$(723.8)_{10} = (\quad)_2$

$$2\,|\,723$$
$$2\,|\,361 - 1$$
$$2\,|\,180 - 1$$
$$2\,|\,90 - 0$$
$$2\,|\,45 - 0$$
$$2\,|\,44 - 1$$
$$2\,|\,22 - 0$$
$$2\,|\,11 - 0$$
$$2\,|\,10 - 1$$
$$2\,|\,5 - 0$$
$$2\,|\,2 - 1$$
$$2\,|\,1 - 0$$
$$2\,|\,0 - 1$$

$0.8 \times 2 = 1.6$

$0.6 \times 2 = 1.2$

$0.2 \times 2 = 0.4$

$0.4 \times 2 = 0.8$

1100

0.8

$(1011000011.1100)_2$

$$8\,|\,723$$
$$8\,|\,90 - 3$$
$$8\,|\,11 - 2$$
$$8\,|\,1 - 3$$
$$0 - 1$$

$0.8 \times 8 = 6.4$

$0.4 \times 8 = 3.2$

$0.2 \times 8 = 1.6$

$0.6 \times 8 = 4.8$

~~1323.~~ $(1323.6314)_8$

~~12230~~

$$5\,|\,723$$
$$5\,|\,144 - 3$$
$$5\,|\,28 - 4$$
$$5\,|\,5 - 3$$
$$5\,|\,1 - 0$$
$$0 - 1$$

$0.8 \times 5 = 4.0$

0.0×5

(10343.4000)

Scanned with CamScanner

Complements

→ Reddish Complement (R's complement)

→ Diminished " ((R-1)'s ")

decimal - 10's Complement
9's Complement

Octal - 8's complement
7's "

Binary - 2's complement
1's "

Hexa - 16's
15's

If n is a positive number to the base r with integer part of n digits than the r's complement of N is defined as $(r^n - N)$.

$(7634)_{10}$ = $r \Rightarrow n$

$n = 4$
$r = 10$

10's Complement = $10^4 - 7634$ = 2366.

~~76.34~~

100-76.34

23.66

$(11011)_2$

$2^5 \rightarrow 32$ $\underline{\underline{32}}$

2	32	
2	16	0
2	8	0
2	4	0
2	2	0
2	1	0
	0	1

$$
\begin{array}{r}
100000 \\
-\ 0011011 \\
\hline
00101 \\
\hline
\end{array}
$$

$(00101)_2$

$$
\begin{array}{r}
3856 \\
1947 \\
\hline
1909 \\
\hline
\end{array}
$$

If N is a +ve no., base r and n integer and m is fractional digits, then $r-1$ complement is $r^{n}-1-m$ or $(r^n - r^m - N)$ or $(r^n - 1 - N)$

Add 1 to $(r-1)$'s complement to get r's complement.

1's complement + 1 = 2's complement

Substraction

Step 1 → Equate the no. of digits of both given no.'s, if they are not equal apply the padding of zero to make them equal.

Step - 2 → Find r's complement of subtraend, minuend ko minus nahi

Step 3 → Add minuend fluid. If sum produces carry then discard carry. and if the sum does not produce any end around carry, take the r's complement to the result of sum and place a -ve sign in front of it.

Q → 765329 - 4250

~~765329~~
~~4250~~ → minuend
76532
04250 → Subtrend.

r's Complement of subtraend.

$10^5 - 04250$
⇒ 95750

95750
76532
~~1~~72282

04250 - 76532

10's comp. of 76532

$10^5 - 76532$

~~37570~~ 23468

9999	
100000	
73532	
26468	

$$\begin{array}{r} 23468 \\ 04250 \\ \hline 27718 \end{array}$$

↓ 10's comp.

$10^5 - 27718$

-72282

$$\begin{array}{r} 100000 \\ 27718 \\ \hline 72282 \end{array}$$

11101 − 1100

2' c → 01100

$$\begin{array}{r} 10011 \\ +1 \\ \hline 10100 \end{array}$$

$$\begin{array}{r} 10100 \\ 11101 \\ \hline +\ \cancel{1}10001 \end{array}$$

Anand Kumar ,

Binary Codes

- Weighted
  - Binary
  - BCD
    - 8421
    - 2421
    - 3321
    - 5421
- Non weighted
  - Excess
  - Gray
- Reflected Codes
- Sequential Codes
- Alpha numeric Codes
  - ASCII
  - EBCDIC
- Error detecting and Collecting
  - Parity
  - Hamming

2 bit 3 bit 4 bit

2 bit

0	0
0	1
1	0
1	1

8 → 3 bit

0	0
0	1
0	0
0	1
1	0
1	1
1	0
1	1

Digital system are require to handle & manipulate data which may consisting of alphabets or numerical. These alphabets, numerals or special characters are to be converted to binary format.

weighted Codes :- In which each position of No. represents a specific wt. for 576

576
↓ ↳ wt. of 10.
wt. of 100.

Non weighted Codes → Codes which are not assigned with any specific wt. to each digit position like Excess 3 or gray Code.

BCD addition :-

Step 1 → Add 2 BCD No.'s using ordinary binary addition.

Step 2 → If 4 bit sum = or less than 9 then no correction is needed the sum is in proper BCD form.

Step 3 → If the 4 bit sum is > 9 or if the carry is generated from the 4 bit sum then the sum is invalid. To correct this invalid sum at 0110 to 4 bit sum and if any carry result from the addition then add 1 to the next higher BCD digit.

Q→ 175
326

0001 0110 0101
0011 0010 0110
+ 0100 1001 1011 → 79 80 add 6.

1
0110

>9 add 6
0100 1010 0001
+ 1 0110

$\underbrace{0101}_{5}$ $\underbrace{0000}_{0}$ $\underbrace{0001}_{1}$

24
18

0010 0100
0001 1000
0011 1100
1 0110
0100 0010
4 2

0 →	0	0	0	0	0
1 →	0	0	0	1	1
3 →	0	0	1	1	2
2 →	0	0	1	0	3
6 →	0	1	1	0	4
7 →	0	1	1	1	5
5 →	0	1	0	1	6
4 →	0	1	0	0	7
12 →	1	1	0	0	8
13 →	1	1	0	1	9
15 →	1	1	1	1	10
14 →	1	1	1	0	11
10 →	1	0	1	0	12
11 →	1	0	1	1	13
9 →	1	0	0	1	14
8 →	1	0	0	0	15

42 → 63

Error detecting and Correcting Codes

→ Hamming Codes → Parity

$N = m + P$

- P → no. of Parity bits
- m → no. of msg. bits.
- N → Total no. of bits in Codes.

7	6	5	4	3	2	1
m_7	m_6	m_5	P_4	m_3	P_2	P_1
111	110	101	100	011	010	001

When data is transmitted through a channel the noise generated by various sources existing in the channel corrupting it. So data can be recovered from the noise with the help of error detecting and correcting codes.

→ Hamming ~~by~~ code is one of the codes used to find error and correcting it.

Parity bits - They are used to identify the error in the ~~da~~ transmitted data word. It is of two type odd parity and even parity.

odd parity - no. of 1's should be odd
even parity - no. of 1's should be even

10110 → odd parity
10111 → even

$$\boxed{m \leq 2^p - p - 1}$$

$p = 2$

$4 \leq 2^2 - 2 - 1$

$4 \leq 1$.

$4 \leq 2^3 - 3 - 1$

$4 = 4$.

$m \leq 2^5 - 5 - 1$

26

$5 \leq 2^p - p - 1$

$6 \leq 2^p - p$

1001

m_7	m_6	m_5	P_4	m_3	P_2	P_1
1	0	0	1	1	0	0

$P_1 \rightarrow 1, 3, 5, 7$ (1 at 1st position)
$P_2 \rightarrow 2, 3, 6, 7$ (1 at 2nd position)
$P_4 = 4, 5, 6, 7.$ (1 at 3rd position.

$P_1 \rightarrow 101\underline{0}$

$P_2 \rightarrow 100\underline{0}$

$P_3 \rightarrow 100\underline{1}$

$\boxed{1001100}$

Q → Assume the code word 1001100 is transmitted and 1000100 is received prove that the errors occoured and find out the correction bit if even parity is used.

m_7	m_6	m_5	P_4	m_3	P_2	P_1
111	110	101	100	011	010	001
1	0	0	0	1	0	0

$P_1 \rightarrow 1, 3, 5, 7$ ~~0100~~ 1010 — 0

$P_2 \rightarrow 2, 3, 6, 7$ 1010 — 0

$P_3 \rightarrow 4, 5, 6, 7$ 1000 × 1

$\underline{100} \rightarrow \underline{4}$ bit is wrong m = 1101 / odd parity

m7	m6	m5	P4	m3	P2	P1
1	1	0	1	1	0	1
0111	110	101	100	011	010	001
1	0	0	1	1	0	1

$P_1 \rightarrow 1, 3, 5, 7$ $P_2 \rightarrow 2, 3, 6, 7$

$P_4 \rightarrow 4, 5, 6, 7.$

1001101

$P_1 \rightarrow 1, 3, 5, 7 \rightarrow 1101$ 0

$P_2 \rightarrow 2, 3, 6, 7 \rightarrow 0101$ 1

$P_4 \rightarrow 4, 5, 6, 7 \rightarrow 1001$ 1

110 → 6 bit is wrong

$m_i = 2^P - P - 1$

9 =

Rx → 10101

m9	P8	m7	m6	m5	P4	m3	P2	P1
1	1	0	1	0	1	1	0	0
1001	1000	0111	110	0101	100	011	010	001
0	0	0	0	0	1	1	0	0

0

1001
+1
1010
0
0001

$P_1 \rightarrow 1, 3, 5, 7, 9$ 10010

$P_2 \rightarrow 2, 3, 6, 7$ 10.0 0 1

$P_4 \rightarrow 4, 5, 6, 7$ 0 1 0 1

$P_8 \rightarrow 8, 9$ 11

1110

$\boxed{110101100}$

Bunk Pager - The Social Notebook

00 101100

$P_1 \rightarrow 0, 1, 0, 0, 1 \rightarrow 0$

$P_2 \rightarrow 0, \cancel{0}, \phi, 0 \rightarrow 0$

$P_4 \rightarrow 1, 0, 1, 0 \rightarrow 0$

$P_8 \rightarrow 0, 1 \rightarrow 1$

Boolean Algebra :-

$A+0 = A$

$A+1 = 1$

$A \cdot A = A$

$A \cdot \bar{A} = 0$

$A + \bar{A} = 1$

$A(B+C) = AB + AC$

$A + BC = (A+B)(A+C)$

$\bar{\bar{A}} = A$

1) Number system and their conversion.
2) Basic Arithmetic of no. systems
3) Binary Codes (BCD, Gray code, Excess 3, 8421) (Hamming code)
4) BCD and excess 3 arithmatics
5) Basic gates and universal gates.
6) Implimentation of Boolean func. universal gates
7) K map upto 4 variables

Excess 3 → Add 3 to BCD code.

Solving of Boolean expression using boolean Algebra

Anand Kumar

Sum of Products :- (SoP)
Product of Sums :- (PoS)

A	B	C	m	M
0	0	0	$\bar{A}\bar{B}\bar{C}$	$A+B+C$
0	0	1	$\bar{A}\bar{B}C$	$A+B+\bar{C}$
0	1	0	$\bar{A}B\bar{C}$	$A+\bar{B}+C$
0	1	1	$\bar{A}BC$	$A+\bar{B}+\bar{C}$
1	0	0	$A\bar{B}\bar{C}$	$\bar{A}+B+C$
1	0	1	$A\bar{B}C$	$\bar{A}+B+\bar{C}$
1	1	0	$AB\bar{C}$	$\bar{A}+\bar{B}+C$
1	1	1	ABC	$\bar{A}+\bar{B}+\bar{C}$

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

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

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

$$\cancel{y = ABC + \bar{A}BC + AB\bar{C}}$$

$$y = \underset{111}{ABC} + \underset{110}{AB\bar{C}} + \underset{010}{\bar{A}B\bar{C}} + \underset{100}{A\bar{B}\bar{C}}$$

$$f(A,B,C) = \Sigma m(7, 6, 2, 4)$$

Σm	A	B	C	Y
0	0	0	0	1
1	0	0	1	1
2	0	1	0	0
3	0	1	1	1
4	1	0	0	1
5	1	0	1	0
6	1	1	0	1
7	1	1	1	1

$$\Sigma m\,(0,1,3,4,6,7)$$

$$\pi(m(2,5))$$

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

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

$$\pi = AB + AC + \bar{A}B + \bar{B}C + \bar{A}C + AC$$

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

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

$\cancel{ABC} + \cancel{AB\bar{C}} + \cancel{AB\bar{C}} + \cancel{A\bar{B}\bar{C}} + \cancel{\bar{A}\bar{B}C} + \cancel{\bar{A}\bar{B}\bar{C}} + \cancel{A\bar{B}\bar{C}} + \cancel{\bar{A}\bar{B}\bar{C}} + \bar{A}BC + \cancel{\bar{A}\bar{B}C} + \cancel{ABC} + \bar{A}BC$

$ABC + AB\bar{C} + A\bar{B}\bar{C} + \bar{A}\bar{B}C + \bar{A}\bar{B}\bar{C} + \bar{A}BC$

111 110 100 001 000 011

7 6 4 1 0 3

$\Sigma m(0, 1, 3, 4, 6, 7)$

This implies SOP & POS are complementary.

