classmate
Date
Page

⇒ Signal :- Variation of any parameter with respect to time is called as 'Signal'.

→ Continuous Signal - For particular ~~to~~ range if it continuously changes then it is called ..

→ Discrete Signal - Not continuous.

→ Digital and Discrete both are different.

⇒ Digital signal — In this signal there are two logic systems. (1) Positive (2) Negative.
→ In positive if range is -5 to 5 then 5 is high level and -5 is low level.
→ In Negative if range is -5 to 0 then -5 is high and 0 is low.

⇒ Adv. of Digital Signal -

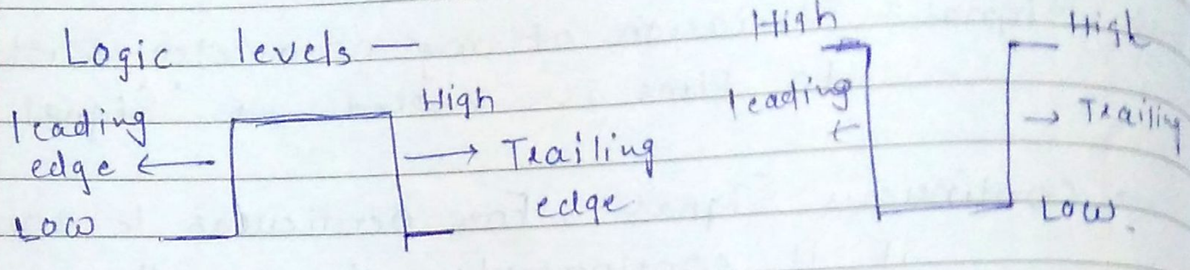
(1) Easier to Design.
(2) Information storage
(3) Accuracy & Precision.
(4) More Versatile. (Programmable / flexible)
(5) Less affected by noise.
(6) More Digital circuits can be fabricated on ~~the~~ IC.

0 - 0.8 low ~~level~~ stage
0.8 - 2 Indeterminate stage
2 - 5 High stage

⇒ Disadvantages - Processing time increases. (Analog - Digital - Analog)

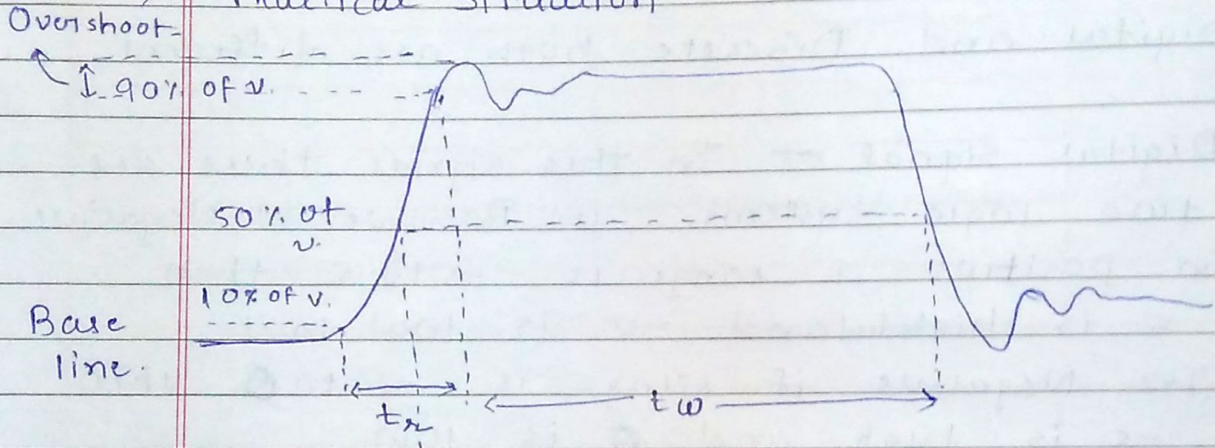
REAL WORLD IS ANALOG

⇒ Logic levels —



Positive Logic. Negative logic.

⇒ Practical Situation —



→ t_r (Rise time) - Time Duration between 10% of v to 90% of voltage

→ t_w (Width of Pulse) - 50% of leading edge
50% of ~~of~~ Trailing edge

⇒ Elements of ~~Logic~~ digital logic —

→ System Design.
→ Circuit design.
→ Logic design.

⇒ Functions of Digital Logic —

→ Arithmatic
- Adder
- Substraction.
- Multiplication.
- Division

→ Code converters

→ Encoders.

→ Decoders.

→ MUX (Multiplexer)

→ DMUX (De-Multiplexer)

Problems —

(1) Find decimal equivalent of $(231.23)_4$.

→ $(231.23)_4 \longrightarrow (\quad)_{10}$.

$$\begin{array}{ccccc} 2 & 3 & 1. & 2 & 3 \\ 4^2 & 4^1 & 4^0 & 4^{-1} & 4^{-2} \end{array}$$

$= 2 \times 16 + 12 + 1 . \left(\frac{2}{4} + \frac{3}{16}\right)$

$= 32 + 13 . \left(\frac{8}{16} + \frac{3}{16}\right) = (45.068)_{10}$.

(2) Count from 0 to 9 in radix 5.

$(0)_{10} \rightarrow (\quad)_5 = 0$

$(1)_{10} \rightarrow (\quad)_5 = 1$

$(2)_{10} \rightarrow (\quad)_5 = 2$

$= 3$

$= 4$

$$\begin{array}{r|l} 5 & 5 \quad 0 \\ & 1 \quad 1 \end{array} = 10$$

$$\begin{array}{r|l} 5 & 6 \quad 1 \\ & 1 \end{array} = 11$$

$= 12$

$= 13$

$= 14$

⇒ Substraction using 2's complement.

(1) 11010 − 10000.

Here, 1's comp. 10000

01111

2's comp. 01111

+ 1

10000

Using 1's co[mp]

11010

① 01111

01001

→ 0100[1]

Add to LSB

= 0101[1]

If no carry take ... put negative ...

→ ~~Here, carry is not generated so again take 2's comp. and put (−) sign.~~

~~10000~~

~~2's comp. 10000~~

①

11010

10000

01010

→ Here carry is generated. Ans. = 01010

(2) 11010 − 01101.

01101

1's 10010

2's 1

10011

Ans. = 01101

11010

10011

① 01101

1's comp. method.

11010

10010

① 01100

1

$\boxed{01101}$

(3) 100 − 11000.

1's	11000
	00111
2's	1
	01000

00100
01000
01100

→ Here carry is not generated. so again take 2's comp.

	01100
1's.	10011
2's	1
	$\boxed{= 10100}$

00100
00111
01011

~~10100~~
~~1~~
~~10101~~

Here carry is not gen
01011
1's. $\boxed{10100}$

(4) 1010100 − 1010100.

1's	1010100
2's	0101011
	1
	0101100

1010100
0101100
(1) 0000000

Ans. = 0000000.

1010100
0101011
1111111
1's $\boxed{0000000.}$

4 2 1

⇒ Signed Binary no.

→ Sign magnitude.
→ Signed 1's comp.
→ Signed 2's comp.

→ e.g. We want represent (−14) by all 3 method

−14	=	1 0001110	→	Sign Magnitude
1's	=	0111 0001		
2's	=	1000 1110		

sign → Magnitude

⇒ Multiplication —

(1) $(011)_2 \times (110)_2$

```
    011
    110
  -----
  01100
  00110
  -----
  10010
```

Ans: 10010.

(2) $(1110)_2 \times (1010)_2$

```
      1110
      1010
  --------
   1110000
   0000000
     11100
  --------
  10001100
```

Ans: 10001100.

Scanned by CamScanner

⇒ Division —

(1) $(11011011)_2 / (110)_2$ (2) $(110101101)_2 / (101)_2$

```
        10010                      1010101
110 | 11011011           101 | 110101101
      110                          101
      ---------                    -----
         110                       00110
         110                         101
         ---------                   -----
             11                      00111
                                       101
                                       -----
                                       00101
                                         0101
                                         -----
                                         0100
```

⇒ −45 (signed no. so, we have to consider sign and Magnitude both).

−45 = ~~110101101~~
= 1 0 1 0 1 1 0 1
= 1 1 0 1 0 0 1 0 (1's comp. Representation).
= 1 1 0 1 0 0 1 1 (2's comp.

⇒ −73.75

73.75

2	73	1
2	36	0
2	18	0
2	9	1
2	4	0
2	2	0
	1	1

0.75 × 2 = 1.50 (1)
0.50 × 2 = 1.00 (1

sign (+) = 0 1001001 . 1100

1's comp. = 0 0110110 . 0011

2's comp. = 0 0110110 . 0011
+ 1

00110110 . 0100

= 00110110 . 0100

→ Before point we have to try to keep 4, 8, 12... bits and further after point.

