

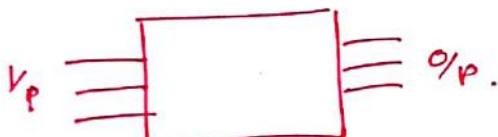
Digital Electronics

Combinational
Circuit

Sequential circuit

→ O/p is only depending on present V_p .

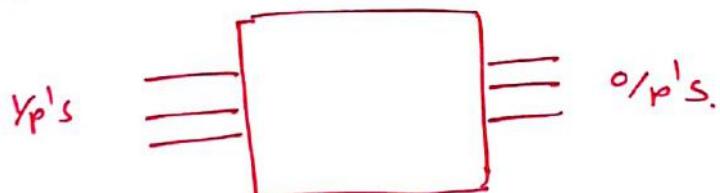
→ O/p is depending on present V_p and past O/p.



- No feedback
- No memory

- Eg
- HA/FA
- HS/FS
- Mux/Demux
- Encoder/Decoder
- Code Converter.

- Designing of combinational Circuits.



Step 1 - Determine & Define total V_p 's and total O/p's of the circuit.

Step-2 - Make truth table that defines relationship b/w V_p 's and O/p's.

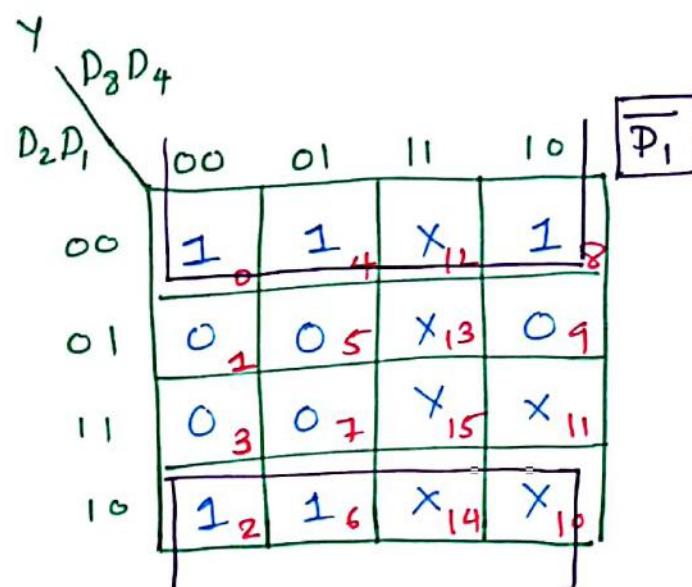
Step-3 - Determine boolean eqⁿ using K-map.

Step-4 - Based on boolean eqⁿ, we can form circuit.

Combinational Circuit designing example 83

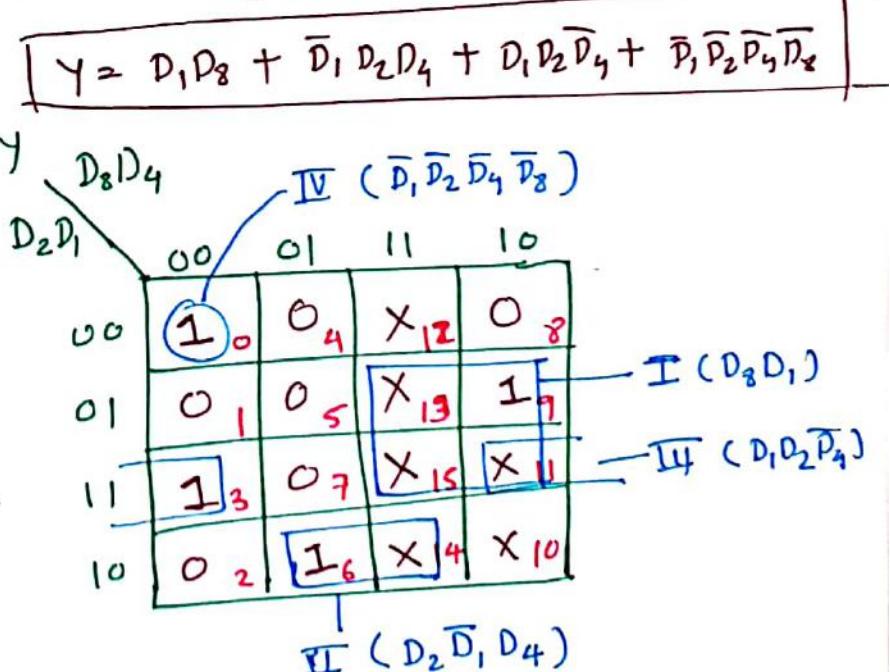
The minimal function that can detect "divisible by 2" with 8421 BCD [$D_8 D_4 D_2 D_1$] is given by $\overline{P_1}$

	$D_8 D_4 D_2 D_1$	Y
0 -	0 0 0 0	1
1 -	0 0 0 1	0
2 -	0 0 1 0	1
3 -	0 0 1 1	0
4 -	0 1 0 0	1
5 -	0 1 0 1	0
6 -	0 1 1 0	1
7 -	0 1 1 1	0
8 -	1 0 0 0	1
9 -	1 0 0 1	0
10 -	1 0 1 0	x
:	:	:
15 -	1 1 1 1	x

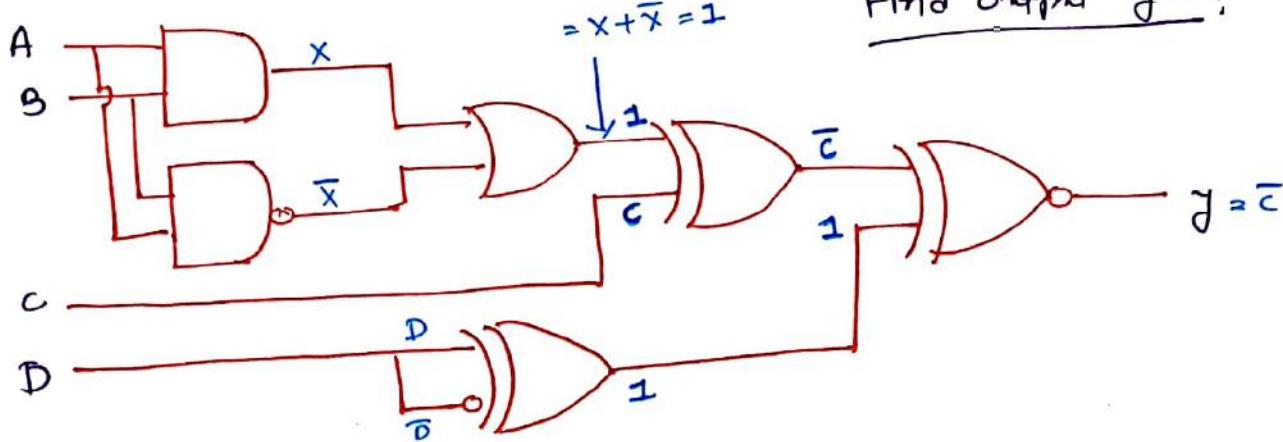


The minimal function that can detect "divisible by 3" with 8421 BCD [$D_8 D_4 D_2 D_1$] is given by _____

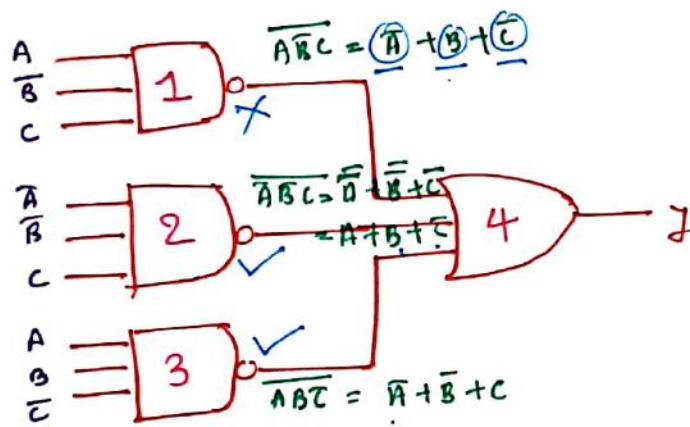
	$D_8 D_4 D_2 D_1$	Y
0 -	0 0 0 0	1
1 -	0 0 0 1	0
2 -	0 0 1 0	0
3 -	0 0 1 1	1
4 -	0 1 0 0	0
5 -	0 1 0 1	0
6 -	0 1 1 0	1
7 -	0 1 1 1	0
8 -	1 0 0 0	0
9 -	1 0 0 1	1
10 -	1 0 1 0	x
:	:	:
15 -	1 1 1 1	x



1)

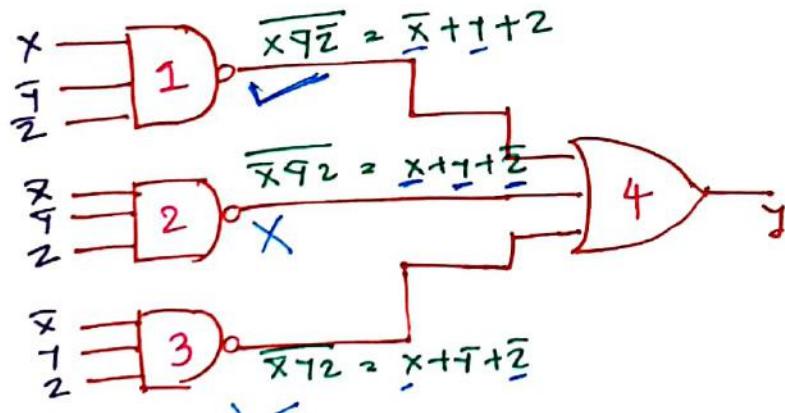
Find output $J = ?$

2)



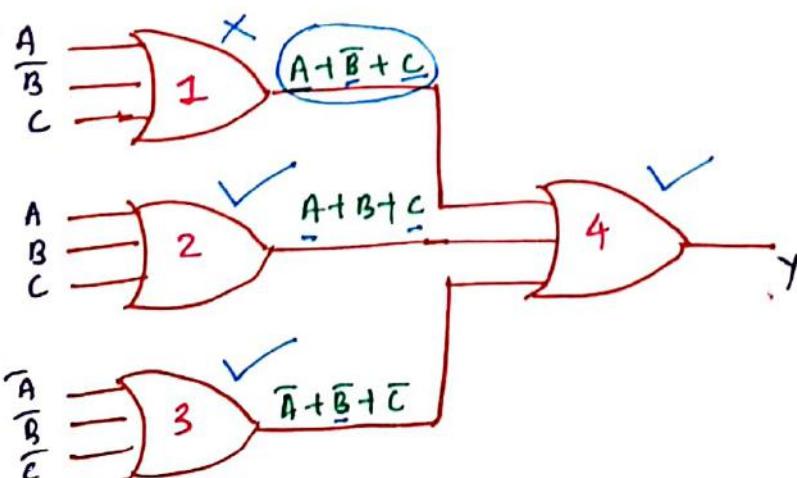
what is redundant
gate in given
combinational circuit?

3)



what is redundant
gate in given
combinational circuit?

4)

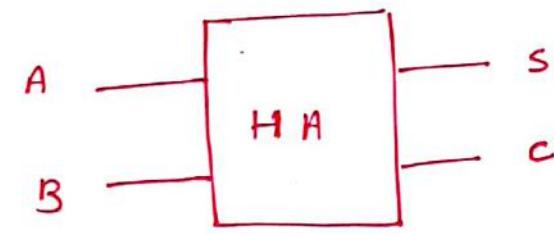


what is redundant
gate in given
combinational
circuit?

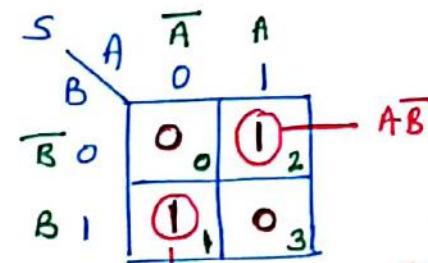
HALF ADDER

86

→ Half Adder we perform two bit addition.

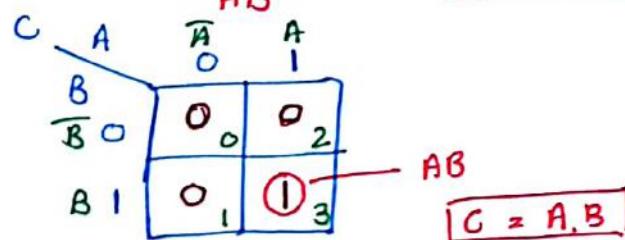


A	B	S	C
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

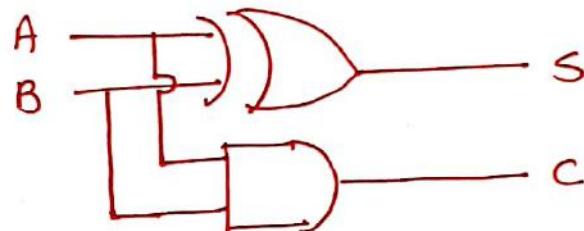


$$S = A\bar{B} + \bar{A}B$$

$$S = A \oplus B$$



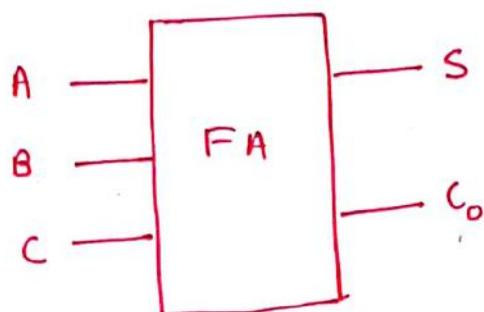
$$C = A \cdot B$$



Full Adder

87

- Full Adder circuit is used to perform 3 bits addition.



A	B	C	S	C_o
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

S	AB	$\bar{A}\bar{B}$	$\bar{A}B$	AB	$A\bar{B}$
C_o	0 0	1 2	0 6	1 4	
C_i	1 1	0 3	1 7	0 5	

Below the table:

- $\bar{A}\bar{B}C$
- $\bar{A}BC$
- ABC
- $A\bar{B}C$

$$\begin{aligned}
 \rightarrow S &= \bar{A}\bar{B}C + \bar{A}BC + ABC + A\bar{B}\bar{C} \\
 &= \bar{A}(\bar{B}C + BC) + A(BC + \bar{B}\bar{C}) \\
 &= \bar{A}(B \oplus C) + A(B \oplus C) \\
 &= \bar{A}(B \oplus C) + A(\bar{B} \oplus C)
 \end{aligned}$$

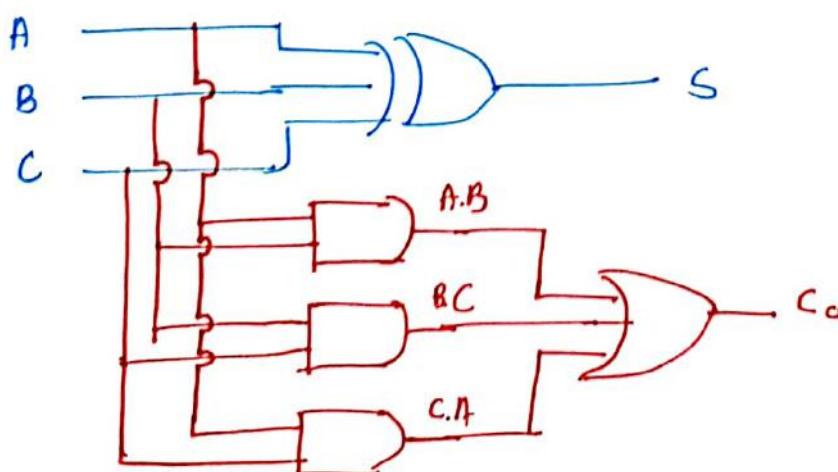
$$S = A \oplus B \oplus C$$

C_o	AB	$\bar{A}\bar{B}$	$\bar{A}B$	AB	$A\bar{B}$
C_o	0 0	0 2	1 6	0 4	
C_i	0 1	1 3	1 7	1 5	

Below the table:

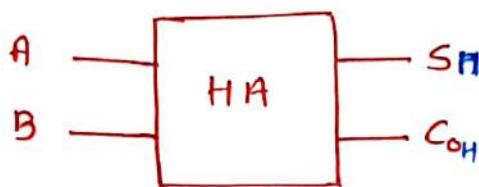
- CB
- AB
- AC

$$C_o = AB + B.C + A.C$$



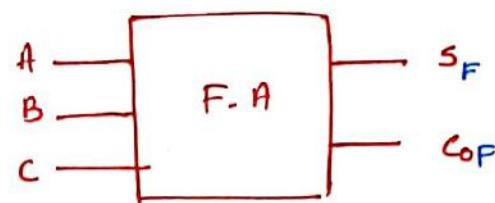
Full Adder circuit using Half Adder.

88



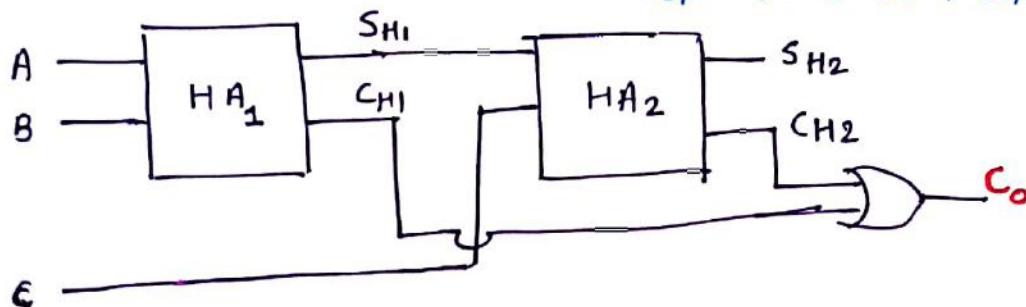
$$S_H = A \oplus B$$

$$C_{OH} = A \cdot B$$



$$S_F = A \oplus B \oplus C$$

$$C_{OF} = AB + BC + CA$$



$$- S_{H_1} = A \oplus B = A\bar{B} + \bar{A}B$$

$$C_{H_1} = A \cdot B$$

$$| \quad S_{H_2} = S_{H_1} \oplus C = A \oplus B \oplus C$$

$$\begin{aligned} C_{H_2} &= S_{H_1} \cdot C \\ &= (A\bar{B} + \bar{A}B) \cdot C \\ &= A\bar{B}C + \bar{A}B \cdot C \end{aligned}$$

$$- C_O = C_{H_2} + C_{H_1}$$

$$\Rightarrow A \cdot B + \overline{ABC} + \overline{ABC}$$

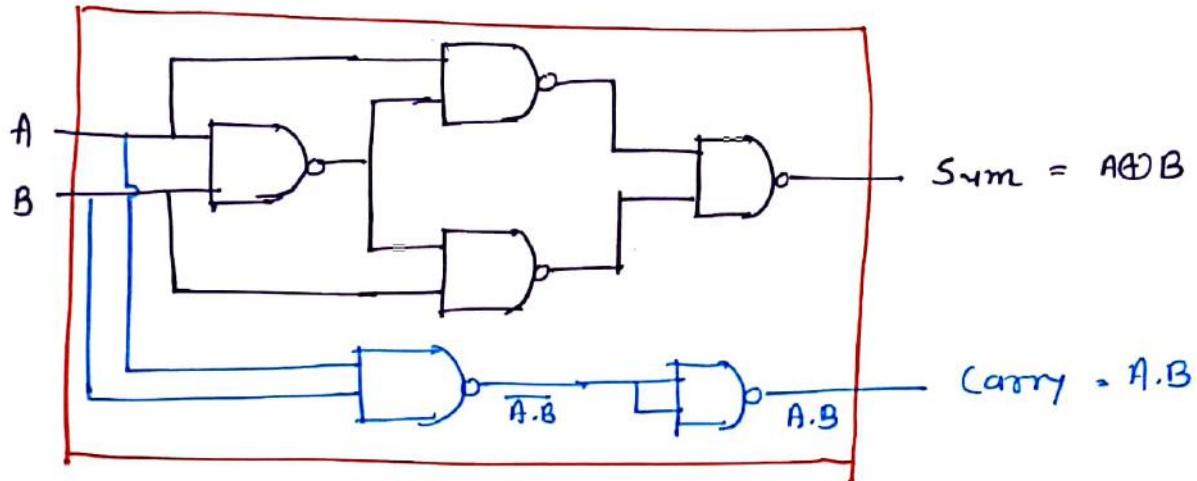
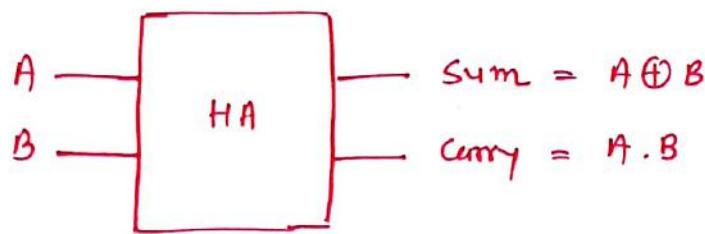
$$\Rightarrow A \cdot B + BC + AC$$

$$C_O$$

	AB	$\bar{A}\bar{B}$	$\bar{A}B$	AB	$A\bar{B}$
C	00	01	11	10	
0					
1	I	I	I	I	I
	CB	AB	AC		

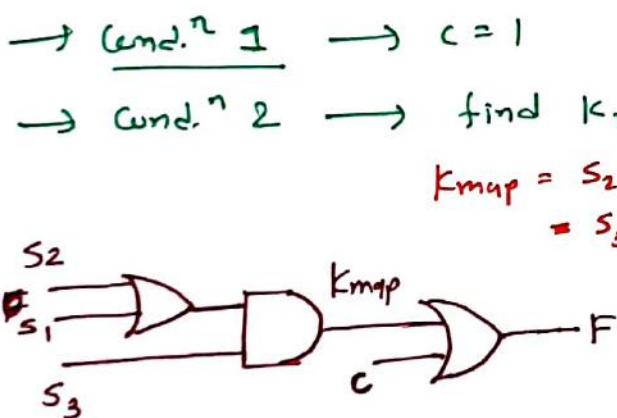
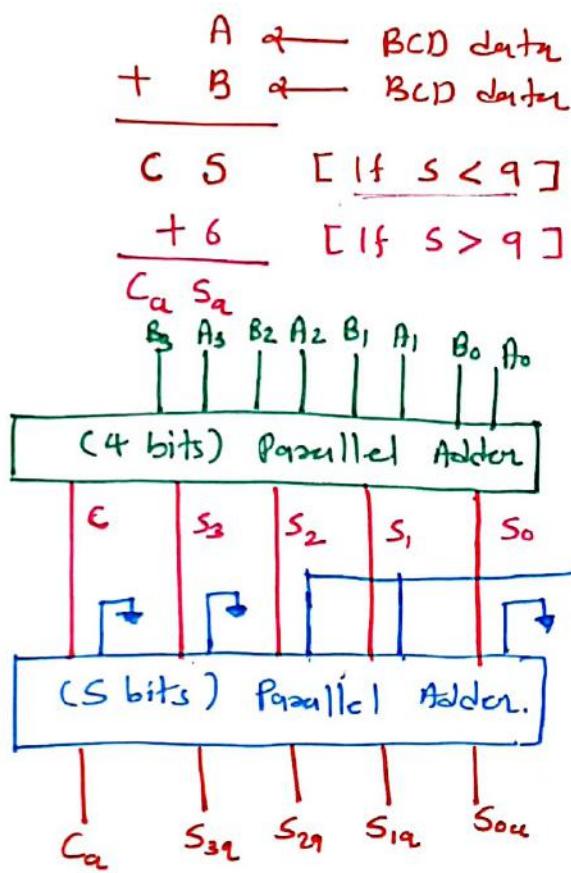
HALF ADDER BY NAND GATES

89



BCD Adder.

