2 17/1/22

Module 1

Result = carry * base + remainder

① $(AF2)_{16}$
$(35C)_{16}$
E4E

$14 = 0 \times 16 + 14$
$20 = 1 \times 16 + 4$
$14 = 0 \times 16 + 14$

A = 10
B = 11
C = 12
D = 13
E = 14
F = 15

highest digit in any number system = base - 1

② $(2FCD)_{16}$
$(3A2B)_{16}$
$(123A)_{16}$
7C32

$34 = 2 \times 16 + 2$
$19 = 1 \times 16 + 3$
$28 = 1 \times 16 + 12$
$7 = 0 \times 16 + 7$

HW
③ $(1F2E)_{16}$
$(2F31)_{16}$
$(3C2A)_{16}$
8A89

$25 = 1 \times 16 + 9$
$42 = 2 \times 16 + 10$

Binary Addition

① 01110
10110
10111
111011

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

HW
② 11101
10111
10101
1010001

Octal addition

① $\begin{array}{r} {}^{2}\,{}^{2}\,{}^{1}\;\; \\ 3461 \\ 2671 \\ 6767 \\ \hline 15341 \end{array}$

$20 = 2 \times 8 + 4$
$19 = 2 \times 8 + 3$
$13 = 1 \times 8 + 5$

HW ② $\begin{array}{r} {}^{2}\,{}^{2}\,{}^{1}\;\; \\ 2431 \\ 3763 \\ 7776 \\ \hline 16412 \end{array}$

Subtraction:

Decimal no. system

① $\begin{array}{r} {}^{2}\;{}^{10}\;\;\;\; \\ 7\,\cancel{3}\,\cancel{1}\,\cancel{2}\,12 \\ 1\,2\,9\,7 \\ \hline 6\,\cancel{2}0\,1\,5 \end{array}$

Hexadecimal subtraction

① $\begin{array}{r} E\;\; B{+}16\;\; C{+}16\;\; D{+}16 \\ \cancel{F}\;\cancel{C}\;\cancel{D}\;\cancel{D} \\ 2\,F\,E\,E \\ \hline C\,C\,E\,F \end{array}$

$13+16-14 = 15$
$12+16-14 = 14$
$11+16-15 = 12$

HW ② $\begin{array}{r} 2\;\; 1{+}16\;\; 0{+}16\;\; C{+}16 \\ \cancel{3}\;\cancel{2}\;\cancel{1}\;\cancel{C} \\ 1\,9\,5\,D \\ \hline 1\,6\,9\,F \end{array}$

Binary subtraction

① $\begin{array}{r} {}^{0\,\cancel{0}^2}\;\; \\ 1\,0\,\cancel{1}\,\cancel{1}\,\cancel{0}^{2} \\ -\,1\,0\,0\,1\,1 \\ \hline 0\,0\,0\,1\,1 \end{array}$

HW ② $\begin{array}{r} {}^{0\;\cancel{0}^2\;1\;1}\;\; \\ 1\,\cancel{1}\,\cancel{1}\,\cancel{0}\,\cancel{0}\,\cancel{0}^{2} \\ 1\,0\,1\,0\,1\,1 \\ \hline 0\,0\,1\,1\,0\,1 \end{array}$

Octal subtraction

①

$$
\begin{array}{r}
\overset{2}{\cancel{3}}\,\overset{5+8}{\cancel{6}}\,\overset{1+8}{\cancel{2}}\,\overset{4+8}{\cancel{5}}\,\overset{(1+8)\ 9}{\cancel{1}} \\
2\ 7\ 6\ 7\ 6 \\
\hline
0\ 6\ 3\ 5\ 3 \\
\hline
\end{array}
$$

HW ②

$$
\begin{array}{r}
\overset{3}{\cancel{4}}\,\overset{8}{\cancel{1}}\,\overset{9}{\cancel{2}}\,\overset{10}{\cancel{3}}\,\overset{9}{\cancel{1}} \\
2\ 6\ 7\ 4\ 6 \\
\hline
1\ 2\ 2\ 6\ 3 \\
\hline
\end{array}
$$

Multiplication :-

$$
\begin{array}{r}
(567)_8 \\
\times (36)_8 \\
\hline
4312 \\
2145\ \ \\
\hline
25762 \\
\hline
\end{array}
$$

$6\times7 = 42 = 5\times8+2$

$6\times6 = 36+5$

$= 41 = 5\times8+1$

$6\times5 = 30+5$

$= 35 = 4\times8+3$

$3\times7 = 21 = 2\times8+5$

$3\times6 = 18+2$

$20 = 2\times8+4$

$3\times5 = 15+2$

$= 17 = 2\times8+1$

Binary multiplication

①

$$
\begin{array}{r}
(10110)_2 \\
\times (110)_2 \\
\hline
00000 \\
10110\ \ \\
10110\ \ \ \ \\
\hline
(10000100)_2 \\
\hline
\end{array}
$$

HW ②

$$
\begin{array}{r}
(11010)_2 \\
\times (1101)_2 \\
\hline
110110 \\
000000\ \ \\
110110\ \ \ \ \\
110110\ \ \ \ \ \ \\
\hline
1010111110 \\
\hline
\end{array}
$$

Decimal to Binary Conversion

① $(231.25)_{10} \rightarrow (\ \ )_2$

$$
\begin{array}{l}
2\,|\,231 \\
2\,|\,115-1 \\
2\,|\,57-1 \\
2\,|\,28-1 \\
2\,|\,14-0 \\
2\,|\,7-0 \\
2\,|\,3-1 \\
2\,|\,1\ 0 \\
\ \ \ \ 0-1
\end{array}
$$

$(011100111)_2$

$0.25\times2 = 0.50 \rightarrow 0$

$0.5\times2 = 1.0 \rightarrow 1$

$0.0\times2 = 0$

$(0.01)_2$

$(231.25)_{10} \rightarrow (011100111.01)_2$

② $(612.4)_{10} \rightarrow (\ \ )_2$

$$\begin{array}{r|l} 2 & 612 \\ 2 & 306-0 \\ 2 & 153-0 \\ 2 & 76-1 \\ 2 & 38-0 \\ 2 & 19-0 \\ 2 & 9-1 \\ 2 & 4-1 \\ 2 & 2-0 \\ 2 & 1-0 \\ & 0-1 \end{array}$$

$(1001100100)_2$

$0.4 \times 2 = 0.8$ 0

$0.8 \times 2 = 1.6$ 1

$0.6 \times 2 = 1.2$ 1

$0.2 \times 2 = 0.4$ 0

(repeat again)

$(0.0110)_2$

$(612.4)_{10} \rightarrow (1001100100.0110)_2$

Decimal to Hexadecimal conversion

① $(19628.2)_{10} \rightarrow (\ \ )_{16}$

$$\begin{array}{r|l} 16 & 19628 \\ 16 & 1226-12 \\ 16 & 76-10 \\ 16 & 4-12 \\ & 0-4 \end{array}$$

$(4CAC)_{16}$

$0.2 \times 16 = 3.2$ 3

$0.2 \times 16 = 3.2$ 3

$(0.33)_{16}$

$(4CAC.33)_{16}$

HW ② $(76293.125)_{10} \rightarrow (\ \ )_{16}$

Decimal to Octal

① $(3952.4)_{10} \rightarrow (\ \ )_8$

$$\begin{array}{r|l} 8 & 3952 \\ 8 & 494-0 \\ 8 & 61-6 \\ & 7-5 \end{array}$$

$(7560)_8$

$0.4 \times 8 = 3.2$ 3

$0.2 \times 8 = 1.6$ 1

$0.6 \times 8 = 4.8$ 4

$0.8 \times 8 = 6.4$ 6

$0.4 \times 8 =$ (repeat)

$= (0.3146)_8$

$(7560.3146)_8$.

HW ② $(3ACB.AB)_{16} \rightarrow (\ \ )_{10}$.

HW ③ $(3765.65)_8 \rightarrow (\ \ )_{10}$

$(3ACB.AB)_{16}$
3 2 1 0 -1 -2

$= 3\times16^3 + A\times16^2 + C\times16 + B\times16^0$
$+ A\times16^{-1} + B\times16^{-2}$

HW
(4) $(4561.35)_8 \longrightarrow (\ )_{10}$

HW
(5) $(11011.1011)_2 \longrightarrow (\ )_{10}$

Hexadecimal to Binary to Octal

(1) $(5ACD.BB)_{16} \rightarrow (\ )_2 \rightarrow (\ )_8$

5 A C D . B B

$(0101\ 1010\ 1100\ 1101 . 1011\ 1011)_2$

0 101 101 011 001 101 . 101 110 110

$(5\ 5\ 3\ 1\ 5 . 5\ 6\ 6)_8$

HW
(2) $(3ABC.AC)_{16} \rightarrow (\ )_2 \rightarrow (\ )_8$

HW
(3) $(01101111011.101101)_2 \rightarrow (\ )_8 \rightarrow (\ )_{16} \rightarrow$ (pair up 4 digits each)

(pair up 3 digits each)

18/01/2022

(1) $(3716.563)_8 \rightarrow (\ )_2 \rightarrow (\ )_{16}$

3 7 1 6 . 5 6 3

. ~~1011 100~~

$(011\ 111\ 001\ 110 . 101\ 110\ 011)_2$

0111 1100 1110 . 1011 1001 1000 take extra zeroes

7 12 14 11 9 8

$(7CE.B98)_{16}$

(2) $(5ACD.BB)_{16} \rightarrow (\ )_2 \rightarrow (\ )_8$

5 A C D . B B

$(0101\ 1010\ 1100\ 1101 . 1011\ 1011)_2$

$0\underbrace{101}_{5}\underbrace{101}_{5}\underbrace{011}_{3}\underbrace{001}_{1}\underbrace{101}_{5}.\underbrace{101}_{5}\underbrace{110}_{6}\underbrace{110}_{6}$

$(55315.566)_8$

Convertions

① $(327)_8 \to (\ )_5$

$\to (\ )_{10} \to$

$\underset{8^2}{3}\underset{8^1}{2}\underset{8^0}{7} \Rightarrow 7+16+192 = (215)_{10}$

5 | 215
5 | 43 - 0
5 | 8 - 3
5 | 1 - 3
0 - 1

$= (1330)_5$

② $(12332)_4 \to (\ )_3$

$\underset{4^4}{1}\underset{4^3}{2}\underset{4^2}{3}\underset{4^1}{3}\underset{4^0}{2} = 2+12+48+128+256 = (446)_{10}$

3 | 446
3 | 148 - 2
3 | 49 - 1
3 | 16 - 1
3 | 5 - 1
1 - 2

$= (121112)_3$

HW

③ $(5621)_7 \to (\ )_9$

Complements:

1's & 2's, 3's & 4's, 5's & 6's, 7's & 8's, 9's & 10's, 11's & 12's, 13's & 14's, 15's & 16's complements.

1's & 2's

① $(110101)_2$

001010 → 1's complement
+1
001011 → 2's complement

② $(1011010)_2$

0100101 → 1's
+ 1
0100110 → 2's

7's & 8's

① $(67123)_8$ → 0 to 7.

$$\begin{array}{r} 77777 \\ -\ 67123 \\ \hline 10654 \end{array} \rightarrow 7\text{'s complement}$$

$$\begin{array}{r} +\ \ 1 \\ \hline 10655 \end{array} \rightarrow 8\text{'s complement}$$

HW

② $(235400)_8$ → 0 to 7

$$\begin{array}{r} 777777 \\ -\ 235400 \\ \hline 542377 \end{array} \rightarrow 7\text{'s complement}$$

$$\begin{array}{r} +\ \ 1 \\ \hline 542400 \end{array} \rightarrow 8\text{'s complement}$$

9's & 10's

① $(169700)_{10}$ → 0 to 9

$$\begin{array}{r} 999999 \\ 169700 \\ \hline 830299 \end{array} \rightarrow 9\text{'s complement}$$

$$\begin{array}{r} +\ \ 1 \\ \hline 830300 \end{array} \rightarrow 10\text{'s complement}$$

15's & 16's

① $(2BCDE)_{16}$ → 0 to 15

$$\begin{array}{ccccc} 15 & 15 & 15 & 15 & 15 \\ 2 & B & C & D & E \\ \hline D & 4 & 3 & 2 & 1 \end{array} \rightarrow 15\text{'s complement}$$

$$\begin{array}{r} +\ 1 \\ \hline D\ 4\ 3\ 2\ 2 \end{array} \rightarrow 16\text{'s complement}$$

Binary Subtraction using 1's complement

$A-B \rightarrow A+(-B) = y$

- In Addition if carry is 1, then add +1 with your y, Answer will be positive.

- If carry is 0, Answer is negative

① $(1011)_2 - (0101)_2$

A = 1011

B = 0101

↳ 1010 → 1's complement

y = A + (-B)

$$\begin{array}{r} 1011 \\ 1010 \\ \hline ①\,0101 \\ +\ 1 \\ \hline 0110 \\ \hline \end{array}$$

carry (+ve)

Final answer = 0110

② $(0110)_2 - (1010)_2$

A = 0110

B = 1010 ↳ 0101

$$\begin{array}{r} 0110 \\ 0101 \\ \hline 1011 \\ \hline \end{array}$$

no carry (-ve)

Final answer = 1's complement of 1011

= -(0100)

HW ③ $(1110)_2 - (0110)_2$

HW ④ $(0111)_2 - (1001)_2$

Binary subtraction using 2's complement

A - B = y = A + (-B)

↳ 2's complement

Carry is 1 → +ve number & neglect carry

Carry is 0 → -ve number

A = 1001

B = 0101 → 1011 (2's complement)

$$\begin{array}{r} 1001 \\ 1011 \\ \hline ①\,0100 \end{array}$$

carry (+ve)

Final answer = 0100.

② $(0110)_2 - (1010)_2$

A = 0110

B = 1010 → 0110 (2's complement)

$$\begin{array}{r} 0110 \\ 0110 \\ \hline 1100 \\ \hline \end{array}$$ no carry (-ve)

Final answer = 2's complement of 1100

$$= \begin{array}{r} 0011 \\ 1 \\ \hline 0100 \\ \hline \end{array}$$

HW ③ $(1000)_2 - (1010)_2$

HW ④ $(1101)_2 - (0111)_2$

19/1/22

Classification of codes :

1) Weighted codes :

Each position is having fixed value in weighted codes.

Ex:- Binary, 8421, 2421

$$\begin{array}{cccc} a_3 & a_2 & a_1 & a_0 \\ \uparrow & \uparrow & \uparrow & \uparrow \\ 8 & 4 & 2 & 1 \end{array} \qquad \begin{array}{cccc} b_3 & b_2 & b_1 & b_0 \\ | & | & | & | \\ 2 & 4 & 2 & 1 \end{array}$$

Position is defined.

2) Non-weighted codes :

It doesn't have any fixed value

Ex:- Gray code, XS-3

3) Reflective code :

Self complementary code

Ex:- 9 complement is 0

8 complement is 1

7 complement is 2

~~8~~ 6 complement is 3

5 complement is 4

Decimal 8421

0 →	0000
1 →	0001
2 →	0010
3 →	0011
4 →	0100
5 →	1011
6 →	1100
7 →	1101
8 →	1110
9 →	1111

5 can also be written as
2421
0101
4+1=5

4) Sequential code:

Next data will be incremented by 1

Ex:- Binary, XS-3, 8421.

5) Alpha numeric code;

ASCII

American standard code for Information Interchange

6) Error correcting code or Detecting code:

Hamming code, cyclic code

BCD (Binary Code Decimal)

In this case, each decimal digit are seperated by 4 digits bits binary code

↳ 4 bits = nibble

- As in decimal digit are ranging from 0 to 9, we will represent 1 nibble by BCD code.

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

BCD code of decimal digit

Conversion of BCD from decimal

1) $(149)_{10} \rightarrow (\ )_{BCD}$

$$\begin{array}{ccc} 1 & 4 & 9 \\ 0001 & 0100 & 1001 \end{array}$$

$(0001\ 0100\ 1001)_{BCD}$

2) $(638)_{10} \rightarrow (0110\ 0011\ 1000)_{BCD}$

3) $(378)_{10} \rightarrow (0011\ 0111\ 1000)_{BCD}$

Conversion of BCD to decimal

① $(10100)_{BCD} \rightarrow (\ )_{10}$

extra zeroes $\underbrace{\underline{00}10}\ \underbrace{1000}$

$(1\ 4)_{10}$

② $(1000100101)_{BCD}$

$= \underbrace{0010}\ \underbrace{0010}\ \underbrace{0101}$

$= (225)_{10}$

Comparison of Binary & BCD code

Decimal	Binary	BCD
$(15)_{10}$	$(1111)_2$	$(0001\ 0101)_{BCD}$

BCD is less efficient than binary

more bits are required to represent BCD

Homework

① Represent the following decimal into BCD

1) $(37)_{10}$

2) $(271)_{10}$

3) $(141)_{10}$.

2) Represent following BCD into decimal

① 010011

② 10110110110101

③ 1011010110.

BCD Addition

Rules :-

1) If nibble to nibble carry is 1 then add 0110 to that nibble.

2) If addition of nibble with nibble is greater than >9 then add 0110 to the nibble.