⇒ 46 − 14 (8 bit 2's complement)

46 = 0 0101110

−14 = 1 0001110

1 1110001 (1's comp.)
+ 1

1 1110010 (2's comp.)

46 + (−14) = 0 0101110
+ 1 1110010

1 0 0100000

Scanned by CamScanner

⇒ -75 + 26

26 = 0 0011010

-75 = 1 1001011
1 0100100
1 0100101

00011010
10100101

~~10111111~~

-75 = 1 1001011
1 0110100
1 0110101

0 0011010
1 0110101
---
(1) 1001111

Here carry is not generated.

01000000
1

01000001

0 0110000
1
---
0 0110001

= (-49).

⇒ -45.75 + 87.5. (Using 12-bit 2's)

-45 = 1 0101101
= 1 1010010
= 1 1010011

0.75 × 2 = 1.50 1
0.50 × 2 = 1.00 1

-45.75 = 1 1010011. 1100.

Scanned by CamScanner

87 = 01010111

0.5×2 = 1.0   1

87.5 = 01010111.1000

  01010111. 1000
+ 11010011. 1100

1 00101011. 0100

Ans. 00101011. 0100.

⇒ 27.125 − 79.625.

27 = 00011011
0.125 × 2 = 0.250   0
0.250 × 2 = 0.500   0.
0.500 × 2 = 1.000   1

27.125 = 00011011.0010

−79 = 1 1001111. 101 = 1 1001111.101
~~0.625~~ = 1 0110000   1 0110000.010
1 0110001

1 0110000.011

0.625 × 2 = 1.250   1
0.250 × 2 = 0.500   0
0.500 × 2 = 1.000   1

~~−79.625 = 1 0110001~~. 1010.

0 0011011.0010
+ 1 0110000.0010

1 1001101.1000

carry is not generated.

1 0110010.0011
+ 1

1 0110010.0100

⇒ −31.5 − 93.125

−31 → 1 0011111

0.5 × 2 = 1.00 1.

−31.5 = 1 0011111.1000
= 1 1100000.0111
+ 1

1 1100000.1000

−93 = 1 1011101

0.125 × 2 = 0.250 0.
0.250 × 2 = 0.500 0.
0.500 × 2 = 1.000 1

−93.125 = 1 1011101.0010
1 0100010.1101
+ 1

1 0100010.1110

$$
\begin{array}{r}
\ 1\ 1\ \ \ \ \ \ \ 1\ \ \ \ \ \ \ \ \\
=\ 1\ 1100000.1000 \\
1\ 0100010.1110 \\
\hline
1\ 1\ 00000111.0110
\end{array}
$$

→ Here carry is generated but 8th digit is 1 means negative. so again we have to take 2's comp.

⇒ 9's complement –

→ ~~For~~ first subtract the value with 9's with same digit that value have

3465 782.54

$$
\begin{array}{r}
9\ 9\ 9\ 9 \\
-\ 3\ 4\ 6\ 5 \\
\hline
6\ 5\ 3\ 4
\end{array}
\quad (9\text{'s comp.})
\qquad\qquad
\begin{array}{r}
999.99 \\
-\ 782.54 \\
\hline
217.45
\end{array}
\ (9\text{'s})
$$

$$
\begin{array}{r}
1 \\
\hline
6\ 5\ 3\ 5
\end{array}
\quad (10\text{'s comp.})
$$

→ In 1's comp. if carry is generated then we were adding carry at LSB and answer is positive. Same rule can be applied for 9's and 10's (2's) complement

classmate
Date
Page

(1) 745.81 using 9's comp.
- 436.62

-436.62 = 999.99
- 436.62
563.37

.38

745.81
563.37
1 309.18 (Positive ⇒ Carry is generated)
1
309.19

Ans: 309.19.

(2) 436.62 - 745.81.

-745.81 = 999.99
- 745.81
254.18

436.62
254.18
690.80 (negative → carry is not generated)

→ Again take 9's comp. of ans.

999.99
690.80
309.19 (negative)

Scanned by CamScanner

(3) 2928.54 − 416.73 (using 10's comp.)

$$\begin{array}{r} 9999.99 \\ -\ 0416.73 \\ \hline 9583.26 \end{array}$$ (9's)

$$\begin{array}{r} 2928.54 \\ -\ 9583.26 \\ \hline 345.28 \end{array}$$

$$\begin{array}{r} 9583.26 \\ 1 \\ \hline 9583.27 \end{array}$$ (10's comp.)

$$\begin{array}{r} 2928.54 \\ 9583.27 \\ \hline 1\ 2511.81 \end{array}$$

Ans. 2511.81 (positive)

(4) −2928.54 + 0416.73

$$\begin{array}{r} 9999.99 \\ -\ 2928.54 \\ \hline 7071.45 \\ 1 \\ \hline 7071.46 \end{array}$$ (10's comp.)

$$\begin{array}{r} 7071.46 \\ 0416.73 \\ \hline 7488.19 \end{array}$$

carry is not generated.

$$\begin{array}{r} 9999.99 \\ 7488.19 \\ \hline 2511.80 \end{array}$$

(negative

Weighted codes

	8	4	2	1
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1

⇒ Classification of Binary code –

(1) Alphanumeric
- ASCII
- EBCDIC
- Hollerith

(2) Numeric
- Weighted codes
- Non-weighted codes

Weighted codes
- Binary
- BCD
  - 8421
  - 2421
  - 3321
  - 5211
  - 84-2-1
  - 74-2-1

e.g. 16

0001 0110.

= 00010110.

Numeric
- Weighted
- Non-weighted
- Self-complementing

  e.g. 2421, 5211

  if we know code of N then 9-N can be find out by changing 0to1 and 1to0.
- Sequential
- Error detecting & correcting
- Cyclic

  e.g. Gray code

  0001

  0010

  0100.

Here for 0-4 we have to give preference from LSB & then MSB

We have to start from

	8421	2421	5211	6 4 2 -3	XS-3
0	0000	0000	0000	0000	0011
1	0001	0001	0001	0101	010
2	0010	0010	0011	0010	0101
3	0011	0011	0101	[illegible]	0110
4	0100	0100	0111	0100	0111
5	0101	1011	1000	1011	100
6	0110	1100	1010	1000	100
7	0111	1101	1100	1101	101
8	1000	1110	1110	1010	101
9	1001	1111	1111	1111	110

Self complementing

9 − N → by simply changing 0 → 1, 1 → 0.

⟹ BCD Addition −

```
         1 1
  25     0010  0101
+ 13   + 0001  0011
       ------------
         0011  1000
```

→ Here carry is not generated and codes are legal (0 − 9) is that is the final answer.

Scanned by CamScanner

(1)
$$\begin{array}{r} \overset{1}{0}110 \quad \overset{111}{0111} \quad 1001.\ 0110 \\ 0101 \quad 0011 \quad 0110.\ 1000 \\ \hline 1011 \quad 1010 \quad 1111.\ 1110 \end{array}$$

→ Here, codes are ~~the~~ illegal (9+).
so, we will add 6.

$$\begin{array}{r} 1011 \quad 1010 \quad 1111.\ 1110 \\ 0110 \quad 0110 \quad 0110.\ 0110 \\ \hline 1\ 0001 \quad 0000 \quad 0101\ 0100 \\ 1 \quad\quad 1 \quad\quad 1 \\ \hline 0001\ 0010 \quad 0001 \quad 0110.\ 0100 \end{array}$$

Ans: 0001 0010 0001 0110. 0100.

(2) BCD Subtraction —

(i) 38 − 15.

38 = 0011 1000 , 15 = 0001 0101

~~−15 = 9999 9999~~
~~0001 0101~~

~~−15 = 0001 0101~~
~~1110 1010~~ (1's comp.)

$$\begin{array}{lr} 38 & 0011\ 1000 \\ -15 & -\ 0001\ 0101 \\ \hline & 0010\ 0011 \end{array}$$

Ans: 0010 0011.

(ii) 206.7 − 147.8

206.7	0010	0000	0110	.	0111
− 147.8	0001	0100	0111	.	1000
	0000	1011	1110	.	1111

→ code is illegal (9+)

0000	1011	1110	.	1111
−	0110	0110	.	0110
0000	0101	1000	.	1001

Ans: 0000 0101 1000 . 1001

⇒ 679.6 − 885.9 using 9's comp.

