July 24, 2024

Number systems

1) Binary → $(\ )_2 = 0, 1$
2) Octal → $(\ )_8 = 0$ to 7
3) Decimal → $(\ )_{10} = 0$ to 9
4) Hexadecimal → $(\ )_{16} = 0$ to 9, A – F

Q → $(19)_{10} \rightarrow (\ )_2$

2	19	
2	9	1
2	4	1
2	2	0
2	1	0
	0	1

$= (10011)_2$

Q → $(2497)_{10} = (\ )_5$

5	2497	
5	499	2
5	99	4
	0	9

$= \boxed{(942)_5}$

Ans

Q→ $(1247.78)_{10} = (\quad)_8$

↓

8	1247	
8	155	7
8	19	3
8	2	3
	0	2

$(2337.617)_8$

$= (2337.617)_8$ Ans.

$0.78 \times 8 = 6.24 = 6$

$0.24 \times 8 = 1.92 = 1$

$0.92 \times 8 = 7.36 = 7$

Q→ 1) $(243.75)_{10} \rightarrow (\quad)_3$

Q-2.) $(12332)_4 \rightarrow (\quad)_2$

Q-3) $(726.105)_8 \rightarrow (\quad)_2$

Q-4) $(7AD.C2)_{16} \rightarrow (\quad)_2$

Q-5) $(111010110101.10011)_2 \rightarrow (\quad)_4$

1.) $(243.75)_{10} = (\quad)_3$

3	243	
3	81	0
3	27	0
3	9	0
3	3	0
3	1	0
	0	1

$(100000.210)_3$

$0.75 \times 3 = 2.35 = 2$

$0.35 \times 3 = 1.05 = 1$

$0.05 \times 3 = 0.15 = 0$

2.) $(12332)_4 = (\quad)_2$

$(12332)_4 \longrightarrow (\quad)_{10} \longrightarrow (\quad)_2$

$= 1 \times 4^4 + 2 \times 4^3 + 3 \times 4^2 + 3 \times 4^1 + 2 \times 4^0$

$= 256 + 128 + 48 + 12 + 2$

$= (446)_{10}$

2	446	
2	223	0
2	111	1
2	55	1
2	27	1
2	13	1
2	6	1
2	3	0
2	1	1
	0	1

$= (110111110)_2$

3.) $(726.105)_8 = (111010110.001000101)_2$

4.) $(7AD.C2)_{16} = (011110101101.11000010)_2$

5.) $(011110101101.10011)_2$

$= (13223 1.212)_4$

Q→ $(43.12)_5 \longrightarrow (\cdot)_7$

$(43.12)_5 \rightarrow (\quad)_{10} \rightarrow (\quad)_7$

$= (4\times5^{1} + 3\times5^{0} + 1\times5^{-1} + 2\times5^{-2})$

$= 20 + 3 + 0.2 + 0.08$

$= (23.28)_{10}$

$(23.28)_{10} \rightarrow (\quad)_7$

7	23	
7	3	2
	0	3

$(32.165)_7$

$0.28\times7 = 1.96 = 1$

$0.96\times7 = 6.72 = 6$

$0.72\times7 = 5.04 = 5$

Q→ Design a warning system of a car manufacturing company which has three inputs light, day and engine.

1) Understand the problem
2) Truth table on the basis of input

Assuming for day: 0→ Night, 1→ Day; for light 0→ OFF, 1→ ON; 0→ OFF, 1→ ON

maxterm boolean		L	D	E	W	minterm boolean
LDE	m_0	0	0	0	0	$\bar{L}\bar{D}\bar{E}$
$LD\bar{E}$	m_1	0	0	1	1	$\bar{L}\bar{D}E$
$L\bar{D}E$	m_2	0	1	0	0	$\bar{L}D\bar{E}$
$L\bar{D}\bar{E}$	m_3	0	1	1	0	$\bar{L}DE$
$\bar{L}DE$	m_4	1	0	0	1	$L\bar{D}\bar{E}$
$\bar{L}D\bar{E}$	m_5	1	0	1	0	$L\bar{D}E$
$\bar{L}\bar{D}E$	m_6	1	1	0	1	$LD\bar{E}$
$\bar{L}\bar{D}\bar{E}$	m_7	1	1	1	1	LDE

3) Boolean expression comprising of AND (.), OR (+), NOT (¯ or

For SOP (sum of products) [Minterm]

↳ The value of all products to be summed/added should '1'.

Therefore '0' is taken as compliment whereas '1' is as it is

For POS (product of sum) [Maxterm].

↳ The value of all sums to be multiplied has to be '0'. Therefore '0' is taken as it is whereas '1' is compliment

As per SOP: $F(LDE) = m_1 + m_4 + m_6 + m_7 = \Sigma(1, 4, 6, 7)$

As per POS: $F(LDE) = M_0 + M_2 + M_3 + M_5 = \Pi(0, 2, 3, 5)$

Boolean [Taking minterms]:-

$= \bar{L}\bar{D}E + L\bar{D}\bar{E} + LD\bar{E} + LDE$

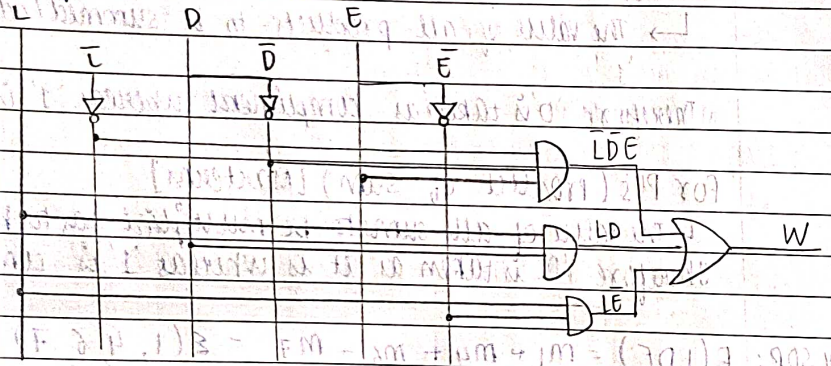
* K-map follows gray code [change in only one bit]

4) Minimise the boolean expression
→ By boolean algebra
→ By K-map

L \ DE	00	01	11	10
0		1		
1	1		1	1

$W = \bar{L}\bar{D}E + LD + L\bar{E}$

5) Implement the circuit



Q- Bank locker system : Bank manager, cashier 1, cashier 2
locker opens only if manager is present.

Truth Table :-

M	A	B	L
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

Logically,
simplified Boolean expression :-
$= M(A+B)$

Q- Design a system having three inputs where a system will always give a high signal when the inputs are odd.

ly 25, 2024

K-Map

1) F(A, B) No. of cells = $2^2 = 4$

A \ B	0	1
0	0	1
1	2	3

	AB
m_0	00
m_1	01
m_2	10
m_3	11

2) F(A, B, C)

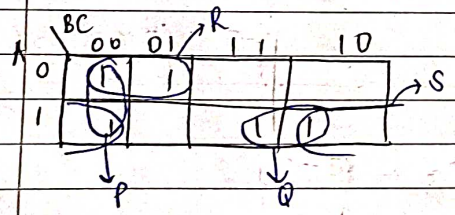
A \ BC	00	01	11	10
0	0	1	3	2
1	4	5	7	6

Implicant : Group of 1s

Prime implicant : largest no. of group of 1s.

Essential prime implicants : A group where one operand has not been paired with any other group

Redundant prime implicants are ignored.



$P = \Sigma(0, 4)$

$Q = \Sigma(7, 6)$

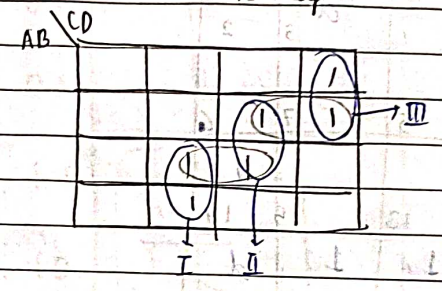
$R = \Sigma(0, 1)$

$S = \Sigma(4, 6)$

Redundant prime implicant.

Essential Prime implicant : Group of 1s where atleast single 1 cannot be combined with any other group.

Q→ Four variable K-Map



* Prime implicants → I, II, III
* Essential prime → I, III
* Reducdant → II (or non-essential prime implicant)

Q→ $F(A, B, C, D) = \Sigma m(0, 2, 3, 7, 5, 11, 13, 15)$

AB \ CD	00	01	11	10
00	1 (0)	1 (1)	1 (3)	1 (2)
01	(4)	1 (5)	1 (7)	(6)
11	(12)	1 (13)	1 (15)	(14)
10	(8)	(9)	1 (11)	(10)

I, II, III

$F = BD + CD + \overline{A}\,\overline{B}\,\overline{D}$

Essential prime implicant : all 3

1 = A

0 = A

Q → $f(A, B, C, D) = \Sigma m(0, 2, 5, 7, 9, 11) + d(3, 8, 10, 12, 14)$

AB \ CD	00	01	11	10
00	1 (0)	(1)	d (3)	1 (2)
01	(4)	1 (5)	1 (7)	(6)
11	d (12)	(13)	(15)	d (14)
10	d (8)	1 (9)	1 (11)	d (10)

$F = A\bar{B} + \bar{B}\bar{D} + \bar{A}BD$

Five variable K-map

For five variable K-map, make 2 maps for A = 0 and 1

BC \ DE	00	01	11	10
00	0	1	3	2
01	4	5	7	6
11	12	13	15	14
10	8	9	11	10

A = 0

BC \ DE	00	01	11	10
00	16	17	19	18
01	20	21	23	22
11	28	29	31	30
10	24	25	27	26

A = 1

July 30, 2024

Q → $Y = m_1 + m_3 + m_5 + m_7 + m_8 + m_9 + m_{12} + m_{13}$
$= \Sigma(1, 3, 5, 7, 8, 9, 12, 13)$

AB \ CD	00	01	11	10
00	(0)	1 (1)	1 (3)	(2)
01	(4)	1 (5)	1 (7)	(6)
11	1 (12)	1 (13)	(15)	(14)
10	1 (8)	1 (9)	(11)	(10)

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

Q → $ABCD + A\bar{B}\bar{C}\bar{D} + A\bar{B}\bar{C} + \bar{A}B$. simplify using K-map

→ canonical expression (where all the inputs have been used either directly or complemented).

Otherwise the expression is minimal expression.

{Eg: The above is minimal}

Eg: $Y = A + B$

This expression is both → minimal SOP/standard SOP
→ canonical POS

: $Y = AB$

→ minimal POS
→ canonical SOP

Solu → $ABCD + A\bar{B}\bar{C}\bar{D} + A\bar{B}C(D+\bar{D}) + AB(C+\bar{C})(D+\bar{D})$

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

$= ABCD + A\bar{B}\bar{C}\bar{D} + A\bar{B}CD + A\bar{B}C\bar{D} +$

$ABCD + ABC\bar{D} + AB\bar{C}D + AB\bar{C}\bar{D}$

$= ABCD + A\bar{B}\bar{C}\bar{D} + A\bar{B}CD + A\bar{B}C\bar{D} + ABC\bar{D} + AB\bar{C}D + AB\bar{C}\bar{D}$

1111 1000 1011 1010 1110 1101 1100

$= m_{15} + m_8 + m_{11} + m_{10} + m_{14} + m_{13} + m_{12}$

$= \Sigma(15, 8, 11, 10, 14, 13, 12)$

AB \ CD	00	01	11	10
00	0		3	2
01	4	5	7	6
11	1 12	1 13	1 15	1 14
10	8	9	11	10

$= \boxed{A\bar{D} + AB + AC}$ Any

For 5 variable K-map

Q→ $F = \Sigma(1, 5, 6, 7, 9, 13, 14, 15, 17, 18, 19, 21, 22, 23, 25, 29)$

BC \ DE	00	01	11	10
00	0	1 1	3	2
01	4	5	1 7	1 6
11	12	1 13	1 15	1 14
10	8	1 9	11	10

A = 0

BC \ DE	00	01	11	10
00	16	1 17	1 19	1 18
01	20	1 21	1 23	1 22
11	28	1 29	31	30
10	24	1 25	27	26

A = 1

$= \bar{D}E + \bar{A}CD + A\bar{B}D$

Q→ $F(A,B,C,D,E) = \Sigma m(0, 1, 4, 5, 16, 17, 21, 25, 29)$

BC \ DE	00	01	11	10
00	1 0	1 1	3	2
01	1 4	1 5	7	6
11	12	13	15	14
10	8	9	11	10

A = 0

BC \ DE	00	01	11	10
00	1 16	1 17	19	18
01	20	1 21	23	22
11	26	1 29	31	30
10	24	1 25	27	26

A = 1

$F = \bar{B}\bar{D} + \bar{A}\bar{B}D + A\bar{D}E$

Q→ $F(P,Q,R,S,T) = \Sigma(0, 2, 4, 7, 8, 10, 12, 16, 18, 20, 23, 24, 25, 26, 27, 28)$

BC \ DE	00	01	11	10
00	1 0	1 1	3	1 2
01	1 4	5	1 7	6
11	1 12	13	15	14
10	1 8	9	11	1 10

A = 0

BC \ DE	00	01	11	10
00	1 16	17	19	1 18
01	1 20	21	1 23	22
11	1 28	29	31	30
10	1 24	1 25	1 27	1 26

A = 1

$F = \bar{D}\bar{E} + \bar{B}CDE + \bar{C}\bar{E} + AB\bar{C}$

July 31, 2024

Q→ $Y = \Sigma m(1, 5, 7, 13, 14, 15, 17, 18, 21, 22, 25, 29) + \Sigma d(6, 9, 19, 23, 30)$

BC \ DE	00	01	11	10
00	0	1 1	3	2
01	4	1 5	1 7	d 6
11	12	1 13	1 15	1 14
10	8	d 9	11	10

A = 0

BC \ DE	00	01	11	10
00	16	1 17	d 19	1 18
01	20	1 21	d 23	1 22
11	28	1 29	31	d 30
10	24	1 25	27	26

A = 1

$Y = \bar{D}E + \bar{A}CD + A\bar{B}D$

Q→ $Y(PQRST) = \Sigma m(0, 2, 4, 7, 8, 10, 12, 16, 18, 20, 23, 24, 25, 26, 27, 28)$

Both SOP and POs:-

SOP:-

QR \ ST	00	01	11	10
00	1 0	1	3	1 2
01	1 4	5	1 7	6
11	1 12	13	15	14
10	1 8	9	11	1 10

P = 0

QR \ ST	00	01	11	10
00	1 16	17	19	1 18
01	1 20	21	1 23	22
11	1 28	29	1 31	30
10	1 24	1 25	1 27	1 26

P = 1

$Y = \bar{S}\bar{T} + \bar{R}\bar{T} + PQ\bar{R} + \bar{Q}RST$

POs:- (1, 3, 5, 6, 9, 11, 13, 14, 15, 17, 19, 21, 22, 29, 30, 31)

Q+R \ S+T	00	01	11	10
00	0	0 1	0 3	2
01	4	0 5	7	0 6
11	12	0 13	0 15	0 14
10	8	0 9	0 11	10

P = 0

QR \ S+T	00	01	11	10
00	16	0 17	0 19	18
01	20	0 21	23	0 22
11	28	0 29	0 31	0 30
10	24	25	27	26

P = 1

~~$Y = (\bar{Q}+\bar{R}+\bar{T})\cdot(\bar{Q}+\bar{S}+\bar{T})\cdot(R+S+\bar{T})\cdot(Q+R+T) + (\bar{P}$~~

$Y = (Q + R + \bar{T})\cdot(Q + S + \bar{T})\cdot(\bar{R} + \bar{S} + T)\cdot(\bar{Q} + \bar{R} + \bar{T}) + (P + \bar{Q} + \bar{T})$

Q- $F(A,B,C,D) = \Sigma m(0,1,2,4,5,6,10,13,14,18,21,22,24,26,29,30)$

BC \ DE	00	01	11	10
00	1 0	1 1	3	1 2
01	1 4	1 5	7	1 6
11	12	1 13	15	1 14
10	8	9	11	1 10

A = 0

BC \ DE	00	01	11	10
00	16	17	19	1 18
01	20	1 21	23	1 22
11	28	1 29	31	1 30
10	1 24	25	27	1 26

A = 1

$Y = DE + \bar{A}\bar{B}\bar{D} + C\bar{D}E + AB\bar{C}\bar{E}$

AB \ CD	00	01	11	10
00	1 0	1 1	1 3	
01	4	5	1 7	6
11	12	13	1 15	14
10	1 8	9	1 11	10

$CD + \bar{B}\bar{C}$

Quine-McCluskey method or Tabular method

$Y(A,B,C,D) = \Sigma m(0,1,3,7,8,9,11,15)$

Step 1:

Decimal	Minterm	Binary A	B	C	D
0	m0	0	0	0	0
1	m1	0	0	0	1
3	m3	0	0	1	1
7	m7	0	1	1	1
8	m8	1	0	0	0
9	m9	1	0	0	1
11	m11	1	0	1	1
15	m15	1	1	1	1

Step 2: Group minterm and its binary (based on no of 1s)

Group	Minterm	Binary A	B	C	D
0	m0	0	0	0	0
1	m1	0	0	0	1
	m8	1	0	0	0
2	m3	0	0	1	1
	m9	1	0	0	1
3	m7	0	1	1	1
	m11	1	0	1	1
4	m15	1	1	1	1

matching means change of a single bit.

Step 3

Group	Matched Pairs	A	B	C	D	
0 & 1	m_0, m_1	0	0	0	_	✓
	m_0, m_8	_	0	0	0	✓
1 & 2	m_1, m_3	0	0	_	1	✓
	m_1, m_9	_	0	0	1	✓
	m_8, m_9	1	0	0	_	✓
2 & 3	m_3, m_7	0	_	1	1	✓
	m_3, m_{11}	_	0	1	1	✓
	m_9, m_{11}	1	0	_	1	✓
3 & 4	m_7, m_{15}	_	1	1	1	✓
	m_{11}, m_{15}	1	_	1	1	✓

Step 4:

Group	Matched term/Pairs	A	B	C	D	
0	m_0, m_1, m_8, m_9	_	0	0	_	$\bar{B}\bar{C}$
	$m_0\ m_8\ m_1\ m_9$	_	0	0	_	
1	$m_1\ m_3\ m_9\ m_{11}$	_	0	_	1	
	$m_1\ m_9\ m_3\ m_{11}$	_	0	_	1	$\bar{B}D$
2	$m_3\ m_7\ m_{11}\ m_{15}$	_	_	1	1	CD
	$m_3\ m_{11}\ m_7\ m_{15}$	_	_	1	1	

Prime implicants: $\bar{B}\bar{D}$, $\bar{B}D$, CD.

	Prime implicant (minterm)	0	1	3	7	8	9	11	15
✓	$\bar{B}\bar{C}$ ($m_0\ m_1\ m_8\ m_9$)	X	X			X	X		
	$\bar{B}D$ ($m_1\ m_3\ m_9\ m_{11}$)		X	X			X	X	
✓	CD ($m_3\ m_7\ m_{11}\ m_{15}$)			X	X			X	X
		✓			✓	✓			✓

Essential prime implicants: $\bar{B}\bar{C} + CD$

Cause at 1 value not being used twice eg: 0, 7, 8, 15 in this example

~~Cella 0, 2, 3, 5, 7, 11, 13, 15~~

Q→ $Y = \Sigma m(0, 2, 3, 5, 7, 11, 13, 15)$

Decimal	Minterm	A (8)	B (4)	C (2)	D (1)
0	m_0	0	0	0	0
2	m_2	0	0	1	0
3	m_3	0	0	1	1
5	m_5	0	1	0	1
7	m_7	0	1	1	1
11	m_{11}	1	0	1	1
13	m_{13}	1	1	0	1
15	m_{15}	1	1	1	1

Group	Minterm	A	B	C	D
0	m_0	0	0	0	0
1	m_2	0	0	1	0
2	m_3	0	0	1	1
	m_5	0	1	0	1
3	m_7	0	1	1	
	m_{11}	1	0	1	
	m_{13}	1	1	0	
	m_{15}	1	1	1	

Group	Matched Pairs	A B C D
0 & 1	m_0 m_2	0 0 – 0

August 1, 2024

Q7 $F(A,B,C,D) = \Sigma(1,2,3,7,8,9,10,11,14,15)$ by Tabular/Quine Mccluskey method.

Step 1:

Decimal	Minterm	A	B	C	D
1	m_1	0	0	0	1
2	m_2	0	0	1	0
3	m_3	0	0	1	1
7	m_7	0	1	1	1
8	m_8	1	0	0	0
9	m_9	1	0	0	1
10	m_{10}	1	0	1	0
11	m_{11}	1	0	1	1
14	m_{14}	1	1	1	0
15	m_{15}	1	1	1	1

Step 2:

Group	Minterm	A	B	C	D
1	m_1	0	0	0	1 ✓
	m_2	0	0	1	0 ✓
	m_8	1	0	0	0
2	m_3	0	0	1	1 ✓
	m_9	1	0	0	1 ✓
	m_{10}	1	0	1	0 ✓
3	m_7	0	1	1	1 ✓
	m_{11}	1	0	1	1 ✓
	m_{14}	1	1	1	0 ✓
4	m_{15}	1	1	1	1 ✓

Step 3:

Group	matched pairs	A	B	C	D
1 & 2	m_1, m_3 ✓	0	0	–	1
	m_1, m_9 ✓	–	0	0	1
	m_2, m_3 ✓	0	0	1	–
	m_2, m_{10} ✓	–	0	1	0
	$m_8 m_9$ ✓	1	0	0	–
	$m_8 m_{10}$ ✓	1	0	–	0
2 & 3	m_3, m_7 ✓	0	–	1	1
	m_3, m_{11} ✓	–	0	1	1
	m_9, m_{11} ✓	1	0	–	1
	m_{10}, m_{11} ✓	1	0	1	–
	m_{10}, m_{14}	1	–	1	0
3 & 4	m_7, m_{15} ✓	–	1	1	1
	m_{11}, m_{15} ✓	1	–	1	1
	m_{14}, m_{15} ✓	1	1	1	–

Step 4:

Group	matched pairs	A	B	C	D
1	m_1, m_3, m_9, m_{11}	–	0	–	1
	m_1, m_9, m_3, m_{11}	–	0	–	1
2	m_2, m_3, m_{10}, m_{11}	–	0	1	–
	m_2, m_{10}, m_3, m_{11}	–	0	1	–
3	m_8, m_9, m_{10}, m_{11}	1	0	–	–
	m_8, m_{10}, m_9, m_{11}	1	0	–	–
4	m_3, m_7, m_{11}, m_{15}	–	–	1	1
	m_3, m_{11}, m_7, m_{15}	–	–	1	1
5	$m_{10}, m_{11}, m_{14}, m_{15}$	1	–	1	–
	$m_{10}, m_{14}, m_{11}, m_{15}$	1	–	1	–

Prime implicants: $\bar{B}D$, $\bar{B}C$, $A\bar{B}$, CD, AC

Prime implicants	1	2	3	7	8	9	10	11	14	15
$\bar{B}D$ ($m_1 m_3 m_9 m_{11}$)	X		X			X		X		
$\bar{B}C$ ($m_2 m_3 m_{10} m_{11}$)		X	X				X	X		
$A\bar{B}$ (m_8, m_9, m_{10}, m_{11})					X	X	X	X		
CD ($m_3 m_7 m_{11} m_{15}$)			X	X				X		X
AC ($m_{10} m_{11} m_{14} m_{15}$)							X	X	X	X
	✓	✓	–	✓	✓				✓	

$Y = \bar{B}D + \bar{B}C + CD + A\bar{B} + AC$

AB \ CD	00	01	11	10
00	0	1 1	1 3	1 2
01	4	5	1 7	6
11	12	13	1 15	1 14
10	1 8	1 9	1 11	1 10

$CD + A\bar{B} + AC + \bar{B}C + \bar{B}D$

August 6, 2024

Q→ $F(ABCD) = \Sigma m(0,6,8,13,14) + d(2,4,10)$ using tabular method.