① $(8)_{10} + (7)_{10} = (?)$

```
  1000
  0111
  ----
  1111 >9
+ 0110
  ----
 10101
extra 0001 0101 = (15)10
        1    5
```

② $(9)_{10} + (8)_{10} = (?)_{10}$

```
   1001
   1000
   ----
 ① 0001
Carry 0110
   ----
  1 0111
extra 0001 0111 = (17)10
        1    7
```

③ $(89)_{10} + (79)_{10} = (?)_{10}$

```
   1000 1001
   0111 1001
   ---------
  10000 0010
   0110 0110
   ---------
extra
  0001 0110 1000        => (168)10.
```

(4) $(279)_{10} + (789)_{10} = (?)_{10}$

```
       0010 0111 1001
       0111 1000 1001
      ----------------
       1010 0000 0010
       0110 0110 0110
extra ----------------
  000 1 0000 0110 1000
  1     0    6    8
```

$(1068)_{10}$

(5) 279
+ 781

```
 0010 0111 1001
 0111 1000 0001
----------------
 1001 1111 1010
  9    15   10

 1001 1111 1010
 0000 0110 0110
----------------
 1 01 0010 0000
  10   6    0
```

106

if second time we got > 9 then we don't need to consider nibble to nibble carry

(6) $(189)_{10} + (265)_{10}$

(7) $(742)_{10} + (688)_{10}$

```
 0111 0100 0010
 0110 1000 1000
----------------
 1101 1100 1010
 0110 0110 0110
----------------
 1 0100 0011 0000
 1  4    3    0
```

```
 0001 1000 1001
 0010 0110 0101
----------------
 0011 1110 1110
      0110 0110
----------------
 0100 0101 0100
  4    5    4
```

Binary to BCD conversion

For any number system to BCD conversion, First we had to convert given no.system into decimal then we convert decimal system into BCD code

Binary $\rightarrow (\ )_{10} \rightarrow$ BCD

① $(110011001)_2 \rightarrow (\ )_{BCD}$

$$\begin{array}{ccccccccc} 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 \\ 2^8 & 2^7 & & & 2^4 & 2^3 & & & 2^0 \\ 256 & 128 & & & 16 & 8 & & & 1 \end{array} = (409)_{10} = (0100\ 0000\ 1001)_{BCD}$$

② (11011101100.0101)

$$11011101100.0101 = (1772.3125)_{10}$$

$2^{10}\ 2^9\ 2^7\ 2^6\ 2^5\ 2^3\ 2^2\ \ 2^{-2}\ 2^{-4}$

$\downarrow$

$(0001\ 0111\ 0111\ 0010.0011\ 0001\ 0010\ 0101)_2$

③ $(1101101)_2$
④ $(101010110)_2$
⑤ $(762.6)_8$

} HW

BCD to binary

$\rightarrow (\ )_{10} \rightarrow$

① $\underbrace{0010}_{2}\ \underbrace{1001}_{9}\ \underbrace{0010}_{2} = (292)_{10}$

$$\begin{array}{l|l} 2 & 292 \\ 2 & 146-0 \\ 2 & 73-0 \\ 2 & 36-1 \\ 2 & 18-0 \\ 2 & 9-0 \\ 2 & 4-1 \\ 2 & 2-0 \\ & 1-0 \end{array} = (100100100)_2$$

② $\underbrace{0101}_{5}\ \underbrace{1000}_{8}\ \underbrace{1001}_{9}$

$$\begin{array}{l|l} 2 & 589 \\ 2 & 294-1 \\ 2 & 147-0 \\ 2 & 73-1 \\ 2 & 36-1 \\ 2 & 18-0 \\ 2 & 9-0 \\ 2 & 4-1 \end{array} \quad \begin{array}{l|l} 2 & 2-0 \\ & 1-0 \end{array}$$

$(1001001101)_2$

(3) $(1010110)_{BCD} \rightarrow (\ )_2$ } HW
(4) $(10110100)_{BCD} \rightarrow (\ )_2$

21/1/22

2421 BCD code
↳ weightage code

- BCD code { }
- Sometimes it is referred as 2^*-4-2-1 code

we have to give higher priority to lower bits and lower priority to higher one

Decimal	2^*-4-2-1 code
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1110
9	1111

Not self complementary

	2421
0	0000
1	0001
2	0010
3	0011
4	0100
5	1011
6	1100
7	1101
8	1110
9	1111

→ Self complementary

Here after 4 we are giving higher priority to higher bits

We are writing the bits inorder to satisfy the assumptions

BCD codes

7421 5421 3321 $84\bar{2}\bar{1}$ $74\bar{2}\bar{1}$

which is self complementing and which is not

$\begin{matrix} 7 & 4 & 2 & 1 \\ | & | & | & | \\ w_0 & w_1 & w_2 & w_3 \end{matrix}$

$w_0 + w_1 + w_2 + w_3 = 9$ then self complementary

$7+4+2+1 = 14 \neq 9$

$3+3+2+1 = 9$

$8+4-2-1 = 9$

$7+4-2-1 = 8 \neq 9$

$5+4+2+1 = 12 \neq 9$

3321, $84\bar{2}\bar{1}$ are self complementary

7421, 5421, $74\bar{2}\bar{1}$ are not self complementary

H.W

① Convert $(145)_{10}$ to 2421 code $= (0001\ 0100\ 1011)_{2421}$

② $(0100\ 0100\ 1110)_{2421}$ to $(\ )_{10}$ $= (4\ 4\ 8)_{10}$

Excess code (Xs^3)

To have a basic conversion of Decimal to Xs^3

1) Decimal No $\rightarrow$ $\overset{8421}{BCD}$ $\xrightarrow[0011]{Add}$ Xs^3

Eg: $4 \rightarrow 0100 \xrightarrow[0011]{+} 0111$ Xs^3

Decimal No	8421 (binary)	Xs^3
0	0000	0: 0011 3
1	0001	1: 0100 4
2	0010	2 0101 5
3	0011	3 0110 6
4	0100	4 0111 = 7
5	0101	5: 1000 = 8
6	0110	6: 1001 = 9
7	0111	7: 1010 = 10
8	1000	8: ~~1000~~ 1011 = 11
9	1001	9: 1100 = 12

Xs^3 → always unweighted

→ Self complementary.

$(66)_{Xs^3} \xrightarrow[+3]{+3} (99)_{10}$

Conversion of decimal no. in Xs^3 code

① $(36)_{10}$ — $(\overset{8421}{0011\ 0110})_{8421}$ $\xrightarrow[0011]{Add}$ $= (0110\ 1001)_{Xs^3}$

```
  0011 0110
  0011 0011
  ---------
  0110 1001
```

② $(395)_{10}$ - 0011 1001 0101
+0011 0011 0011
0110 1100 1000

HW

1) $(27)_{10}$
2) $(452)_{10}$
3) $(110101 1)_2$

} convert to $X S^3$ code

Ex3 addition (Ex^3) Ex^3 & XS^3 both are same excess code

1. Convert given no into Ex^3
2. Add given Ex^3 no.
3. See if there is the carry from nibble-nible

CY=1 - Add 0011
CY=0 - subtract 0011

① $(3)_{10} + (6)_{10}$

Step1

$(3)_{10} \xrightarrow{8421} 0011 \xrightarrow[0011]{Add} 0110$

$(6)_{10} \xrightarrow{8421} 0110 \xrightarrow[0011]{} 1001$

Step2

0110
1001
1111

no carry subtract 0011

1111
0011
1100

② $(38)_{10} + (28)_{10}$

$(38)_{10} \rightarrow (0011\ 1000)_{8421} \rightarrow (0110\ 1011)_{XS^3}$

$(28)_{10} \rightarrow (0010\ 1000)_{8421} \rightarrow (0101\ 1011)_{XS^3}$

no carry

0110 1011
0101 1011
1100 0110
(-) 0011 0011
1001 1001

Identification of self complementary code

9 — 0
8 → 1
7 — 2
6 — 3
5 — 4

So, if weighted code is given $(w_3 w_2 w_1 w_0)$

$w_3 + w_2 + w_1 + w_0 = 9$ ✓ self complementary

HW check by normal process (by writing codes)

7421 : 7+4+2+1 = 14 ≠ 9
5921 = 5+9+2+1 = 17 ≠ 9
3321 = 3+3+2+1 = 9
$84\bar{2}\bar{1}$ = 8+4−2−1 = 9
$74\bar{2}\bar{1}$ = 7+4−2−1 = 8 ≠ 9

ASCII code

American standard code for Information Interchange.
- It is 7 bits code
- total 127 different characters we represent by ASCII

→ 0 to 126

IFS Question

35 → #
36 → \$
37 → %
38 → &
48-57 → 0 to 9 {48 = 0, 49 = 1, 50 = 2, ... 57 = 9}
65-90 → A to Z {65 = A & 90 = Z}
97-122 → (a to z) {a = 97 = a, b = 98, ... z = 122}
126 → ~

Q: (1001101 1000111 1100111) → 103

77, 71

Find its ASCII code

GMg M, G, g

③ B C 4

B = 66 — 1000010

C = 67 — 1000011

4 = 52 — ~~0110011~~ 110100

HW

1) 100111 001111 110111 → convert to ASCII code

2) ASCII — 'Bb48'
↳ binary?

Gray code

- Frank Gray
- unweighted code (age)
- reflected binary code
- unit distance code
- It is cyclic code

1 bit GC

b_0
0
1

2 bit GC

b_1	b_0
0	0
0	1
1	1
1	0

→ Reflective Mirror

3 bit GC

$b_2\ b_1\ b_0$ → $1+2+4$ = 7 ⑨ 0 to 8

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

2×4 / 8

just last b_2 line will be complemented

4 bit

$b_3\ b_2\ b_1\ b_0$ — 8+4+2+1 ⑮ 0 to 15

b_3	b_2	b_1	b_0
0	0	0	0
0	0	0	1
0	0	1	1
0	0	1	0
0	1	1	0
0	1	1	1
0	1	0	1
0	1	0	0

RM

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

RM

8+8 = 16 — RM

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

Binary to GC

① take MSB as it is

② XOR MSB with next bit

③ Repeat ②

$b_3\ b_2\ b_1\ b_0$

$G_3\ G_2\ G_1\ G_0$

$G_3 = b_3$

$G_2 = b_3 \oplus b_2$

$G_1 = b_2 \oplus b_1$

$G_0 = b_1 \oplus b_0$

XOR → even no. of one's = 0.

Convert $(1001100)_2$ to GC

(10010100)

1 1 0 1 0 1 0

$1 \oplus 0 = 1$

$1 \oplus 1 = 0$

② $(0100110101)_2$ to GC

0 1 1 0 1 0 1 1 1 1

HW

1) $(1011010)_2$ – GC

2) $(346)_6$ to GC.

24/1/22

Gray to Binary Conversion

Step 1 - Take MSB as it is

Step 2 - XOR MSB to next bit by Gray code

Step 3 - Repeat step 2

$g_3\ g_2\ g_1\ g_0$

$B_3\ B_2\ B_1\ B_0$

$B_2 = B_3 \oplus g_2$

$= g_3 \oplus g_2$

$B_1 = B_2 \oplus g_1$

$= g_3 \oplus g_2 \oplus g_1$

$B_0 = B_1 \oplus g_0$

$= g_3 \oplus g_2 \oplus g_1 \oplus g_0$

Convert 01101011 Gray code into binary

0 1 1 0 1 0 1 1

$(0\ 1\ 0\ 0\ 1\ 1\ 0\ 1)_2$

② 10110110 → GC → B

1 0 1 1 0 1 1 0

1 1 0 1 1 0 1 1

Identification of Base and Radix of a given number

① $52/4 = 12$

$[5\times b^1 + 2\times b^0] = [1\times b^1 + 2\times b^0][4\times b^0]$

$5b^1 \cancel{+2b^0}$ $(5b^1+2) = (b^1+2)4$

$\boxed{b^1 = 6}$

② $24+17=40$

$[2\times b^1 + 4\times b^0] + [1\times b^1 + 7\times b^0] = [4\times b^1 + 0\times b^0]$

$(2b^1+4)+(b^1+7) = 4b^1$

$2b^{12}+18b^1+28 = 4b^1$

$2b^{12}+14b^1+28=0$

$b^{12}+7b^1+14=0$

$b^1 = 11$

$3b^1+11=4b^1$

$\boxed{11=b^1}$

③ $\sqrt{22} = 6$

$\sqrt{2\times b^1 + 2\times b^0} = 6\times b^0$

$2b^1+2 = 36$

$b^1 = \cancel{12}\ 17$

we have to take just real part / not decimal part

Total bits required to represent a number

1) How many bits required to represent 12 digit octal number.

$2^n \geq b^x$ 12.

$n \rightarrow$ no. of ~~di~~ bits required

$b \rightarrow$ base of number

$x \rightarrow$ power (no. of digits)

$\log 2^n \geq \log b^n\ (8^{12})$

$n\log 2 \geq 12\log 8$

$n \geq 12\left(\frac{\log 8}{\log 2}\right)$

$\boxed{n \geq 36.}$

(2) 8 bit Hexadecimal

$$2^n \geq 16^8$$
$$n \log 2 \geq 8 \log(16)$$
$$n \geq 8 \times 4$$
$$\boxed{n \geq 32}$$

(3) 15 digit decimal number

$$2^n \geq 10^{15}$$
$$n \log 2 \geq 15 \log 10$$
$$n \geq 15 \left(\frac{\log 10}{\log 2}\right)$$
$$n \geq 49.82 \simeq 50.$$

(4) How many bits required to represent $(2679)_{10}$.

~~$2^n \geq 10^4 \Rightarrow$ till 9999~~ but here number is specified.

$$\cancel{n \geq 4 \frac{\log 10}{\log 2}}$$
$$\cancel{n \geq 13.28, n =}$$

$$2^n \geq 2679$$
$$n \log 2 \geq \log 2679$$
$$\text{lor } n \geq \frac{\log(2679)}{\log 2}$$
$$11.38 \simeq 12$$

For base 8 till 7777. (4 digits)

no base is involved here

<u>HW</u> How many bits are required to represent

1) 10 digit decimal
2) 11 digit with base 5
3) $(129)_{12}$

Minimum <u>Decimal</u> <u>Equivalent</u>

(1) Find out the minimum decimal equivalent of $(D17)_x$

Base = Max Digit + 1
$= 13 + 1 = 14$

Minimum decimal equivalent will be:

$$\overset{2\,1\,0}{D17}$$
$$= D \times 14^2 + 1 \times 14^1 + 7 \times 14^0$$
$$= 2548 + 14 + 7$$
$$= (2569)_{10}$$

2) Find minimal decimal equivalent of $(731)_x$

Base = 8

$$\begin{matrix} 7 & 3 & 1 \\ 8^2 & 8^1 & 8^0 \end{matrix}$$

$= 7\times8^2+3\times8+1$

$= 448+24+1$

$= (473)_{10}$

3) $(3241)_x$

Base = 5

$= 3\times5^3+2\times5^2+4\times5+1$

$= (446)_{10}$

HW Find minimal decimal equivalent

1) $(3162)_x$

2) $(G6)_x$ ⟶ Gate 2018 G = 16

3) $(11)_x$.

Hamming Code

- This code was given by RW Hamming code
- Error correcting code, - It is used to detect and correct error
- In Hamming code we send data along with parity bits or Redundant bit
- Its representation is (n,k) code — n: total bits, k: message bits
- Parity bits ⇒ $p = n-k$
- To identify Parity bits, It should satisfy $\underline{2^P \geq P+K+1}$

For k=4, message bits, we have to identify, what will be no. of Parity bits.

$2^P \geq P+K+1$

$\boxed{2^P \geq P+5}$

$2^3 \geq 3+5 \Rightarrow 8=8$ ✓ $\boxed{P=3}$

P=3, k=4

$n = 3+4 = 7$

code = (7,4)

7	6	5	4	3	2	1
D_7	D_6	D_5	P_4	D_3	P_2	P_1
			2^2		2^1	2^0

~~$P_1 = 2^0 = 1$~~
~~$P_2 = 2^{2} = 1$~~

$P_1 = 2^0 = 1$
$P_2 = 2^1 = 2$ $(p^0 p^1 p^2)$
$P_4 = 2^2 = 4$

$P_1 \rightarrow$ check 1 bit & skip 1 bit $D_3 D_5 D_7$ (XOR)
$P_2 \rightarrow D_3 D_6 D_7$ (XOR)
$P_4 \rightarrow D_5 D_6 D_7$ (XOR).

Generation of Hamming code

① 5 bits data 01101 is given, Represent given data in hamming code

$k = 5$ bits
$2^P \geq P + k + 1$
$2^P \geq P + 6$
$2^4 \geq 4 + 6$ ✓ $P = 4$ bits

$n = 5 + 4 = 9$ bits
code - (9, 5) code

9	8	7	6	5	4	3	2	1
D_9	P_8	D_7	D_6	D_5	P_4	D_3	P_2	P_1
0		1	1	0		1		

$P_1 = 2^0 = 1$
$P_2 = 2^1 = 2$
$P_4 = 2^2 = 4$
$P_8 = 2^3 = 8$

$P_1 = D_3 D_5 D_7 D_9$ - ~~$0 \oplus 1 \oplus 0 \oplus 1$~~ $= 0$
$P_2 = D_3 D_5 D_7 D_9$ - ~~$1 \oplus 0 \oplus 0$~~ $= 1$
$P_4 = D_5 D_6 D_7$ - $0 \oplus 1 \oplus 1 = 0$
$P_8 = D_9$ - 0.

Final answer ← | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 0 |

$D_9\ D_7\ D_6\ D_5\ D_3$
0 1 1 0 1

$P_1 = D_3 \oplus D_5 \oplus D_7 \oplus D_9$
$= 1 \oplus 0 \oplus 1 \oplus 0$
$= 0$

$P_2 = D_3 \oplus D_6 \oplus D_7$
$= 1 \oplus 1 \oplus 1$
$= 1$

$P_4 = D_5 \oplus D_6 \oplus D_7 = 0 \oplus 1 \oplus 1$

check skip check skip check skip check skip check

D_9	P_8	D_7	D_6	D_5	P_4	D_3	P_2	P_1

skip check skip check

0 ← $P_1 = D_3 D_5 D_7 D_9$
1 ← $P_2 = D_3 D_6 D_7$
0 ← $P_4 = D_5 D_6 D_7$

0 ← $P_8 = D_9$

HW

(2). 6 digit (bits) data 001101,
Represent the given data in HC.

25/1/22

Hamming code error detection and error correction

- If received hamming code is 1110101 with even parity then detect and correct error.

3 parity bits – $P_1 = 2^0 = 1$
$P_2 = 2^1 = 2$
$P_4 = 2^2 = 4$

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

For P_1 association {check 1 bit and skip 1 bit}

$P_1 = D_3 D_5 D_7$ (XOR)

For P_2 association {check 2 bits and skip 2 bits}

$P_2 = D_3 D_6 D_7$ (XOR)

For ~~P~~ For P_4 association {check 4 bits and skip 4 bits}

$P_4 = D_5 D_6 D_7$ (XOR)

Given: 1110101

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

if it is correct then Take that parity as 0.

