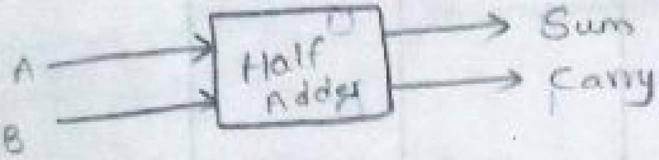
unit-3

Combinational digits Circuits

Half Adder :-

Half Adder is used for adding two bits. It contains Sum and Carry

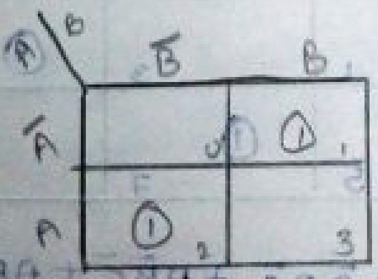
Block diagram



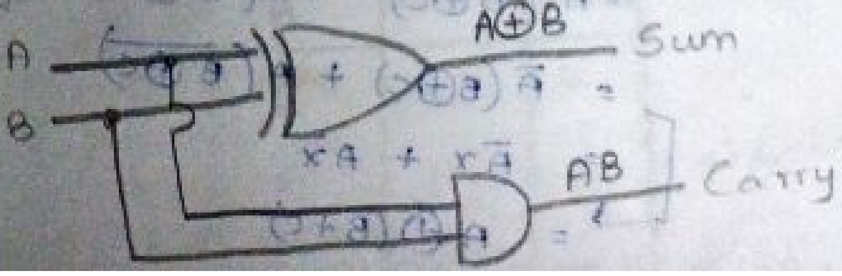
Truth Table

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

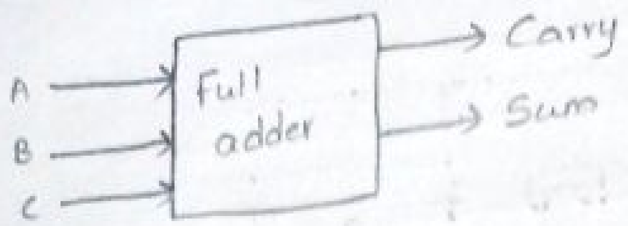
Sum :-



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



Full adder



Truth table

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

K-map Simplification

A \ BC	$\bar{B}\bar{C}$	$\bar{B}C$	BC	$B\bar{C}$
$\bar{A}$	(0)	1 (1)	(3)	1 (2)
A	1 (4)	(5)	1 (7)	(6)

$$\text{Sum} = \bar{A}\bar{B}C + \bar{A}B\bar{C} + A\bar{B}\bar{C} + ABC$$

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

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

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

$$= \bar{A}x + A\bar{x}$$

$$= A \oplus (B + C)$$

$$B \oplus C = \overline{B \odot C}$$

$$B \odot C = \overline{B \oplus C}$$

$$B \oplus C = x$$

$$\overline{B \oplus C} = \bar{x}$$

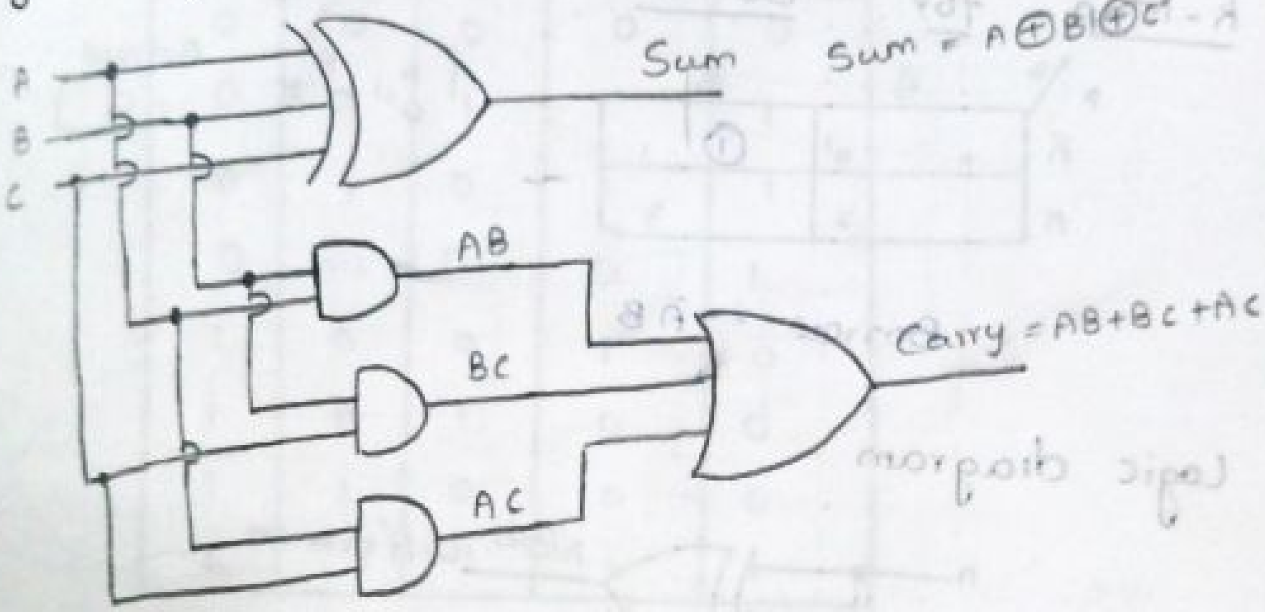
$$\text{Sum} = A \oplus B \oplus C$$

for Carry

A \ BC	$\bar{B}\bar{C}$	$\bar{B}C$	BC	$B\bar{C}$
$\bar{A}$	0	1	1 3	2
A	4	1 5	1 7	1 6

Carry = AB + BC + AC

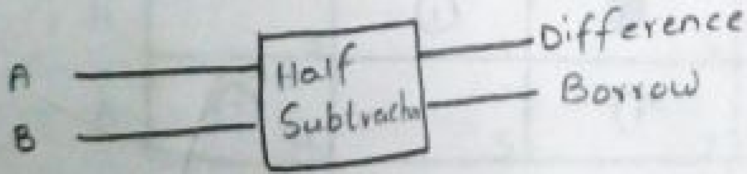
logic diagram



Half Subtractor :-

Half adder is used for subtracting two bits.