Step 1:

Decimal	Minterm	A (8)	B (4)	C (2)	D (1)
0	m_0	0	0	0	0
2	m_2	0	0	1	0
4	m_4	0	1	0	0
6	m_6	0	1	1	0
8	m_8	1	0	0	0
10	m_{10}	1	0	1	0
13	m_{13}	1	1	0	1
14	m_{14}	1	1	1	0

Step 2:

Group	Minterm	A	B	C	D
0	m_0	0	0	0	0 ✓
1	m_2	0	0	1	0 ✓
	m_4	0	1	0	0 ✓
	m_8	1	0	0	0 ✓
2	m_6	0	1	1	0 ✓
	m_{10}	1	0	1	0 ✓
3	m_{13}	1	1	0	1
	m_{14}	1	1	1	0 ✓

Step 3:

Group	Minterm	A	B	C	D	
0	$m_0\ m_2$	0	0	–	0	✓
	$m_0\ m_4$	0	–	0	0	✓
	$m_0\ m_8$	–	0	0	0	✓
1	$m_2\ m_6$	0	–	1	0	✓
	$m_2\ m_{10}$	–	0	1	0	✓
	$m_4\ m_6$	0	1	–	0	✓
	$m_8\ m_{10}$	1	0	–	0	✓
2	$m_6\ m_{14}$	–	1	1	0	✓
	$m_{10}\ m_{14}$	1	–	1	0	✓

Step 4:

Group	~~[illegible]~~ matched pair	A	B	C	D
0	$m_0\ m_2\ m_4\ m_6$	0	–	–	0
	$m_0\ m_2\ m_8\ m_{10}$	–	0	–	0
	$m_0\ m_4\ m_2\ m_6$	0	–	–	0
	$m_0\ m_8\ m_2\ m_{10}$	–	0	–	0
1	$m_2\ m_6\ m_{10}\ m_{14}$	–	–	1	0
	$m_2\ m_{10}\ m_6\ m_{14}$	–	–	1	0

Prime implicants:- $\bar{A}\bar{D}$, $\bar{B}\bar{D}$, $C\bar{D}$

Prime implicant	Minterm	0	2	4	6	8	10	13	14
$\bar{A}\bar{D}$	$m_0 m_2 m_4 m_6$	X	X	X	X				
$\bar{B}\bar{D}$	$m_0 m_2 m_8 m_{10}$	X	X			X	X		
$C\bar{D}$	$m_2 m_6 m_{10} m_{14}$		X		X		X	X	X
						✓		✓	✓

dont care

$ABC\bar{D}$ ← m_{13} (to be written)

$Y = \bar{B}\bar{D} + C\bar{D} + ABC\bar{D}$ Ans.

$Y = xz + wx + wxy$

Verification:-

AB \ CD	00	01	11	10
00	1 0	1 1	3	d 2
01	d 4	5	7	1 6
11	12	1 13	15	1 14
10	1 8	9	11	d 10

$\bar{C}\bar{D} + \bar{B}\bar{D} + ABCD$

Q→ $F(w,x,y,z) = m(1,3,4,5,9,10,11) + \Sigma\phi(6,8)$ → dont care

Decimal	minterm	8 w	4 x	2 y	1 z
1	m1	0	0	0	1
3	m3	0	0	1	1
4	m4	0	1	0	0
5	m5	0	1	0	1
6	m6	0	1	1	0
8	m8	1	0	0	0
9	m9	1	0	0	1
10	m10	1	0	1	0
11	m11	1	0	1	1

Group	minterm	w	x	y	z
1	m1	0	0	0	1 ✓
	m4	0	1	0	0 ✓
	m8	1	0	0	0 ✓
2	m3	0	0	1	1 ✓
	m5	0	1	0	1 ✓
	m6	0	1	1	0 ✓
	m9	1	0	0	1 ✓
	m10	1	0	1	0 ✓
3	m11	1	0	1	1 ✓

Group	matched pairs	w	x	y	z
1	m1 m3	0	0	–	1 ✓
	m1 m5	0	–	0	1 ✓
	m1 m9	–	0	0	1 ✓
	m4 m5	0	1	0	– ✓
	m4 m6	0	1	–	0 ✓
	m8 m9	1	0	0	– ✓
	m8 m10	1	0	–	0 ✓
2	m3 m11	–	0	1	1 ✓
	m9 m11	1	0	–	1 ✓
	m10 m11	1	0	1	– ✓

Group	matched pairs	A	B	C	D
1	m1 m3 m9 m11	–	0	–	1
	m1 m9 m3 m11	–	0	–	1
	m8 m9 m10 m11	1	0	–	–
	m8 m10 m9 m11	1	0	–	–

PI	minterm	1 ✓	3 ✓	4	5	9 ✓	10 ✓	11 ✓
$\bar{x}z$	1, 9, 3, 11	X	X			X		X
$w\bar{x}$	8, 9, 10, 11					X	X	X
$\bar{w}\bar{y}z$	1, 5	X			X			
$\bar{w}x\bar{y}$	4, 5			X	X			
$\bar{w}x\bar{z}$	4, 6			X				
			✓				✓	

$Y = xz + wx + \bar{w}xy$

→ value containing minterm 4 and 5 (they were not used?)

wx \ yz	00	01	11	10
00	0	1	1	2
01	1	1	7	6
11	12	13	15	14
10	8	9	11	10

$= w\bar{x} + \bar{x}z + \bar{w}x\bar{y}$

August 7, 2024

Combinational circuit

→ A circuit where output only depends on input

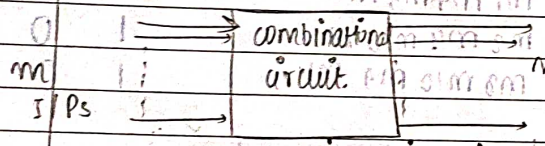
→ Logic gates combined together for a specific output depending upon the inputs without memory being used

$O/P = fn(I/P)$

sequential circuit

→ Output depends not only on present input, but also previous output.

→ It uses memory.



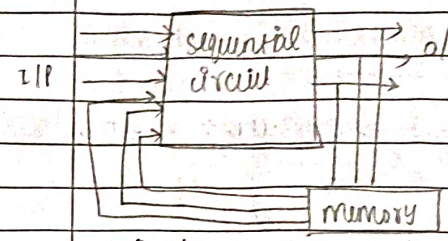
Applications of combinational ckt :-

1) → Adder
→ subtractor
→ multiplier
→ Parity
→ comparator

2) Transmission/Reception (TX/RX)
→ MUX (multiplexer)
→ DEMUX (Demultiplexer)
→ Encoder
→ Decoder

3) code conversion
→ BCD
→ Gray code

$\overline{yxw} + xw + yx = Y$



{Feedback circuit}
{or positive feedback network}

Applications of sequential circuit:-

→ Flip Flop
→ Registers
→ Counter

ADDER

↳ combinational circuit which is used to add numbers (BCD, decimal, binary)

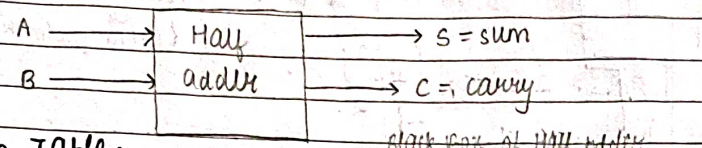
Applications :
- ALU
- Increment
- Decrement
- Address calculation / Indices

Types : 1) Half adder
2) Full adder

1) Half Adder

It is a combinational circuit which takes two bits as input and adds it, does not consider carry for addition.

$Y = xz + wz + wxy$



~~black box of Half adder~~

Truth Table:-

A	B	S	Cout
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

as $1+1 = 2$
if > no., subtract base (binary)
$= 2 - 2 = 0$
no. of times subtraction $= 1$
$\therefore$ carry $= 1$
but sum $= 0$

To find Boolean expression:-

A \ B	0	1
0	0	1
1	2	3

sum = S

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

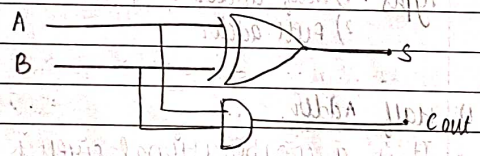
$S = A \oplus B$

A \ B	0	1
0	0	
1	2	1 3

no K-map required for carry?

$C_{out} = AB$

$S = A \oplus B$
$C_{out} = AB$

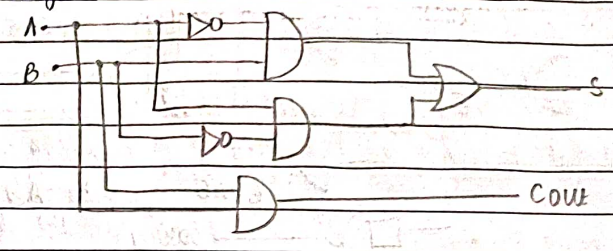


circuit of Half Adder

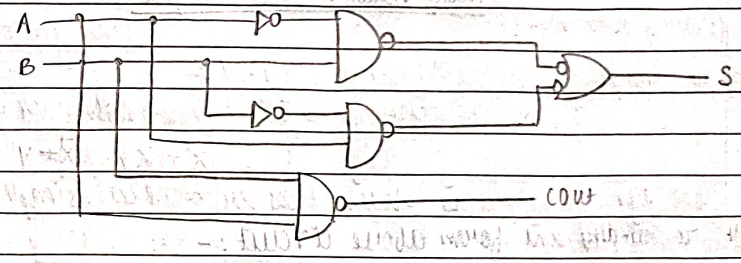
Make this circuit using NAND gate:-

Basic gate:-

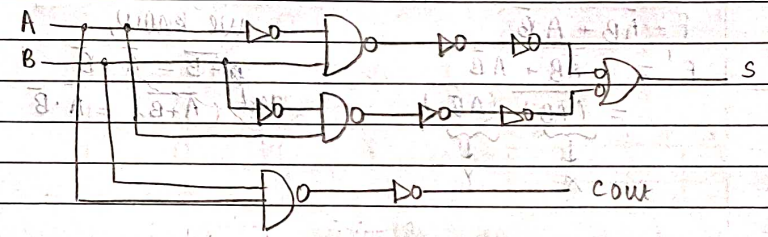
Step 1:



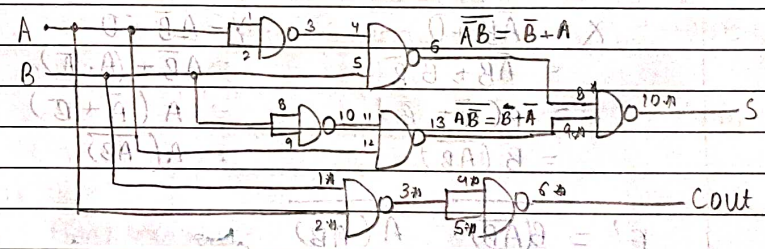
Step 2:



Step 3:

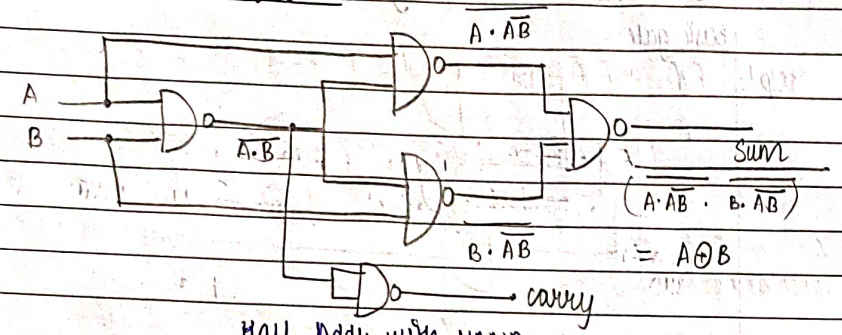


Step 4:



can be further simplified to following circuit?

* Minimum no. of gates :-



Half Adder with NAND gate

{practice with NOR}

⇒ To simplify previously drawn circuit :-

We know,

$\bar{X} + XY = \bar{X} + Y$

{same output is obtained as per Boolean simplification}

August 8, 2024 To simplify and form above circuit :-

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

$F' = \overline{\bar{A}B + A\bar{B}}$

$= \underbrace{(\overline{\bar{A}B})}_{X} \cdot \underbrace{(\overline{A\bar{B}})}_{Y}$

We know,

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

and $(\overline{A+B}) = \bar{A} \cdot \bar{B}$

$X = \bar{A}B + 0$
$= \bar{A}B + B \cdot \bar{B}$
$= B(\bar{A} + \bar{B})$
$= B(\overline{AB})$

$Y = A\bar{B} + 0$
$= A\bar{B} + (A \cdot \bar{A})$
$= A(\bar{A} + \bar{B})$
$= A(\overline{AB})$

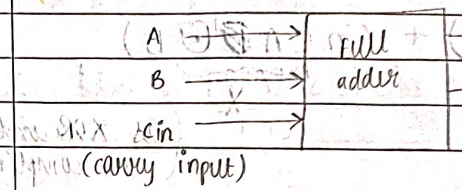
$F' = \overline{B(\overline{AB})} \cdot \overline{A(\overline{AB})}$

$= [(B(AB)')' \cdot (A \cdot (AB)')']'$ → one NAND gate

{whenever two product terms have whole bar, use NAND gate}

Full Adder

A combinational circuit which adds two bits with previous carry.



Block Box of Full adder

Truth Table

A	B	Cin	S	Cout
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

For sum :-

A \ BCin	00	01	11	10
0	0	1	3	2
1	4	5	7	6

Chess Board type of configuration

→ $A \oplus B \oplus C_{in}$

For carry :-

A \ BCin	00	01	11	10
0	0	1	3	2
1	4	5	7	6

→ $BCin + ACin + AB$

For sum :-

Final expression : $A \oplus B \oplus Cin$

How to obtain :-

$S = \Sigma(1, 2, 4, 7)$

* odd no. of 1s → XOR gate [in Truth table] show 1.

[XOR is an odd 1s detector]

$$= \overset{(1)}{\bar{A}\bar{B}C_{in}} + \overset{(2)}{\bar{A}B\bar{C}_{in}} + \overset{(4)}{A\bar{B}C_{in}} + \overset{(7)}{ABC_{in}}$$

$$= C_{in}(\bar{A}B + A\bar{B}) + C_{in}[\bar{A}\bar{B} + AB]$$

$$= \bar{C}_{in}\underbrace{(A \oplus B)}_{X} + C_{in}\underbrace{(A \odot B)}_{\bar{X}}$$

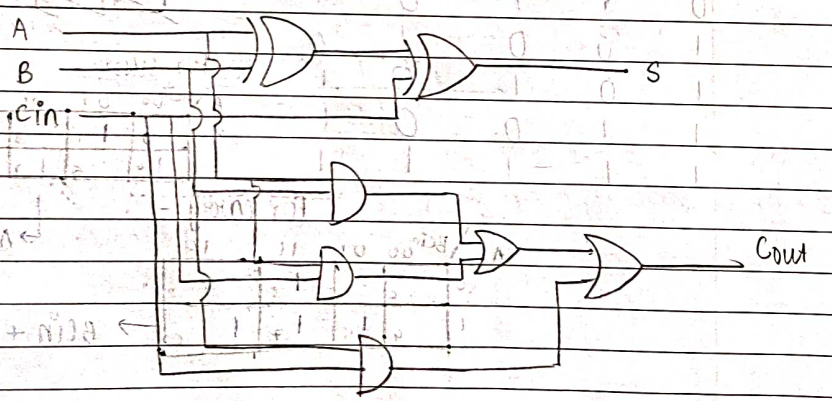
{as XOR and XNOR are compliments}

$$= \bar{C}_{in}X + C_{in}\bar{X}$$

$$= C_{in} \oplus X$$

$$= \boxed{C_{in} \oplus A \oplus B}$$ → Chess Board configuration output.

Circuit:-



Another way of writing carry:-

A \ BCin				
	0	1	1 (2)	2
	4	1 (5)	1 (3)	1 (6)

$$AB + \bar{A}BC_{in} + A\bar{B}C_{in}$$

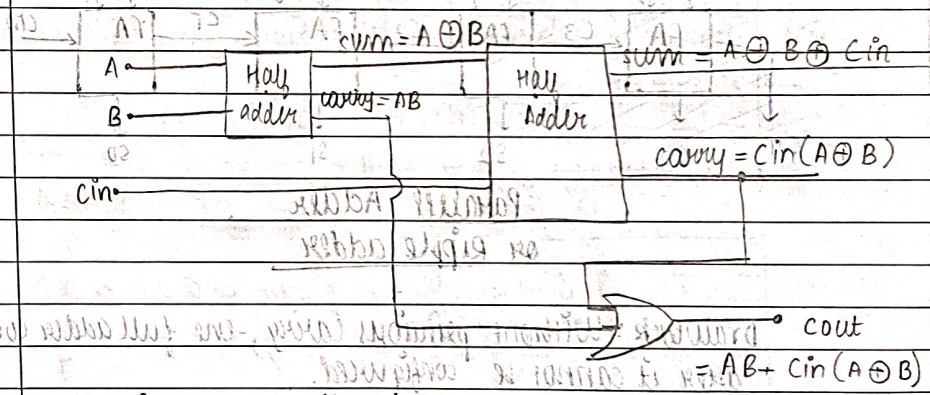
$$= AB + C_{in}(\bar{A}B + A\bar{B})$$

$$\text{carry} = \boxed{AB + C_{in}(A \oplus B)}$$

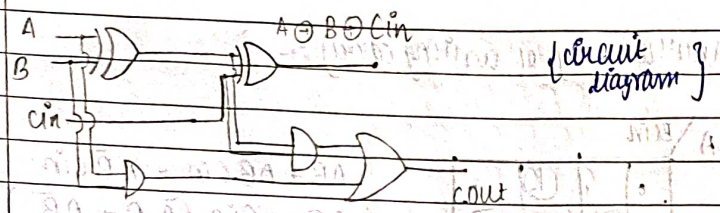
Full Adder using Half Adder

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

$$C = AB + BC_{in} + AC_{in} = AB + C_{in}(A \oplus B)$$



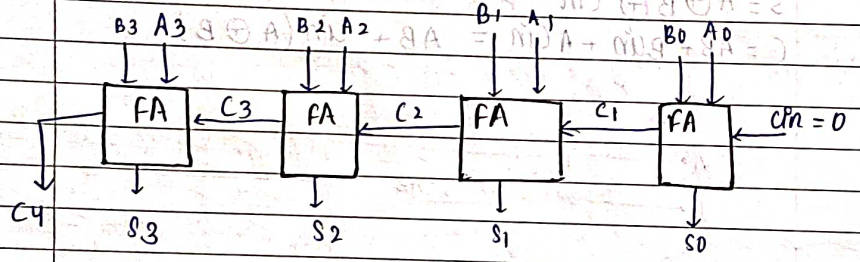
Black Box of Full adder using Half adder



{circuit diagram}

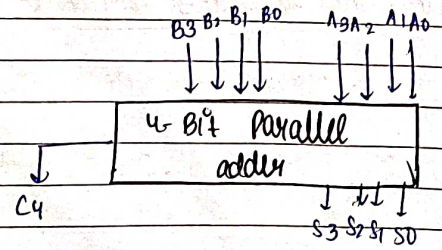
For addition of four bits at a time: we need four full adders.

$$\begin{array}{r} C_3\ C_2\ C_1\ \ \\ A_3\ A_2\ A_1\ A_0 \\ +\ B_3\ B_2\ B_1\ B_0 \\ \hline C_4\ S_3\ S_2\ S_1\ S_0 \end{array}$$

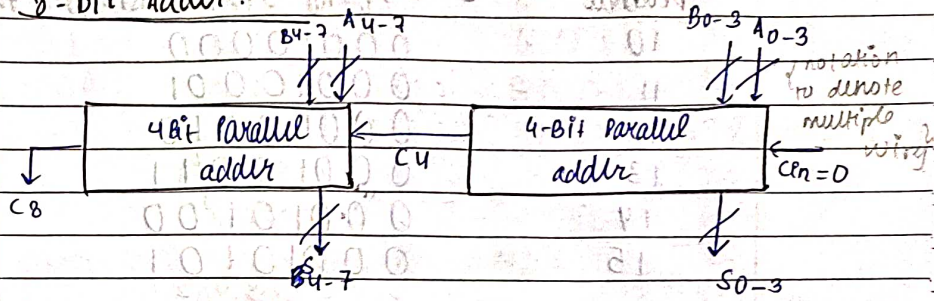


Parallel Adder
or Ripple adder

Drawback: Without previous carry, the full adder connected after it cannot be configured.



8-bit Adder :-



(8 bit parallel adder)

→ also called 8-4-2-1

BCD {Binary Coded Decimal}

Binary representation of decimal number, $n = 10$ (range → 0 to 9)

In BCD, every digit will be expressed by 4 binary bits.

Eg: $(9)_{10} \longrightarrow (1001)_{BCD}$

Decimal	Z_8	Z_4	Z_2	Z_1
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

as every digit needs to be represented by 4 bits. __/__/__

Decimal	{no 8421 exists for these}
10	00010000
11	00010001
12	00010010
13	00010011
14	00010100
15	00010101
	8421 8421

{each digit follows 8421 individually}

Decimal to BCD

→ $(56)_{10} = (01010110)_{BCD}$

→ $(153)_{10} = (000101010011)_{BCD}$

BCD to Decimal

→ $(01100111)_{BCD} = (67)_{10}$

→ (001111 1111)

↳ This is not proper BCD as 15 is (1111). (Invalid BCD)

Addition of BCD :-

1) sum is ≤ 9, answer is valid.

steps for addition : i) convert Decimal to BCD
ii) Add the no.s (Binary addition)

__/__/__

Eg: $(3)_{10} + (5)_{10}$

↓ ↓

0011 0101

```
  1 1 1
  0 0 1 1
+ 0 1 0 1
  1 0 0 0
```

= 8 which is ≤ 9

∴ valid BCD

2) sum > 9, answer is invalid.
To make it valid, add binary of 6 (0110) to the sum.
{no.s from 10 to 15 are 6}

Eg: $(4)_{10} + (6)_{10}$

↓ ↓

(0100) (0110)

```
  0 1 0 0
+ 0 1 1 0
  1 0 1 0
```

$= 10 > 9$

to make it valid

```
  1 0 1 0
+ 0 1 1 0
0 0 0 1 0 0 0 0
  1     0
```

$= 10$

3) sum is < 9 but Final carry $= 1$

Eg: $(9)_{10} + (8)_{10}$

↓ ↓

1001 1000

```
  1 0 0 1
+ 1 0 0 0
1 0 0 0 1
```

add 6 because although sum < 9 but carry = 1

```
  1 0 0 0 1
    0 1 1 0
  0 0 0 1 0 1 1 1   = 17
```

Q→ $(56)_{10} + (55)_{10}$

```
  1 1 1
  0 1 0 1   0 1 1 0
+ 0 1 0 1   0 1 0 1
  1 0 1 0   1 0 1 1
+ 0 1 1 0   0 1 1 0
  1 0 0 0 1 0 0 0 1   = 111
```

add 6 to both as >9

Q→ $(68)_{10} + (73)_{10}$

```
  1 1
  0 1 1 0   1 0 0 0
+ 0 1 1 1   0 0 1 1
  1 1 0 1 1 1 0 1 1
  0 1 1 0   0 1 1 0
  0 0 0 1 0 1 0 0   0 0 0 1   = 141
```

add 6:

4 bits → nibble

Q→ $(99)_{10} + (99)_{10}$

```
  1 1 → auxiliary carry
  1 0 0 1   1 0 0 1
+ 1 0 0 1   1 0 0 1
  1 0 0 1 1 0 0 1 0
  0 1 1 0   0 1 1 0
  1 1 0 0 1 1 0 0   = 198 Ans.
```

final carry

add 6 to both

Q→ $(54)_{10} + (64)_{10}$

```
  1
  0 1 0 1   0 1 0 0
+ 0 1 1 0   0 1 0 0
  1 0 1 1   1 0 0 0
  0 1 1 0
  1 0 0 0 1 1 8     = 118
```

add 6 only to first

BCD adder Maximum sum that can be obtained 9+9+1 = 19

K	z_8	z_4 (Binary)	z_2	z_1	C	S_8	S_4 (BCD)	S_2	S_1	Decimal
0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	1	0	0	0	0	1	1
					0	0	0	1	0	2

Max. value that can be obtained = 9+9+1 = 19

BCD adder

Binary K	$Z_8 Z_4 Z_2 Z_1$	BCD C	$S_8 S_4 S_2 S_1$	Decimal
0	0000	0	0000	0
0	0001	0	0001	1
0	0010	0	0010	2
0	0011	0	0011	3
0	0100	0	0100	4
0	0101	0	0101	5
0	0110	0	0110	6
0	0111	0	0111	7
0	1000	0	1000	8
0	1001	0	1001	9
0	1010	1	0000	10
0	1011	1	0001	11
0	1100	1	0010	12
0	1101	1	0011	13
0	1110	1	0100	14
0	1111	1	0101	15
1	0000	1	0110	16
1	0001	1	0111	17
1	0010	1	1000	18
1	0011	1	1001	19

need not be minimized.