$P_1 = D_3 \oplus D_5 \oplus D_7$

$1 = 1 \oplus 1 \oplus 1 = 1$ (correct parity)

Then $P_1 = 0$.

$P_2 = D_3 \oplus D_6 \oplus D_7$

If parity is wrong then take it as 1

$0 = 1 \oplus 1 \oplus 1 = 1$ (Wrong parity/false parity)

Then $P_2 = 01$

$P_4 = D_5 \oplus D_6 \oplus D_7$

$0 = 1 \oplus 1 \oplus 1 = 1$ (false parity)

$P_4 = 1$

With respect to false parity, we can identify the location of error

$P_4 P_2 P_1 = 110 = 6$

we have error.

E = | 0 | 1 | 0 | 0 | 0 | 0 | 0 | $\oplus$ R = | 1 | 1 | 1 | 0 | 1 | 0 | 1 | = | 1 | 0 | 1 | 0 | 1 | 0 | 1 |

error ... received ... Corrected data

IEEE 754 floating point single precision 32-bit format

Step I : Convert given no. into binary

Step II : Repeat given no. into scientific notation

Step III : IEEE 754 floating point single precision 32

X — sign: +ve = 0, -ve = 1

(xxxxxxxx) 8 bit of exponent = exponent bias + power

(xxxx . . . xx) 23 bit Mantessa

① $(3066.25)_{10}$

Following above mentioned rule

2 | 3066
2 | 1533 - 0
2 | 766 - 1
2 | 383 - 0
2 | 191 - 1
2 | 95 - 1
2 | 47 - 1
2 | 23 - 1
2 | 11 - 1
2 | 5 - 1
2 | 2 - 1
1 - 0

101101111010

$0.25 \times 2 = 0.5$ — 0
$0.5 \times 2 = 1.0$ — 1
$0.0 \times 2 = 0.0$

(0.01)

$[3066.25] = [101111111010.01]_2$

Step2: $(101111111010.01)_2$

$1.01111111010 01 \times 2^{11} \rightarrow$ Scientific form

Step3:

As this is +ve, sign = 0

$= (2^{k-1} - 1) + 11$

↓ exponent bias.

127
+11
138

2^7 10000000
128 10001010
+10 2^3 2^1

$= 2^7 - 1 + 11$

$= 138 \rightarrow 10001010$

23-bit mantessa

(0111111101001.0000)

↓

23 bit mantessa

(zero padding).

HW

(2) $(231.75)_{10}$

$$
\begin{array}{l|l}
2 & 231 \\
2 & 115 - 1 \\
2 & 57 - 1 \\
2 & 28 - 1 \\
2 & 14 - 0 \\
2 & 7 - 0 \\
2 & 3 - 1 \\
2 & 1 - 1 \\
 & 0 - 1
\end{array}
$$

11100111

$0.75 \times 2 \Rightarrow 1.50 = 1$

$0.5 \times 2 = 1.0$ — 1

$0.0 \times 2 = 0.$

0.11

$(11100111.11)_2$

1.110011111×2^7

IEEE 754 floating point double precision 64-bit format

Step I - convert given no. to binary

Step II - Represent in scientific notation

Step III - IEEE 754 double precision 64 bit

X	(xxxx xxxx xxx)	(xxxx .. xx)
S	11 bit of exponent	52 bit
+ve = 0	= exponent bias + power	Mantessa
-ve = 1		

Q) $(A0F9.0FB)_{16}$.

A 0 F 9 . 0 F B

1010 0000 1111 1001 . 0000 1111 1011

$(1010\,0000\,1111\,1001.\,0000\,1111\,1011)_2$

Step 2: $(1.010\,0000\,1111\,1001\,0000\,1111\,1011) \times 2^{15}$

Step 3: As this is +ve, sign = 0

$= (2^{k-1} - 1) + 15$

$= (2^{10} - 1) + 15$

$= 2^{10} + 14$

$= \downarrow$

$$
\begin{array}{r}
10000000000 \\
+ \quad 1110 \\
\hline
10000001110
\end{array}
\rightarrow \text{exponent 11-bits.}
$$

52 bit mantessa → (010 0000 1111 1001 0000 1111 1011 .. 00)

↳ 52 bit

HW

Q [FAFA.01] → IEEE 754, 64-bit format

DeMorgan's Theorem

- Used to simplify the boolean expression
- Digital circuits (DC) can be implemented in many ways
- Digital circuits are simply made by these basic operators
  1) Boolean AND $A, B \rightarrow A\cdot B$ ⌒ output is high only when all inputs are high
  2) Boolean OR $A, B \rightarrow A+B$ – output is high, if any input is high
  3) Boolean NOT $A \rightarrow \bar{A}$ ⌊ invert of input

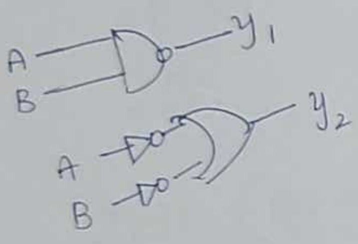
DeMorgans law

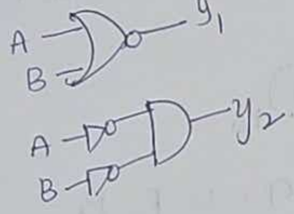
$\overline{A\cdot B} = \bar{A}+\bar{B}$

complement of product is the sum of complements

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

complement of sum is product of complements.





Truth tables

28/1/22

Consensus Theorem

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

LHS

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

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

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

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

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

Reference : NOT → $\bar{\bar{A}} = A$

AND → $A \cdot A = A$

$A \cdot 0 = 0$

$A \cdot 1 = A$

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

OR → $A + A = A$

$A + 0 = A$

$A + 1 = 1$

$A + \bar{A} = ~~A~~ \ 1$

Distributive law

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

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

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

Demorgan law

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

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

Theorem 2

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

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

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

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

$= ACB + B\bar{A}B + BCB + ACC + B\bar{A}C + BCC$

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

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

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

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

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

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

$=$ RHS

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

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

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

Boolean Algebra examples :

1) $\overline{\bar{A}B + \bar{A} + AB}$ $\qquad AB = X \qquad$ or $\qquad \overline{\bar{A}B + \bar{A} + AB}$

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

$= \bar{\bar{X}} \cdot \overline{1+\bar{A}}$ $\qquad\qquad = AB \cdot A \cdot \overline{AB}$

$= \bar{1} = 0$ $\qquad\qquad = 0$

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

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

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

$= A[(B+\bar{C})(\bar{A}) + 0]$

$= 0$

3) $A(B+\bar{C}(\overline{AB + A\bar{C}}))$ $\qquad$ HW

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

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

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

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

$= A(B+\bar{C}\bar{A} + 0)$

$= AB$

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

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

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

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

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

$= 0$

5) $(\overline{A+\overline{BC}})(A\bar{B} + A\bar{B}C)$ $\qquad$ HW

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

$= 0$

6) $A + \bar{B}C\,(A + \overline{\bar{B}C})$

7) If $\bar{A} + AB = 0$, then find values of A & B

$$\bar{A} + AB = 0 \Rightarrow \bar{A} + B = 0$$

$$\Downarrow$$

$$A = 1,\ B = 0$$

8) $A\bar{B}\,(1 + BA) + \overline{\bar{A}B}$

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

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

$= \bar{B} \cdot 1 \to 1?$

9) $\overline{ABC}\,(\overline{A + B} + C) \to (\overline{A + B + C})$?

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

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

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

$=$

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

$\overline{ABC}\,(\overline{A + B + C})$

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

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

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

10) If $x = 1$ is logical question

$$[x + z\,[\bar{y} + [\bar{z} + x\bar{y}])]]\,[\bar{x} + \bar{z}\,(x + y)] = 1$$

a) $y = z$

b) $y = \bar{z}$

c) $y = z = 1$

~~d)~~ $= z = 0$

11) If we have 3 variables A, B & C. Find the output $y = 1$ for majority of '1' in A, B & C also minimize the function.

A	B	C	y
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

majority is 1

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

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

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

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

$= C[B + A\bar{B}] + AB\bar{C}$ $\qquad$ $AB + BC + CA$

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

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

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

$= AC + B(A+C) \Rightarrow AC + AB + BC$

Dual and self Dual of BF

Dual Digits:-

AND with OR

1 with 0

0 with 1

$A \to B$, $B \to A$, $A \to A$, $B \to B$

Q) Find out dual of $(A+B)(A\bar{B}+C)(B+C) = (A+B)(\bar{B}+C)$

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

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

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

Self Dual: If Dual function is same function then it is known as self Dual

1) $F = AB + BC + AC$, is F is self Dual?

$F_d = (A+B)(B+C)(A+C)$ $\qquad$ $(A+B)(B+C) = B + AC$

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

$= BA + BC + AC + AC$

$= BA + BC + AC = F$

F is self dual

2) For n no. of variable total self dual $= 2^{2^{n-1}}$ $\qquad$ FORMULA $2^{2^{n-1}}$

Q) For $n=2$ variables. Find out total self duals?

$= 2^{2^{2-1}} = 2^2 = 4$

Q) $n=5$

$= 2^{2^{5-1}} = 2^{2^4} = 2^{16} = 65536$

SOP - Sum of product

[DNF - Disjunctive normal form]

Eg: $Y = AB + A\bar{B}C + A\bar{B}\bar{C}$

POS - Product of sum

[CNF - Conjunctive Normal Form]

Eg:- $Y = (A+B)(A+C)(\bar{A}+\bar{B})$

Standard ← SSOP

1. Each product term contain all the variable of the function

2. $F(A,B,C) = ABC + A\bar{B}C + AB\bar{C}$

SSOP ✓

not SOP

→ standard

SPOS

1. Each sum term contain all the variable of the function

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

SPOS ✓

not POS

Min term and Max term

Min term: Each individual term in SSOP is called Minterm

Max term: Each individual term in SPOS is called as Maxterm

For 3 variables

it makes all one's

it makes all zero's

F	A	B	C	Minterm SSOP	Maxterm SPOS
0	0	0	0	$\bar{A}\bar{B}\bar{C} - m_0$	$A+B+C - m_0$
1	0	0	1	$\bar{A}\bar{B}C - m_1$	$A+B+\bar{C} - m_1$
0	0	1	0	$\bar{A}B\bar{C} - m_2$	$A+\bar{B}+C - m_2$
1	0	1	1	$\bar{A}BC - m_3$	$A+\bar{B}+\bar{C} - m_3$
0	1	0	0	$A\bar{B}\bar{C} - m_4$	$\bar{A}+B+C - m_4$
1	1	0	1	$A\bar{B}C - m_5$	$\bar{A}+B+\bar{C} - m_5$
0	1	1	0	$AB\bar{C} - m_6$	$\bar{A}+\bar{B}+C - m_6$
1	1	1	1	$ABC - m_7$	$\bar{A}+\bar{B}+\bar{C} - m_7$

if function is this

bar to 0

bar to 1

$F_{min} = \bar{A}\bar{B}\bar{C} + \bar{A}B\bar{C} + A\bar{B}\bar{C} + AB\bar{C}$

$\bar{A}\bar{B}C + \bar{A}BC + A\bar{B}C + ABC = \Sigma m(1,3,5,7)$

$= \Sigma m(0,2,4,6)$

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

$= \Pi_m(1,3,5,7)$ $(A+B+C)(A+\bar{B}+C)(\bar{A}+B+C)(\bar{A}+\bar{B}+C)$

$\Pi m(0,2,4,6)$

31/1/22

- each individual term in SPOS is Max term
- each individual term in SSOP is Min term

(1,3,4,6,7)

f'	F	A	B	C	Min term, 1 SSOP	Maxterm, 0 SPOS
0	0	0	0	0	$\bar{A}\bar{B}\bar{C} - m_0$	$(A+B+C) - m_0$
1	1	0	0	1	$\bar{A}\bar{B}C - m_1$	$(A+B+\bar{C}) - m_1$
0	0	0	1	0	$\bar{A}B\bar{C} - m_2$	$(A+\bar{B}+C) - m_2$
1	1	0	1	1	$\bar{A}BC - m_3$	$(A+\bar{B}+\bar{C}) - m_3$
1	0	1	0	0	$A\bar{B}\bar{C} - m_4$	$(\bar{A}+B+C) - m_4$
0	1	1	0	1	$A\bar{B}C - m_5$	$(\bar{A}+B+\bar{C}) = m_5$
1	0	1	1	0	$AB\bar{C} - m_6$	$(\bar{A}+\bar{B}+C) - m_6$
1	1	1	1	1	$ABC - m_7$	$(\bar{A}+\bar{B}+\bar{C}) = m_7$

(0,2,5)

$f'_{min} = \Sigma m(0,2,5)$

$f'_{max} = \Pi m(1,3,4,6,7)$

$F_{min} = \Sigma m\{0,2,4,6\}$

$F_{max} = \Pi m(1,3,5,7)$

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

$= m_1 + m_3 + m_5 + m_7$

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

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

$= m_0 + m_2 + m_3 + m_{\cancel{4}\,6}$

$= \Sigma m(0,2,3,6)$

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

$= m_0 + m_2 + \cancel{m_3 +} m_7$ $\quad m_0 \cdot m_2 \cdot m_3 \cdot m_7$

$= \Pi m(0,2,3,7)$

f'_{min} $F = \bar{A}\bar{B}C + \bar{A}BC + A\bar{B}\bar{C} + AB\bar{C} + ABC$ ✓ correct

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

f'_{max} $F = (A+B+C)(A+\bar{B}+C)(\bar{A}+B+\bar{C})$

$= \prod m(0,2,5)$

Maxterm is ~~AND~~ and Minterm is complement to eachother.

SOP to SSOP conversion

Step 1: Identify the missing variables in product term

Step 2: Multiply [Missing variable and its complement)

Step 3: Neglect the repeated term

step 1) $F(A,B,C) = \underset{\downarrow C}{AB} + A\bar{B}C + \underset{\downarrow B}{AC}$

step 2) $F(A,B,C) = A.B.(C+\bar{C}) + A\bar{B}C + A(B+\bar{B})C$

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

step 3) $F(A,B,C) = \underset{m_7}{ABC} + \underset{m_6}{AB\bar{C}} + \underset{m_5}{A\bar{B}C}$

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

Q: $F(A,B,C,D) = \underset{\downarrow C\&D}{AB} + \underset{B\&D}{AC} + ABC\bar{D}$ — HW

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

HW → $F(A,B,C,D) = \underset{\downarrow B\&C\&D}{A} + \underset{A\&B}{CD}$

POS to SPOS conversion

Step 1: Identify the missing variable

$$F(A,B,C) = \underbrace{(\bar{A}+\bar{B})}_{C}\ \underbrace{A}_{B\ \&\ C}\ (A+B+\bar{C})$$

Step 2: Add with that variable and its compliment seperately

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

repeated

Step 3: Neglect the repeated terms

$$F(A,B,C) = \underbrace{(A+\bar{B})}_{C}\ \underbrace{(\bar{A}+C)}_{B}\ (\bar{A}+\bar{B}+C)$$

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

$$= m2 \quad m3 \quad m4 \quad m6$$

$$= \Pi m(2,3,4,6)$$

SSOP to SPOS / SPOS to SSOP } conversion

1) $F(A,B,C) = \Sigma m(0,1,4,7)$

Convert SSOP to SPOS

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

$\Pi m(2,3,5,6)$ (Complement of SSOP)

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

2) $F(A,B,C) = \Pi m(1,2,6)$

$n=3$

$2^n = 8$

(0 to 7)

Convert SPOS to SSOP

$$\Pi m(1,2,6) = (A+B+\bar{C})(A+\bar{B}+C)(\bar{A}+\bar{B}+C)$$

$$= \Sigma m(0,3,4,5,7)$$

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

3) $F(A,B,C) = ABC + AB\bar{C} + \bar{A}BC$

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

$\downarrow$

$\Pi m(0,1,2,4,5)$

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

HW → ④ $F(A,B,C) = (A+\bar{B}+C)(\bar{A}+B+\bar{C})(A+B+\bar{C})$

Convert to SSOP

SPOS

Gate question

Q. If $n=3$ variables then total min max term $= 8 \rightarrow 2^n$

total max term $= 8 - 2^n$

Total term $= 2^{2^n} = 2^{2^3} = 2^8 = 256$

Self dual $= 2^{2^{n-1}}$

Q. If $y(A,B,C) = A + \bar{B}C$, Find SSOP & SPOS?

B & C   A

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

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

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

SPOS $\Rightarrow \Pi m(0,2,3)$

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

HW

Q $y = (A+B)(A+C)$ {Find SPOS & SSOP}. Find total min and max term.

$\downarrow$ C $\downarrow$ B

Logic Gates Introduction

There are three basic categories of logic gates

⇒ Basic logic gates
AND, OR, NOT.

⇒ Universal
NAND, NOR

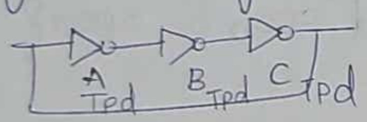
⇒ Arithmetic (secondary) logic gates
XOR, XNOR

NOT gate

x	y
0	1
1	0

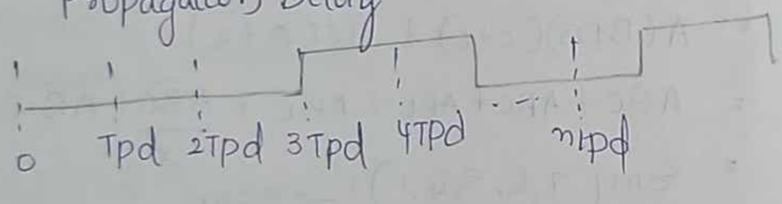
Application of NOT gate:-

1) Not gate as ring oscillator



y (odd no. of inverter)
0/1

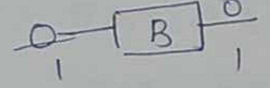
Propagation Delay



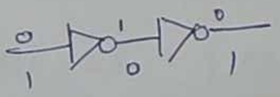
$T = 2n\,T_{pd}$ ⟶ no. of inverters

$f = \frac{1}{2n\,T_{pd}}$ $\quad f = 1/T$

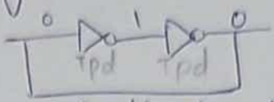
2) NOT gate also behave as Buffer.