91



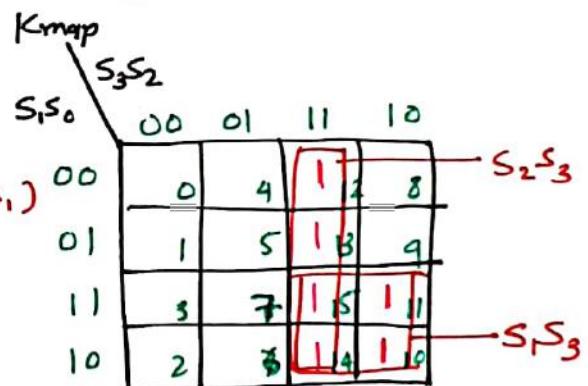
(6 → 00110)

[S < 9]

(5 > 9)
(c = 0)

(S > 9)
(c = 1)

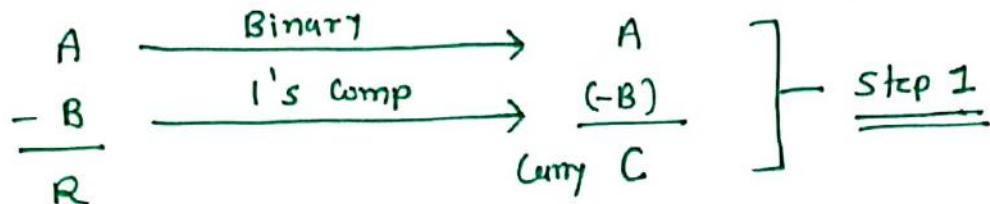
C	S ₃	S ₂	S ₁	S ₀	
0	0	0	0	0	→ 0
0	0	0	0	1	→ 1
0	0	0	1	0	→ 2
0	0	0	1	1	→ 3
0	0	1	0	0	→ 4
0	0	1	0	1	→ 5
0	0	1	1	0	→ 6
0	0	1	1	1	→ 7
0	1	0	0	0	→ 8
0	1	0	0	1	→ 9
0	1	0	1	0	→ 10
0	1	0	1	1	→ 11
0	1	1	0	0	→ 12
0	1	1	0	1	→ 13
0	1	1	1	0	→ 14
0	1	1	1	1	→ 15
1	0	0	0	0	→ 16
1	0	0	0	1	→ 17
1	0	0	1	0	→ 18



S < 9	15 < S < 9	S > 9 (c = 1)
A = 4	A = 5	
B = 4	B = 6	
$\frac{8}{}$	$\frac{11}{}$	
$\frac{01000}{\uparrow\uparrow\uparrow\uparrow\uparrow}$	$\frac{01011}{\uparrow\uparrow\uparrow\uparrow\uparrow}$	
C S ₃ S ₂ S ₁ S ₀	C S ₃ S ₂ S ₁ S ₀	

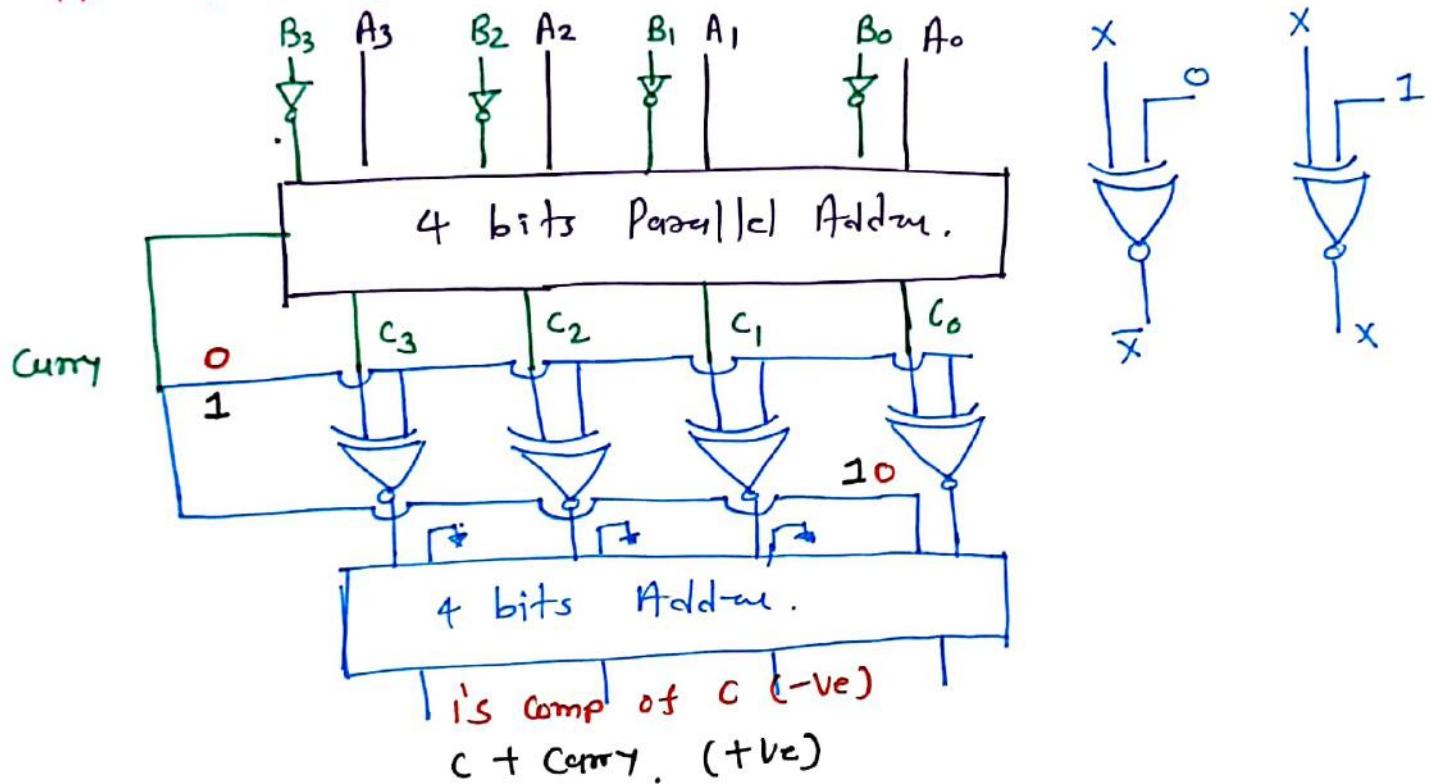
1's Complement Subtraction using parallel adder

92



- If Carry = 1 [Result is +ve] $\Rightarrow R = C + \text{Carry}$

If Carry = 0 [Result is -ve] $\Rightarrow R = 1\text{'s comp. of } C.$



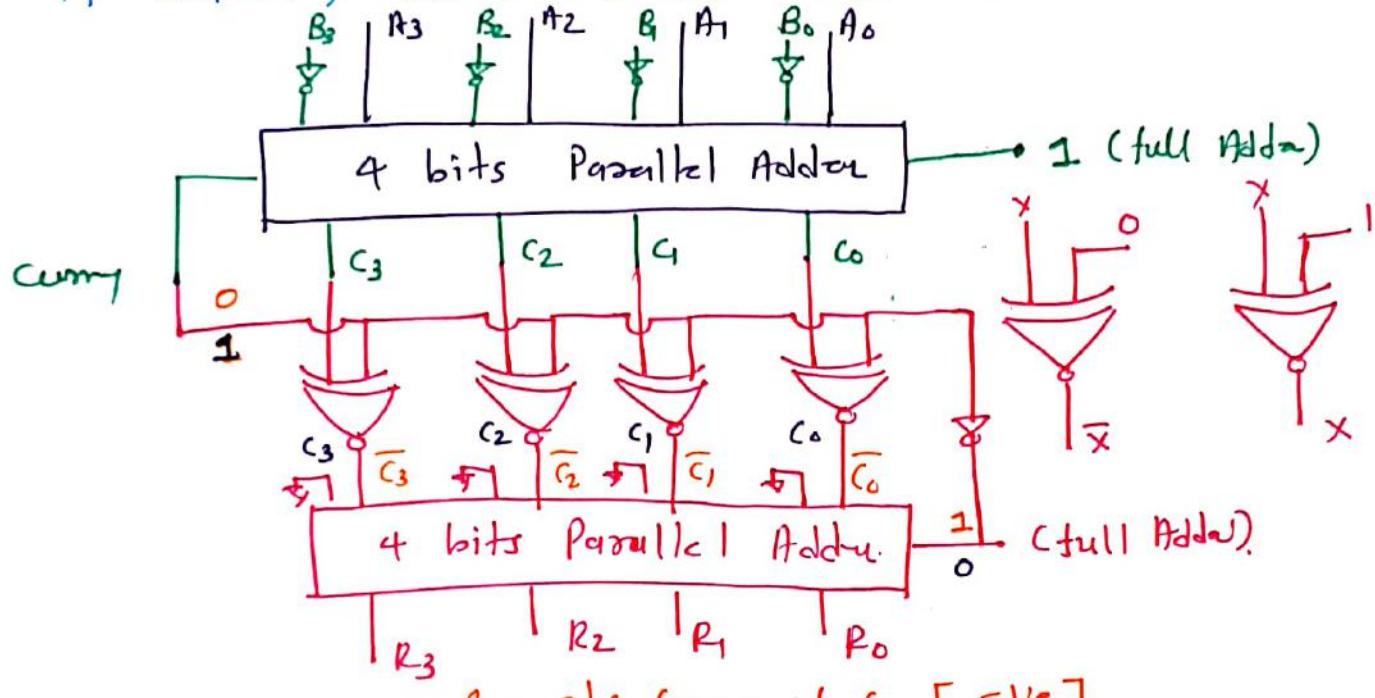
2^1 's Complement Subtraction using Parallel Adder

$$\begin{array}{r}
 A \\
 -B \\
 \hline
 R
 \end{array}
 \quad \begin{array}{c}
 \xrightarrow{\text{Binary}} A \\
 \xrightarrow{\text{2's Comp.}} -B \\
 \text{Carry } C
 \end{array}
 \quad \boxed{\text{Step 1}}$$

Q3

→ If Carry = 0, Result is (-Ve), $\Rightarrow R = 2^1$'s Comp. of C.] Step 2

If Carry = 1, Result is (+Ve), $\Rightarrow R = S$



$R = 2^1$'s Comp. of C [-Ve].

$R = C [+Ve]$.

Half Subtractor

→ Half subtractor is performing two bits subtraction.

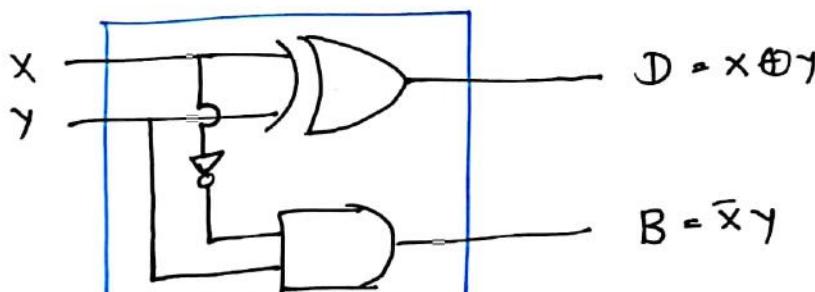
		X	- Y	B D
		(X-Y)		
		X	Y	B D
0	0	0	0	0 0
0	1	1	1	1 1
1	0	1	0	1 0
1	1	0	0	0 0

D	X	\bar{X}	X
Y	0	0	1
Y	1	1	0

$$D = X\bar{Y} + \bar{X}Y \\ = X \oplus Y$$

B	X	\bar{X}	X
Y	0	0	1
Y	1	1	0

$$B = \bar{X}Y \quad \begin{array}{l} \text{for } Y-X \\ B = X\bar{Y} \end{array}$$

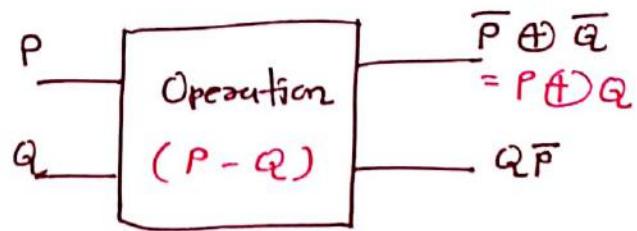
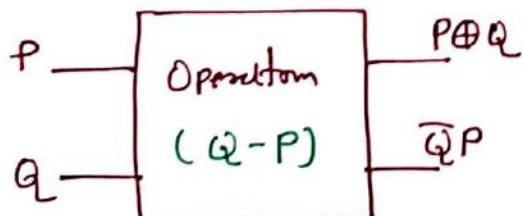


Half Adder

$$S = X \oplus Y$$

$$C = X \bar{Y}$$

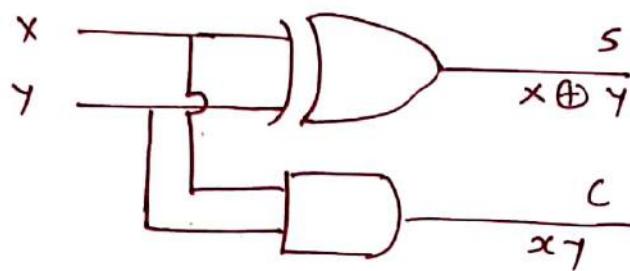
Half Sub.



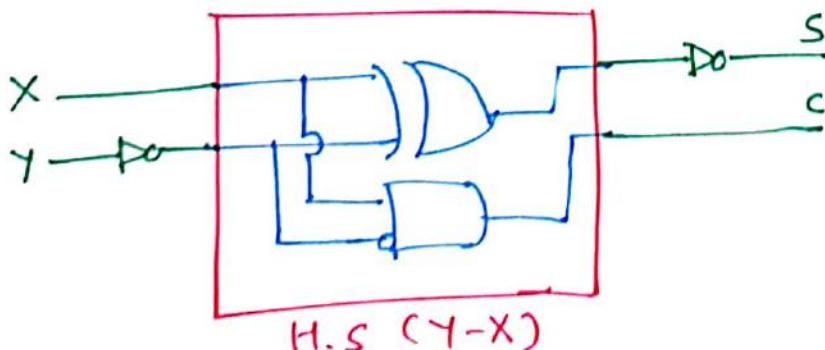
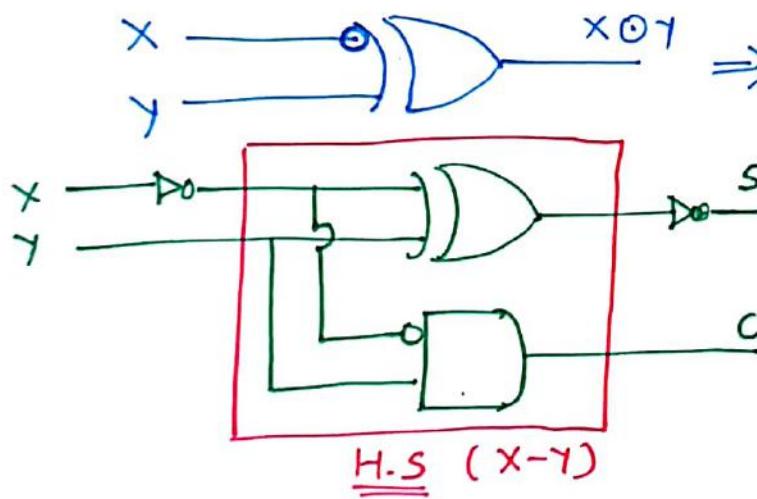
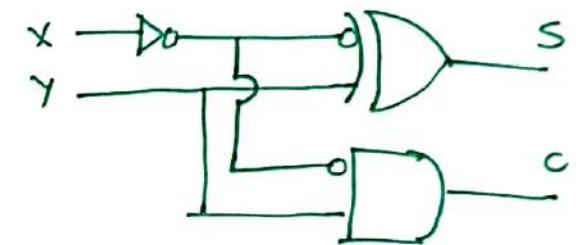
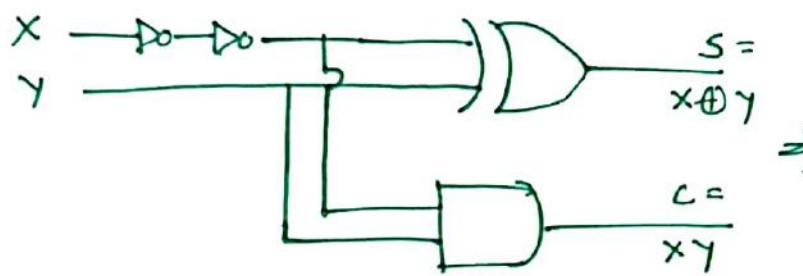
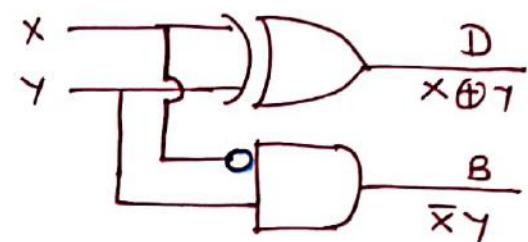
$$\begin{aligned} \bar{P} \oplus \bar{Q} &= \bar{P}\bar{Q} + \bar{\bar{P}}\bar{Q} \\ &= \bar{P}Q + P\bar{Q} \\ &= P \oplus Q \end{aligned}$$

Half Adder using Half Subtractor 95

→ Half Adder



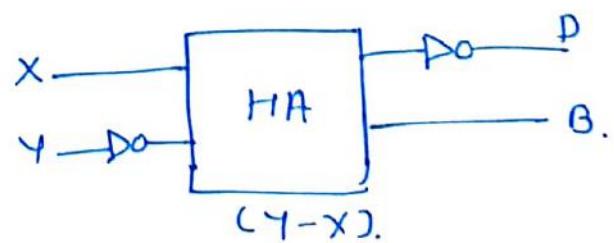
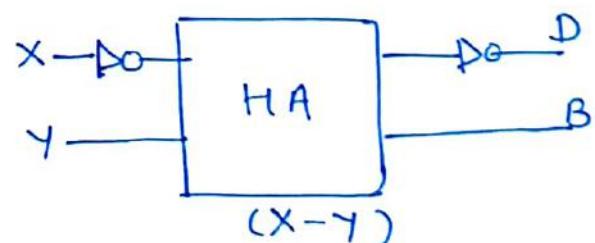
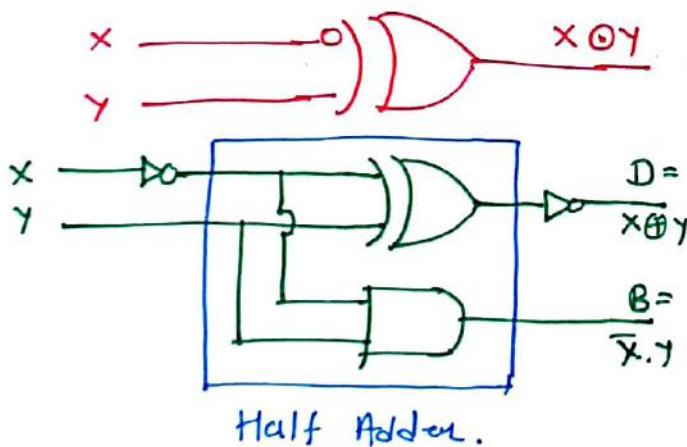
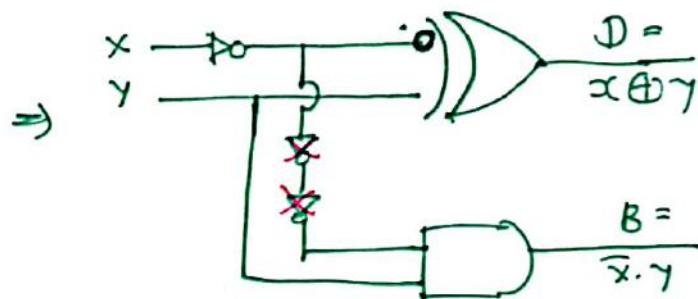
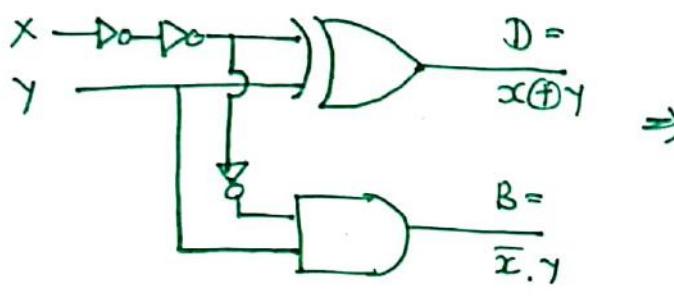
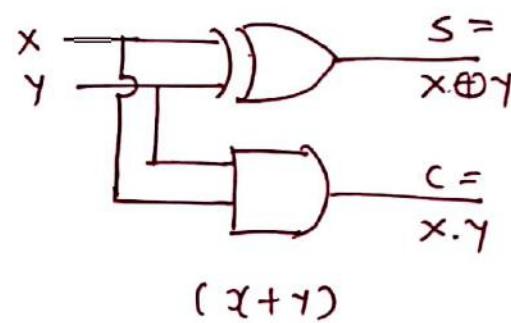
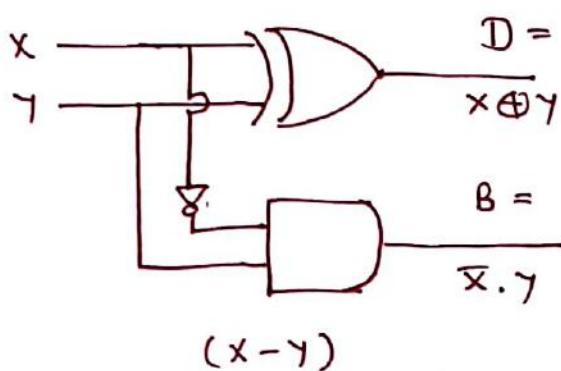
→ Half Subtractor



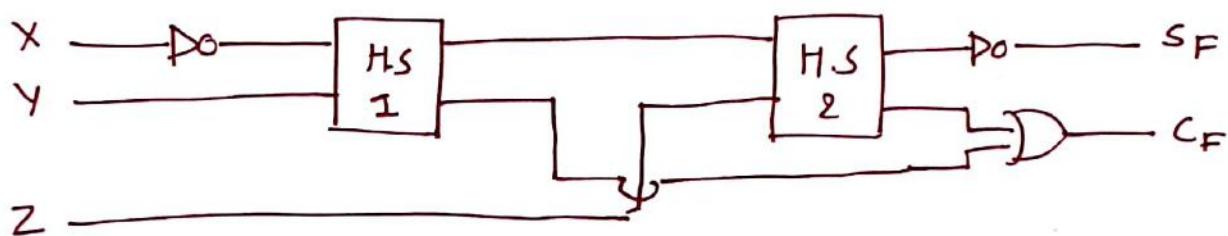
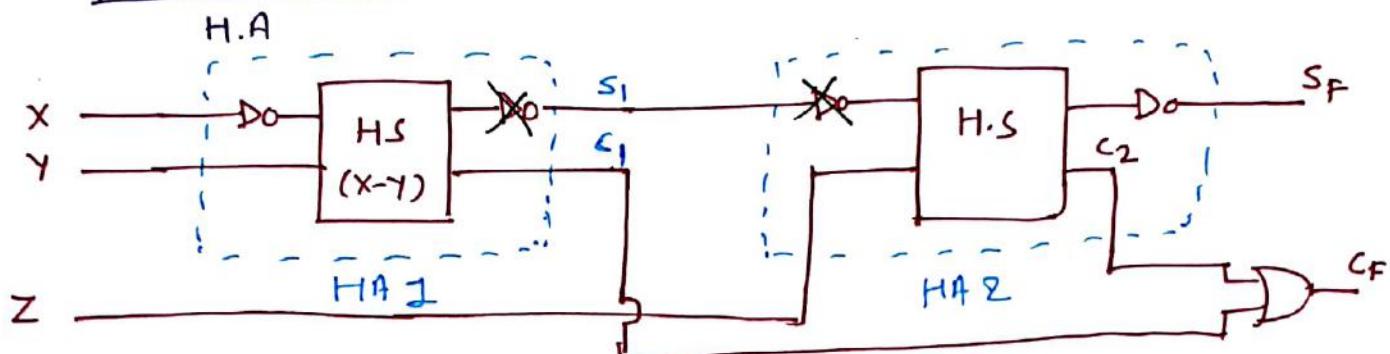
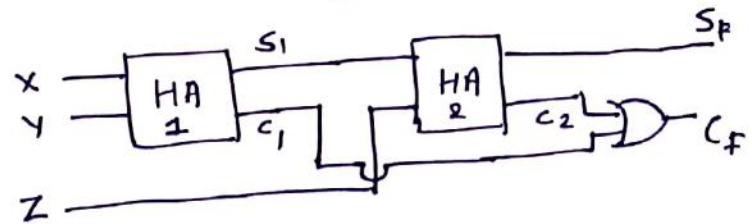
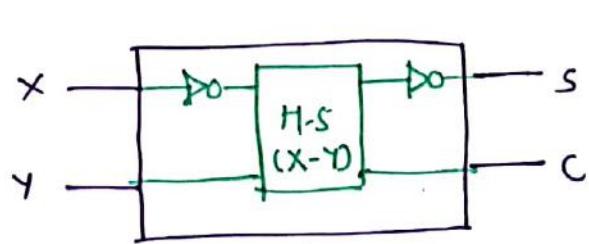
Half Subtractor using Half Adder. 16