It contains ~~sum~~ diff and barrow



Truth table

A	B	Diff	Borrow
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

⇒ 0 − 1 = −1 → Negative

⇒ 1 − 0 = 1

⇒ 1 − 1 = 0

K-Map for Difference

A	B	Diff
0_{2^1}	1_{2^0}	1
1_{2^1}	0_{2^0}	1

A \ B	$\bar{B}$	B
$\bar{A}$	$_0$	1 $_1$
A	1 $_2$	$_3$

$$\text{Difference} = A\bar{B} + \bar{A}B$$
$$= A \oplus B$$

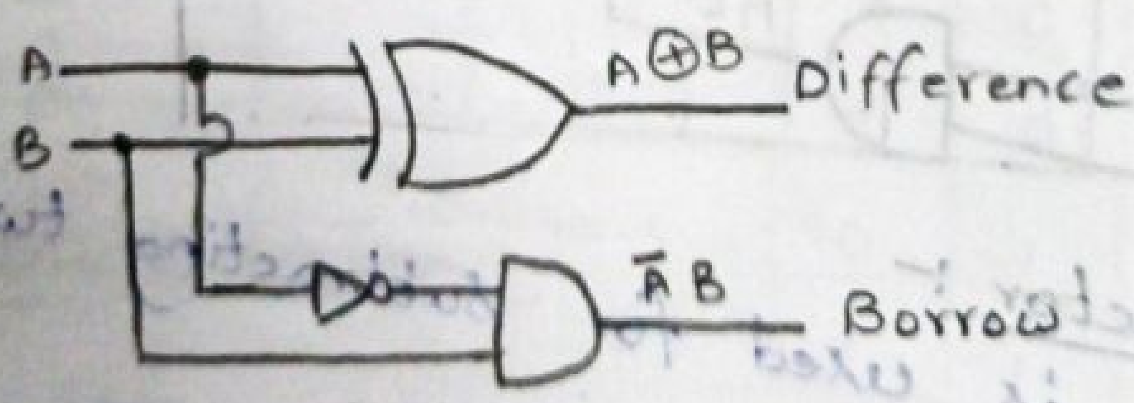
K-Map for Borrow

A	B	Borrow
0_{2^1}	1_{2^0}	1

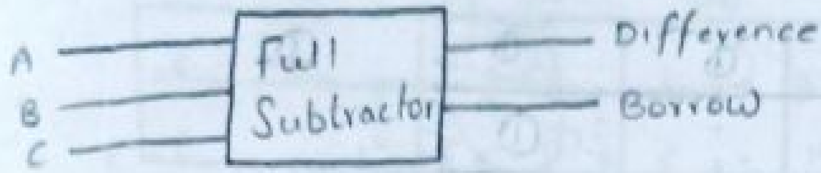
A \ B	$\bar{B}$	B
$\bar{A}$	$_0$	1 $_1$
A	$_2$	$_3$

$$\text{Borrow} = \bar{A}B$$

Logic diagram



Full Subtractor



Truth table

A	B	C	Diff	Borrow
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

K-map Simplification for Difference

A \ BC	$\bar{B}\bar{C}$	$\bar{B}C$	BC	$B\bar{C}$
$\bar{A}$	0	1 (1)	1 (3)	1 (2)
A	1 (4)	5	1 (7)	6