For no.s 16 to 19, $C = K = 1$

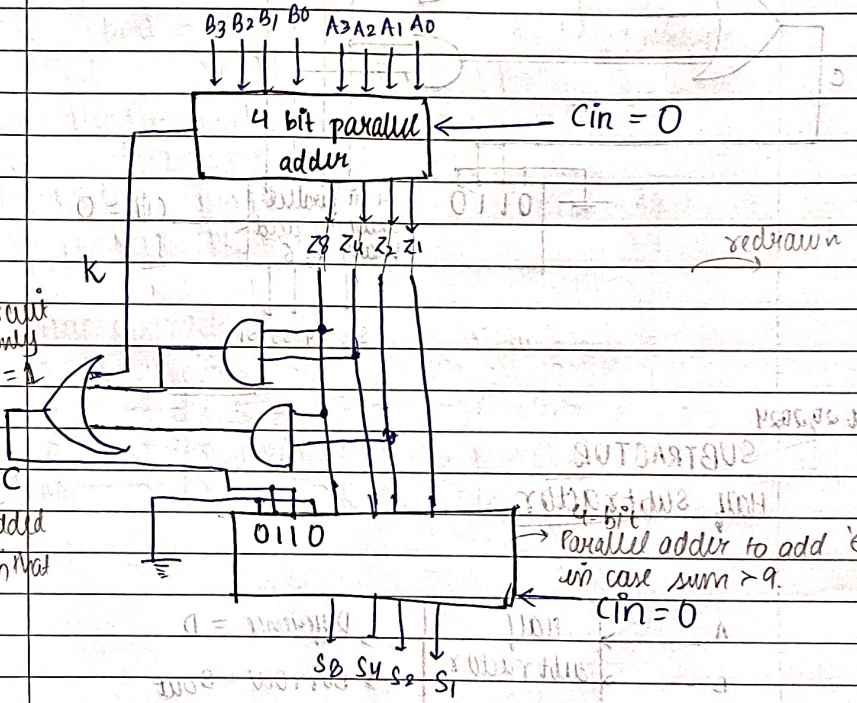
similarly, for 0 to 9, $C = K = 0$

* The only change is observed from no.s 10 to 15 where $C = 1$ and $K = 0$.

Make a K-Map to minimize the expression where change is observed :-

$Z_8 Z_4$ \ $Z_2 Z_1$	00	01	11	10
00	0	1	3	2
01	4	5	7	6
11	1 (12)	1 (13)	1 (15)	1 (14)
10	1 (8)	1 (9)	1 (11)	1 (10)

$C = K + Z_8 Z_4 + Z_8 Z_2$

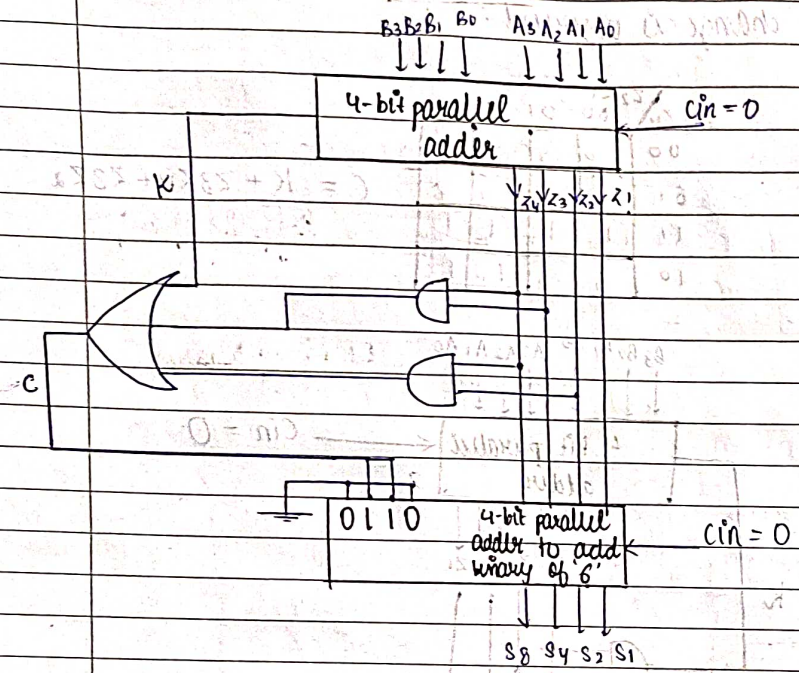


redrawn

* This circuit works only when C = 1

as 6 will be added only in that case

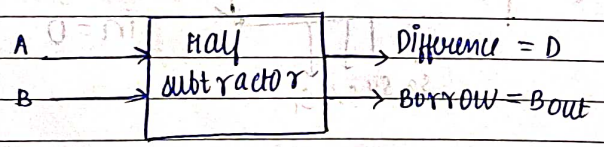
Redrawn :-



August 20, 2024

SUBTRACTOR

Half subtractor



* make HA and FA using NOR gates only. [NAND also] and HS and FS also

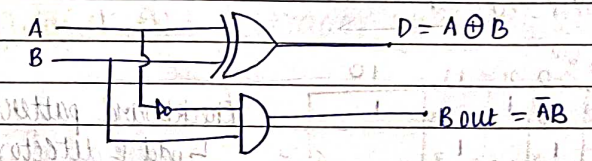
A	B	D	Bout
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

$D = \bar{A}B + A\bar{B}$ $\quad \Sigma(1,2) \rightarrow$ minterms

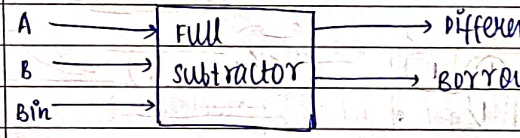
$= A \oplus B$

$B_{out} = \bar{A}B$ $\quad \Sigma(1) \rightarrow$ minterm

circuit Diagram :



Full subtractor



A	B	Bin	D	Bout
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

To simplify expressions:-

FOR difference,

A \ B Bin	00	01	11	10
0	0	1 1	3	1 2
1	1 4	5	1 7	6

{checkboard pattern}
{odd 1 detector}

$D = A \oplus B \oplus Bin$

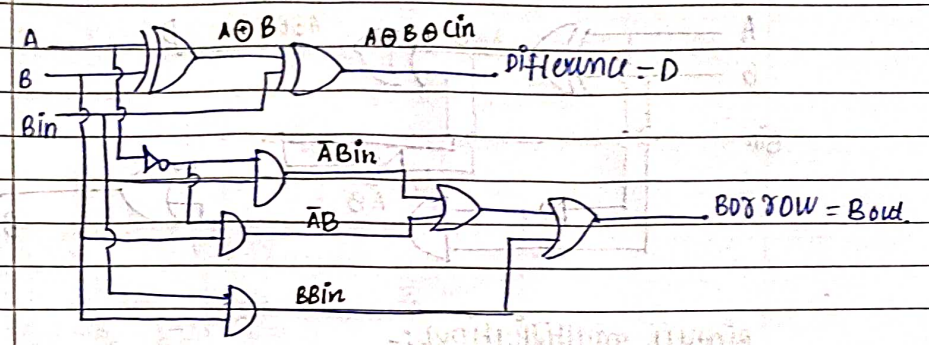
FOR BORROW,

A \ B Bin	00	01	11	10
0	0	1 1	1 3	1 2
1	4	5	1 7	6

$\Sigma(1,2,3,7)$

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

Circuit:-



To make a Full subtractor using Half subtractors

Another expression for BORROW:

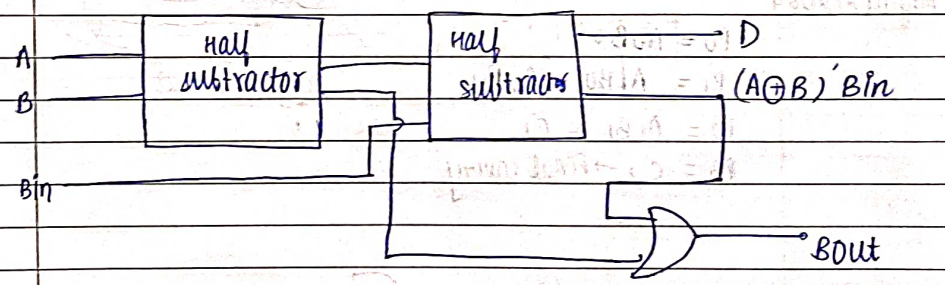
A \ B Bin	00	01	11	10
0	0	1	1 3	1 2
1	4	5	1 7	6

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

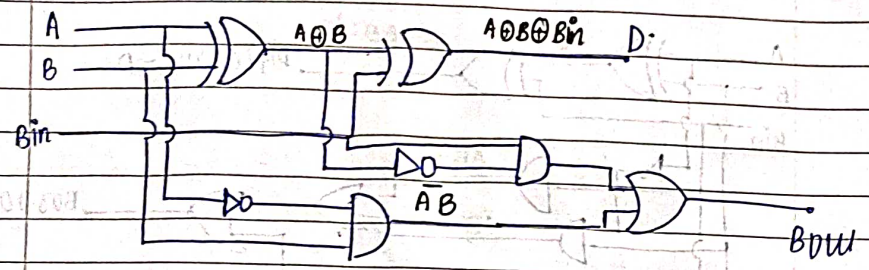
$= \bar{A}B + Bin(\bar{A}B + AB)$

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

{XNOR is complement of XOR}



Circuit.

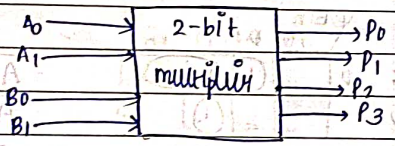


Binary multiplication:-

Eg:

$$\begin{array}{r} A_1 \quad A_0 \\ \times\ B_1 \quad B_0 \\ \hline (A_1B_0)\ 0 \quad 0\ (A_0B_0) \\ (A_1B_1)\ 1 \quad (A_0B_1) \\ \hline 1 \quad 1 \quad 0 \\ \hline \text{(carry) } P_3 \quad P_2 \quad P_1 \quad P_0 \end{array}$$

* For a 2-bit multiplier, max. output = 4



Black Box

August 21, 2024

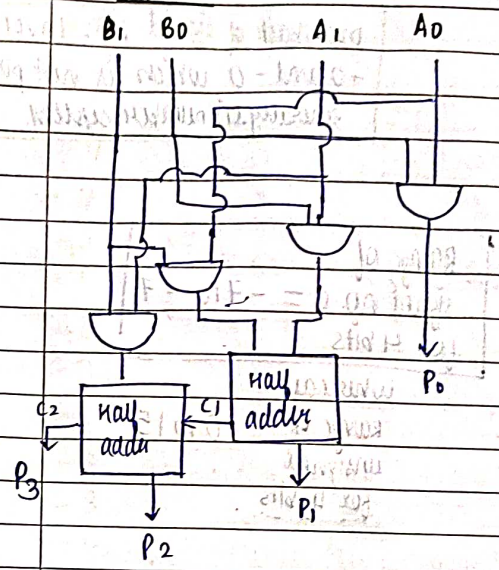
$P_0 = A_0B_0$

$P_1 = A_1B_0 + A_0B_1$

$P_2 = A_1B_1 + C_1$

$P_3 = C_2 \rightarrow$ Final carry.

Circuit:-



Signed Numbers

The last bit denotes sign, positive → zero (0)
negative → one (1)

sign bit

Eg: +5 = 0101

−5 = 1101

signed bit

+7	0111
+6	0110
+5	0101
+4	0100
+3	0011
+2	0010
+1	0001
+0	0000
-0	1000
-1	1001
-2	1010
-3	1011
-4	1100
-5	1101
-6	1110
-7	1111

Drawback of signed bit: Involves +0 and -0 which is not possible generally in number system.

Range of signed no.s = -7 to +7 for 4 bits

whereas range of unsigned for 4-bits = 0 to 15

1's complement :-

+5 = 0101

-5 = 1010 {complemented terms}

2's complement :- {preferred one}

→ 1's complement + 1

*Final carry is discarded in 2's complement

Eg: -5 = 1010 (1's complement)

+ 0001

1011

*Overflow occurs in signed numbers (rarely in unsigned)

	1's complement	2's complement
+7		
+6		
+5	same as signed	same as signed
+4		
+3		
+2		
+1		
+0		
-0	1111	0000
-1	1110	1111
-2	1101	1110
-3	1100	1101
-4	1011	1100
-5	1010	1011
-6	1001	1010
-7	1000	1001
	For this case: -8	1000

as one binary? {zero not zero}

Range of 2's complement :- +7 to -8

Eg: (+7)+(-5) → 0111

+ 1001 {2's complement}

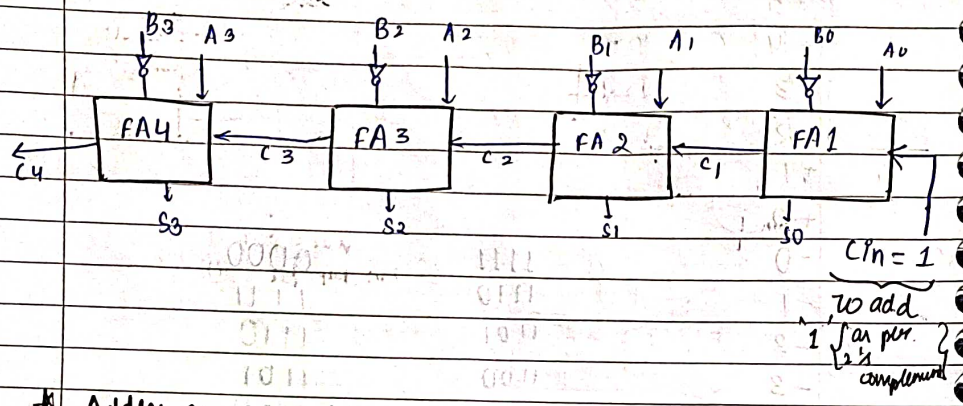
1 0010 } 2

discarded

~~Adder as a subtractor~~

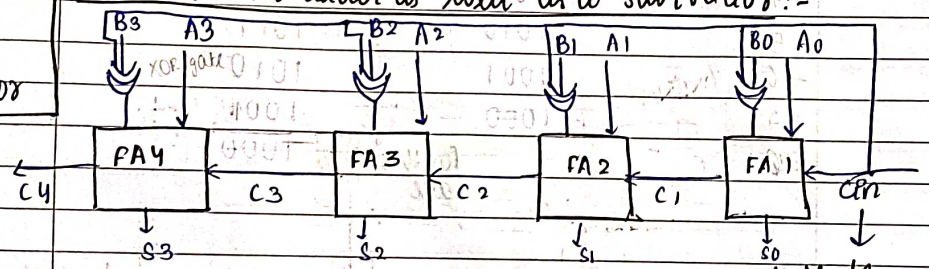
4-bit Parallel Adder (as well as subtractor)

* Adder as a subtractor :-



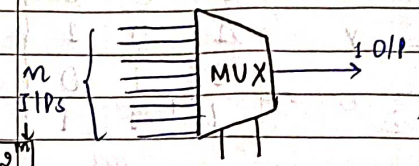
* Adder as an adder as well as a subtractor :-

Parallel Adder/ subtractor



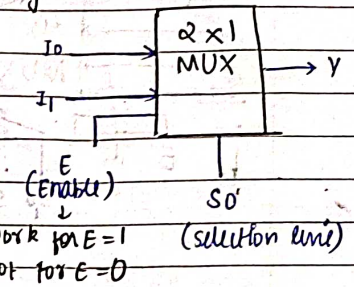
Imp. ** Adder: Cin = 0
Subtractor: Cin = 1

V.Imp. **MULTIPLEXER (MUX)**
↳ Data selector



$n = 2^m$
no. of I/P = 2^m
no. of selection line

Eg:

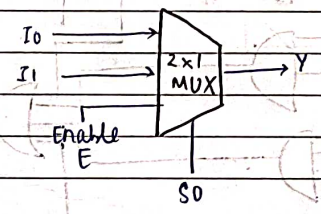


will work for E = 1 and not for E = 0

↳ Also converts parallel to serial.

* Multiple inputs → one output as per the selection line.

2 x 1 MUX

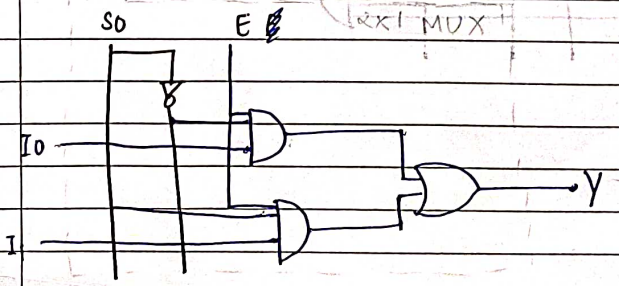


Truth Table :-

E	S_0	Y
0	X	0
1	0	I_0
1	1	I_1

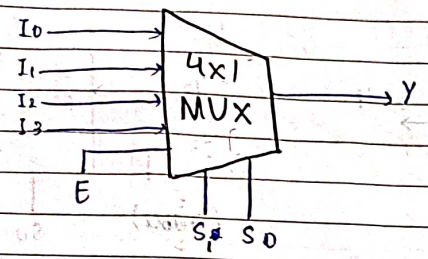
$Y = E\overline{S_0} I_0 + E S_0 I_1$

Circuit :-



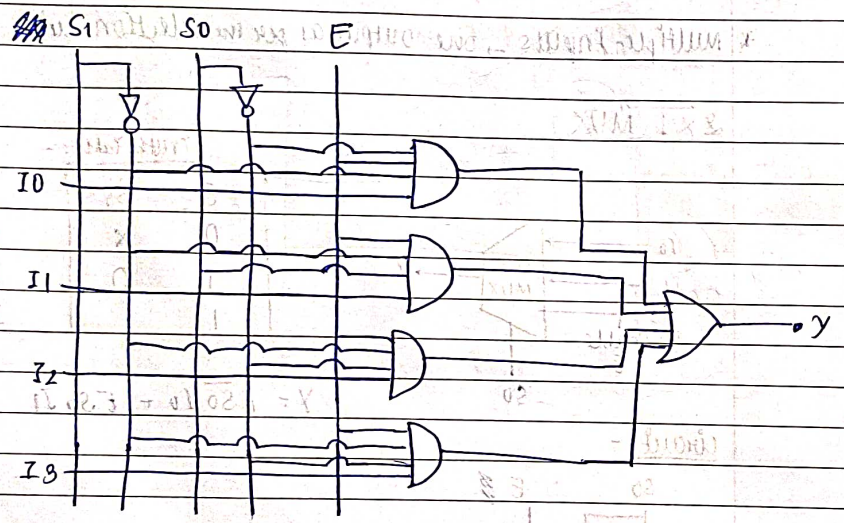
Also design by using NAND gate :-

4x1 MUX

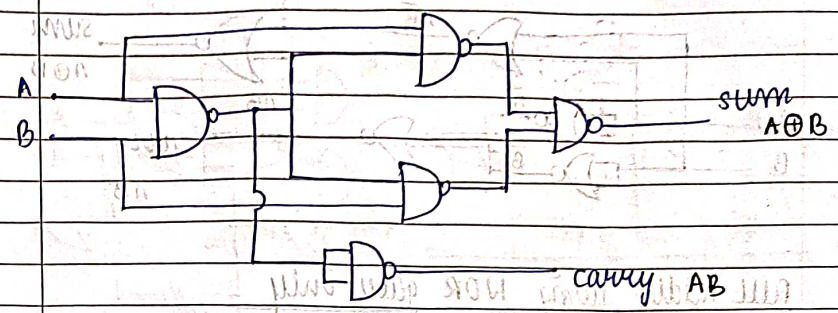


E	S_1	S_0	Y
0	x	x	0
1	0	0	I_0
1	0	1	I_1
1	1	0	I_2
1	1	1	I_3

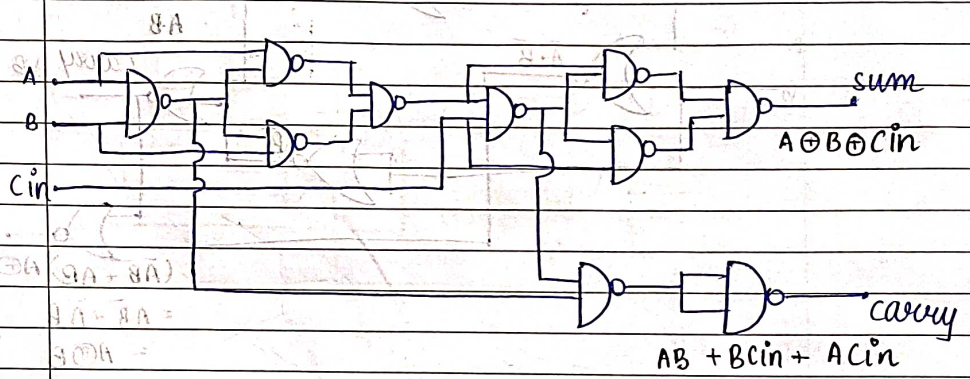
$$Y = E\bar{S_0}\bar{S_1} I_0 + E\bar{S_1}S_0 I_1 + E S_1 \bar{S_0} I_2 + E S_1 S_0 I_3$$



