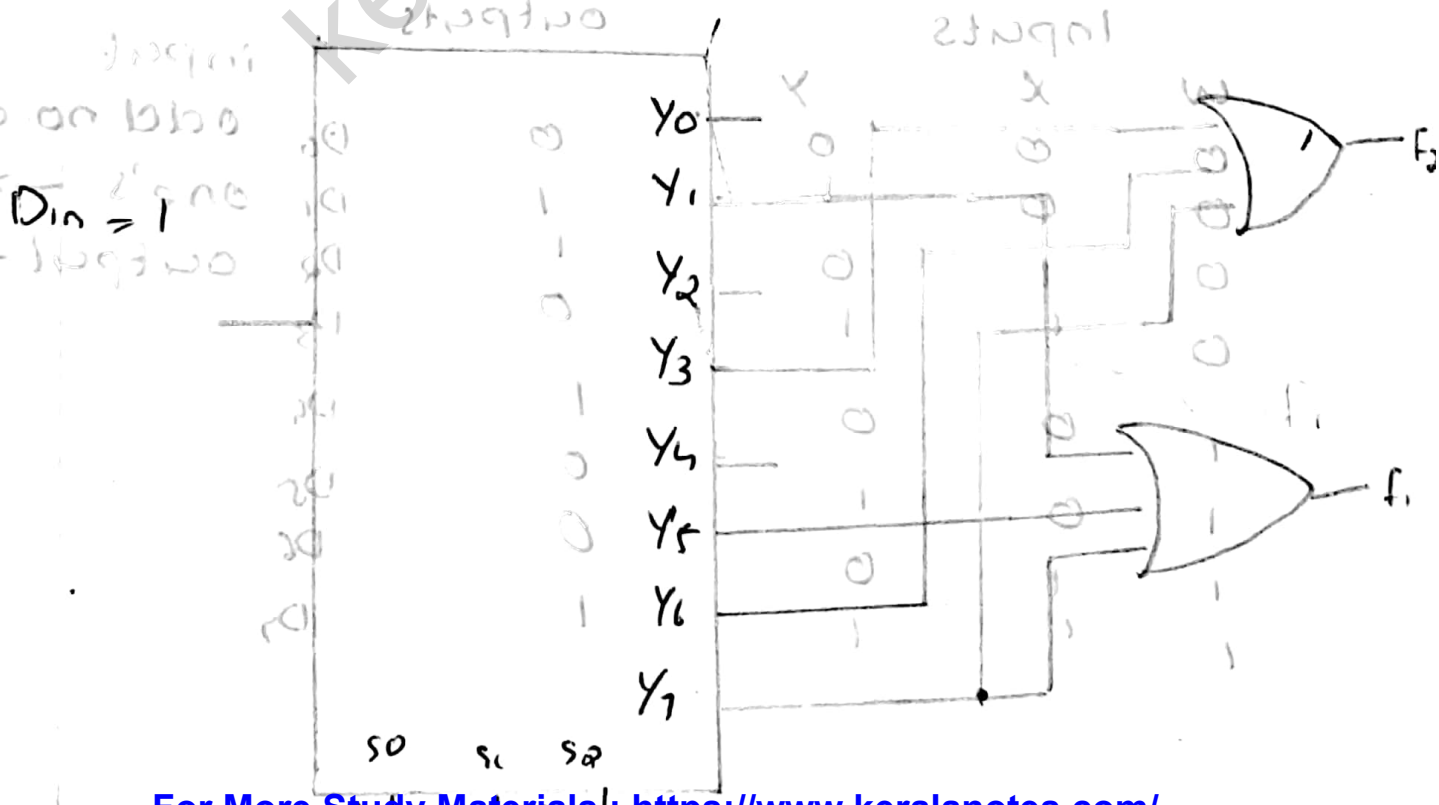
For More Study Materials : https://www.keralanotes.com/

KeralaNotes



6. Use a Demultiplexer implement the following functions $f_1 = \Sigma m(1,5,7)$

$f_2 = \Sigma m(3,6,7)$



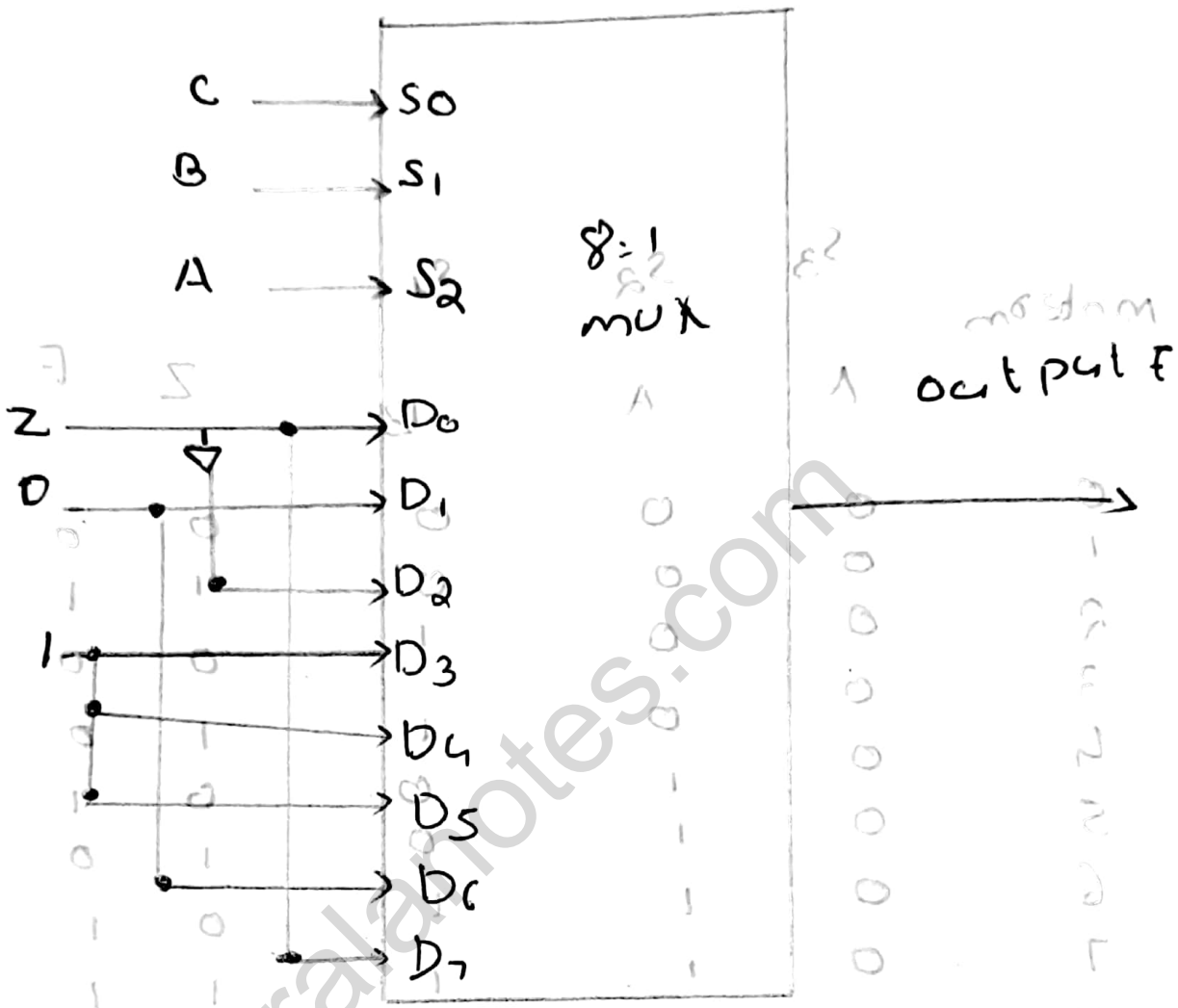
For More Study Materials : https://www.keralanotes.com/

7 Implement the function $F(w, x, y, z) = \Sigma m(1, 4, 6, 7, 8, 9, 10, 11, 15)$ using 8:1 MUX

Minterm	S_3	S_2	S_1				
	A	A	B	Z	F		
0	0	0	0	0	0	F=Z	D_0
1	0	0	0	1	1		
2	0	0	1	0	0	F=0	D_1
3	0	0	1	1	0		
4	0	1	0	0	1	F=$\bar{Z}$	D_2
5	0	1	0	1	0		
6	0	1	1	0	1	F=1	D_3
7	0	1	1	1	1		
8	1	0	0	0	1	F=1	D_4
9	1	0	0	1	1		
10	1	0	1	0	1	F=1	D_5
11	1	0	1	1	1		
12	1	1	0	0	0	F=0	D_6
13	1	1	0	1	0		
14	1	1	1	0	0	F=Z	D_7
15	1	1	1	1	1		

KeralaNotes
keralanotes.com
For More Study Materials : https://www.keralanotes.com/

KeralaNotes



keralanotes.com

Decoder and Encoder

Decoder

- Decoder is a multiple input multiple output logic circuit which converts coded inputs to coded output, where the input and output are different.

For More Study Materials : https://www.keralanotes.com/

KeralaNotes

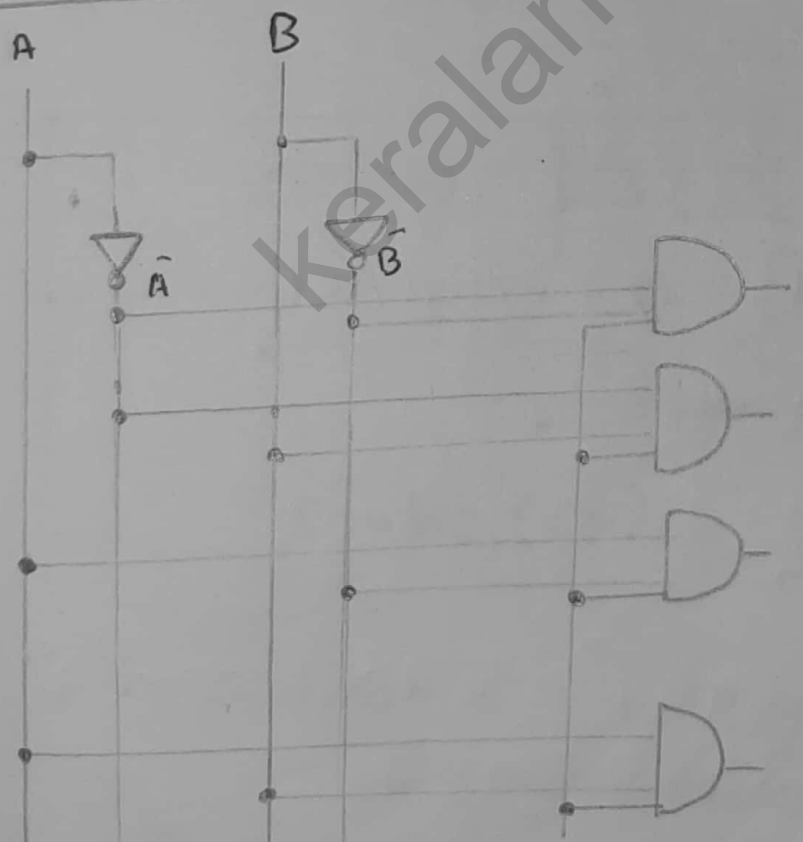
- Decoder consist of n input and 2^n maximum possible output