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

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

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

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

$= B\oplus(A\oplus C)$

$= A\oplus B \quad B\oplus(A+C)$

$= A\oplus B\oplus C$

$A\bar{C}+\bar{A}C = A\oplus C$

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

$A\odot C = \overline{A\oplus C}$

$A\oplus C = \overline{A\odot C}$

$A\oplus C = x$

$\overline{A\oplus C} = \bar{x}$

$\bar{B}x + B\bar{x}$

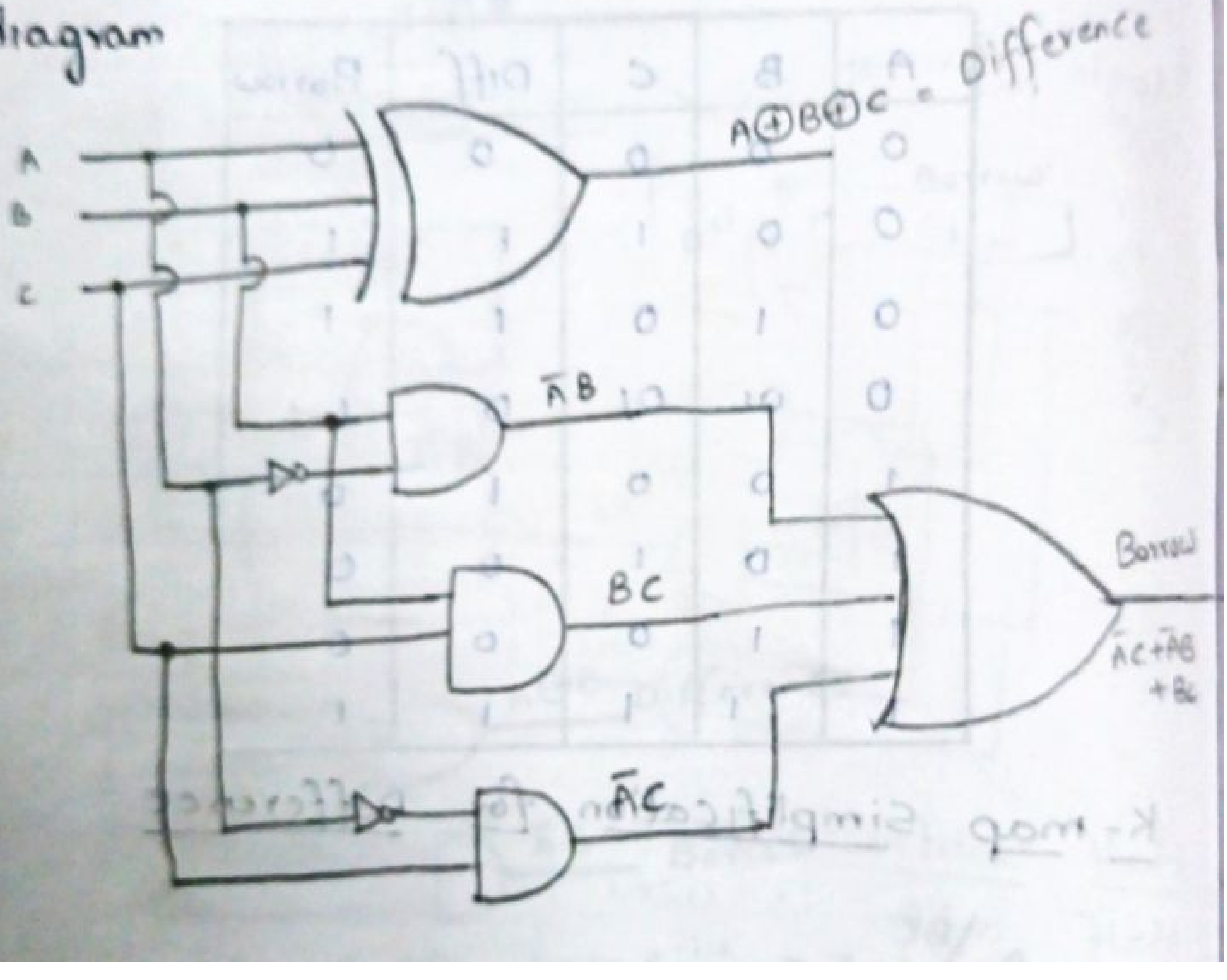
$= B\oplus x$

K-Map for Borrow

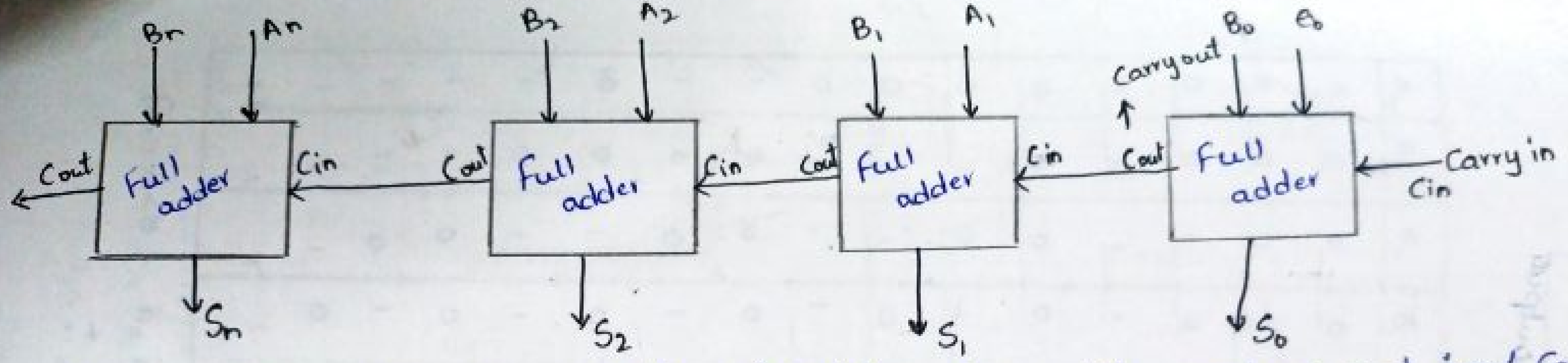
A \ BC	$\bar{B}\bar{C}$	$\bar{B}C$	BC	$B\bar{C}$
$\bar{A}$	0	1 1	1	1 2
A	4	5	1 7	6

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

Logic diagram



parallel binary adder



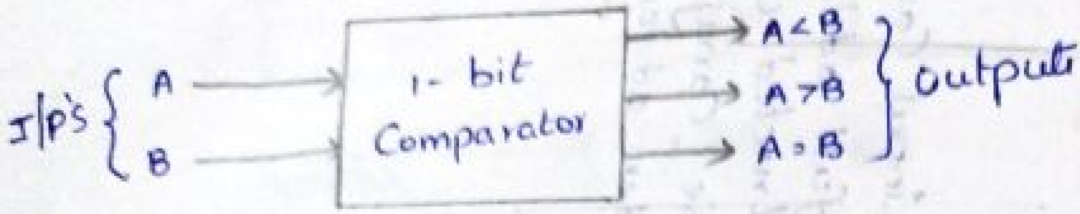
→ Single full adder is capable of adding two 1 bit numbers and input carry. In order to add binary numbers with more than 1 bit additional full adder may be use. For this purpose a 'n' bit parallel adder can be constructed using number of full adder circuit connected in parallel. In this the carry out of each adder is connected to the carry input of the next higher order adder.

If Suppose $n=4$

	A_3	A_2	A_1	A_0
	1	0	0	0
	B_3	B_2	B_1	B_0
	1	1	0	0
Carry ← 1	0	1	0	0

Magnitude Comparator

The Comparator is used for Comparing the number of bit.



A lo Value B lo Value Compare Chayyali Condition denki true ayithe andhulo 1 mugiladhi

A	B	A<B	A>B	A=B
0	0	0	0	1
0	1	1	0	0
1	0	0	1	0
1	1	0	0	1

K-map

for A<B

A	B	A<B
0	1	1

$= \bar{A}B$ (A ki 0 kanuka $\bar{A}$)

for A>B

A	B	A>B
1	0	1

$= A\bar{B}$

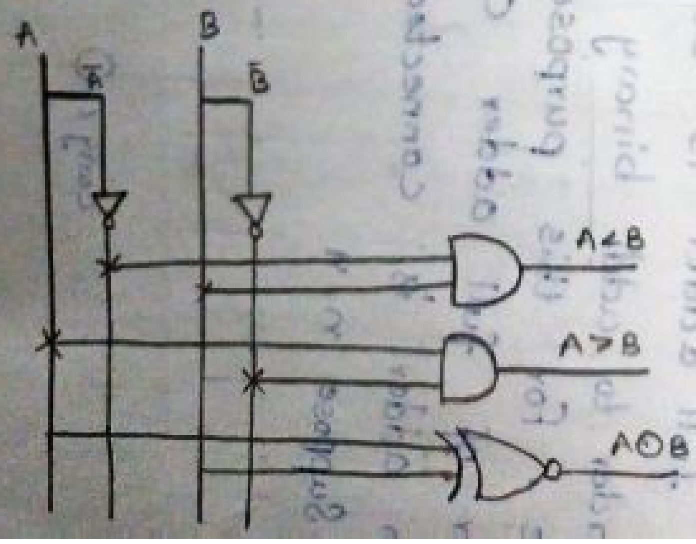
for A=B

A	B	A=B
0	0	1
1	1	1

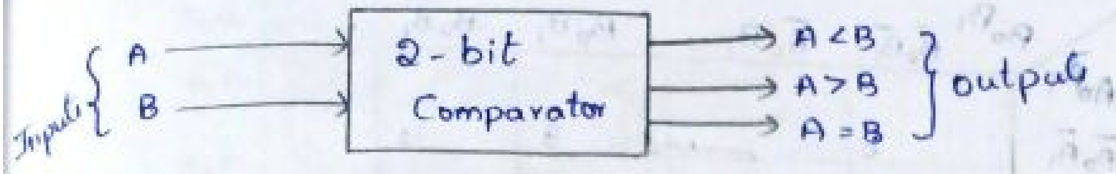
$= \bar{A}\bar{B} + AB$

$= A \odot B$

Logic diagram



2-bit Comparator



A_0	A_1	B_0	B_1	A<B	A>B	A=B
0	0	0	0	0	0	1
0	0	0	1	1	0	0
0	0	1	0	1	0	0
0	0	1	1	1	0	0
0	1	0	0	0	1	0
0	1	0	1	0	0	1
0	1	1	0	1	0	0
0	1	1	1	1	0	0
1	0	0	0	0	1	0
1	0	0	1	0	1	0
10	0	1	0	0	0	1
1	0	1	1	1	0	0
1	1	0	0	0	1	0
1	1	0	1	0	1	0
1	1	1	0	0	1	0
1	1	1	1	0	0	1