Q → $\overline{\bar{A}B + \bar{A} + AB}$

$\overline{(AB)}\, A.\overline{(\bar{A}B)}$

0

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

~~$AB + \bar{A}C + A\bar{B}C . AB + A\bar{B}C . C$~~

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

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

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

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

~~AB~~ 1

AB \ CD	$C+D$ 00	$C+\bar D$ 01	$\bar C+\bar D$ 11	$\bar C+D$ 10
$A+B$ 00	X 0	X 1	X 3	X 2
$A+\bar B$ 01	0 4	0 5	0 7	0 6
$\bar A+\bar B$ 11	0 12	13	15	X 14
$\bar A+B$ 10	0 8	X 9	X 11	10

$A\,(C+D).$

AB \ CD	$\bar C\bar D$	$\bar C D$	CD	$C\bar D$
$\bar A\bar B$	X 0	1 1	1 3	X 2
$\bar A B$	4	X 5	1 7	6
AB	12	13	1 15	14
$A\bar B$	8	9	1 11	10

$\bar A\bar B + CD$

$f(A,B,C,D,E) = \Sigma m(0,1,2,3,8,12,15,16,17,18,19,23,\cancel{25}\,28,31)$

$\bar A$

BC \ DE	$\bar D\bar E$ 00	$\bar D E$ 01	DE 11	$D\bar E$ 10
$\bar B\bar C$ 00	1 0	1 1	1 3	1 2
$\bar B C$ 01	4	5	7	6
BC 11	1 12	13	1 15	14
$B\bar C$ 10	1 8	9	11	10

A

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

1 2 3 5

$f \to \bar B\bar C + \bar A B\bar D\bar E + BCDE + BC\bar D\bar E + A\bar B D\bar E$

Q → ~~Σm = (0, 1, 2, 3, 8, 12, 15, 16, 17, 18, 19, 22, 28)~~,

$\Sigma m = (0, 4, 8, 12, 16, 18, 20, 22) + d(24, 26, 28, 30, 31)$.

$\bar{A}$

BC \ DE	00 ($\bar{D}\bar{E}$)	01 ($\bar{D}E$)	11 (DE)	10 ($D\bar{E}$)
$\bar{B}\bar{C}$ 00	1 (0)	1	3	2
$\bar{B}C$ 01	1 (4)	5	7	6
BC 11	1 (12)	13	15	14
$B\bar{C}$ 10	1 (8)	9	11	10

A

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

$\bar{D}\bar{E}$ ~~+ $AD\bar{E}$~~

$\bar{A}(\bar{D} + AD)$

$\bar{E}((A+\bar{D}))$

$\boxed{\bar{E}\bar{D} + A\bar{E}}$

Q1 using Mc Cluskey method (Tabulation method)

Step 1 → list all the minterms in their equivalent binary form.

Step 2 → arrange the minterm's based on the no. of 1's.

Step 3 → Compare each binary no. from 1 group to another and if they differs only by 1 bit position put — mark and copy the remaining term. and place ✓ after each comparison

Step-4 → Apply the same process described in step 3 for the resultant column and continue the cycles until a single pass through cycle yields no further elimination.

Step - 5 - list all the unticked prime implicants and form a prime implication chart. Each prime implicant is represented in a row and each mintum in a column. The cross mark are placed in each row to show the comparison of mintum that makes a prime implicant.

→ Search for single cross mark column and select the prime implicant corresponding to that.

→ Search for multiple dot columns one by one & if the corresponding mintum is already included in the final expression, ignore the mintum and go to next multidot column. otherwise include the corresponding prime implicant in the final expression.

$F(A,B,C,D) = \Sigma m(0,2,3,6,7,8,10,12,13)$

0000,	0010,	0011,	0110,	0111,	1000,	1010,	1100,	1101
m_0	m_2	m_3	m_6	m_7	m_8	m_{10}	m_{12}	m_{13}

✓ m_0	0000	00-0 (0,2)	discard 2 bit change
✓ m_2	0010	-000 (0,8)	
✓ m_8	1000	001- (2,3)	
✓ m_3	0011	0-10 (2,6)	
✓ m_6	0110	-010 (2,10)	
✓ m_{10}	1010	10-0 (8,10)	
✓ m_{12}	1100	1-00 (8,12)	
✓ m_7	0111		
✓ m_{13}	1101		

0 – 1 1 (3, 7)
0 1 1 – (6, 7)
1 1 0 – (12, 13)

	A B C D		
0 0 – 0 (0,2)	– 0 – 0 (0,2,8,10)		$\bar{B}\bar{D}$ (0,2,8,10)
– 0 0 0 (0,8)	– 0 – 0 (0,8,2,10)		
0 0 1 – (2,3)	0 – 1 – (2,3,6,7)		$\bar{A}C$ (2,3,6,7)
0 – 1 0 (2,6)	0 – 1 – (2,6,3,7)		
– 0 1 0 (2,10)			
1 0 – 0 (8,10)			
* 1 – 0 0 (8,12)			
0 – 1 1 (3,7)			
0 1 1 – (6,7)			
* 1 1 0 – (12,13)			

Prime Implicants	0	2	3	6	7	8	10	12	13
$\bar{B}\bar{D}$ (0,2,8,10)	X	X				X	X		
$\bar{A}C$ (2,3,6,7)		X	X	X	X				
$A\bar{C}\bar{D}$ (8,12)						X		X	
$AB\bar{C}$ (12,13)								X	X

$AB\bar{C} + \bar{B}\bar{D} + \bar{A}C + \bar{B}\bar{D}$

$\boxed{AB\bar{C} + \bar{B}\bar{D} + \bar{A}C}$ → Saare literals aagaye.

AB \ CD	$\bar C\bar D$ 00	$\bar C D$ 01	CD 11	$C\bar D$ 10
$\bar A\bar B$ 00	1 (0)	1 (1)	1 (3)	1 (2)
$\bar A B$ 01	(4)	(5)	1 (7)	1 (6)
AB 11	1 (12)	1 (13)	(15)	(14)
$A\bar B$ 10	1 (8)	(9)	(11)	1 (10)

$\cancel{\bar A\bar B + \bar A CD + A\bar B\bar D + AB\bar C}$

$\cancel{\bar A B}$ $\bar A C + AB\bar C + \bar B\bar D$.

Q → $F = \Sigma m(0,1,9,15,24,29,30) + d(8,11,31)$

Q → $\Sigma m(1,3,7,11,15) + d(0,2,5)$

~~m_1 0001~~	m_0 0000	000- (0,1)	ABCD		
~~m_3 0011~~	m_1 0001	00-0 (0,2)	00--	(0,1,2,3)	$\bar A\bar B$
~~m_7 0111~~	m_2 0010	00-1 (1,3)	00--	(0,2,1,3)	
~~m_{11} 1011~~	m_3 0011	0-01 (1,5)	0--1	(1,3,5,7)	$\bar A D$
~~m_{15} 1111~~	m_5 0101	001- (2,3)	0--1	(1,5,3,7)	
	m_7 0111	0-11 (3,7)	--11	(3,7,11,15)	CD
	m_{11} 1011	-011 (3,11)			
	m_{15} 1111	01-1 (5,7)			
		-111 (7,15)			
		1-11 (11,15)			

Prime Implicant	1	3	7	11	15	0	2	5
$\bar A\bar B$	X	X				X	X	
$\bar A D$	X	X	X					X
CD		X	X	X	X			

→ essential prime implicants

$\cancel{\bar A\bar B}$ $\bar A D + CD$

All no's must be included if not ~~repla~~ ~~ceh~~ the don't care then.

$\Sigma(0,1,9,15,24,29,30) + d(8,11,31)$

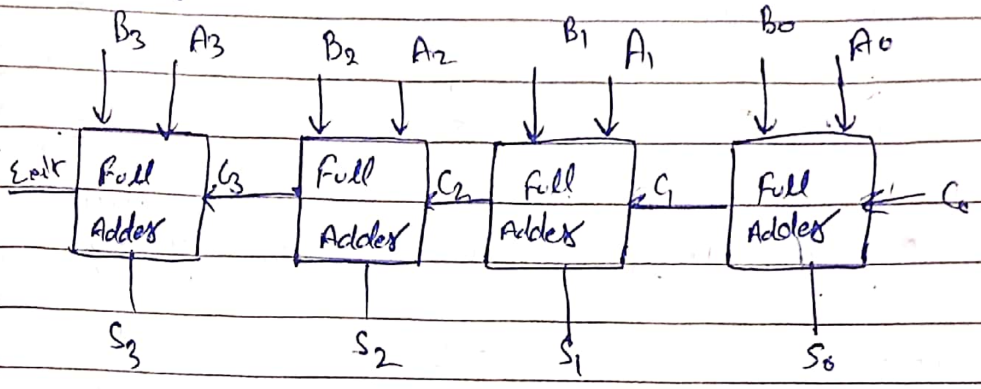
00000, 00001, 01001, 01111, 11000, 11101, 11110, 01000, 01011, 11111
0, 1, 9, 15, 24, 29, 30, 8, 11, 31

m_0 00000	0000– (0,1)	0 – 00 – (0,8,24)
m_1 00001	0–000 (0,8)	0 – 00 – (0,8,1,9)
m_8 01000	0–001 (1,9)	
m_9 01001	0100– (8,9)	
m_{24} 11000	–1000 (8,24)	
m_{11} 01011	010–1 (9,11)	
	01–11 (11,15)	
m_{30} 11110	1111– (30,31)	
m_{15} 01111	–1111 (15,31)	
m_{31} 11111	111–1 (29,31)	

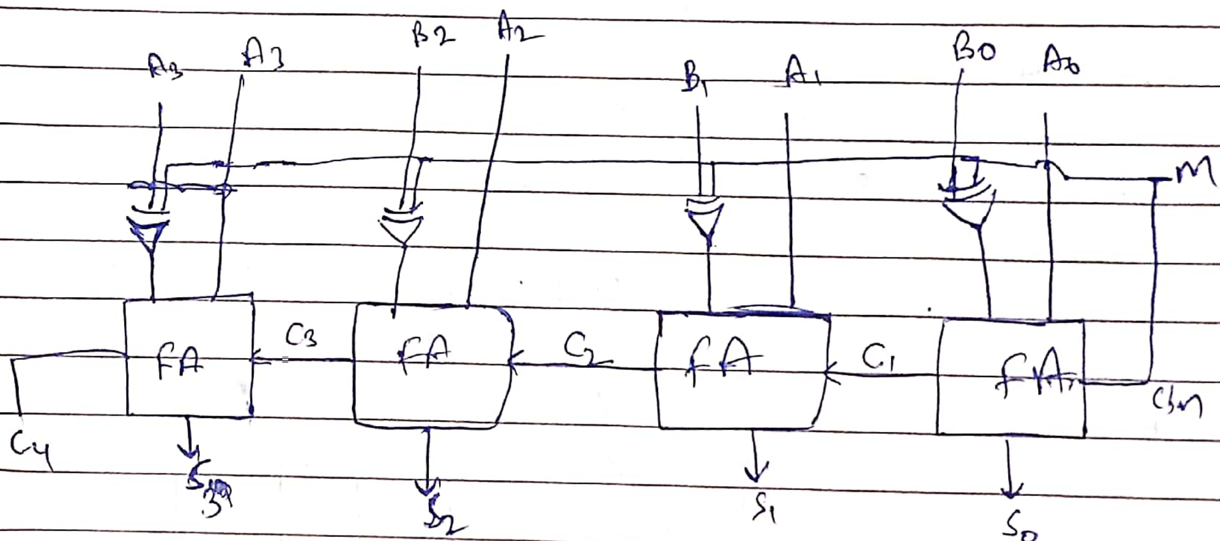
prime implicant	0	1	9	15	24	29	30	8	11	31
ABC	X	X			X			X		
$B\bar{C}\bar{D}\bar{E}$					X			X		
$A\bar{B}CE$			X						X	
ABDE				X					X	
ABCD							X			X
BCDE				X						X
ABCE						X				X

ABC + $A\bar{B}CE$ + ABCD + BCDE + ABDE

Binary Parallel Adder :-



A3	A2	A1	Ao
B3	B2	B1	Bo
S3	S2	S1	So



If m = 0 circuit is adder.
If m = 1 circuit is substractor.

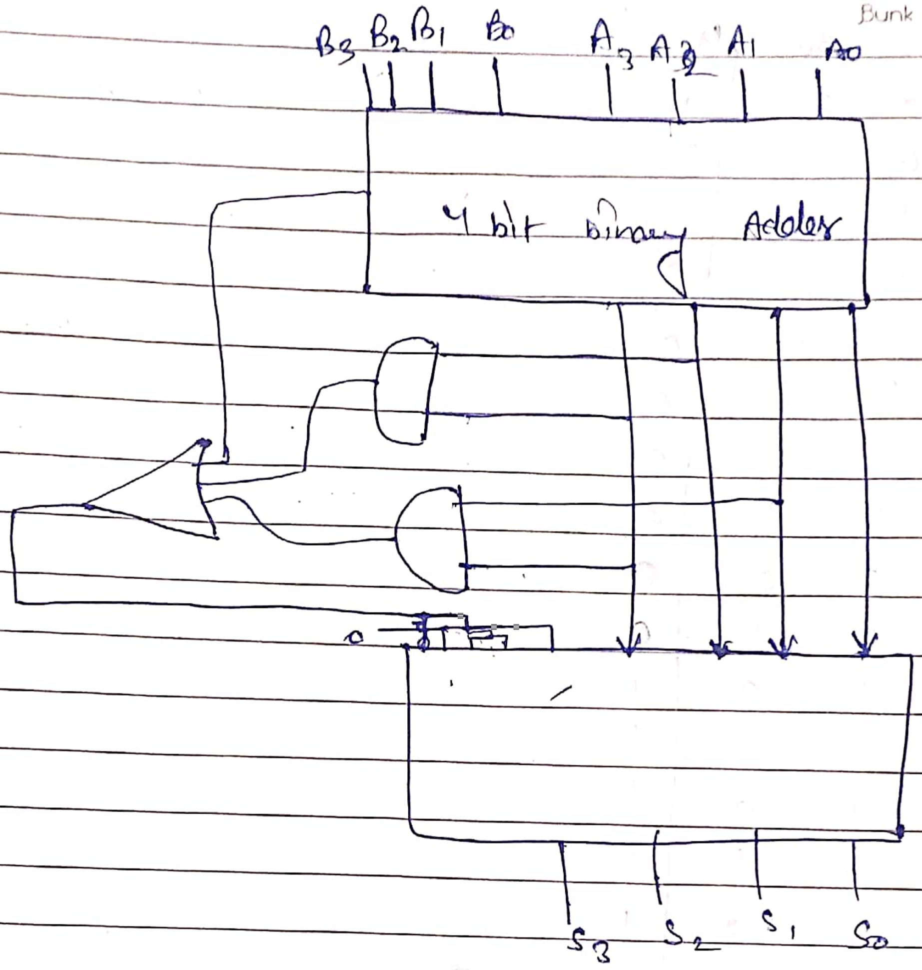
BCD Adder :-

	S_3	S_2	S_1	S_0	y
0	0	0	0	0	0
1	0	0	0	1	0
2	0	0	1	0	0
3	0	0	1	1	0
4	0	1	0	0	0
5	0	1	0	1	0
6	0	1	1	0	0
7	0	1	1	1	0
8	1	0	0	0	0
9	1	0	0	1	0
10	1	0	1	0	1
11	1	0	1	1	1
12	1	1	0	0	1
13	1	1	0	1	1
14	1	1	1	0	1
15	1	1	1	1	1