Eg: $2^2=4$ $2^3=8$

2-4 / 3-8 / 4-16

Binary Decoder (2 to 4 Decoder)

Inputs			outputs			
Enable	A	B	Y_3	Y_2	Y_1	Y_0
1	0	0	0	0	0	1
1	0	1	0	0	1	0
1	1	0	0	1	0	0
1	1	1	1	0	0	0



$Y_0 = \bar{A}\bar{B}$

$Y_1 = \bar{A}B$

$Y_2 = A\bar{B}$

$Y_3 = AB$

Enable (EN)

keralanotes.com

For More Study Materials : https://www.keralanotes.com/

3 to 8 decoder

	Inputs				output							
	Enable	A	B	C	Y_7	Y_6	Y_5	Y_4	Y_3	Y_2	Y_1	Y_0
$\bar{A}\bar{B}\bar{C}$	1	0	0	0	0	0	0	0	0	0	0	1
$\bar{A}\bar{B}C$	1	0	0	1	0	0	0	0	0	0	1	0
$\bar{A}B\bar{C}$	1	0	1	0	0	0	0	0	0	1	0	0
$\bar{A}BC$	1	0	1	1	0	0	0	0	1	0	0	0
$A\bar{B}\bar{C}$	1	1	0	0	0	0	0	1	0	0	0	0
$A\bar{B}C$	1	1	0	1	0	0	1	0	0	0	0	0
$AB\bar{C}$	1	1	1	0	0	1	0	0	0	0	0	0
ABC	1	1	1	1	1	0	0	0	0	0	0	0

$Y_0 = \bar{A}\bar{B}\bar{C}$

$Y_1 = \bar{A}\bar{B}C$

$Y_2 = \bar{A}B\bar{C}$

$Y_3 = \bar{A}BC$

$Y_4 = A\bar{B}\bar{C}$

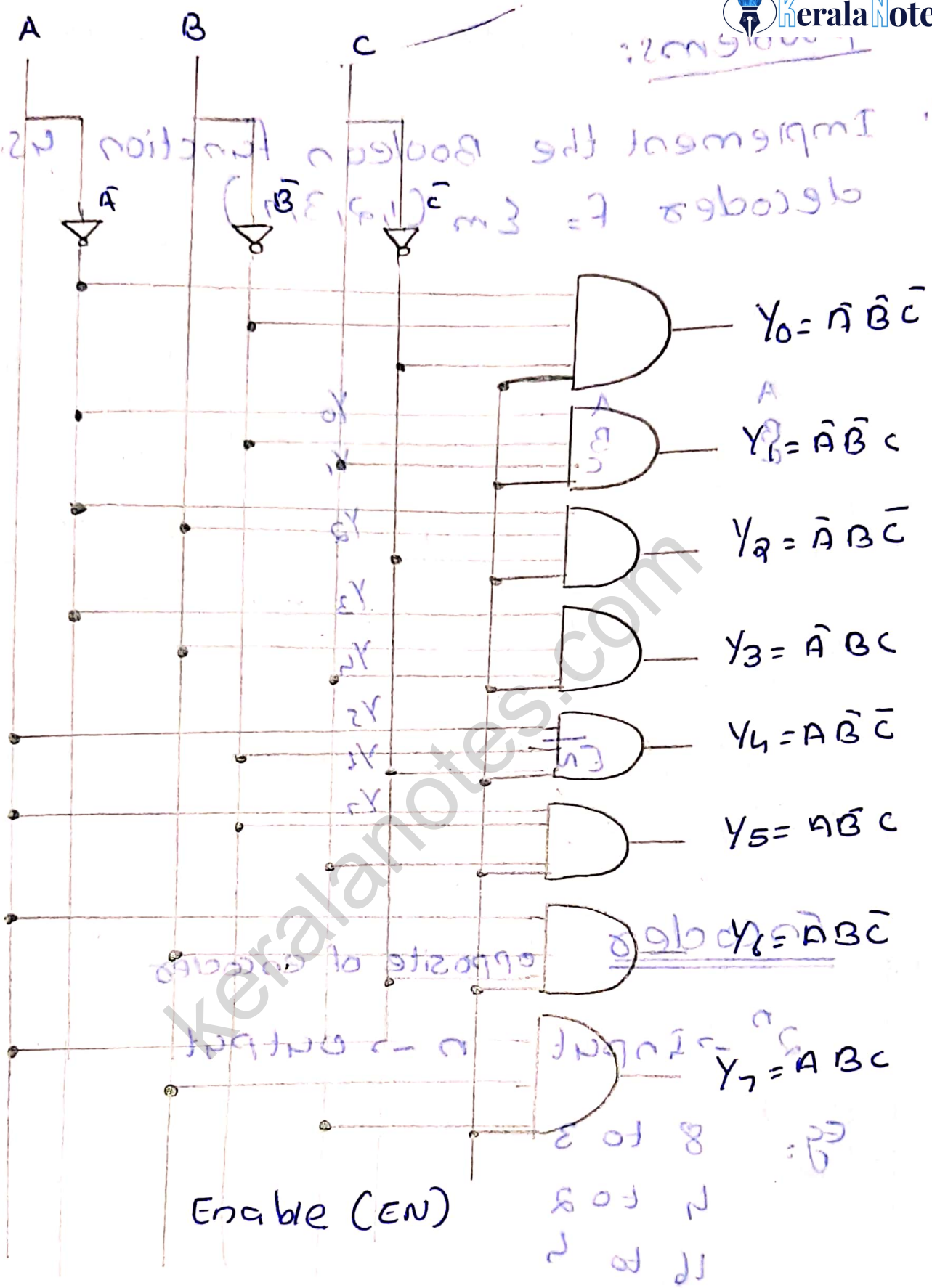
$Y_5 = A\bar{B}C$

$Y_6 = AB\bar{C}$

$Y_7 = ABC$

KeralaNotes
keralanotes.com
For More Study Materials : https://www.keralanotes.com/

KeralaNotes



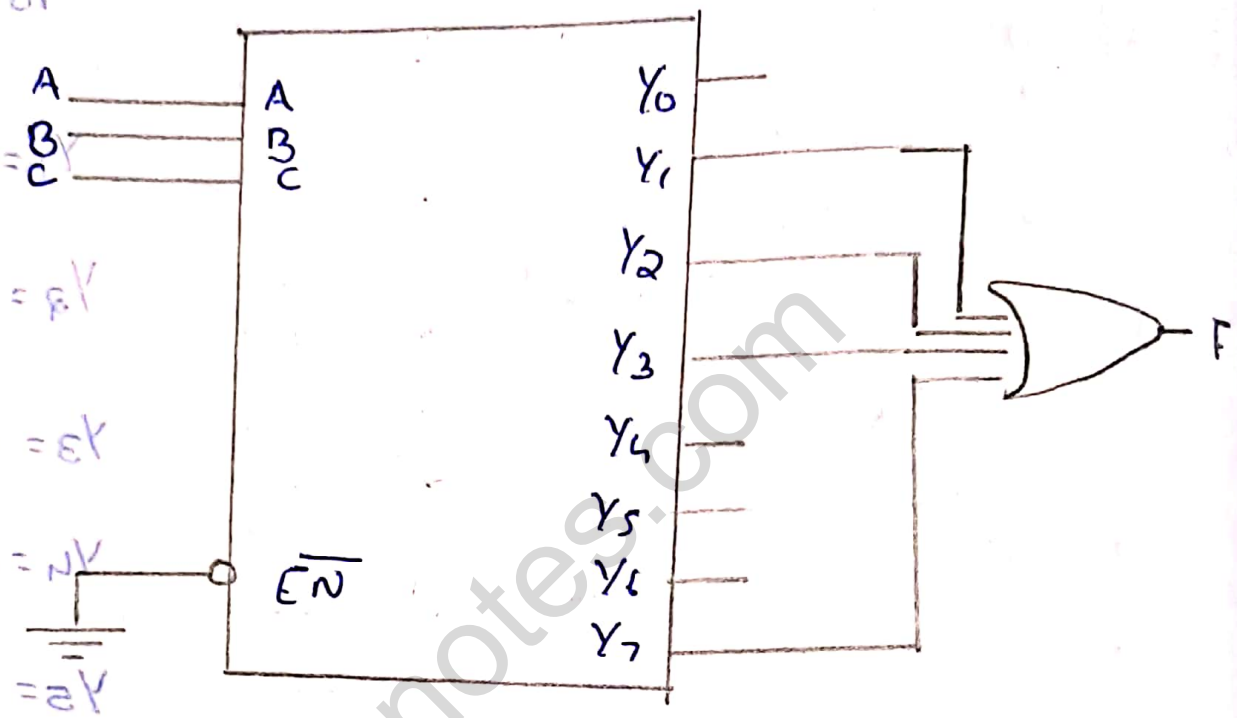
keralanotes.com

For More Study Materials : https://www.keralanotes.com/

KeralaNotes

Problems:

1. Implement the Boolean function using decoder $F = \Sigma m(1,2,3,7)$



Encoder

opposite of encoder

$2^n \rightarrow$ input $\quad n \rightarrow$ output

Eg: 8 to 3

4 to 2

16 to 4

- Decoder consist of n output and 2^n maximum possible input

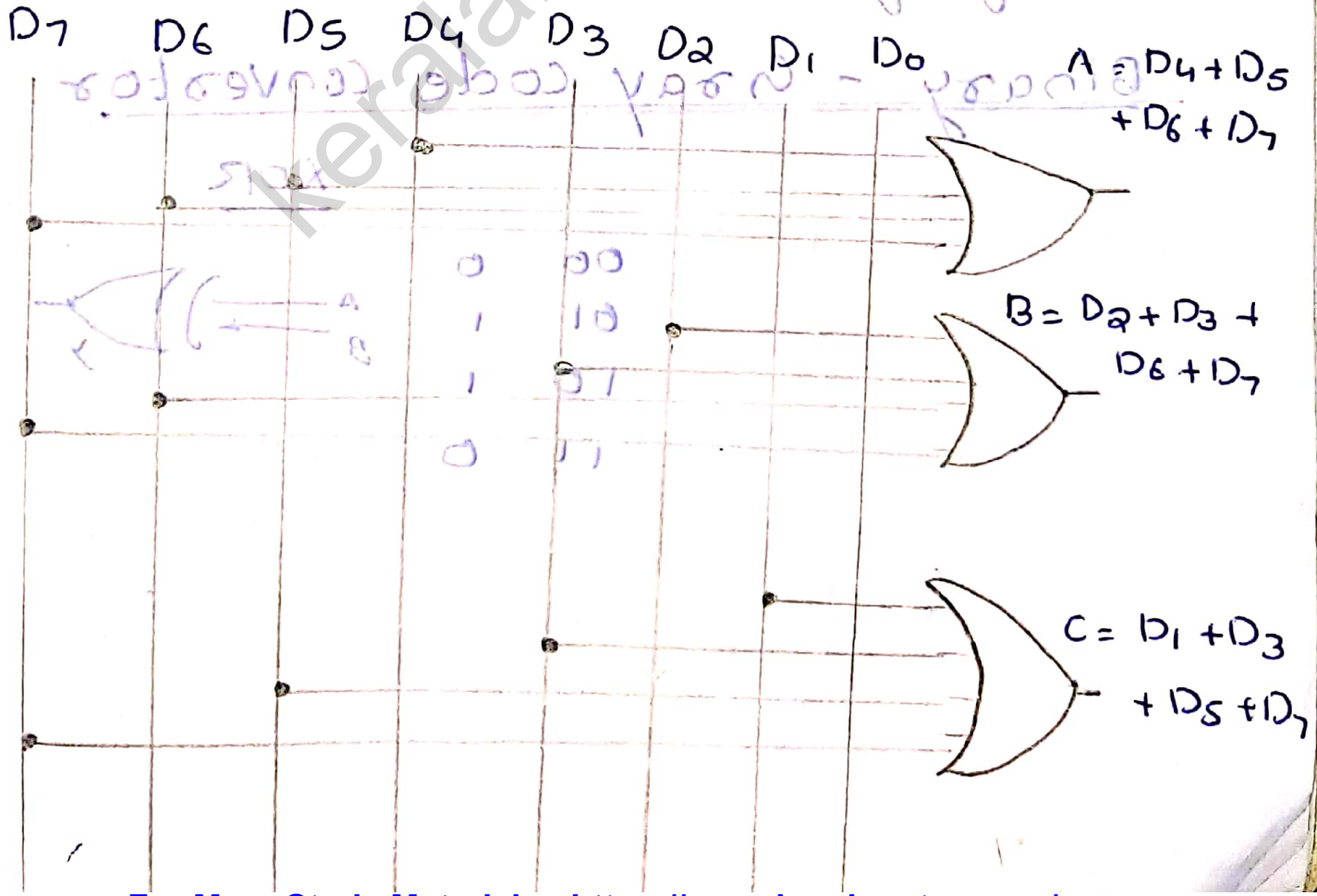
keralanotes.com

For More Study Materials : https://www.keralanotes.com/

KeralaNotes

8 to 3 Encoder (Octal to Binary)

Inputs								outputs		
D_0	D_1	D_2	D_3	D_4	D_5	D_6	D_7	A	B	C
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



keralanotes.com

For More Study Materials : https://www.keralanotes.com/

KeralaNotes

Code convertors

- Convert one to another code

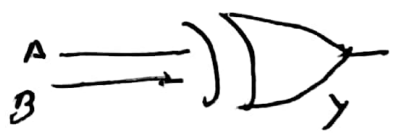
Steps:

1. Truth table showing relationship between input and output
2. For Each output code determine the simplified boolean expression using KMAP.
3. Realize the code convertor using Logic gates.

Binary - Gray code convertor

00	0
01	1
10	1
11	0

XOR



keralanotes.com

For More Study Materials : https://www.keralanotes.com/

KeralaNotes

MSA LSA

$B_3 \xrightarrow{\oplus} B_2 \xrightarrow{\oplus} B_1 \xrightarrow{\oplus} B_0$

Eg:

1	0	1	1
↓	↓	↓	↓
1	1	1	0
G_3	G_2	G_1	G_0

XOR of B_3 B_2

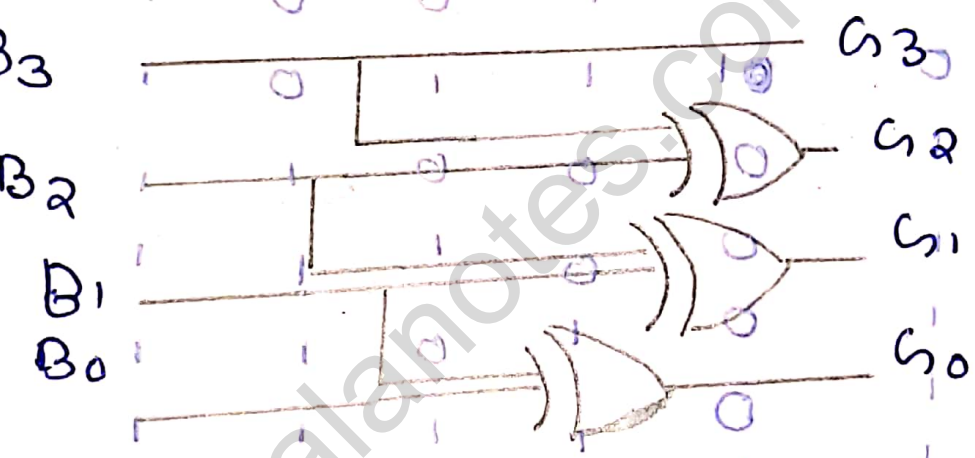
XOR

B_3 B_2

$= 1$

(1) $G_3 = B_3$ (2) $G_2 = B_3 \oplus B_2$

(3) $G_1 = B_2 \oplus B_1$ (4) $G_0 = B_1 \oplus B_0$



Eg:- 1 1 0 0

↓ ↓ ↓ ↓

1 0 1 0

keralanotes.com

For More Study Materials : https://www.keralanotes.com/

KeralaNotes

Decimal	Binary code				Grey ~~Decimal~~ code			
	D	C	B	A	G_3	G_2	G_1	G_0
0	0	0	0	0	0	0	0	0
1	0	0	0	1	0	0	0	1
2	0	0	1	0	0	0	1	1
3	0	0	1	1	0	0	1	0
4	0	1	0	0	0	1	1	0
5	0	1	0	1	0	1	1	1
6	0	1	1	0	0	1	0	1
7	0	1	1	1	0	1	0	0
8	1	0	0	0	1	1	0	0
9	1	0	0	1	1	1	0	1
10	1	0	1	0	1	1	1	1
11	1	0	1	1	1	1	1	0
12	1	1	0	0	1	0	1	0
13	1	1	0	1	1	0	1	1
14	1	1	1	0	1	0	0	1
15	1	1	1	1	1	0	0	0

Grey to Binary convertor

G_3	G_2	G_1	G_0
1	1	0	0
1	0	0	0
B_3	B_2	B_1	B_0

① $B_3 = G_3$ ② $B_2 = B_3 \oplus G_2$

keralanotes.com

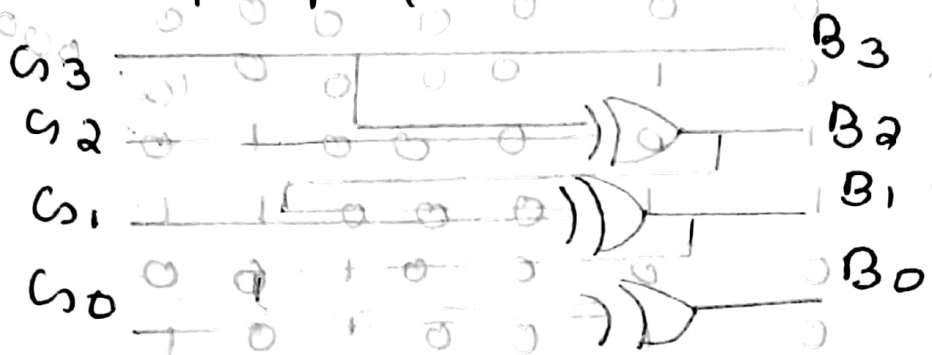
For More Study Materials : https://www.keralanotes.com/

KeralaNotes

(3) $B_1 = B_2 \oplus G_1$ (4) $B_0 = B_1 \oplus G_0$

Eg: 1 0 0 0

1 1 1 1



Gray code				Binary code			
G_3	G_2	G_1	G_0	D	C	B	A
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1
0	0	1	1	0	0	1	0
0	0	1	0	0	0	1	1
0	1	1	0	0	1	0	0
0	1	1	1	0	1	0	1
0	1	0	1	0	1	1	0
0	1	0	0	0	1	1	1
1	1	0	0	1	0	0	0
1	1	0	1	1	0	0	1
1	1	1	1	1	0	1	0
1	1	1	0	1	0	1	1
1	0	1	0	1	1	0	0
1	0	1	1	1	1	0	1
1	0	0	1	1	1	1	0
1	0	0	0	1	1	1	1

keralanotes.com

For More Study Materials : https://www.keralanotes.com/

Binary to BCD converter

KeralaNotes

000 1 0101

5 bit

14

000 1 01

Binary code				BCD code				
D	C	B	A	B_4	B_3	B_2	B_1	B_0
0	0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0	1
0	0	1	0	0	0	0	1	0
0	0	1	1	0	0	0	1	1
0	1	0	0	0	0	1	0	0
0	1	0	1	0	0	1	0	1
0	1	1	0	0	0	1	1	0
0	1	1	1	0	0	1	1	1
1	0	0	0	0	1	0	0	0
1	0	0	1	0	1	0	0	1
1	0	1	0	1	0	0	0	0
1	0	1	1	1	0	0	0	1
1	1	0	0	1	0	0	1	0
1	1	0	1	1	0	0	1	1
1	1	1	0	1	0	1	0	0
1	1	1	1	1	0	1	0	1