→ Half subtractor

→ Half Adder



Full Adder using Half Subtractor 97



Full Subtractor 98

→ Full Subtractor performs 3 bits subtraction.

$$(A - B - C) \text{ or } A - (B + C)$$

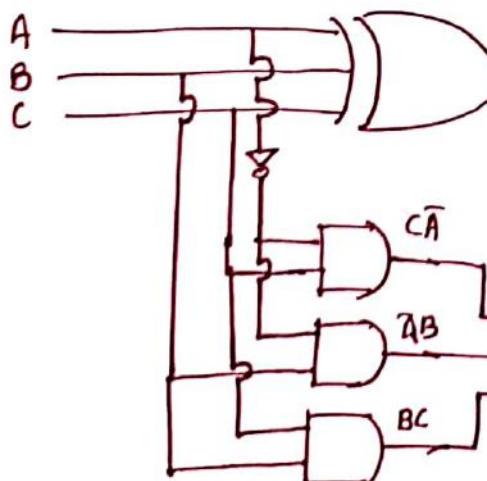
A	B	C	D	B_o
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	1
1	0	0	1	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

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

$$\begin{aligned}
 D &= \bar{A}\bar{B}C + \bar{A}B\bar{C} + ABC + A\bar{B}\bar{C} \\
 &= \bar{A}(\bar{B}C + B\bar{C}) + A(BC + \bar{B}\bar{C}) \\
 &= \bar{A}(B \oplus C) + A(B \odot C) \\
 &= A \oplus B \oplus C
 \end{aligned}$$

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

$$B_o = \bar{C}\bar{A} + \bar{A}B + BC$$



$$D = A \oplus B \oplus C$$

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

$$\text{Carry} = AB + BC + CA$$

Full Adder.

$$\begin{aligned}
 B_o &= \bar{C}\bar{A} + (\bar{A}B + BC) \\
 &= (A - B - C)
 \end{aligned}$$

Full Adder using Full Subtractor 188

→ Full Adder. → Full Subtractor

$$[A + B + C]$$

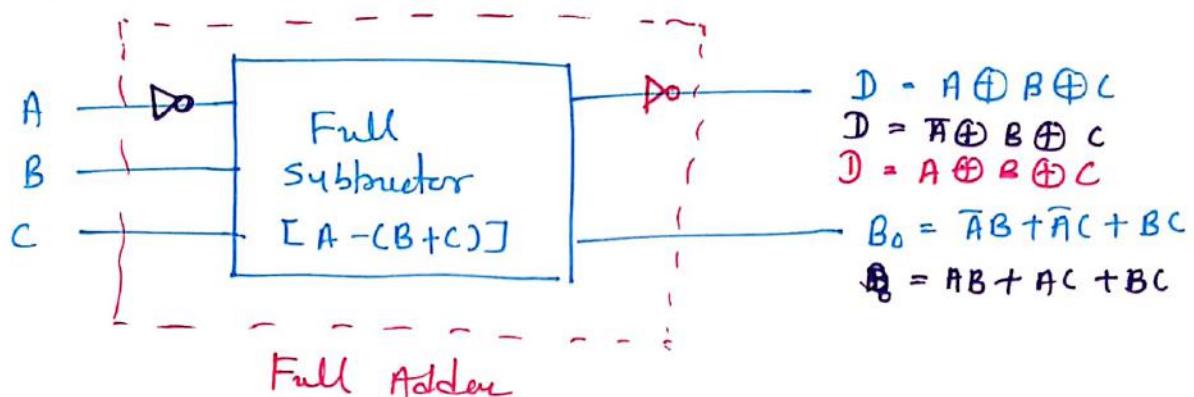
$$[A - (B+C)]$$

$$S = A \oplus B \oplus C$$

$$D = A \oplus B \oplus C$$

$$C_0 = AB + BC + CA$$

$$B_0 = \bar{A}B + \bar{A}C + BC$$



Full subtractor using Half subtractor [10]

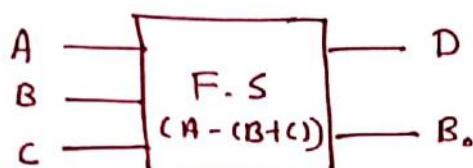
→ Full subtractor

$$[A, B, C]$$

$$[A - (B + C)]$$

$$\rightarrow D = A \oplus B \oplus C$$

$$B_o = \overline{AB} + \overline{AC} + BC$$



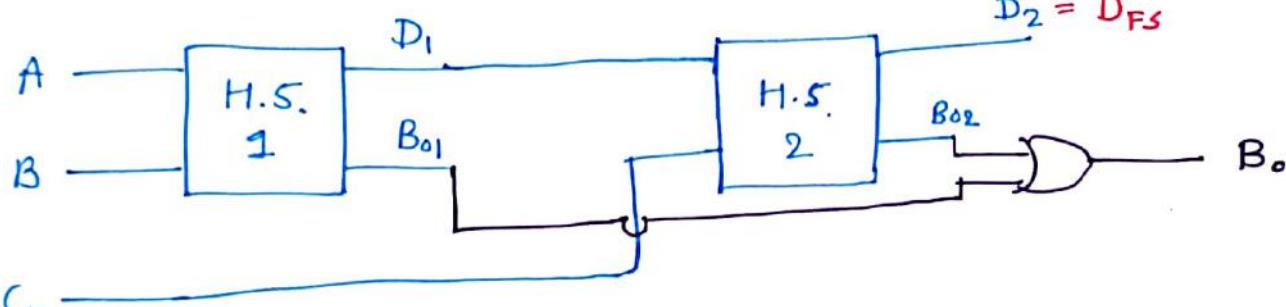
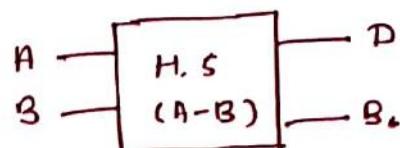
→ Half subtractor

$$[A, B]$$

$$[A - B]$$

$$\rightarrow D = A \oplus B$$

$$B_o = \overline{AB}$$



H.S. 1

$$D_1 = A \oplus B$$

$$B_{o1} = \overline{AB}$$

H.S. 2

$$D_2 = D_1 \oplus C = A \oplus B \oplus C$$

$$B_{o2} = \overline{D_1} \cdot C$$

$$\rightarrow B_o = B_{o1} + B_{o2}$$

$$= \overline{AB} + \overline{D_1} \cdot C$$

$$= \overline{AB} + (\overline{AB} \cdot B) \cdot C$$

$$= \overline{AB} + (AB + \overline{AB}) \cdot C$$

$$= \overline{AB} + ABC + \overline{A}\overline{B}C$$

$$= \overline{AB} + \overline{AC} + BC$$

	AB	C	B _o
00	0	0	0
01	0	1	1
11	1	1	1
10	1	0	0

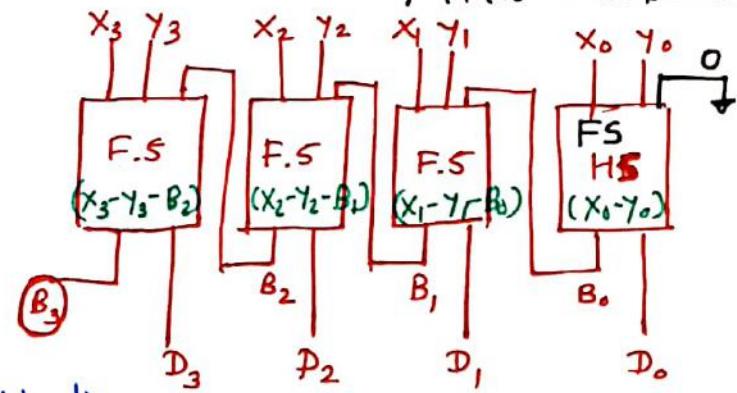
Parallel Subtractor using

B_3	B_2	B_1	B_0	O
X	X_3	X_2	X_1	X_0
$- Y$	Y_3	Y_2	Y_1	Y_0

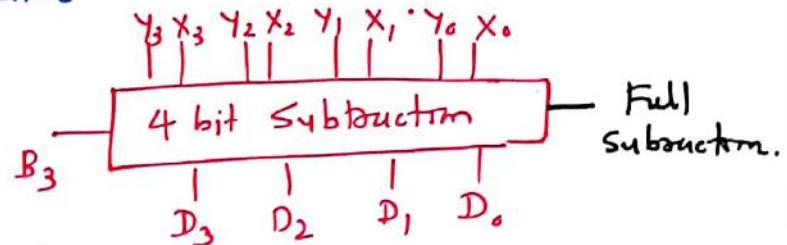
$B_0 \text{ D}$

$(X - Y - B)$
Full Subtraction.

Full Subtractor / Half subtractor.



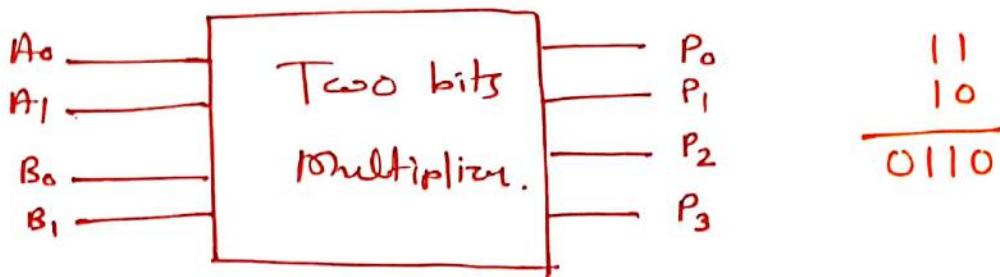
$(x_0 - y_0)$
Half Subtraction



Full
Subtraction.

102

2 Bits Multiplier using Half Adder 1.03



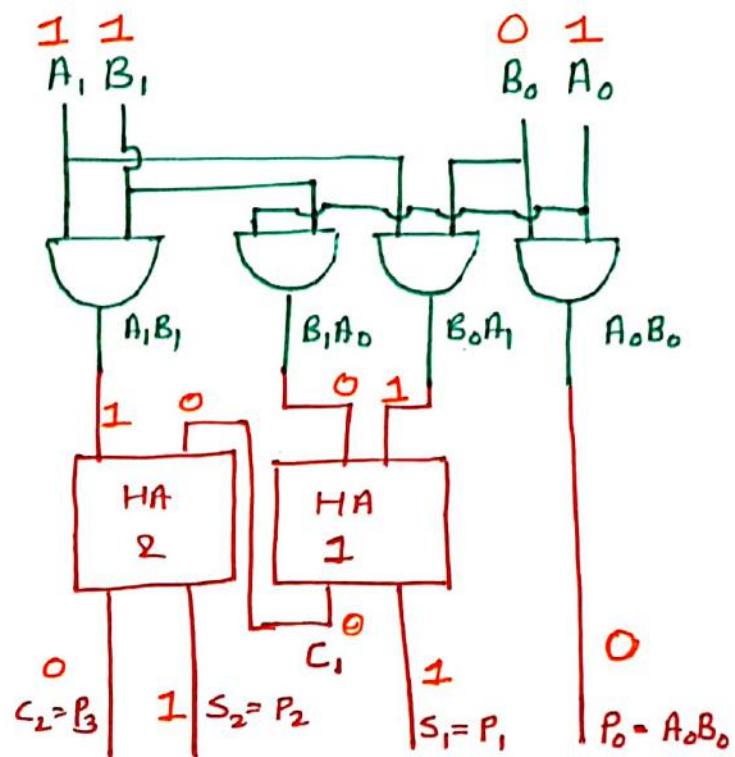
$$\begin{array}{r}
 A_1 \quad A_0 \\
 \times \quad \quad \quad B_1 \quad B_0 \\
 \hline
 C_1 \quad B_0A_1 \quad B_0A_0 \\
 C_2 \quad A_1B_1 \quad B_1A_0 \quad X \\
 \hline
 P_3 \quad P_2 \quad P_1 \quad P_0
 \end{array}$$