(no feedback)



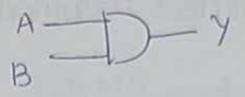
3) NOT gate as Bistable Multivibrator



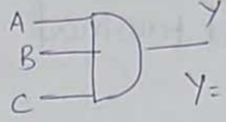
feedback path → like buffer with feedback

1/2/22

AND gate



$Y = A \cdot B$ (two terminal)

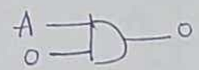


$Y = ABC$ (# multiterminal)

If all inputs are high, then o/p is high else o/p is zero.

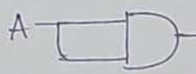
A	B	Y
0	0	0
0	1	0
1	0	0
1	1	1

Properties of AND gate :-



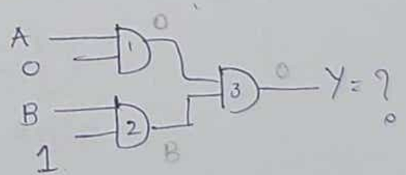
$Y = 0$

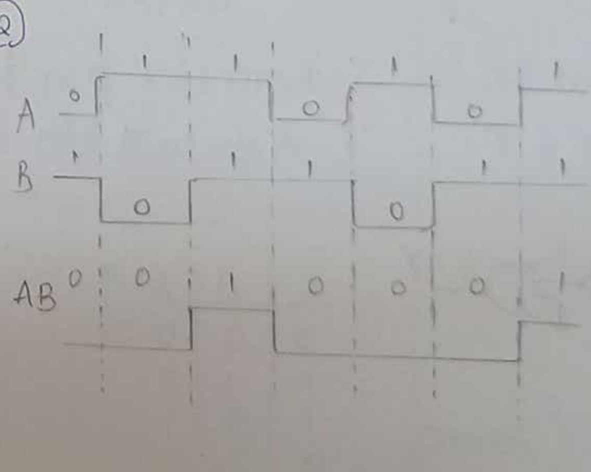
$Y = A$



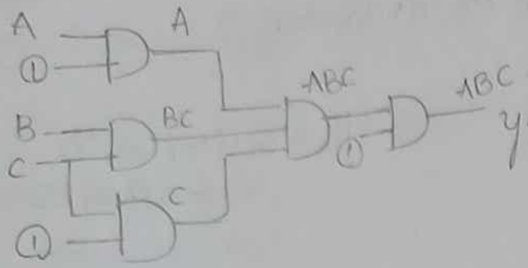
$Y = A$

Example:

① 

② 

③



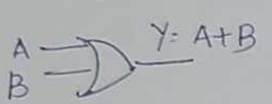
y.ABC

* For Transistor-transistor logic (TTL) non connected terminal at V_p th' active high '1' input

* ECL (Emitter coupled Logic) non connected terminal at V_p inactive low '0'

For ECL, $Y=0$ in above problem

OR gate

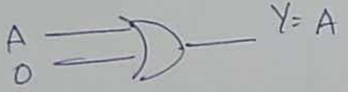


In OR gate if any input is '1' then output is '1' else o/p is '0'.

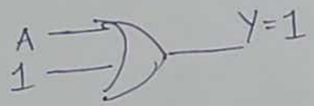
A	B	Y
0	0	0
0	1	1
1	0	1
1	1	1

Properties

①



②

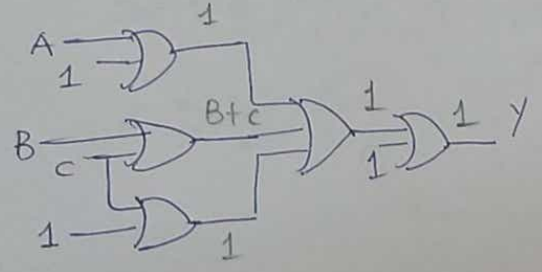


③

Example:

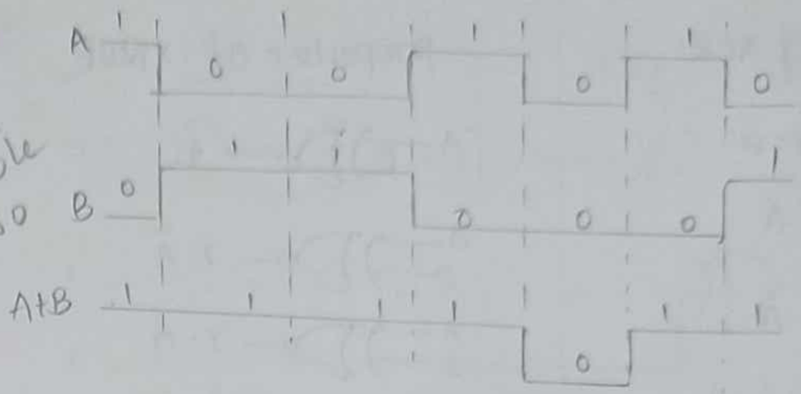
①.

TTL

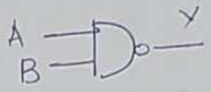


ECL $A+(B+C)+C = A+B+C$.

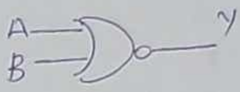
(2) HW Truth table also



NAND and NOR gate



$Y = \overline{A \cdot B}$



$Y = \overline{A + B}$

In NAND gate, if any i/p is '0' then o/p will be '1' else o/p will be '0'.

In NOR gate, if any i/p is '1' then o/p will be '0' else o/p will be '1'

NAND

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

NOR

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

XOR and XNOR gate

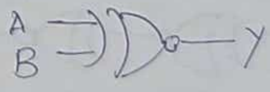
$Y = A \oplus B$

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

If no. of '1' at input is odd then o/p is '1' else o/p is '0'

*

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



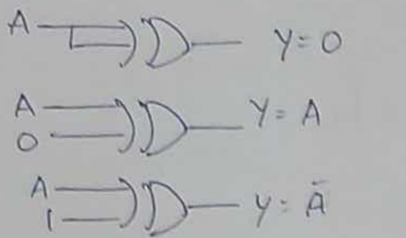
$Y = A \odot B$

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

If no. of '1' at i/p is even then o/p is '1' else output is '0'.

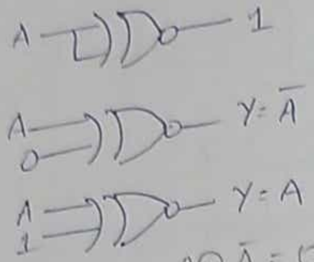
A	B	Y
0	0	1
0	1	0
1	0	0
1	1	1

Properties of XOR



$A \oplus \bar{A} = 1$

Properties of XNOR



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

Examples:

① 

What is the equivalent gate of given circuit?

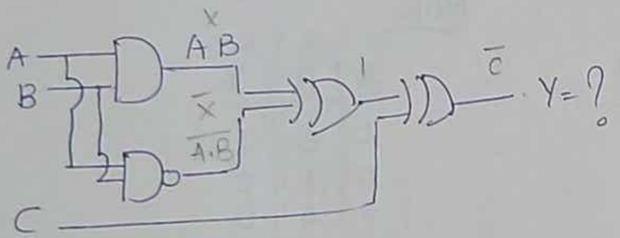
$Y = A \oplus \bar{B}$ [For $A \oplus B = A\bar{B} + \bar{A}B$]

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

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

$= A \odot B$

↳ XNOR

② 

$x \oplus \bar{x}$

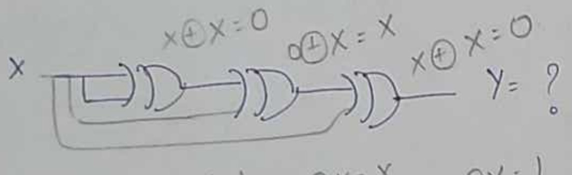
$x\bar{\bar{x}} + \bar{x}\bar{x}$

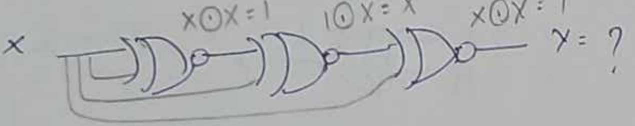
$xx + \bar{x}\bar{x}$

$x + \bar{x}$

1

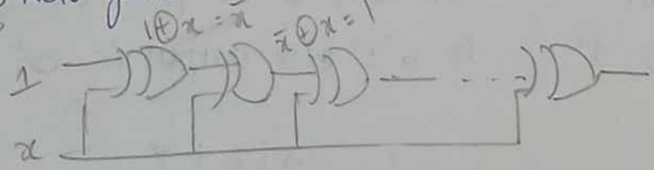
$1 \oplus C$

③ HW 



④ If the input of digital circuit consisting of cascade of 20 XOR gate. Then what will be the output

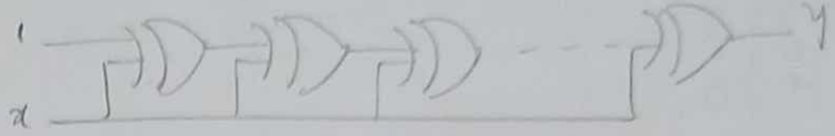
HW Solve in next class



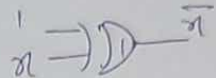
a) 0 b) 1 c) $\bar{x}$ d) x

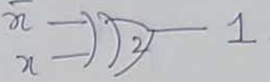
TTL, $x \oplus 1 = \bar{x}$ $\bar{x} \oplus x = 1$

2/2/22



x	y	py
0	0	0
0	1	1
1	0	1
1	1	0.



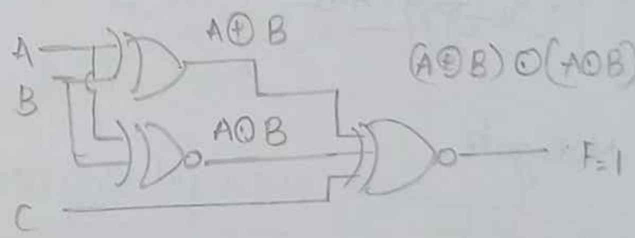


Output of 1, 3, 5, 7, 9, 11, 13, 15, 17, 19 gates is $\bar{x}$

Output of 2, 4, 6, 8, 10, 12, 14, 16, 18, 20 gates is 1

Ⓠ Find output F to be 1 in the logic circuit, what input combination is appropriate.

1) A = 1, B = 1, C = 0
2) A = 1, B = 0, C = 0
3) A = 0, B = 1, C = 0
4) A = 0, B = 0, C = 1. ✓



$(A \oplus B) \odot (A \odot B) \odot C$.

$A \oplus B$ / $A \odot B$ One will be '1' and other will be '0' So, C should be 1.

H.W

Q) If input to digital circuit consisting of cascade of 20 XNOR gates, then what will be the output y?

$1 \odot x = x$ $\quad x \odot x = 1$

y = ? 1.

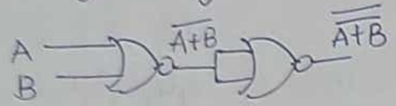
NOR as Universal gate

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

1) Not gate by NOR gate

$y = \bar{A}$

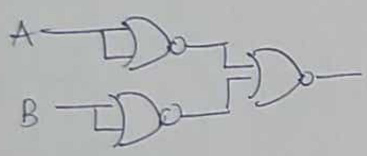
2) OR gate by NOR gate



$y = A+B$

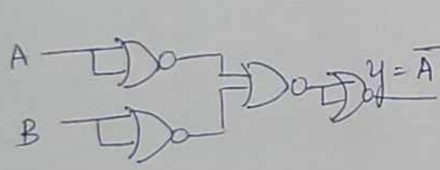
3) AND gate by NOR gate

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

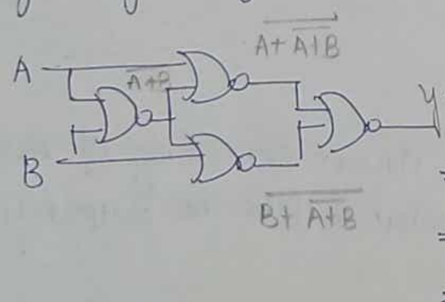


$y = A \cdot B$

4) NAND gate by NOR gate



5) XNOR gate by NOR gate



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

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

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

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

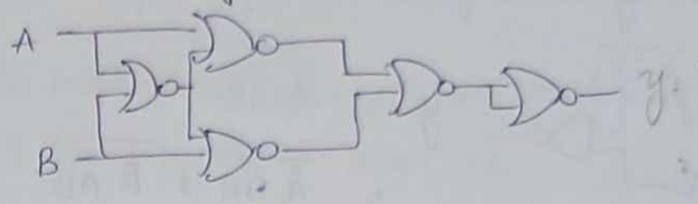
$= (\overline{\bar{A}B}) \cdot (\overline{\bar{B}A})$

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

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

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

6) XOR gate by NOR gate



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

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

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

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

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

$=$ XOR

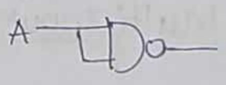
NAND as Universal gate

~~XY~~

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

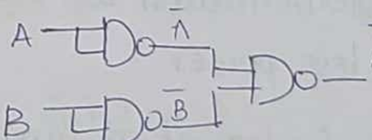
For nand gate if any input is '0', o/p is '1' else '0'

1) NOT gate by NAND gate



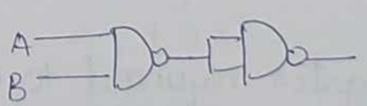
$y = \bar{A}$

2) OR gate by NAND gate



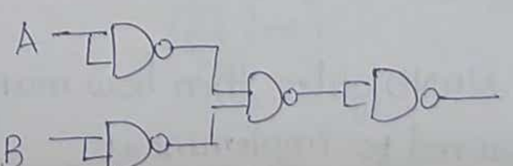
$\overline{\bar{A}\cdot\bar{B}} = A+B = y$

3) AND gate by NAND gate



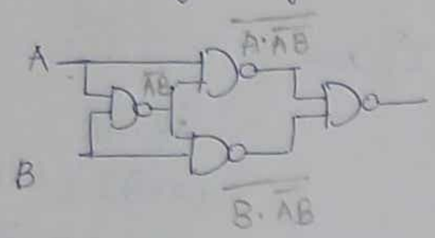
$y = A \cdot B$

4) NOR gate by NAND gate



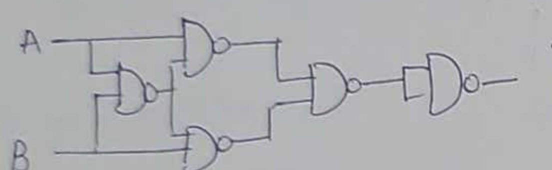
$y = \overline{A+B}$

5) EXOR gate by NAND gate



$$y = \overline{\overline{A\cdot\overline{AB}}\cdot\overline{B\cdot\overline{AB}}}$$
$$= \overline{\overline{A\cdot\overline{AB}}} + \overline{\overline{B\cdot\overline{AB}}}$$
$$= A\cdot\overline{AB} + B\cdot\overline{AB}$$
$$= \overline{AB}\,(A+B)$$
$$= (\bar{A}+\bar{B})(A+B)$$
$$= A\bar{A} + \bar{A}B + \bar{B}A + B\bar{B}$$
$$= A\bar{B} + \bar{A}B = XOR$$

6) XNOR gate by NAND gate



$$y = \overline{A\bar{B}+\bar{A}B}$$
$$= \overline{A\bar{B}}\cdot\overline{\bar{A}B}$$
$$= (\bar{A}+B)(A+\bar{B})$$
$$= \bar{A}\bar{B} + AB = XNOR$$

Minimum two input NAND gates for Multi Input AND and Multiinput NAND gate

* In NAND and NOR gate implementation we should prefer NAND gate as it consumes less power.

$2(n-1)$ - $(2\cancel{n}-1)$ (two i/p NAND to implement n input AND)

- $(2n-3)$ (two i/p NAND to implement n input NAND).