— same

For B_0

DC \ BA	00	01	11	10
00	0	1	1	0
01	0	1	1	0
11	0	1	1	0
10	0	1	1	0

$B_0 = A$

For B_1

DC \ BA	00	01	11	10
00	0	0	1	1
01	0	0	1	1
11	1	1	0	0
10	0	0	0	0

$B_1 = DCB + \bar{D}B$

keralanotes.com

For More Study Materials : https://www.keralanotes.com

KeralaNotes

For B_2

DC \ BA	00	01	11	10
00	0	0	0	0
01	1	1	1	1
11	0	0	1	1
10	0	0	0	0

$D_2 = \bar{D}C + CB$

For B_3

DC \ BA	00	01	11	10
00	0	0	0	0
01	0	0	0	0
11	0	0	0	0
10	1	1	0	0

$D_3 = D\bar{C}\bar{B}$

For B_4

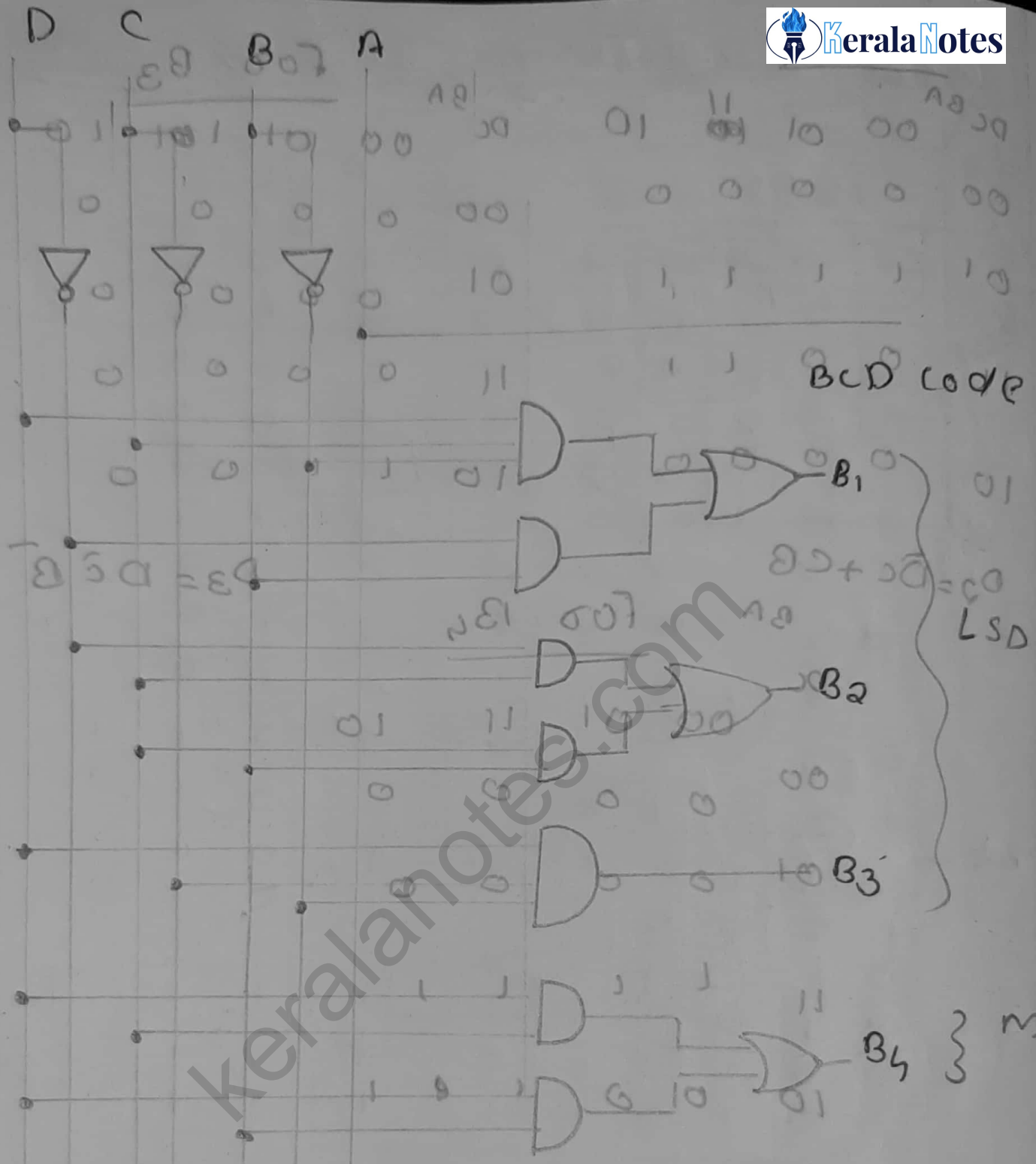
DC \ BA	00	01	11	10
00	0	0	0	0
01	0	0	0	0
11	1	1	1	1
10	0	0	1	1

$B_4 = DC + DB$

keralanotes.com

For More Study Materials : https://www.keralanotes.com/

KeralaNotes



BCD to Excess 3 [A + 3]

1 → 1 + 3 = 4

keralanotes.com

For More Study Materials : https://www.keralanotes.com/

KeralaNotes

Decimal	B_3	B_2	B_1	B_0	E_3	E_2	E_1	E_0
0	0	0	0	0	0	0	1	1
1	0	0	0	1	0	1	0	0
2	0	0	1	0	0	1	0	1
3	0	0	1	1	0	1	1	0
4	0	1	0	0	0	1	1	1
5	0	1	0	1	1	D	0	0
6	0	1	1	0	1	D	0	1
7	0	1	1	1	1	D	1	0
8	1	0	0	0	1	D	1	1
9	1	0	0	1	1	1	0	0

0-9 only available 10-15 → Don't care

B_1B_0 For E_3

B_3B_2			
0	0	0	0
0	1	1	1
X	X	X	X
1	1	X	X

Don't care 10-15 → X

B_1B_0 For E_2

B_3B_2			
0	1	1	1
1	0	0	0
X	X	X	X
0	1	X	X

B_1B_0 For E_1

B_3B_2			
1	0	1	0
1	0	1	0
X	X	X	X
1	0	X	X

B_1B_0 For E_0

B_3B_2			
1	0	0	1
1	0	0	1
X	X	X	X
1	0	X	X

keralanotes.com

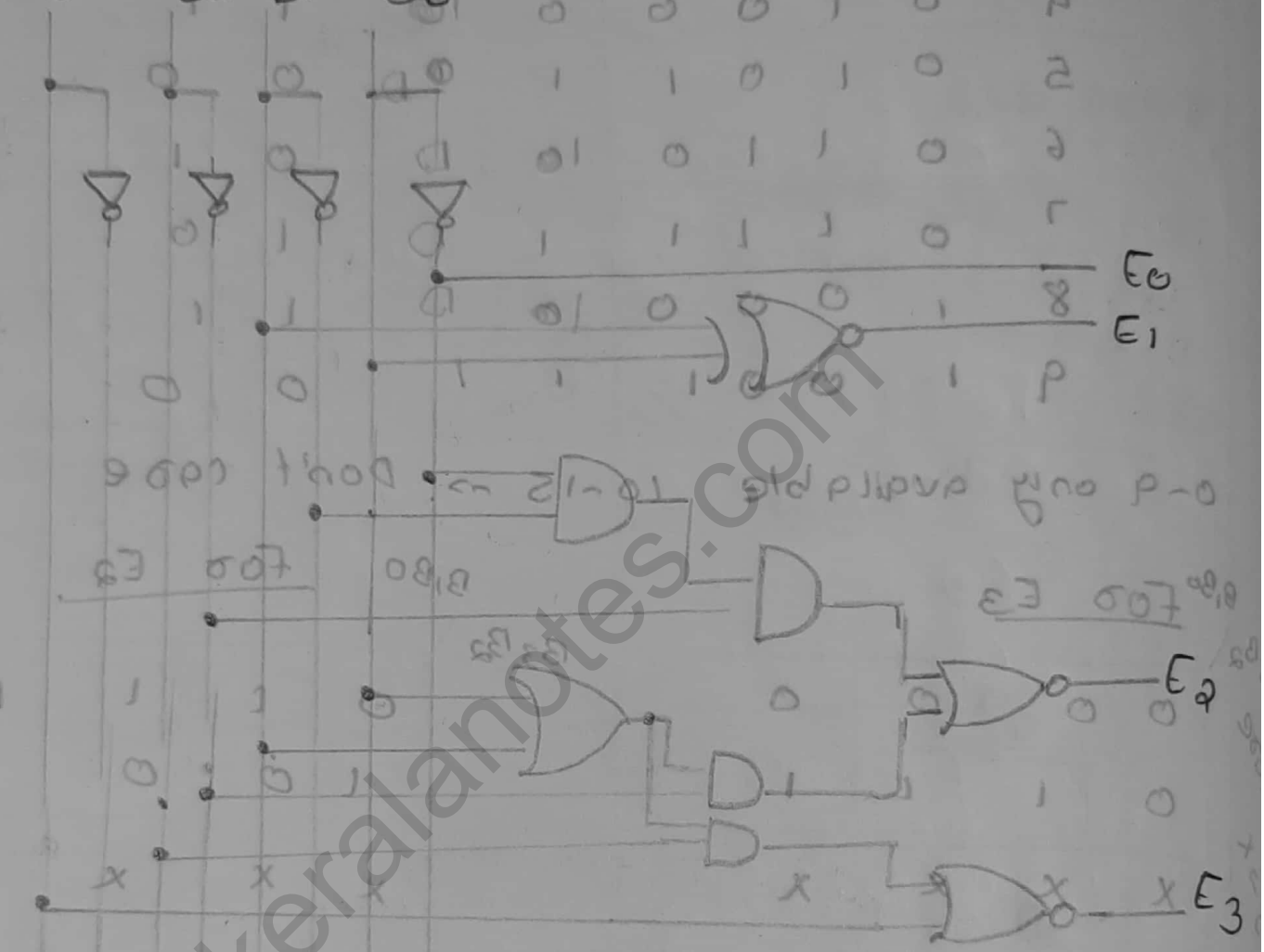
For More Study Materials : https://www.keralanotes.com/

$E_3 = B_3 + B_2(B_0 + B_1)$ $\quad E_2 = B_2 \overline{B_1} \overline{B_0} + B_2(B_0 + B_1)$

$E_1 = \overline{B_1}\,\overline{B_0} + B_1 B_0$

$= B_1 \odot B_0$

$E_0 = \overline{B_0}$



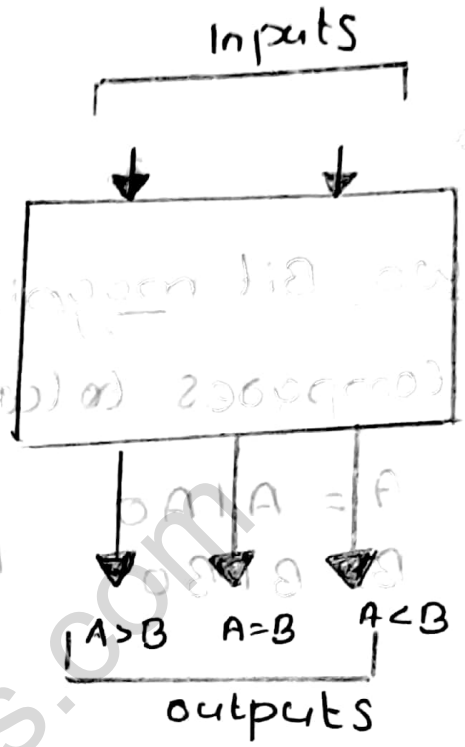
Magnitude comparators

- Combinational circuit that compares two digital or binary numbers (A & B) and determines their relative magnitudes in order to find out whether one number is equal, less than or greater than the other digital number

KeralaNotes
keralanotes.com
For More Study Materials : https://www.keralanotes.com/

KeralaNotes

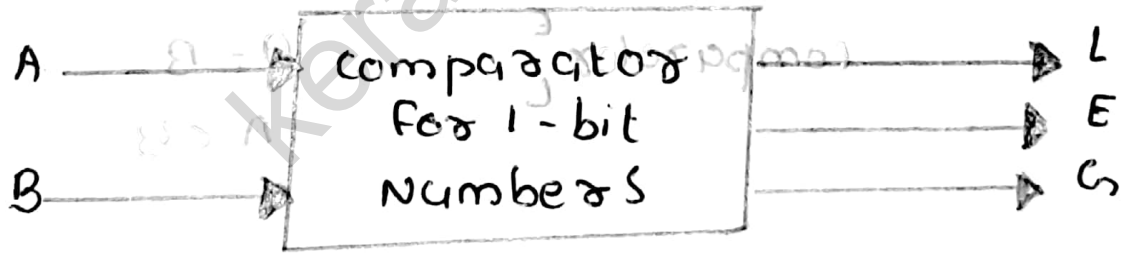
- three binary are used to indicate the outcome of the comparison as $A>B$, $A<B$, or $A=B$