$$\rightarrow P_0 = B_0A_0$$

$$P_1 = HA_1[B_0A_1, B_1A_0]$$

$$P_2 = HA_2[C_1, A_1B_1]$$

$$P_3 = C_2$$



Excess-3 Addition by Parallel Adder

10th

Data

Step-1 - take two Excess-3 data A, B

A

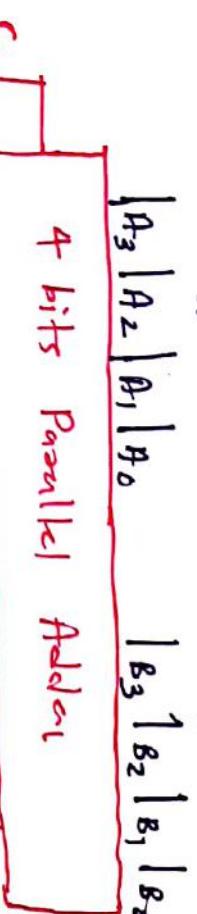
B

Data

Step-2 - Add Two data A & B

B

$$A + B = S \text{ (carry } C)$$



Step-3 - Check

If C = 0

B

Ans = S - 3

A

-3 → -0011

A

↓ 1's comp

1100

$\begin{array}{r} +1 \\ \hline \end{array}$ ↓ 2's comp

1101

Data

B

Data

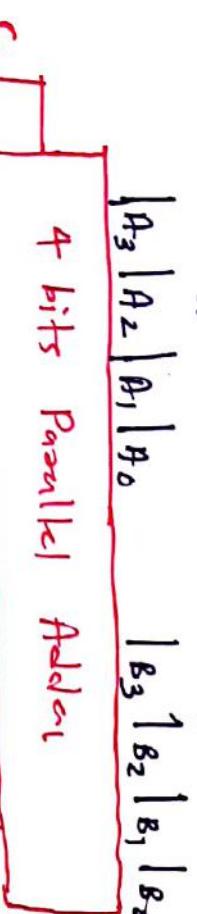
Step-1 - take two Excess-3 data A, B

B

Step-2 - Add Two data A & B

B

$$A + B = S \text{ (carry } C)$$



Step-3 - Check

If C = 1

B

Ans = S + 3

A

+3 → 0011

A

↓ 1's comp

1100

$\begin{array}{r} +1 \\ \hline \end{array}$ ↓ 2's comp

1101

Data

B

Data

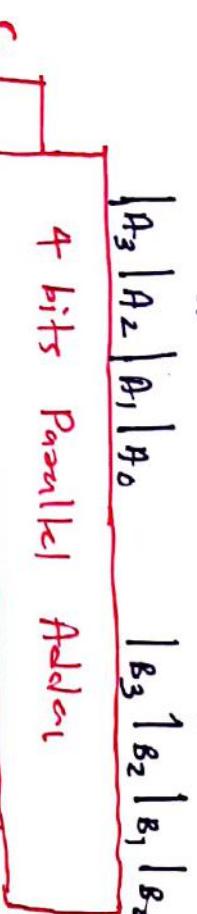
Step-1 - take two Excess-3 data A, B

B

Step-2 - Add Two data A & B

B

$$A + B = S \text{ (carry } C)$$



Step-3 - Check

If C = 1

B

Ans = S + 3

A

-3 → -0011

A

↓ 1's comp

1100

$\begin{array}{r} +1 \\ \hline \end{array}$ ↓ 2's comp

1101

Data

B

Data

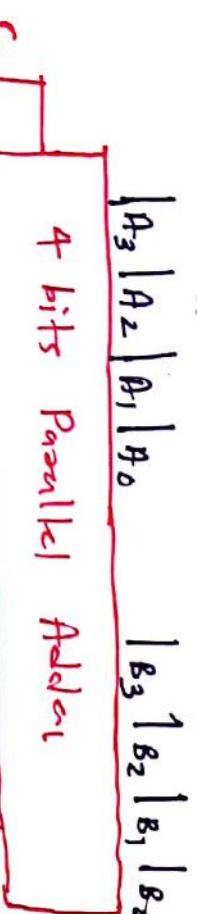
Step-1 - take two Excess-3 data A, B

B

Step-2 - Add Two data A & B

B

$$A + B = S \text{ (carry } C)$$



Step-3 - Check

If C = 1

B

Ans = S + 3

A

-3 → -0011

A

↓ 1's comp

1100

$\begin{array}{r} +1 \\ \hline \end{array}$ ↓ 2's comp

1101

Data

B

Data

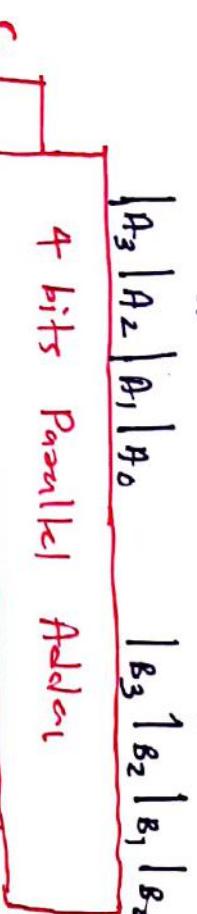
Step-1 - take two Excess-3 data A, B

B

Step-2 - Add Two data A & B

B

$$A + B = S \text{ (carry } C)$$



Step-3 - Check

If C = 1

B

Ans = S + 3

A

-3 → -0011

A

↓ 1's comp

1100

$\begin{array}{r} +1 \\ \hline \end{array}$ ↓ 2's comp

1101

Data

B

Data

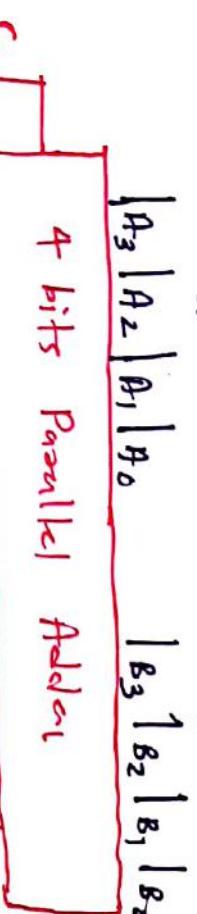
Step-1 - take two Excess-3 data A, B

B

Step-2 - Add Two data A & B

B

$$A + B = S \text{ (carry } C)$$



Step-3 - Check

If C = 1

B

Ans = S + 3

A

-3 → -0011

A

↓ 1's comp

1100

$\begin{array}{r} +1 \\ \hline \end{array}$ ↓ 2's comp

1101

Data

B

Data

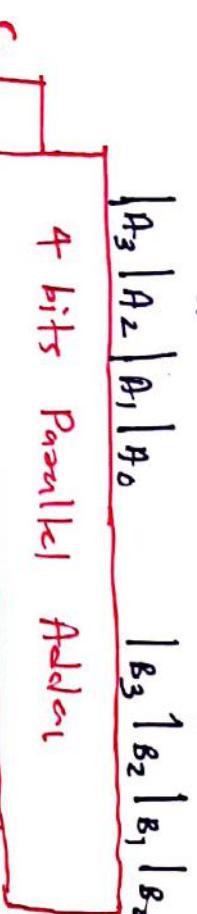
Step-1 - take two Excess-3 data A, B

B

Step-2 - Add Two data A & B

B

$$A + B = S \text{ (carry } C)$$



Step-3 - Check

If C = 1

B

Ans = S + 3

A

-3 → -0011

A

↓ 1's comp

1100

$\begin{array}{r} +1 \\ \hline \end{array}$ ↓ 2's comp

1101

Data

B

Data

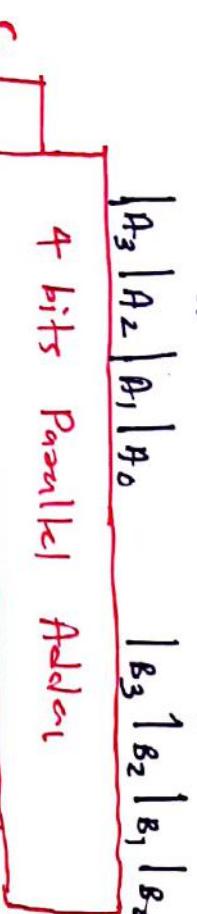
Step-1 - take two Excess-3 data A, B

B

Step-2 - Add Two data A & B

B

$$A + B = S \text{ (carry } C)$$



Step-3 - Check

If C = 1

B

Ans = S + 3

A

-3 → -0011

A

↓ 1's comp

1100

$\begin{array}{r} +1 \\ \hline \end{array}$ ↓ 2's comp

1101

Data

B

Data

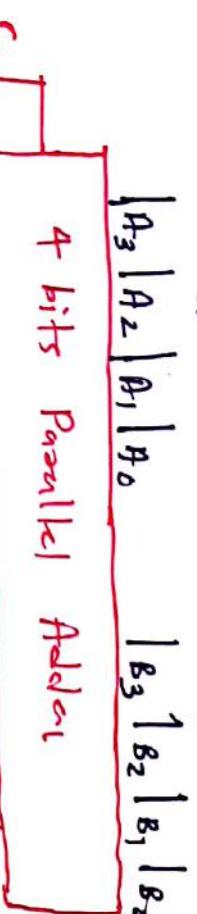
Step-1 - take two Excess-3 data A, B

B

Step-2 - Add Two data A & B

B

$$A + B = S \text{ (carry } C)$$



Step-3 - Check

If C = 1

B

Ans = S + 3

A

-3 → -0011

A

↓ 1's comp

1100

$\begin{array}{r} +1 \\ \hline \end{array}$ ↓ 2's comp

1101

Data

B

Data

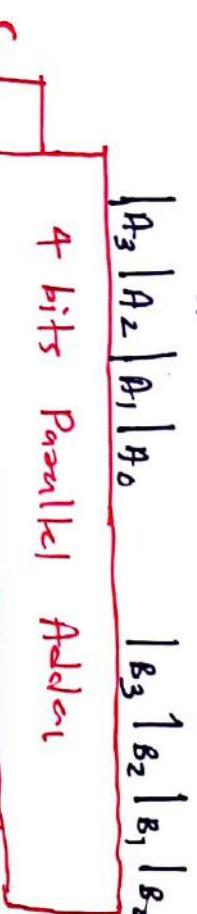
Step-1 - take two Excess-3 data A, B

B

Step-2 - Add Two data A & B

B

$$A + B = S \text{ (carry } C)$$



Step-3 - Check

If C = 1

B

Ans = S + 3

A

-3 → -0011

A

↓ 1's comp

1100

$\begin{array}{r} +1 \\ \hline \end{array}$ ↓ 2's comp

1101

Data

B

Data

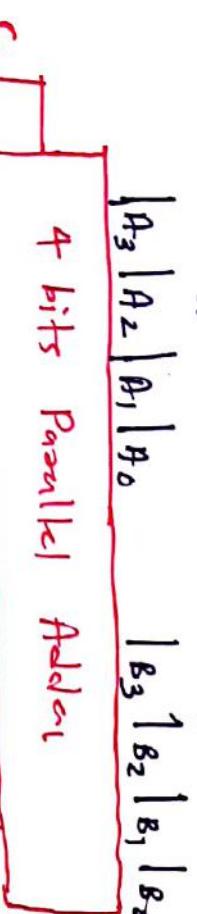
Step-1 - take two Excess-3 data A, B

B

Step-2 - Add Two data A & B

B

$$A + B = S \text{ (carry } C)$$



Step-3 - Check

If C = 1

B

Ans = S + 3

A

-3 → -0011

A

↓ 1's comp

1100

$\begin{array}{r} +1 \\ \hline \end{array}$ ↓ 2's comp

1101

Data

B

Data

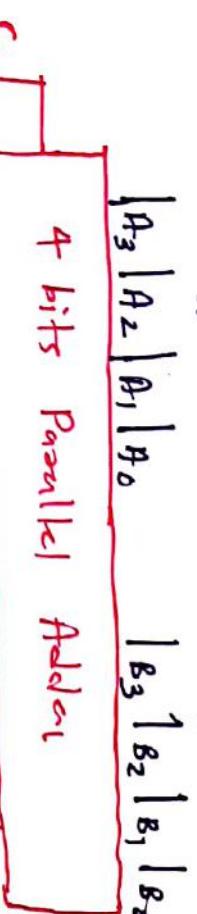
Step-1 - take two Excess-3 data A, B

B

Step-2 - Add Two data A & B

B

$$A + B = S \text{ (carry } C)$$



Step-3 - Check

If C = 1

B

Ans = S + 3

A

-3 → -0011

A

↓ 1's comp

1100

$\begin{array}{r} +1 \\ \hline \end{array}$ ↓ 2's comp

1101

Data

B

Data

Step-1 - take two Excess-3 data A, B

B

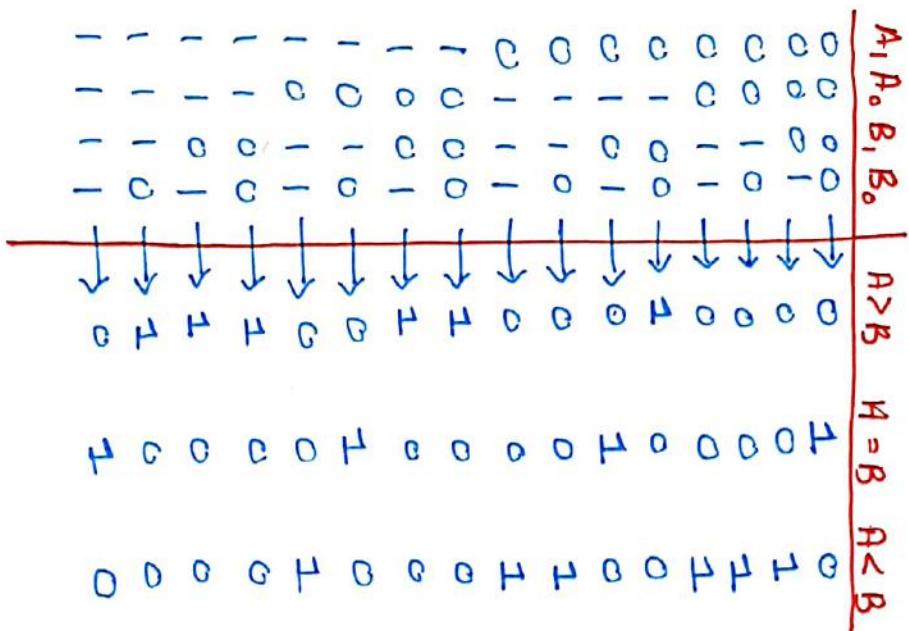
Step-2 - Add Two data A & B

B

$$A + B = S \text{ (carry } C)$$



2 bits Comparator 105



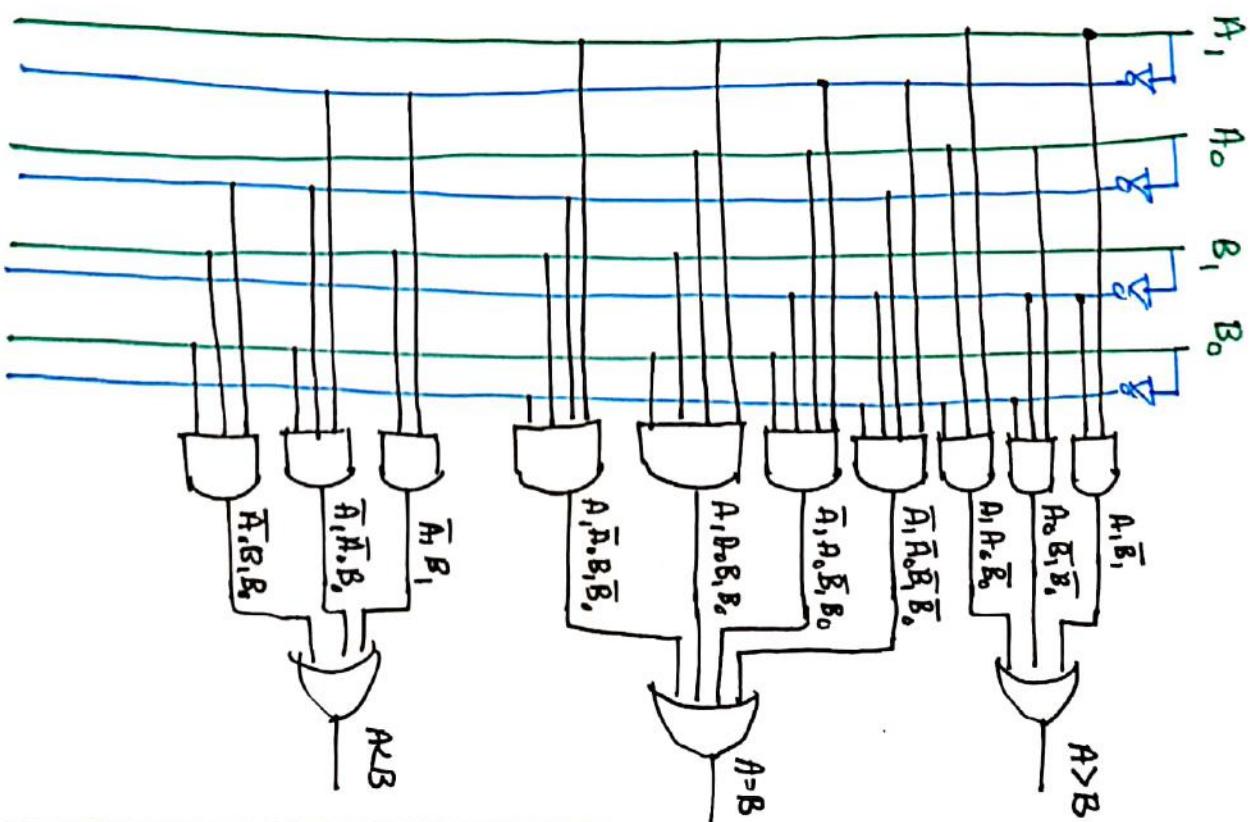
$$(A > B) = A_1 \bar{B}_1 + A_0 B_1 \bar{B}_0 + A_1 A_0 \bar{B}_0.$$

$A > B$	A_1, A_0	B_1, B_0	$A_1 \bar{B}_1$	$A_0 B_1 \bar{B}_0$	$A_1 A_0 \bar{B}_0$
00	00	00	1	0	0
01	01	00	1	1	0
10	10	00	1	0	1
11	11	00	1	1	1

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

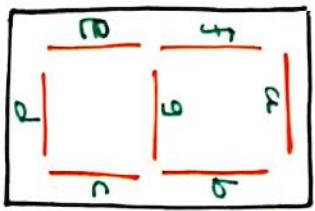
$A < B$	A_1, A_0	B_1, B_0	$\bar{A}_1 \bar{A}_0$	$\bar{B}_1 \bar{B}_0$	$\bar{A}_1 A_0$	$\bar{B}_1 B_0$	$A_1 \bar{A}_0$	$\bar{B}_1 \bar{B}_0$	$A_1 A_0 \bar{B}_1$	$\bar{B}_1 B_0$
00	00	00	1	1	0	0	0	1	0	0
01	01	00	1	1	1	0	1	1	1	0
11	11	00	1	1	1	1	1	1	1	1
10	10	00	1	1	1	1	0	1	1	1

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



Seven Segment Display Decoder

106



Digit	b_3	b_2	b_1	b_0	a	b	c	d	e	f	g
0	-	-	-	-	0	0	0	0	0	0	0
1	-	-	-	0	0	0	0	0	0	0	0
2	-	-	1	0	1	0	1	0	0	0	0
3	-	-	1	1	1	0	1	0	1	0	0
4	-	1	1	0	0	1	1	1	0	0	0
5	-	1	0	1	1	1	0	1	1	0	0
6	-	1	0	1	1	1	1	0	1	0	0
7	-	0	1	1	0	1	1	1	0	0	0
8	0	1	1	1	1	1	1	1	1	1	1
9	1	1	1	1	0	1	1	1	1	0	1

Other	00	01	10	11	00	01	10	11	00	01	10
-	-	-	-	-	0	0	0	0	0	0	0
-	0	0	0	0	0	0	0	0	0	0	0
-	1	0	0	0	0	0	0	0	0	0	0
-	1	0	0	0	0	0	0	0	0	0	0
-	0	1	0	0	0	0	0	0	0	0	0
-	1	0	0	0	0	0	0	0	0	0	0
x	x	-	-	-	0	1	0	1	0	1	0
x	x	-	-	-	1	0	1	0	1	0	1
x	x	-	-	-	0	1	0	1	0	1	0
x	x	-	-	-	1	0	1	0	1	0	1
x	x	-	-	-	0	1	0	1	0	1	0
x	x	-	-	-	0	0	1	0	1	0	1
x	x	-	-	-	0	0	0	1	0	1	0
x	x	-	-	-	0	0	0	0	1	0	1
x	x	-	-	-	0	0	0	0	0	1	0

b_3b_2	00	01	10	11
00	00	X ₁₄	X ₁₀	X ₁₁
01	01	X ₁₅	X ₁₃	X ₁₂
11	11	X ₁₉	X ₁₇	X ₁₈
10	10	X ₁₂	X ₁₄	X ₁₆

$$g = b_3 + \overline{b}_1b_2 + b_1\overline{b}_0 + b_1\overline{b}_2$$

b_3b_2	00	01	10	11
00	10	X ₁₄	X ₁₂	X ₁₃
01	01	X ₁₅	X ₁₃	X ₁₁
11	11	X ₁₉	X ₁₇	X ₁₈
10	10	X ₁₂	X ₁₄	X ₁₆

$$c = b_1\overline{b}_0 + \overline{b}_0\overline{b}_2$$

$$f = b_3 + \overline{b}_0b_2 + \overline{b}_0\overline{b}_1 + \overline{b}_1\overline{b}_2$$

b_3b_2	00	01	10	11
00	10	X ₁₄	X ₁₂	X ₁₃
01	01	X ₁₅	X ₁₃	X ₁₁
11	11	X ₁₉	X ₁₇	X ₁₈
10	10	X ₁₂	X ₁₄	X ₁₆

$$d = b_3 + b_1\overline{b}_0 + \overline{b}_1\overline{b}_2 + \overline{b}_0\overline{b}_1$$

$$q = b_1 + b_3 + b_0b_2 + \overline{b}_0\overline{b}_2$$

$$b = \overline{b}_2 + \overline{b}_1\overline{b}_0 + b_1\overline{b}_0$$

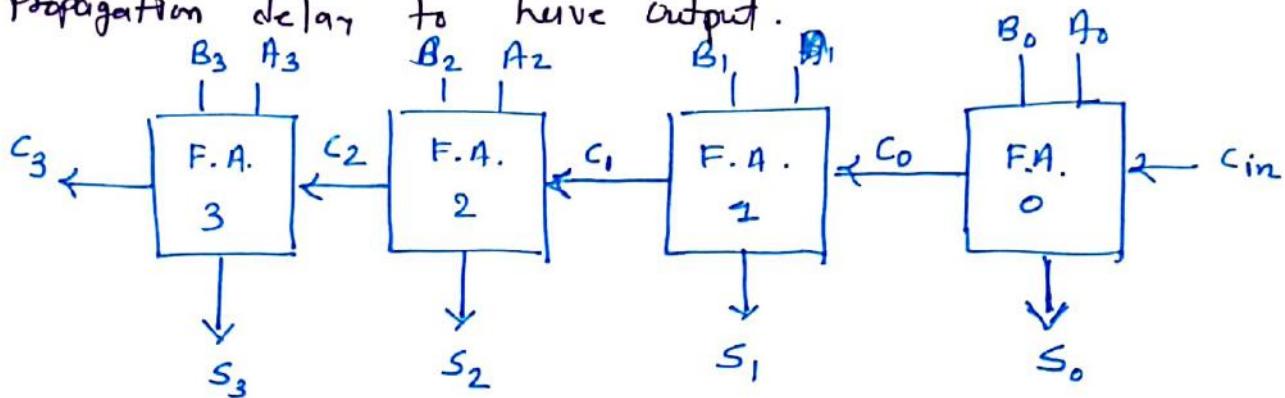
b_3b_2	00	01	11	10
00	10	X ₁₂	X ₁₄	X ₁₈
01	01	X ₁₅	X ₁₃	X ₁₉
11	11	X ₁₇	X ₁₅	X ₁₄
10	10	X ₁₆	X ₁₄	X ₁₀

$$f = b_3 + \overline{b}_0b_2 + \overline{b}_0\overline{b}_1 + \overline{b}_1\overline{b}_2$$

$$b = \overline{b}_2 + \overline{b}_1\overline{b}_0 + b_1\overline{b}_0$$

Carry Look Ahead Adder [CLA Adder] 107

- CLA Adder is better than Full Adder in terms of Speed.
- By Full Adder when we have parallel Adder, it takes Propagation delay to have output.



A_0	B_0	C_{in}	C_0
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

$\overbrace{A_0 B_0}^{A_0 \bar{B}_0 C_{in}}$

$$\begin{aligned} \rightarrow C_0 &= A_0 B_0 + A_0 \bar{B}_0 C_{in} + \bar{A}_0 B_0 C_{in} \\ &= A_0 B_0 + C_{in} (A_0 \bar{B}_0 + \bar{A}_0 B_0) \\ &= A_0 B_0 + C_{in} (A_0 \oplus B_0) \end{aligned}$$

$$\rightarrow C_i = \underbrace{A_i B_i}_{\text{Carry generator } G_i} + \underbrace{C_{i-1}}_{\text{Carry propagator } P_i} (\underbrace{A_i \oplus B_i}_{G_i \oplus P_i})$$

$$\rightarrow [C_i = [G_i + C_{i-1} P_i]]$$

$$\rightarrow i=0$$

$$C_0 = G_0 + C_{-1} P_0$$

$$\rightarrow i=1$$

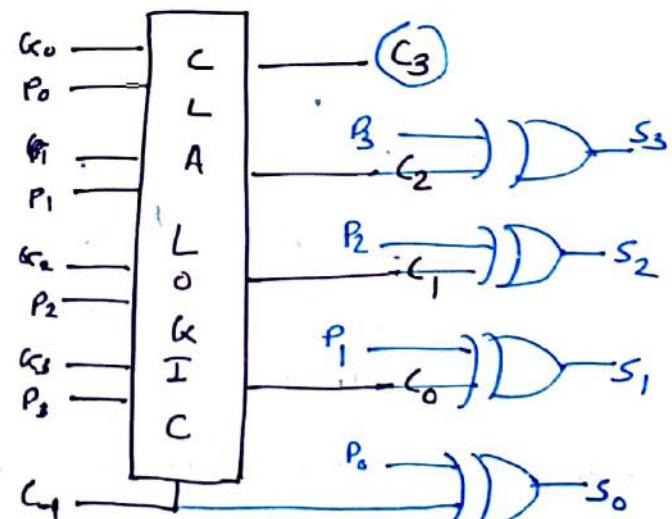
$$\begin{aligned} C_1 &= G_1 + C_0 P_1 = G_1 + [G_0 + C_{-1} P_0] P_1 \\ &= G_1 + P_1 G_0 + P_1 P_0 C_{-1} \end{aligned}$$

$$\rightarrow i=2$$

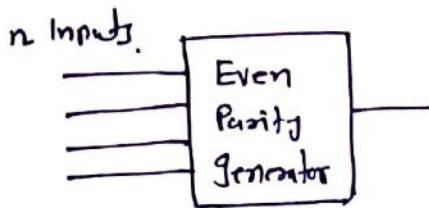
$$\begin{aligned} C_2 &= G_2 + C_1 P_2 \\ &= G_2 + P_2 [G_1 + P_1 G_0 + P_1 P_0 C_{-1}] \\ &= G_2 + P_2 G_1 + P_2 P_1 G_0 + P_2 P_1 P_0 C_{-1} \end{aligned}$$

$$\rightarrow i=3$$

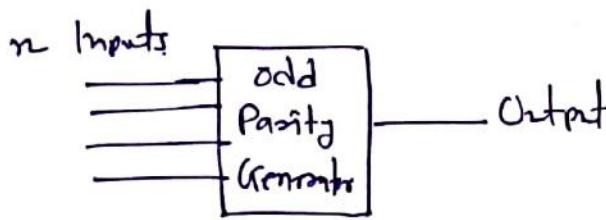
$$\begin{aligned} C_3 &= G_3 + C_2 P_3 \\ &= G_3 + P_3 [G_2 + P_2 G_1 + P_2 P_1 G_0 + P_2 P_1 P_0 C_{-1}] \\ &= G_3 + P_3 G_2 + P_3 P_2 G_1 + P_3 P_2 P_1 G_0 + P_3 P_2 P_1 P_0 C_{-1} \end{aligned}$$



Even Parity Generator / Odd Parity Generator 108



- For even parity generator
Output = 1 when number of 1's at b_p is odd.



- For odd parity generator
Output = 1, when number of 1's at b_p is even.

b_3	b_2	b_1	b_0	O/P (Even parity) P
0	0	0	0	0 (0)
0	0	0	1	1 (1)
0	0	1	0	1 (2)
0	0	1	1	0 (3)
0	1	0	0	1 (4)
0	1	0	1	0 (5)
0	1	1	0	0 (6)
0	1	1	1	1 (7)
1	0	0	0	1 (8)
1	0	0	1	0 (9)
1	0	1	0	1 (10)
1	0	1	1	0 (11)
1	1	0	0	0 (12)
1	1	0	1	1 (13)
1	1	1	0	1 (14)
1	1	1	1	0 (15)

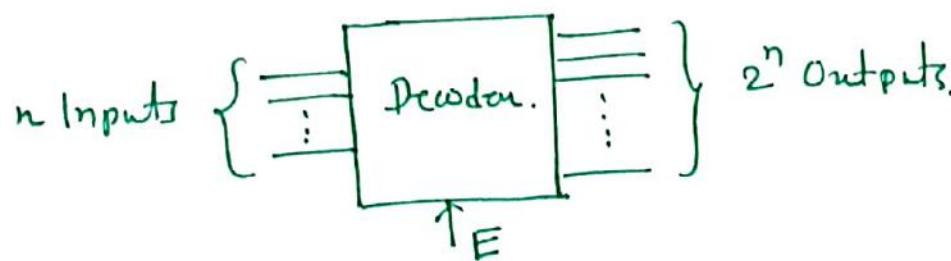
$Q_{\text{odd}} = \overline{b_3 \oplus b_2 \oplus b_1 \oplus b_0}$

P	$b_3 b_2$	00	01	11	10
00	0	1	0	1	0
01	1	0	1	0	1
11	0	1	1	0	1
10	1	1	0	1	0

$$P(b_0 b_1 b_2 b_3) = b_0 \oplus b_1 \oplus b_2 \oplus b_3$$

Decoder Basics and 2×4 Decoder 109

- Decoder decodes n Inputs to 2^n Outputs.

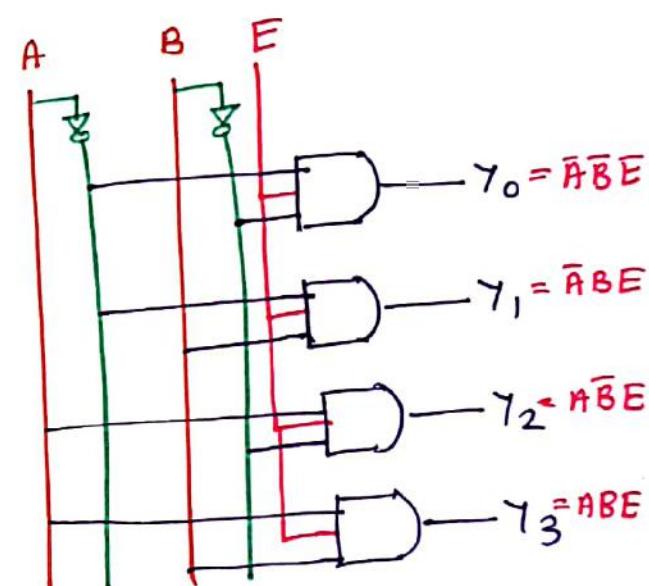
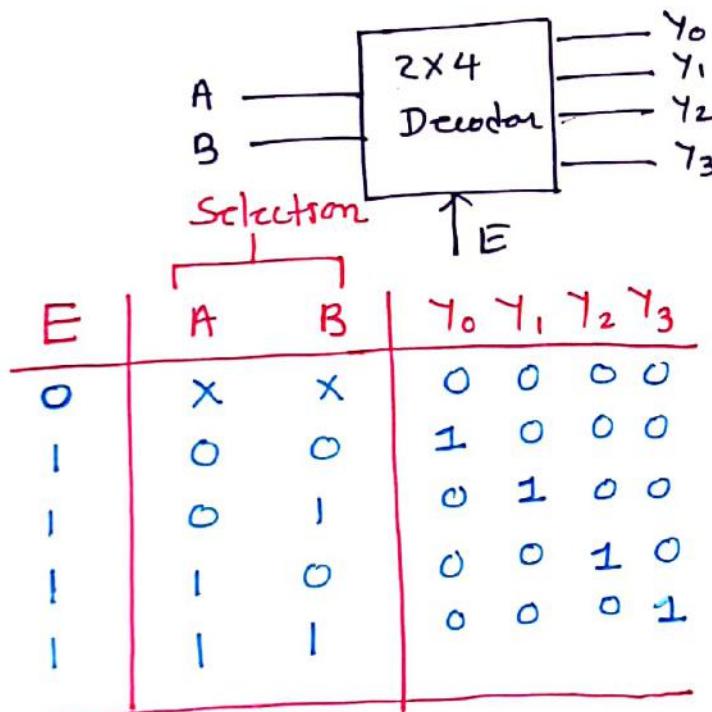


E = Enable.

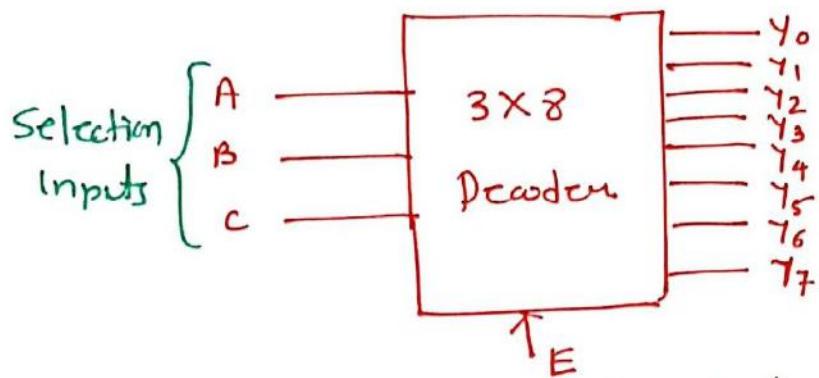
If $E=0$, Decoder is disabled.