679.6 = 0110 0111 1001 . 0110

885.9 =

~~9999 9999 9999 . 9999~~

~~1000 1000 0101 . 1001~~

~~8999 . 8999 9898 . 8998~~

999.9
885.9
114.0 (9's comp.)

114.0 = 0001 0001 0100 . 0000

$$
\begin{array}{r}
\ \ 111 \\
0110\ \ 0111\ \ 1001.\ 0110 \\
+\ 0001\ \ 0001\ \ 0100.\ 0000 \\
\hline
0111\ \ 1000\ \ 1101\ .0110 \\
1\ \ \ 0110 \\
\hline
\end{array}
$$

Ans: 0111 1001 0011 . 0110

⇒ ~~342.7 - 108.9 using~~

→ Here, carry is not generated. so again we have to take 9's comp.

7 9 3 . 6

$$
\begin{array}{r}
9\ 9\ 9\ 9 \\
-\ 7\ 9\ 3\ .\ 6 \\
\hline
2\ 0\ 6\ .\ 3
\end{array}
$$

$\boxed{\text{Ans: } 0010\ \ 0000\ \ 0110.\ \ 0011}$

⇒ 342.7 - 108.9 using 10's comp.

342.7 = 0011 0100 0010. 0111

$$
\begin{array}{r}
999.9 \\
-\ 108.9 \\
\hline
891.0 \\
1 \\
\hline
891.1
\end{array}
$$
(10's comp.)

classmate
Date
Page

891.1 = 1000 1001 0001. 0001

so, 0011 0100 0010. 0111
1000 1001 0001. 0001
1011 1101 0011. 1000
0110 0110

Ans: (1) 0010 0011 , 0011 .1000

⇒ Gray code —

e.g. 1011. (Binary).

(MSB) 1110 (Gray).

1011 (Binary).

(n = MSB
$G_n = B_n$ → XOR
$G_{n-1} = B_n \oplus B_{n-1}$
$G_{n-2} = B_{n-1} \oplus B_{n-2}$)

($B_n = G_n$
$B_{n-1} = B_n \oplus G_{n-1}$)

Decimal	Binary	Gray.
0	0000	0000
1	0001	0001
2.	0010	0011
3	0011	0010
4	0100	0110
5	0101	0111
6	0110	0101
7	0111	0100
8	1000	1100
9	1001	

Scanned by CamScanner

⇒ Error Selecting codes –

odd perity system :- No. of 1s are odd.
even perity system :- No. of 1s are even

→ At ~~reciever~~ transmitter we know that which perity system we have but if at reciever if we get even perity system, then we can say that there is some error in our code.

	Perity	
1100	1	→ odd perity system.
1101	1	→ even " " .

⇒ Block perity – (Error correcting code) –

		P
	01011011	0
	10010101	1
	01101110	0
	10001101	1
	01001100	0
P	10011110	1

→ Here In row, if we get error and error is column then we can get errored code and we will take complement of that code.

$\Rightarrow$ 7-bit Hamming code -

e.g. P_1 P_2 D_3 P_4 D_5 D_6 D_7

at 2^0 = 1^{st} parity
at 2^1 = 2^{nd} parity
at 2^2 = 3^{rd} parity.

$\rightarrow$ And we know the data code

P_1	P_2	D_3	P_4	D_5	D_6	D_7	P_8
		1		1	1	0	

2^0 $\rightarrow$ $1 \overset{+1}{,} 3 \overset{+1}{,} 5, 7$

2^1 $\rightarrow$ $2, 3 \overset{+2}{,} 6, 7$

2^2 $\rightarrow$ $4, 5, 6, 7 \; {+4}$

if we have odd parity.

	at 3	at 5	at 7	
1	1	1	0	(odd Parity)
	at 3	at 6	at 7	
1	1	1	0	
	at 5	at 6	at 7	
1	1	1	0	

for odd $\rightarrow$ 1 1 1 0
for even $\rightarrow$ 0 1 1 0

$\Rightarrow$

P_1	P_2	D_3	P_4	D_5	D_6	D_7	P_8	D_9	D_{10}	D_{11}
1	0	1	0	0	1	0	1	1	1	0

D_{12}	D_{13}	D_{14}	D_{15}
1	0	1	1

$\rightarrow$ 1 3 5 7 9 11 13 15

= 1 1 0 0 1 0 0 1 = $C_1 = 0$

$\rightarrow$ 2, 3, 6, 7, 10, 11, 14, 15 = 0 1 1 0 1 0 1 1 = 1 $\rightarrow C_2$

$\rightarrow$ 4, 5, 6, 7, 12, 13, 14, 15 = 0 0 1 0 1 0 1 1 = 0 $\rightarrow C_3$

$\rightarrow$ 8, 9, 10, 11, 12, 13, 14, 15 = 1 1 1 0 1 0 1 1 = 0 $\rightarrow C_2$

$C_2 = 1$. (Error in 2nd position).

so, Corrected Ans: 1 $\bar{0}$ 1 0 0 1 0 1 1 1 0 1 0 1 1.

Scanned by CamScanner

Boolean Algebra & Minimization

⇒ Commutative law ⇒ $A + B = B + A$

$A \cdot B = B \cdot A$

⇒ Associative Law ⇒ $A + (B + C) = (A + B) + C$

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

⇒ Distributive Law ⇒ (i) $A \cdot (B + C) = (A \cdot B) + (A \cdot C)$

(ii) $A + (B \cdot C) = (A + B) \cdot (A + C)$

⇒ Absorption Law ⇒ $A + AB$

$= A \cdot 1 + A \cdot B$

$= A(1 + B)$

$= A \cdot 1$

$= A$

So, $A + AB = A$

(ii) $A \cdot (A + B)$

⇒ $A \cdot A + AB$

⇒ $A + AB$

⇒ A

So, $A \cdot (A + B) = A$.

Scanned by CamScanner

⇒ Redundant Literal Rule (RLR) —

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

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

$= (1)\cdot(A+B)$

$= A+B$

So, $\underline{A + \bar{A}B = A+B}$

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

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

$= 0 + A\cdot B$

$= A\cdot B$

So, $\underline{A\cdot(\bar{A}+B) = AB}$

⇒ Consensus Law (Included factor Theorem) —

complements

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

Proof: $AB + \bar{A}C + BC\cdot(A+\bar{A})$

$= AB + \bar{A}C + BCA + BC\bar{A}$

$= AB(1+C) + \bar{A}C(1+B)$

$= \underline{AB + \bar{A}C.}$

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

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

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

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

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

$= \bar{A}B\cdot B + ACB + \bar{A}BC + AC\cdot C$

$= \bar{A}B + ACB + \bar{A}BC + AC$

$= \bar{A}B \cdot \bar{A}B + B \cdot Ac + \bar{A}c$

$= \bar{A}B(1+C) + AC(1+C)$

$= \bar{A}B + AC$ $\quad = \bar{A}B + AC + A\cdot\bar{A}$

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

<u>OR</u> $(A+B)(\bar{A}+C)\cdot(B+C)$

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

$= [A\cdot\bar{A} + B\cdot\bar{A} + AC + BC](B+C) + (A\cdot\bar{A})$

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

$=$

⇒

One exp. can be obtain from the other by replacing each ~~pair~~ 0 with 1 & 1 with 0 & + with · & · with +

⇒ Transposition theorem —

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

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

$= A\cdot\bar{A} + AB + \bar{A}C + BC$

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

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

Que. $AB + \overline{AC} + A\overline{B}c\,(AB + c)$

$= AB + \overline{AC} + A\overline{B}C \cdot AB + A\overline{B}C \cdot C$

$= AB + \overline{A} + \overline{C} + A\overline{B} \cdot C$ $\qquad \overline{A} + A\overline{B}C = \overline{A} + \overline{B}C$

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

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

$= \overline{A} + \overline{C} + 1$

$= \overline{A} + 1$

$= 1$

Que. $\overline{\overline{A\overline{B}} + ABc + A(B + A\overline{B})}$

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

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

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

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

$= \overline{\overline{(\overline{B} + c)} + 1} \qquad = \overline{1}$

$= 0$

$\Rightarrow$ (a) If $\overline{A}B + c\overline{D} = 0$

prove, $AB + \overline{c}(\overline{A} + \overline{D})$

$= AB + BD + \overline{B}\overline{D} + \overline{A}C\overline{D}$

~~(b)~~ ~~$\overline{ABC}+$~~ $\qquad$ $\cancel{R.H.S. = AB + BD + \overline{B}\overline{D} + \overline{A}C\overline{D}}$

$AB + \overline{c}(\overline{A} + \overline{D})$ $\qquad$ $\cancel{= AB + D(B + \overline{A}\overline{C}) + \overline{B}\overline{D}}$

$= AB + \overline{AC} + \overline{CD}$ $\qquad =$

$=$

condition: $\bar{A}B + C\bar{D} = 0.$

(1) L.H.S = $AB + \bar{C}(\bar{A}+\bar{D})$

$= AB + BD + \bar{B}\bar{D} + \bar{A}\bar{C}D + 0$

R.H.S : $AB + BD + \bar{B}.\bar{D} + \bar{A}\bar{C}D + \bar{A}\cdot A$

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

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

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

$= \bar{A}\bar{B}[C+\bar{C}] + \bar{A}B[\bar{C} + C] + A\bar{B}\bar{C}$

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

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

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

$\bar{A} + AX$

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

$\bar{A} + X$

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

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

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

Scanned by CamScanner

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

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

$= (\bar{A}B + AC)(\bar{\bar{B}} \cdot \overline{AC})$ [∵ Transposition]

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

$=$

⇒ Boolean function representation —

(1) SOP (sum of product)

e.g. $AB + BC$.