Single bit magnitude comparator

- comparitors used to compare two bits; two numbers each of single bit is called single bit comparator

$L(A<B)$, $E(A=B)$, $G(A>B)$



Truth table

A_0	B_0	L (less than)	E (equal)	G (greater)
0	0	0	1	0
0	1	1	0	0
1	0	0	0	1
1	1	0	1	0

Boolean Expression

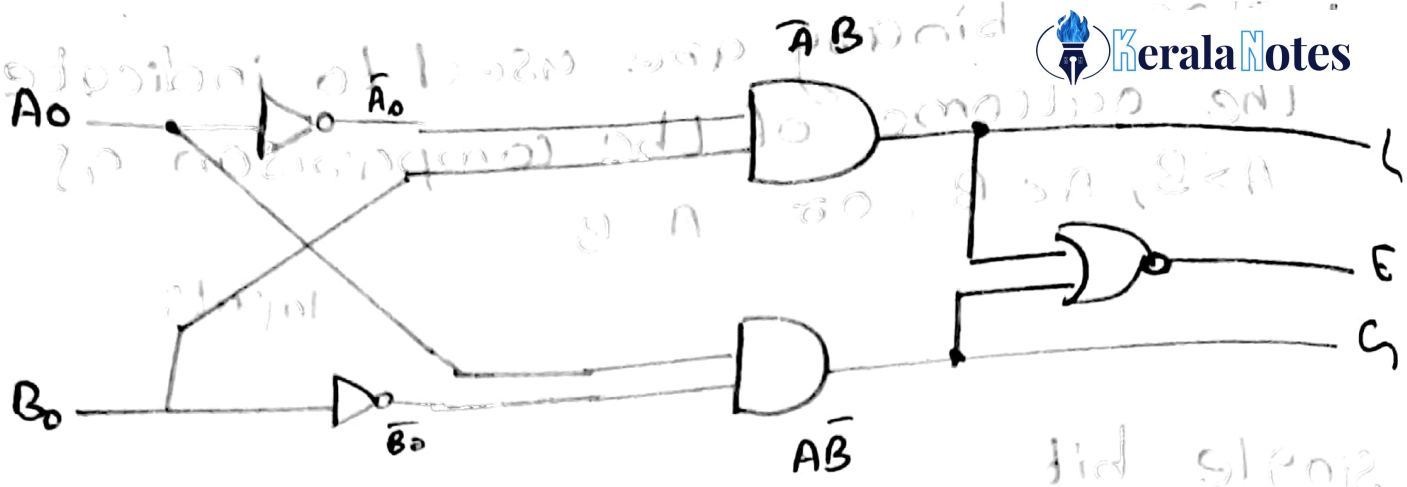
$E = \bar{A}\cdot\bar{B} + AB = \overline{A \oplus B}$

$L = \bar{A}B$

$G = A\bar{B}$

For More Study Materials : https://www.keralanotes.com/

KeralaNotes

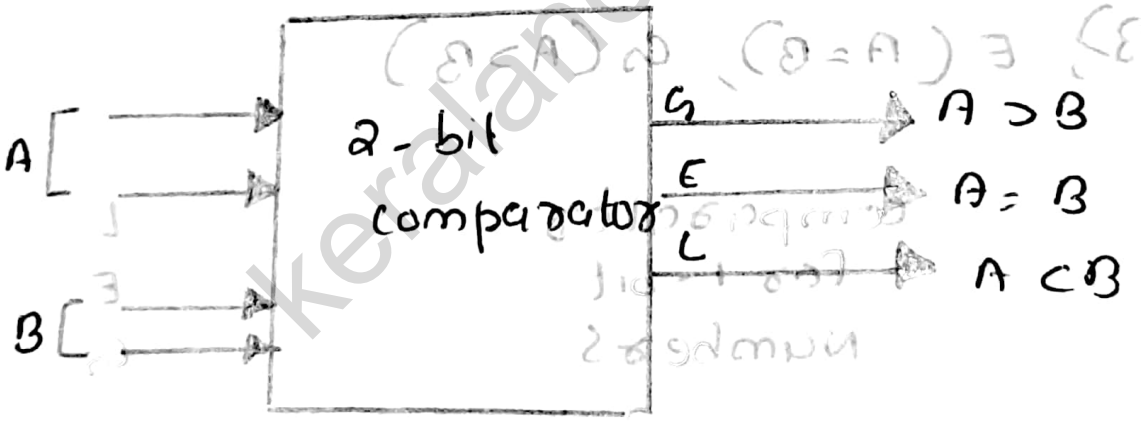


Two Bit magnitude comparator

- compares two binary numbers,

A = A1A0
B = B1B0

produced 3 output
G (G = 1 if A > B),
E (E = 1 if A = B)
L (L = 1 if A < B)



keralanotes.com

For More Study Materials : https://www.keralanotes.com/

KeralaNotes

Inputs				outputs		
A_1	A_0	B_1	B_0	$A>B$	$A=B$	$A<B$
0	0	0	0	0	1	0
0	0	0	1	0	0	1
0	0	1	0	0	0	1
0	0	1	1	0	0	1
0	1	0	0	1	0	0
0	1	0	1	0	1	0
0	1	1	0	0	0	1
0	1	1	1	0	0	1
1	0	0	0	1	0	0
1	0	0	1	1	0	0
1	0	1	0	0	1	0
1	0	1	1	0	0	1
1	1	0	0	1	0	0
1	1	0	1	1	0	0
1	1	1	0	1	0	0
1	1	1	1	0	1	0

keralanotes.com

A > B

A_1A_0 \ B_1B_0	00	01	11	10
00	0	0	0	0
01	1	0	0	0
11	1	1	0	1
10	1	1	0	0

A = B

A_1A_0 \ B_1B_0	00	01	11	10
00	1	0	0	0
01	0	1	0	0
11	0	0	1	0
10	0	0	0	1

For More Study Materials : https://www.keralanotes.com/

KeralaNotes

A < B

A_1A_0 \ B_1B_0	00	01	11	10
00		1	1	1
01			1	1
11				
10			1	

(G)

$A > B \rightarrow A_0 \bar{B}_1 \bar{B}_0 + A_1 \bar{B}_1 + A_1 A_0 \bar{B}_0$

$A = B \rightarrow \bar{A}_1 \bar{A}_0 \bar{B}_1 \bar{B}_0 + \bar{A}_1 A_0 \bar{B}_1 B_0 + A_1 A_0 B_1 B_0$

(E)

$A_1 \bar{A}_0 B_1 \bar{B}_0$

$= \bar{A}_1 \bar{B}_1 (\bar{A}_0 \bar{B}_0 + A_0 B_0) +$

$A_1 B_1 (A_0 B_0 + \bar{A}_0 \bar{B}_0)$

$= (A_0 B_0 + \bar{A}_0 \bar{B}_0)(A_1 B_1 + \bar{A}_1 \bar{B}_1)$

$= (A_0 \text{ Ex-NOR } B_0)(A_1 \text{ Ex-NOR } B_1$

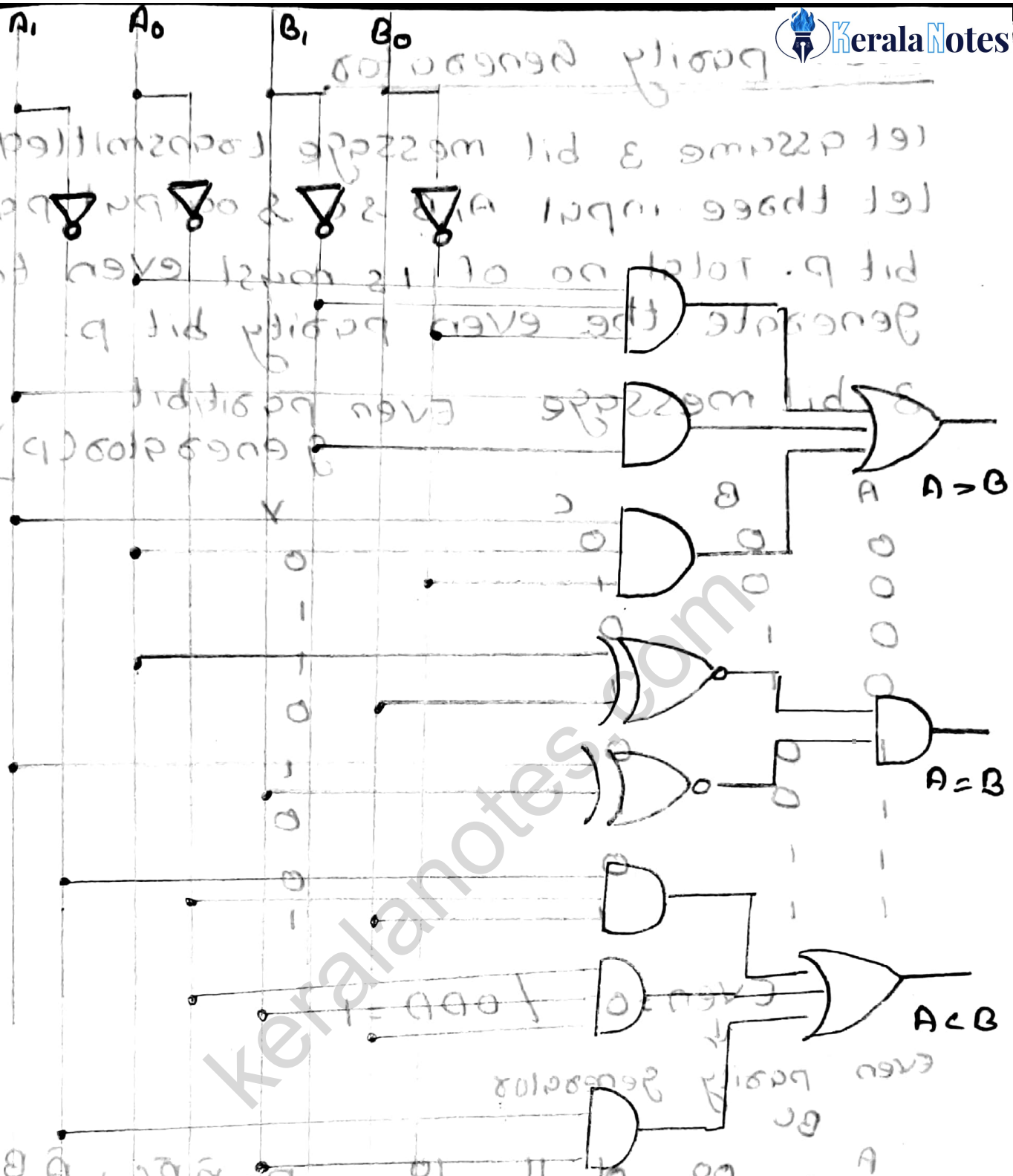
$A < B \rightarrow \bar{A}_1 B_1 + \bar{A}_0 B_1 B_0 + \bar{A}_1 \bar{A}_0 B_0$