If $E=1$, Decoder is Enabled.

- 2×4 Decoder.



3 X 8 Decoder 110



E is enable terminal

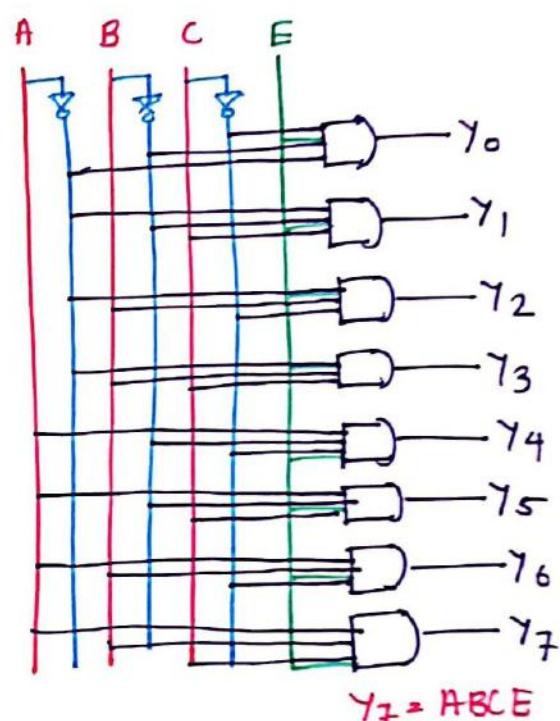
E = 0, Decoder is disabled.

E = 1, Decoder is Enabled.

E	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

Selection Inputs

Outputs.

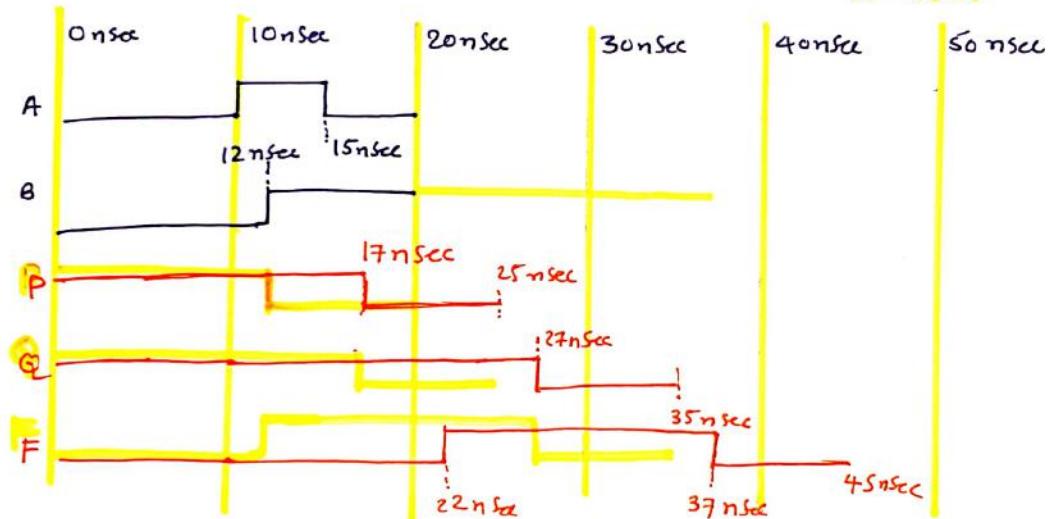
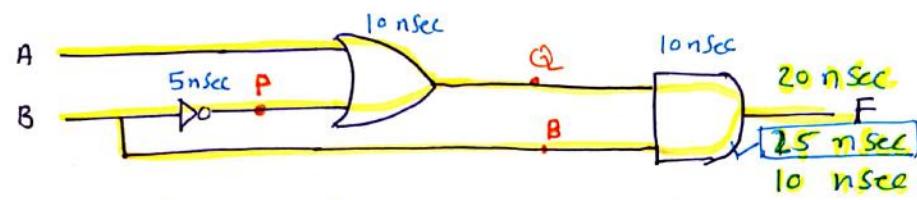


Combinational Circuit Output Waveform with delay at gates. [Part 1]

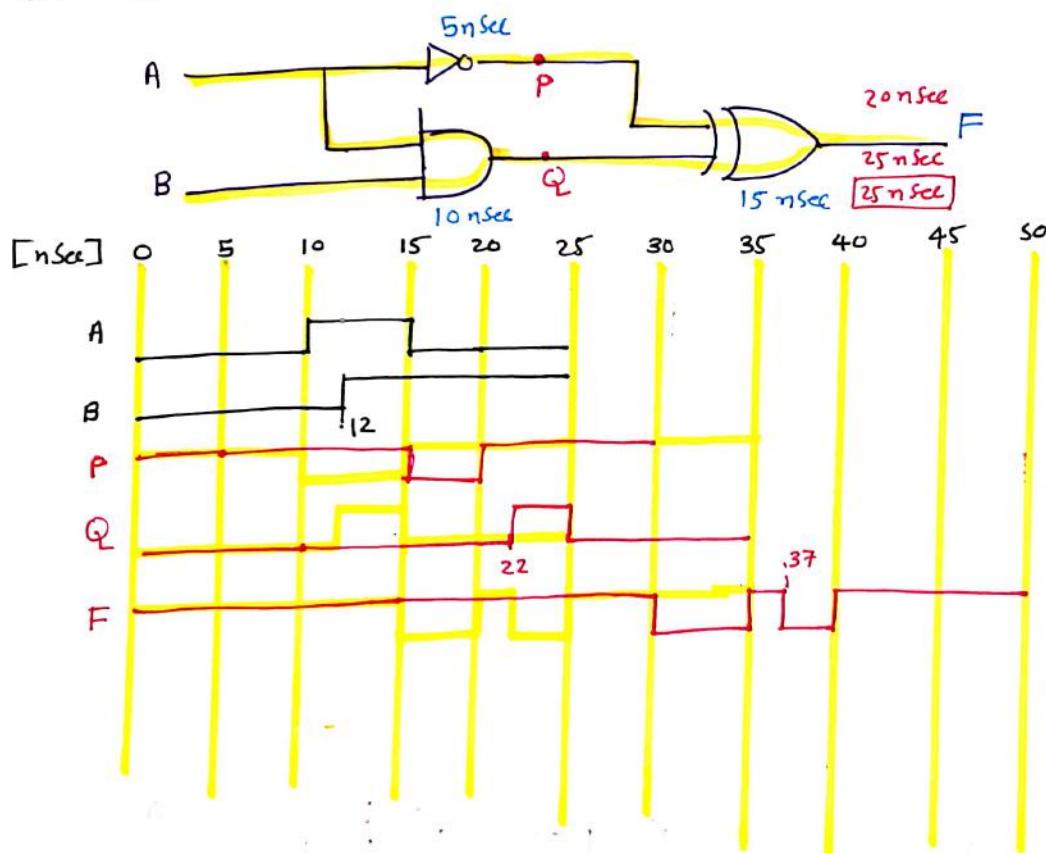
III

F.

- 1] Find output waveform
- 2] Find total delay of given combinational circuit.

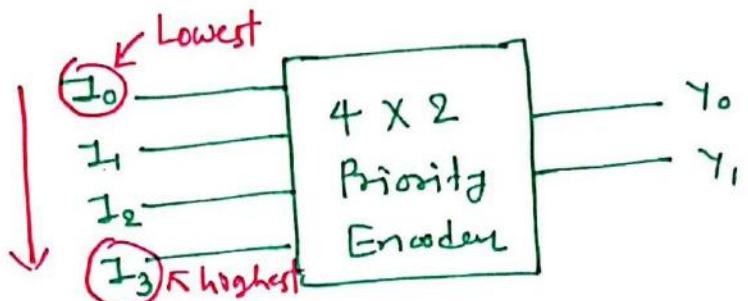


Combinational circuit Output waveform with delay at gates [Part 2].
1/2



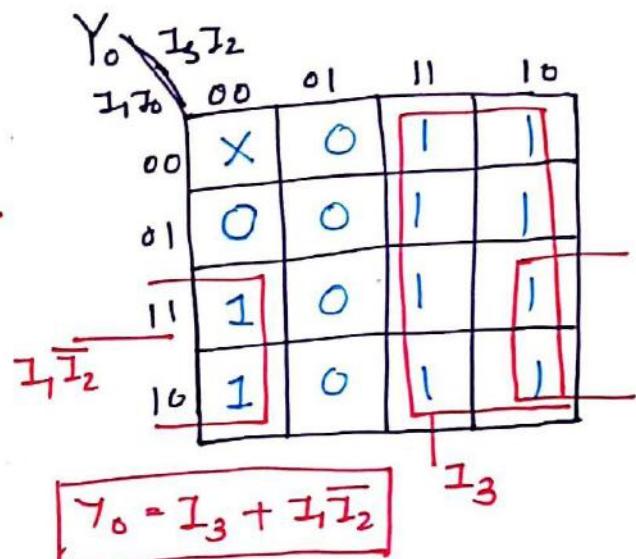
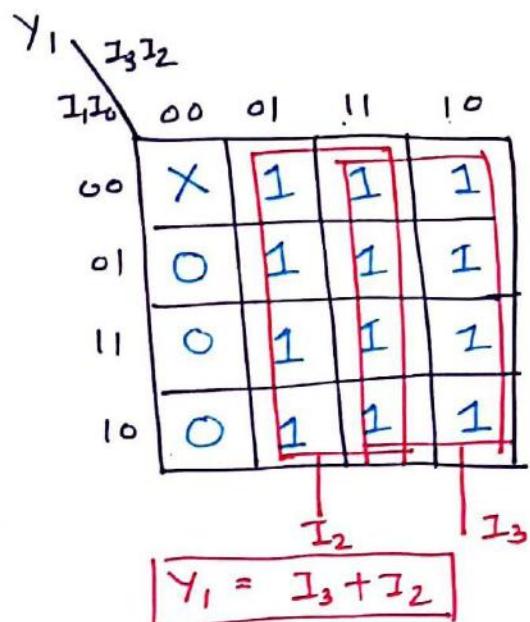
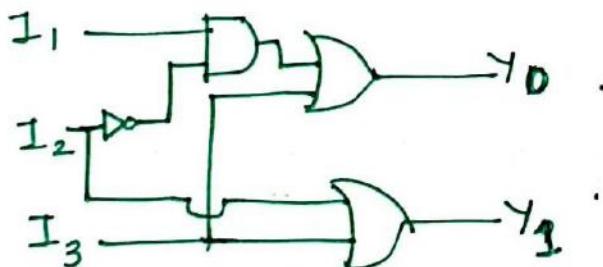
- 1] Find output waveform F.
- 2] Find total delay of given combinational circuit.

Priority Encoder 11E



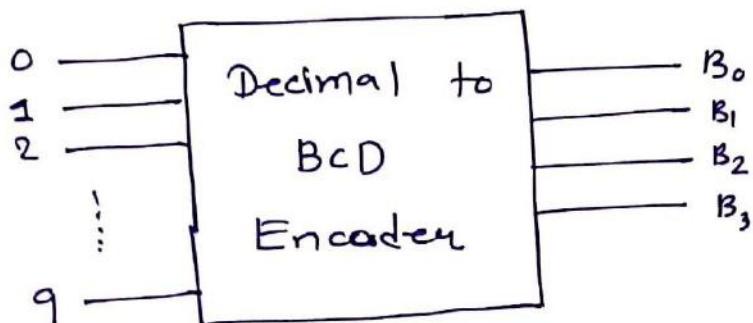
$$\rightarrow n = 4 = 2^m \Rightarrow m = 2$$

I_3	I_2	I_1	I_0	Y_1	Y_0
0	0	0	0	X	X
0	0	0	1	0	0
0	0	1	X	0	1
0	1	X	X	1	0
1	X	X	X	1	1



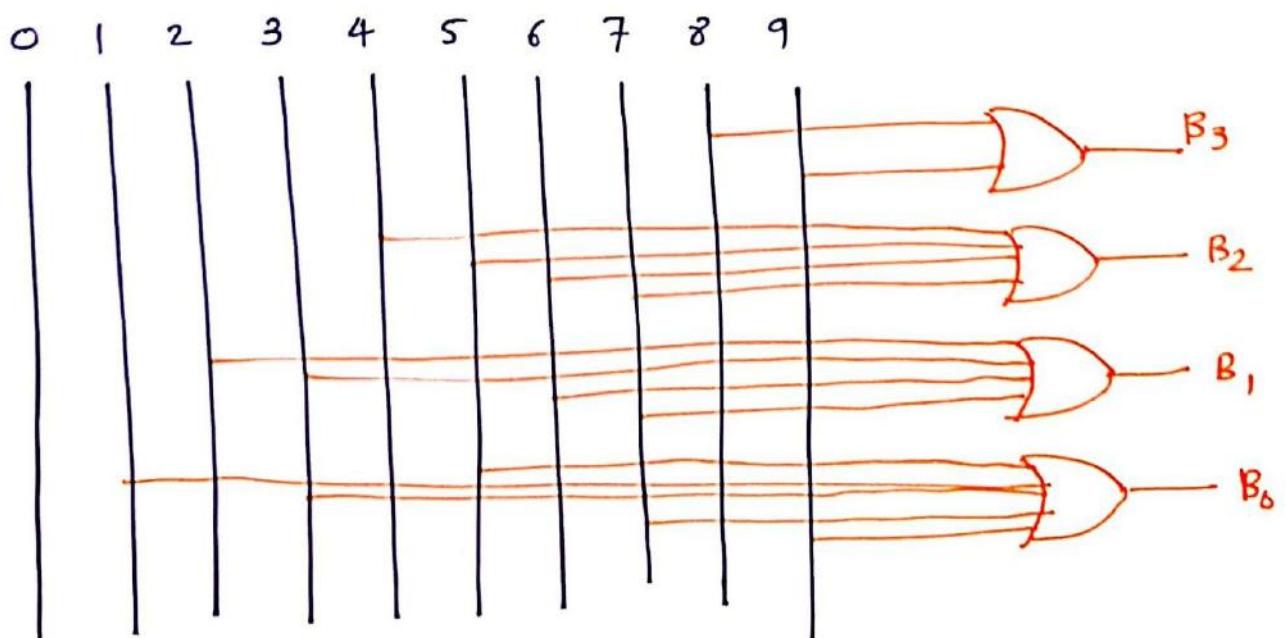
Decimal to BCD Encoder 114

→ Decimal [0 to 9] to BCD [0000 to 1001] Encoder.

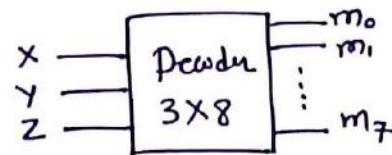


Input Decimal Number	O/P			
	B_3	B_2	B_1	B_0
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1

$$\begin{aligned}
 B_3 &= \textcircled{8} + \textcircled{9} \\
 B_2 &= \textcircled{4} + \textcircled{5} + \textcircled{6} + \textcircled{7} \\
 B_1 &= \textcircled{2} + \textcircled{3} + \textcircled{6} + \textcircled{7} \\
 B_0 &= \textcircled{1} + \textcircled{3} + \textcircled{5} + \textcircled{7} + \textcircled{9}
 \end{aligned}$$

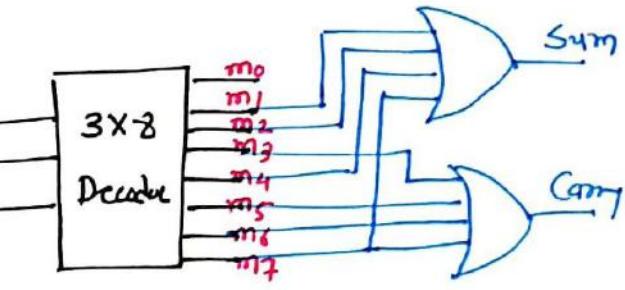


Full Adder using Decoder 115



$$m = 2^n = 2^3 = 8$$

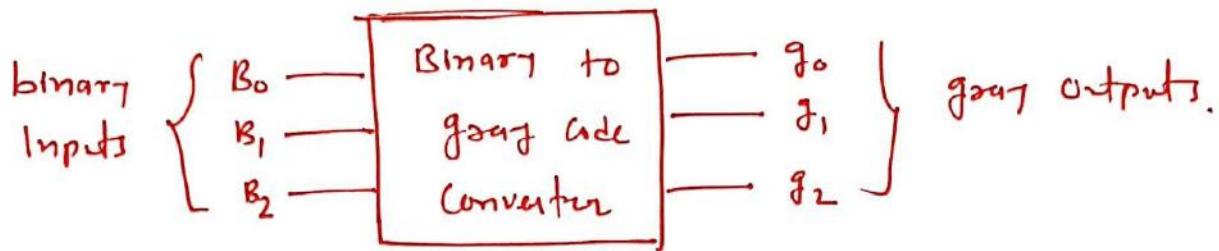
X	Y	Z	S	C	O/P
0	0	0	0	0	m_0
0	0	1	1	0	m_1
0	1	0	1	0	m_2
0	1	1	0	1	m_3
1	0	0	1	0	m_4
1	0	1	0	1	m_5
1	1	0	0	1	m_6
1	1	1	1	1	m_7



$$S = m_1 + m_2 + m_4 + m_7$$

$$C = m_3 + m_5 + m_6 + m_7$$

Binary to Gray Code Converter 116



Inputs			Outputs		
B_2	B_1	B_0	g_2	g_1	g_0
0	0	0	0	0	0
0	0	1	0	0	1
0	1	0	0	1	1
0	1	1	0	1	0
1	0	0	1	1	0
1	0	1	1	1	1
1	1	0	1	0	1
1	1	1	1	0	0

$g_2 = B_2 B_1$

B_0	00	01	11	10
0	0	0	1	1
1	0	0	1	1

$g_1 = \overline{B}_2 B_1 + B_2 \overline{B}_1$

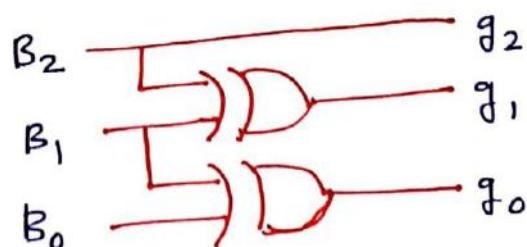
$$g_1 = B_2 \oplus B_1$$

B_0	00	01	11	10
0	0	1	0	1
1	0	1	0	1

$g_0 = B_0 \overline{B}_1 + \overline{B}_0 B_1$

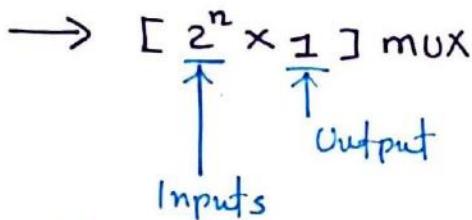
$$g_0 = B_0 \oplus B_1$$

$B_0 \overline{B}_1$	00	01	11	10
0	0	1	1	0
1	1	0	0	1

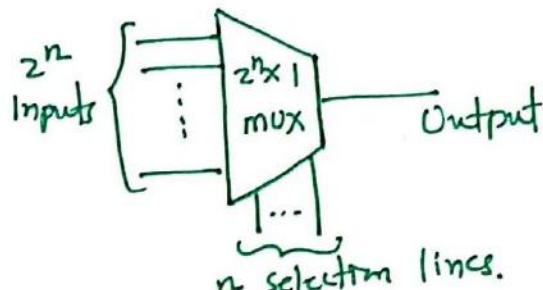
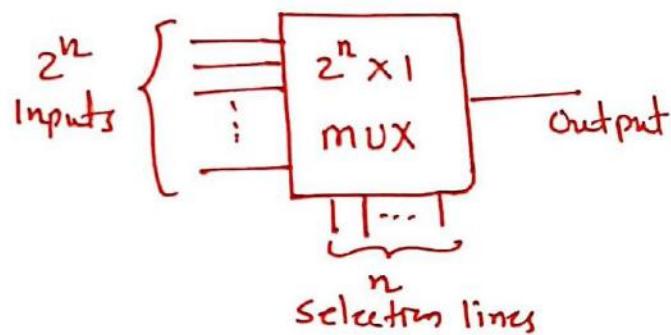


Multiplexer [MUX]

117



→ n is number of selection lines.



→ MUX IC's → MSI Scale IC's.

Advantages

- 1) Complexity is less
- 2) Cost is less
- 3) Less wiring.

Applications

- 1) Data Selector
- 2) Switch
- 3) Combinational Circuits.

Types → $[2^n \times 1]$ MUX

$[2 \times 1]$ MUX
 $n=1$

$[4 \times 1]$ MUX
 $n=2$

$[8 \times 1]$ MUX
 $n=3$

$[16 \times 1]$ MUX
 $n=4$

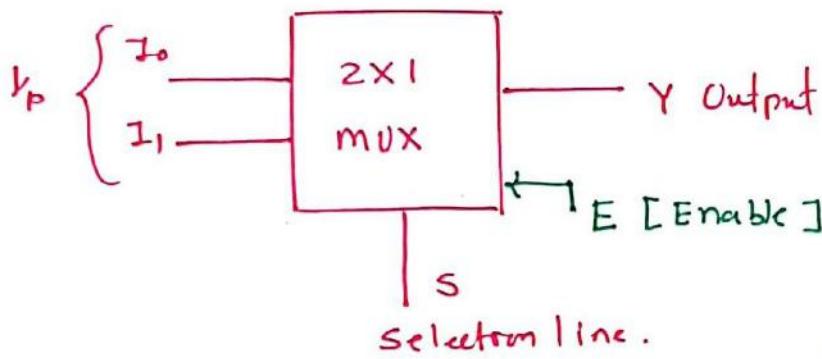
$[32 \times 1]$ MUX
 $n=5$

2×1 Multiplexer 118

→ $[2^n \times 1]$ MUX

→ $n=1$, 1 selection line

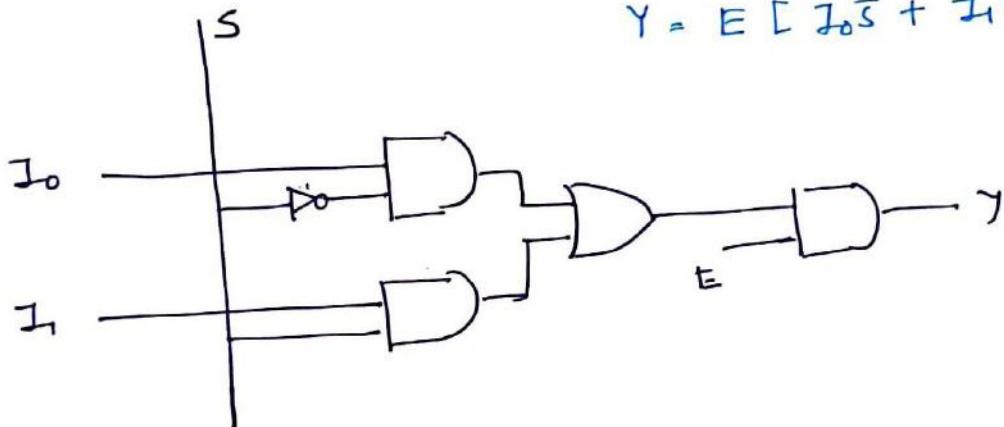
→ 2 Inputs, 1 output.