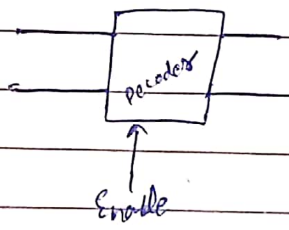
S_1S_0 $S_1\bar{S_0}$

S_3S_2 \ S_1S_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	8	9	1 (11)	1 (10)

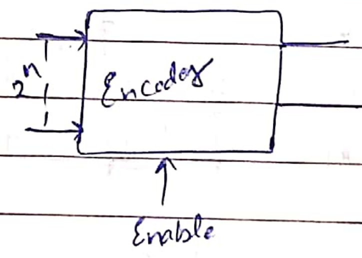
$S_3S_2 + S_3S_1$



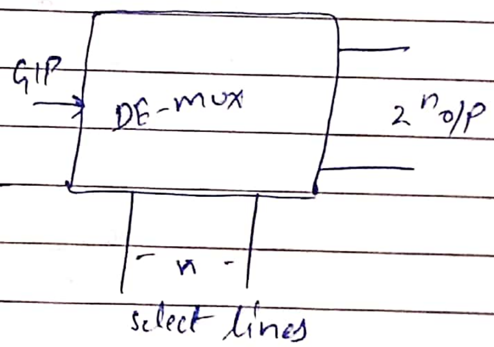
B3 B2 B1 B0
A3 A2 A1 A0
4 bit binary Adder
0
S3 S2 S1 S0



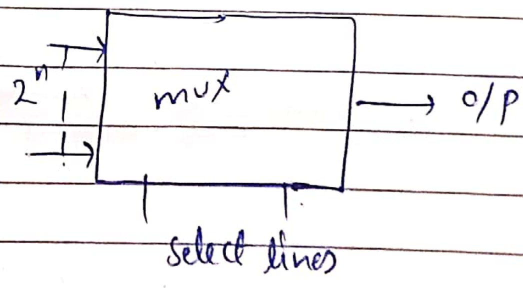
Decoder
Enable



2^n
Encoder
Enable

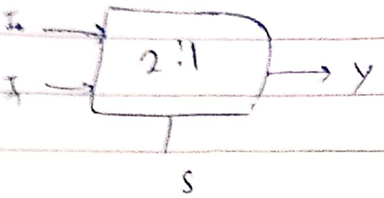


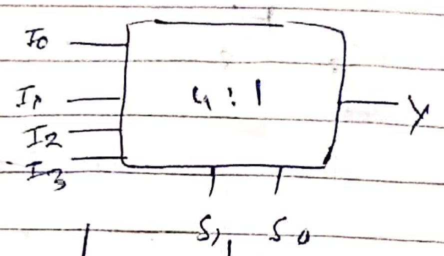
G/P
DE-mux
2^n o/p
n
select lines



2^n
mux
o/p
select lines

MULTIPLEXER





$S = 0 \quad Y = I_0$

$S = 1 \quad Y = I_1$

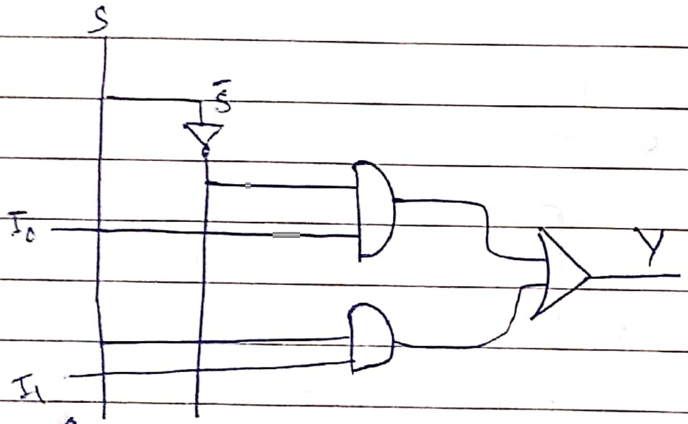
$Y = \bar{S} I_0 + S I_1$

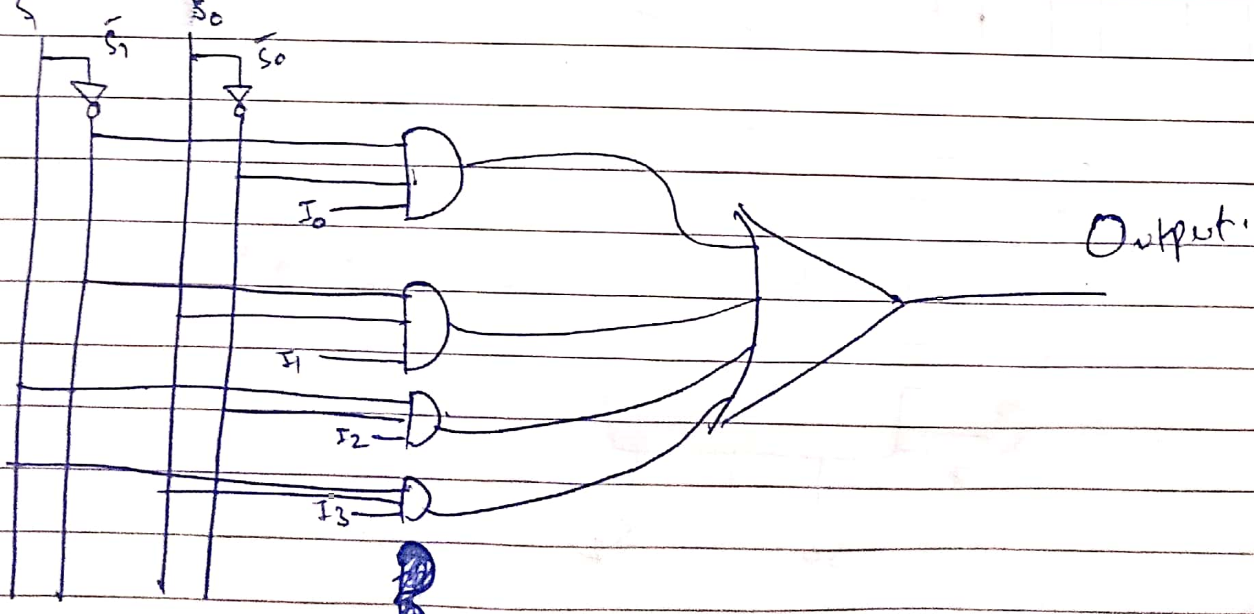
S_1	S_0	Y
0	0	I_0
0	1	I_1
1	0	I_2
1	1	I_3

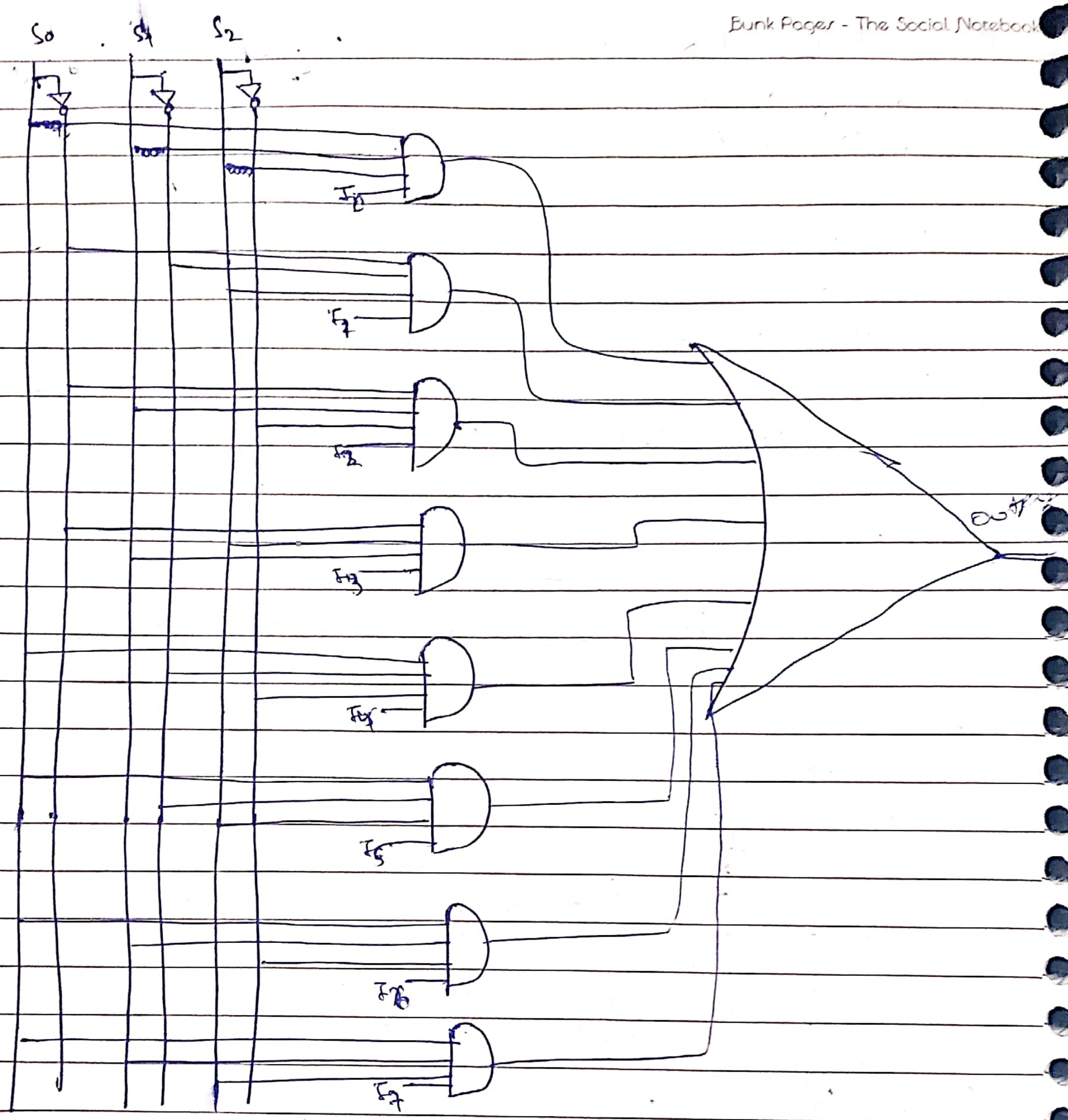
$\cancel{S = I_0 - \bar{S} I_0 + S I_1}$

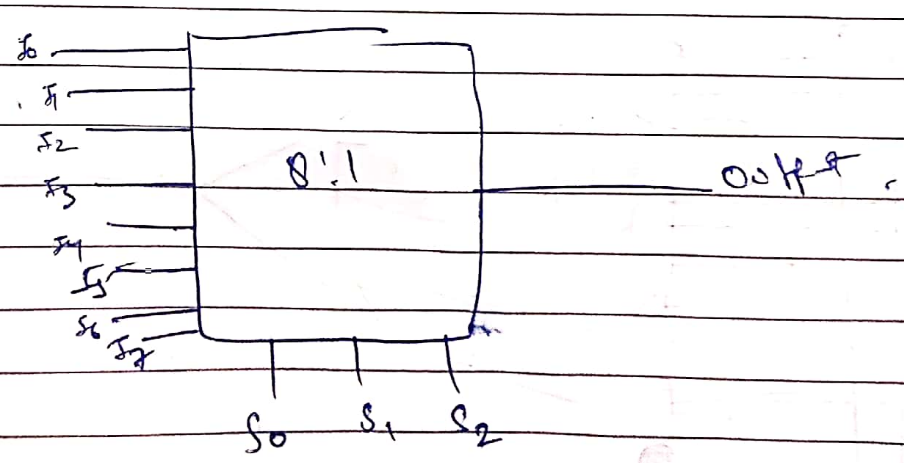
$S = \bar{S_1}\bar{S_0} I_0 + \bar{S_1} S_0 I_1 + S_1 \bar{S_0} I_2 + S_1 S_0 I_3$

$S = I_0 +$





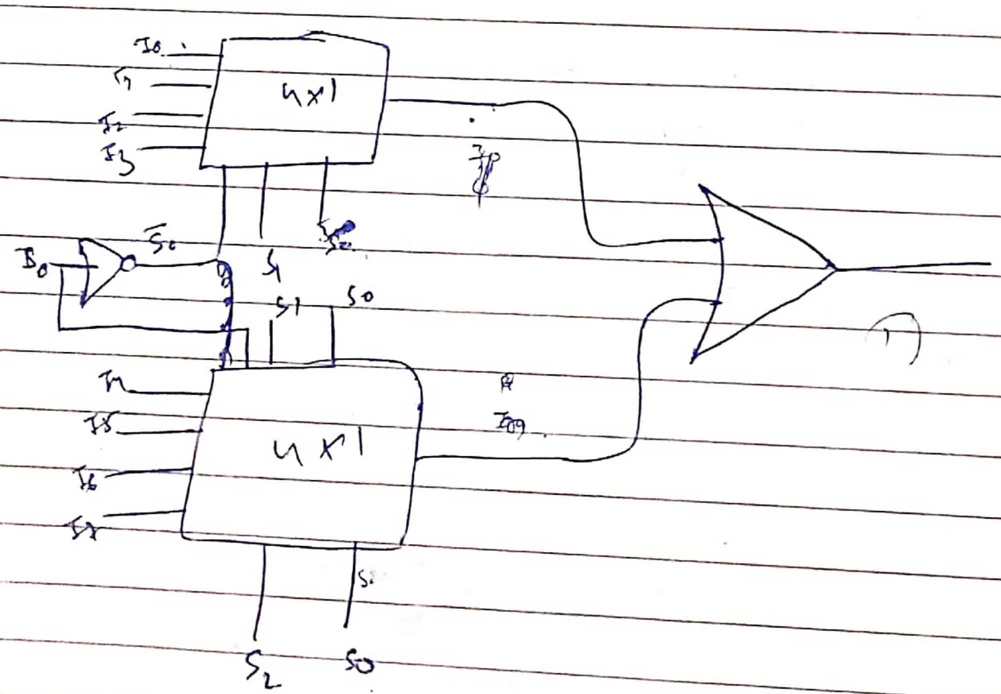

S0
S1
S2
I0
I1
I2
I3
I4
I5
I6
I7
Output


I0
I1
I2
I3
I4
I5
I6
I7
8:1
Output
S0
S1
S2

S_0	S_1	S_2	output
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

$Y = \bar{S_0}\bar{S_1}\bar{S_2} I_0 + \bar{S_0}\bar{S_1} S_2 I_1 - - - -$

Q → 8:1 using 2 × (4:1).