(2) POS (Product of sum)

e.g. $(A+B)\cdot(B+C)$.

(3) Standard SOP (canonical representation)

→ A function is called canonical if every term in function contain every variable which are going to be used in boolean function.

(4) Standard POS (canonical).

e.x.

A	B	C	Y	Minterms	Maxterms
0	0	0	0	$\bar{A}\,\bar{B}\,C$	$A+B+C$
0	0	1	1	$\bar{A}\bar{B}\,C$	$A+B+\bar{C}$
0	1	0	1		
0	1	1	0		
1	0	0	0		
1	0	1	1		
1	1	0	1		
1	1	1	0		

so, $\Sigma m(1,2,5,6)$ [Min terms]

$\Pi M(0,3,4,7)$ [Max terms].

⇒ Canonical SOP — (canonical rep. is used in K-map).

$$f(A,B,C) = \bar{A}B + \bar{B}C$$
$$= \bar{A}B(C+\bar{C}) + \bar{B}C(A+\bar{A})$$
$$= \bar{A}BC + \bar{A}B\bar{C} + \bar{B}CA + \bar{B}C\bar{A}.$$

⇒ canonical POS –

$$(A+\bar{B})(B+\bar{C})$$
$$= [(A+\bar{B}) + C\cdot\bar{C}]\,[(B+\bar{C})+(A\cdot\bar{A})]$$
$$= (A+\bar{B}+C)\cdot(A+\bar{B}+\bar{C})\cdot(B+\bar{C}+A)\cdot(B+\bar{C}+\bar{A})$$

Scanned by CamScanner

⇒ K-map —

(1) $f = A'C + A'B + AB'C + BC$

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

$= A'CB + A'CB' + \underline{A'BC} + A'BC' + AB'C$

$+ ABC + A'BC.$

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

$= \underline{C + A'BC'} = C + A'B.$ (RLR).

(2) $f = x'yz + x'yz' + xy'z' + xy'z.$

z \ xy	00	01	11	10
0	0	1	0	1
1	0	1	0	1

$x'y + xy'$

$=$

(3) $f(x,y,z) = \Sigma(0,2,4,5,6)$. [Min-terms]

$z \backslash xy$	00	01	11	10
0	1	0	0	1
1	0	1	1	1

$z \backslash xy$	00	01	11	10
0	1	0	0	1
1	1	1	0	1

$= \cancel{z+} \; \cancel{\bar{x}z + \bar{x}\bar{y}\bar{z} + \bar{x}y}$

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

$= \bar{x}z + \bar{y}[\underbrace{\bar{x}\bar{z} + x}_{(x+\bar{z})}]$

$= \bar{x}z + \bar{y}(x+\bar{z})$

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

$= \bar{x}z + \bar{y}(\overset{\bar{A}B + A}{\bar{x}\bar{z} + x})$

$= \bar{x}z + \bar{y}(x+\bar{z})$

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

$\bar{x}[\bar{y}\bar{z} + z] + x\bar{y}$

$= \bar{x}[z + \bar{y}] + x\bar{y}$

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

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

$\Rightarrow$ $f(A,B,C,D) = \Sigma\,(0,1,2,5,8,9,10)$.

SOP

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

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

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

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

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

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

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

$=$

	$C+D$	$C+\bar{D}$	$\bar{C}+\bar{D}$	$\bar{C}+D$
$A+B$				
$A+\bar{B}$	0			
$\bar{A}+\bar{B}$	0	0	0	0

⇒ Five Variable K-map —

$\bar{A}$ BC\DE	00	10	11	10
00	m_0	m_1	m_3	m_2
01	m_4	m_5	m_7	m_6
11	m_{12}	m_{13}	m_{15}	m_{14}
10	m_8	m_9	m_{11}	m_{10}

A				
	m_{16}	m_{17}	m_{19}	m_{18}
	m_{20}	m_{21}	m_{23}	m_{22}
	m_{28}	m_{29}	m_{31}	m_{30}
	m_{24}	m_{25}	m_{27}	m_{26}

$f = \Sigma m(0,2,3,10,11,12,13,16,17,18,19,20,21,26,27)$

→ Here we will put K-maps above each other and then we will make groups
⇒ And we will make a grouping for individual K-maps also
→ Overlapping of groups are not allowed.

Que $\Sigma m(0,1,4,5,6,13,14,15,22,24,25,28,29,30,31)$

$\bar{A}$

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

A

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

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

$= CD[B + \bar{B}\bar{E}] + AB\bar{D} + \bar{B}\bar{D}\bar{A} + \bar{A}BC\bar{D}E$

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

~~$= BC(D + \bar{D}\bar{A}E)$~~

⇒ Don't care combinations —

→ In BCD codes only 0 to 9 numbers are allowed. so, In k-map we have to m_{10} to m_{15} cells as 'X' don't care sign.

e.g.

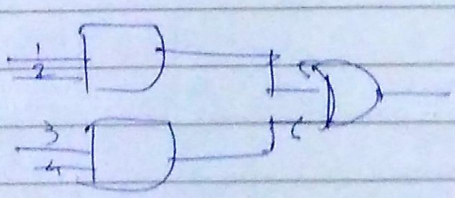
	1		
	1	1	
X	X	X	X
		X	X

e.g. $f(A,B,C,D) = \Sigma m(0,1,2,3,4,5) + d(10,11,\ldots 15)$

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

6 numbers

$\overline{A}\,\overline{B} + AB\overline{C}$ (6 inputs → Then called as Real minimal exp.)



(2) $f(A, B, C, D) = \Pi M(6, 7, 8, 9) \cdot d(10, 11, \ldots)$

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

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

$\cancel{= AB + \bar{C}B + AC}$

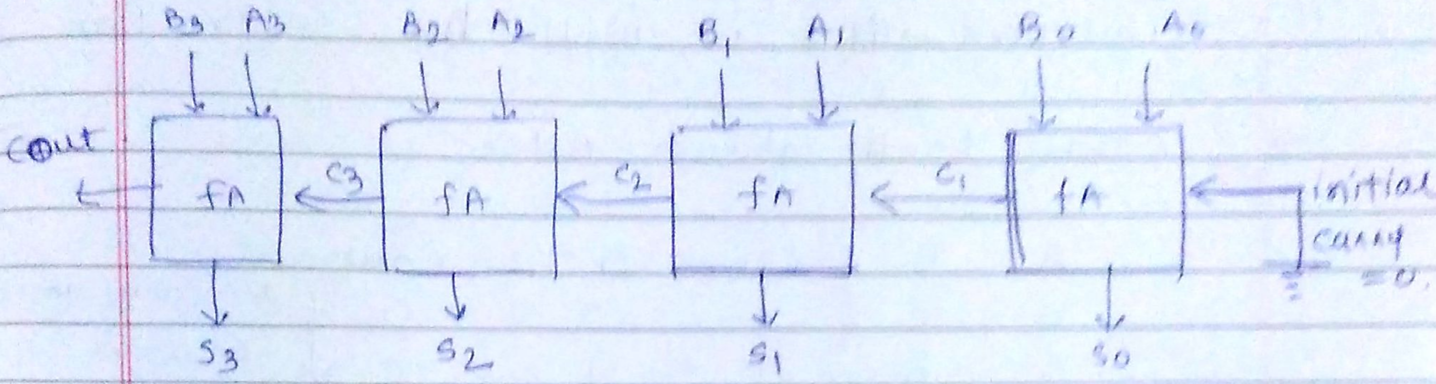
$AB + AC + \bar{C}B + D'$

Scanned by CamScanner

Binary Parallel Adder.