Q) How many two i/p NAND gates required to implement 4 i/p AND gate

$$2(n-1) = 2\times(4-1) = 6 \quad \cancel{8-1=7} \quad \cancel{4\cdot 8-1=7}$$

Q) If you have 4 i/p NAND gates then how many 2 i/p NAND gates are required to implement it

$$2n-3 = 2\times 4-3 = 8-3 = 5$$

3) Find 2 i/p NAND gate for given boolean function

$F = A \cdot B \cdot C \cdot \overline{D}$

→ To implement NOT gate we need only one NAND gate

→ 4 i/p AND gate → $2(n-1) = 6$

Total $= 6+1 = 7$

4) $F = \overline{\overline{A} \cdot B \cdot \overline{C}}$

~~3 i/p AND gate → $2(3-1) = 4$~~

~~$\overline{A}, \overline{C} \to 2$~~

~~$\overline{\overline{A} \cdot B \cdot \overline{C}} \to 1$~~

~~Total $= 4+2+1 = 7$~~

$\overline{A}, \overline{C} \to 2$

3 i/p NAND gate

$\hookrightarrow 2n-3$

$= 6-3 = 3$

Total $= 3+2 = 5$

HW 5) Identify min 2 i/p NAND gate

A) $F_1 = AB\overline{C} \Rightarrow 5$

B) $F_2 = A\overline{B\overline{C}} \Rightarrow 5$

C) $F_3 = \overline{\overline{A} \cdot \overline{B}} \Rightarrow 3$

6) Min no. of 2 i/p NAND gate for logical expression

$F = (\overline{a+b})(c+b)$

1) 3 2) 4 3) 5 4) 6

DeMorgan's law $\overline{a+b} = \overline{a} \cdot \overline{b}$

$\overline{a} + \overline{b} = \overline{a \cdot b}$

$= \overline{a} \cdot \overline{b}(c+b)$

$=$ Let, $A = \overline{a} \cdot \overline{b} \Rightarrow 1$ NAND

$= A(c+b)$

$= A \cdot C + A \cdot b$

$= \overline{\overline{A \cdot C + A \cdot b}}$

$= \overline{(\overline{A \cdot C})(\overline{A \cdot b})}$

$= \overline{A \cdot C} \to 1$

$\overline{A \cdot b} \to 1$

$\overline{\overline{A \cdot C} \cdot \overline{A \cdot b}} \to 1$

Total $= 4$

7) Min 2 i/p NAND gates for $F = w\overline{x}y\overline{z}$

a) 4 b) 5 c) 6 d) 7

$2(4-1) = 6 + 2 = \underline{8}$

Total 2 i/p NAND gates = 8

Min = 6 (exclude those two $\bar{x}, \bar{z}$)

8) Find Min 2 i/p NAND gates for $F = \overline{\bar{A}\bar{B}\bar{C}D}$

Total = 3 + (2×4−3)

= 3+5 = 8

Min

Step-1: Resolve Boolean Expression

Step-2: Check NAND operation in given expression

Step-3: If no NAND operation then by double complement, we can use deMorgan's

Step-4: Atleast use multi input AND rule or use multi input NAND rule

$\overline{AB+BC+CA} = \overline{AB}\,\overline{BC}\,\overline{CA} \rightarrow 3 + (2(3-1))$

$= 3+4$

$= 7$

$= \overline{\overline{\overline{AB}\,\overline{BC}\,\overline{CA}}}$

$\overline{\overline{AB}\,\overline{BC}\,\overline{CA}} \rightarrow 2(3)-3$

$= 3+3 = 6$

$AB+BC+CA$

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

$= \overline{\overline{AB}\,\overline{BC}\,\overline{CA}}$

$2(3)-3$

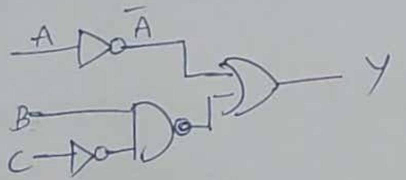
$= 3+3 = 6$

Boolean Expression to NAND gate Implementation

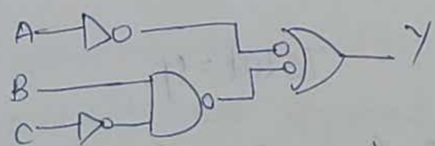
1) $Y = \bar{A} + B\bar{C}$

Step1: Implement given Boolean expression in terms of basic gates of (AND, OR, NOT) AOI

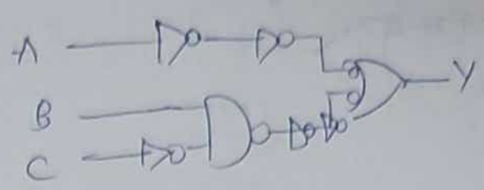
↳ inverter



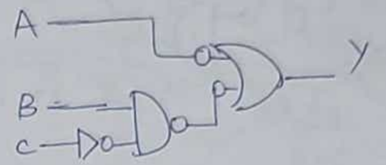
Step2: Apply bubble to o/p of AND gate to i/p of OR gate



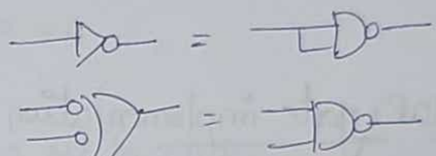
Step3: Apply NOT gate at all places where bubble inserted



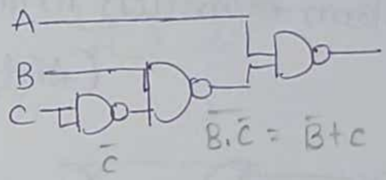
Step4: Cancel extra NOT gates



Step5: NAND gate equivalent of the ckt



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



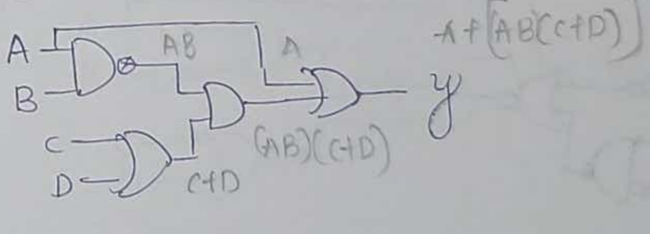
$Y = \overline{A(\bar{B}+C)}$

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

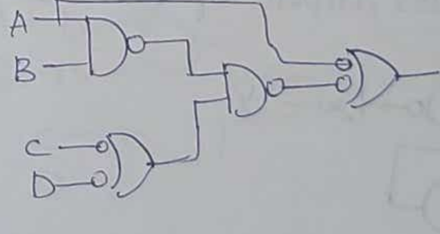
$= \bar{A} + B\bar{C} = Y$

XOR → NAND (Try it)

AOI to NAND implementation

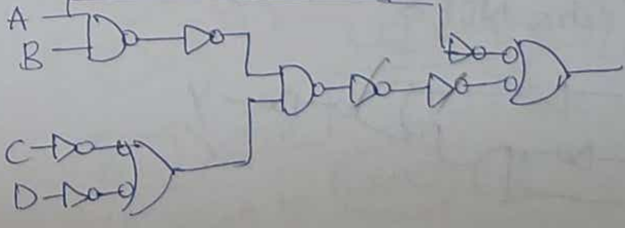


Step1:

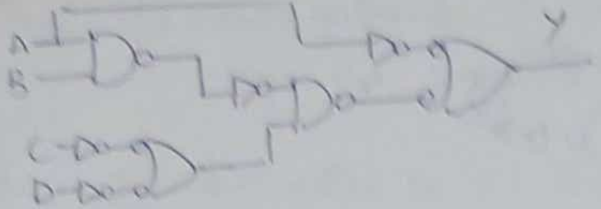


Apply Bubble
① o/p of AND gate
② i/p of OR gate

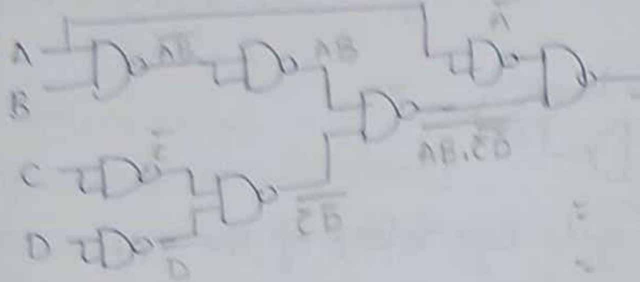
Step2:



Step 3:



Step 4:



$\overline{\bar{A} \cdot \overline{AB \cdot \bar{C}\bar{D}}}$

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

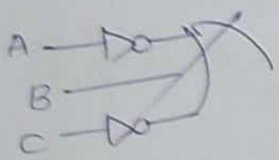
$= A + (AB)(C+D)$

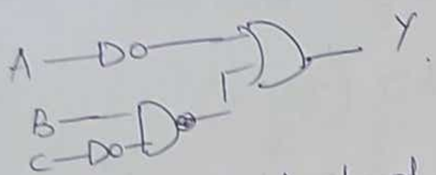
$= A + ABC + ABD$

Boolean expression for NOR gate implementation

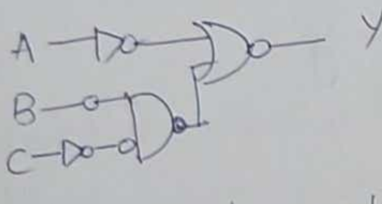
$Y = \bar{A} + B\bar{C}$

Step 1: Implement given boolean expression in term of AOI (And, OR, NOT)

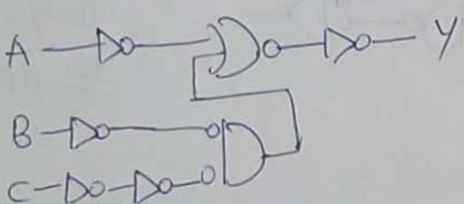




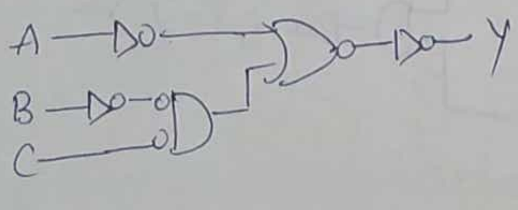
Step 2: Add bubbles at o/p of OR and i/p of AND



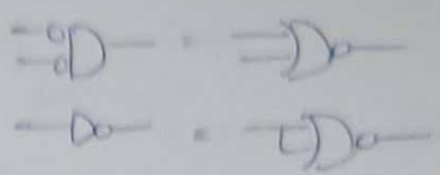
Step 3: Connect NOT gates in place of bubbles.



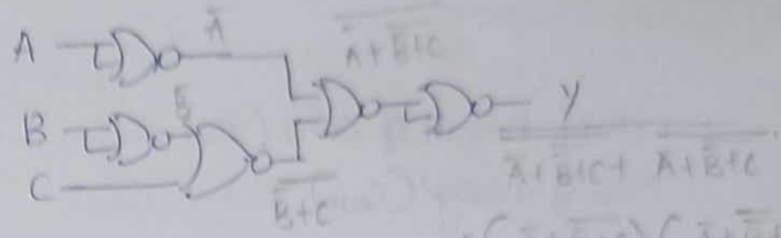
Step 4: Cancel extra NOT's.



Step 5: NOR gate equivalent



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



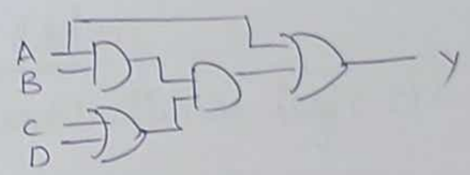
$\overline{\overline{\bar{A}+\overline{\bar{B}+C}} + \overline{\bar{A}+\overline{\bar{B}+C}}}$

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

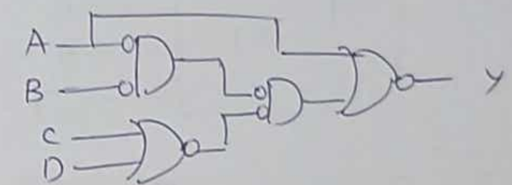
$= (\bar{A}+\overline{\bar{B}+C})$

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

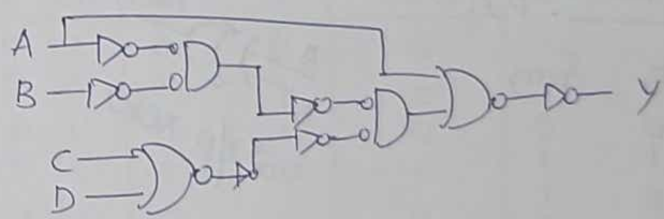
AOI (A, O, N) to NOR



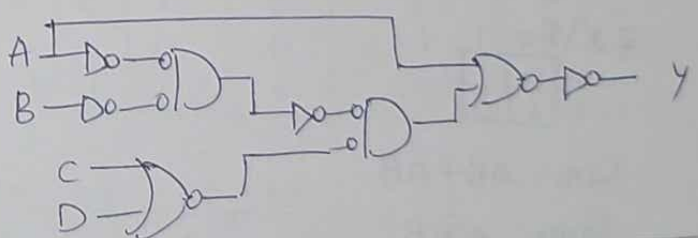
Step 1:



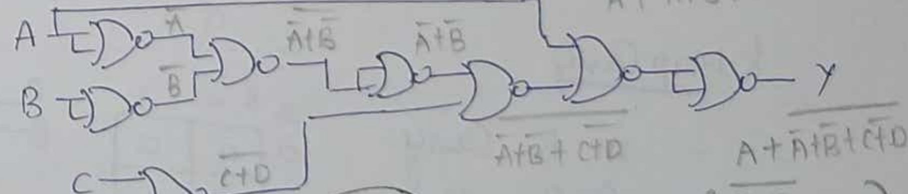
Step 2:



Step 3:



Step 4:



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

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

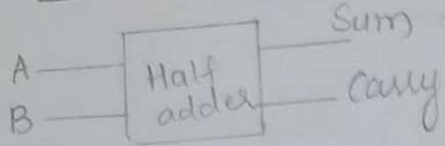
$= A + (AB\cdot C+D)$

$= A + ABC + ABD$

Preference is given to NAND

Adders :- BCD, EX-3, GC, Binary etc

Half Adder



LSB of Result of sum (Σ_0)
MSB of Carry (C_{out})

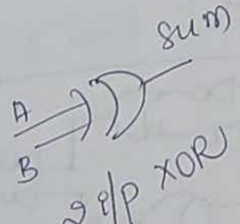
```
          1 1 0 0
          1 0 1 0
carry (1) 0 1 1 0
```

Half Adder Truth Table

Input		Output	
A	B	Sum	Carry
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1
		↓ XOR	↓ AND

K-Map (truth Map)

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



A\B	0	1
0	0	1
1	1	0

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

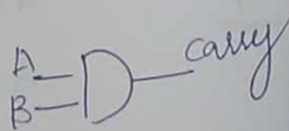
$\text{Sum} = A \oplus B$

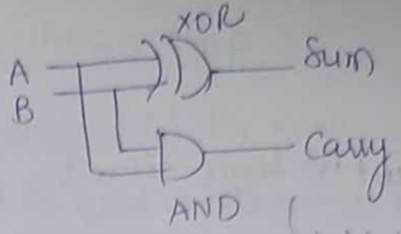
K-Map Carry

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

A\B	0	1
0	0	0
1	0	1

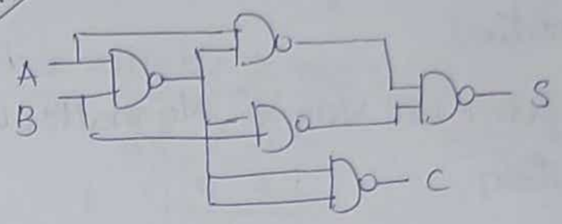
$\text{Carry} = A \cdot B$



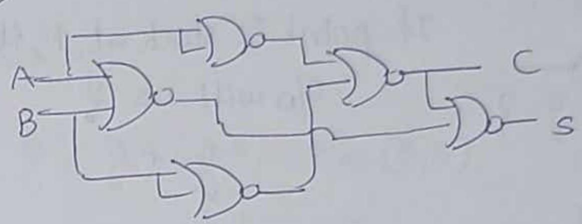


↳ Make half adder using NAND & NOR gates

NAND

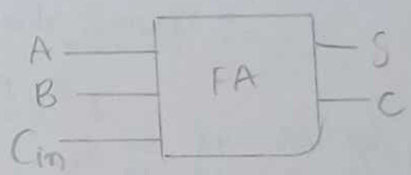


NOR



Full Adder

Combinational logic ckt



Complex compared to Half adder

A	B	C_{in}	S	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

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

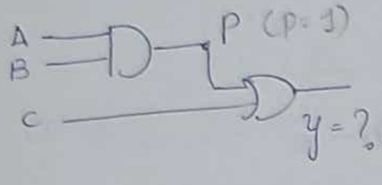
9/2/22

Stuck at 1 and stuck at 0 Fault in Logic circuit

Stuck at 1: o/p at given point will stay '1' no matter what is the ckt connection.

Stuck at 0: o/p at given point will stay '0', No matter what is the ckt connection.

Ex:- ①



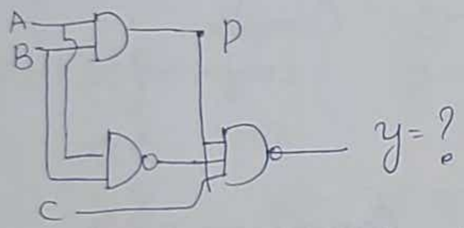
If point is stuck at 1, the o/p will be ?

$y = 1$

If $P = 0$, $y = C$

Note: For OR gate, if any i/p is logic 1, then o/p will be '1' logic.

②

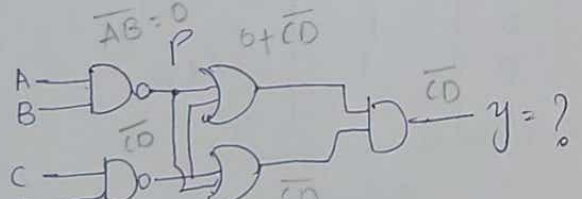


if P is stuck at 0, the o/p will be ? $y = 1$.

~~P is~~

Note:- For NAND gate, if any i/p is logic 0, the o/p will be logic 1.

③



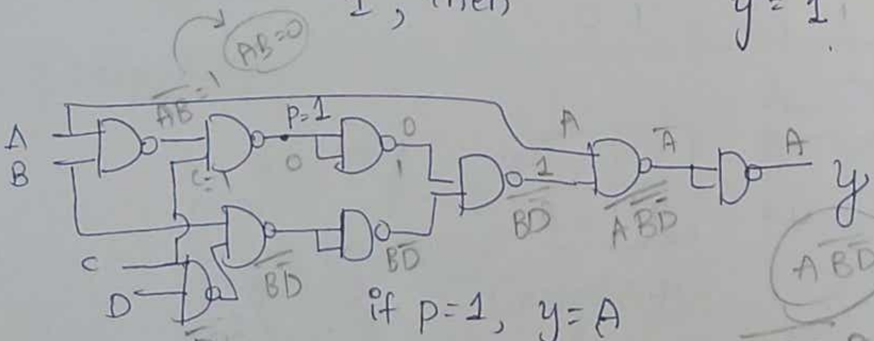
$\overline{AB} + \overline{CD}$

$0 + \overline{CD}$

if P is stuck at 0, then o/p will be $y = \overline{CD}$

1, then $y = 1$

④



if $P = 1$, $y = A$

$\overline{A\overline{BD}}$ → if P=0

$\overline{\overline{CD}\,B \cdot A}$

KARNAUGH MAP

- Developed by Mr. Karnaugh in 1953.
- It is used to minimize boolean expression.
- It is prepared based on Gray code

Two variable k-map :-

B \ A	$\bar{A}$ 0	A 1
$\bar{B}$ 0	0	2
B 1	1	3

A	B	
0	0	– 0
0	1	– 1
1	0	– 2
1	1	– 3

$cell = 2^n = 2^2 = 4$

Q :- $y = (A, B) = \Sigma m(0,3)$

B \ A	0	1
0	1	
1		1

→ locating by k-map or representing

3-Variable k-map :-

1 → SOP
0 → POS.