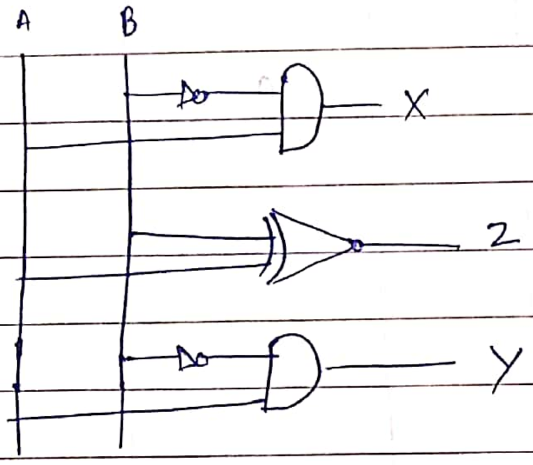
1 bit Comparator :-

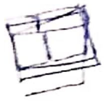
A	B	A < B	A > B	A = B
0	0	0	0	1
0	1	1	0	0
1	0	0	1	0
1	1	0	0	1

$X = \bar{A}B.$

$Y = A\bar{B}$

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



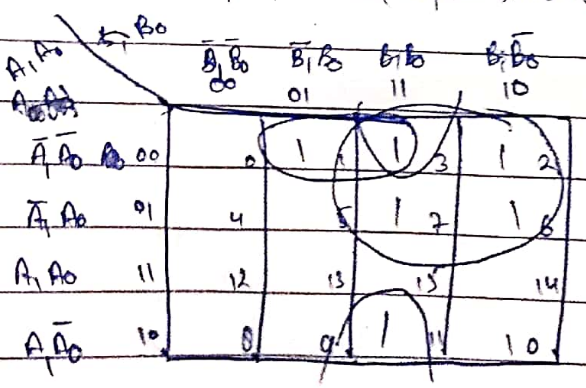


	A_1	A_0	B_1	B_0	$X = A<B$	$Y : A>B$	$Z = A=B$
0	0	0	0	0	0	0	1
1	0	0	0	1	1	0	0
2	0	0	1	0	1	0	0
3	0	0	1	1	1	0	0
4	0	1	0	0	0	1	0
5	0	1	0	1	0	0	1
6	0	1	1	0	1	0	0
7	0	1	1	1	1	0	0
8	1	0	0	0	0	1	0
9	1	0	0	1	0	1	0
10	1	0	1	0	0	0	1
11	1	0	1	1	1	0	0
12	1	1	0	0	0	1	0
13	1	1	0	1	0	1	0
14	1	1	1	0	0	1	0
15	1	1	1	1	0	0	1

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

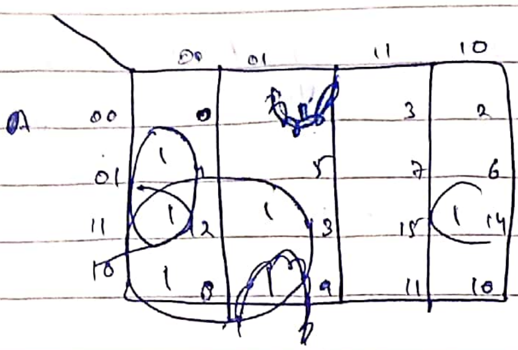
$$X = \bar{A}_1 B_1 + (A_1 \odot B_1) \cdot \bar{A}_0 B_0$$

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



$$\bar{A}_1 B_1 + \bar{A}_1 \bar{A}_0 B_0 + \bar{A}_0 B_1 B_0$$

$\overline{A}B + AB$



$\overline{A_0}\,\overline{B_1}\,B_0 + A_1A_0\overline{B_0}(B_1+\overline{B_1})$

$A_1\overline{B_1} + \cancel{\overline{A_0}\,\overline{B_1}\,\overline{B_0}} + \overline{A_0}\,\overline{B_1}\,B_0$

$+ A_1A_0B_0$

$\cancel{A_0\overline{B_1} + A_1A_0\overline{B_0}}$

$\cancel{A_0(\overline{B_1} + A_1\overline{B_0})}$

$\cancel{A_0((\overline{B_1}+A_1)(\overline{B_1}+\overline{B_0}))}$

$\cancel{A_0A_1 + A_0A_1}$

$\cancel{A_0(\overline{B_1}+A_1)(\overline{B_1}+\overline{B_0})}$

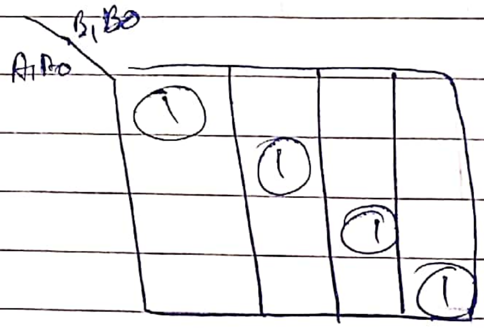
$\cancel{(\overline{B_1}+A_1)A_0\overline{B_1}} +$

$\cancel{A_0\overline{B_0}}$

$\cancel{A_0\overline{B_1} + A_1A_0\overline{B_1}} +$

$\cancel{A_0\overline{B_0}}$

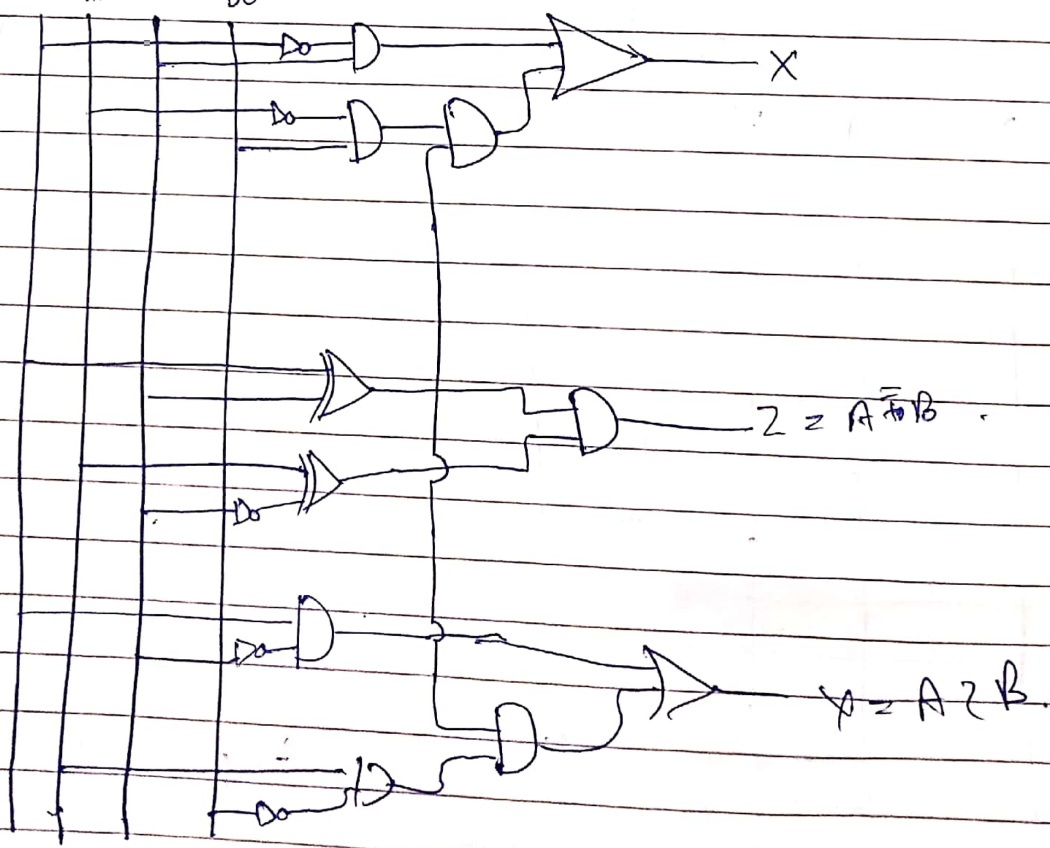
$Y = A_1\overline{B_1} + (A_1 \odot B_1)A_0\overline{B_0}$



$A_1\overline{B_1} +$

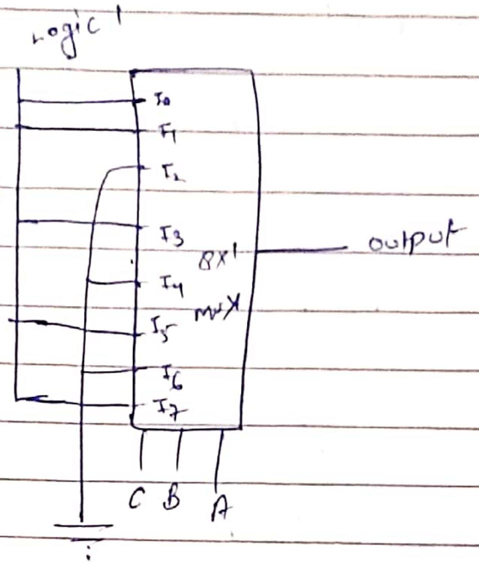
$Z = (A_1 \odot B_1) \cdot (A_0 \odot B_0)$

$A_1 \quad A_0 \quad B_1 \quad B_0$



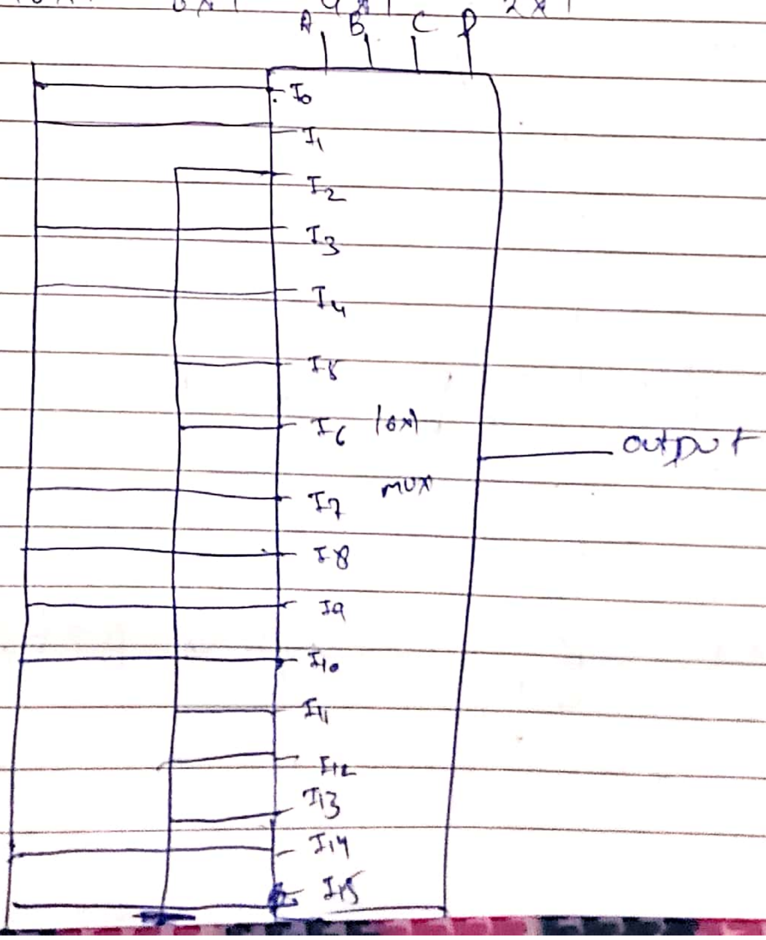
Limits Implementation of boolean function using multiplexer.