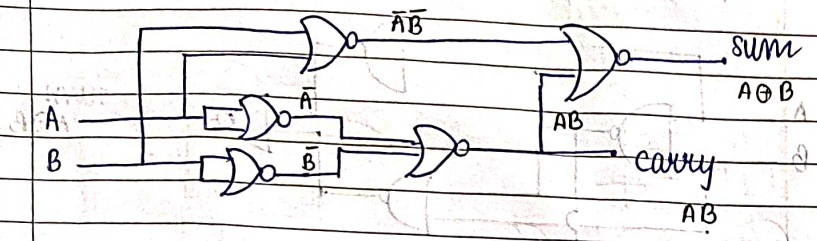
Half Adder using NAND gates only



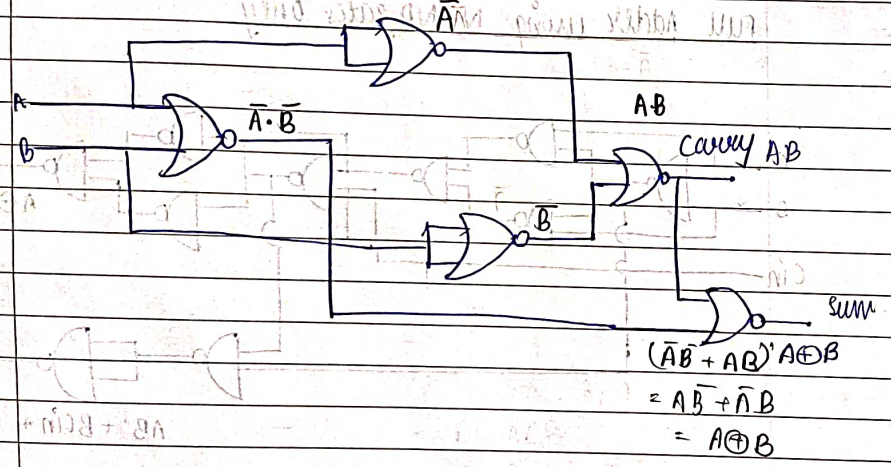
Full Adder using NAND gates only



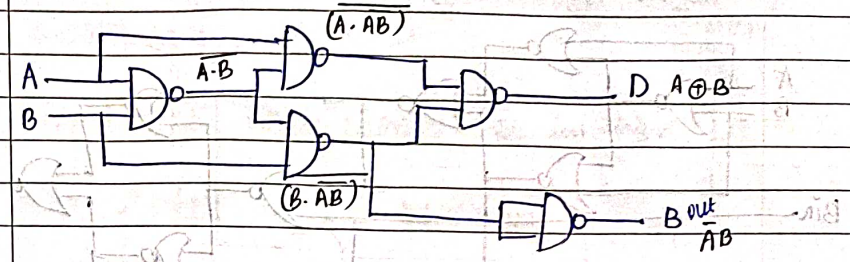
Half Adder using NOR gates only



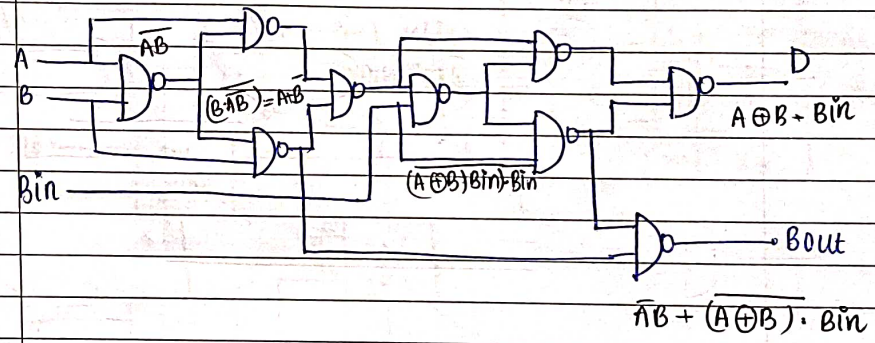
Full Adder using NOR gate only



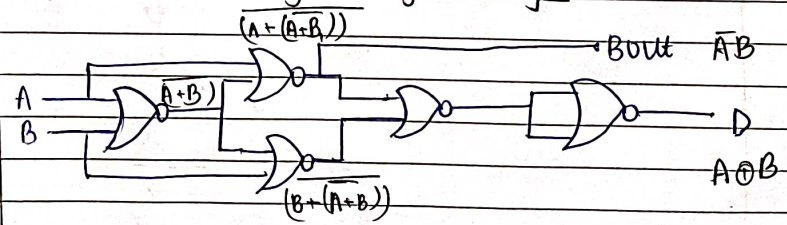
Half subtractor using NAND gates only



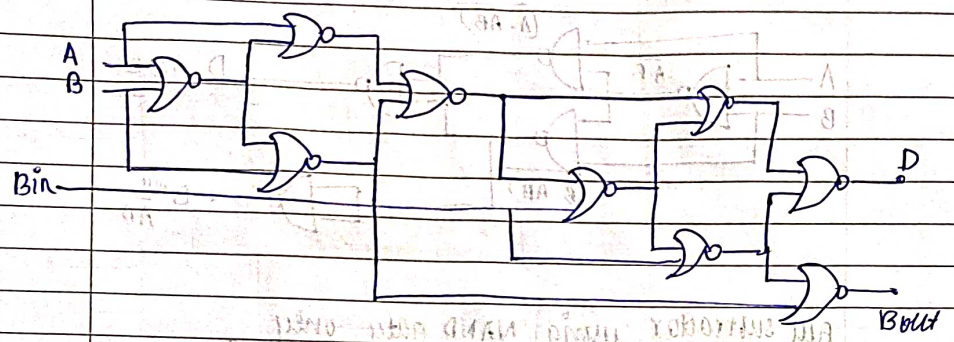
Full subtractor using NAND gate only



Half subtractor using NOR gates only



Full subtractor using NOR gates only



August 22, 2024

Q- Design a 4x1 MUX using a 2x1 MUX. {concept of MUX Tree}

Required → 4, Available → 2

Step 1: $\frac{\text{Required MUX}}{\text{Available MUX}} = \frac{4}{2} = \boxed{2}$ 1st MUX (→ we need 2 MUX)

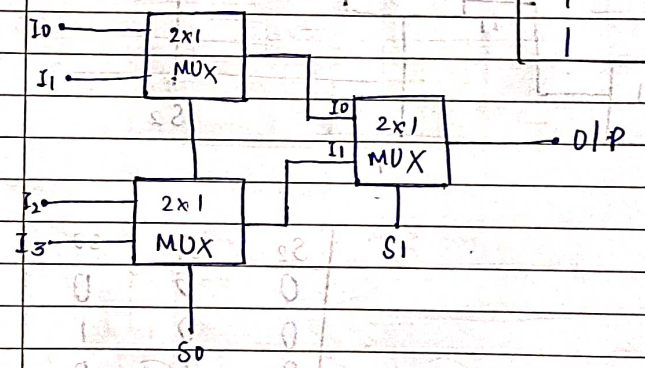
Divide until values becomes 1 or less than 1.

$= \frac{2}{2} = \boxed{1}$ 2nd MUX (only 1)

Total MUX = No. of 1st MUX + No. of 2nd MUX
= 2 + 1
= 3

4x1 MUX Truth Table:

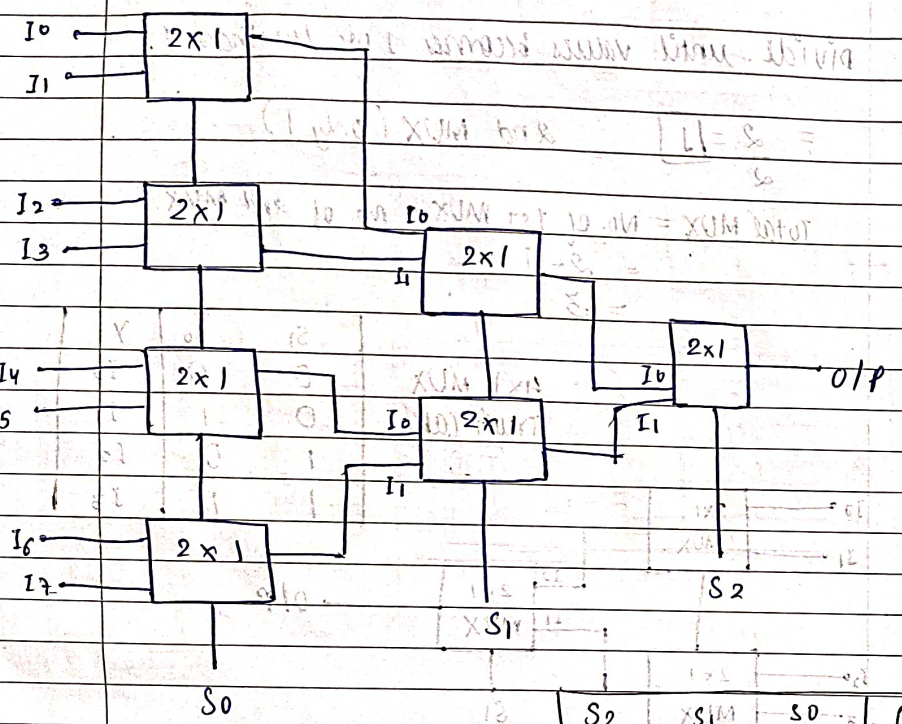
S_1	S_0	Y
0	0	I_0
0	1	I_1
1	0	I_2
1	1	I_3



Q→ Design 8x1 MUX by 2x1

layer/Level 1 layer/Level 2 layer/Level 3

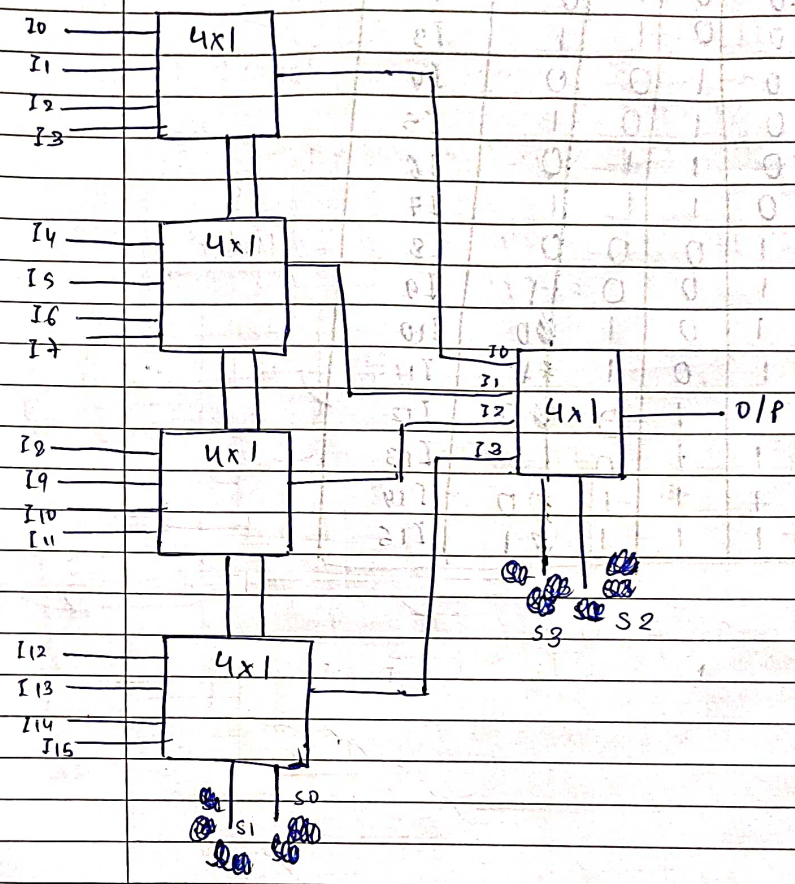
Soln → $\frac{8}{2} = \boxed{4}$, $\frac{4}{2} = \boxed{2}$, $\frac{2}{2} = \boxed{1}$



S_2	S_1	S_0	O/P
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

Q→ 16x1 MUX by 4x1 MUX

Soln → $\frac{16}{4} = \boxed{4}$, $\frac{4}{4} = \boxed{1}$



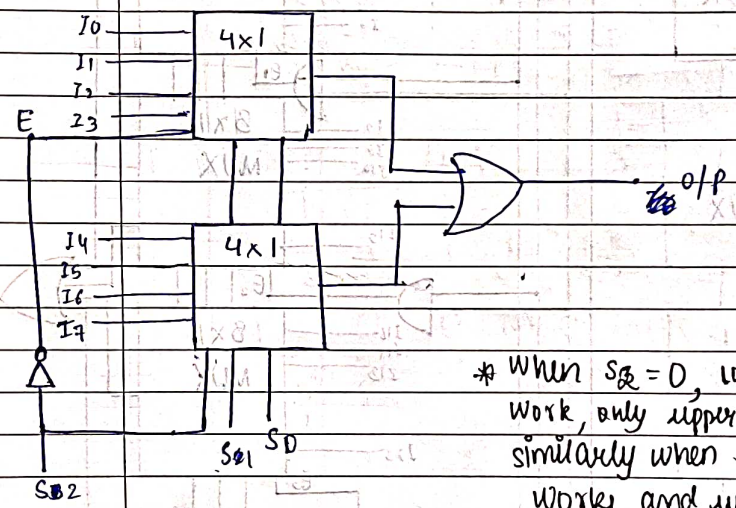
→ Table

S_3	S_2	S_1	S_0	O/P
0	0	0	0	I_0
0	0	0	1	I_1
0	0	1	0	I_2
0	0	1	1	I_3
0	1	0	0	I_4
0	1	0	1	I_5
0	1	1	0	I_6
0	1	1	1	I_7
1	0	0	0	I_8
1	0	0	1	I_9
1	0	1	0	I_{10}
1	0	1	1	I_{11}
1	1	0	0	I_{12}
1	1	0	1	I_{13}
1	1	1	0	I_{14}
1	1	1	1	I_{15}

8×1 MUX using 4×1

$\frac{8}{4} = \boxed{2}$, $\frac{2}{4} = \boxed{0.5}$

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



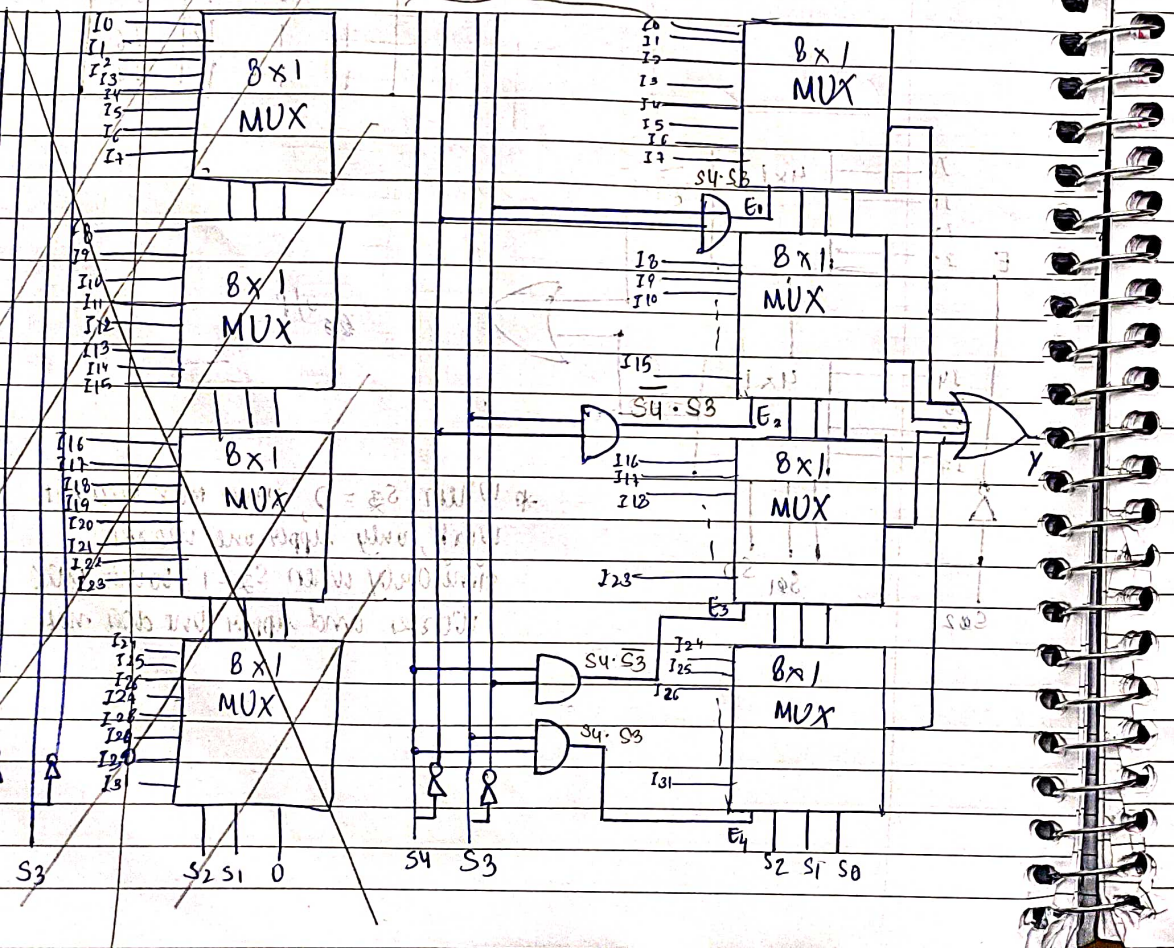
* When $S_2 = 0$, lower MUX will not work, only upper one works. similarly when $S_2 = 1$, lower MUX works and upper one does not.

0 → 32x1 by 8x1
0 → 16x1 by 8x1
0 → 32x1 by 16x1

32x1 MUX using 8x1 MUX

$\frac{32}{8} = \boxed{4}$ $\frac{4}{8} = \boxed{0.5}$

can be replaced with a 4x1 MUX



~~16x1 MUX using 8x1 MUX~~

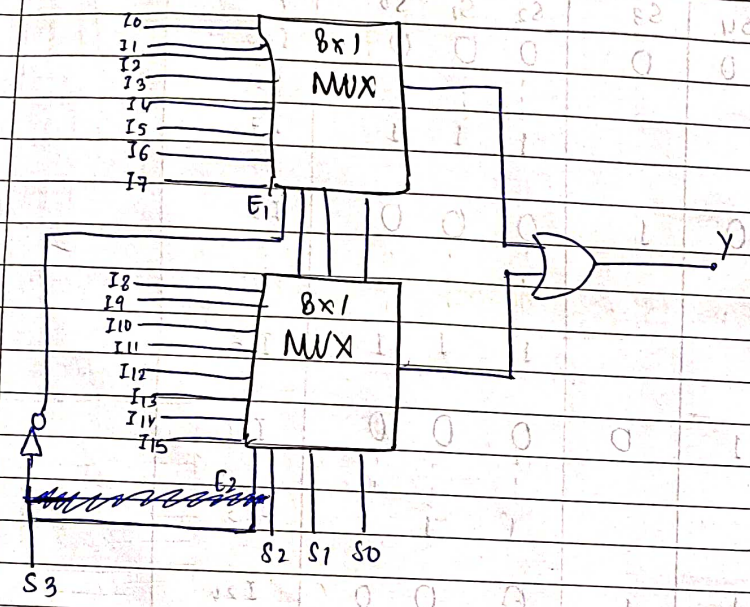
~~$\frac{16}{8} = \boxed{2}$ $\frac{2}{8} = \boxed{0.25}$~~

Truth table :-

S4	S3	S2	S1	S0	Y
0	0	0	0	0	I0
		⋮	⋮	⋮	⋮
		1	1	1	I7
0	1	0	0	0	I8
		⋮	⋮	⋮	⋮
		1	1	1	I15
1	0	0	0	0	I16
		⋮	⋮	⋮	⋮
		1	1	1	I23
1	1	0	0	0	I24
		⋮	⋮	⋮	⋮
		1	1	1	I31

16x1 MUX using 8x1 MUX

$\frac{16}{8} = \boxed{2}$ $\qquad \frac{2}{8} = \boxed{0.25}$



August 27, 2024

Q. $F(A,B,C) = \Sigma m(1\ 3\ 5\ 6)$ by MUX.

~~To choose~~ To choose right MUX, $\boxed{n+1 = 3}$ → no. of variables

{can use other MUX but choosing a MUX appropriately this method is best.}

n → no. of selection lines

no. of variables = 3

$n = 2$

MUX $= 2^2 = 4$

∴ 4x1 MUX

S_1	S_0	O/P
0	0	I_0
0	1	I_1
1	0	I_2
1	1	I_3

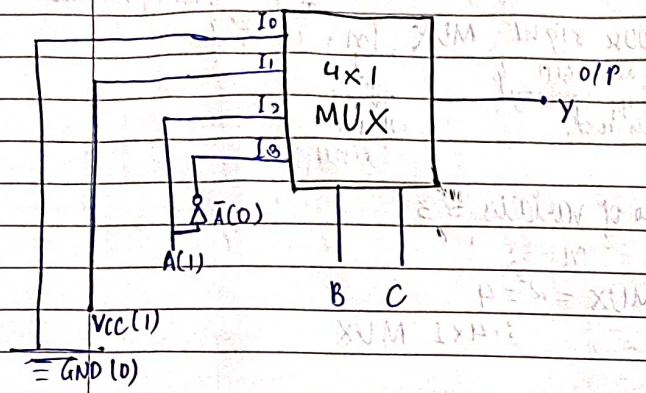
Input selection lines.

* A B C

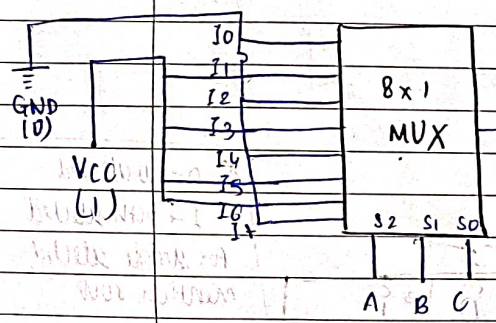
* 0 → no selected
1 → both selected
For single selected, mention row

Implementation Table

		$\bar{B}\bar{C}$	$\bar{B}C$	$B\bar{C}$	BC
	A	I_0	I_1	I_2	I_3
$\bar{A}$	0	0	1	2	3
A	1	4	5	6	7
		0	1	$\bar{A}$	A
		I_0	I_1	I_2	I_3



By using 8x1 MUX :- {more efficient for the given question}



By using 2x1 MUX :-

S_0	Y
0	I_0
1	I_1

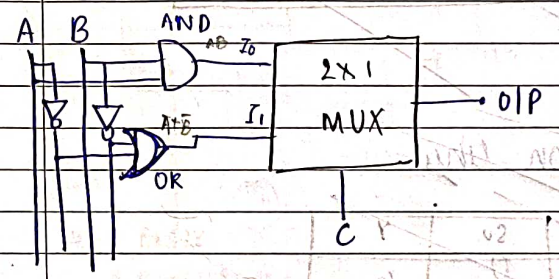
A	B	0 (I0 lc)	1 (I_1)
0	0	0	1
0	1	2	3
1	0	4	5
1	1	6	7
		AB	$\bar{A}+\bar{B}$

simplification of I_1 : $I_1 = A\bar{B} + \bar{A}B + A\bar{B}$

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

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

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



Practice Q → $F(A, B, C) = \Sigma m(1, 2, 6, 7)$ using :-

- 8x1 MUX
- 4x1 MUX
- 2x1 MUX

done later

Q- $F(A, B, C, D) = \Sigma(0, 1, 3, 4, 7, 8, 9, 11, 14, 15)$ by using 8x1, 4x1 and 2x1 MUX.