0 0 → 0
0 1 → 1
1 0 → 2
1 1 → 3

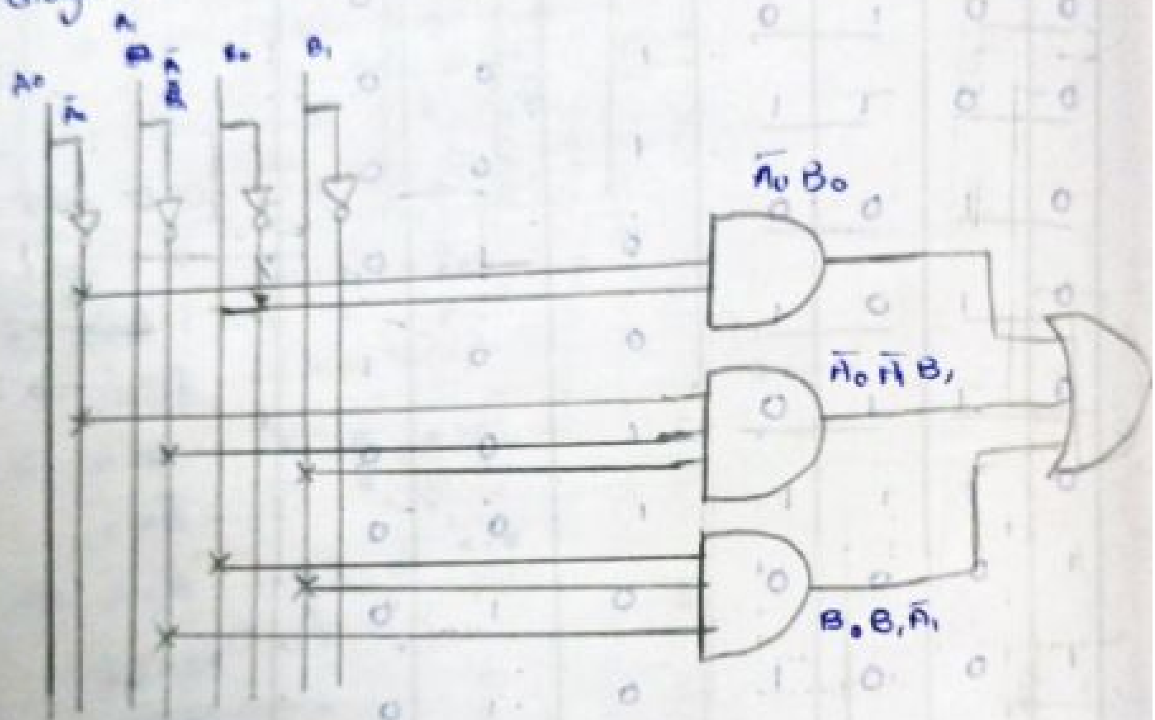
for A<B :- $\bar{A}_0 \bar{A}_1 \bar{B}_0 B_1 + \bar{A}_0 \bar{A}_1 B_0 \bar{B}_1 + \bar{A}_0 \bar{A}_1 B_0 B_1 + \bar{A}_0 A_1 B_0 \bar{B}_1$
$+ \bar{A}_0 A_1 B_0 B_1 + A_0 B \bar{A}_1 B_0 B_1$

K-map for A<B :-

$A_0A_1 \backslash B_0B_1$	$\bar B_0\bar B_1$	$\bar B_0 B_1$	B_0B_1	$B_0\bar B_1$
$\bar A_0\bar A_1$	0	1 1	1 3	1 2
$\bar A_0 A_1$	4	5	1 7	1 6
A_0A_1	12	13	15	14
$A_0\bar A_1$	8	9	1 11	10

$$Y = \bar A_0 B_0 + \bar A_0 \bar A_1 B_1 + B_0 B_1 \bar A_1$$

Logic diagram



for A=B :-

$A_0A_1 \backslash B_0B_1$	$\bar B_0\bar B_1$	$\bar B_0 B_1$	B_0B_1	$B_0\bar B_1$
$\bar A_0\bar A_1$	1 0	1	3	2
$\bar A_0 A_1$	4	1 5	7	6
A_0A_1	12	13	1 15	14
$A_0\bar A_1$	8	9	11	1 10

$$= \bar A_0\bar A_1 \bar B_0\bar B_1 + \bar A_0 A_1 \bar B_0 B_1 + A_0 A_1 B_0 B_1 + A_0 \bar A_1 B_0 \bar B_1$$

$$= \bar A_0 \bar B_0 (\bar A_1 \bar B_1 + A_1 B_1) + A_0 B_0 (A_1 B_1 + \bar A_1 \bar B_1)$$