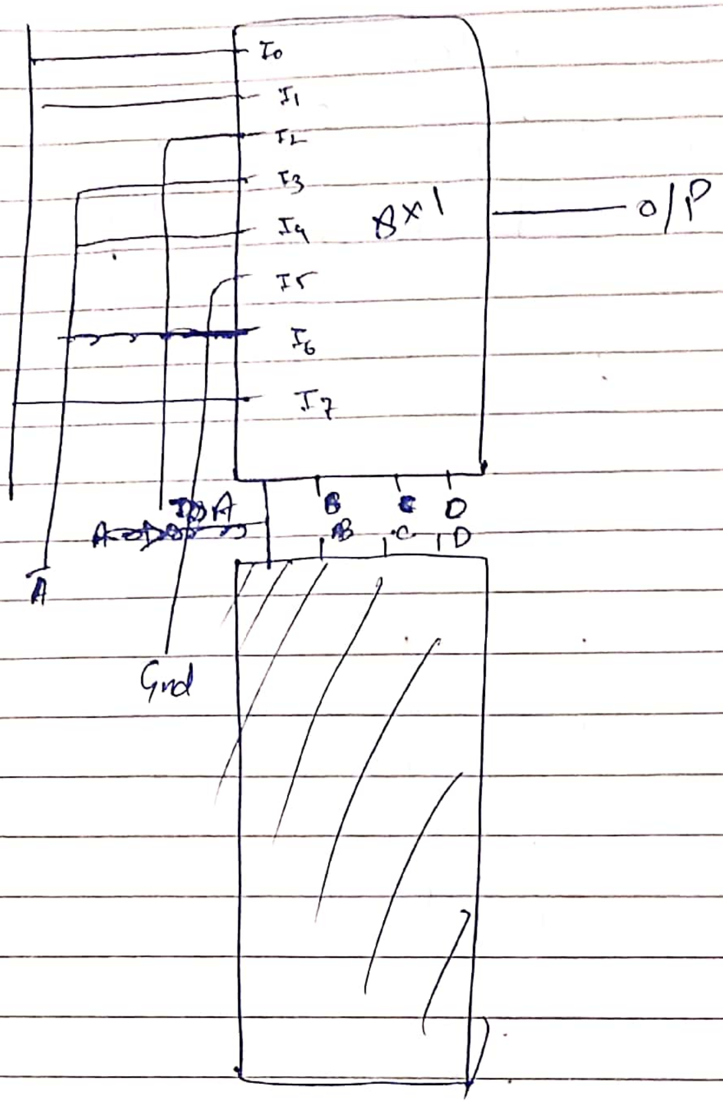
$$F(A,B,C) = \Sigma m(0,1,3,5,7)$$



Q → $F(A,B,C,D) = \Sigma m(0,1,3,4,7,8,9,10,14,15)$

16x1 8x1 4x1 2x1





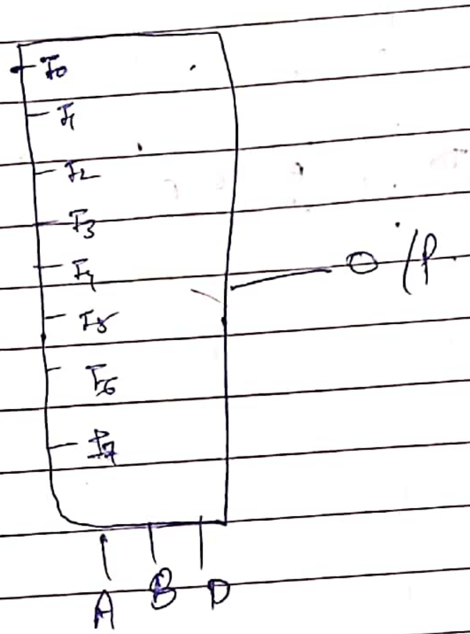
		000	001	010	011	100	101	110	111
0	$\bar{A}$	0	1	2	3	4	5	6	7
1	A	8	9	10	11	12	13	14	15
		1	1	A	$\bar{A}$	$\bar{A}$	0	A	1

Q4 Ckt. using B as variable.

$\bar{B}$	0	1	2	3	8	9	10	11
B	4	5	6	7	12	13	14	15
	1	$\bar{B}$	0	1	$\bar{B}$	$\bar{B}$	1	B

Q → Take C as a variable.

$\bar{C}$	0	1	4	5	8	9	12	13
C	2	3	6	7	10	11	14	15
	$\bar{C}$	1	$\bar{C}$	C	1	$\bar{C}$	C	C



Q → 4x1.

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

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

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

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

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

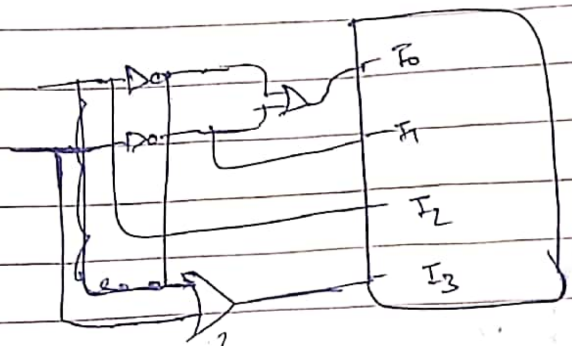
1

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

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

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

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



		$\bar{D}$	D
$\bar{A}\bar{B}\bar{C}$	000	0	2
$\bar{A}\bar{B}C$	001	1	3
$\bar{A}B\bar{C}$	010	4	6
$\bar{A}BC$	011	5	7
$A\bar{B}\bar{C}$	100	8	10
$A\bar{B}C$	101	9	11
$AB\bar{C}$	110	12	14
ABC	111	13	15

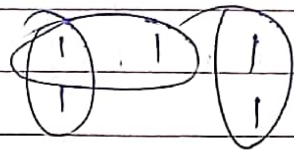
000	0	1	0
001	2	3	1
010	4	5	2
011	6	7	3
100	8	9	4
101	10	11	5
110	12	13	6
111	14	15	7

~~ABC~~ + A

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

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

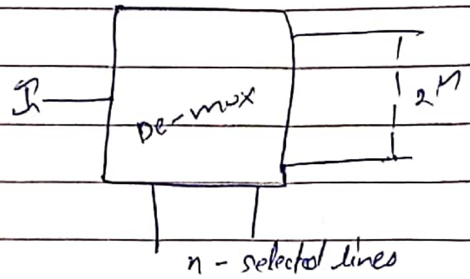
A \ BC	$\bar{B}\bar{C}$ 00	01	BC 11	$B\bar{C}$ 10
$\bar{A}$ 0	1	1	1	2
A 1	1	5	1	6



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

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

De-MUX



1x4
1x8
1x16

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

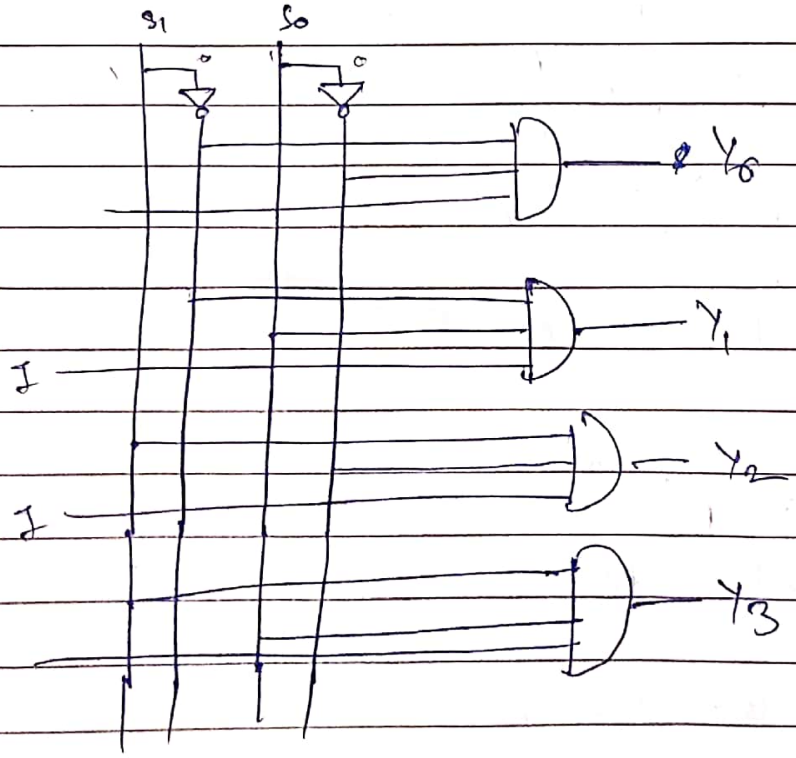
$\cancel{\bar{S}_1 \bar{S}_0 Y_0 + \bar{S}_1 S_0 Y_1 + S_1 \bar{S}_0 Y_2}$

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

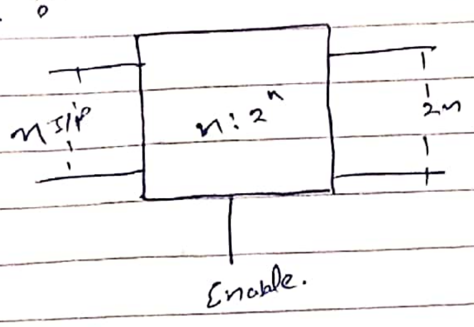
$Y_1 = \bar{S}_1 S_0 I$

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

$Y_3 = S_1 S_0 I$



Decoder :-



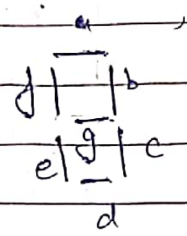
E	A	B	Y_3	Y_2	Y_1	Y_0
0	X	X	X	X	X	X
1	0	0	0	0	0	1
1	0	1	0	0	1	0
1	1	0	0	1	0	0
1	1	1	1	0	0	0

3:8

E	A	B	C	Y_7	Y_6	Y_5	Y_4	Y_3	Y_2	Y_1	Y_0
0	X	X	X	X	X	X	X	X	X	X	X
	0	0	0	0	0	0	0	0	0	0	1
	0	0	1	0	0	0	0	0	0	1	0
	0	1	0	0	0	0	0	0	1	0	0
	0	1	1	0	0	0	0	1	0	0	0
	1	0	0	0	0	0	1	0	0	0	0
	1	0	1	0	0	1	0	0	0	0	0
	1	1	0	0	1	0	0	0	0	0	0
	1	1	1	1	0	0	0	0	0	0	0

BCD to 7 segment Converter :-

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



	A	B	C	D	a	b	c	d	e	f	g
0	0	0	0	0	1	1	1	1	1	1	0
1	0	0	0	1	0	1	1	0	0	0	0
2	0	0	1	0	1	1	0	1	1	0	1
3	0	0	1	1	1	1	1	1	0	0	1
4	0	1	0	0	0	1	1	0	0	1	1
5	0	1	0	1	1	0	1	1	0	1	1
6	0	1	1	0	1	0	1	1	1	1	1
7	0	1	1	1	1	1	1	0	0	0	0
8	1	0	0	0	1	1	1	1	1	1	1
9	1	0	0	1	1	1	1	1	0	1	1

0 → abcdef

1 → bc

$a = \Sigma m(0, 2, 3, 5, 6, 7, 8, 9) + d(10, 11, 12, 13, 14, 15).$

AB \ CD		$\bar C \bar D$ 00	$\bar C D$ 01	CD 11	$C\bar D$ 10
$\bar A \bar B$	00	1 (0)	(1)	1 (3)	1 (2)
$\bar A B$	01	(4)	1 (5)	1 (7)	1 (6)
AB	11	X (12)	X (13)	X (15)	X (14)
$A\bar B$	10	1 (8)	1 (9)	X (11)	X (10)

$\bar A C + A\bar C + \bar A \bar B \bar D$

$+ B \bar C D$

$A + C + BD + \bar B \bar D$

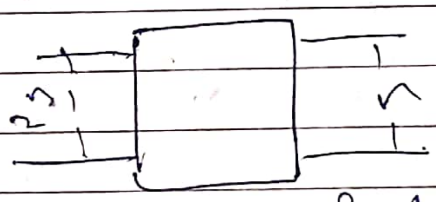
	$\bar{C}\bar{D}$	$\bar{C}D$	CD	$C\bar{D}$
$\bar{A}\bar{B}$	1	1	1	1
$\bar{A}B$	1	.	1	8
AB	X	X	X	X
$A\bar{B}$	1	1	X	X

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

Encoder :-

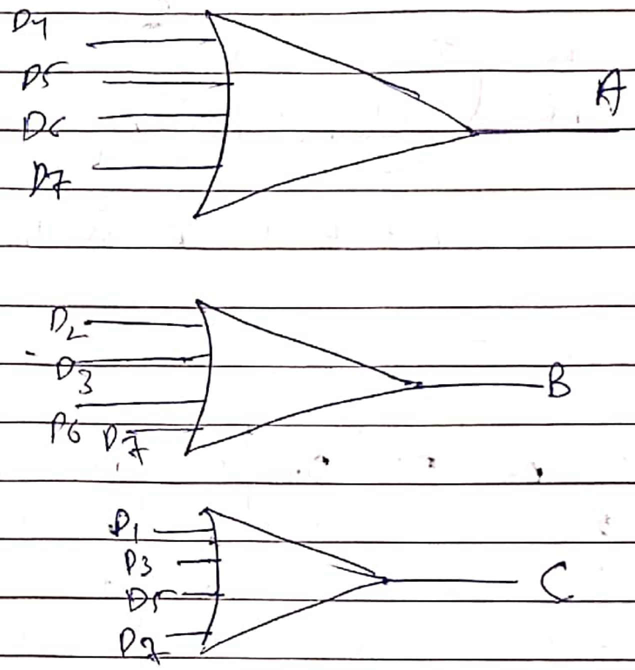
n = 3.