Half Adder, full Adder

Half subtractor, full subtractor.

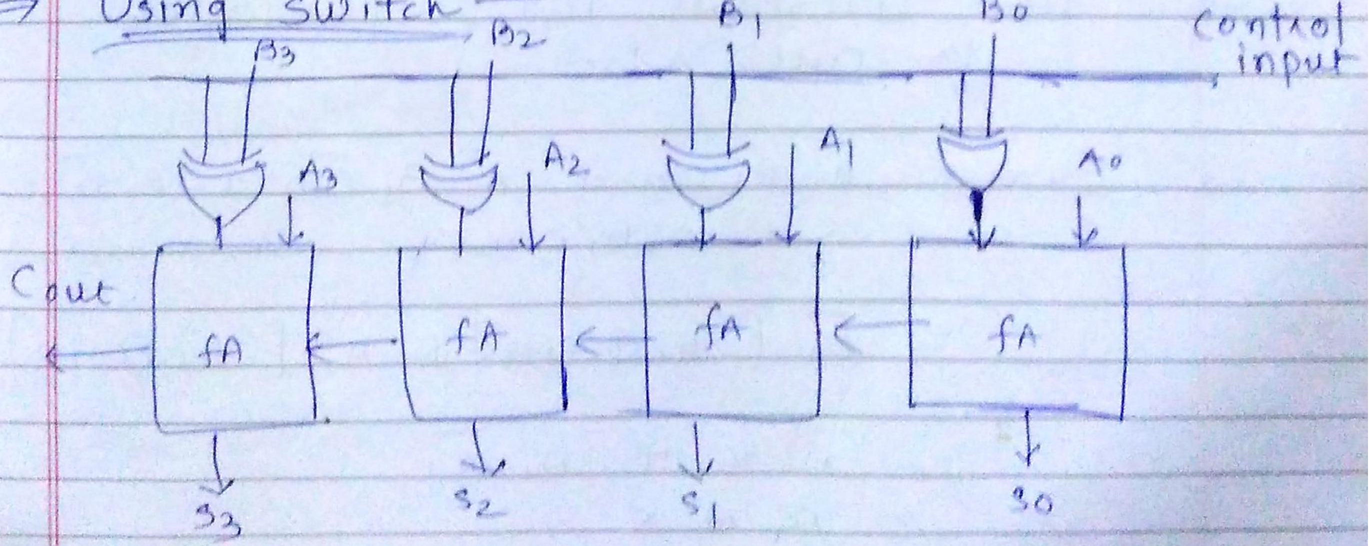


⇒ Disadvantage -

→ If any one full adder has some time delay then it would be multiply by n(no. of fA). so, n would be total time delay.

→ If we want to do subtraction then take invertion of B_0, B_1, B_2 and B_3 and take initial carry as One.

⇒ Using switch —



When control input = 0, it will act as full adder otherwise B input will be inverted and it will be subtraction.

⇒ Carry Look ahead Adder —

A	B	Cin	S	Cout	
0	0	0	0	0	No carry generation, Cout = 0.
0	0	1	1	0	
0	1	0	1	0	Carry propogation (P), Cout = Cin
0	1	1	0	1	
1	0	0	1	0	
1	0	1	0	1	
1	1	0	0	1	carry generation (G), cout = 1.
1	1	1	1	1	

→ Carry Generation $(G_i) = A_i B_i \overline{C_{in}} + A_i B_i C_{in} = A_i B_i \; (1) = A_i B_i$

Carry propogation $(P_i) = \overline{A_i} B_i + A_i \overline{B_i}$

$= (A_i \oplus B_i)$.

→ Here, (i) shows for a perticular ith Full Adder.

→ $Sum = \overline{A_i}\,\overline{B_i}\,C_{in} + \overline{A_i} B_i \overline{C_{in}} + A_i \overline{B_i}\,\overline{C_{in}} + A_i B_i C_{in}$

$= \overline{A_i} [B_i \oplus C_{in}] + A_i [B_i \odot C_{in}]$

$= \overline{A_i} X + A_i \overline{X}$

$= A_i \oplus X$

$\therefore Sum = A_i \oplus B_i \oplus C_{in}$.

→ $\cancel{C_{out} = A_i B_i + C_{in}(A_i \oplus B_i)}$

$\cancel{= \overline{\overline{A_i B_i}} \; \overline{\overline{C_{in}(A_i \oplus B_i)}}}$

→ $C_{out} = A_i B_i + C_{in}(A_i \oplus B_i)$

$\therefore C_{out} = G_i + C_{in} P_i$

so, $C_0 = G_0 + P_0 C_{(-1)}$ $\quad (\because [C_{in}]_{(i-1)}$ ahead carry).

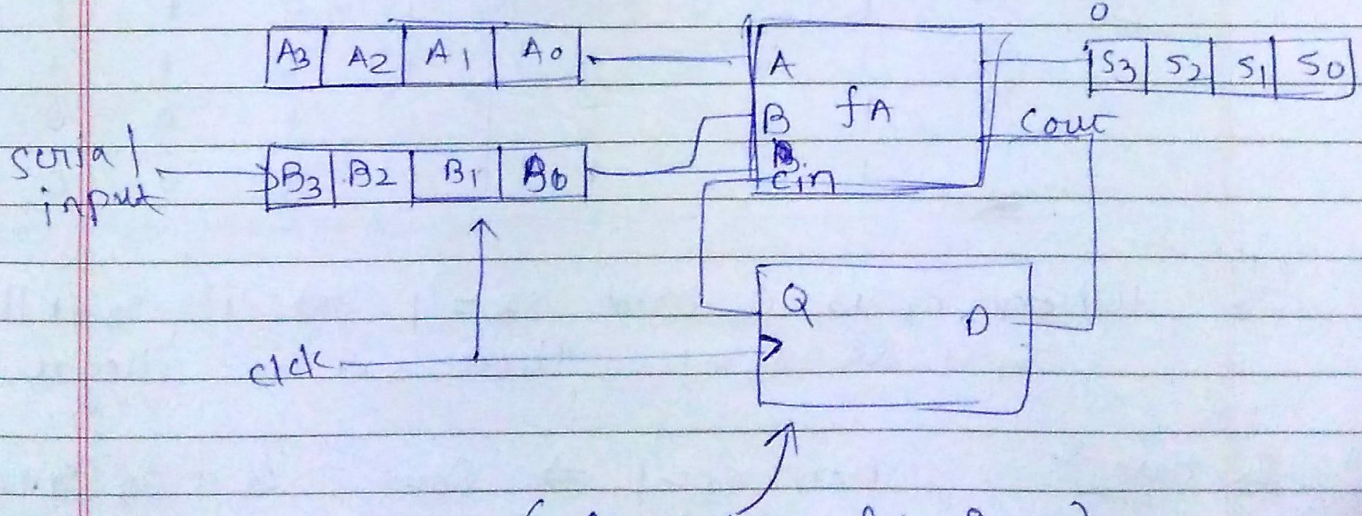
$C_1 = G_1 + P_1 C_0$

$= G_1 + P_1 [G_0 + P_0 C_{-1}]$.

→ Here, C_1 is dependent only on Cinput. so, propogation delay would n't be considered.