$$= (\bar A_0 \bar B_0 + A_0 B_0)(\bar A_1 \bar B_1 + A_1 B_1)$$

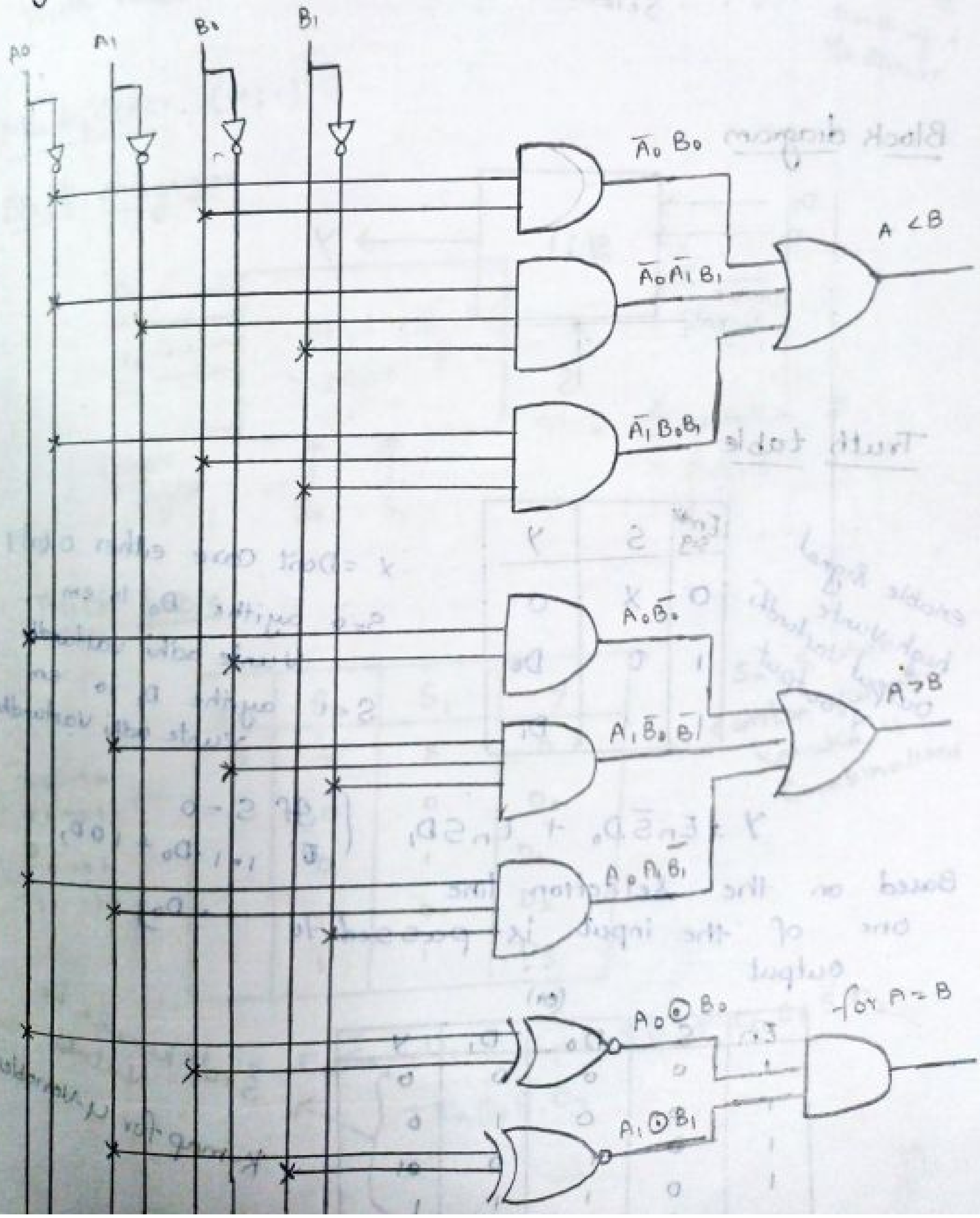
$$= (A_0 \odot B_0)(A_1 \odot B_1)$$

for A > B

$A_0A_1 \backslash B_0B_1$	$\bar{B}_0\bar{B}_1$	$\bar{B}_0B_1$	B_0B_1	$B_0\bar{B}_1$
$\bar{A}_0\bar{A}_1$	0	1	3	2
$\bar{A}_0A_1$	1 (4)	5	7	6
A_0A_1	1 (12)	1 (13)	15	1 (14)
$A_0\bar{A}_1$	1 (8)	1 (9)	11	10

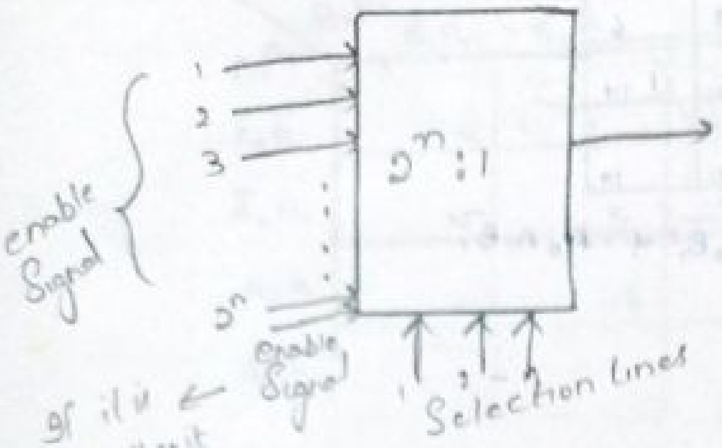
$$Y = A_0\bar{B}_0 + A_1\bar{B}_0\bar{B}_1 + A_0A_1\bar{B}_1$$

Logic diagram



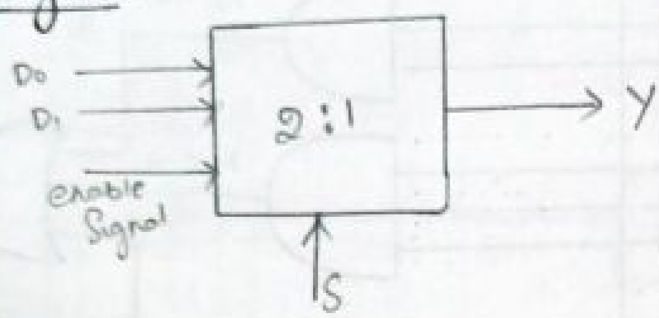
Multiplexer :- (MUX)

(2:1)



If it is high then it result o/p

Block diagram



Truth table

Enable Sig	S	Y
0	X	0
1	0	D_0
1	1	D_1

enable Signal high vunte output vastundhi Y-output

X = Don't care either 0 (or) 1

S=0 ayithe D_0 lo em vunte adhi vastundhi

S=1 ayithe D_1 lo em vunte adhi vastundhi

$$Y = E_n\bar{S}D_0 + E_nSD_1$$

[If S=0
$1\cdot1\cdot D_0 + 1\,0\,D_1$
$= D_0$]