E	S	Y
0	X	0
1	0	I_0
1	1	I_1

$$Y = I_0 \bar{S} E + I_1 S E$$

$$Y = E [I_0 \bar{S} + I_1 S]$$

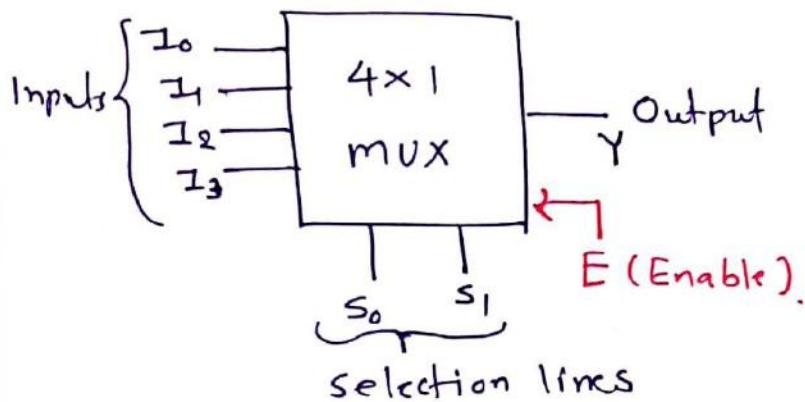


4x1 MUX

119

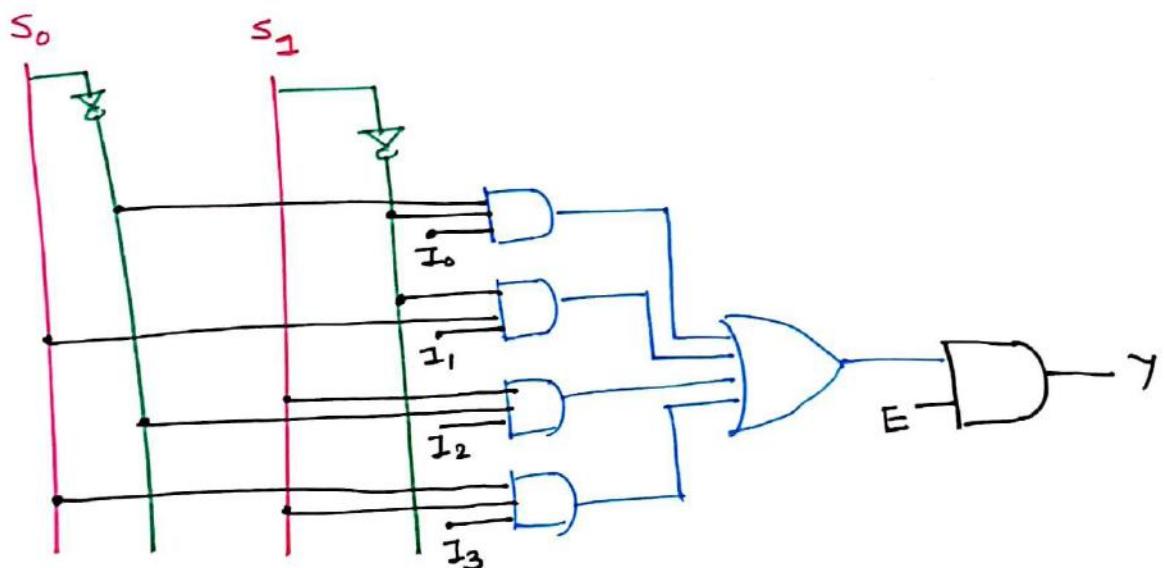
→ $[2^n \times 1]$ MUX

⇒ $n=2$, 2 Selection lines.



E	S ₁	S ₀	Y
0	X	X	0
1	0	0	I ₀
1	0	1	I ₁
1	1	0	I ₂
1	1	1	I ₃

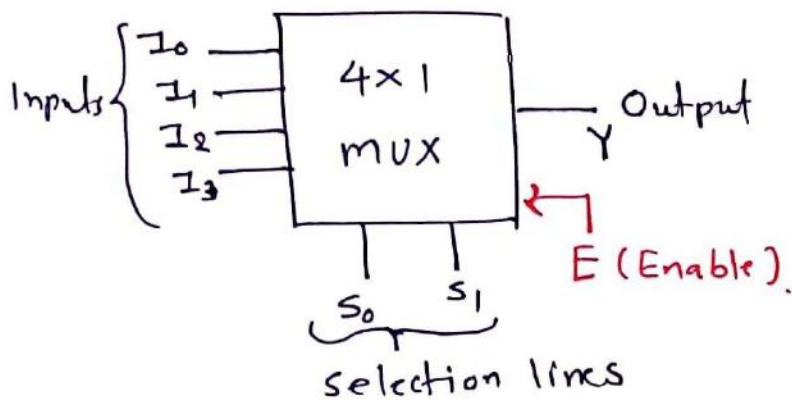
$$Y = E \cdot [\bar{S}_1 \bar{S}_0 I_0 + \bar{S}_1 S_0 I_1 + S_1 \bar{S}_0 I_2 + S_1 S_0 I_3]$$



4x1 MUX 119

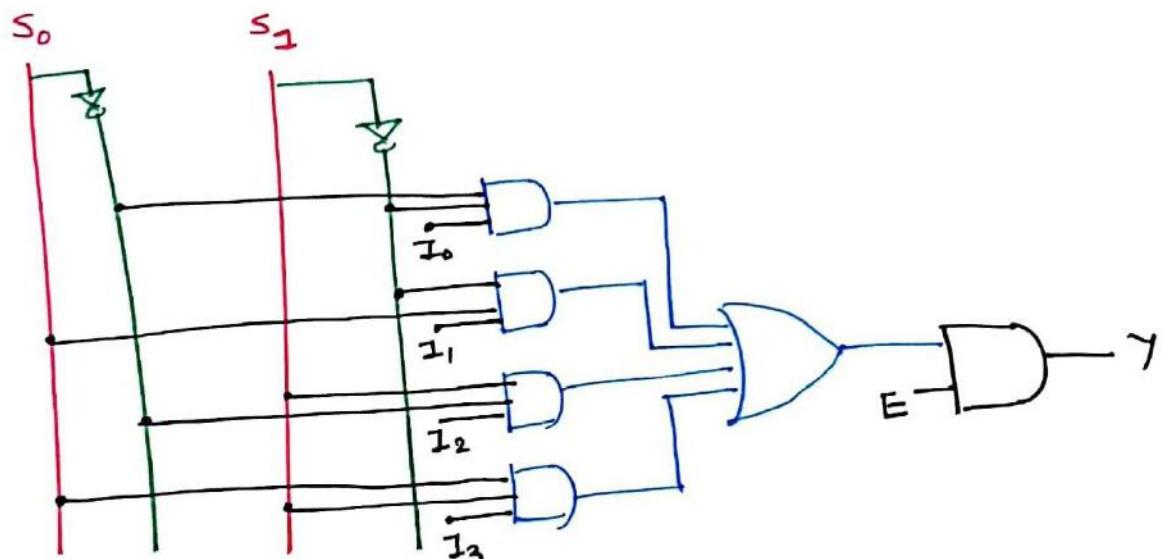
→ $[2^n \times 1]$ MUX

⇒ $n = 2$, 2 Selection lines.



E	S ₁	S ₀	Y
0	X	X	0
1	0	0	I ₀
1	0	1	I ₁
1	1	0	I ₂
1	1	1	I ₃

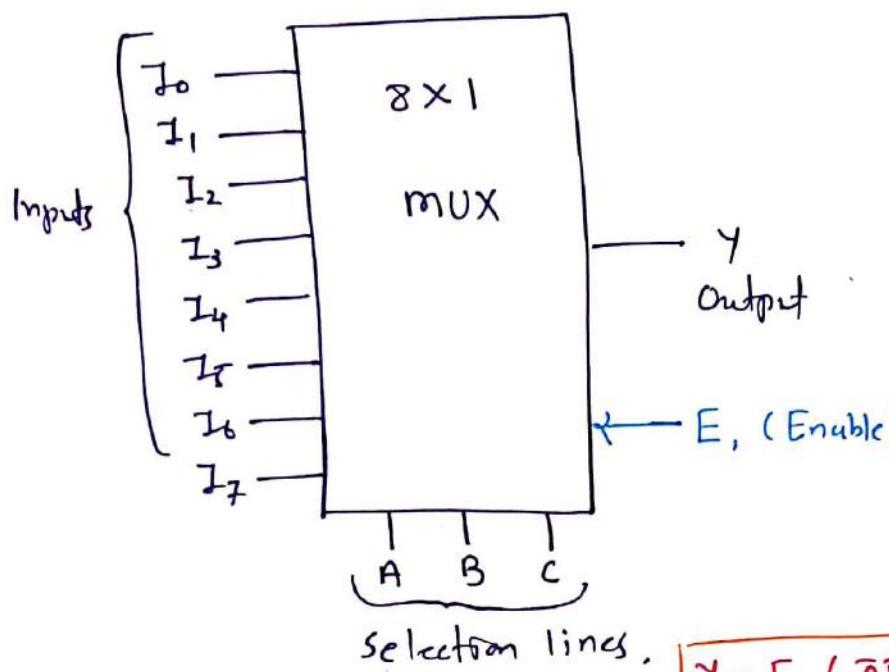
$$Y = E \cdot [\bar{S}_1 \bar{S}_0 I_0 + \bar{S}_1 S_0 I_1 + S_1 \bar{S}_0 I_2 + S_1 S_0 I_3]$$



8 X 1 MUX 120

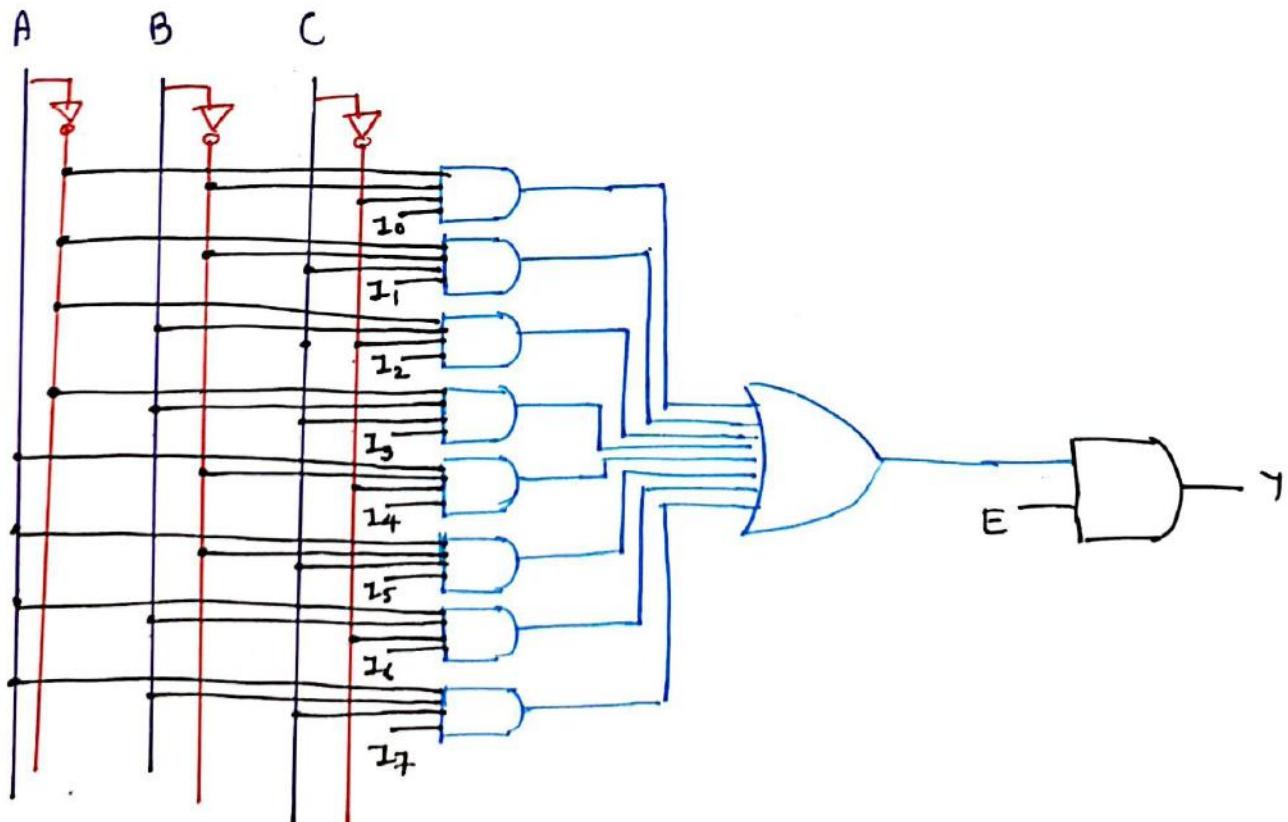
→ $[2^n \times 1]$ MUX

→ $n=3$, 3 selection lines.



E	A	B	C	Y
0	x	x	x	0
1	0	0	0	I_0
1	0	0	1	I_1
1	0	1	0	I_2
1	0	1	1	I_3
1	1	0	0	I_4
1	1	0	1	I_5
1	1	1	0	I_6
1	1	1	1	I_7

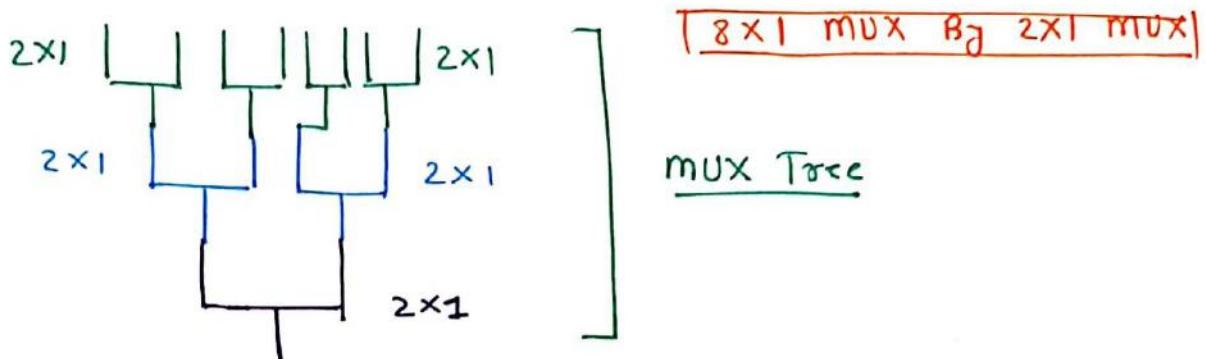
$$Y = E \cdot (\bar{A}\bar{B}\bar{C}I_0 + \bar{A}\bar{B}CI_1 + \bar{A}BCI_2 + ABCI_3 + A\bar{B}\bar{C}I_4 + A\bar{B}CI_5 + AB\bar{C}I_6 + ABCI_7)$$



MUX Tree

12

→ MUX Tree is used to obtain higher order MUX using Lower Order MUX.



1] How many 4x1 MUX is req'd to make 32x1 MUX.

$$\frac{32}{4} = \frac{8}{1}$$

$= 8 + 2 + 1$

$= 11$

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

$$\frac{2}{4} = 0.5 < 1$$

2] How many 2x1 MUX is req'd to get 64x1 MUX.

$$\frac{64}{2} = \frac{32}{1}$$

$= 32 + 16 + 8 + 4 + 2 + 1$

$= 63$

$$\frac{32}{2} = \frac{16}{1}$$

$$\frac{16}{2} = \frac{8}{1}$$

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

$$\frac{4}{2} = \frac{2}{1}$$

$$\frac{2}{2} = \frac{1}{1}$$

$$\frac{64}{8} = \frac{8}{1}$$

$= 8 + 1$

$= 9$

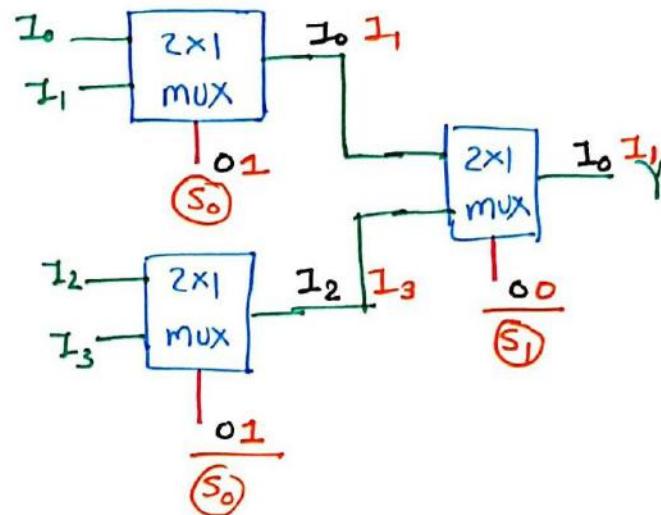
3] How many 8x1 MUX
req'd to get 64x1 MUX.

Design 4x1 MUX using 2x1 MUX. 122

→ 1st find, How many 2x1 MUX req'd to get 4x1 MUX.

$$\begin{array}{r} \frac{4}{2} = 2 \\ \hline \frac{2}{2} = 1 \end{array} \quad = 2 + 1 \\ = 3$$

S_1	S_0	Y
0	0	I_0
0	1	I_1
1	0	I_2
1	1	I_3

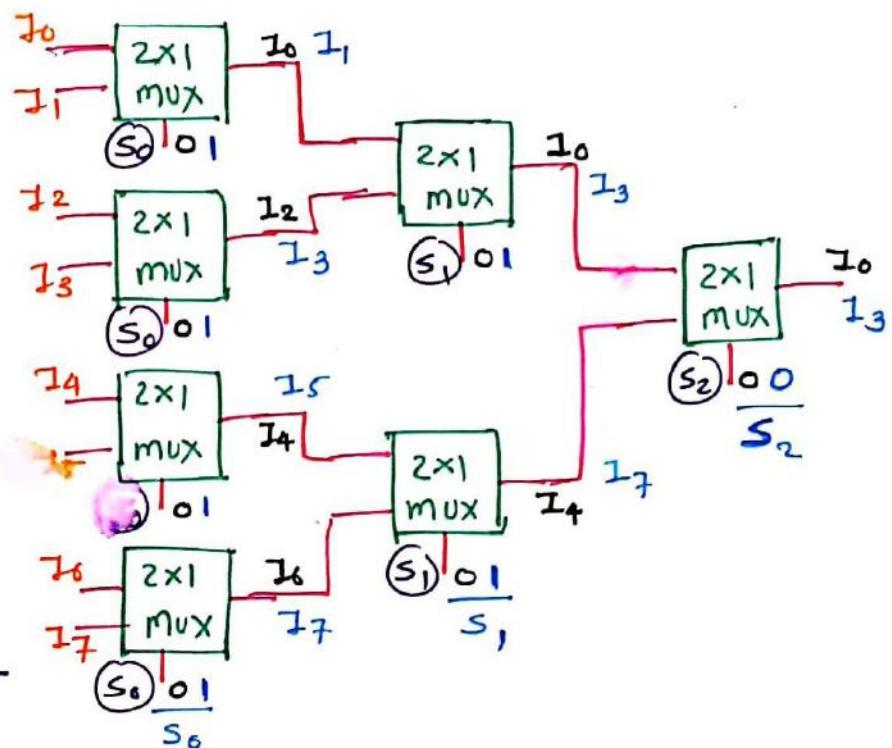


Design 8x1 MUX using 2x1 MUX 123

→ 1st Find, How many 2x1 MUX is req'd to get 8x1 MUX

$$\begin{array}{rcl} \frac{8}{2} & = & 4 \\ \hline \frac{4}{2} & = & 2 \\ \hline \frac{2}{2} & = & 1 \end{array}$$
$$= 4 + 2 + 1$$
$$= 7$$

S_2	S_1	S_0	γ
0	0	0	I_0
0	0	1	I_1
0	1	0	I_2
0	1	1	I_3
1	0	0	I_4
1	0	1	I_5
1	1	0	I_6
1	1	1	I_7



8×1 MUX using 4×1 MUX 12h

→ 1st step is to find, How many 4×1 MUX is req'd to get 8×1 MUX

$$\frac{8}{4} = 2$$

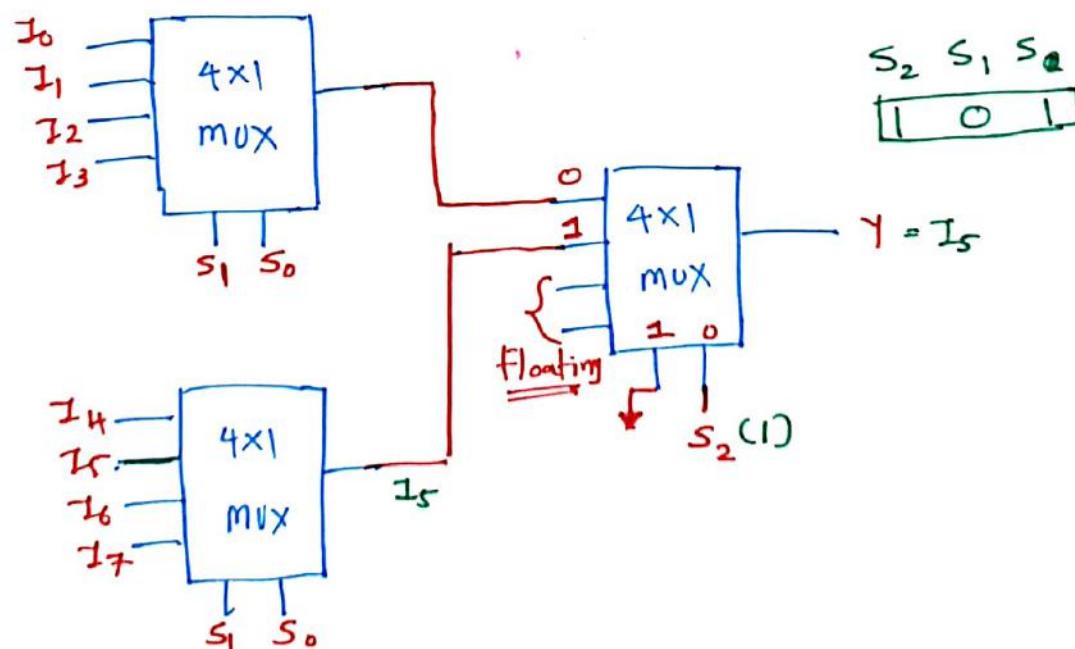
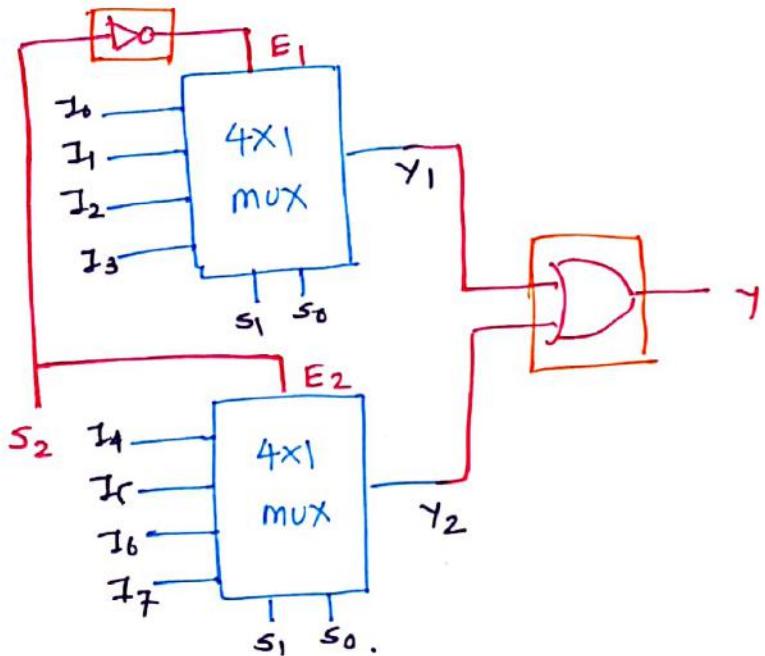
$$\frac{2}{4} = 0.5 < 1$$

$$= 2 + 0.5$$

$$= 2.5$$

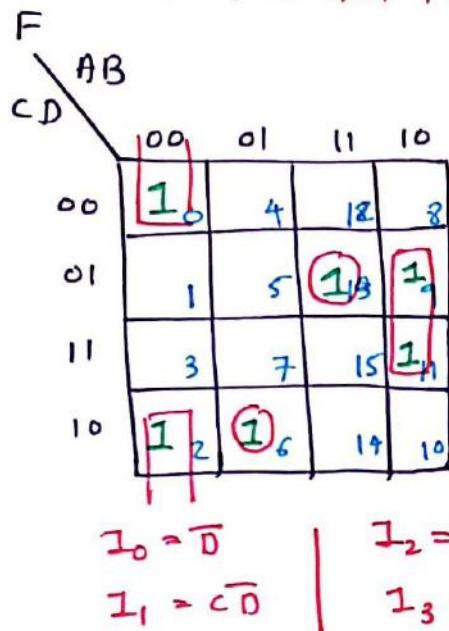
= 3 [without Additional gates]

s_2	s_1	s_0	Y
0	0	0	I ₀
0	0	1	I ₁
0	1	0	I ₂
0	1	1	I ₃
1	0	0	I ₄
1	0	1	I ₅
1	1	0	I ₆
1	1	1	I ₇

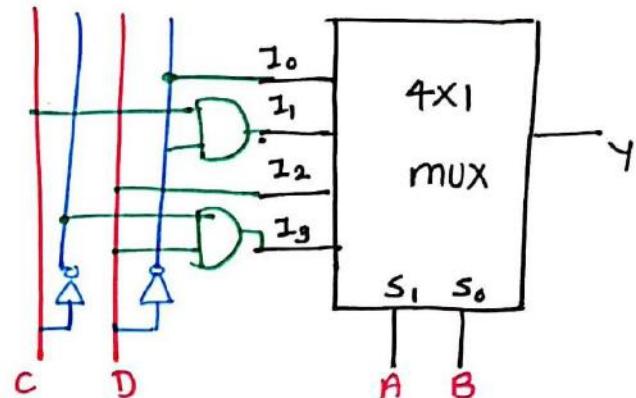


SOP Implementation by MUX 125

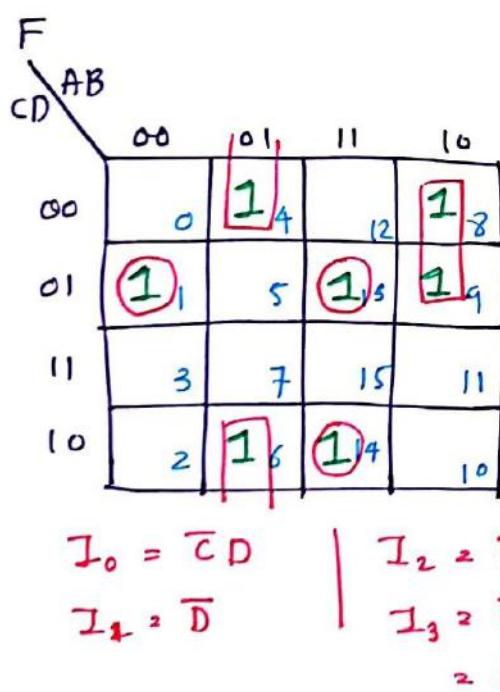
Function $F(A, B, C, D) = \sum_m(0, 2, 6, 7, 11, 13)$ make it by 4x1 MUX.