$(1) = \underline{A_i B_i}$

⇒ Serial Adder –



(As a delay flip-flop)

⇒ BCD Adder —

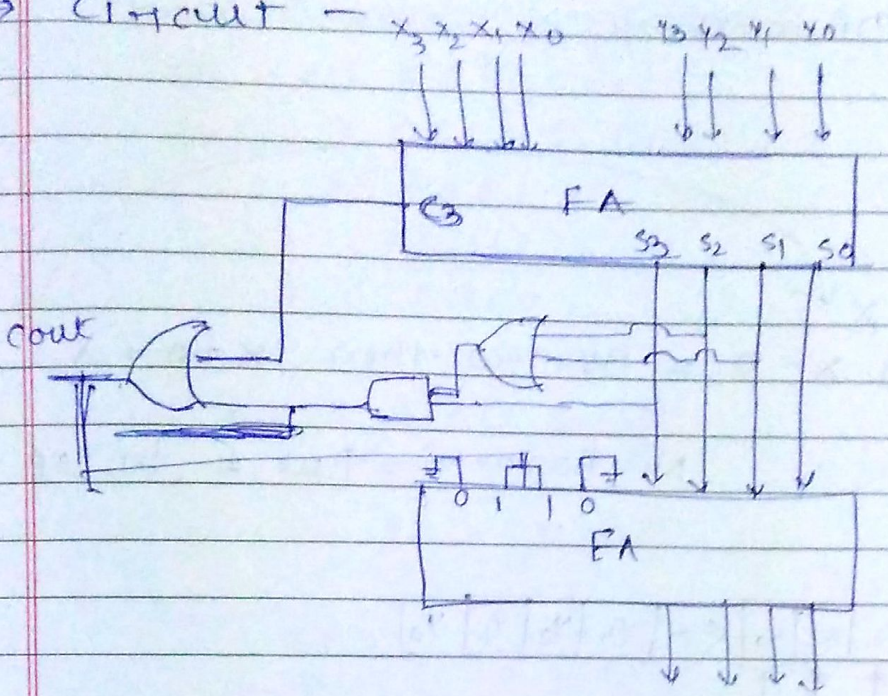
	C_3	S_3	S_2	S_1	S_0	C_{out}	S_3	S_2	S_1	S_0
	Uncorrected Sum					Corrected Sum				
0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	1	0	0	0	0	1
2	0	0	0	1	0	0	0	0	1	0
3	0	0	0	1	1	0	0	0	1	1
4										
5										
6										
7										
8										
9										
	C_3	S_3	S_2	S_1	S_0					
10	0	1	0	1	0	1	0	0	0	0
11	0	1	0	1	1	1	0	0	0	1
12	0	1	1	0	0	1	0	0	1	0
13	0	1	1	0	1	1	0	0	1	1
14	0	1	1	1	0	1	0	1	0	0
15	0	1	1	1	1	1	0	1	0	1
16	1	0	0	0	0	1	0	1	1	0
17	1	0	0	0	1	1	0	1	1	1
18	1	0	0	1	0	1	1	0	0	0
19	1	0	0	1	1	1	1	0	0	1

Case-1

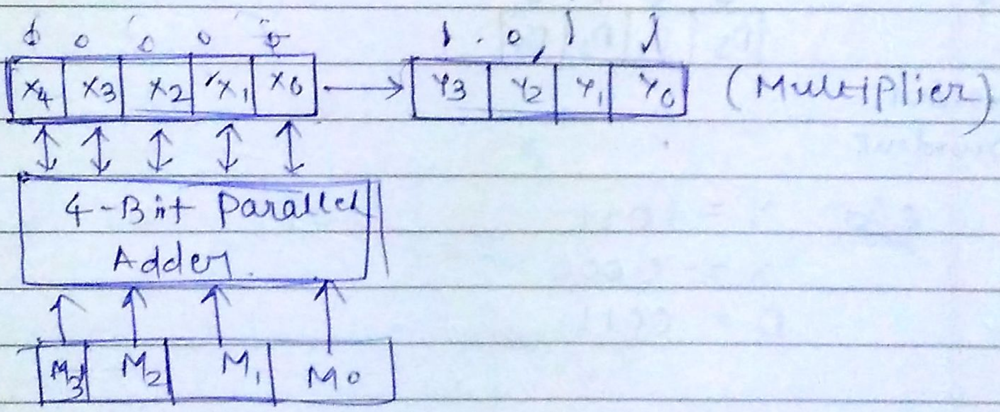
→ When C_3 is 0 and $S_3 = 1$ ⇒ if $S_2 = 1 \,||\, S_1 = 1 \,||$
$S_1 = 1$ && $S_2 = 1$ then it is illegal code

→ **Case-2** when $C_3 = 1$ ⇒ $C_{out} = C_3 + S_3(S_2 + S_1)$
then illegal code

⇒ Circuit —



⇒ Binary Multiplier —



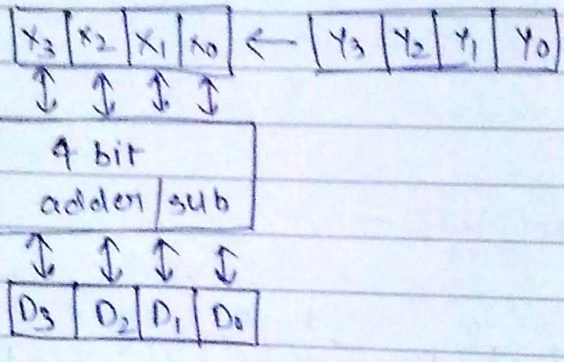
III initially x_0 to $x_4 = 0$.

⇒ Binary Division —

shift X

then $X - D$ = Borrow then $X + D = X$

No Borrow = put 1 in LSB of y



D) Y

D = Divider.

Y = Divident.

X = 0.

e.g.

$Y = 1011$

$X = 0000$

$D = 0011$

$X = 0000\ 1011$

←

$X = \boxed{0001}\ 0110$

$$
\begin{array}{rr}
 & {\scriptstyle 1\,1\,1\,0} \\
X - D = & 0001 \\
- & 0011 \\
\hline
1 & 1110
\end{array}
$$

. .

here Borrow is generated

so, $X = X + D$

$= 0001 + 0011$

$= 0001$

$+ 0011$

$X = 0100$

$XY = 0100\ 0110.$

$\leftarrow$ 2 shifts.

$= 100\boxed{0}\,1100$

0 1 1 1 0

1000

$-\ 0011$

0101 (NO BORROW)

$Y = \cancel{0111}\ 1101$

$XY = 1000\ 1101$

$\leftarrow$ 3 shifts

$= 0001\ \boxed{}\ 1011$

0001

$-\ 0011$

1 1110.

1 1

0001

$+\ 0011$

$X = 0100$

$XY = 0100\ 1011$

$\leftarrow$

$= 1001\ 0110$

⇒ BCD to binary conversion —

	BCD				Binary			
	B_3	B_2	B_1	B_0	B_{i3}	B_{i2}	B_{i1}	B_{i0}
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

B_3B_2 \ B_1B_0	00	01	11	10
00				
01				
11	x	x	x	x
10	1	1	x	x

$\therefore\ B_{i3} = B_3.$

Similarly $B_{i2} = B_2$

$B_{i1} = B_1$

$B_{i0} = B_0.$

exa. Design a circuit to detect decimal num 0, 1, 4, 6, 7, 8 in 4-bit XS-3 input.

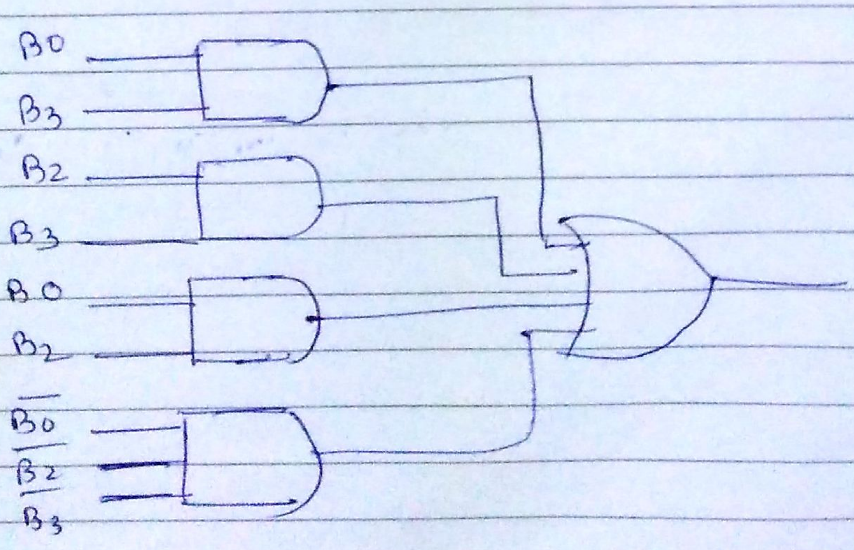
	$B_0\ B_1\ B_2\ B_3$	O/P
0	0 0 1 1	1
1	0 1 0 0	1
2	0 1 0 1	0
3	0 1 1 0	0
4	0 1 1 1	1
5	1 0 0 0	0
6	1 0 0 1	1
7	1 0 1 0	1
8	1 0 1 1	1
9	1 1 0 0	0

$f = \Sigma m(3, 4, 7, 9, 10, 11) + d(0, 1, 13, 14, 15)$

$B_0B_1 \backslash B_2B_3$	00	01	11	10
00	X	X	1	X
01	1		1	
10		X	X	X
11		1	1	1

~~$B_0(B_3 + B_2)$~~

$= B_2 B_3 + B_3 \underline{B_0} + \underline{B_0} B_2 + \overline{B_0}\,\overline{B_2}\,\overline{B_3}$