D	C	B	
S_0	S_1	S_2	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

8x1 MUX

A → input

BCD → selection lines

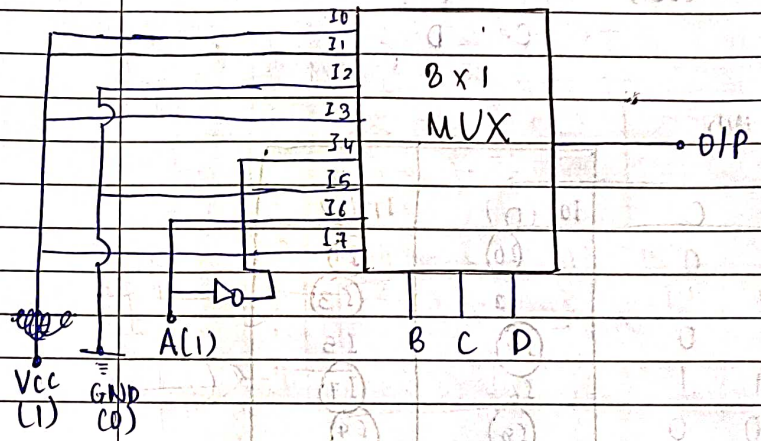
S_2	S_1	S_0	Y
0	0	0	I_0
0	0	1	I_1
0	1	0	I_2
0	1	1	I_3
1	0	0	I_4
1	0	1	I_5
1	1	0	I_6
1	1	1	I_7

For 8x1 MUX

A → I/P

B, C, D → selection lines

	I_0	I_1	I_2	I_3	I_4	I_5	I_6	I_7
A	000 (B C D)	001	010	011	100	101	110	111
0	0	1	2	3	4	5	6	7
1	8	9	10	11	12	13	14	15
	1	1	0	1	$\bar{A}$	0	A	1



4x1 MUX

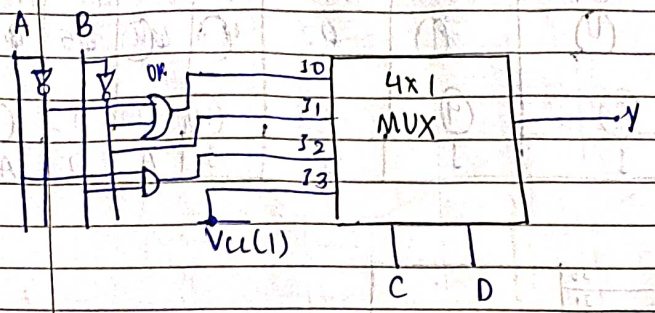
A, B → input

C, D → selection lines.

A	B	I_0 (CD 00)	I_1 (CD 01)	I_2 (CD 10)	I_3 (CD 11)
0	0	0	1	2	3
0	1	4	5	6	7
1	0	8	9	10	11
1	1	12	13	14	15
		$A+B$	$\bar{B}$	AB	1

$I_0 = \bar{A}\bar{B} + \bar{A}B + A\bar{B} = \bar{A} + A\bar{B} = \bar{A} + \bar{B}$

$I_1 = \bar{A}\bar{B} + A\bar{B} = \bar{B}(A + \bar{A}) = \bar{B}$



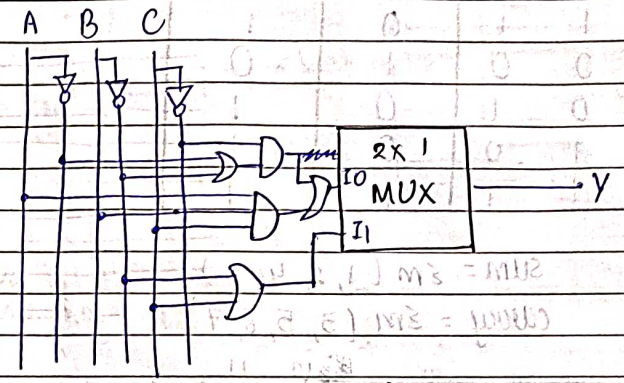
Using 2x1 MUX

A	B	C	$\bar{D}$ I0 (0)	D I1 (1)
0	0	0	I0	I1
0	0	1	I2	I3
0	1	0	I4	I5
0	1	1	I6	I7
1	0	0	I8	I9
1	0	1	I10	I11
1	1	0	I12	I13
1	1	1	I14	I15
			$\bar{C}(\bar{A}+\bar{B})+ABC$	$\bar{B}+C$

$$I0 = \bar{A}\bar{B}\bar{C} + \bar{A}B\bar{C} + A\bar{B}\bar{C} + ABC$$
$$= \bar{A}\bar{C}(\bar{B}+B) + A(\bar{B}\bar{C} + BC)$$
$$= \bar{A}\bar{C} + A\bar{B}\bar{C} + ABC$$
$$= \bar{C}(\bar{A} + A\bar{B}) + ABC$$
$$\boxed{I0 = \bar{C}(\bar{A}+\bar{B}) + ABC}$$

$$I1 = \bar{A}\bar{B}\bar{C} + \bar{A}\bar{B}C + \bar{A}BC + A\bar{B}C + A\bar{B}C + ABC$$

$$= AB + BC + A\bar{B}$$
$$= \bar{B}(\bar{A}+A) + BC$$
$$= \bar{B} + BC$$
$$\boxed{I1 = \bar{B} + C}$$



circuit

[also make full subtractor and use 2×1 MUX also]

August 28, 2024

Q. Design a full adder using 4×1 MUX.

Truth Table:

A	B	Cin	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

sum = Σm (1, 2, 4, 7)

carry = Σm (3, 5, 6, 7)

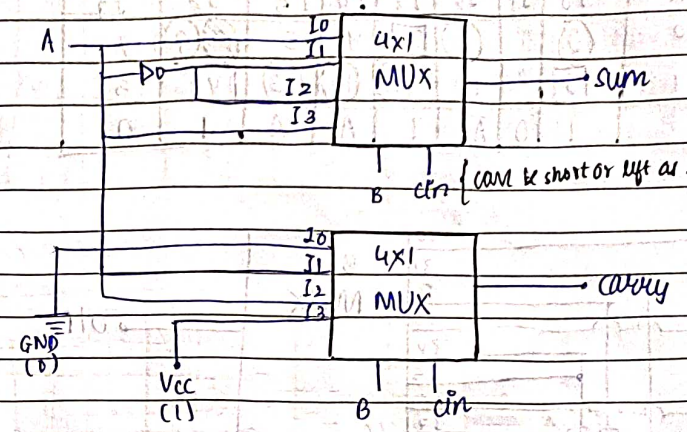
Implementation Table:

A	I_0 (00)	I_1 (01)	I_2 (10)	I_3 (11)
0	0	1	2	3
1	4	5	6	7
	A	$\bar{A}$	$\bar{A}$	A

(for sum)

A	I_0 (00)	I_1 (01)	I_2 (10)	I_3 (11)
0	0	1	2	3
1	4	5	6	7
	0	A	A	1

(for carry)



→ make full subtractor

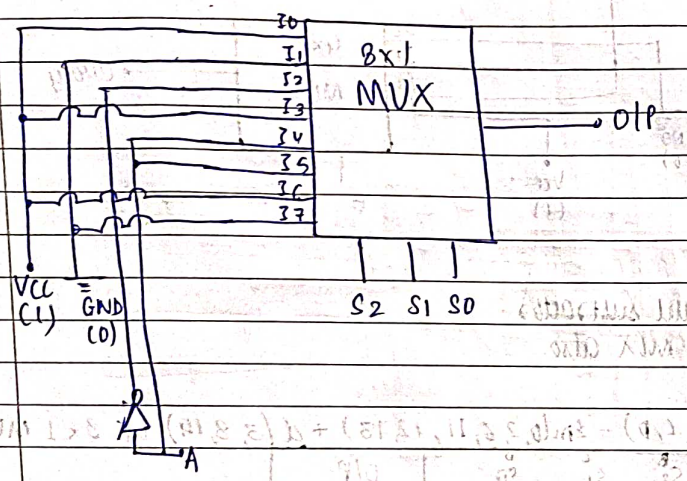
→ use 2×1 MUX also.

Q. F(A, B, C, D) = Σm(0, 2, 6, 11, 12, 13) + d(3, 8, 14) by 8 × 1 MUX

S_2 (B)	S_1 (C)	S_0 (D)	O/P
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

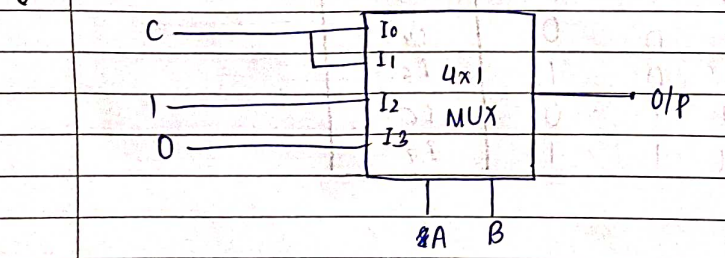
{don't care marked with value ___/___/___}

A	I_0	I_1	I_2	I_3	I_4	I_5	I_6	I_7
0	0	1	2	3	4	5	6	7
1	8	9	10	11	12	13	14	15
	1	0	$\bar{A}$	1	A	A	1	0



Logical expressions from MUX

0 →



___/___/___

soln →

S1 A	S0 B	o/p
0	0	I_0
0	1	I_1
1	0	I_2
1	1	I_3

$I_0 + I_1 + I_2 + I_3$

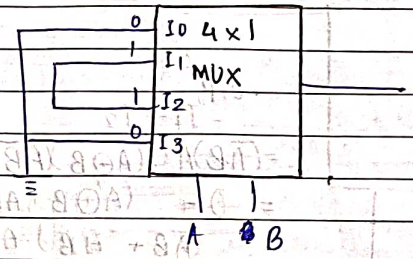
$= \bar{A}\bar{B}C + \bar{A}BC + A\bar{B}(1) + \cancel{AB(0)}^{0}$

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

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

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

0 →

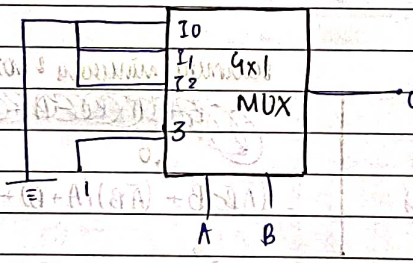


$= \cancel{I_0}^{0} + I_1 + I_2 + \cancel{I_3}^{0}$

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

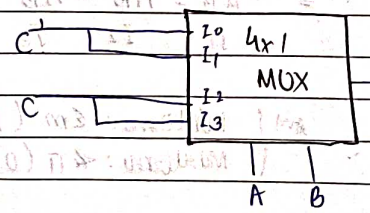
$= \boxed{A \oplus B}$ XOR gate

0 →



$= \boxed{AB}$ AND gate.

0 →



$= I_0 + I_1 + I_2 + I_3$

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

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

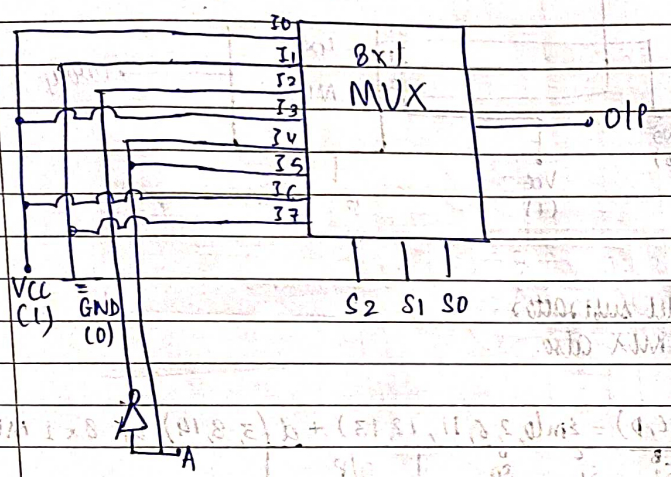
$= \bar{A}\bar{C} + AC$

$= (A \oplus C)'$

$= \boxed{A \odot C}$ XNOR gate.

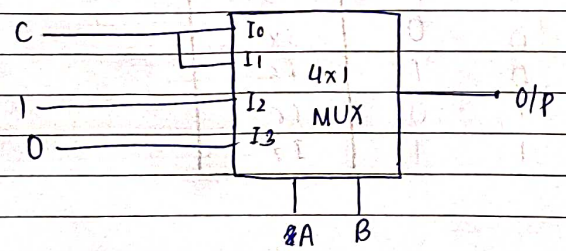
{don't care marked with value ___/___/___}

A	I_0	I_1	I_2	I_3	I_4	I_5	I_6	I_7
0	0	1	2	3	4	5	6	7
1	8	9	10	11	12	13	14	15
	1	0	$\bar{A}$	1	A	A	1	0



Logical expressions from MUX

0 →



___/___/___

soln →

S1 A	S0 B	o/p
0	0	I_0
0	1	I_1
1	0	I_2
1	1	I_3

$I_0 + I_1 + I_2 + I_3$

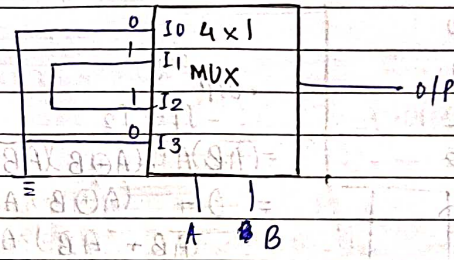
$= \bar{A}\bar{B}C + \bar{A}BC + A\bar{B}(1) + \cancel{AB(0)}^{0}$

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

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

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

0 →

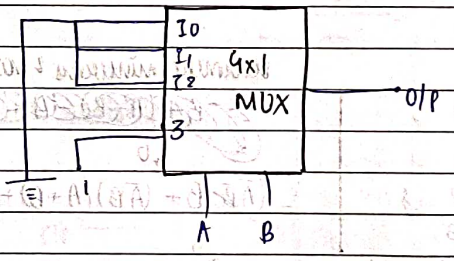


$= \cancel{I_0}^{0} + I_1 + I_2 + \cancel{I_3}^{0}$

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

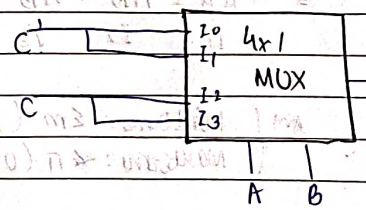
$= \boxed{A \oplus B}$ XOR gate

0 →



$= \boxed{AB}$ AND gate.

0 →



$= I_0 + I_1 + I_2 + I_3$

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

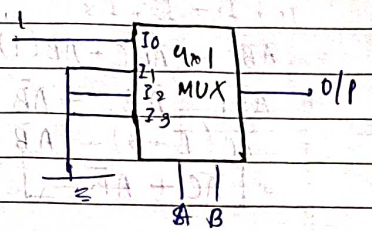
$= \bar{A}\bar{C}(\bar{B}+B) + AC(\bar{B}+B)$

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

$= (A \oplus C)'$

$= \boxed{A \odot C}$ XNOR gate.

Q →

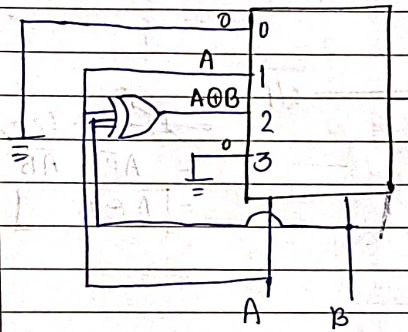


$= \overline{A}\,\overline{B}$

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

NOR gate

Q →



o/p

$= I_1 + I_2$

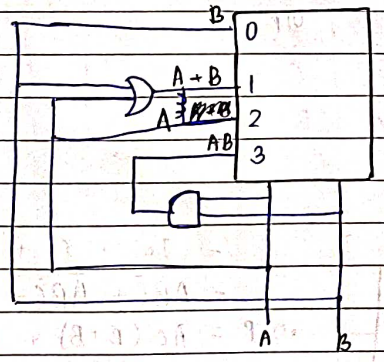
$= (\overline{A}B)A + (A \oplus B)(A\overline{B})$

$= 0 + (A \oplus B)\,A\overline{B}$

$= (\overline{A}B + A\overline{B})\,A\overline{B}$

$= \boxed{A\overline{B}}$

Q →



Determine minterms & maxterms.

$= \overset{0}{(\overline{A}\overline{B})B} + (\overline{A}B)(A+B) + (A\overline{B})A + (AB)$ (no)

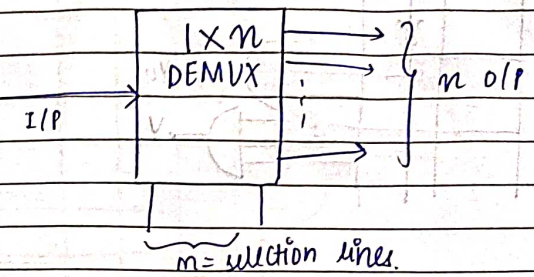
$= \overset{0}{A\overline{A}\overline{B}} + \overline{A}B + A\overline{B} + AB$

$= \underset{I_1}{\overline{A}B} + \underset{I_2}{A\overline{B}} + \underset{I_3}{AB}$

Ans. Minterms : $\Sigma m\,(1, 2, 3)$

Maxterms : $\Pi\,(0)$

De-multiplexer (DEMUX)



For DEMUX :-

I/P = 1

O/P : $n = 2^m$ selection lines

when :-

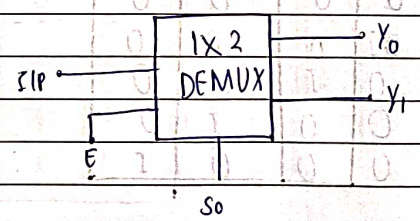
$m = 1$ 1×2 DEMUX

$m = 2$ 1×4

$m = 3$ 1×8

⋮

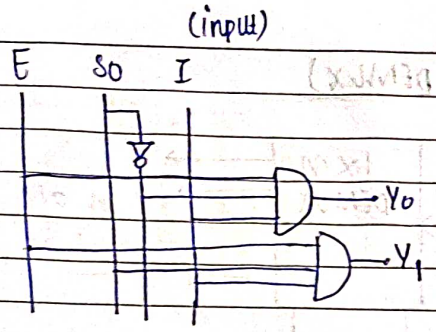
1×2 DEMUX



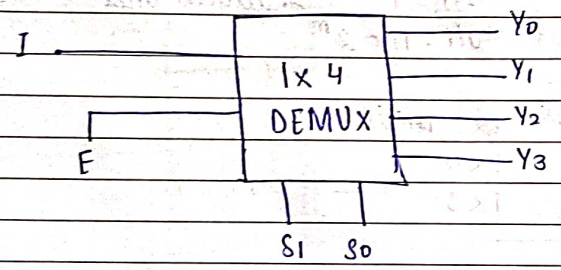
E	S_0	Y_0	Y_1
0	X	0	0
1	0	I	0
1	1	0	I

$Y_0 = E\overline{S_0}I$

$Y_1 = E S_0 I$



1x4 DEMUX



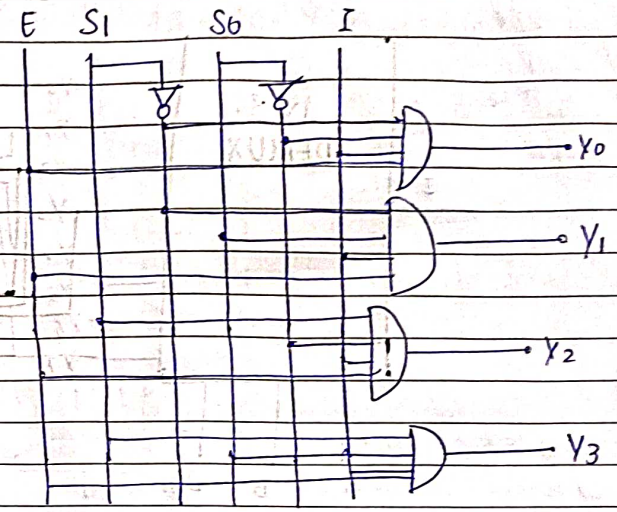
E	S_1	S_0	Y_0	Y_1	Y_2	Y_3
0	X	X	0	0	0	0
1	0	0	I	0	0	0
1	0	1	0	I	0	0
1	1	0	0	0	I	0
1	1	1	0	0	0	I

$Y_0 = \bar{S_1}\bar{S_0}IE$

$Y_1 = \bar{S_1}S_0IE$

$Y_2 = S_1\bar{S_0}IE$

$Y_3 = S_1S_0IE$

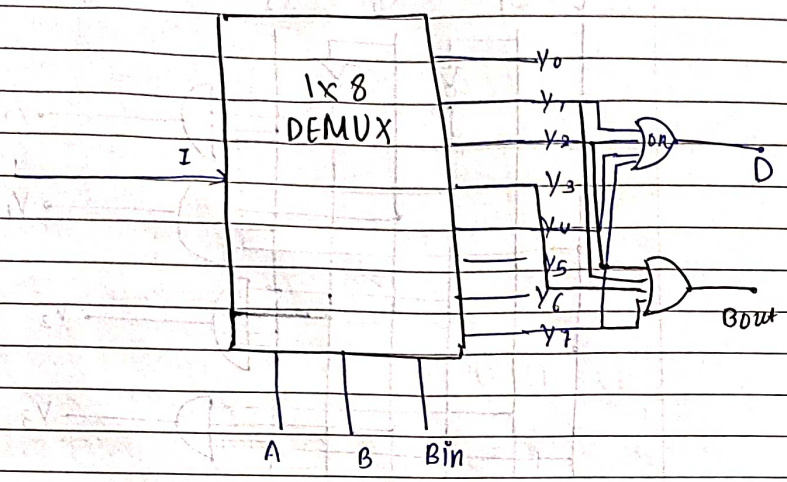


Q- Design a full subtractor by using 1x8 DEMUX.