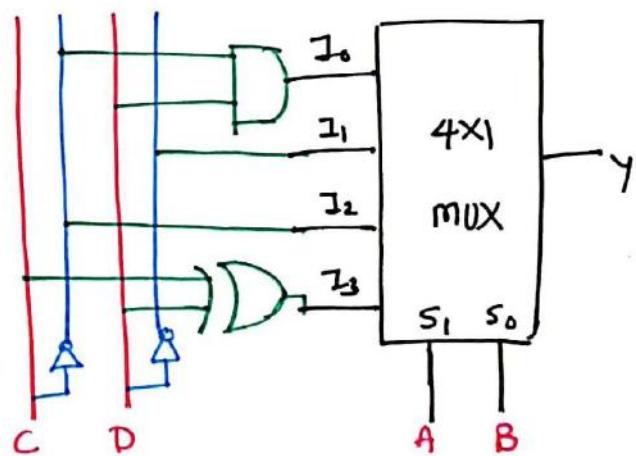
S_1	S_0	Y
0	0	I_0
0	1	I_1
1	0	I_2
1	1	I_3



Function $F(A, B, C, D) = \sum_m(1, 4, 6, 8, 9, 13, 14)$ make it by 4x1 MUX.



S_1	S_0	Y
0	0	I_0
0	1	I_1
1	0	I_2
1	1	I_3



Full Adder by 2x1 MUX

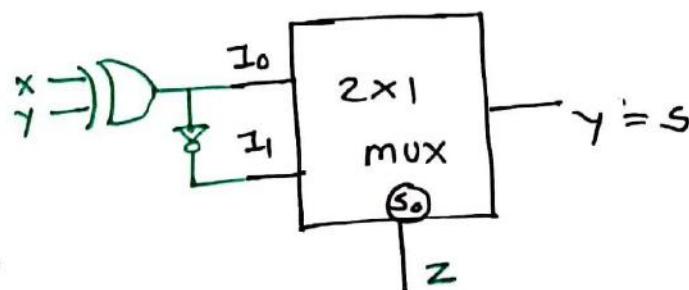
127

x	y	z	s	c_0
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

$S = xy$

z	00	01	11	10
0	0	1	1	1
1	1	1	0	0

$$\begin{array}{c|cc} S_0 & Y \\ \hline 0 & x \oplus y \\ 1 & \bar{x}y + x\bar{y} \\ 1 & \bar{x}\bar{y} + x\bar{y} \\ \hline & = \bar{x} \oplus y \end{array}$$

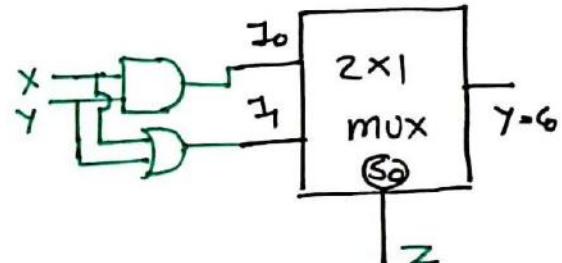


$C_0 = xy$

z	00	01	11	10
0	0	1	1	1
1	1	1	0	0

$$\begin{aligned} I_0 &= x \cdot y \\ I_1 &= \bar{y} + x \end{aligned}$$

S_0	$Y = C_0$
0	I_0
1	I_1



Full Adder by 2x1 MUX

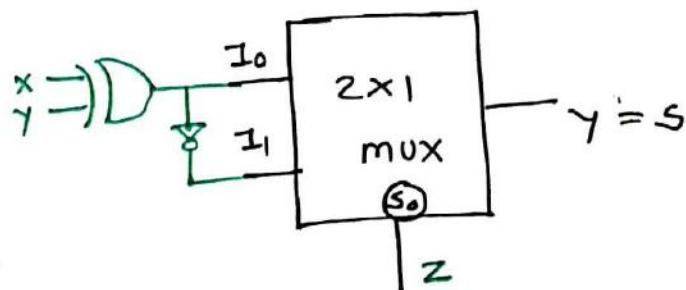
127

X	Y	Z	S	C ₀
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

$S = X \oplus Y$

S	X	Y	00	01	11	10
0	0	0	0	1	1	1
1	1	0	1	1	0	0

S ₀	Y = X \oplus Y
0	I ₀ = $\bar{X}Y + X\bar{Y}$
1	I ₁ = $\bar{X}Y + XY$ = $\overline{X \oplus Y}$



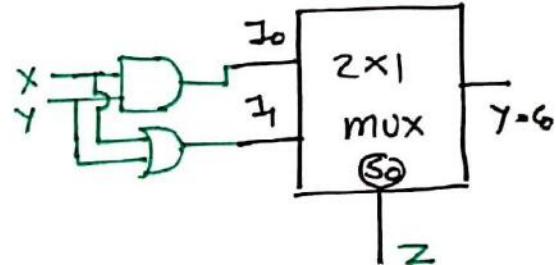
$C_0 = X \cdot Y$

C ₀	X	Y	00	01	11	10
0	0	0	0	1	1	1
1	1	0	1	1	0	0

$I_0 = X \cdot Y$

$I_1 = Y + X$

S ₀	Y = C ₀
0	I ₀
1	I ₁



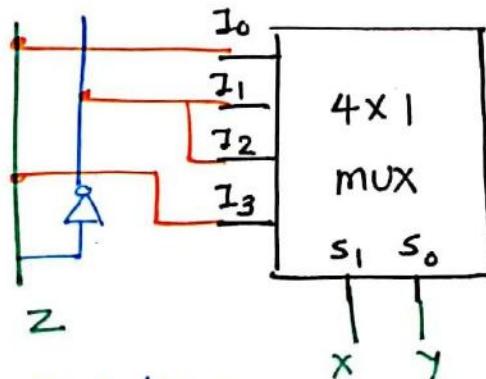
Full Adder using 4x1 MUX

X	Y	Z	S	C
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

$S = X \oplus Y$

00	01	11	10
0	(1)	(1)	(1)
1	(1)	(1)	(1)

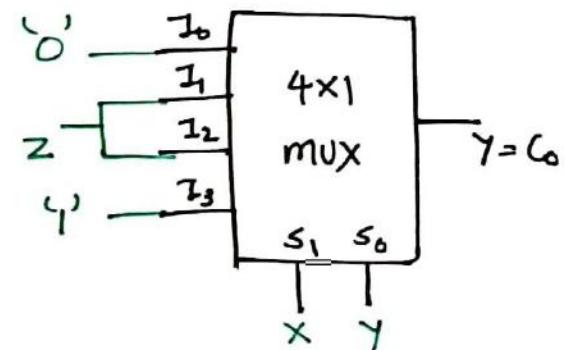
s_1, s_0	$y = s$
00	$I_0 = z$
01	$I_1 = \bar{z}$
10	$I_2 = \bar{z}$
11	$I_3 = z$



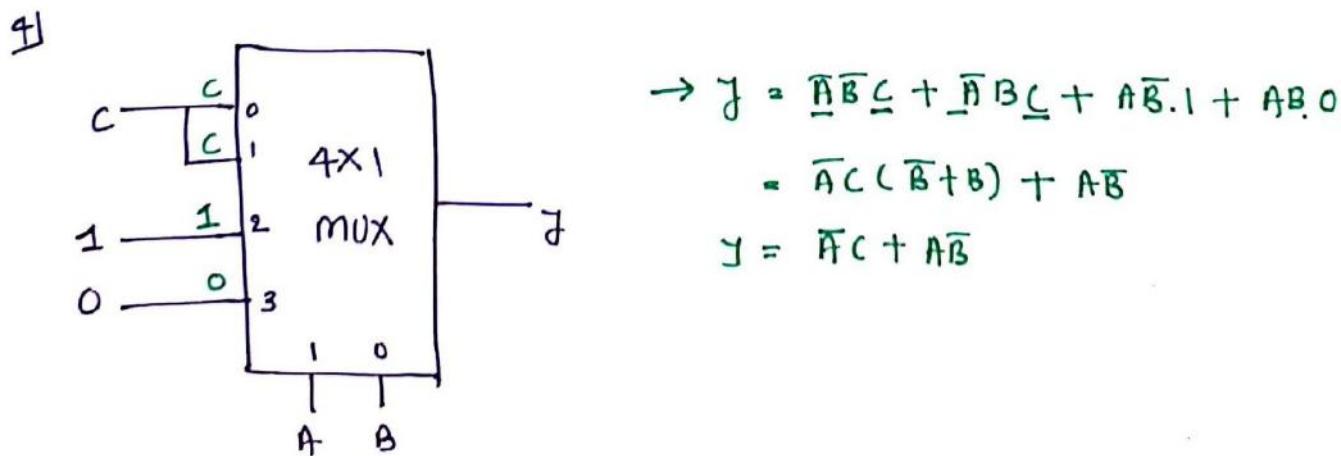
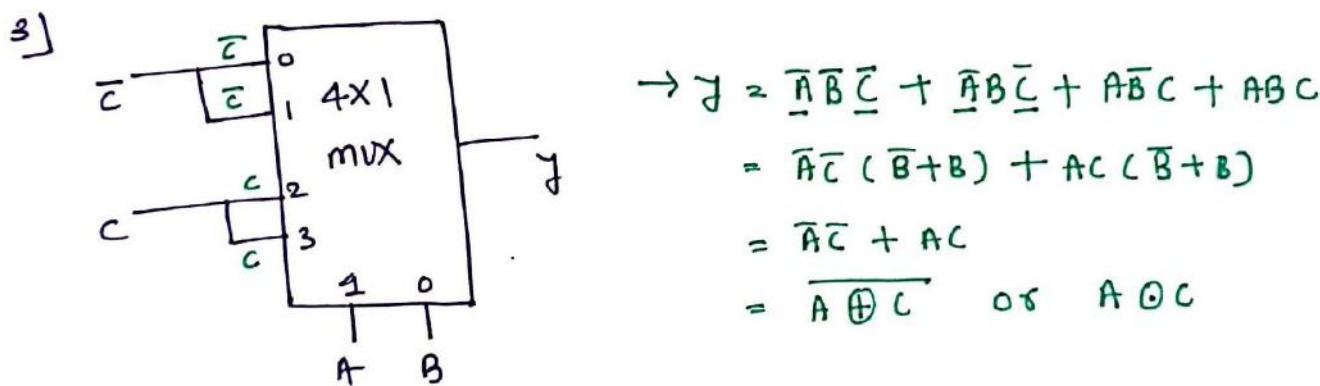
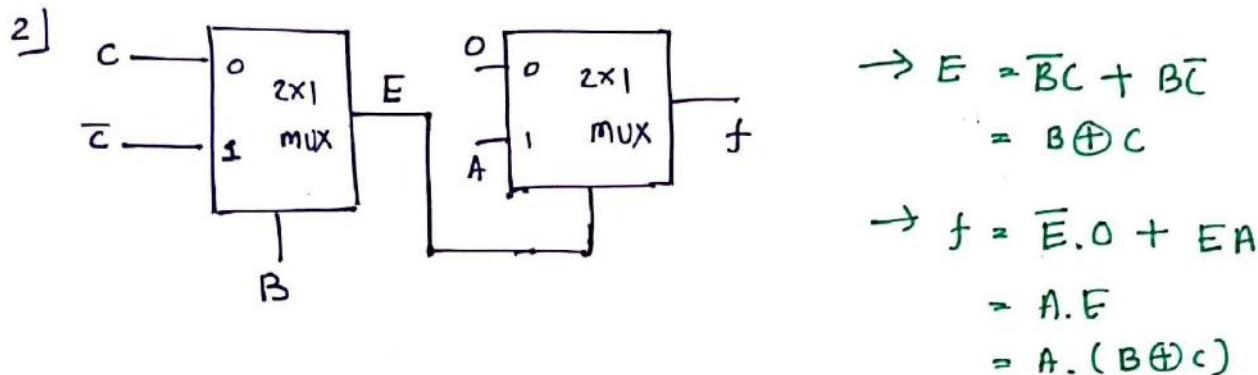
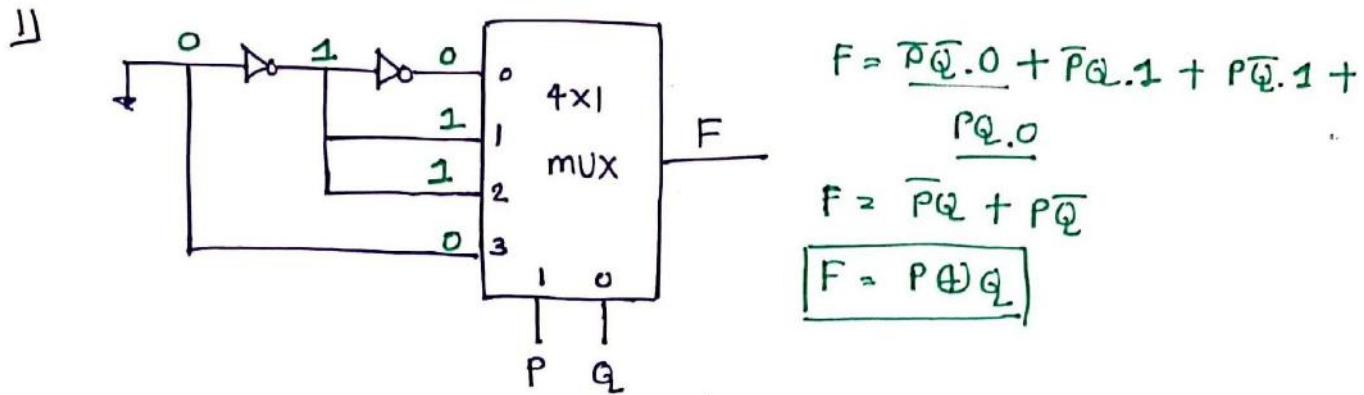
$C = X \oplus Y \oplus Z$

2	00	01	11	10
0			(1)	(1)
1	(1)	(1)	(1)	(1)

s_1, s_0	$y = c_0$
00	$I_0 = 0$
01	$I_1 = z$
10	$I_2 = \bar{z}$
11	$I_3 = 1$



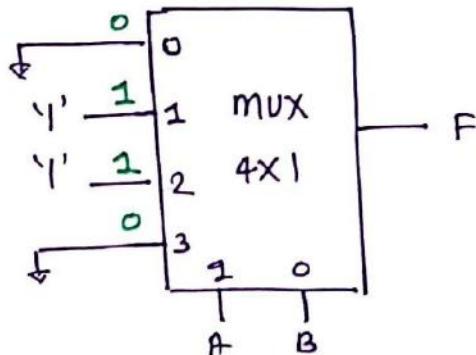
Boolean function from multiplexor circuit 128



Identification of logic gate from multiplexer circuit

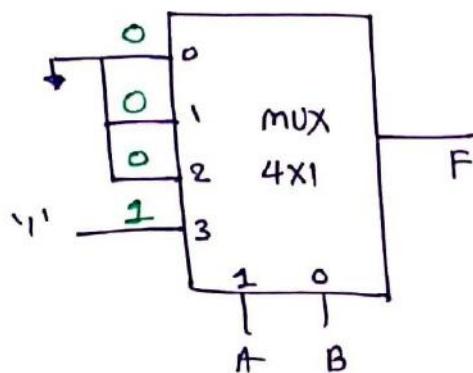
129

1)



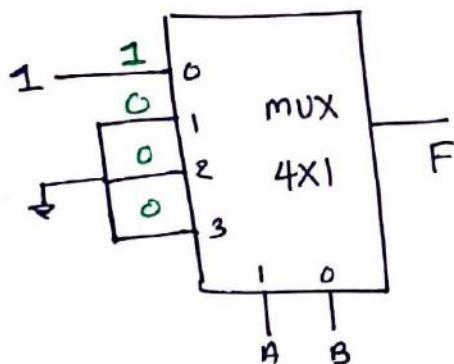
$$\begin{aligned}
 F &= \overline{A}\overline{B} \cdot 0 + \overline{A}B \cdot 1 + A\overline{B} \cdot 1 + AB \cdot 0 \\
 &= \overline{A}B + A\overline{B} \\
 &= A \oplus B \quad [\text{XOR gate}]
 \end{aligned}$$

2)



$$\begin{aligned}
 F &= \overline{A}\overline{B} \cdot 0 + \overline{A}B \cdot 0 + A\overline{B} \cdot 0 + AB \cdot 1 \\
 &= A \cdot B \\
 &= [\text{AND gate}]
 \end{aligned}$$

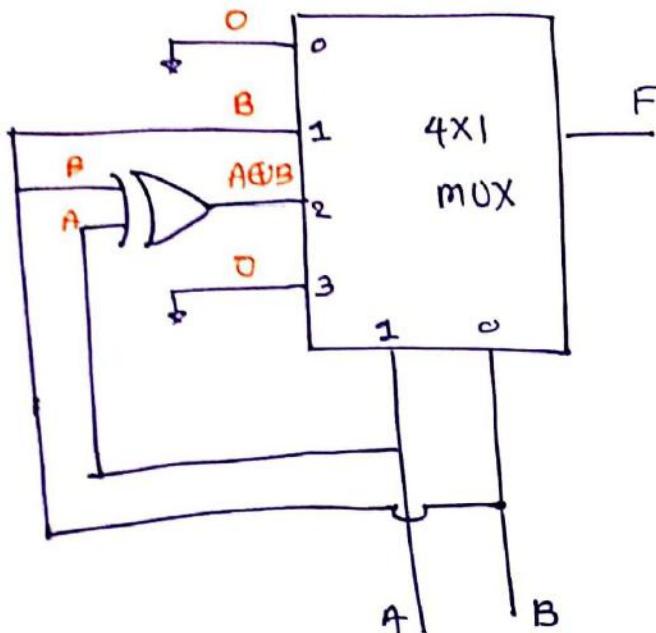
3)



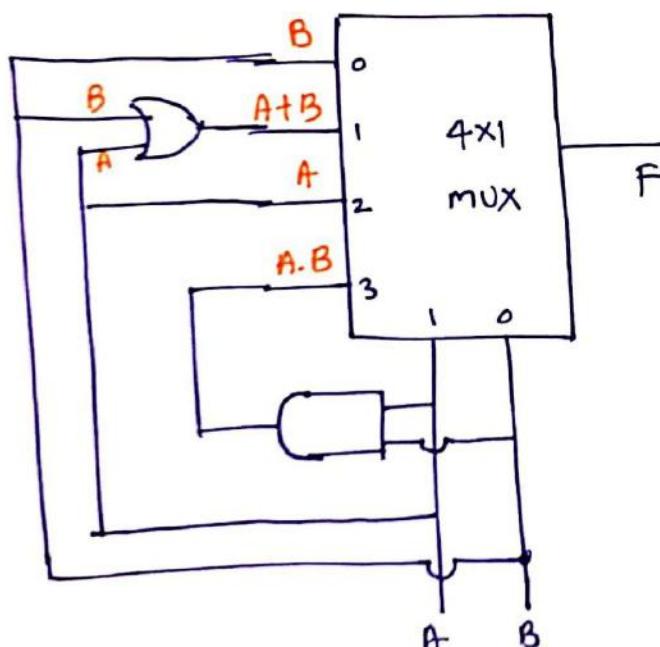
$$\begin{aligned}
 F &= \overline{A}\overline{B} \cdot 1 + \overline{A}B \cdot 0 + A\overline{B} \cdot 0 + AB \cdot 0 \\
 &= \overline{A}\overline{B} \\
 &= (\overline{A} + \overline{B}) \quad [\text{NOR gate}]
 \end{aligned}$$

Examples based on Multiplexor

13-2



$$\begin{aligned}
 F &= \overline{\overline{A}\overline{B}} \cdot 0 + \overline{A}\overline{B} \cdot B + A\overline{B} (A\overline{B} + \overline{A}B) \\
 &\quad + AB \cdot 0 \\
 &= \overline{A}B + A\overline{B}AB + A\overline{B}\overline{A}B \\
 &= \overline{A}B + A\overline{B} + 0 \\
 &= \overline{A}B + A\overline{B} \\
 &= A \oplus B
 \end{aligned}$$



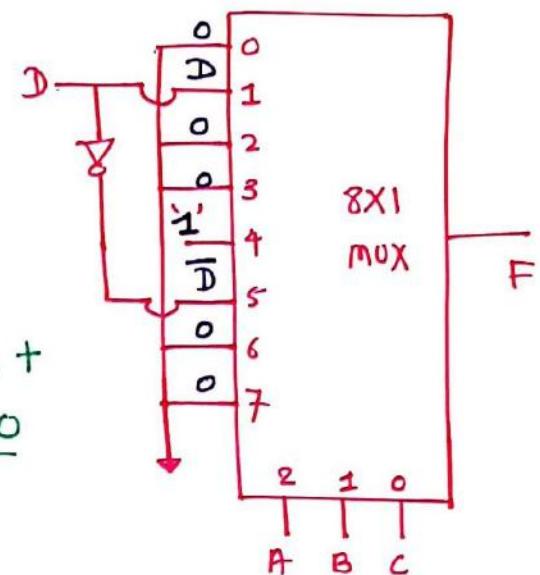
$$\begin{aligned}
 F &= \overline{\overline{A}\overline{B}} \cdot B + \overline{A}B \cdot (A+B) + A\overline{B} \cdot A + AB \cdot AB \\
 &= 0 + \overline{A}BA + \overline{A}BB + A\overline{B} + AB \\
 &= \frac{\overline{A}B}{1} + \frac{A\overline{B}}{2} + \frac{AB}{3} \\
 &= \Sigma_m (1, 2, 3) \\
 &= [0, 1, 2, 3] \\
 &= \Pi_m [0]
 \end{aligned}$$

GATE / ISRO Examples based on Multiplexer 13)

1) Function $F(A, B, C, D)$ can be expressed as

- ✓ a) $\Sigma_m(3, 8, 9, 10)$
- b) $\Sigma_m(3, 8, 10, 14)$
- c) $\Pi(0, 1, 2, 4, 5, 6, 7, 11, 12, 13, 15)$
- d) $\Pi(0, 1, 2, 4, 5, 6, 7, 10, 12, 13, 15)$

$$\begin{aligned} \rightarrow F &= \underline{\bar{ABC}.0} + \underline{\bar{ABC}.D} + \underline{\bar{ABC}.0} + \underline{\bar{ABC}.0} + \\ &\quad \underline{\bar{ABC}.1} + \underline{\bar{ABC}.\bar{D}} + \underline{ABC.0} + \underline{ABC.0} \\ &= \underline{\bar{ABC}D} + \underline{\bar{ABC}} + \underline{\bar{ABC}\bar{D}} \\ &= \underline{\underline{0011}} \quad \underline{\underline{1000}} \quad \underline{\underline{1010}} \\ &\quad 3 \quad 8, 9 \\ &= \Sigma_m(3, 8, 9, 10) \end{aligned}$$



2) Find the correct statements.