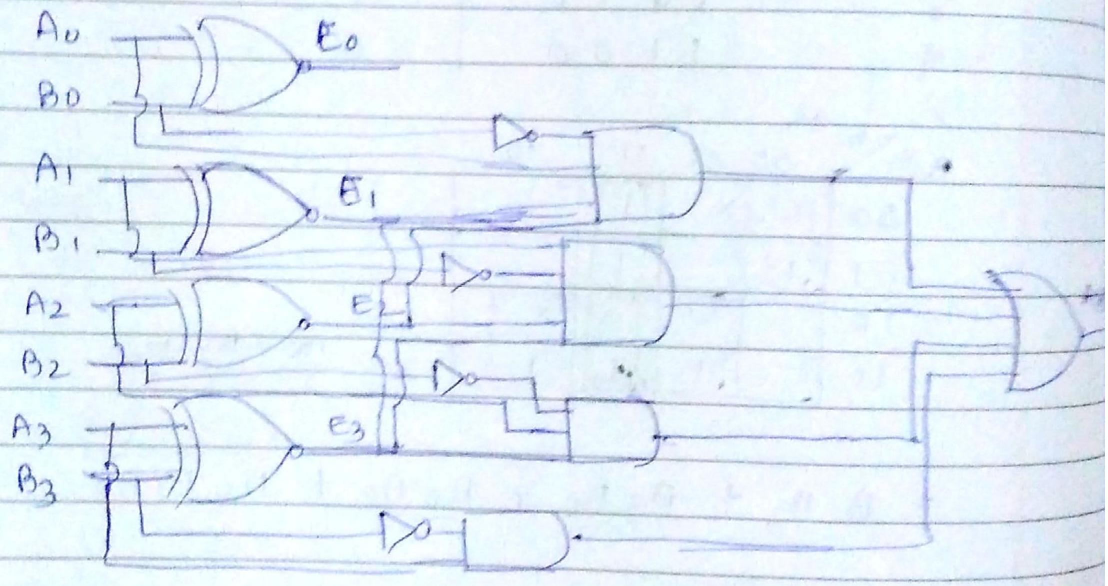
⇒ Comparator —

1-bit
2-bit
4-bit

⇒ 4-bit — $E_0 = A_0 \odot B_0$, $E_1 = A_1 \odot B_1$, $E_2 = A_2 \odot B_2$, E_3

$A > B$: $A_3\overline{B_3} + E_3(A_2\overline{B_2}) + E_3E_2(A_1\overline{B_1})$
$+ E_3E_2E_1(A_0\overline{B_0})$

$A < B$



⇒ Multiplexer –

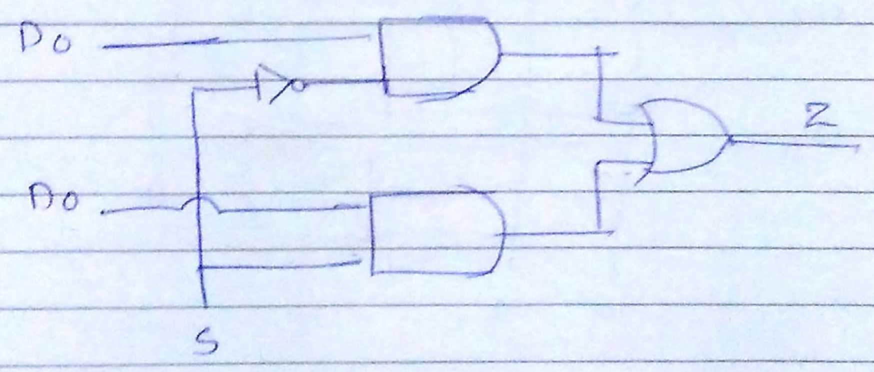
→ Two types : (1) Time division Multiplexer
(2) Frequency division Multiplexer

→ For whole time or frequency slop one input will be buffered and output will be taken.

e.g. 2X1 MUX.

S	OP (Z)
0	D_0
1	D_1

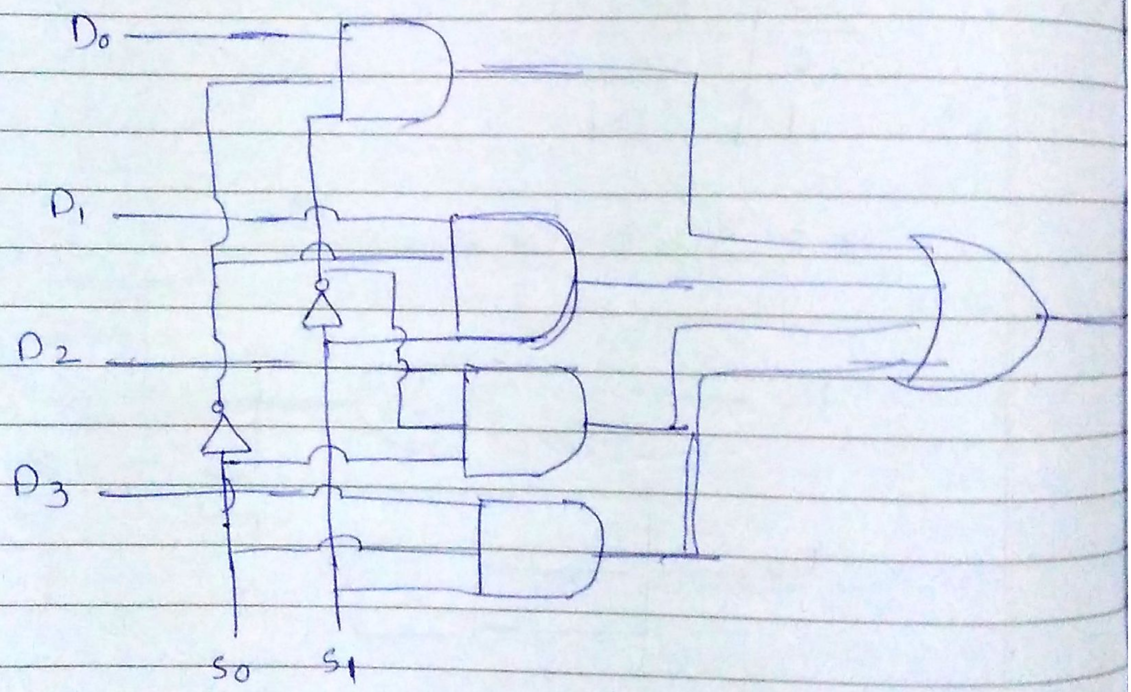
$z = \bar{S}D_0 + SD_1$.



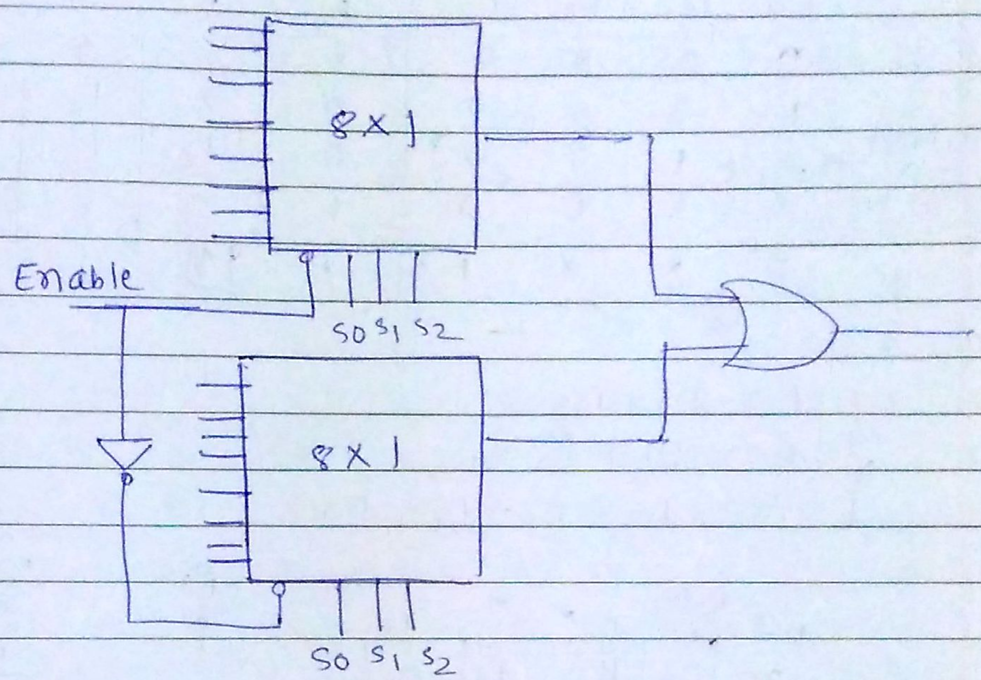
e.x. 4x1 MUX.