$cell = 2^n = 2^3 = 8$

C \ AB	$\bar{A}\bar{B}$ 00	$\bar{A}B$ 01	AB 11	$A\bar{B}$ 10
$\bar{C}$ 0	1 0	2	6 1	4 1
C 1	1 1	3	7	5

A	B	C	
0	0	0	– 0
0	0	1	– 1
0	1	0	– 2
0	1	1	– 3
1	0	0	– 4
1	0	1	– 5
1	1	0	– 6
1	1	1	– 7

$\Rightarrow Y(ABC) = \Sigma m(0,1,4,6)$

4-variable k-map :-

$2^n = 2^4 = 16$.

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

Q :- $Y(A,B,C,D) = \Sigma m(0,1,4,6,10,12)$

K-Map Rules for Grouping cells, K-map rules for formation Boolean function K-map:

1) Group should not contain zero and cells contain 1, must be grouped
2) We can group 1, 2, 4, 8, ... 2^n cells
3) Each group should be as large as possible.
4) Group may overlap.
5) Opposite grouping and corner grouping is allowed.
6) There should be as few groups as possible

① corner group. 1st group. 2nd

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

not changing — me
changing — sie

Boolean expression, $Y = \bar{B}\bar{D} + \bar{A}\bar{D}$

②

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

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

4, 5, 6, 7 → 1
0, 4, 2, 1 → 2
9, 11 → 3

3

Implicants, Prime implicants and Essential prime implicants in k-map :-

- Implicants → The group of 1's is called implicants
- Prime implicant → It is largest possible groups of 1's
- EPI - In the group at least there is single '1' which cannot combine in other way

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

5, 7, 13, 15 – 1 gr

0, 1 – 2 gr

14, 15 – 3 gr

Gr II & Gr III are essential prime implicants

Gr I → PI

K-Map Examples

Q1. In the sum of products function is

$f(x,y,z) = \Sigma m(2, 4, 3, 4, 5)$. Then tell the prime implicants in the following

1) $\bar{x}y, x\bar{y}$
2) $\bar{x}y, x\bar{y}\bar{z}, x\bar{y}z$
3) $\bar{x}yz, \bar{x}yz, x\bar{y}$
d) $\bar{x}y\bar{z}, \bar{x}yz, x\bar{y}z$

z \ xy	00	01 ($\bar{x}y$)	11	10 ($x\bar{y}$)
0	0	1 (2)	6	1 (4)
1	1	1 (3)	7	1 (5)

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

Q2. The k-map for a boolean function is shown below, then the no. of essential prime implicants for the function will be

a) 4
b) 5
c) 6
d) 8

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

0, 2, 8, 10 → corners 1
0, 4 – 2
2, 3 – 3
8, 9 – 4
4, 5 – 5

5 –
3 –
9 – EPI = 4
10 –

Q3. Solve this given Boolean expression using k-map

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

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

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

Q. Find the Boolean Expression of k-map given below

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

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

0, 2, 8, 10 → I → $\bar{B}\bar{D}$
5, 7, 13, 15 → II → BD
or ← 8, 9, 10, 11, → III → $A\bar{B}$

or

9, 11, 13, 15
↳ instead of $A\bar{B}$

$\boxed{AD}$

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

Solve boolean expression by k-map

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

$y = A + C$

Q. Find the Boolean expression for K-Map given below.

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

optional

4,5,6,7   1,3,9,11

$\bar{A}B + D\bar{B} + B\bar{D} \rightarrow 4,12,6,14$

or

$\bar{A}D \rightarrow 1,5,3,7$

K-Map examples on don't care

1) The no. of product term in the minimize sum of product expression obtained through the following k-map is

$\boxed{x \rightarrow \text{don't care}}$

Note:- Don't care is used to increase the size of group but it is not necessary or compulsory to use all don't care.

CD \ AB	$\bar{A}\bar{B}$	$\bar{A}B$	AB	$A\bar{B}$
$\bar{C}\bar{D}$	1	0	0	1
$\bar{C}D$	0	x	0	0
CD	0	0	x	1
$C\bar{D}$	1	0	0	1

a) 2
b) 3
c) 4
d) 5

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

or

$\bar{B}\bar{D} + ACD = y$

2) Solve given boolean k-map (x is don't care)

	$\bar{A}\bar{B}$	$\bar{A}B$	AB	$A\bar{B}$
$\bar{C}\bar{D}$	1	x		1
$\bar{C}D$				
CD				1
$C\bar{D}$	x	1		1

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

0,2,8,10   0,4,2,6   10,11

3) Solve given boolean k-Map [x is don't care]

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

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

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

All don't care conditions are not necessarily taken

or

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

* Don't care is used to make groups as large as possible, it is not compulsory to take all don't care's, it is only compulsory to take all 1's

Q1)

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

X → leave this

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

Q2) $y = \Sigma m(1,2,5,9,13) + \Sigma d(3,6,11,15)$

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

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

or

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

Q3) Solve

	$\bar{A}\bar{B}$	$\bar{A}B$	AB	$A\bar{B}$
$\bar{C}\bar{D}$	1			1
$\bar{C}D$	1		x	x
CD	x	x	1	1
$C\bar{D}$	1			1

else AD

$\bar{B}$

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

$\bar{B} + CD$ } or

$\bar{B} + AD$

K-Map for POS expression

SOP
POS given 0
Find function (f_d)
Put complement of all variable ($\bar{f}$) (y)

Q1) If Boolean function is given by $y = \Sigma m(3, 6)$ then

a) $y = B(A+C)(\bar{A}+\bar{C})$
b) $y = B(A+\bar{C})(\bar{A}+\bar{C})$
c) $y = \bar{B}(A+\bar{C})(\bar{A}+C)$
d) $y = \bar{B}(A+C)(\bar{A}+\bar{C})$

C \ AB	$\bar{A}\bar{B}$	$\bar{A}B$	AB	$A\bar{B}$
$\bar{C}$	0_0	0_2	1_6	0_4
C	0_1	1_3	0_7	0_5

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

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

Q2) If Boolean function is given by

1	0	1	0
0	1	0	1

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

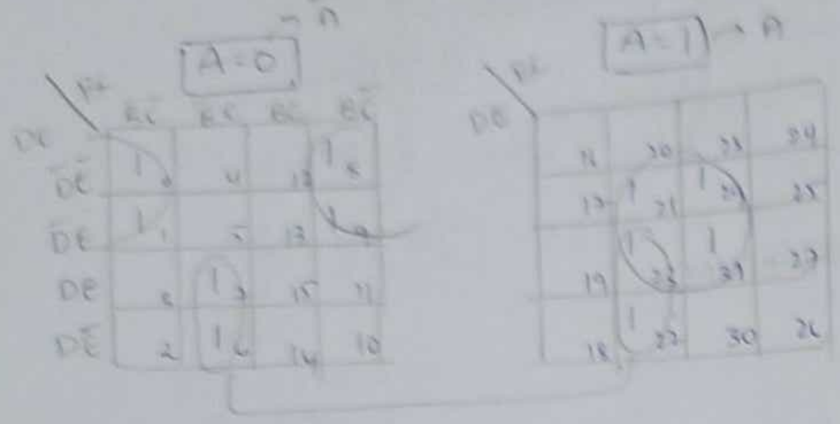
$y = (A+B+\bar{C})(A+\bar{B}+C)(\bar{A}+\bar{B}+\bar{C})(\bar{A}+B+\bar{C})$

5-Variable K-map

$y(A, B, C, D, E) = \Sigma m(0, 1, 6, 7, 8, 9, 21, 22, 23, 29, 31)$

$2^n = 2^5 = 32$

We have to make k map by bisecting it into two groups, to have with one group make A=0, other A=1.



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

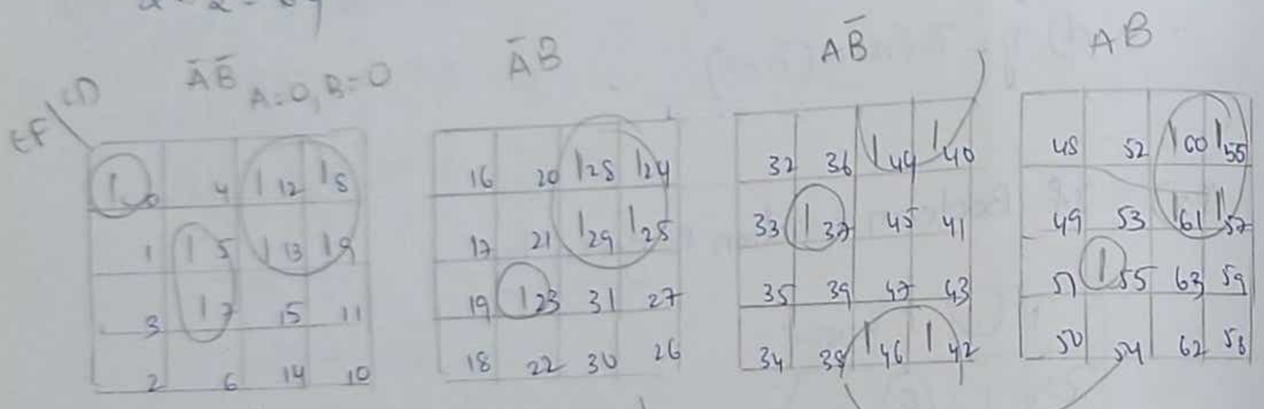
$f(A,B,C,D,E) = \bar{B}CD + \bar{A}\bar{C}\bar{D} + ACE$

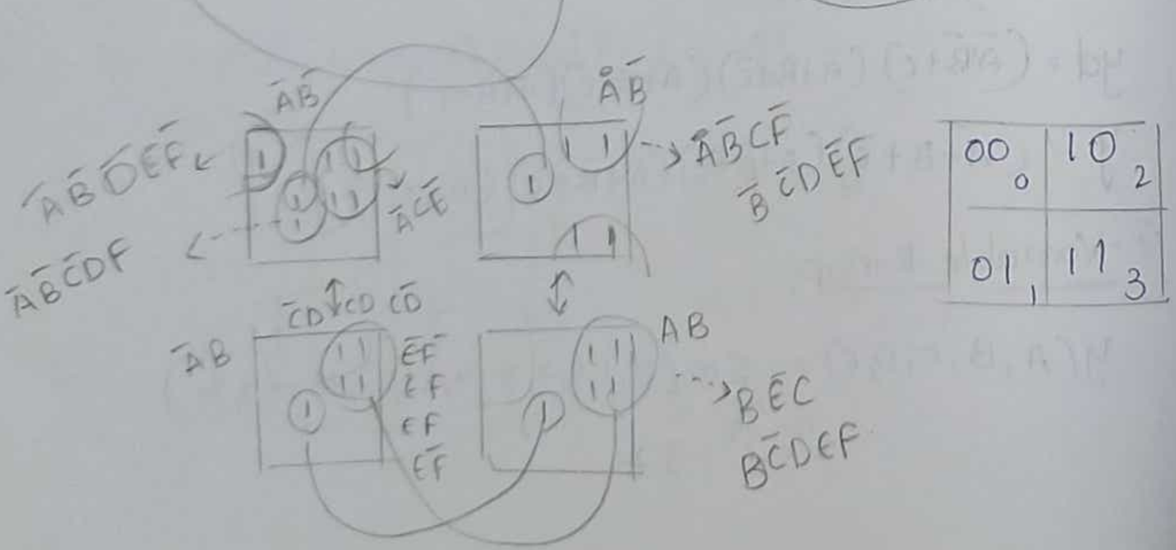
6 Variable k-Map

$f(A,B,C,D,E,F) = \Sigma m(0,5,7,8,9,12,13,23,24,25,28,29,37,40,42,44,46,55,56,57,60,61)$

$n = 6$

$2^n = 2^6 = 64$





Q. The function $f(A,B,C,D) = \Sigma m(3,4,5,7,9,11,13,15)$ is no. of independent variables of

a) B b) C c) A & C d) D.

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

$BD + AD = y$

Q) The standard sum of products of functions of $y = A + \bar{B}C$ is expressed as

a) $\Sigma m(1,4,5,6,7) + d(2,3)$

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

c) $\Sigma m(0,2,3) + d(1,4,5,6,7)$

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

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

Q) Consider the following Boolean function of four variables

$f(w,x,y,z) = \Sigma m(1,3,4,6,9,11,12,14)$

a) Independent of 1 variable

b) 2

c) 3

d) Dependent of all "

	$\bar{w}\bar{x}$	$\bar{w}x$	wx	$w\bar{x}$
$\bar{y}\bar{z}$	0	1 4	1 12	8
$\bar{y}z$	1 1	5	13	1 9
yz	1 3	7	15	1 11
$y\bar{z}$	2	1 6	1 14	10

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

HW

Q) $f(A,B,C,D) = \Sigma m(1,4,5,9,11,12)$

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

b) $AB\bar{C} + ACD + \bar{B}\bar{C}D$

c) $AC\bar{D} + \bar{A}\bar{B}C + A\bar{C}\bar{D}$

d) $\bar{A}BD + AC\bar{D} + BC\bar{D}$

e) none of these

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

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

11/2/22.

Quine Maccasky Method

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

Step1: Represent given no. in Binary

Number (N)	A B C D
0	0 0 0 0
1	0 0 0 1
3	0 0 1 1
7	0 1 1 1
8	1 0 0 0
9	1 0 0 1
11	1 0 1 1
15	1 1 1 1

Step2: Form a group wi based on no. of one's.

Group	Term	A B C D
group-0	0	0 0 0 0
group-1	(1) (8)	0 0 0 1 1 0 0 0
group-2	(3) (9)	0 0 1 1 1 0 0 1
group-3	7) 11	0 1 1 1 1 0 1 1
group-4	15	1 1 1 1

Step3: with the above mentioned datatable we need to match pair with one bit difference

Group	Pair	ABCD
group0	(0-1) (0-8)	000_ _000
group1	(1-3) (1-9) (8-9)	00_1 _001 100_
group2	(3-7) (3-11) (9-11)	0_11 _011 10_1
group3	(7-15) (11-15)	_111 1_11

Step4:

Group	pair	ABCD
group0	(0-1-8-9) (0-8-1-9)	_00_ _00_
group1	(1-3-9-11) (1-9-3-11)	_0_1 _0_1
group2	(3-7-11-15) (3-11-7-15)	__11 __11

Step5:

$$\left.\begin{array}{l} \text{group0} \rightarrow \bar{B}\bar{C} \\ \text{group1} \rightarrow A\bar{B}D \\ \text{group2} \rightarrow CD \end{array}\right\} \text{These are prime implicants}$$

Step 6:

Prime Implicants	Min term	0	1	3	7	8	9	11	15
$\bar{B}\bar{C}$	0-1-8-9	⊗	×			⊗	×		
$\bar{B}D$	1-3-9-11		×	×			×	×	
CD	3-7-11-15			×	⊗			×	⊗

Round up 1 Mark's and draw horizontal lines

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

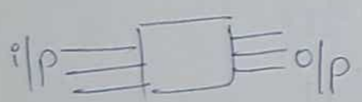
Verify with K-Map

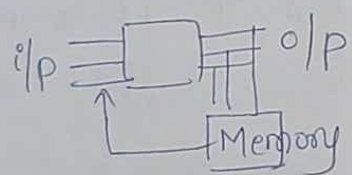
CD \ AB	$\bar{A}\bar{B}$			$A\bar{B}$
$\bar{C}\bar{D}$	1			1
$\bar{C}D$	1			1
CD	1	1	1	1

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

Combinational circuit and Sequential circuit

Combinational ckt	Sequential ckt
- o/p is only depends on present i/p	- o/p is depend on present i/p and past o/p





- No feedback
- No memory

It has either feedback/memory

Ex:- Half Adder / Full adder
Half subtractor / Full subtractor
MUX / DEMUX
Encoder / Decoder
code Counter

Ex:- C
Flipflop

Designing of combinational ckt

i/p ═ [ ] ═ o/p

Step1: Determine and define total inputs and total outputs of the circuit

Step2: Make truth table that defines relationship in between input and output

Step 3: Determine boolean equation using kmap

Step4: Based on boolean equation we can form circuit

① The minimum function that can detect "division of 2" with 8421 BCD [$D_8 D_4 D_2 D_1$] is given by ——

	A	B	C	D	
	D_8	D_4	D_2	D_1	Y (Divisible by 2)
0 -	0	0	0	0	1
-	0	0	0	1	0
2 -	0	0	1	0	1
3 -	0	0	1	1	0
4 -	0	1	0	0	1
5 -	0	1	0	1	0
6 -	0	1	1	0	1
7 -	0	1	1	1	0
8 -	1	0	0	0	1
9 -	1	0	0	1	0
10 -	1	0	1	0	- X
⋮					
15 -	1	1	1	1	- X

D_2D_1 \ D_8D_4	00	01	11	10	
00	1	1	X	1	$\bar{D}_2\bar{D}_1$
01	0	0	X	0	
11	0	0	X	X	
10	1	1	X	X	$D_2\bar{D}_1$

$Y = \bar{D}_1$

② The minimal function that can detect "divisible by 3" with 8421 BCD [$D_8 D_4 D_2 D_1$] is given by ——

$0 \rightarrow \bar{D}_1\bar{D}_2\bar{D}_4\bar{D}_8$

$11, 13, 15 \rightarrow D_1 D_8$

$3, 11 \rightarrow D_1 D_2 \bar{D}_4$

$6, 14 \rightarrow \bar{D}_1 D_2 D_4$

	$\bar{D}_8\bar{D}_4$	$\bar{D}_8 D_4$	$D_8 D_4$	$D_8\bar{D}_4$
$\bar{D}_2\bar{D}_1$	1_0	0_4	X_{12}	0_8
$\bar{D}_2 D_1$	0_1	0_5	X_{13}	1_9
$D_2 D_1$	1_3	0_7	X_{15}	X_{11}
$D_2\bar{D}_1$	0_2	1_6	X_{14}	X_{10}