A	B	Bin	D	Bout
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

$D = \Sigma m(1, 2, 4, 7)$

$B_{out} = \Sigma m(1, 2, 3, 7)$



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	I	0	0	0	0	0	0	0
0	0	1	0	I	0	0	0	0	0	0
0	1	0	0	0	I	0	0	0	0	0
0	1	1	0	0	0	I	0	0	0	0
1	0	0	0	0	0	0	I	0	0	0
1	0	1	0	0	0	0	0	I	0	0
1	1	0	0	0	0	0	0	0	I	0
1	1	1	0	0	0	0	0	0	0	I

Q→ $F(A,B,C) = AB + \bar{A}B + C$ by 1x8 DEMUX

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

{solve single term first eg = C in this case}

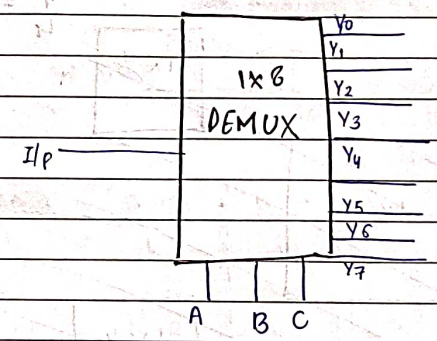
A \ BC	00	01	11	10
0	0	1 (1)	1 (3)	1 (2)
1	4	1 (5)	1 (7)	1 (6)

→ $\bar{A}B$

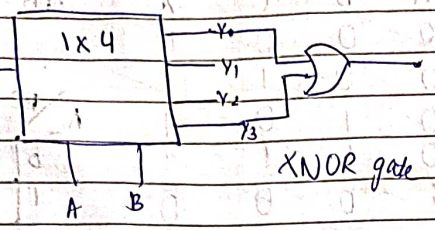
→ AB

C

$\Sigma m(1,2,3,5,6,7)$



Q- $AB + \bar{A}\bar{B}$ by 1×4 DEMUX

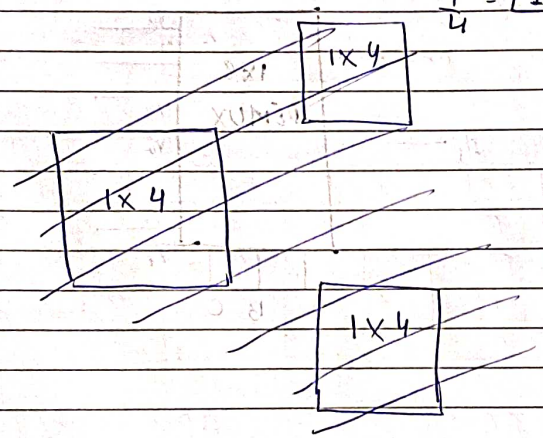


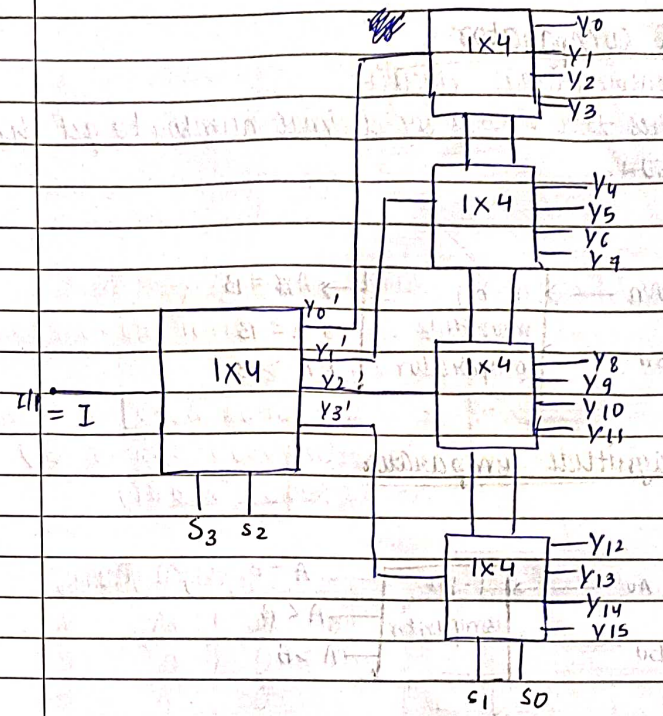
Q- Design a 1×16 DEMUX by using 1×4 DEMUX

no. of DEMUX = $\frac{\text{Required o/p}}{\text{available o/p}} = \frac{16}{4} = \boxed{4}$ level 2

$\frac{4}{4} = \boxed{1}$ level 1

{Reverse of levels}



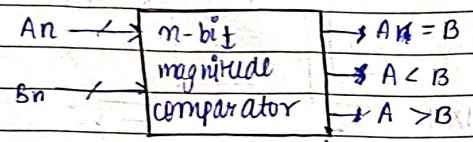


August 30 2024

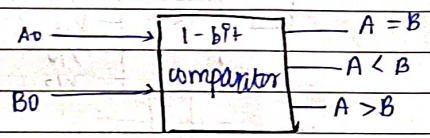
Magnitude comparator

→ It is a combinational circuit.

→ It compares two binary or decimal numbers to get the comparison.



1 bit - magnitude comparator



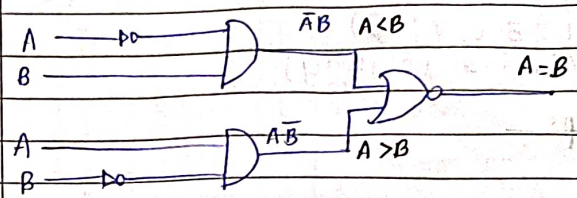
Truth Table :-

A_0	B_0	$A=B$	$A<B$	$A>B$
0	0	1	0	0
0	1	0	1	0
1	0	0	0	1
1	1	1	0	0

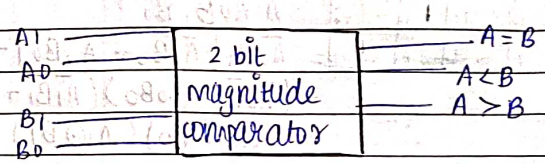
$A=B \Rightarrow \bar{A}\bar{B} + AB = A \odot B = (\bar{A}B + A\bar{B})'$

$A<B \Rightarrow \bar{A}B$

$A>B \Rightarrow A\bar{B}$



2-bit magnitude comparator



Truth Table :-

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

$A=B \Rightarrow \Sigma m(0,5,10,15)$

$A<B \rightarrow \Sigma m(1,2,3,6,7,11)$

$A>B \rightarrow \Sigma m(4,8,9,12,13,14)$

K-maps

A = B:

A_1A_0 \ B_1B_0	00	01	11	10
00	1 (0)	(1)	(3)	(2)
01	(4)	1 (5)	(7)	(6)
11	(12)	(13)	1 (15)	(14)
10	(8)	(9)	(11)	1 (10)

$= \bar{A}_1\bar{A}_0\bar{B}_1\bar{B}_0 + \bar{A}_1A_0\bar{B}_1B_0 + A_1A_0B_1B_0 + A_1\bar{A}_0B_1\bar{B}_0$

$= \bar{A}_1\bar{B}_1[\bar{A}_0\bar{B}_0 + A_0B_0] + A_1B_1[A_0B_0 + \bar{A}_0\bar{B}_0]$

$= (\bar{A}_0\bar{B}_0 + A_0B_0)(\bar{A}_1\bar{B}_1 + A_1B_1)$

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

A < B :

A_1A_0 \ B_1B_0	00	01	11	10
00	(0)	1 (1)	1 (3)	1 (2)
01	(4)	(5)	1 (7)	1 (6)
11	(12)	(13)	(15)	(14)
10	(8)	(9)	1 (11)	(10)

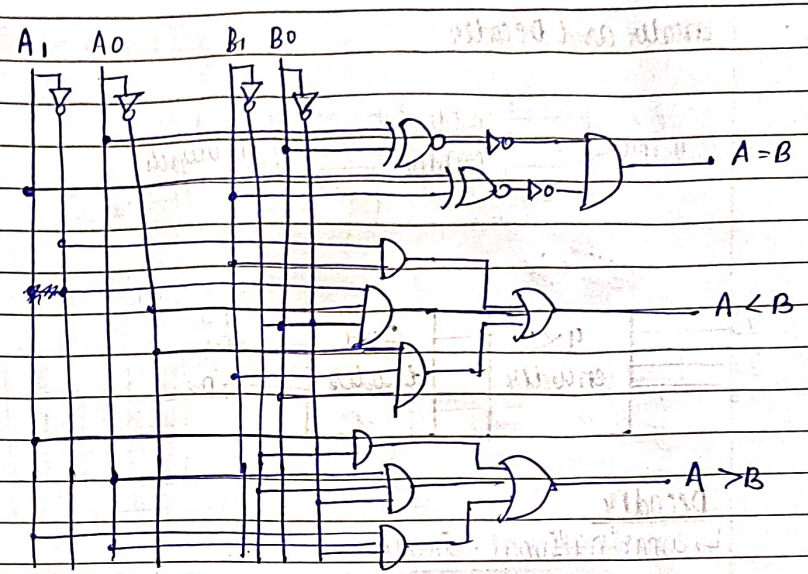
$= \bar{A}_1B_1 + \bar{A}_1\bar{A}_0B_0 + \bar{A}_0B_1B_0$

A > B :

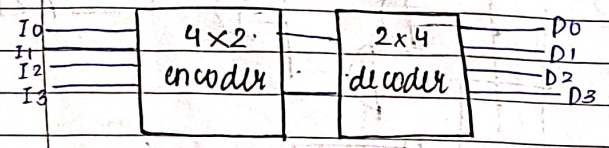
A_1A_0 \ B_1B_0	00	01	11	10
00	(0)	(1)	(3)	(2)
01	1 (4)	(5)	(7)	(6)
11	1 (12)	1 (13)	(15)	1 (11)
10	1 (8)	1 (9)	(11)	(10)

$= A_1\bar{B}_1 + A_0\bar{B}_1\bar{B}_0 + A_1A_0\bar{B}_0$

- make 3-bit & 4-bit comparator

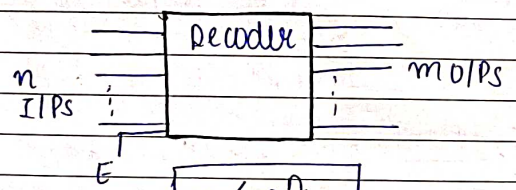


Encoder and Decoder



Decoder

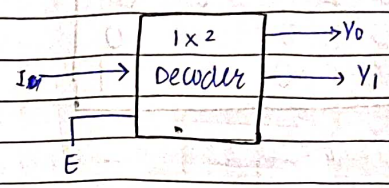
↳ combinational circuit



$m \leq 2^n$

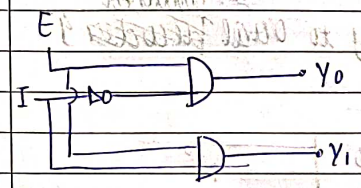
Types : 1 x 2 decoder
2 x 4 decoder
3 x 8 decoder
4 x 16 decoder

1x 2 Decoder

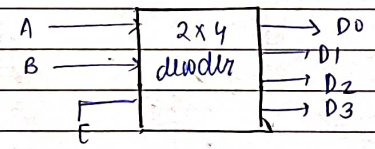


E	I	Y_0	Y_1
0	x	0	0
1	0	1	0
1	1	0	1

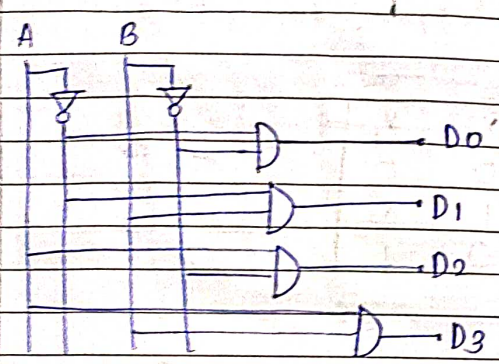
circuit :-



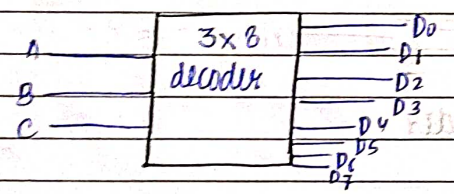
2x4 Decoder



A	B	D_0	D_1	D_2	D_3
0	0	1	0	0	0
0	1	0	1	0	0
1	0	0	0	1	0
1	1	0	0	0	1

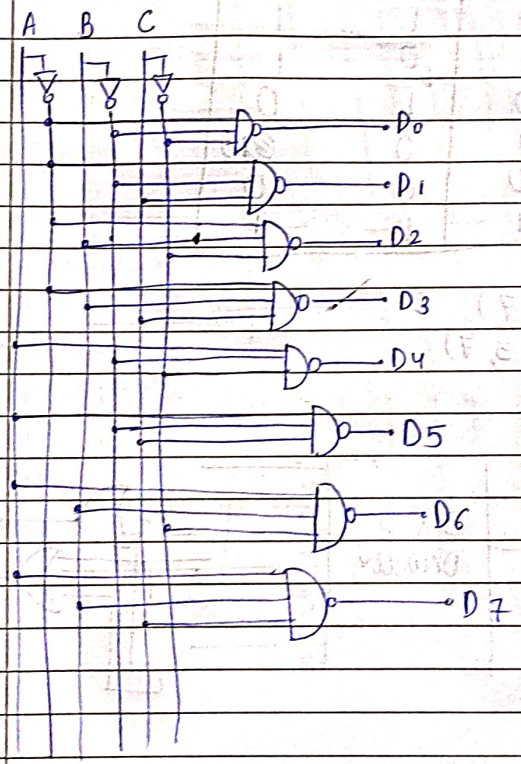


3x8 Decoder {Binary to Octal ~~Decoder~~ converter}



A	B	C	D_0	D_1	D_2	D_3	D_4	D_5	D_6	D_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	0	0	0	1	0	0	0	0
1	0	0	0	0	0	0	1	0	0	0
1	0	1	0	0	0	0	0	1	0	0
1	1	0	0	0	0	0	0	0	1	0
1	1	1	0	0	0	0	0	0	0	1

{converts binary to Octal}



September 3, 2024

Q- BCD to Decimal decoder
↳ 4x16 decoder.
Remaining output will be used as don't care.

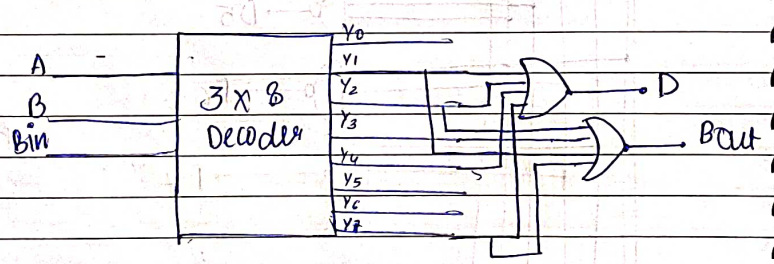
Q- Design a full subtractor by using decoder.

Soln → No. of inputs = 3
~~Decoder~~ suitable decoder = 3x8

A	B	Bin	D	Bout
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

$D = \Sigma m(1, 2, 4, 7)$

$B_{out} = \Sigma m(1, 2, 3, 7)$

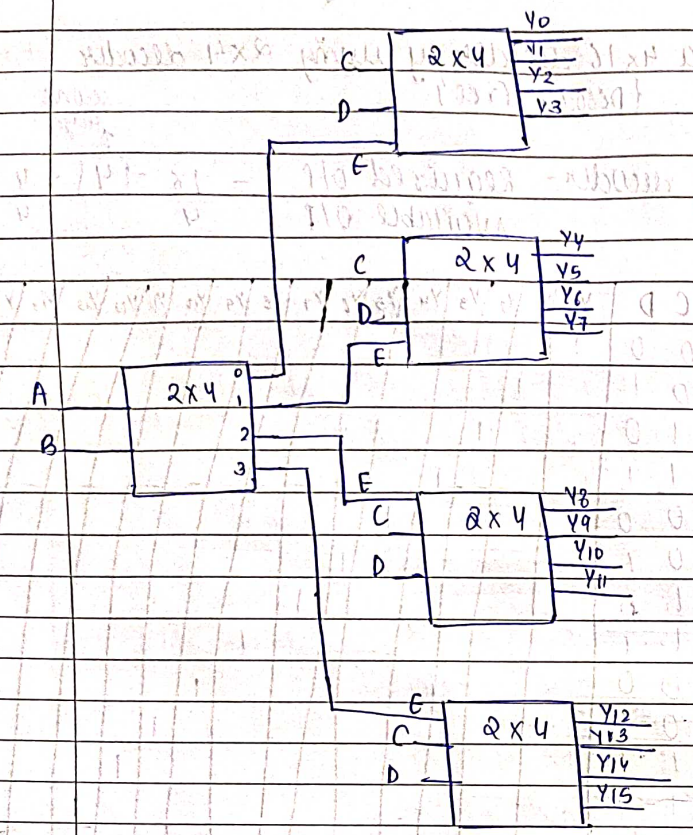


Q- Design a 4x16 decoder by using 2x4 decoder
{Decoder Tree}

$$\text{no. of decoder} = \frac{\text{Required O/P}}{\text{Available O/P}} = \frac{16}{4} = \boxed{4} = \frac{4}{4} = \boxed{1}$$

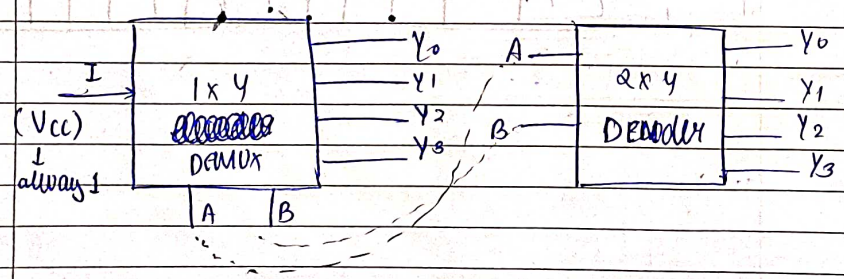
second stage → 4; first stage → 1

A	B	C	D	Y_0	Y_1	Y_2	Y_3	Y_4	Y_5	Y_6	Y_7	Y_8	Y_9	Y_{10}	Y_{11}	Y_{12}	Y_{13}	Y_{14}	Y_{15}
0	0	0	0	1															
0	0	0	1		1														
0	0	1	0			1													
0	0	1	1					1											
0	1	0	0						1										
0	1	0	1							1									
0	1	1	0								1								
0	1	1	1										1						
1	0	0	0											1					
1	0	0	1												1				
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																



Q- Design a DEMUX to a decoder

1x4 2x4



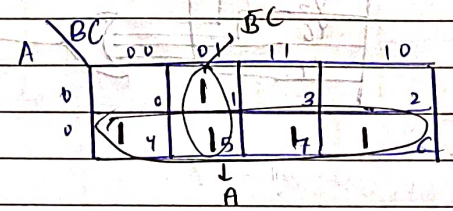
Q- $A + \overline{B}C$ by 3x8 decoder

A	B	C	$\overline{B}$	$\overline{B}C$	$A + \overline{B}C$
0	0	0	1	0	0
0	0	1	1	1	1
0	1	0	0	0	0
0	1	1	0	0	0
1	0	0	1	0	1
1	0	1	1	1	1
1	1	0	0	0	1
1	1	1	0	0	1

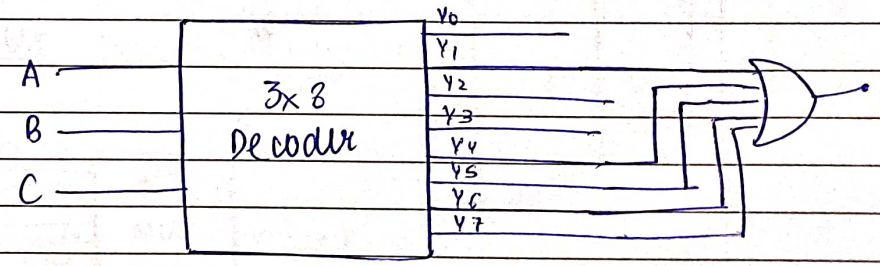
$= \Sigma m(1, 4, 5, 6, 7)$

OR

directly find minterms by K-map



$= \Sigma m(1, 4, 5, 6, 7)$

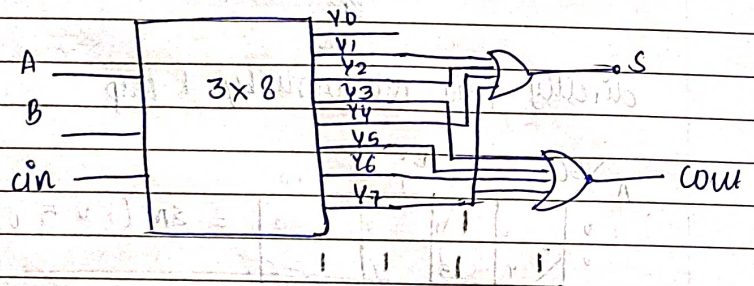


Q- make full adder

A	B	Cin	S	Cout
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

$S = \Sigma m(1, 2, 4, 7)$

$Cout = \Sigma m(3, 5, 6, 7)$



Practice

Q- $F(A, B, C) = \Sigma m(1, 2, 6, 7)$ using :-

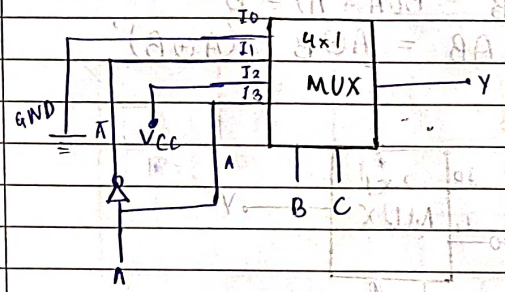
4x1 MUX

no. of variable = 3

S_1	S_0	O/P
0	0	I_0
0	1	I_1
1	0	I_2
1	1	I_3

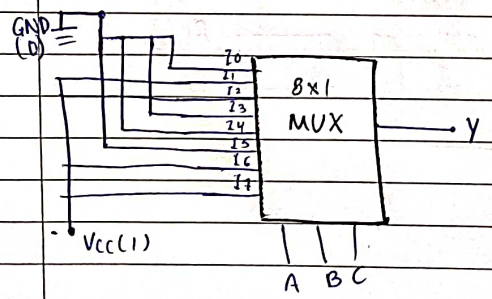
Implementation Table :-

	00	01	10	11
A	I_0	I_1	I_2	I_3
0	0	1	2	3
1	4	5	6	7
	0	$\bar{A}$	1	A



A	B	C	O/P
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

8x1 MUX



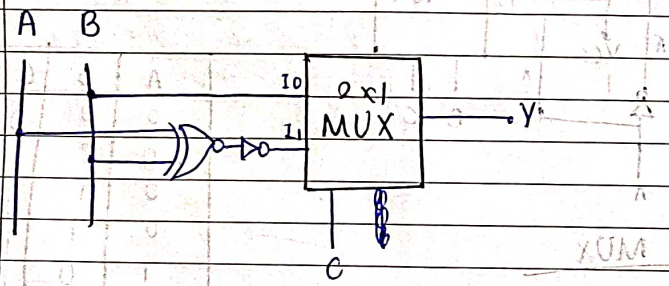
2×1 MUX

C	O/P
0	I_0
1	I_1

A	B	I_0	I_1
0	0	0	1
0	1	2	3
1	0	4	5
1	1	6	7
		B	$(A \oplus B)'$

$I_0 = \bar{A}B + AB = B(A + \bar{A}) = B$

$I_1 = \bar{A}\bar{B} + AB = A \odot B = (A \oplus B)'$



Full subtractor using 4×1 MUX

A	B	Bin	D	Bout
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

$D = \Sigma m(1, 2, 5, 7)$

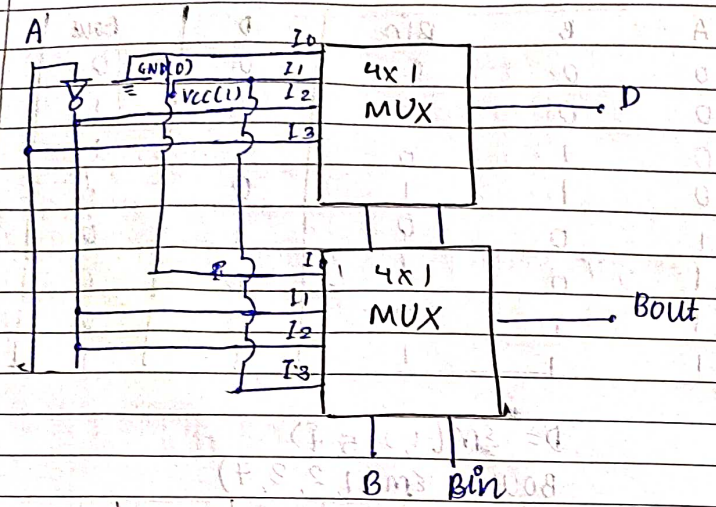
$Bout = \Sigma m(1, 2, 3, 7)$

For D:

	00	01	10	11
A	I_0	I_1	I_2	I_3
0	0	1	2	3
1	4	5	6	7
	0	1	$\bar{A}$	A

For Bout:

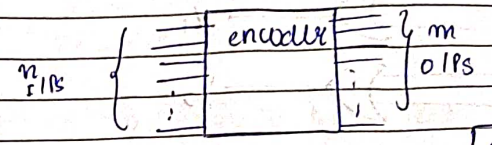
	00	01	10	11
A	I_0	I_1	I_2	I_3
0	0	1	2	3
1	4	5	6	7
	0	$\bar{A}$	$\bar{A}$	1