S_0	S_1	O/P (Z)
0	0	D_0
0	1	D_1
1	0	D_2
1	1	D_3

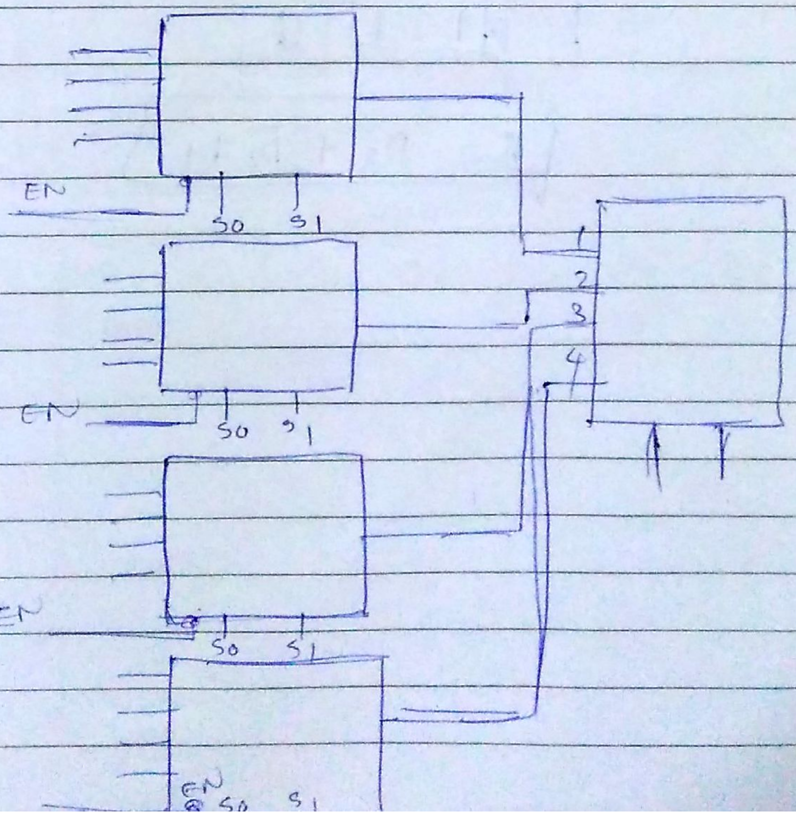
$$Z = \bar{S_0}\,\bar{S_1}\,D_0 + \bar{S_0}\,S_1\,D_1 + S_0\,\bar{S_1}\,D_2 + S_0 S_1 D_3$$



e.x. 16 x 1 Mux using 2 (8 x 1) —



⇒ 16 x 1 MUX using 4 x 1 MUX



00
01
10
11

⇒ Priority Encoder:-

Lowest Priority — Highest Priority

D_0	D_1	D_2	D_3	A	B	V	
0	0	0	0	X	X	0	← No key has been Pressed.
1	0	0	0	0	0	1	
X	1	0	0	0	$\boxed{1}$	1	
X	X	1	0	1	0	1	
X	X	X	1	1	$\boxed{1}$	1	

Key has been Pressed or not →

$$\begin{cases} A = D_2 + D_3 \\ B = D_3 + \overline{D_2}\, D_1 \\ V = D_3 + D_2 + D_1 + D_0 \end{cases}$$

B

$D_0D_1 \backslash D_2D_3$	00	01	11	10
00	0	1	1	0
01	1	1	1	0
11	1	1	1	0
10	0	1	1	0

1111
1001
0001
1101
1011
0011
0101
0111

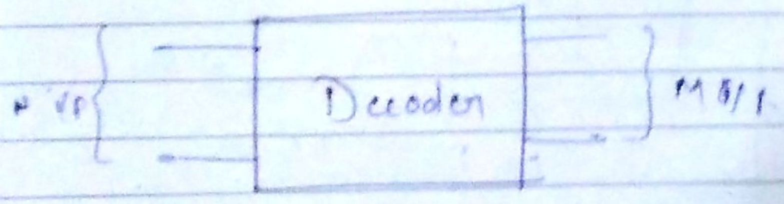
$$\boxed{B = D_3 + \overline{D_2}\, D_1}$$

⇒

⇒ Decimal to BCD Priority encoder with Active ~~low~~ high i/p & o/p

0	1	2	3	4	5	6	7	8	9	D_0	D_1	D_2	D_3	V
1	0	0	0	0	0	0	0	0	0	0	0	0	0	1
x	1	0	0	0	0	0	0	0	0	0	0	0	1	1
x	x	1	0	0	0	0	0	0	0	0	0	1	0	1
x	x	x	1	0	0	0	0	0	0	0	0	1	1	1
x	x	x	x	1	0	0	0	0	0	0	1	0	0	1
x	x	x	x	x	1	0	0	0	0	0	1	0	1	1
x	x	x	x	x	x	1	0	0	0	0	1	1	0	1
x	x	x	x	x	x	x	1	0	0	0	1	1	1	1
x	x	x	x	x	x	x	x	1	0	1	0	0	0	1
x	x	x	x	x	x	x	x	x	1	1	0	0	1	1

⇒ 3 to 8 line decoder :-



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
.				.						
.				.						

e.x. 2 to 4

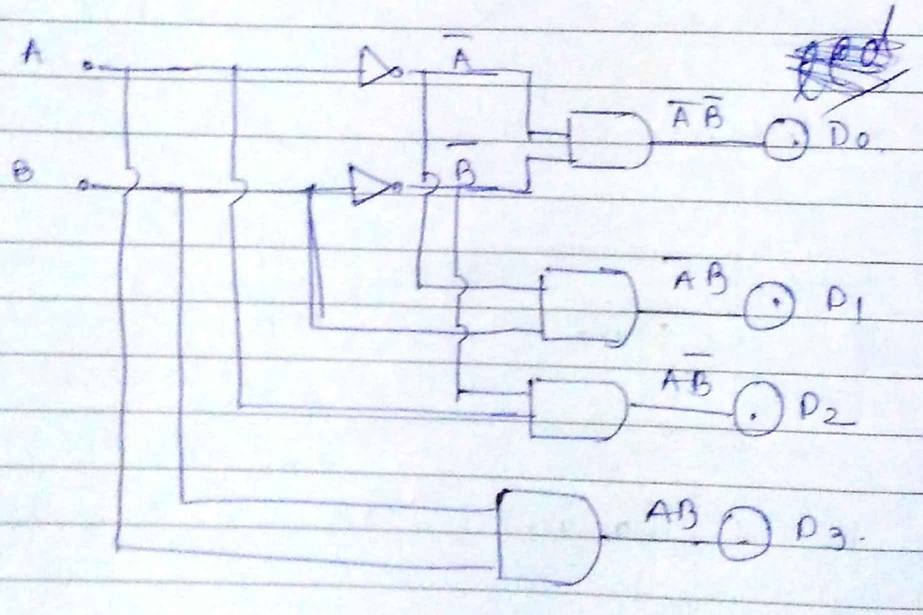
A	B	D_0	D_1	D_2	D_3
0	0	1	0	0	0
0	1	0	1	0	0
1	0	0	0	1	0
1	1	0	0	0	1

$D_0 = \bar{A}\bar{B}$.

$D_1 = \bar{A}B$.

$D_2 = A\bar{B}$

$D_3 = AB$.



⇒ BCD to seven Segment decoder –

D_0	D_1	D_2	D_3	a	b	c	d	e	f	g
0	0	0	0	1	1	1	1	1	1	0
0	0	0	1	0	1	1	0	0	0	0
0	0	1	0	1	1	0	1	1	0	1
0	0	1	1	1	1	1	1	0	0	1
0	1	0	0	0	1	1	0	0	1	1
0	1	0	1	1	0	1	1	0	1	1
0	1	1	0	1	0	1	1	1	1	1
0	1	1	1	1	1	1	0	0	0	0
1	0	0	0	1	1	1	1	1	1	1
1	0	0	1	1	1	1	1	0	1	1

D_0D_1 \ D_2D_3	00	01	11	10
00	1	0	1	1
01	0	1	1	1
11	X	X	X	X
10	1	1	X	X

$$a = D_2 + D_0 + D_1 D_3 + \bar{D_2}\bar{D_3}\bar{D_1}$$