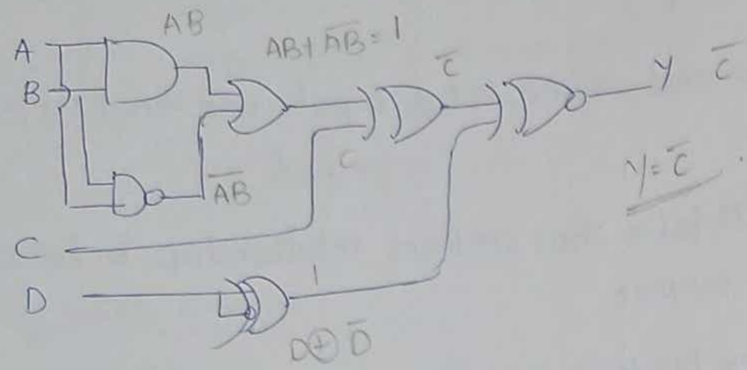
$9, 11, 13, 15 \rightarrow D_1 D_8$

or

$9, 11 \rightarrow D_1 \bar{D}_4 D_8$

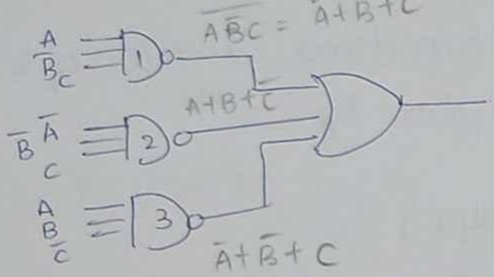
$$Y = D_1D_8 + \bar{D}_1D_2D_4 + D_1D_2\bar{D}_4 + \bar{D}_1\bar{D}_2\bar{D}_4\bar{D}_8$$

③



$Y = \bar{C}$

Redundant gate



Redundant

gate 1 = $\bar{A} + B + \bar{C}$ → X no need, all we have in others

gate 2 = $A + B + \bar{C}$

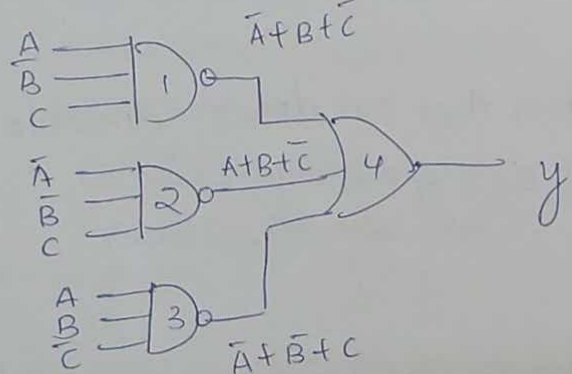
gate 3 = $\bar{A} + \bar{B} + C$

→ Gate 1 is Redundant

We can get the same output by removing that gate

12/2/22

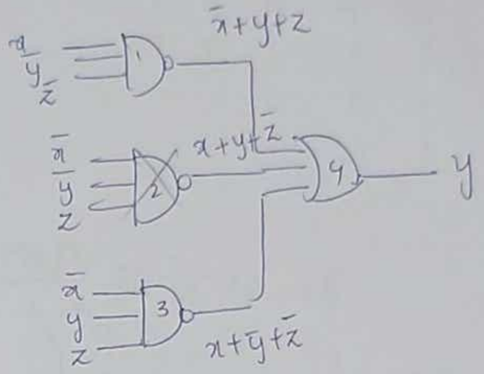
① What is redundant gate in the given combinational ckt?



gate 1 : $\overline{A\bar{B}C} = \bar{A}+B+\bar{C}$ — Redundant gate

gate 2 : $\overline{\bar{A}\bar{B}C} = A+B+\bar{C}$

gate 3 : $\overline{AB\bar{C}} = \bar{A}+\bar{B}+C$

② 

Find the redundant gate in the following combinational circuit

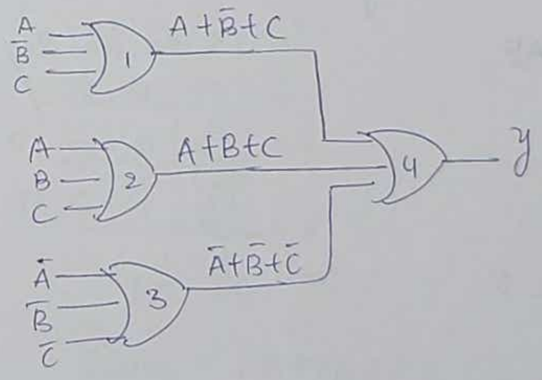
Why it is a combinational ckt. Becoz no feedback

gate 1 : $\overline{x\bar{y}\bar{z}} = \bar{x}+y+z$

gate 2 : $\overline{\bar{x}\bar{y}z} = x+y+\bar{z}$ ⟶ Redundant gate

gate 3 : $\overline{\bar{x}yz} = x+\bar{y}+\bar{z}$

Gate 2 is Redundant

③ 

gate 1 : $A+\bar{B}+C$ ✓ Redundant gate

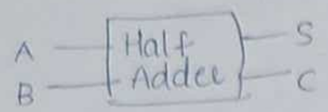
gate 2 : $A+B+C$

gate 3 : $\bar{A}+\bar{B}+\bar{C}$

Gate 1 is Redundandt

Half Adder

> Half adder – two bit addition ckt



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

K-Map of sum

B \ A	$\bar{A}$ 0	A 1
$\bar{B}$ 0	0	1
B 1	1	0

$A\bar{B}$

$\bar{A}B$

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

$= A \oplus B$

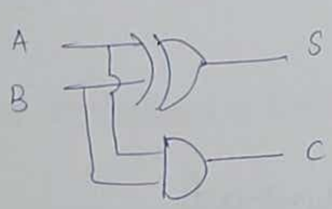
Carry k-Map

B \ A	0	A 1
0	0	0
B 1	0	1

AB

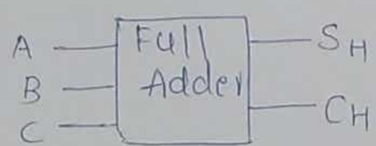
$Carry = A \cdot B$

Half Adder Ckt



Full Adder ckt

> Full Adder – three bit addition ckt



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

K-Map,

Sum,

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

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

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

$$Sum = \bar{A}\bar{B}C + \bar{A}B\bar{C} + ABC + A\bar{B}\bar{C}$$
$$= \bar{A}(\bar{B}C + B\bar{C}) + A(BC + \bar{B}\bar{C})$$
$$= \bar{A}(B \oplus C) + A(B \odot C)$$
$$= \bar{A}(B \oplus C) + A(\overline{B \oplus C})$$
$$= A \oplus B \oplus C$$

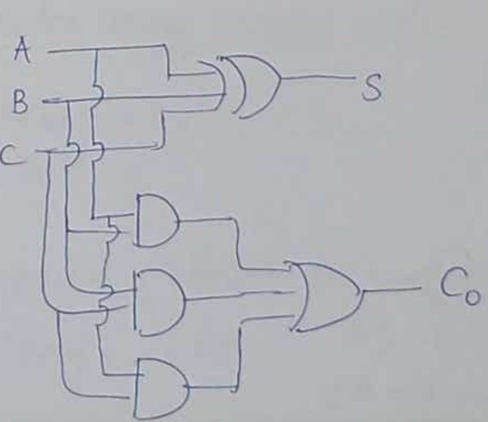
Carry,

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

$BC \quad AB \quad AC$

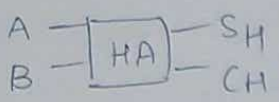
$$Carry_o = CB + AB + AC$$

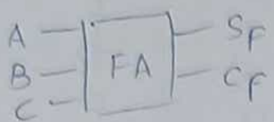
Ckt



Full Adder

Full Adder ckt using Half Adder



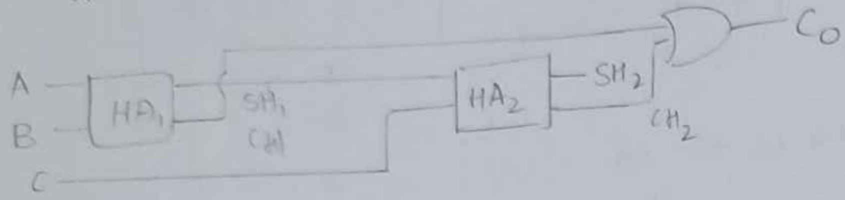


$S_H = A \oplus B$

$C_H = AB$

$S_f = A \oplus B \oplus C$

$C_F = AB + BC + CA$



$SH_1 = A \oplus B$

$CH_1 = AB$

$SH_2 = SH_1 \oplus C$

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

$CH_2 = SH_1 \cdot C$

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

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

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

$C_0 = CH_1 + CH_2$

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

$= AB + BC + AC$

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

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

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

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

$= AB + C(A + B)$

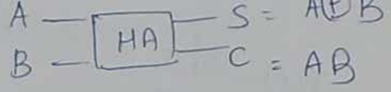
$= AB + AC + BC$

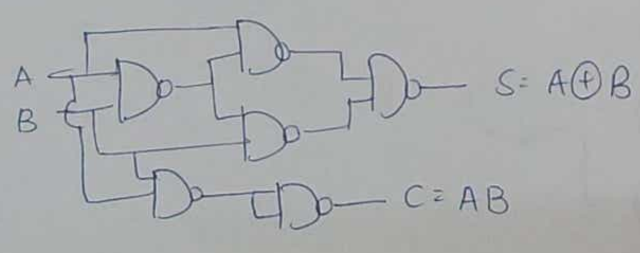
K.Map

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

$AB + BC + CA$

Half Adder by Nand gate



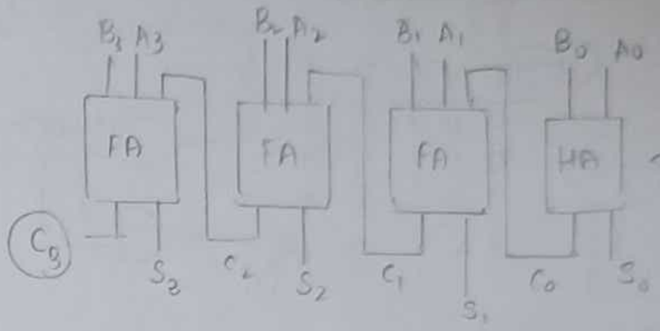


4-Bit parallel Adder

3 bit (FA)

	C_2	C_1	C_0	
A	A_3	A_2	A_1	A_0
B	B_3	B_2	B_1	B_0
C, S	S_3	S_2	S_1	S_0

2bit (HA)



→ we can Replace this HA by FA as C=0.

BCD Adder

BCD data

A → BCD data
B → BCD data

CS [if S<9]
+6 [if S>9]

0110 = 6

C	S_3	S_2	S_1	S_0	
0	0	0	0	0	– 0
0	0	0	0	1	– 1
⋮					
0	1	0	0	1	– 9
⋮					
0	1	1	1	1	– 15
1	0	0	0	0	– 16
1	0	0	0	1	– 17
1	0	0	1	0	– 18

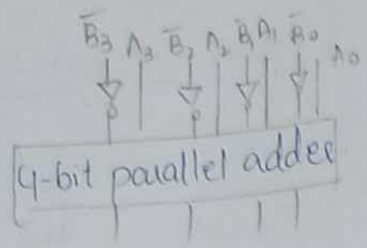
0 to 9: S<9

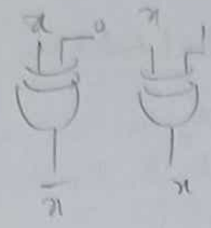
15 to 17: S>9, +6 (Cy=0 / Cy=1)

1's subtractor using parallel adder

A – binary = A_B
B – 1's comp $\overline{B}$

if carry 1 (Result = +ve) R = C + Cy
carry 0 (Result –ve) R = 1's comp of C





Half Subtractor - 2 bit subtraction

x
y
B Diff
Borrow

x	y	D	B
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

Difference

y \ x	$\bar{x}$	x
$\bar{y}$	0	1
y	1	0

$x\bar{y}$

$\bar{x}y$

$D = x\bar{y} + \bar{x}y$
$= x \oplus y$

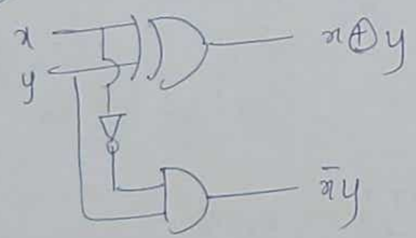
Borrow

	$\bar{x}$	x
$\bar{y}$	0	0
y	1	0

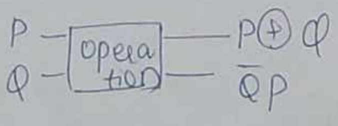
$\bar{x}y$

x-y	y-x
$B = \bar{x}y$	$B = x\bar{y}$

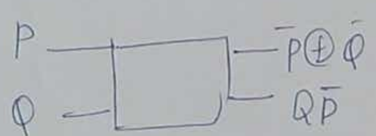
Ckt



Q9



Operation = Q-P

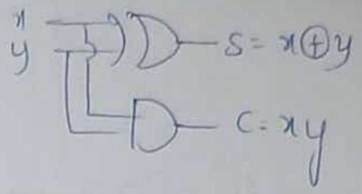


Operation = P-Q

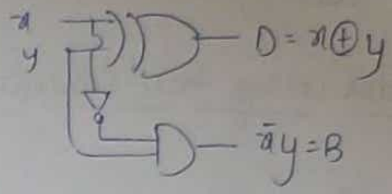
$\bar{P} \oplus \bar{Q} = \bar{P}\bar{\bar{Q}} + \bar{\bar{P}}\bar{Q}$
$= Q\bar{P} + P\bar{Q}$
$= P \oplus Q$

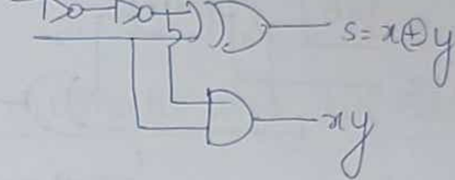
Half Adder using half subtractor

Half Adder

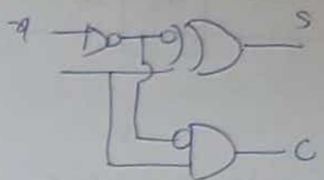


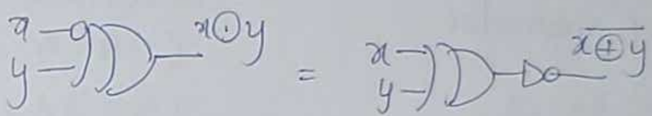
Half Subtractor





$\Rightarrow$



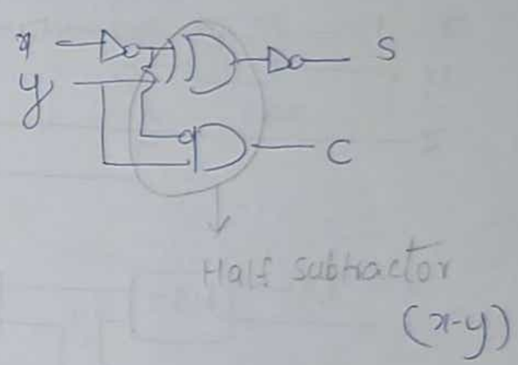


$\Downarrow$

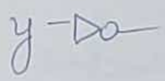
$= \bar{x} \oplus y$

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

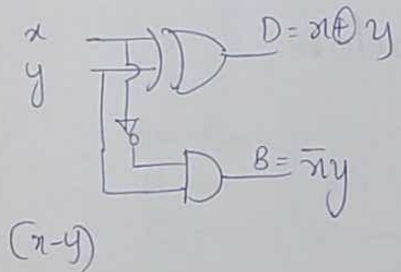
$= xy + \bar{x}\bar{y} = x \odot y$

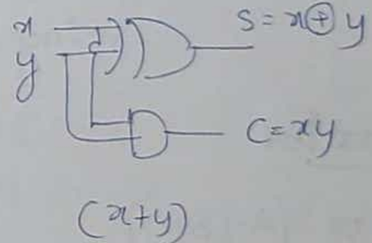


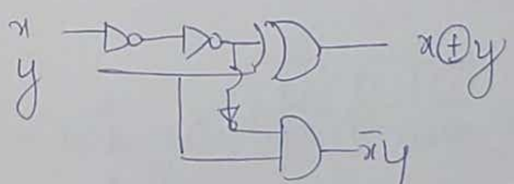
if y-x



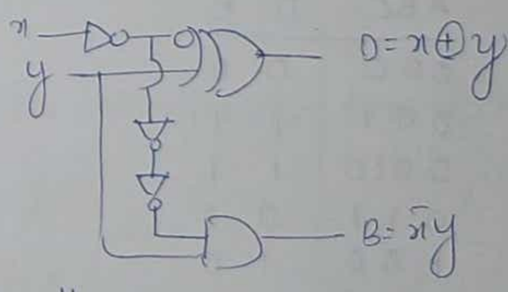
Half subtractor using Half Adder

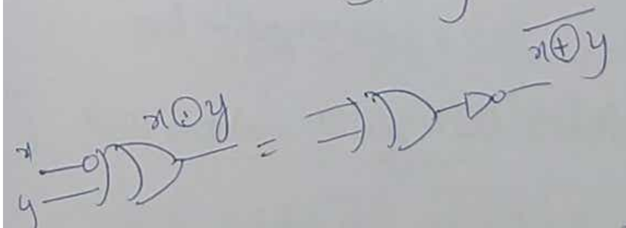




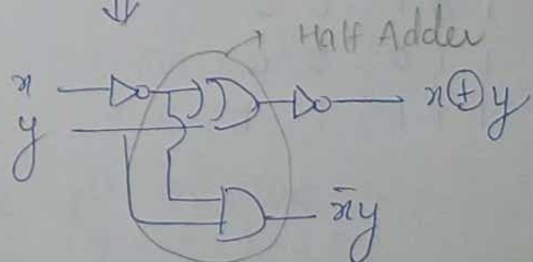


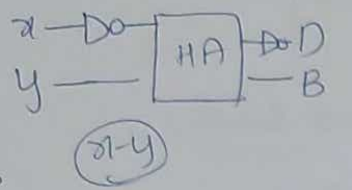
$\Rightarrow$

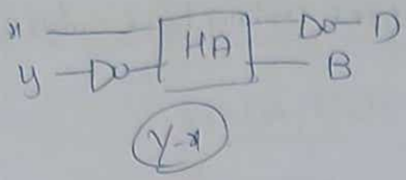




$\Downarrow$

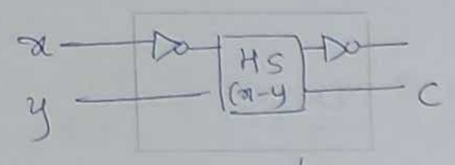




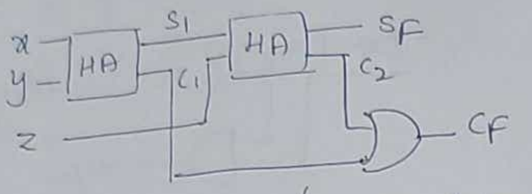


14/2/22

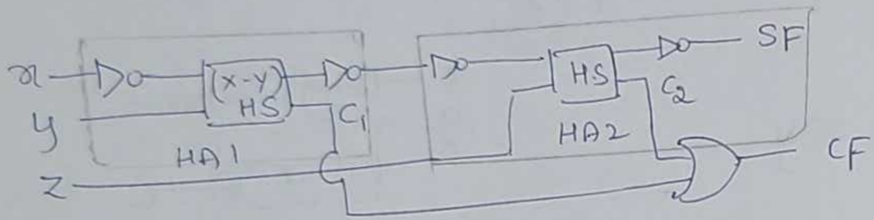
Full Adder using Half Subtractor

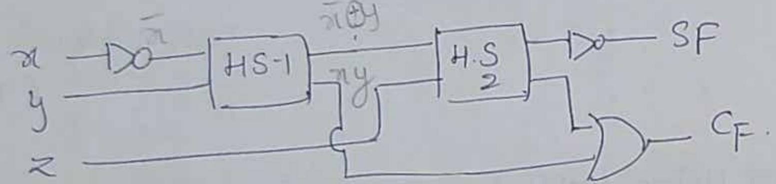


Half adder
using half subtractor



Full adder





Full Adder
using Half subtractor

Full Subtractor

(A-B-C) or (A-(B+C))