D_0	D_1	D_2	D_3	D_4	D_5	D_6	D_7	A	B	C
1	0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0	1
0	0	1	0	0	0	0	0	0	1	0
0	0	0	1	0	0	0	0	0	1	1
0	0	0	0	1	0	0	0	1	0	0
0	0	0	0	0	1	0	0	1	0	1
0	0	0	0	0	0	1	0	1	1	0
0	0	0	0	0	0	0	1	1	1	1



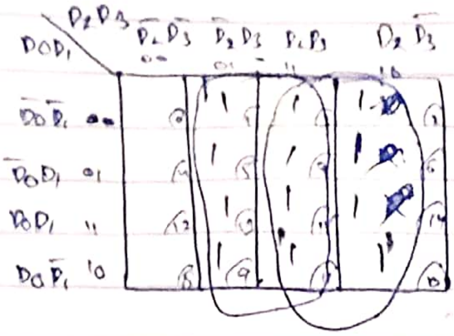
Enable Signal.

$A \rightarrow D_4 + D_5 + D_6 + D_7$



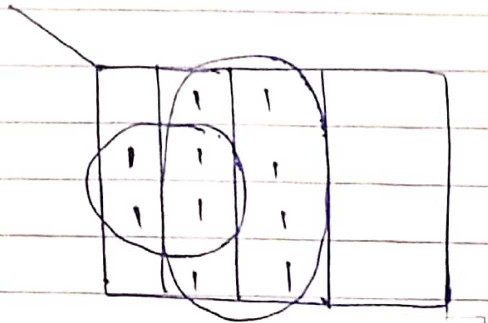
Priority Encoder → If two and more inputs are equal to 1 at the same time then input having the highest priority will be taken for the corresponding output.

D_0	D_1	D_2	D_3	Y_0	Y_1	V
0	0	0	0	X	X	0
1	0	0	0	0	0	1
X	1	0	0	0	1	1
X	X	1	0	1	0	1
X	X	X	1	1	1	1



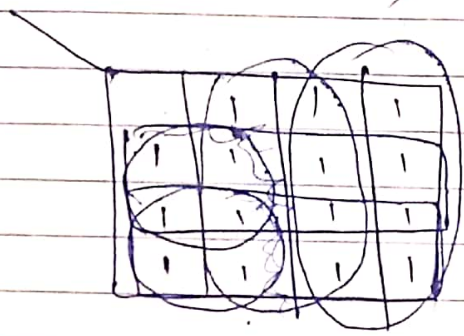
Y_0

D_3 $D_3 + D_2$.



Y_1

$D_1\bar{D_2} + D_3$



N

$D_1 D_2 + D_3 + D_2$

$D_2 + D_3 + D_1 + D_0$

Code Converter :-

Binary to BCD :-

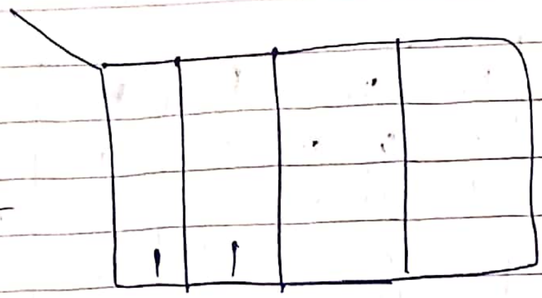
	A	B	C	D	B_4	B_3	B_2	B_1	B_0
0	0	0	0	0		0	0	0	0
1	0	0	0	1		0	0	0	1
2	0	0	1	0		0	0	1	0
3	0	0	1	1		0	0	1	1
4	0	1	0	0		0	1	0	0
5	0	1	0	1		0	1	0	1
6	0	1	1	0		0	1	1	0
7	0	1	1	1		0	1	1	1
8	1	0	0	0		1	0	0	0
9	1	0	0	1		1	0	0	1
10	1	0	1	0	1	0	0	0	0
11	1	0	1	1	1	0	0	0	1
12	1	1	0	0	1	0	0	1	0
13	1	1	0	1	1	0	0	1	1
14	1	1	1	0	1	0	1	0	0
15	1	1	1	1	1	0	1	0	1

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

B_4

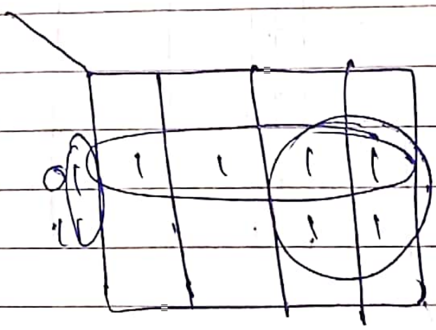
$AB + AC$

B3



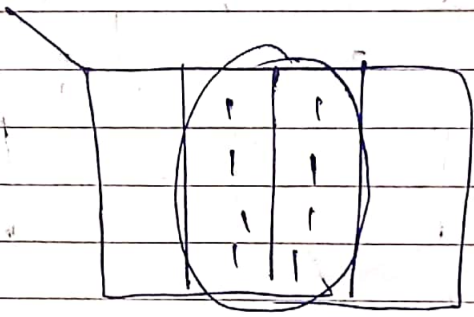
$A\bar{B}\bar{C}$

B4



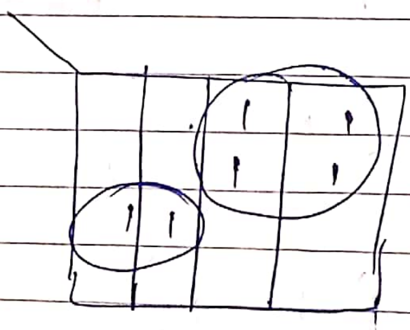
$\bar{A}B + BC$

B6



D

B7



$AB\bar{C} + \bar{A}C$

Scanned with CamScanner

BCD to Excess 3

	A	B	C	D	B_3	B_2	B_1	B_0
0	0	0	0	0	0	0	1	1
1	0	0	0	1	0	1	0	0
2	0	0	1	0	0	1	0	1
3	0	0	1	1	0	1	1	0
4	0	1	0	0	0	1	1	1
5	0	1	0	1	1	0	0	0
6	0	1	1	0	1	0	0	1
7	0	1	1	1	1	0	1	0
8	1	0	0	0	1	0	1	1
9	1	0	0	1	1	1	0	0

B_3

AB \ CD	$\bar{C}\bar{D}$	$\bar{C}D$	CD	$C\bar{D}$
00				
01		1	1	1
11	X	X	X	X
10	1	1	X	X

$BD + BC + A$

~~$\bar{A}B\bar{C} + ABD + \bar{A}BC$~~

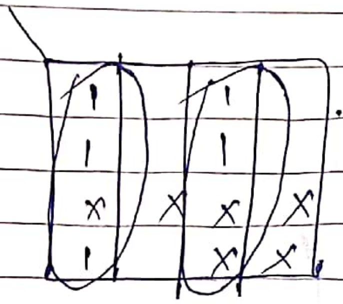
B_2

AB \ CD	$\bar{C}\bar{D}$	$\bar{C}D$	CD	$C\bar{D}$
00		1	1	1
01	1			
11	X	X	X	X
10		1	X	X

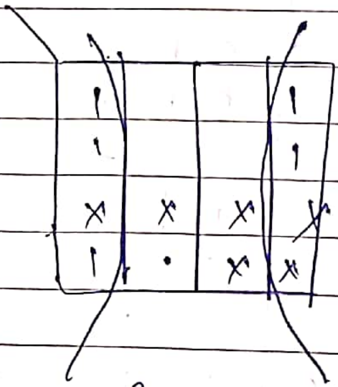
$\bar{B}\bar{C}D + \bar{A}\bar{B}D + \bar{A}\bar{B}C + \bar{A}B\bar{C}\bar{D}$

$\bar{B}(D + C) + B\bar{C}\bar{D}$

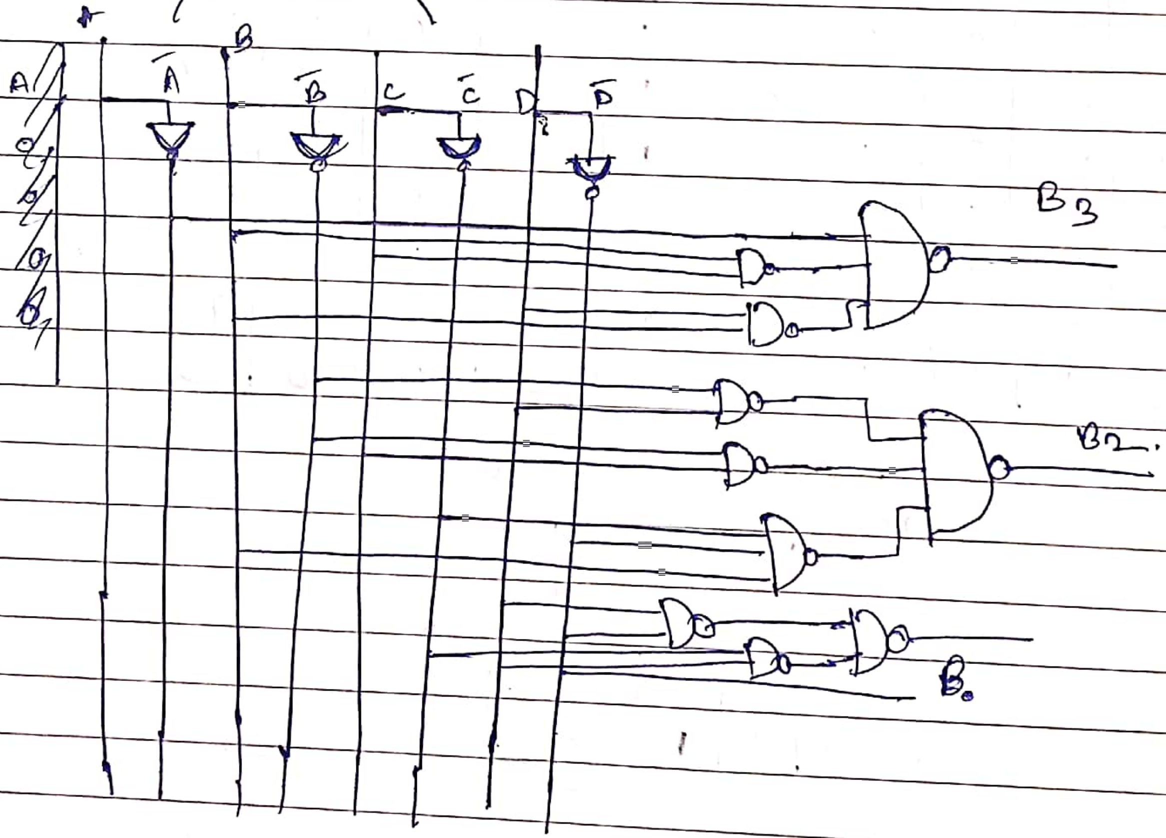
Scanned with CamScanner



$\bar{A}\bar{D} + CD$



$\Rightarrow \bar{D}$



Binary to Gray Converter (4 variable).

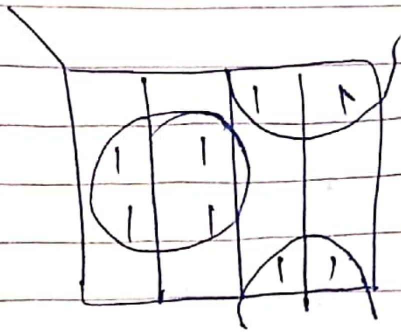
	A	B	C	D	G_1	G_2	G_3	G_4	
0	0	0	0	0	0	0	0	0	0
1	0	0	0	1	0	0	0	1	1
2	0	0	1	0	0	0	1	1	3
3	0	0	1	1	0	0	1	0	2
4	0	1	0	0	0	1	1	0	6
5	0	1	0	1	0	1	1	1	7
6	0	1	1	0	0	1	0	1	5
7	0	1	1	1	0	1	0	0	4
8	1	0	0	0	1	1	0	0	12
9	1	0	0	1	1	1	0	1	13
10	1	0	1	0	1	1	1	1	15
11	1	0	1	1	1	1	1	0	14
12	1	1	0	0	1	0	1	0	10
13	1	1	0	1	1	0	1	1	11
14	1	1	1	0	1	0	0	1	9
15	1	1	1	1	1	0	0	0	8

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

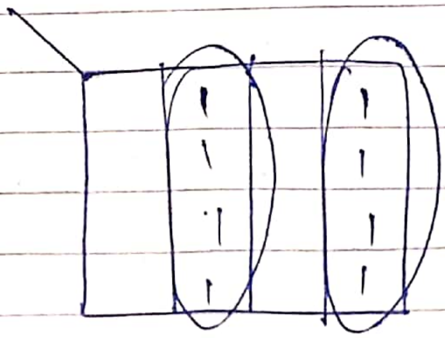
~~AB + AD + AC~~ A.

AB \ CD				
	1	1	1	1
	1	1	1	1

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



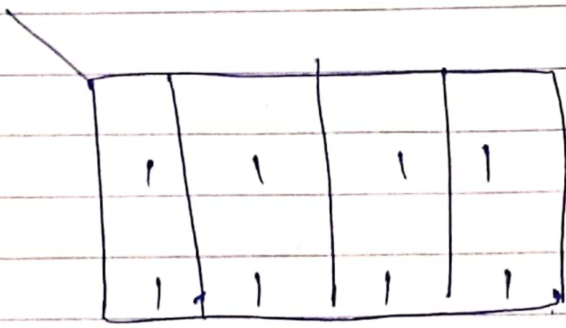
$B\bar{C} + \bar{B}C$ $\quad B \oplus C$



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

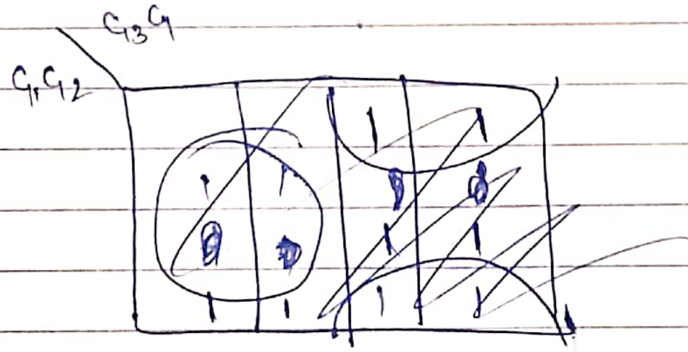
$C_1C_2 \backslash C_3C_4$	$\bar{C}_3\bar{C}_4$	$\bar{C}_3C_4$	C_3C_4	$C_3\bar{C}_4$
$\bar{C}_1\bar{C}_2$	0 (0)	0 (1)	0 (3)	0 (2)
$\bar{C}_1C_2$	(4)	(5)	(7)	(6)
C_1C_2	1 (12)	1 (13)	1 (15)	1 (14)
$C_1\bar{C}_2$	1 (8)	1 (9)	1 (11)	1 (10)

A

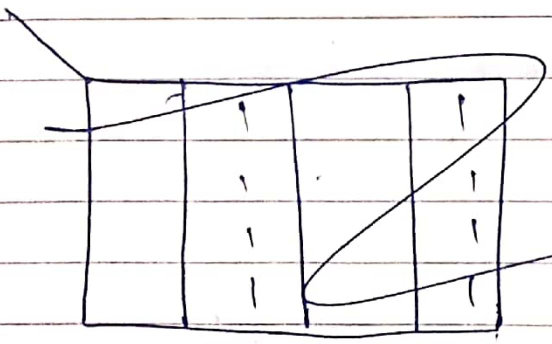


$\bar{C}_1C_2 + C_1\bar{C}_2$.