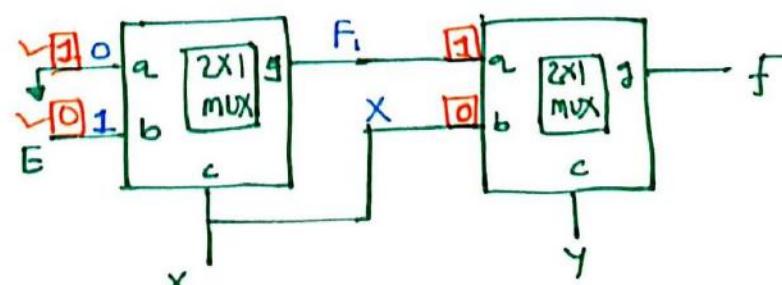
A Multiplexer.

- ✓ a) Selects one of many inputs and transmits on single output.
- ✗ b) Routes the data from single input to many of outputs.
- ✓ c) It converts Parallel data to Serial data.
- ✓ d) is a combinational circuit.

3) If function $g = ac + b\bar{c}$ then find the output f

- (a) $\times \oplus \gamma$
- (b) $XE + X\bar{Y}$
- ✓ (c) $X\bar{Y} + E\bar{X}Y$
- (d) None of these

$$\begin{aligned} - g &= \bar{c}a + cb \times \\ - g_2 &= \bar{c}b + ca \checkmark \end{aligned}$$



$$\begin{aligned} F_1 &= \bar{X}E + X.0 \\ &= \bar{X}E \end{aligned} \quad \rightarrow f = \bar{Y}X + YF_1$$

$$- \bar{Y}X + Y\bar{X}E$$

Coincidence Logic 132

- To get coincidence logic following Condⁿ should get satisfied.

Cond.ⁿ ① Inputs terminals ≥ 2 , Output terminal = 1

Cond.ⁿ ② If all V_p 's = 0, O/P = 1

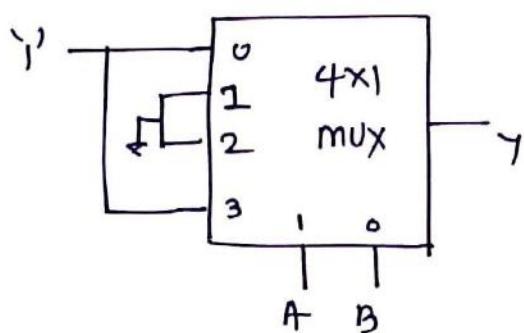
If all V_p 's = 1, O/P = 1

e.g. Two terminal XNOR gate.



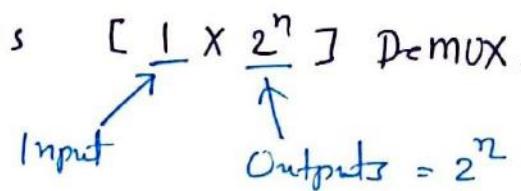
A	B	Y
0	0	1
0	1	0
1	0	0
1	1	1

e.g.

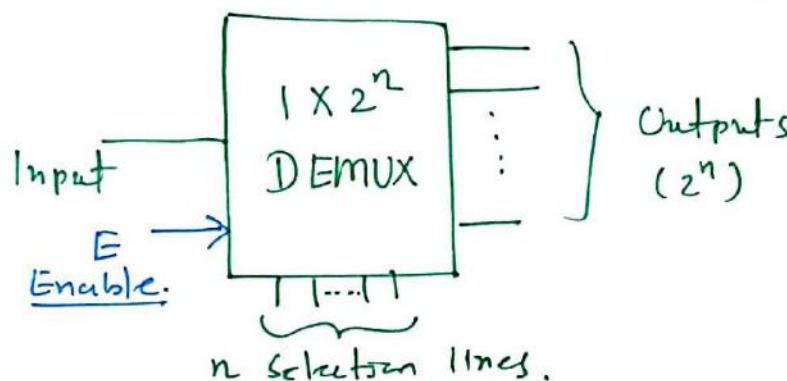


Demultiplexer and 1 to 2 Demultiplexer | 33

- It is having reverse operation to that of multiplexer.
- Basic form of Demultiplexer is $[1 \times 2^n]$ Demux.
- Demultiplexer is said to Parallel converter.



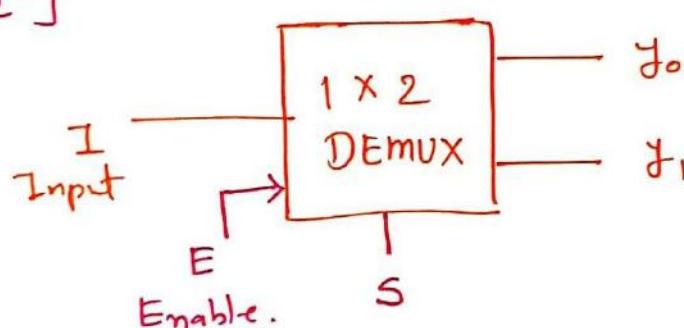
[n = No of selection lines].



- 1 to 2 Demultiplexer

- $[1 \times 2^n]$

- $n = 1$



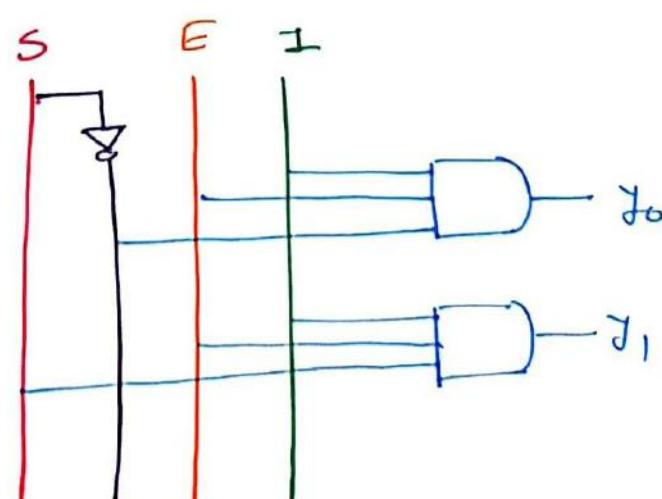
E	S	y_0	y_1
0	X	0	0
1	0	I	0
1	1	0	I

$$- y_0 = I E \bar{S}$$

$$y_1 = I E S$$

Applications

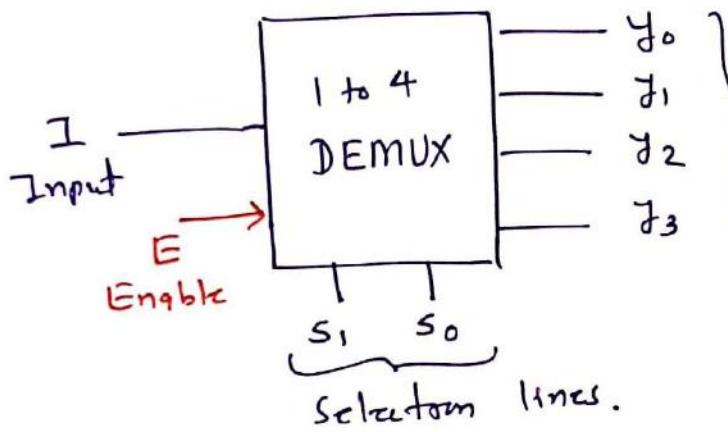
- Data Distribution
- Serial to parallel converter.
- One to many convert.



1 to 4 Demultiplexer | 34
 1 Input 4 Outputs.

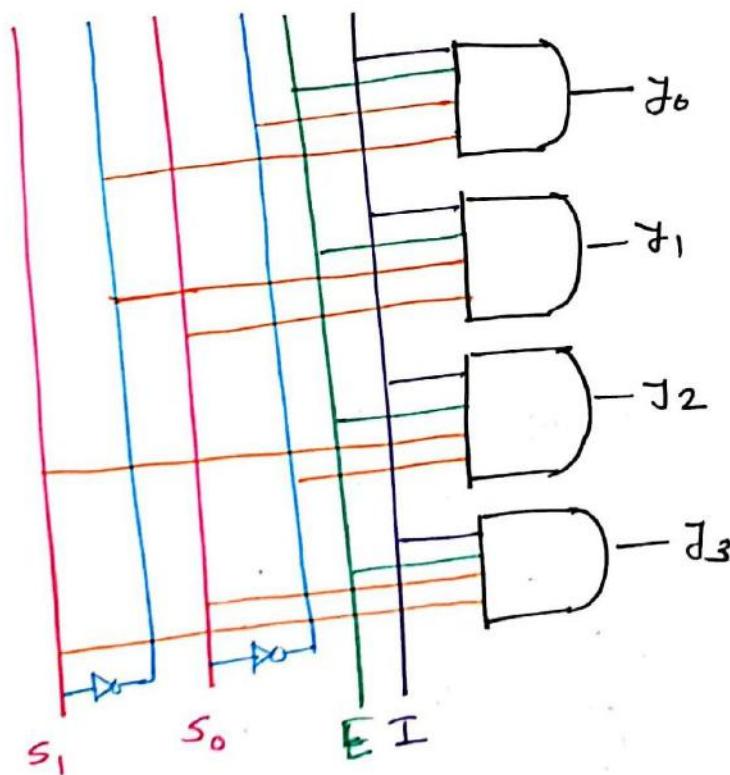
$\rightarrow [1 \times 2^n]$ Demux

$\rightarrow n=2$, Selection lines.



Outputs.

E	S ₁	S ₀	Y ₀	Y ₁	Y ₂	Y ₃
0	X	X	0	0	0	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	0	1



$$Y_0 = I E \bar{S}_1 \bar{S}_0$$

$$Y_1 = I E \bar{S}_1 S_0$$

$$Y_2 = I E S_1 \bar{S}_0$$

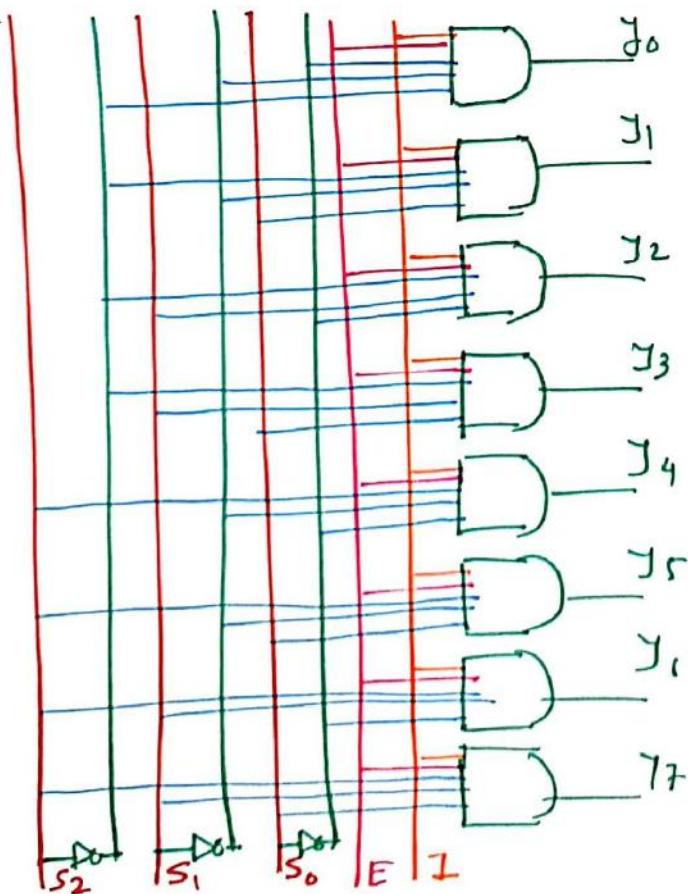
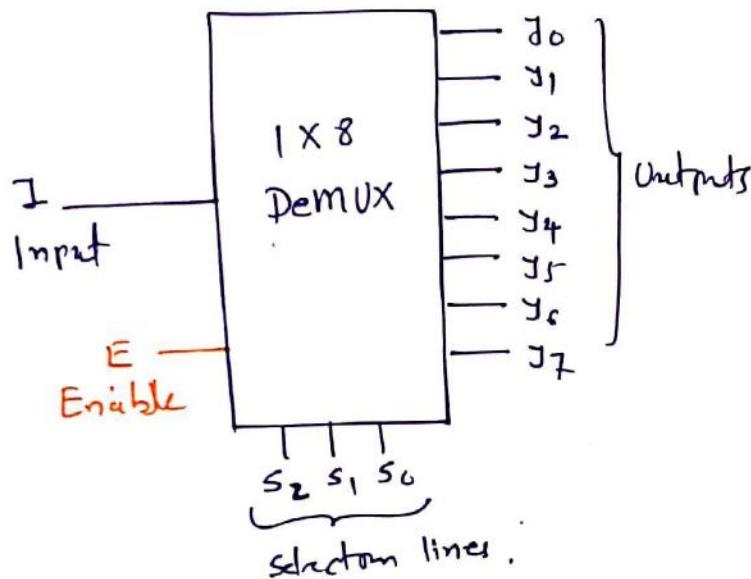
$$Y_3 = I E S_1 S_0$$

1 to 8 Demultiplexer 135

1 Input 8 Outputs.

→ [1 × 2ⁿ] DeMUX

→ n = 3, Selection lines.



E	S ₂	S ₁	S ₀	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 0 0 0 0							
1	0	0	0	1 0 0 0 0 0 0 0 0 0 0 0							
1	0	0	1	1 0 0 1 0 0 0 0 0 0 0 0							
1	0	1	0	0 0 1 0 0 0 0 0 0 0 0 0							
1	0	1	1	0 0 0 1 0 0 0 0 0 0 0 0							
1	1	0	0	0 0 0 0 1 0 0 0 0 0 0 0							
1	1	0	1	0 0 0 0 0 1 0 0 0 0 0 0							
1	1	1	0	0 0 0 0 0 0 1 0 0 0 0 0							
1	1	1	1	0 0 0 0 0 0 0 1 0 0 0 0							

$$Y_0 = IE \bar{S}_2 \bar{S}_1 \bar{S}_0$$

$$Y_1 = IE \bar{S}_2 \bar{S}_1 S_0$$

$$Y_2 = IE \bar{S}_2 S_1 \bar{S}_0$$

$$Y_3 = IE \bar{S}_2 S_1 S_0$$

$$Y_4 = IE S_2 \bar{S}_1 \bar{S}_0$$

$$Y_5 = IE S_2 \bar{S}_1 S_0$$

$$Y_6 = IE S_2 S_1 \bar{S}_0$$

$$Y_7 = IE S_2 S_1 S_0$$

Full Subtractor using 1:8 DeMUX. 136

(A - B - Bin)

	A	B	Bin	D	B_o
m_0	0	0	0	0	0
m_1	0	0	1	1	1
m_2	0	1	0	1	1
m_3	0	1	1	0	1
m_4	1	0	0	1	0
m_5	1	0	1	0	0
m_6	1	1	0	0	0
m_7	1	1	1	1	1

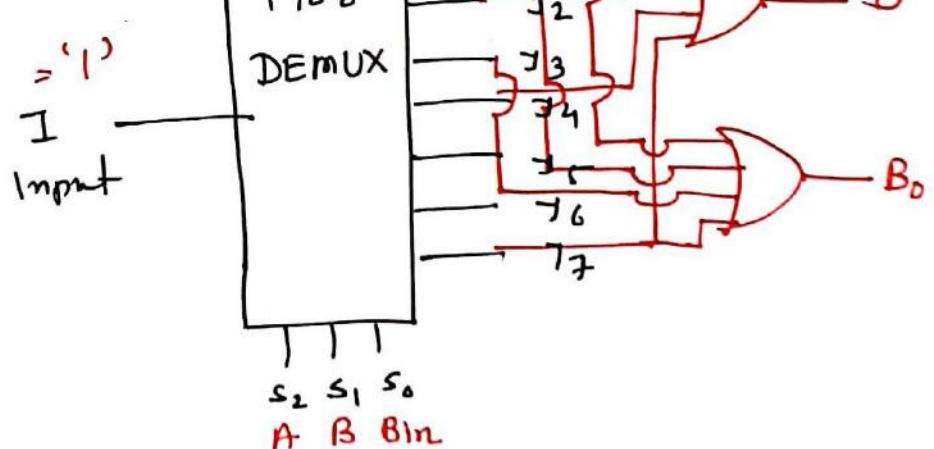
$$\rightarrow D = \sum_m (1, 2, 4, 7)$$

$$B_o = \sum_m (1, 2, 3, 7)$$

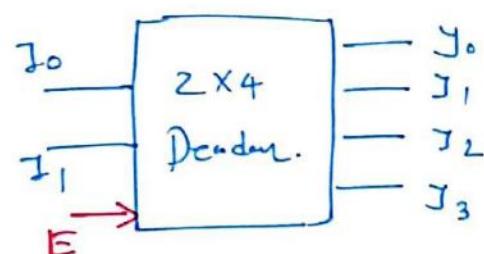
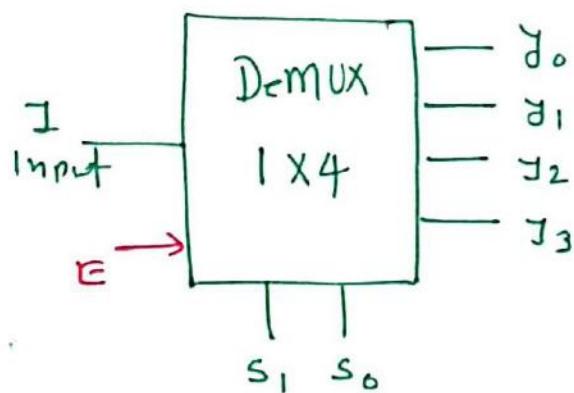
1 Input \uparrow 8 outputs $\rightarrow [1 \times 2^n]$ Demux.
 $\rightarrow n = 3$, select 3 lines.

	s_2	s_1	s_0	y_0	y_1	y_2	y_3	y_4	y_5	y_6	y_7
	0	0	0	1	0	0	0	0	0	0	0
	0	0	1	0	1	0	0	0	0	0	0
	0	1	0	0	0	1	0	0	0	0	0
	0	1	1	1	0	0	1	0	0	0	0
	1	0	0	1	0	0	0	1	0	0	0
	1	0	1	0	1	0	0	0	1	0	0
	1	1	0	0	0	0	0	0	0	1	0
	1	1	1	1	1	1	1	1	1	1	1

= '1'
I
Input



DeMUX as Decoder 137

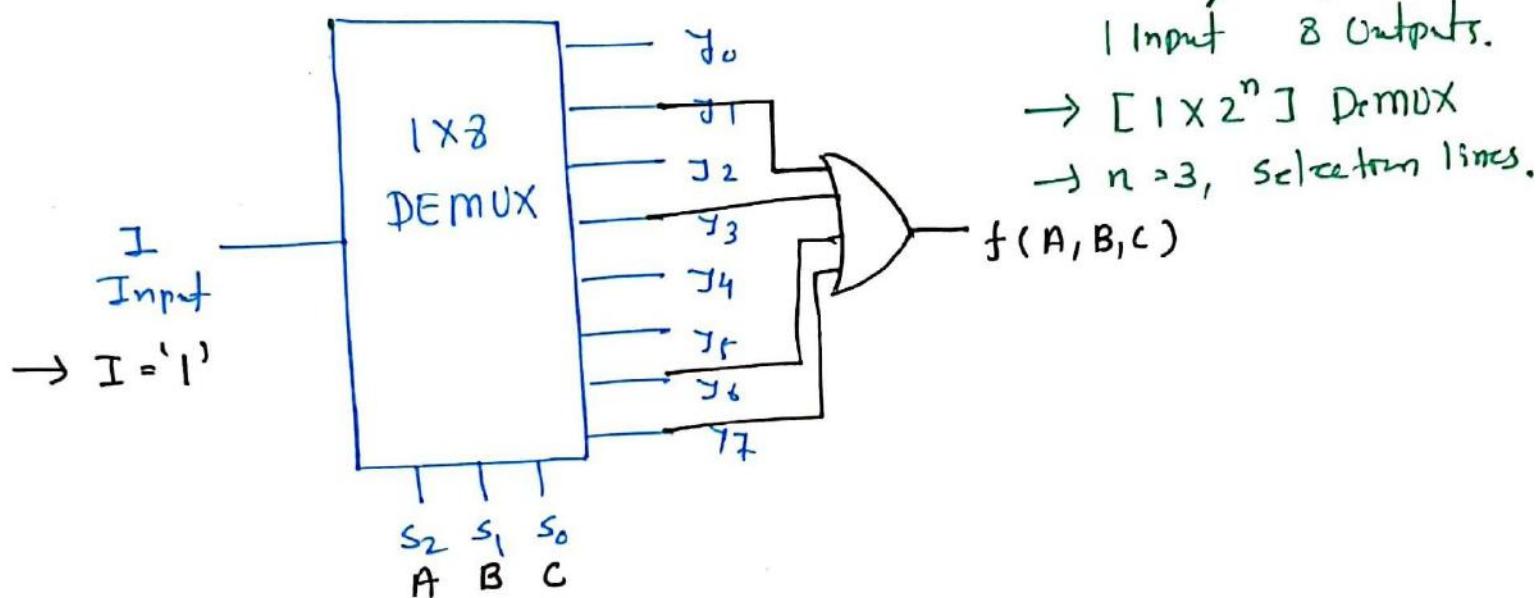


→ If $I = 1$

→ $s_1 s_0 = I_0 I_1$

Implementation of boolean expression using Demux. 138

Implement $f(A, B, C) = \sum_m(1, 3, 6, 7)$ using 1×8 Demux.



- Implement $f(A, B, C) = AB + B\bar{A} + C$ using 1×8 Demux

	AB	C	00	01	11	10
0	0	1	2	16	4	
1	1	1	3	17	15	

$$f(A, B, C) = \sum_m(1, 2, 3, 5, 6, 7)$$

- If $y = \frac{AB}{3} + \frac{\bar{A}\bar{B}}{0}$, Implement using 1×4 Demux

$$\rightarrow y = \sum_m(0, 3)$$