- 1x8 DEMUX
- Adder using DEMUX
- Design 4x1 MUX by NAND gate
- BCD to decimal decoder

September 4, 2024

Encoder

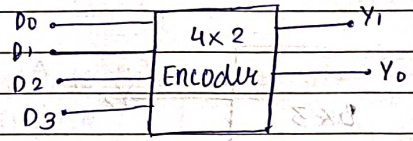


$n <= 2^m$

Eg:
4x2 encoder
8x3 encoder
16x4 encoder

{output will be the binary representation of the input}

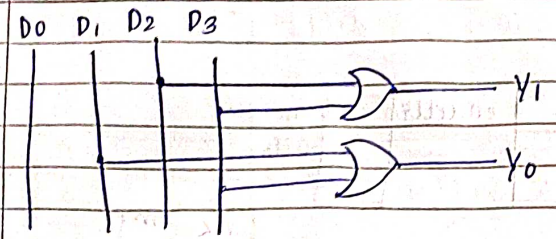
4x2 encoder



Truth Table:

D_0	D_1	D_2	D_3	Y_1	Y_0	decimal:-
1	0	0	0	0	0	→ 0
0	1	0	0	0	1	→ 1
0	0	1	0	1	0	→ 2
0	0	0	1	1	1	→ 3

$Y_1 = D_2 + D_3$
$Y_0 = D_1 + D_3$

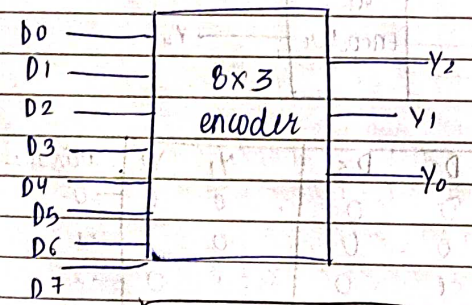


Types of Encoder

1) Priority encoder
2) Octal to Binary encoder
3) Decimal to BCD encoder.
4) Decimal to Binary

Octal to Binary Encoder

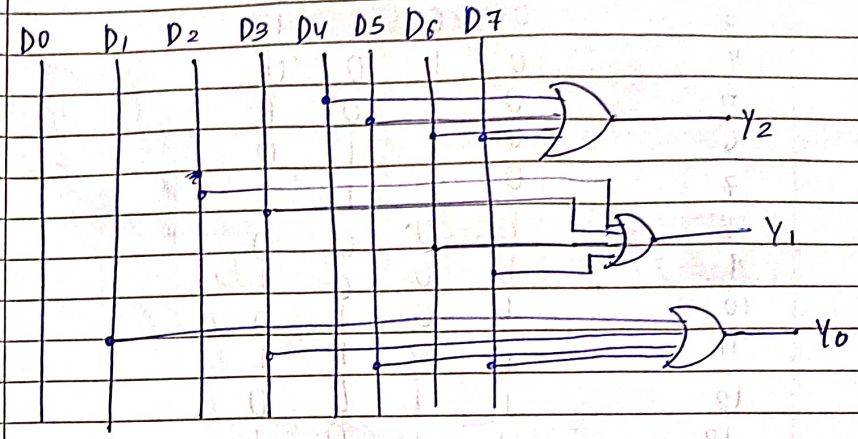
suitable: 8x3 encoder



or make D0 to D7

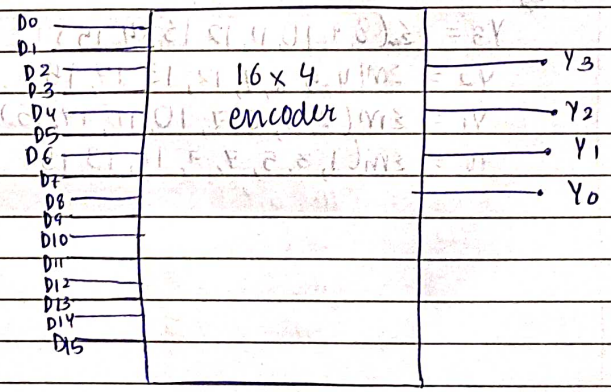
decimal	Y_2	Y_1	Y_0
0	0	0	0
1	0	0	1
2	0	1	0
3	0	1	1
4	1	0	0
5	1	0	1
6	1	1	0
7	1	1	1

$Y_2 = \Sigma m(4, 5, 6, 7)$
$Y_1 = \Sigma m(2, 3, 6, 7)$
$Y_0 = \Sigma m(1, 3, 5, 7)$



Hexadecimal to Binary Encoder

16 x 4 Encoder



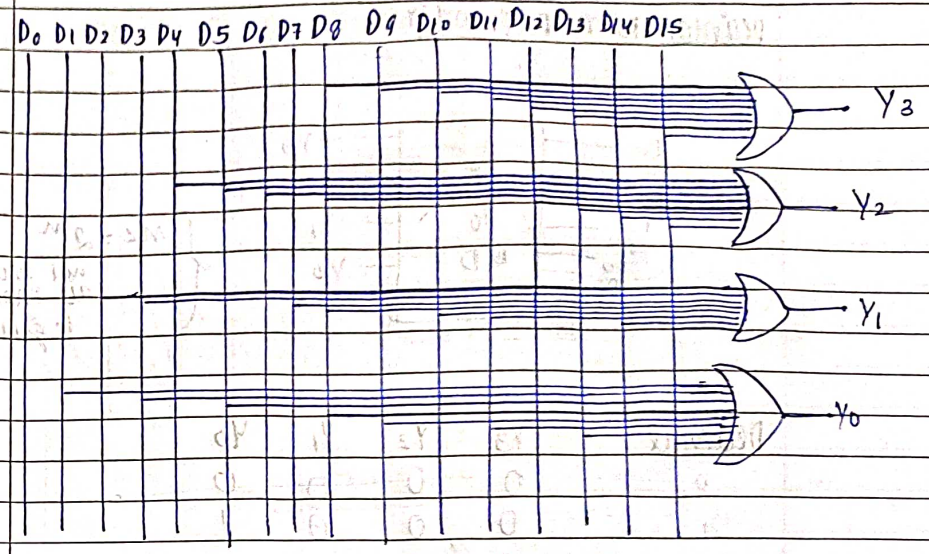
Decimal	Y_3	Y_2	Y_1	Y_0
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1
10	1	0	1	0
11	1	0	1	1
12	1	1	0	0
13	1	1	0	1
14	1	1	1	0
15	1	1	1	1

$Y_3 = \Sigma m(8, 9, 10, 11, 12, 13, 14, 15)$

$Y_2 = \Sigma m(4, 5, 6, 7, 12, 13, 14, 15)$

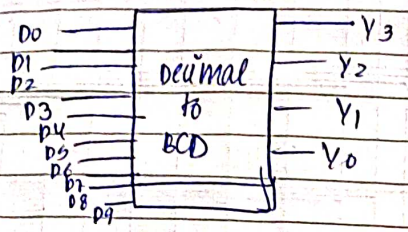
$Y_1 = \Sigma m(2, 3, 6, 7, 10, 11, 14, 15)$

$Y_0 = \Sigma m(1, 3, 5, 7, 9, 11, 13, 15)$



{circuit}

Decimal to BCD Encoder



$n < = 2^m$

not necessarily all inputs are to be given

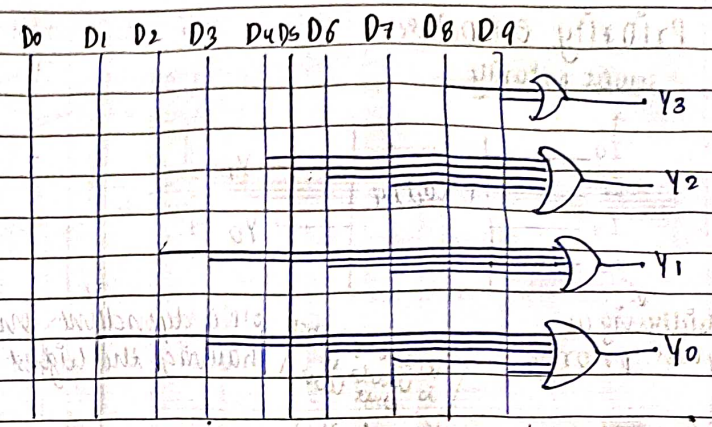
Decimal	Y_3	Y_2	Y_1	Y_0
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1

$Y_3 = \Sigma m(8, 9)$

$Y_2 = \Sigma m(4, 5, 6, 7)$

$Y_1 = \Sigma m(2, 3, 6, 7)$

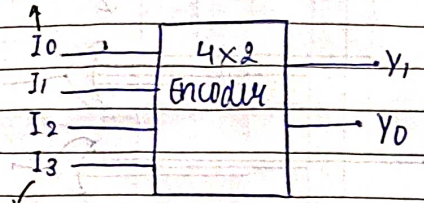
$Y_0 = \Sigma m(1, 3, 5, 7, 9)$



{ circuit }

Priority encoder

lowest priority



considering as highest priority

o/p is dependant on I_3 as it is having the highest priority.

because priority is lowest so dont care

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

For Y_1:

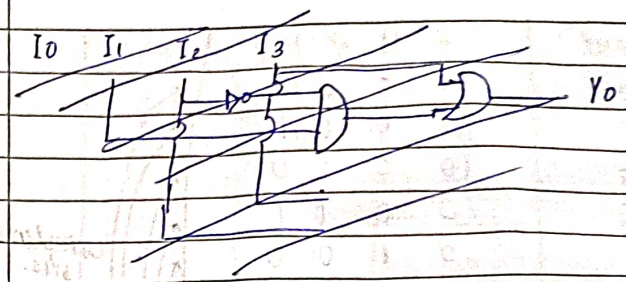
$I_3 I_2$ \ $I_1 I_0$	00	01	11	10
00	d 0	0 1	0 3	0 2
01	1 4	1 5	1 7	1 6
11	1 12	1 13	1 15	1 14
10	1 8	1 9	1 11	1 10

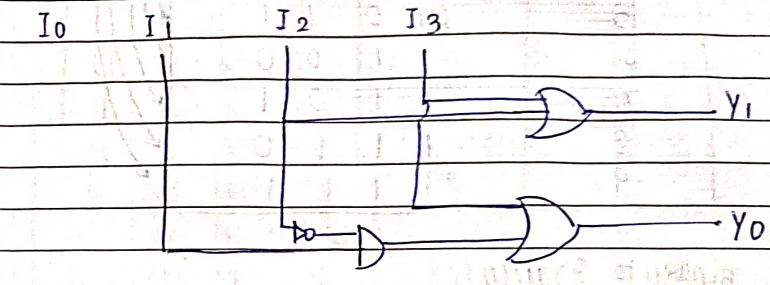
$Y_1 = I_2 + I_3$

For Y_0:

$I_3 I_2$ \ $I_1 I_0$	00	01	11	10
00	d 0	0 1	1 3	1 2
01	0 4	0 5	0 7	0 6
11	1 12	1 13	1 15	1 14
10	1 8	1 9	1 11	1 10

$Y_0 = I_3 + I_2 I_1$





Code conversion

codes {group of symbols}

Classification:

- Weighted codes — Eg: 8421 (BCD), 2421 → positional value
- Non-weighted codes — Eg: Excess-3 or XS-3, Gray code
- Reflective codes (self complementing) — Eg: 2421, XS-3
- Sequential codes — Eg: 8421, XS-3
- Alpha-numeric — Eg: ASCII
- Error correction and detection — Eg: Hamming code

[Parity is an extra bit which can be even or odd]
↳ comes under error detection.

Reflecting / self-complementing code :

Decimal	2	4	2	1
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	1	0	1	1
6	1	1	0	0
7	1	1	0	1
8	1	1	1	0
9	1	1	1	1

complement of bits.

Binary to Graycode

↳ Reflective Binary code.

{ successive bits should have change of only 1-bit }

Binary : 1 0 1 0

as it is ↓ ↓ ↓ ↓

Gray : 1 1 1 1

Eg : 1 0 1 0 1 1

↓ ↓ ↓ ↓ ↓ ↓

1 1 1 1 1 0

For conversion: 3-bit binary to 3-bit Gray code

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

$g_2 = \Sigma m(4,5,6,7)$

$g_1 = \Sigma m(2,3,4,5)$

$g_0 = \Sigma m(1,2,5,6)$

K-map for simplication to obtain expression :-

For g_2 :

b_2 \ b_1b_0	00	01	11	10
0	0	1	3	2
1	1 (4)	1 (5)	1 (7)	1 (6)

$= b_2$

For g_1 :

b_2 \ b_1b_0	00	01	11	10
0	0	1	1 (3)	1 (2)
1	1 (4)	1 (5)	7	6

$= \bar{b_2}b_1 + b_2\bar{b_1}$

$= b_2 \oplus b_1$

For g_0 :

b_2 \ b_1b_0	00	01	11	10
0	0	1 (1)	3	1 (2)
1	4	1 (5)	7	1 (6)

$= \bar{b_1}b_0 + b_1\bar{b_0}$

$= b_1 \oplus b_0$

Binary to graycode

4-bit Binary to Gray code converter

b_3	b_2	b_1	b_0	g_3	g_2	g_1	g_0
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1
0	0	1	0	0	0	1	1
0	0	1	1	0	0	1	0
0	1	0	0	0	1	1	0
0	1	0	1	0	1	1	1
0	1	1	0	0	1	0	1
0	1	1	1	0	1	0	0
1	0	0	0	1	1	0	0
1	0	0	1	1	1	0	1
1	0	1	0	1	1	1	1
1	0	1	1	1	1	1	0
1	1	0	0	1	0	1	0
1	1	0	1	1	0	1	1
1	1	1	0	1	0	0	1
1	1	1	1	1	0	0	0

$g_3 = \Sigma m(8, 9, 10, 11, 12, 13, 14, 15)$

$g_2 = \Sigma m(4, 5, 6, 7, 8, 9, 10, 11)$

$g_1 = \Sigma m(2, 3, 4, 5, 10, 11, 12, 13)$

$g_0 = \Sigma m(1, 2, 5, 6, 9, 10, 13, 14)$

For g_3:

b_3b_2 \ b_1b_0	00	01	11	10
00	0	1	3	2
01	4	5	7	6
11	1 12	1 13	1 15	1 14
10	1 8	1 9	1 11	1 10

$= b_3$

For g_2:

b_3b_2 \ b_1b_0	00	01	11	10
00	0	1	3	2
01	1 4	1 5	1 7	1 6
11	12	13	15	14
10	1 8	1 9	1 11	1 10

$= b_3 b_2 + b_3 \bar{b_2}$

$= b_3 \oplus b_2$

For g_1:

b_3b_2 \ b_1b_0	00	01	11	10
00	0	1	1 3	1 2
01	1 4	1 5	7	6
11	1 12	1 13	15	14
10	8	9	1 11	1 10

$= b_2 \bar{b_1} + \bar{b_2} b_1$

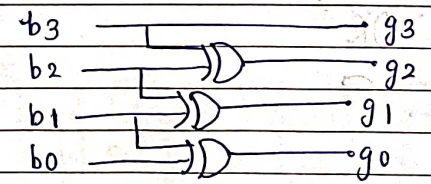
$= b_2 \oplus b_1$

For g_0:

b_3b_2 \ b_1b_0	00	01	11	10
00	0	1 1	3	1 2
01	4	1 5	7	1 6
11	12	1 13	15	1 14
10	8	1 9	11	1 10

$= \bar{b_1} b_0 + b_1 \bar{b_0}$

$= b_1 \oplus b_0$



September 5, 2024

Gray code to Binary

1 0 1 0 → gray code

as it is ↓ ⊕ ⊕ ⊕

1 1 0 0 → binary code

3-bit Gray to Binary code converter

	g_2	g_1	g_0	b_2	b_1	b_0	
0	0	0	0	0	0	0	0
1	0	0	1	0	0	1	1
2	0	1	0	0	1	1	3
3	0	1	1	0	1	0	2
4	1	0	0	1	1	1	7
5	1	0	1	1	1	0	6
6	1	1	0	1	0	0	4
7	1	1	1	1	0	1	5

~~$b_2 = \Sigma m(4,5,6,7)$~~
~~$b_1 = \Sigma m(2,3,4,5)$~~
~~$b_0 = \Sigma m(1,2,4,7)$~~

~~$g_2 = \Sigma m(4,5,6,7)$~~
~~$g_1 = \Sigma m(2,3,4,5)$~~
~~$g_0 = \Sigma m(1,2,4,7)$~~

OR

Table can also be made this way, but the decimal numbering must be taken care of.

	g_2	g_1	g_0	b_2	b_1	b_0	
0	0	0	0	0	0	0	0
1	0	0	1	0	0	1	1
3	0	1	1	0	1	0	2
2	0	1	0	0	1	1	3
6	1	1	0	1	0	0	4
7	1	1	1	1	0	1	5
5	1	0	1	1	1	0	6
4	1	0	0	1	1	1	7

$g_2 = \Sigma m(4,5,6,7)$
$g_1 = \Sigma m(2,3,4,5)$
$g_0 = \Sigma m(1,2,4,7)$

g_2 \ g_1g_0	00	01	11	10
0	0	1	3	2
1	1 4	1 5	1 7	1 6

$\boxed{b_2 = g_2}$

g_2 \ g_1g_0	00	01	11	10
0	0	1	1 3	1 2
1	1 4	1 5	7	6

$= \bar{g_2} g_1 + g_2 \bar{g_1}$
$= g_2 \oplus g_1$
$b_1 = \boxed{b_2 \oplus g_1}$
as $g_2 = b_2$

g_2 \ g_1g_0				
	0	1	3	1 2
	1 4	5	1 7	6

$= \bar{g_2}g_1g_0 + \bar{g_2}g_1\bar{g_0} + g_2\bar{g_1}\bar{g_0} + g_2g_1g_0$

$$= g_0(\bar{g_2}\bar{g_1} + g_2 g_1) + \bar{g_0}(\bar{g_2} g_1 + g_2 \bar{g_1})$$

$$= g_0(g_2 \odot g_1) + \bar{g_0}(g_2 \oplus g_1)$$

$$= g_0 b_1 + \bar{g_0} b_1$$

$$\boxed{b_0 = g_0 \oplus b_1}$$

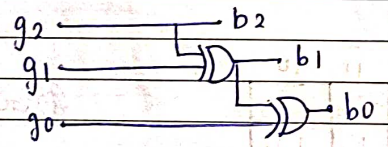
Final expressions:-

$b_2 = g_2$

$b_1 = b_2 \oplus g_1$

$b_0 = b_1 \oplus g_0$

circuit:-



BCD to Gray

	A	B	C	D	W	X	Y	Z
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

10-15 are don't care

$W = \Sigma m(8,9) + \Sigma d(10,11,12,13,14,15)$

$X = \Sigma m(4,5,6,7,8,9) + \Sigma d(10,11,12,13,14,15)$

$Y = \Sigma m(2,3,4,5) + \Sigma d(10,11,12,13,14,15)$

$Z = \Sigma m(1,2,5,6,9) + \Sigma d(10,11,12,13,14,15)$

For W:

AB \ CD	00	01	11	10
00	0	1	3	2
01	4	5	7	6
11	d_{12}	d_{13}	d_{15}	d_{14}
10	8	9	d_{11}	d_{10}

$\Rightarrow W = A$

FOR X:

AB \ CD	00	01	11	10
00	0	1	3	2
01	1 4	1 5	1 7	1 6
11	d 12	d 13	d 15	d 14
10	1 8	1 9	d 11	d 10

$= A + B$ {behaves like XOR only as XOR is add without carry}

FOR Y:

AB \ CD				
	0	1	1 3	1 2
	1 4	1 5	7	6
	d 12	d 13	d 15	d 14
	8	9	d 11	d 10

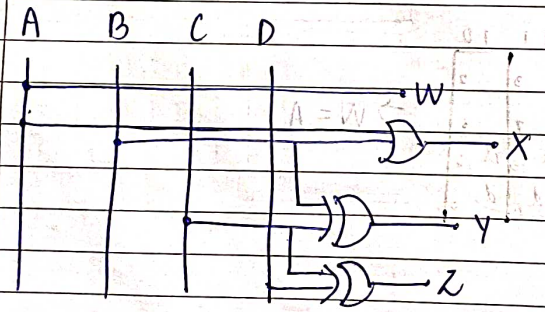
$= B\bar{C} + C\bar{B}$

$= B \oplus C$

FOR Z:

AB \ CD				
	0	1 1	3	1 2
	4	1 5	7	1 6
	d 12	d 13	d 15	d 14
	8	1 9	d 11	d 10

$= C \oplus D$



Practice

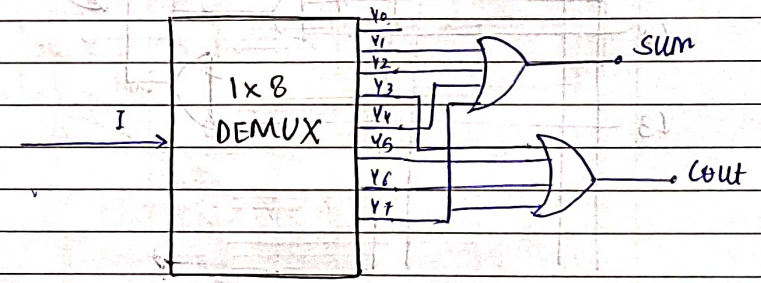
Adder using 1x8 DEMUX

Truth Table:-

A	B	Cin	S	Cout
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

$S = \Sigma(1, 2, 4, 7)$

$Cout = \Sigma(3, 5, 6, 7)$



AB \ CD	00	01	11	10
00	D0 0	D1 1	D3 3	D2 2
01	D4 4	D5 5	D7 7	D6 6
11	d 12	d 13	d 15	d 14
10	D8 8	D9 9	d 11	d 10

~~D0 = AB'C'D'~~

~~D1 = A~~

$D_0 = \bar{A}\bar{B}\bar{C}\bar{D}$

$D_1 = \bar{A}\bar{B}\bar{C}D$

$D_2 = \bar{B}C\bar{D}$

$D_3 = \bar{B}CD$

$D_4 = B\bar{C}\bar{D}$

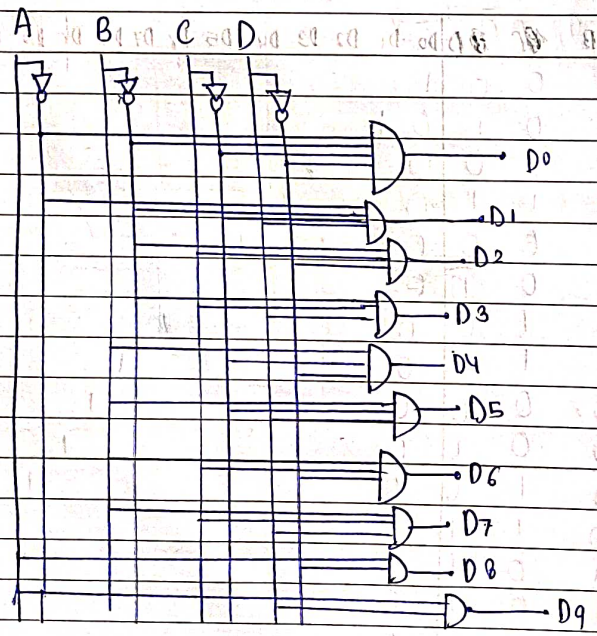
$D_5 = B\bar{C}D$

$D_6 = BC\bar{D}$

$D_7 = BCD$

$D_8 = A\bar{D}$

$D_9 = AD$



~~September 10, 2024~~

BCD to XS-3

$(\quad)_{10} \longrightarrow (\quad)_{BCD} \xrightarrow{\text{add } 3} (\quad)_{XS-3}$

Decimal	BCD A	B	C	D	XS-3 W	X	Y	Z	Decimal
0	0	0	0	0	0	0	1	1	3
1	0	0	0	1	0	1	0	0	4
2	0	0	1	0	0	1	0	1	5
3	0	0	1	1	0	1	1	0	6
4	0	1	0	0	0	1	1	1	7
5	0	1	0	1	1	0	0	0	8
6	0	1	1	0	1	0	0	1	9
7	0	1	1	1	1	0	1	0	10
8	1	0	0	0	1	0	1	1	11
9	1	0	0	1	1	1	0	0	12

(10 to 15 don't care.)