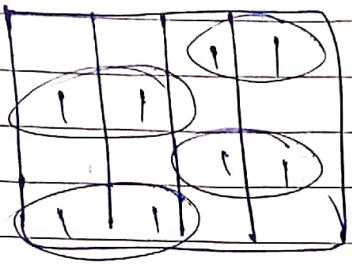
$C_1 \oplus C_2$.



$C_2\bar{C}_3 + C_3\bar{C}_2$.

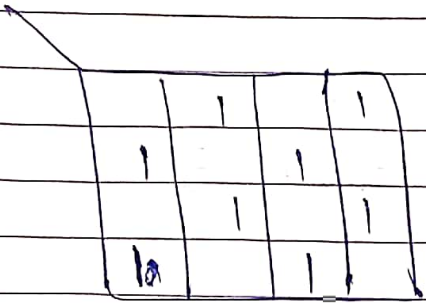


$\bar{C}_3C_4 + C_3\bar{C}_4$.



$\bar{G_1}\bar{G_2}G_3 + \bar{G_1}G_2\bar{G_3} + G_1G_2G_3 + G_1\bar{G_2}\bar{G_3}$.

$G_1 \oplus G_2 \oplus G_3$.



$\bar{G_1}\bar{G_2}\bar{G_3}G_4 + \bar{G_1}\bar{G_2}G_3\bar{G_4} + \bar{G_1}G_2(\bar{G_3}\bar{G_4} + G_3G_4) + G_1G_2(\bar{G_3}G_4 + G_3\bar{G_4}) + G_1\bar{G_2}(\bar{G_3}\bar{G_4} + G_3G_4)$.

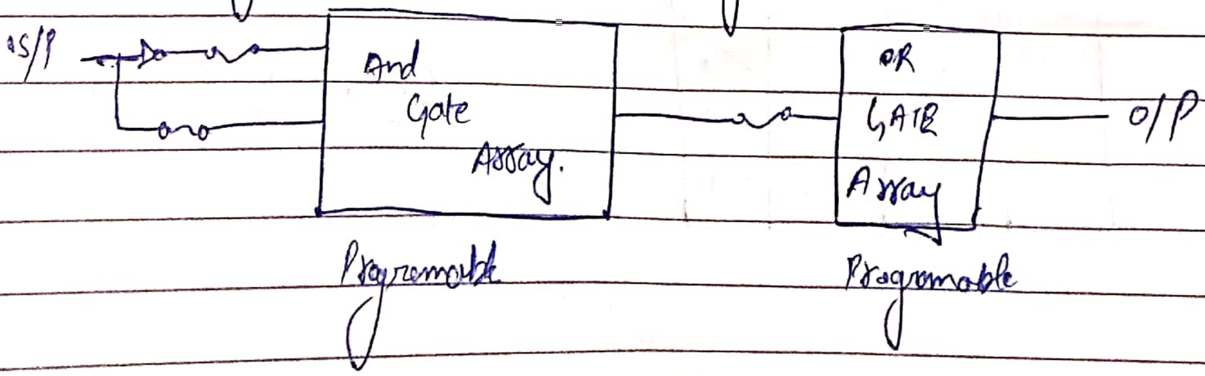
$\cancel{G_3G_4} + (G_3 \oplus G_4)(G_1 \odot G_2) + (G_3 \odot G_4)(G_1 \oplus G_2)$

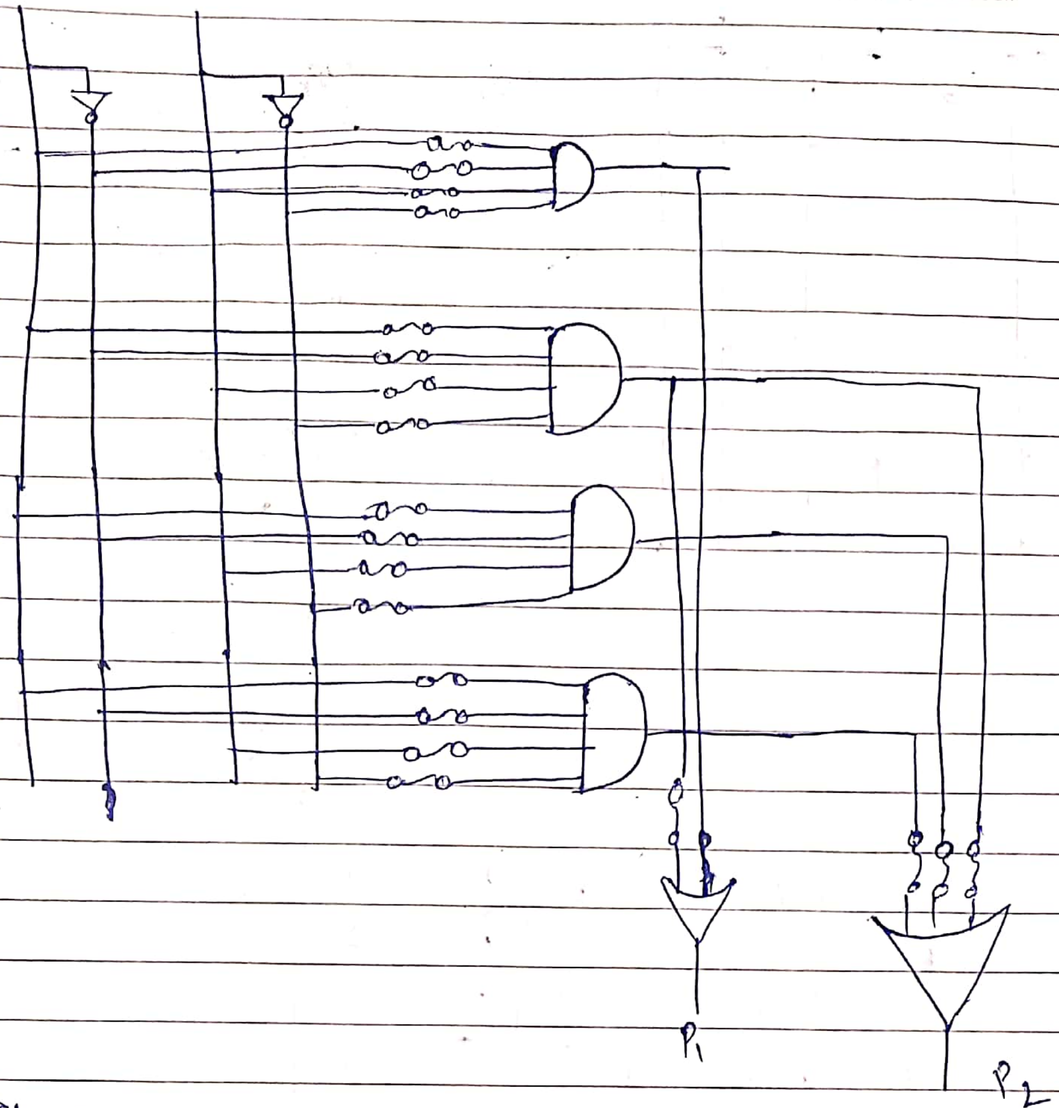
$\Rightarrow G_1 \oplus G_2 \oplus G_3 \oplus G_4$

Programable Logic devices :-

① Simple programable logic devices (SPLD)
② Programable logic array (PLA)
③ Programable array logic (PAL)

Complex PLD.
Field Programable Gate Array (FPGA)

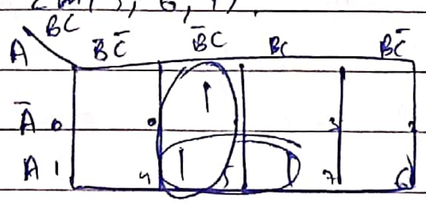




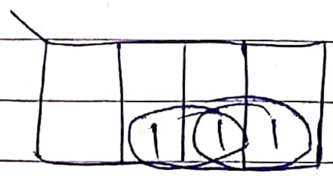
If both and and OR are programmable then it comes under category Programmable logic Array (PLA).

$F_1(A,B,C) = \Sigma m(1,5,7)$

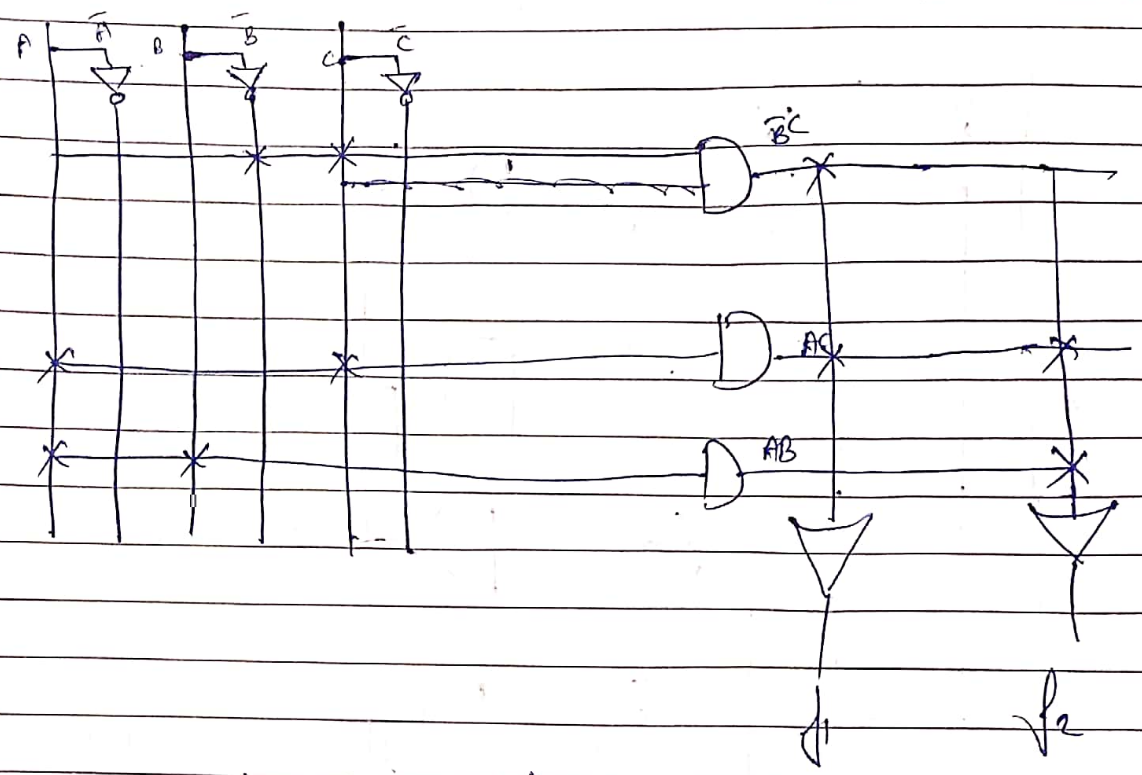
$f_2(ABC) = \Sigma m(5,6,7)$



$\bar{B}C + AC$



$AC + AB$

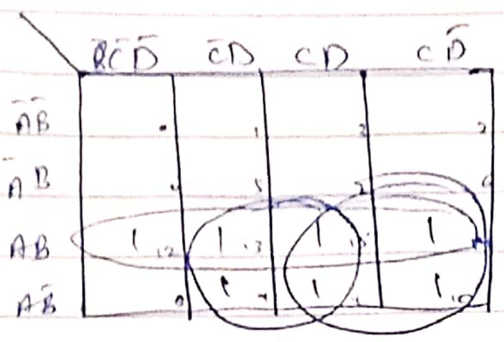


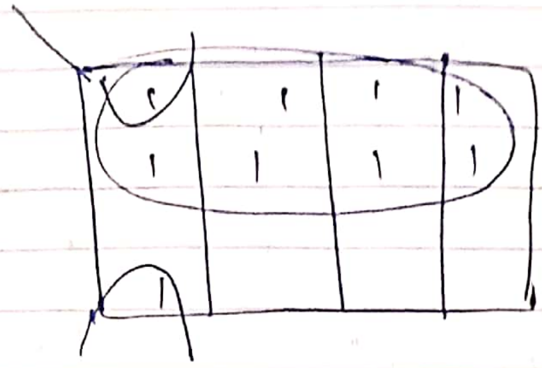
Product Term	A	B	C	O/P f_1	f_2
$\bar{B}C$	–	0	1	1	–
AC	1	–	1	1	1
AB	1	1	–	–	1

Q→ A combinational logic circuit has 4 inputs and 2 outputs f_1 & f_2. The output f_1 gives the high output when input combination is >= 1001.(9) otherwise no output the output f_2 gives high output when input combination is < 1001 otherwise low