Based on the selection line one of the input is passed to output

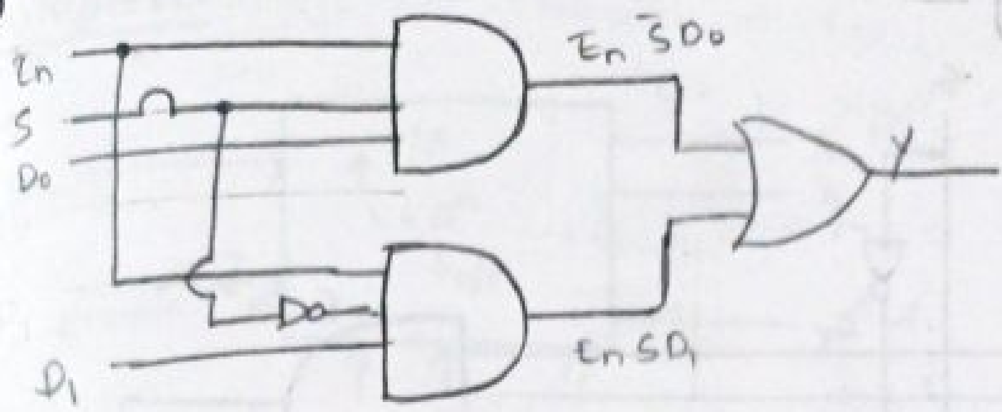
(or)

En	S	D_0	D_1	Y
1	0	0	0	0
1	0	0	1	0
1	0	1	0	1
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

S=0 kanuka D_0 Value

K-map for 4-Variable

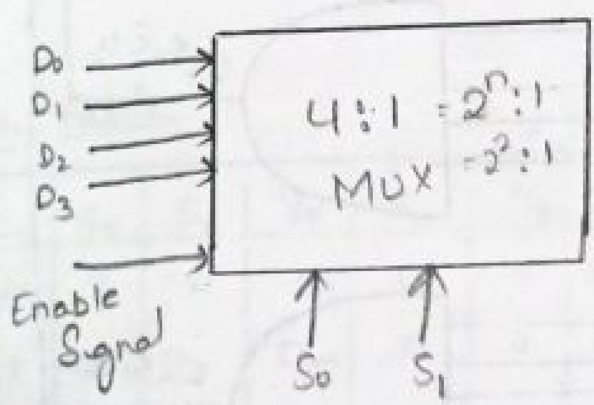
Logic diagram



Truth Table

Multiplexer (4:1)

Block diagram



n=2 Kanuka 2 Signals

Truth table

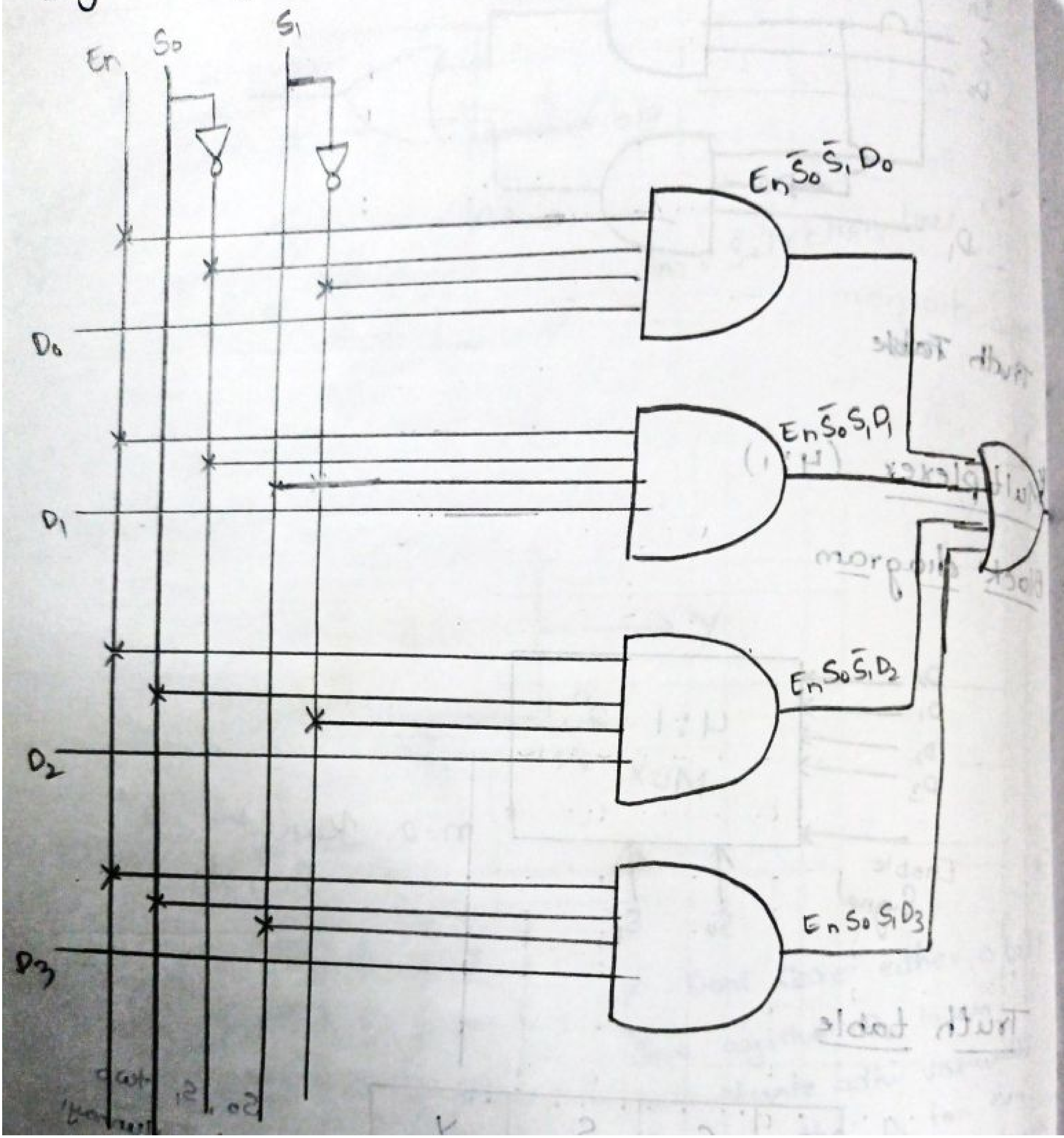
En	S_0	S_1	Y
0	x	x	0
1	0	0	D_0
1	0	1	D_1
1	1	0	D_2
1	1	1	D_3