$W = \Sigma m(5, 6, 7, 8, 9) + d(\quad)$

$X = \Sigma m(1, 2, 3, 4, 9) + d(\quad)$

$Y = \Sigma m(0, 3, 4, 7, 8) + d(\quad)$

$Z = \Sigma m(0, 2, 4, 6, 8) + d(\quad)$

for W:

AB \ CD	00	01	11	10
00	0	1	3	2
01	4	1 5	1 7	1 6
11	12	13	15	14
10	1 8	1 9	11	10

~~W = ABD + ABC + ABC~~

For W:

AB \ CD	00	01	11	10
00	0	1	3	2
01	4	1 5	1 7	1 6
11	d_{12}	d_{13}	d_{15}	d_{14}
10	1 8	1 9	d_{11}	d_{10}

$W = A + BD + BC$

For X

AB \ CD	00	01	11	10
00	0	1 1	1 3	1 2
01	1 4	5	7	6
11	d_{12}	d_{13}	d_{15}	d_{14}
10	8	1 9	d_{11}	d_{10}

$X = \bar{B}D + \bar{B}C + B\bar{C}D$

For Y:

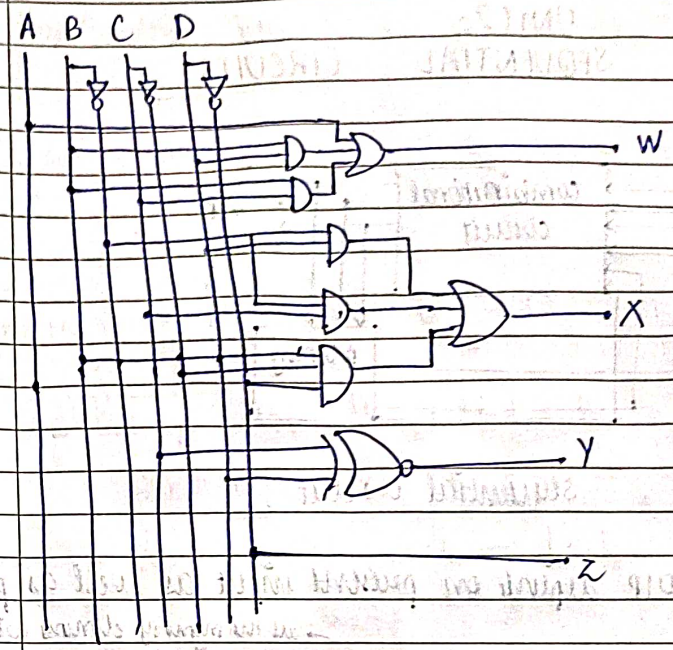
AB \ CD	00	01	11	10
00	1 0	1	1 3	2
01	1 4	5	1 7	6
11	d_{12}	d_{13}	d_{15}	d_{14}
10	1 8	9	d_{11}	d_{10}

$Y = \bar{C}\bar{D} + CD = C \odot D$

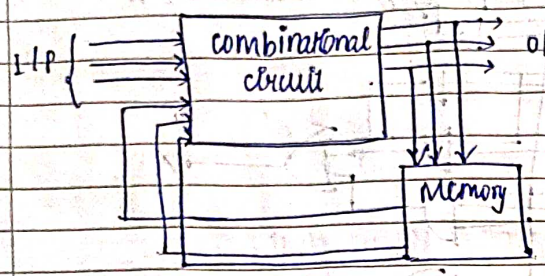
For Z:

AB \ CD	00	01	11	10
00	1 0	1	3	1 2
01	1 4	5	7	1 6
11	d_{12}	d_{13}	d_{15}	d_{14}
10	1 8	9	d_{11}	d_{10}

$Z = \bar{D}$



UNIT 2:
SEQUENTIAL CIRCUITS

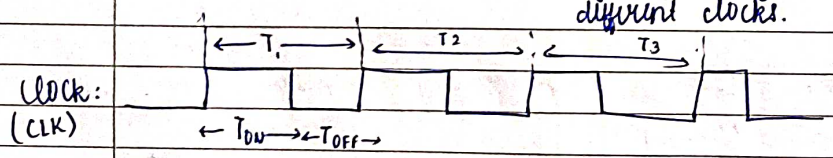


sequential circuit

↓

present O/P depends on present input as well as previous output.

~~Sy~~ sequential circuits → synchronous → all the memory elements will have same clock.
→ Asynchronous → all the memory elements have different clocks.



T_{ON} → Time period for ON
T_{OFF} → Time period for OFF
T → Time period.
Unit : sec

$$f = \frac{1}{T}$$ Unit : Hz

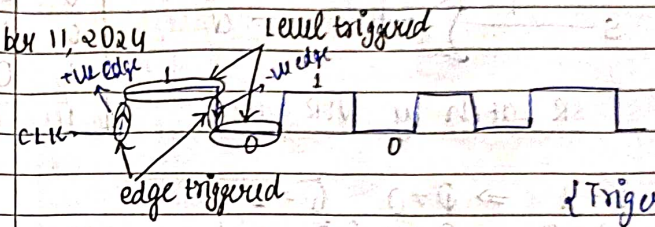
f → frequency

Duty cycle = $\frac{T_{ON}}{T}$

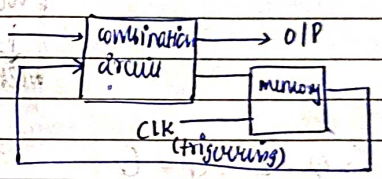
For square pulse, $\frac{T_{ON}}{2T_{ON}} = \frac{1}{2} = 0.5 = 50\%$

{considered as a best clock} as stays 50% for high and 50% for low

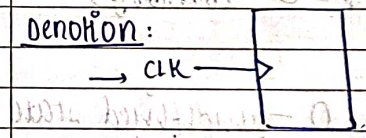
September 11, 2024



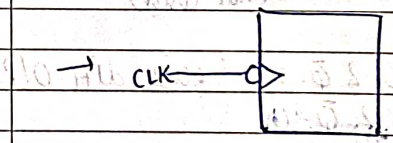
{Triggering → change of state}



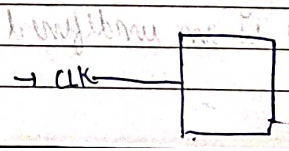
Denotion:



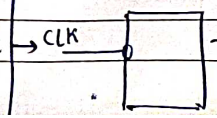
+ve edge triggering



-ve edge triggering



+ve level



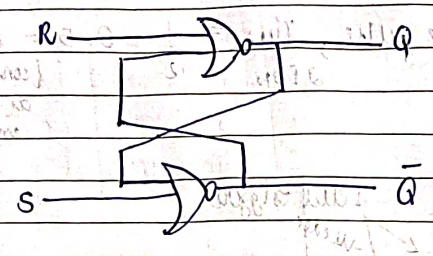
-ve level

SR latch

{R with Q, S with $\bar{Q}$}

Set → S, Reset → R

Using NOR gate



{Imp even single 1 gives o/p = 0}

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

SR latch by NOR gate

case 1: $S=0$, $R=1 \Rightarrow Q=0$, $\bar{Q}=1$

$S=0$, $R=0 \Rightarrow Q=0$, $\bar{Q}=1$ {same o/p as only i/p was changed}

↓ (behaves as)

Memory

case 2: $S=1$, $R=0 \Rightarrow Q=1$, $\bar{Q}=0$

$S=0$, $R=0 \Rightarrow Q=1$, $\bar{Q}=0$ (memory)

case 3: $S=1$, $R=1 \Rightarrow Q=0$, $\bar{Q}=0$ → undefined state (not used)

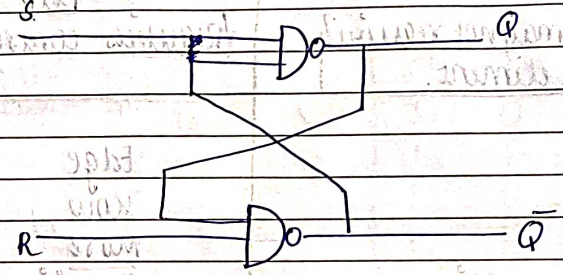
$S=0$, $R=0 \Rightarrow Q=0$ & $\bar{Q}=1$ {give diff. o/ps}
$Q=1$ & $\bar{Q}=0$

∴ $S=1$, $R=1$ is an undefined state

Truth Table for SR Latch

S	R	Q	$\bar{Q}$
0	0	memory	
0	1	0	1
1	0	1	0
1	1	Not used	

Using NAND gate



{S with Q, R with $\bar{Q}$}

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

{Imp. every 0 gives o/p 1 (even single)}

case 1: $S=0$, $R=1 \Rightarrow Q=1$, $\bar{Q}=0$

$S=1$, $R=1 \Rightarrow Q=1$, $\bar{Q}=0$ (Memory)

case 2: $S=1$, $R=0 \Rightarrow Q=0$, $\bar{Q}=1$

$S=1$, $R=1 \Rightarrow Q=0$, $\bar{Q}=1$ (memory)

case 3: $S=0$, $R=0 \Rightarrow Q=1$, $\bar{Q}=1$ {Not used}

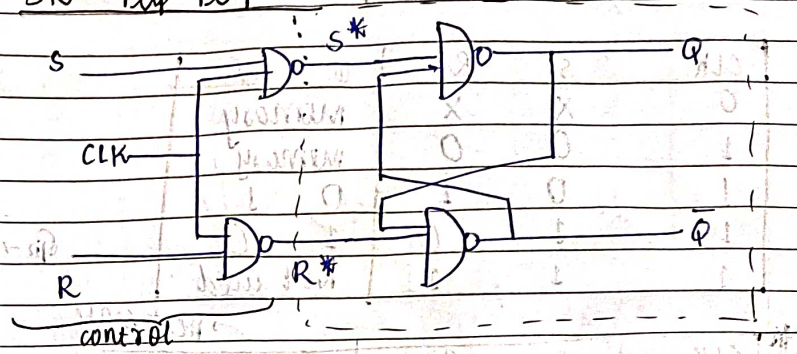
Imp. Truth Table for SR Latch using NAND

S	R	Q	$\bar{Q}$
0	0	Not used	
0	1	1	0
1	0	0	1
1	1	Memory	

Parameters	Latch	Flip Flops
	{May or may not require control element.}	{Requires control element}
1) Triggering	Level	Edge
2) Speed	Fast	Slow
3) No. of gates	Less	More
4) Sensitivity	Pulse duration	Transition
5) Designing	Easy	complex
6) Application	computing, storage	Registers, counters, storage devices

Similarity: → Storage (1 bit)
→ Bistable multivibrator

SR Flip Flops



Truth table of NAND (Latch):

S*	R*	Q	$\bar{Q}$
0	0	Not used	
0	1	1	0
1	0	0	1
1	1	Memory	

$S^* = \overline{S \cdot CLK}$
$= \bar{S} + \overline{CLK}$

$R^* = \overline{R \cdot CLK}$
$= \bar{R} + \overline{CLK}$

Case 1: CLK = 0
$S^* = \bar{S} + 1 = 1$
$R^* = \bar{R} + 1 = 1$
{Memory}

Case 2: CLK = 1
$S^* = \bar{S} + 0 = \bar{S}$
$R^* = \bar{R} + 0 = \bar{R}$

I) $S = 0, R = 0 \Rightarrow S^* = 1, R^* = 1$ (memory)

Truth Table of SR Flip Flop:-

CLK	S	R	Q	$\bar{Q}$
0	X	X	Memory	
1	0	0	Memory	
1	0	1	0	1
1	1	0	1	0
1	1	1	Not used	

Q_n → present state

Other way of making this Truth Table *

CLK	S	R	Q_{n+1}
0	X	X	Q_n
1	0	0	Q_n
1	0	1	0
1	1	0	1
1	1	1	Not used

Q_{n+1} → next state

made using this table

Characteristic Table

For CLK = 1

Q_n	S	R	Q_{n+1}
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	X
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	X

Imp.

Excitation Table

{ made using characteristic table }

Q_n	Q_{n+1}	S	R
0	0	0	X
0	1	1	0
1	0	0	1
1	1	X	0

* Using characteristic table, $Q_{n+1} = \Sigma m(2,4,6) + d(3,7)$

Q_n \ SR	00	01	11	10
0			d	1
1	1		d	1

$\boxed{Q_{n+1} = S + \bar{R}Q_n}$

* Using excitation table,

S =

Q_n \ Q_{n+1}	0	1
0		1
1		d

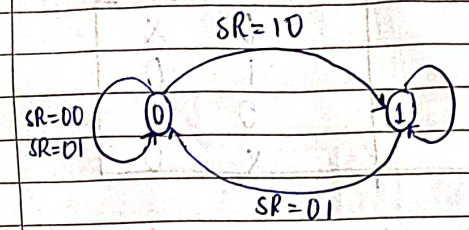
$\Rightarrow \boxed{S = Q_{n+1}}$

R =

Q_n \ Q_{n+1}	0	1
0	d	
1	1	

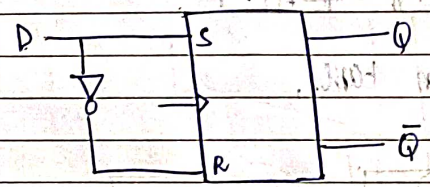
$\Rightarrow \boxed{R = \overline{Q_{n+1}}}$

State Diagram {using excitation Table}

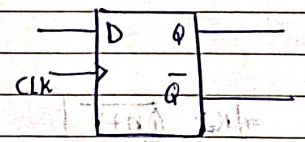


D Flip Flop

↳ Data {to obtain same o/p as I/p data}



⇓ simplified



Data Flip Flop

Truth Table

CLK	D	Q	$\bar{Q}$
0	X	Memory	
1	0	0	1
1	1	1	0

Truth Table :

CLK	D	Qn+1
0	X	Qn
1	0	0
1	1	1

Characteristic Table : CLK=1

Qn	D	Qn+1
0	0	0
0	1	1
1	0	0
1	1	1

Imp. Excitation Table

Qn	Qn+1	D
0	0	0
0	1	1
1	0	0
1	1	1

K-map

characteristic table :

Qn \ D	0	1
0		1
1		1

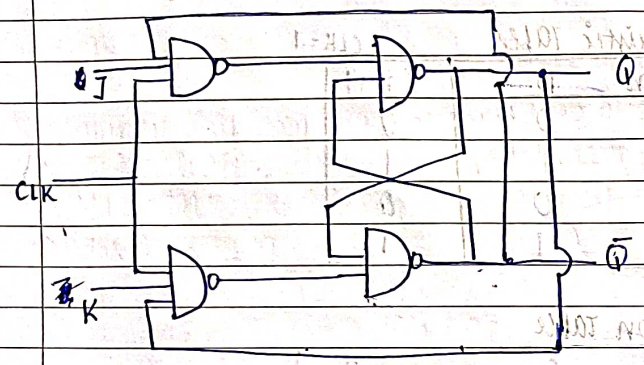
$\boxed{Q_{n+1} = D}$

Excitation table :

Qn \ Qn+1	0	1
0		1
1		1

$\boxed{D = Q_{n+1}}$

JK Flip Flop



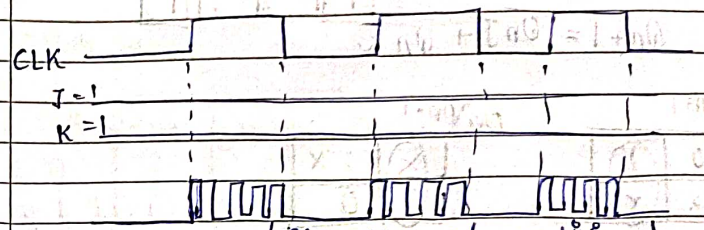
Truth table:

CLK	J	K	Qn+1
0	X	X	Qn
1	0	0	Qn
1	0	1	0
1	1	0	1
1	1	1	$\bar{Q}_n$

* All cases are same as SR flip flop except J = 1 ; K = 1.

For J = 1 , K = 1 ; assuming initially Q = 0, $\bar{Q}$ = 1

Q = 0, 1, 0, 1 ...
$\bar{Q}$ = 1, 0, 1, 0 ... } race-around.



(Race around condition)

Characteristic Table

Qn	J	K	Qn+1
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	0

September 13 2024

Imp :
Excitation Table

Qn	Qn+1	J	K
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

September 13, 2024

KMAP.

characteristic table :-

Q_n \ JK	00	01	11	10
0	0	1	1	1
1	1	5	7	1

$Q_{n+1} = \overline{Q_n} J + Q_n \overline{K}$

Excitation table

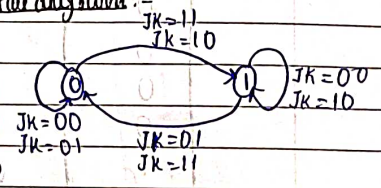
Q_n \ Q_{n+1}	0	1
0	0	1
1	X	X

Q_n \ Q_{n+1}	0	1
0	X	X
1	1	0

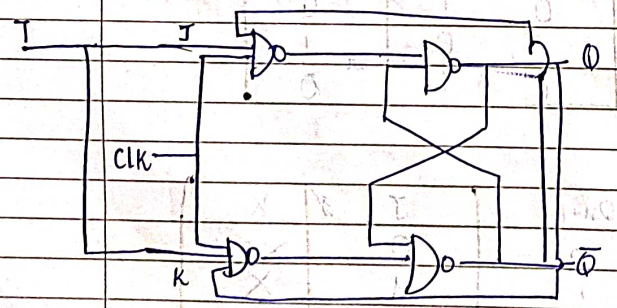
$J = Q_{n+1}$ $K = \overline{Q_{n+1}}$

* How to overcome race around condition :-
→ T/2 > delay
→ Master slave JK flip flop
→ Edge triggering

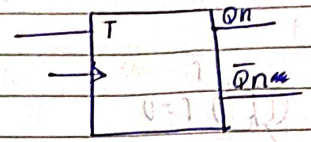
State diagram :-



T (Toggle) Flip Flop



(just like JK but J and K are shorted
In JK the case of toggle is when
J = 1, K = 1
$Q_{n+1} = \overline{Q_n}$
(toggle condition)



Truth table

CLK	T	Q_{n+1}
0	X	Q_n
1	0	Q_n
1	1	$\overline{Q_n}$

characteristic table

Q_n	T	Q_{n+1}
0	0	0
0	1	1
1	0	1
1	1	0

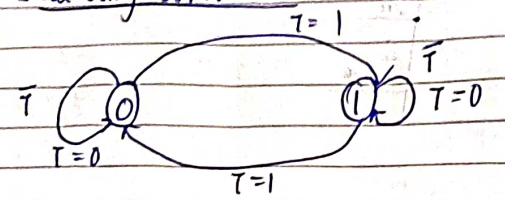
$Q_{n+1} = Q_n \oplus T$

Excitation table

Q_n	Q_{n+1}	T
0	0	0
0	1	1
1	0	1
1	1	0

$T = Q_n \oplus Q_{n+1}$

State diagram :-



Steps of conversion from one to another :

1) Identify the available flip flop and required flip flop.
2) Make the characteristic table of required flip flop.
3) Make the excitation table of available flip flop.
4) Write / solve the boolean expression.
5) Make the circuit.

JK to D

1) Available : JK
   Required : D

2) ~~Excitation~~ characteristic Table of D :-

Qn	D	Qn+1
0	0	0
0	1	1
1	0	0
1	1	1

3) Excitation table of JK :-

Qn	Qn+1	J	K
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

Use this to make characteristic table (include J & K) :-

4)

Qn	D	Qn+1	J	K
0	0	0	0	X
0	1	1	1	X
1	0	0	X	1
1	1	1	X	0

previous table of step 2.

Qn \ D	0	1
0	0	1
1	X	X

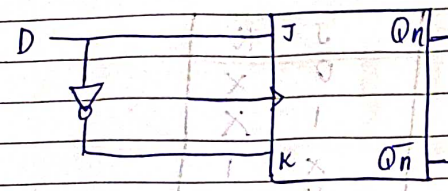
$J = D$

Qn \ D	0	1
0	X	X
1	1	0

$K = \bar{D}$

Boolean expression.

5)



September 17, 2024

JK using SR

1) Available: SR
Required: JK

2) characteristic Table of JK :-

Qn	J	K	Qn+1	S	R
0	0	0	0	0	X
0	0	1	0	0	X
0	1	0	1	1	0
0	1	1	1	1	0
1	0	0	1	X	0
1	0	1	0	0	1
1	1	0	1	X	0
1	1	1	0	0	1

3) Excitation table of SR

Qn	Qn+1	S	R
0	0	0	X
0	1	1	0
1	0	0	1
1	1	X	0

4) Substituted in 2

For S :-

Qn \ JK	00	01	11	10
0	0	1	1 (3)	1 (2)
1	X (4)	5	7	X (6)

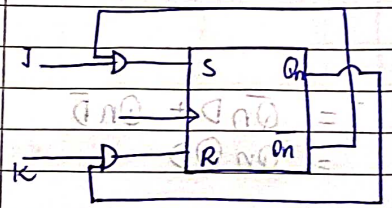
$S = \overline{Q_n} J$

For R :-

Qn \ JK	00	01	11	10
0	X (0)	X (1)	3	2
1	4	1 (5)	1 (7)	6

$R = Q_n K$

5)



T to D Flip Flop

1) Available : T
Required : D

a) Characteristic Table of D.

Qn	D	Qn+1	T
0	0	0	0
0	1	1	1
1	0	0	1
1	1	1	0

3) Excitation Table of T

Qn	Qn+1	T
0	0	0
0	1	1
1	0	1
1	1	0

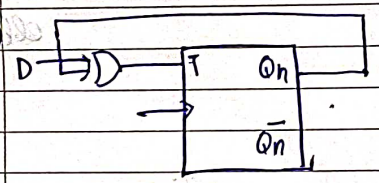
4) (In 2)

Qn \ D	0	1
0		1
1	1	

$T = \bar{Q_n} D + Q_n \bar{D}$

$= Q_n \oplus D$

5)



JK using T

1) Available: T

Required: JKT

2) Characteristic Table of JK

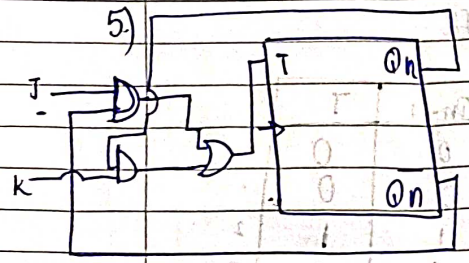
Qn	J	K	Qn+1	T	
0	0	0	0	0	0
0	0	1	0	0	1
0	1	0	1	1	2
0	1	1	1	1	3
1	0	0	1	0	4
1	0	1	0	1	5
1	1	0	1	0	6
1	1	1	0	1	7

3) Excitation Table of T

Qn	Qn+1	T
0	0	0
0	1	1
1	0	1
1	1	0

4) (In 2)

Qn \ JK	00	01	11	10
0	0	1	1 (3)	1 (2)
1	1 (4)	1 (5)	1 (7)	0

$T = \bar{Q_n} J + Q_n K$

5)