(L)

keralanotes.com

For More Study Materials : https://www.keralanotes.com/

KeralaNotes



Parity Generator

- used for error detecting

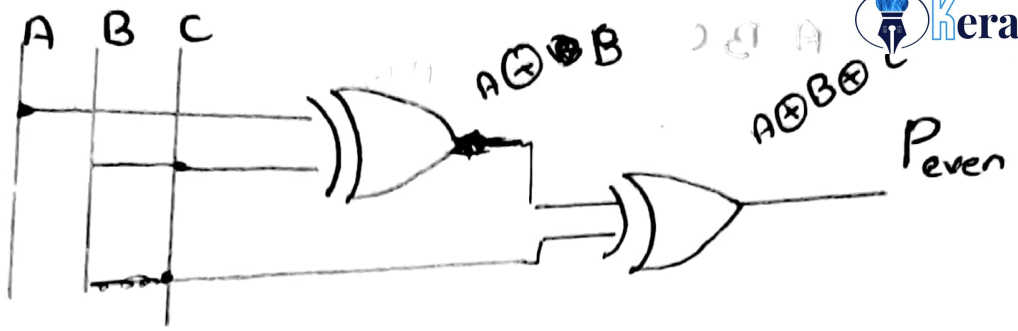
1, 3, 5, 7 → odd parity

0, 2, 4, 6 → Even parity

keralanotes.com

For More Study Materials : https://www.keralanotes.com/

KeralaNotes



Odd parity Generator

ODD = 0 even = 1

3 bit message			Even parity bit generator (P)
A	B	C	Y
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	0

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

$$P_{odd} = (\bar{A}\bar{B}\bar{C} + A\bar{B}C) + (\bar{A}BC + AB\bar{C})$$

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

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

$$= \bar{C}\bar{X} + CX$$

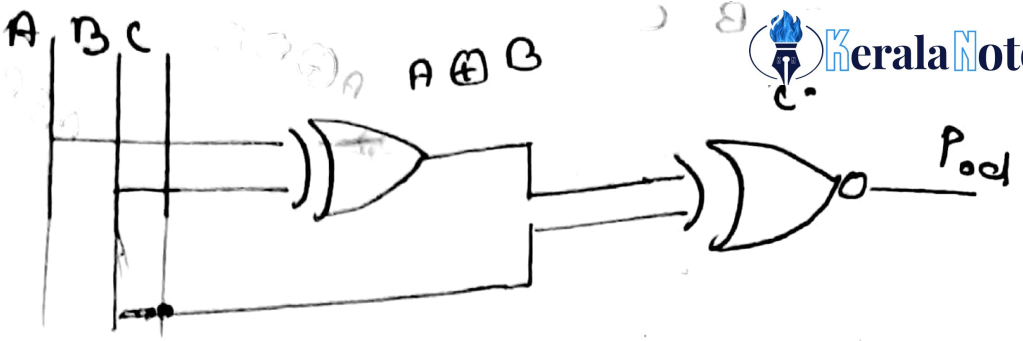
$$= C \odot X$$

$$\boxed{P_{odd} = C \odot \overline{A \oplus B}}$$

keralanotes.com

For More Study Materials : https://www.keralanotes.com/

KeralaNotes



Parity checker

(1) Even parity checker

* If parity Error checker (PEC) = 1 ⇒ Error occurs

* PEC = 0 ⇒ No Error

4-bit received message				parity error checker C_p
A	B	C	P_{parity}	
0	0	0	0	0
0	0	0	1	1 error
0	0	1	0	1 error
0	0	1	1	0
0	1	0	0	1
0	1	0	1	0
0	1	1	0	0
0	1	1	1	1
1	0	0	0	1
1	0	0	1	0
1	0	1	0	0
1	0	1	1	1
1	1	0	0	0
1	1	0	1	1
1	1	1	0	0
1	1	1	1	1

keralanotes.com

For More Study Materials : https://www.keralanotes.com/

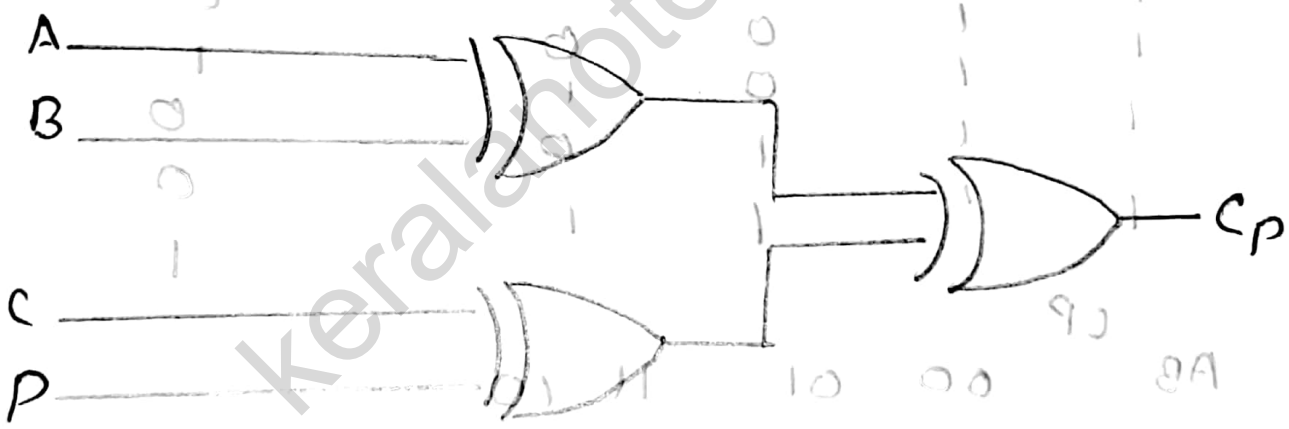
KeralaNotes

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

$$PEC = \bar{A}\bar{B}(\bar{C}D + C\bar{D}) + \bar{A}B(\bar{C}\bar{D} + CD) + AB(CD + C\bar{D}) + AB(\bar{C}\bar{D} + CD)$$

$$= \bar{A}\bar{B}(C \oplus D) + \bar{A}B(\overline{C \oplus D}) + AB(C \oplus D) + A\bar{B}(\overline{C \oplus D})$$

$$\boxed{PEC = (A \oplus B) \oplus (C \oplus B)}$$



(ii) ODD parity checker

IF

* PEC = 1 error (odd no. of ones)
  PEC = 0 no error

keralanotes.com

4-bit recieved message

A	B	C	parit	parity check C_p
0	0	0	0	1
0	0	0	1	0
0	0	1	0	0
0	0	1	1	1
0	1	0	0	0
0	1	0	1	1
0	1	1	0	1
0	1	1	1	0
1	0	0	0	0
1	0	0	1	1
1	0	1	0	1
1	0	1	1	0
1	1	0	0	1
1	1	0	1	0
1	1	1	0	0
1	1	1	1	1

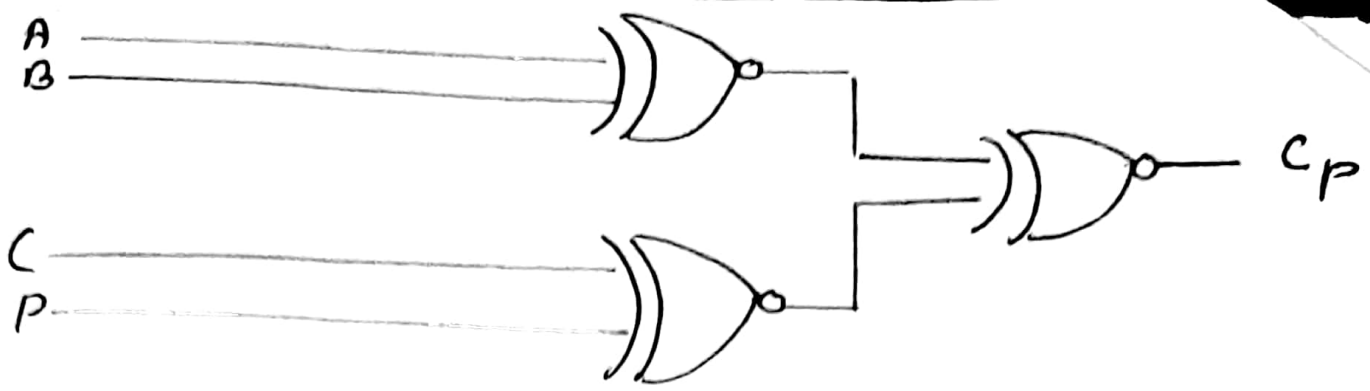
No error

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

PEC = (A Ex-NOR B) Ex-NOR (C Ex-NOR D)

$(A \odot B) \odot (C \odot D)$

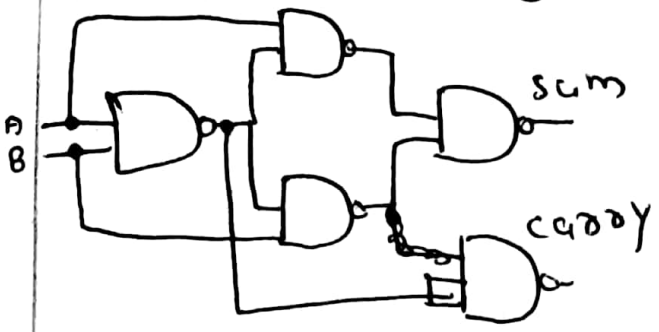
KeralaNotes
keralanotes.com
For More Study Materials : https://www.keralanotes.com/