00 → 0
01 → 1
10 → 2
11 → 3

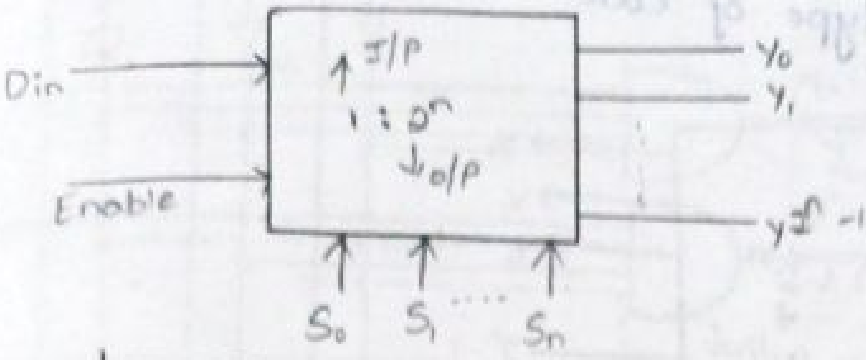
S_0, S_1 two Combin vunnayi Kanuka 4 Combinations

$$Y = E_n \bar{S}_0 \bar{S}_1 D_0 + E_n \bar{S}_0 S_1 D_1 + E_n S_0 \bar{S}_1 D_2 + E_n S_0 S_1 D_3$$

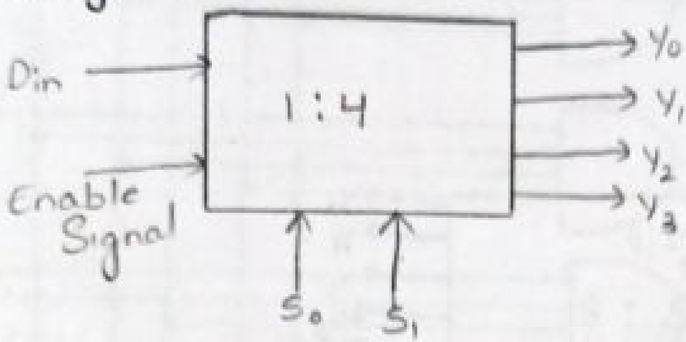
Logic diagram



Demultiplexer :-



Block diagram



Truth table

En	S_0	S_1	Y_0	Y_1	Y_2	Y_3
0	X	X	0	0	0	0
1	0	0	Din	0	0	0
1	0	1	0	Din	0	0
1	1	0	0	0	Din	0
1	1	1	0	0	0	Din

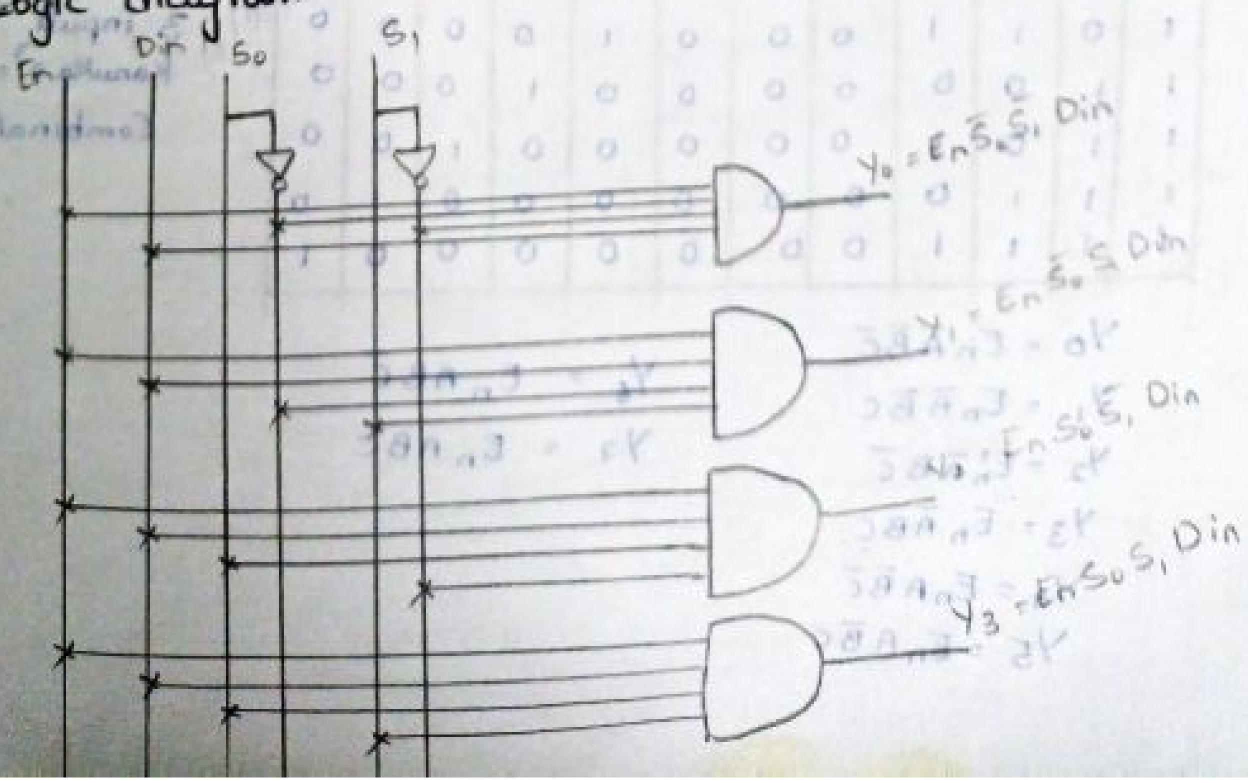
$Y_0 = E_n \bar{S}_0 \bar{S}_1 D_{in}$

$Y_1 = E_n \bar{S}_0 S_1 D_{in}$

$Y_2 = E_n S_0 \bar{S}_1 D_{in}$

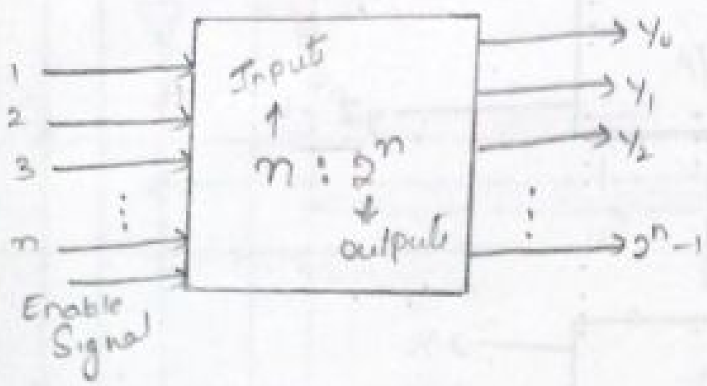
$Y_3 = E_n S_0 S_1 D_{in}$

Logic diagram

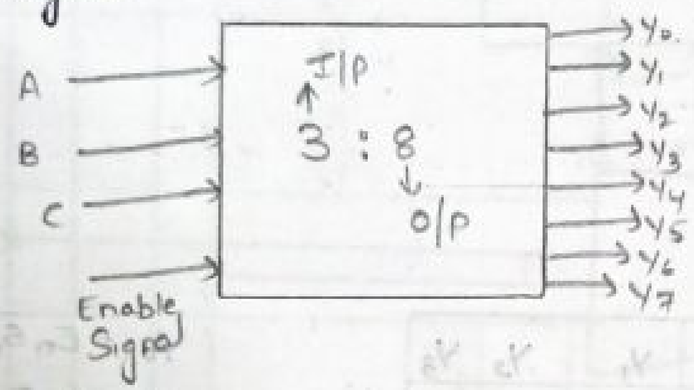


Decoder :-

Decoder are used to Convert one type of Code to another type of code.