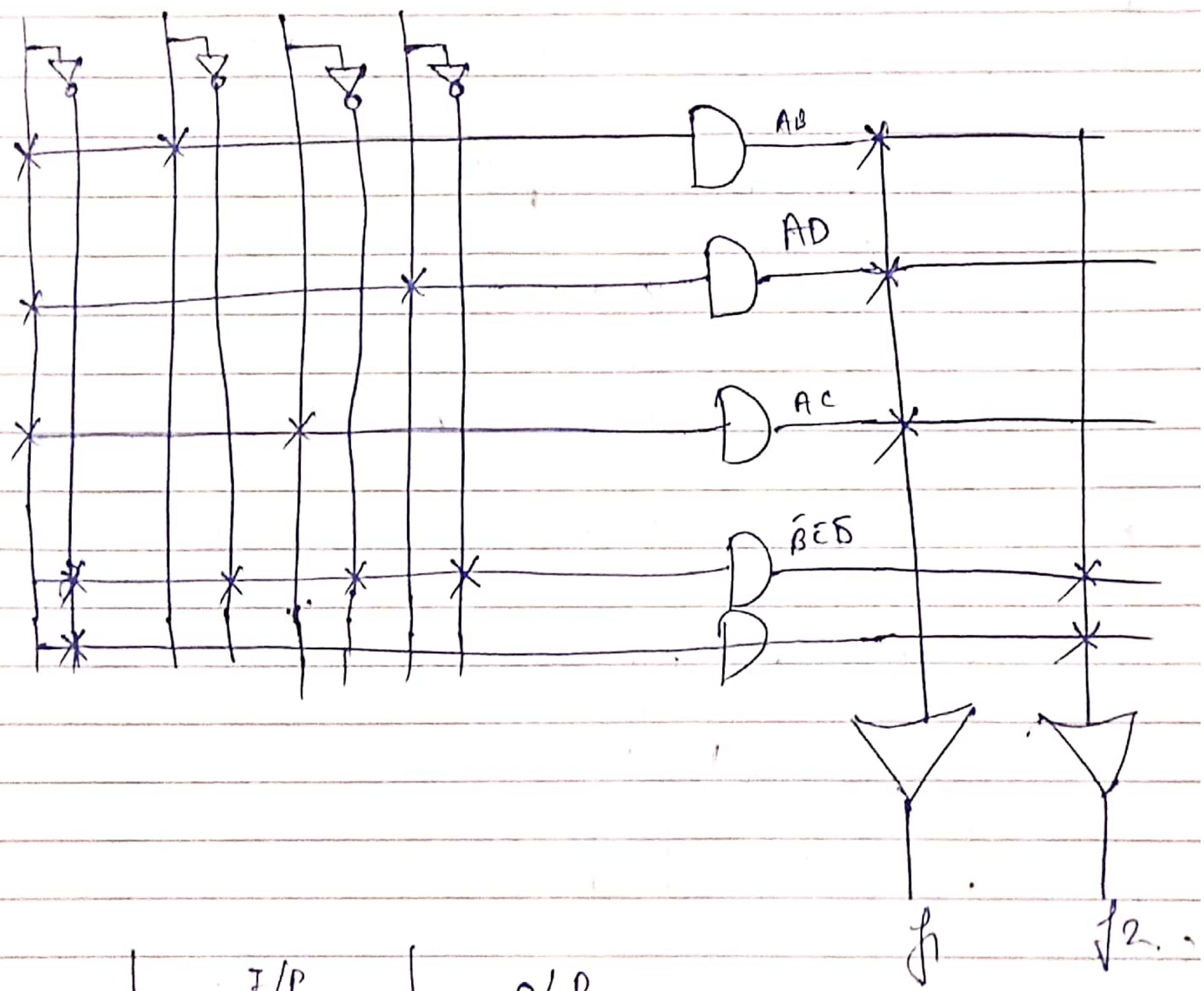
$f_1 = \Sigma m(9, 10, 11, 12, 13, 14, 15)$

$f_2 = \Sigma m(0, 1, 2, 3, 4, 5, 6, 7, 8)$



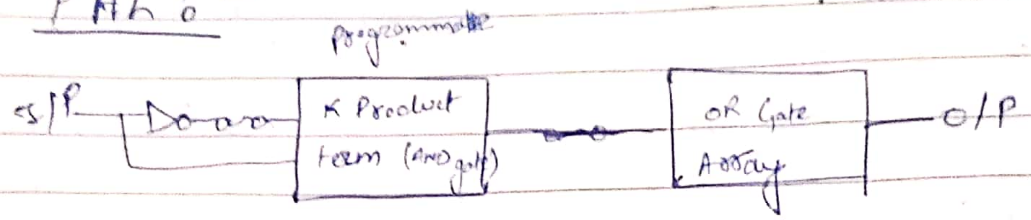


$AB + AD + AC$ $\qquad$ $\bar{A} + A\bar{B}\bar{C}\bar{D}$



	I/P				O/P	
	A	B	C	D	f_1	f_2
AD	1	–	–	1	1	–
AC	1	–	1	–	1	–
AB	1	1	–	–	1	–
$\bar{B}\bar{C}\bar{D}$	–	0	0	0	–	1
$\bar{A}$	0	–	–	–	–	1

PAL :-

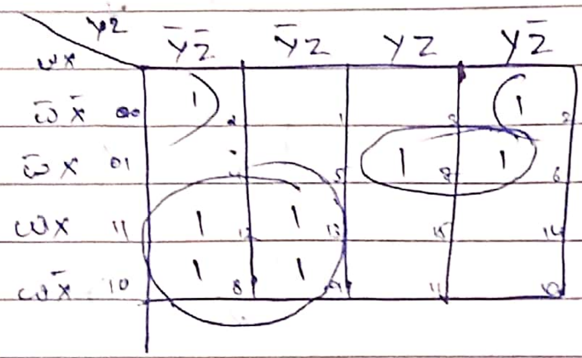


$A(w, x, y, z) = \Sigma m(0, 2, 6, 7, 8, 9, 12, 13)$

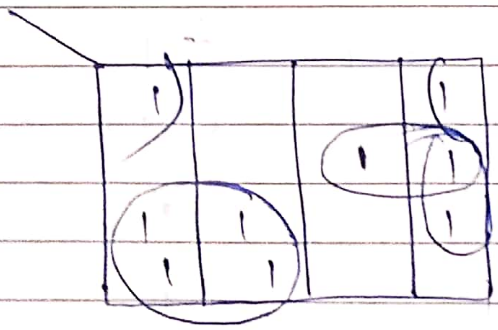
$B(w, x, y, z) = \Sigma m(6, 7, 8, 9, 0, 2, 6, 12, 13, 14)$

$C(w, x, y, z) = \Sigma m(1, 3, 4, 6, 10, 12, 13)$

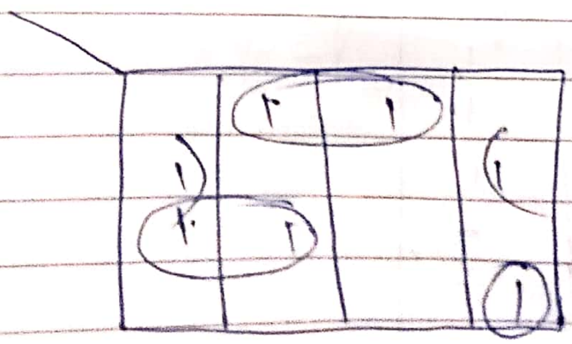
$D(w, x, y, z) = \Sigma m(1, 3, 4, 6, 9, 12, 14)$



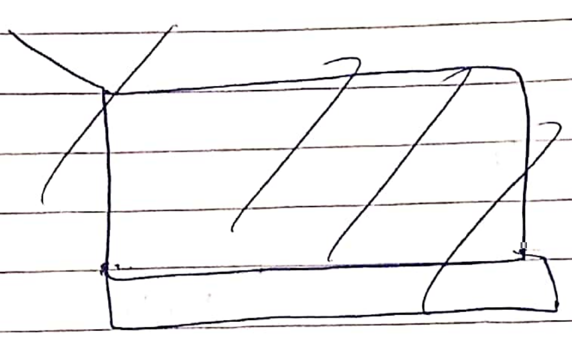
$A = w\bar{y} + \bar{w}\bar{x}\bar{z} + \bar{w}xy$

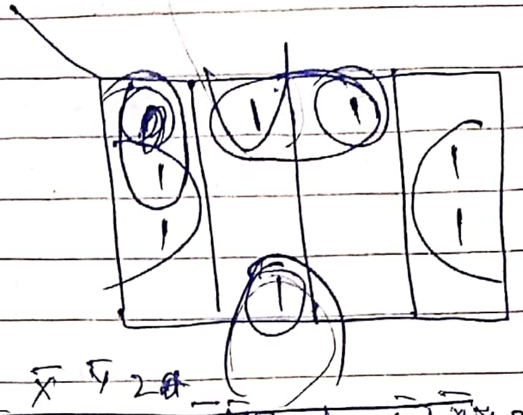


$B = w\bar{y} + \bar{w}\bar{x}\bar{z} + \bar{w}xy + xy\bar{z}$



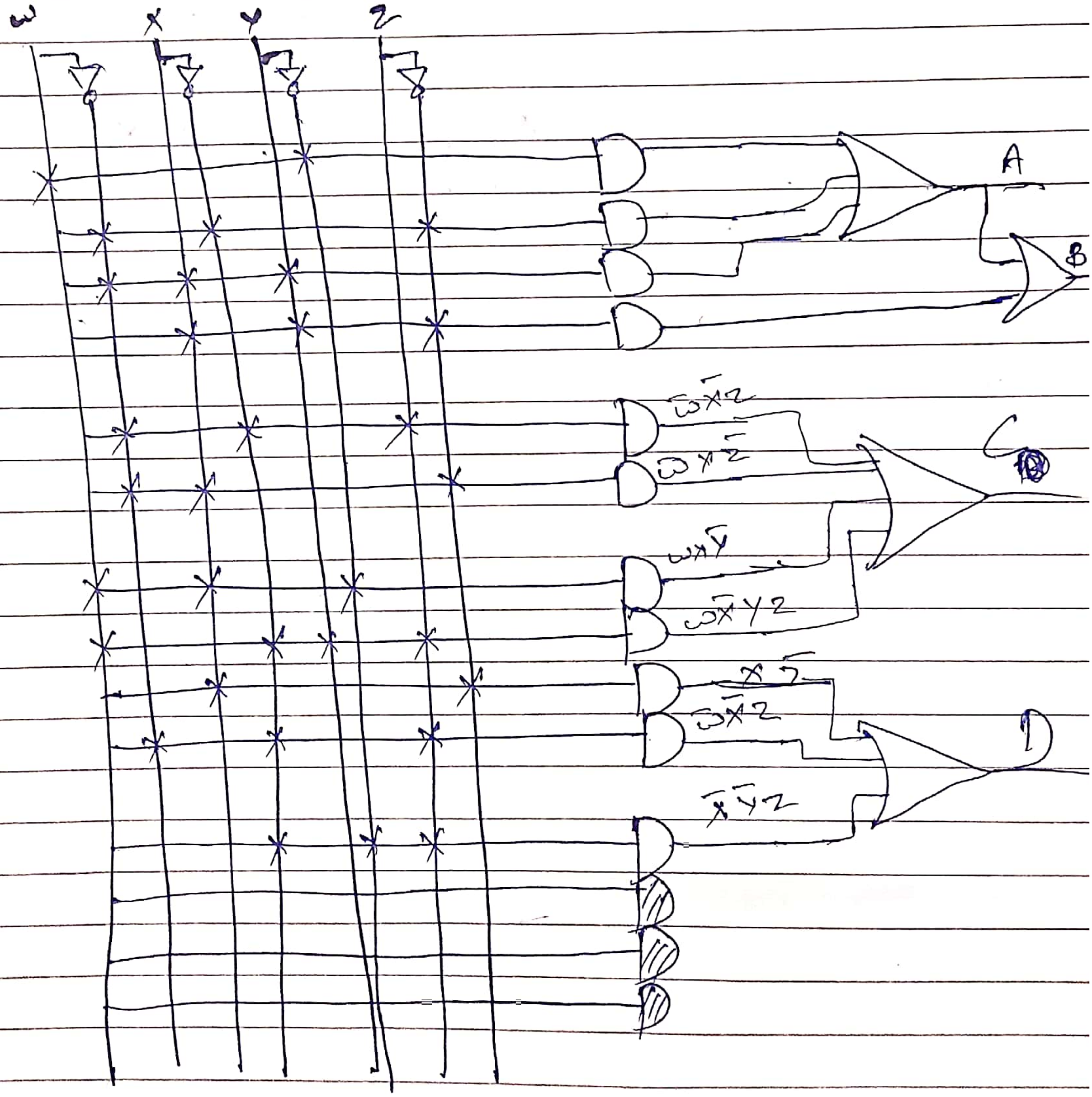
$C = wx\bar{y} + \bar{w}\bar{x}z + \bar{w}x\bar{z} + w\bar{x}y\bar{z}$





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

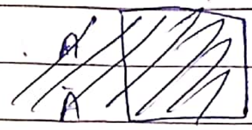
$D = x\bar{z} + \bar{w}\bar{y}\bar{z} + w\bar{x}\bar{y}z + \bar{w}\bar{x}yz$

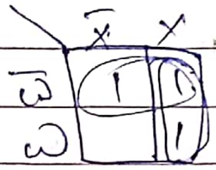


ROM - (Read only memory) → And fixed but or programable.

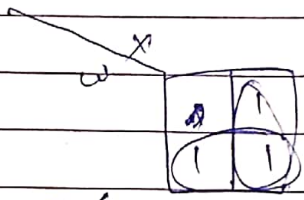
$A = \sum m(\cancel{0}, 2, 3)$

$B = \sum m(1, 2, 3)$





$\bar{w} + x$



$w + x$