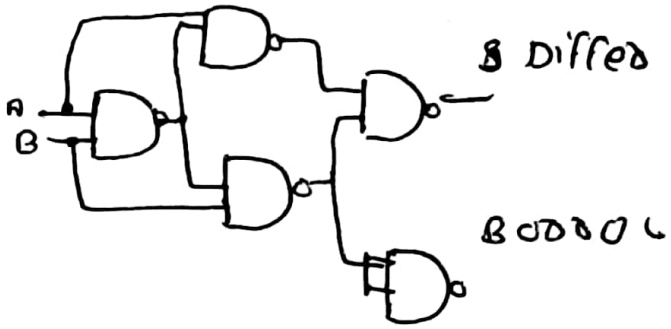
keralanotes.com

KeralaNotes

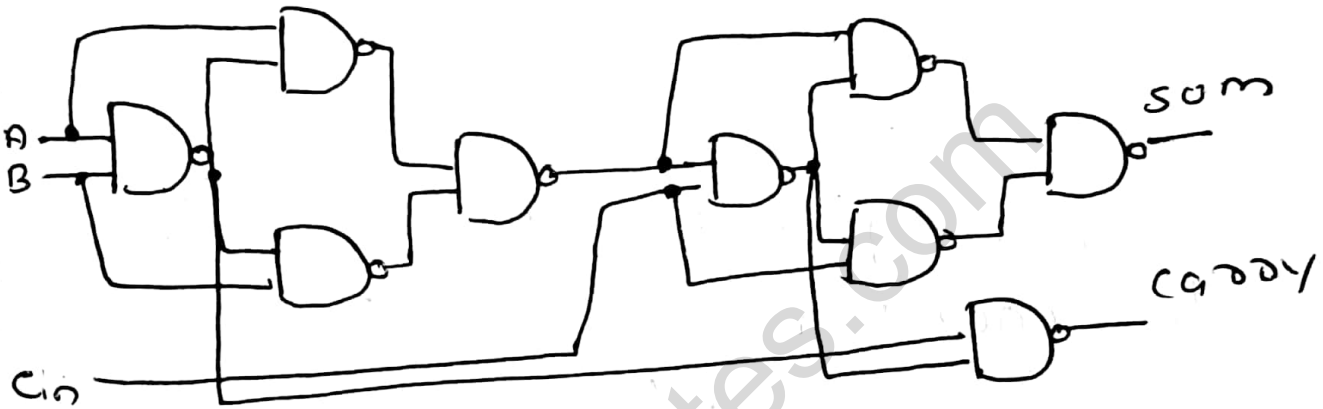
Half adder using NAND



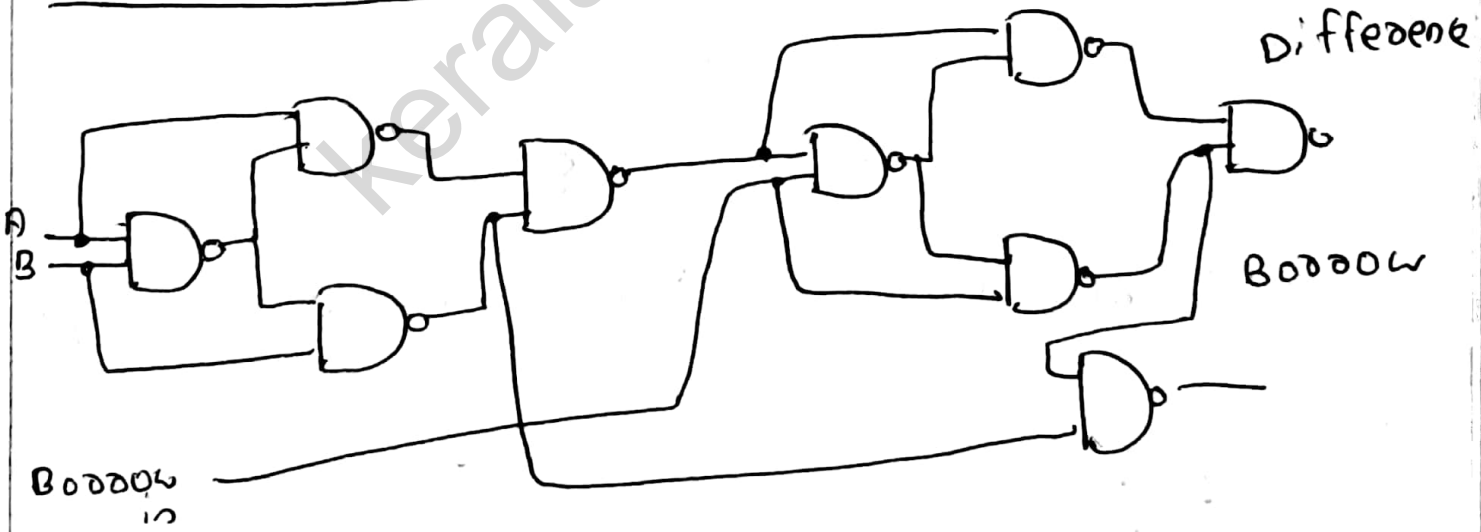
Half Subtractor



Full Adder



Full subtractor



keralanotes.com

For More Study Materials : https://www.keralanotes.com/

KeralaNotes

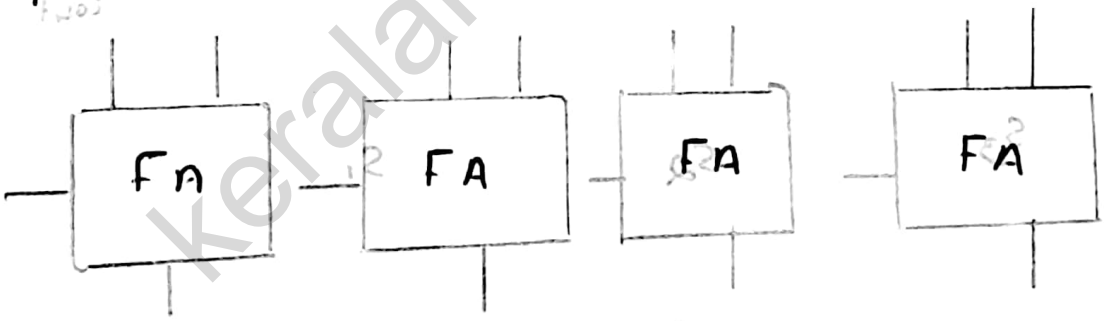
Combinational logic circuit

Binary parallel Adder

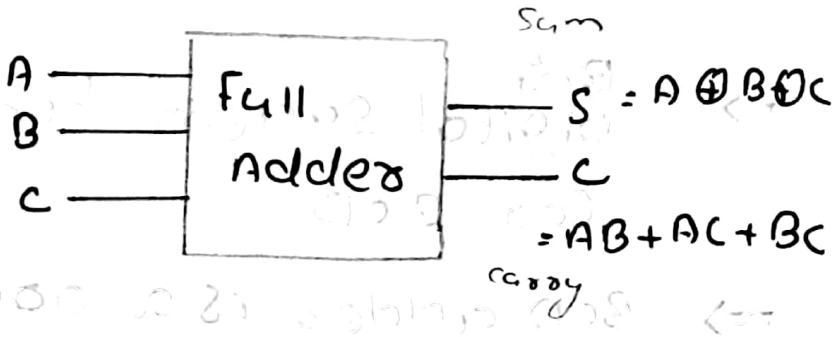
Binary parallel adder is a digital circuit that add two Binary numbers in parallel form and produces the arithmetic sum of those numbers in parallel form

-> It consist of full adders connected in a chain

-> Inter connection of full adder circuit provide 4 bit parallel adder



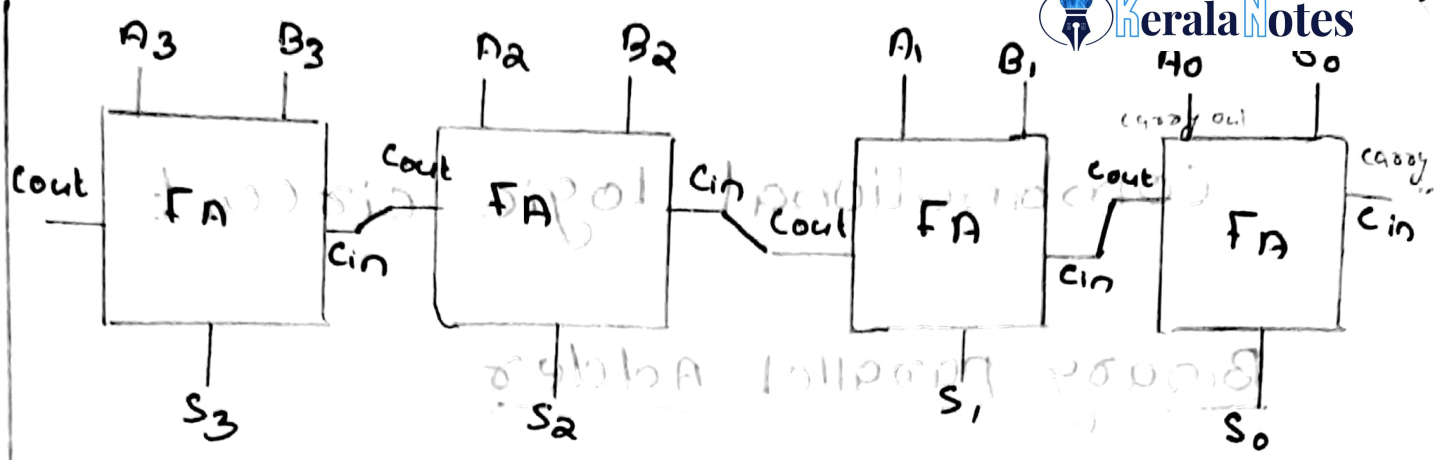
Full adder



-> The carry o/p of each adder is connected to the carry i/p of the next higher ordered adder

For More Study Materials : https://www.keralanotes.com/

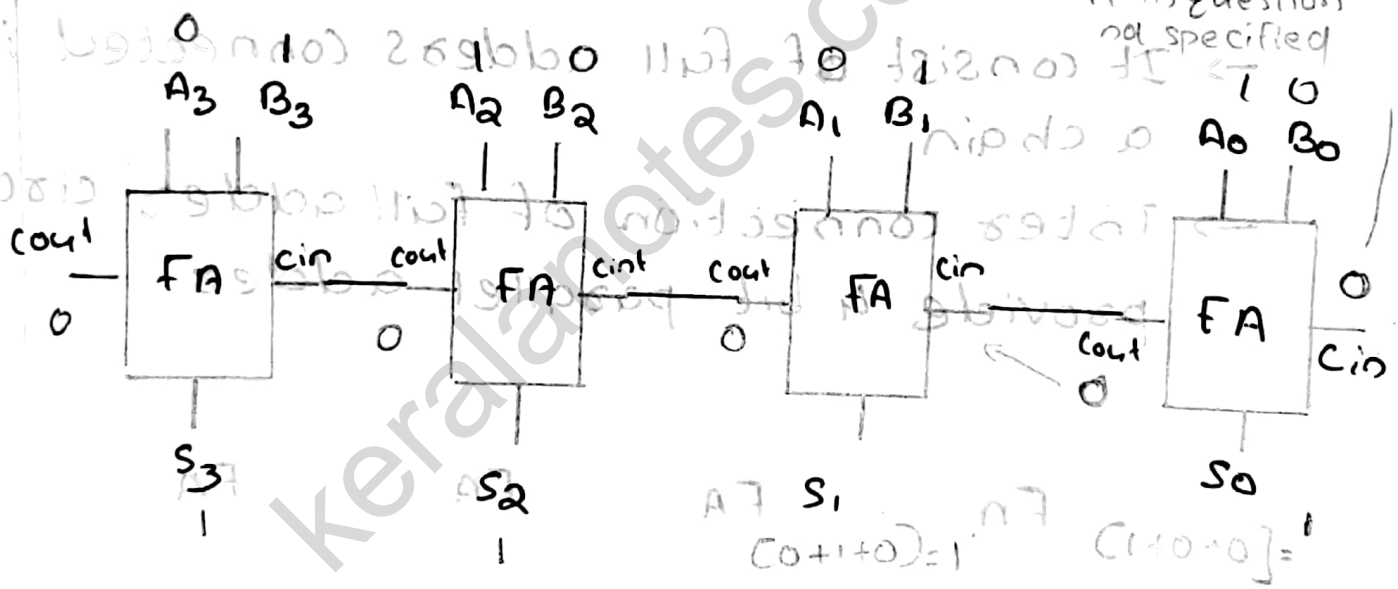
KeralaNotes



Eg:

	(MSB)			(LSB)
A	A_3 0	B_2 1	A_1 0	A_0 1
B	1 B_3	0 B_2	1 B_1	0 B_0
	1	1	1	1

initial will be 0 if in question not specified



BCD Adders

⇒ ~~Bid~~ Digital system handle number in for BCD

⇒ BCD adder is a circuit adds ~~to~~ two BCD ~~digital~~ digits and produces a sum digit also in BCD

keralanotes.com

For More Study Materials : https://www.keralanotes.com/

KeralaNotes

S_3	S_2	S_1	S_0	Y
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	0
0	1	0	1	0
0	1	1	0	0
0	1	1	1	0
1	0	0	0	0
1	0	0	1	0
1	0	1	0	1
1	0	1	1	1
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1

$\frac{6+}{13}$ 0001 0011

9 1001

$S_3 S_2$ \ $S_1 S_0$	$\bar{S}_1\bar{S}_0$ 00	$\bar{S}_1 S_0$ 01	$S_1 S_0$ 11	$S_1\bar{S}_0$ 10
$\bar{S}_3\bar{S}_2$ 00	0	0	0	0
$\bar{S}_3 S_2$ 01	0	0	0	0
$S_3 S_2$ 11	1	1	1	1
$S_3\bar{S}_2$ 10	0	0	1	1

$Y = S_3 S_2 + S_3 S_1$

To implement BCD Adders

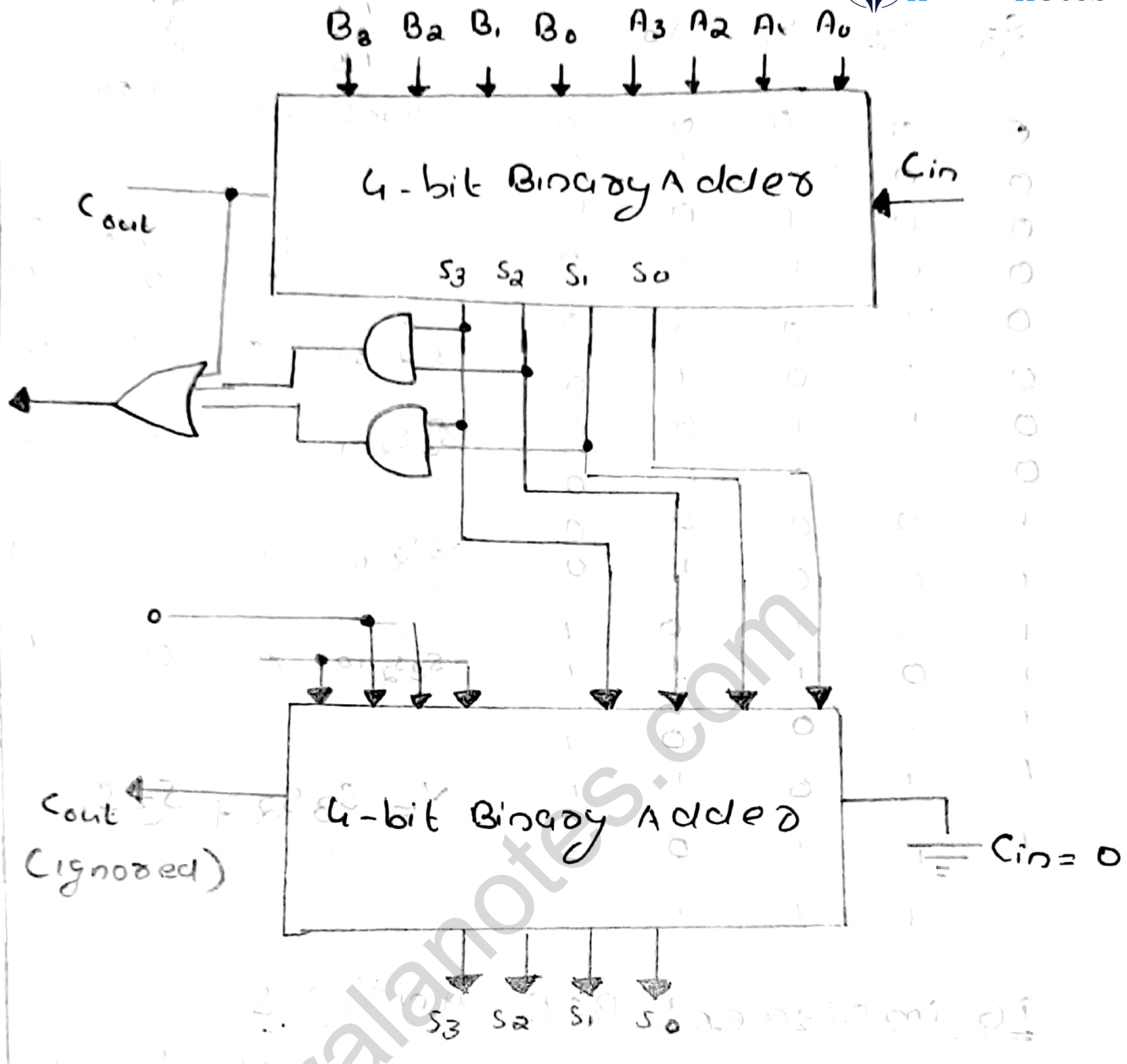
1. 4-Bit Binary Adder for initial addition
2. Logical circuit to detect sum generator than 9
3. one more 4 bit adders to add 0110 in the sum is greater than 9 or carry is 1

- sum > 9 → invalid
  sum < 9, carry
  to valid add 0110 (6)

keralanotes.com

For More Study Materials : https://www.keralanotes.com/

KeralaNotes



Block diagram for BCD Adder

Carry Look Ahead Adder

- one method of speeding up the process by eliminating inter stage carry delay is called carry look ahead adder/ fast adder.

- consider a full adder carry look ahead adder

$S = A \oplus B \oplus C$

$C = AB + AC + BC$

keralanotes.com

For More Study Materials : https://www.keralanotes.com/

KeralaNotes

$$S_i = P_i \oplus C_i$$

$$C_{i+1} = G_i + P_i C_i$$

G_i -> carry generate

P_i -> carry propogate

$$C_2 = C_1 + 1$$

$$C_2 = G_1 + P_1 C_1$$

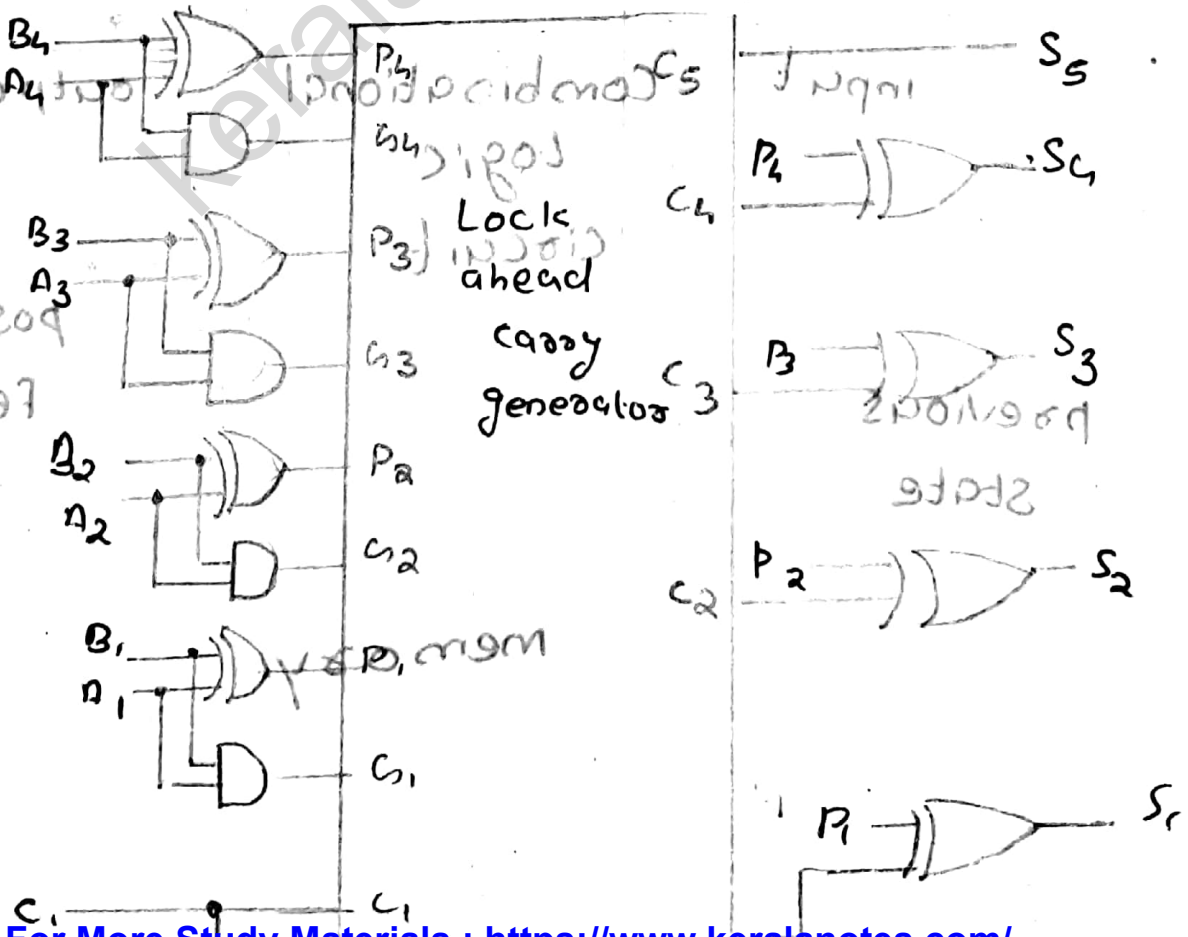
$$C_3 = G_2 + P_2 C_2 = G_2 + P_2 (G_1 + P_1 C_1)$$

$$= G_3 + P_2 G_1 + P_2 P_1 C_1$$

$$C_4 = G_3 + P_3 C_3 = G_3 + P_3 (G_3 + P_2 G_1 + P_2 P_1 C_1)$$

$$= \cancel{G_3 + P_3 G_3 + P_2 G_1 + P_2 P_1} C_1$$

$$= G_3 + P_3 G_3 + P_3 P_2 G_1 + P_3 P_2 P_1 C_1$$



keralanotes.com

For More Study Materials : https://www.keralanotes.com/