Block diagram



Truth table

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

000 → 0
001 → 1
010 → 2
⋮
111 → 7

3 inputs Kanuka $2^3 = 8$ Combinations

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

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

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

$Y_3 = E_n \bar{A} B C$

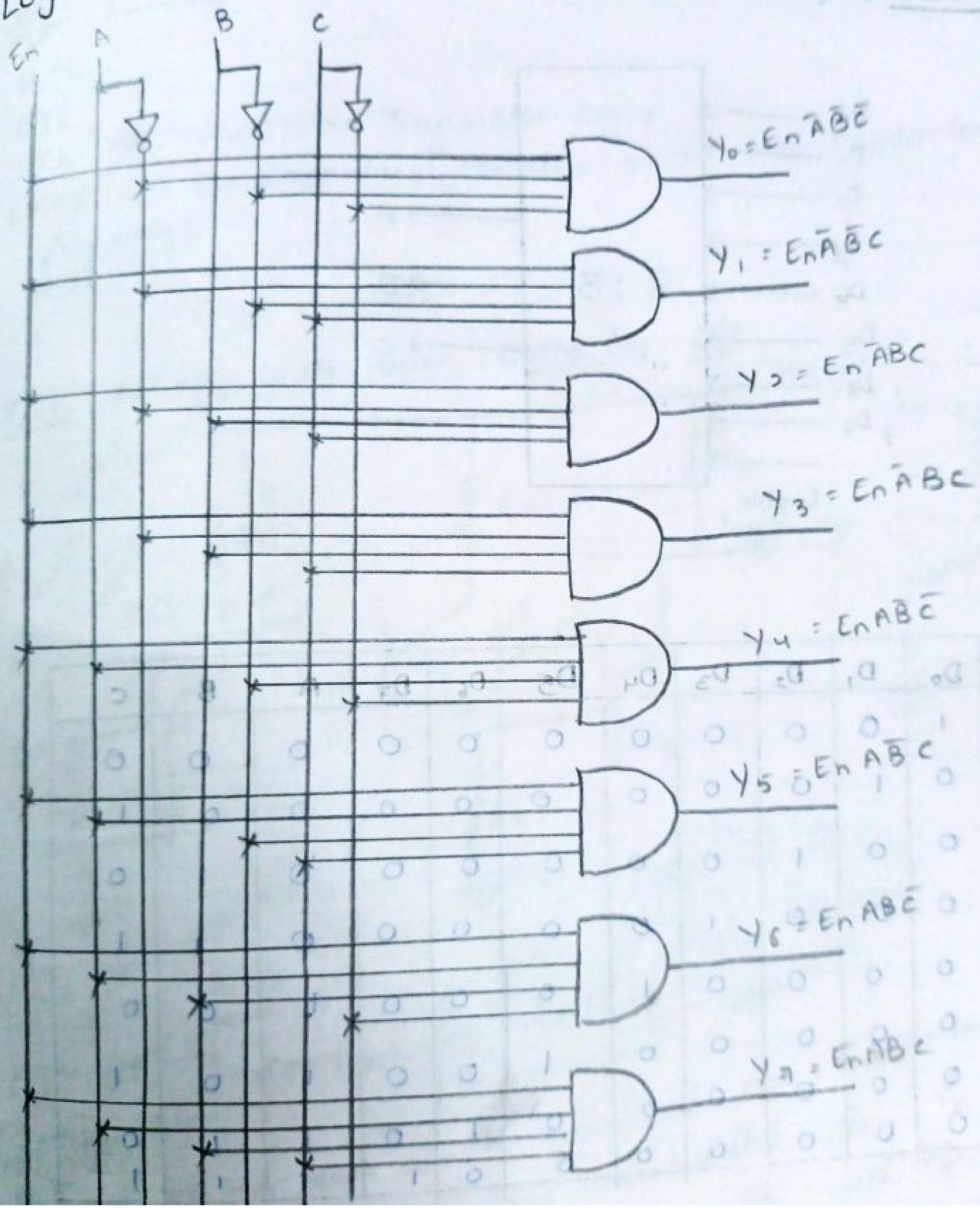
$Y_4 = E_n A \bar{B} \bar{C}$

$Y_5 = E_n A \bar{B} C$

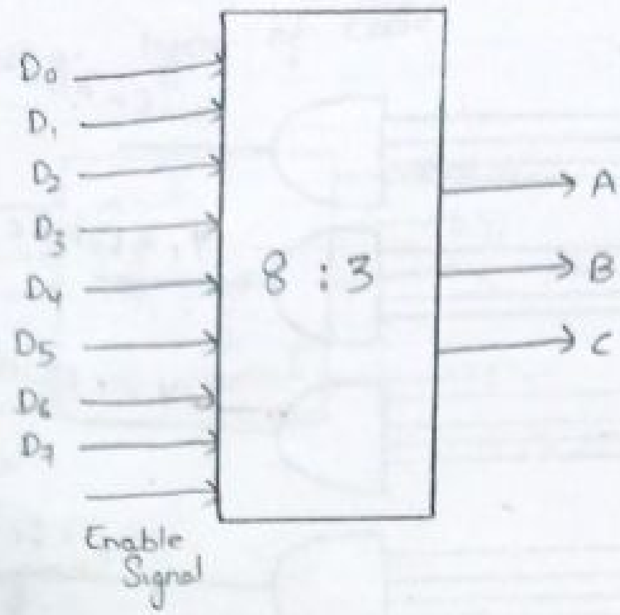
$Y_6 = E_n A B \bar{C}$

$Y_7 = E_n A B C$

Logic diagram



Encoder :- (8 to 3)



$D_0 \rightarrow 000$
$D_1 \rightarrow 001$

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

$$A = D_4 + D_5 + D_6 + D_7$$

$$B = D_2 + D_3 + D_6 + D_7$$

$$C = D_1 + D_3 + D_5 + D_7$$