ABC	D	B
000	0	0
001	1	1
0010	1	1
011	0	1
100	1	0
101	0	0
110	0	0
111	1	1

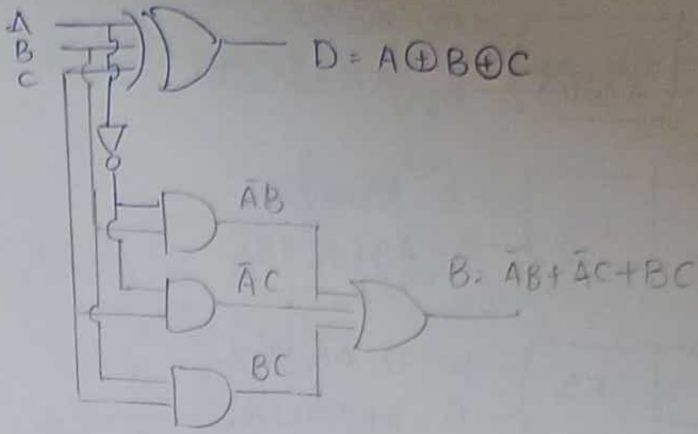
C \ AB				
	0	1	0	1
	1	0	1	0

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

$= A \oplus B \oplus C$

$\bar{C}$	0	1	0	0
C	1	1	1	0

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



Full subtractor using full Adder

Full subtractor

$A - (B + C)$

$D = A \oplus B \oplus C$

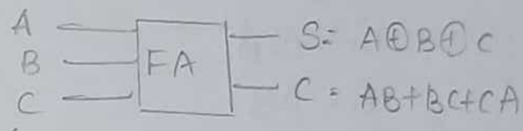
$B = \bar{A}B + \bar{A}C + BC$

Full Adder

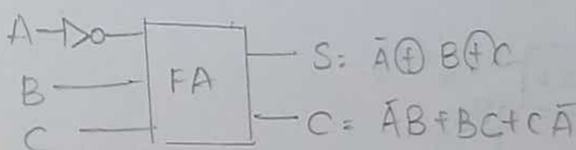
$A + B + C$

$S = A \oplus C \oplus B$

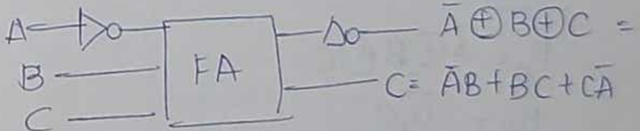
$C = AB + BC + CA$



⇓



⇓



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

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

$\bar{A}[\overline{\bar{B} \odot \bar{C}}] + \bar{\bar{A}}[\bar{B} \odot \bar{C}]$

$\bar{A}[\overline{B \oplus C}] + \bar{\bar{A}}[B \oplus C]$

$A \oplus B \oplus C$

→ $\overline{XOR}$ gate has inverter transfer characteristics from o/p to i/p and viceversa

Full Adder using Full subtractor

Full Adder

$= A + B + C$

$S = A \oplus B \oplus C$

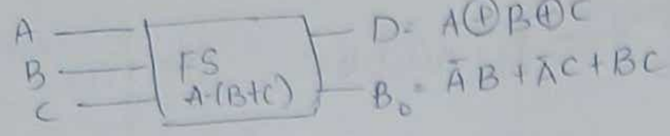
$C = AB + BC + CA$

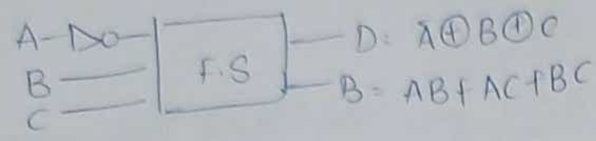
Full subtractor

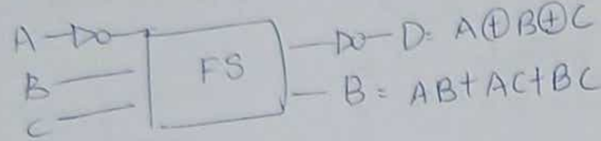
$= A - (B + C)$

$D = A \oplus B \oplus C$

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







XOR gate characteristics to transfer inverter from i/p to o/p and vice versa

Full substractor using half substractor

Full substractor

$A-(B+C)$

$D = A \oplus B \oplus C$

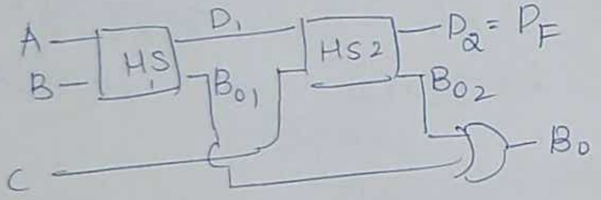
$B_o = \bar{A}B + C\bar{A} + BC$

Half subtractor

$A-B$

$D = A \oplus B$

$B = \bar{A}B$



$D_1 = A \oplus B$

$B_{o_1} = \bar{A}B$

HS_2

$D_2 = A \oplus B \oplus C$

$B_{o_2} = \bar{D_1}C$

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

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

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

$B_{o_1} + B_{o_2} = \bar{A}B + ABC + \bar{A}\bar{B}C$

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

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

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

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

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

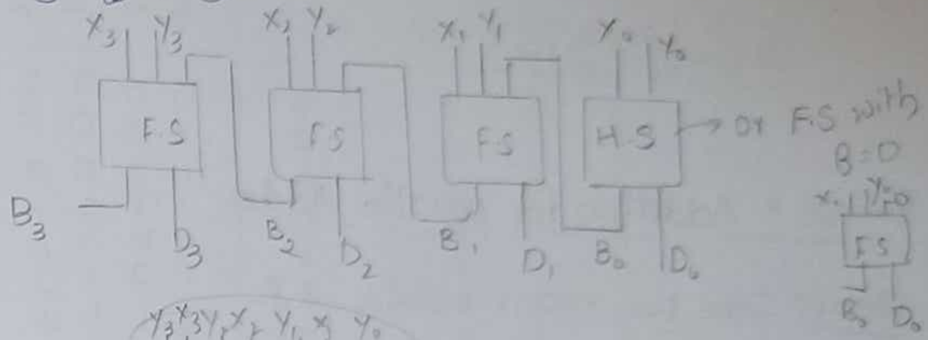
$\bar{A}\bar{B}$	$\bar{A}B$	AB	$A\bar{B}$
	1		
1	1	1	

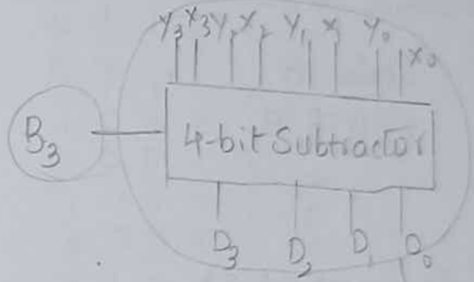
$\bar{A}\bar{B} + \bar{A}C + BC$

Parallel subtractor using Full subtractor

x
$-Y$
$B_0\ D$ — full subtractor

$B_2\ B_1\ B_0$
$X_3\ X_2\ X_1\ X_0$ — half subtractor
$Y_3\ Y_2\ Y_1\ Y_0$
B_3 — $D_3\ D_2\ D_1\ D_0$

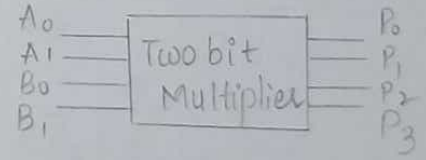




what it signifies ?

16/2/22

Two bit Multiplier using Half Adder



$A_0\ BA_1$
$B_0\ \ B_1$
$C_1\ \ B_1A_0\ \ B_0A_1$
$C_2\ B_0A_0\ \ B_0A_1\ \ \times$

$A_1\ A_0$
$B_1\ B_0$
$C_1\ \ B_0A_1\ \ B_0A_0$
$C_2\ \ B_1A_1\ \ B_1A_0\ \ \times$

$P_0 = B_0A_0$

$P_1 = HA_1\ [B_0A_1, B_1A_0)$

$P_2 = HA_2\ [C_1, A_1B_1]$

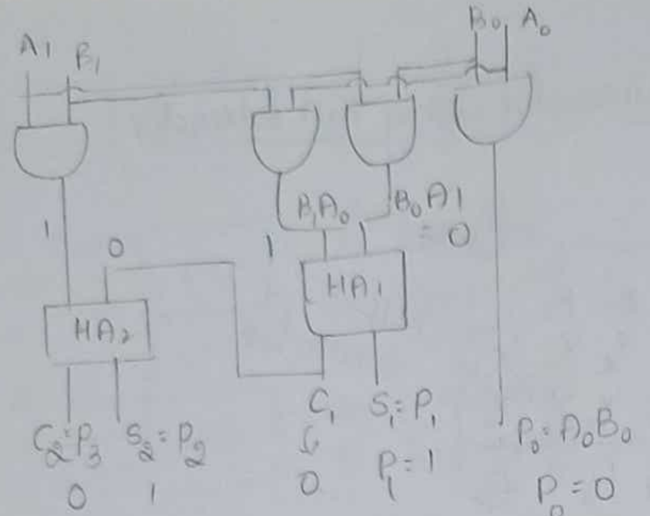
$P_3 = C_2 - HA_2$

11
10

00
11

0110

$B_0 = 0$ $B_1 = 1$
$A_0 = 1$ $A_1 = 1$



Verified, : $\boxed{0110}$

Excess 3 Addition For parallel Adder

Step 1: Take two excess 3 data A, B.

Data A | Data B
$A_3|A_2|A_1|A_0$ | $B_3|B_2|B_1|B_0$

Step-2: Add two A & B

$A+B = S$ (Assume as S)

$A+B = S$ (carry c)

check C

$C=0$ or $C=1$

$S-3$ $S+3$

0011

-3

-0011

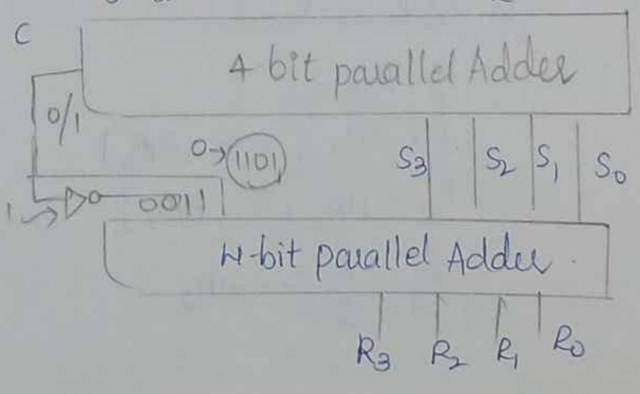
1's complement

1100 + 1

2's — 1101

Data A Data B

$B_3|A_2|A_1|A_0$ $B_3|B_2|B_1|B_0$.

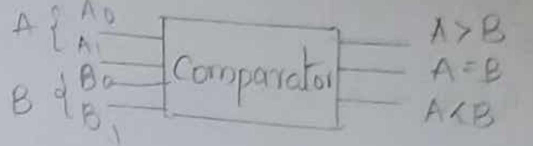


HW

Q) Design a reversible Excess-3 adder and subtractor

↳ try

2-bit comparator



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

A > B

K-map

B_1B_0 \ A_1A_0		$\bar{A}_1A_0$	A_1A_0	$A_1\bar{A}_0$
$\bar{B}_1\bar{B}_0$	0_0	1_4	1_{12}	1_8
$\bar{B}_1B_0$	0_1	0_5	1_{13}	1_9
	0_3	0_7	0_{15}	0_{11}
$B_1\bar{B}_0$	0_2	0_6	1_{14}	0_{10}

$A_1\overline{B_1} + A_0\overline{B_1}\,\overline{B_0} + A_1A_0\overline{B_0}$

A = B

B_1B_0 \ A_1A_0			
1			
	1		
		1	
			1

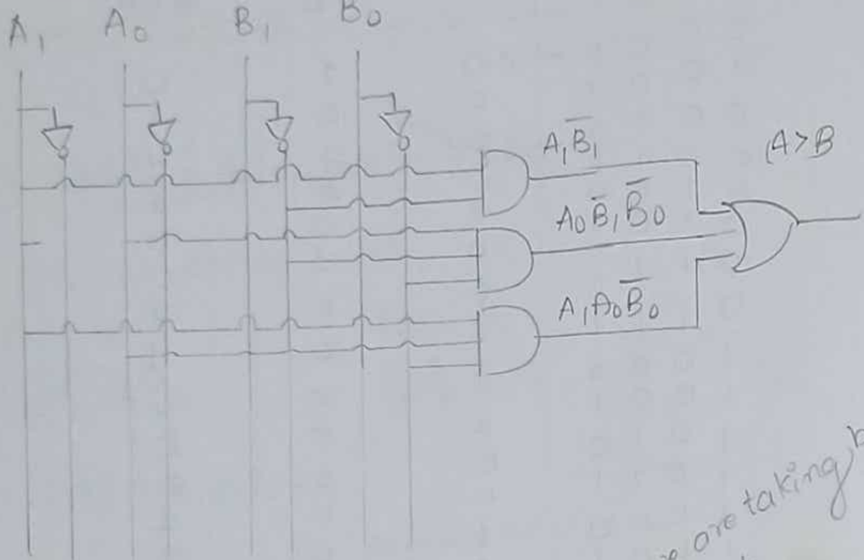
$\bar{A}_1\bar{A}_0\bar{B}_0\bar{B}_1 + \bar{A}_1A_0\bar{B}_1B_0 + A_1A_0B_1B_0$
$+ A_1\bar{A}_0B_1\bar{B}_0$

A<B

A<B

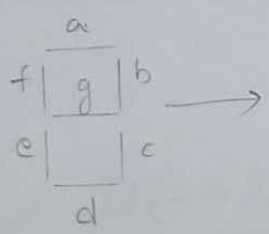
B_1B_0 \ A_1A_0				$A_1\bar A_0$
$\bar B_1 B_0$	1			
$B_1 B_0$	1	1		1
$B_1 \bar B_0$	1	1		

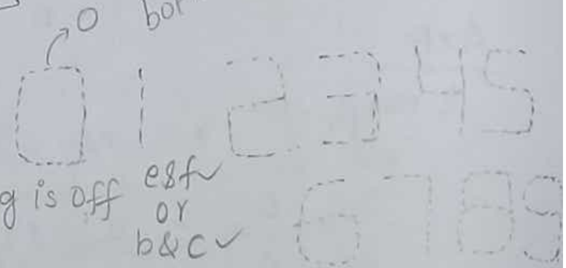
$$\{ \bar A_1 B_1 + \bar A_1 \bar A_0 B_0 + \bar A_0 B_1 B_0 \}$$



HW Represent A=B, A<B

Seventh segment Display Decoder [now we are taking b,c both are ckt



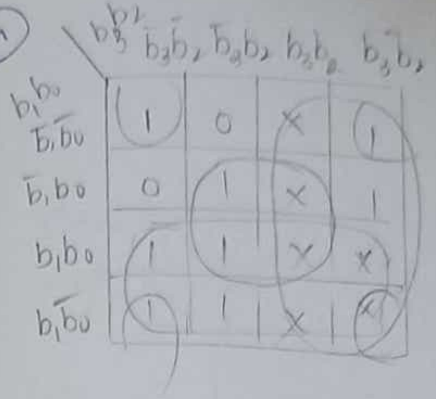


g is off e&f ✓ or b&c ✓

Digits	b_3	b_2	b_1	b_0		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	01	0	1	1

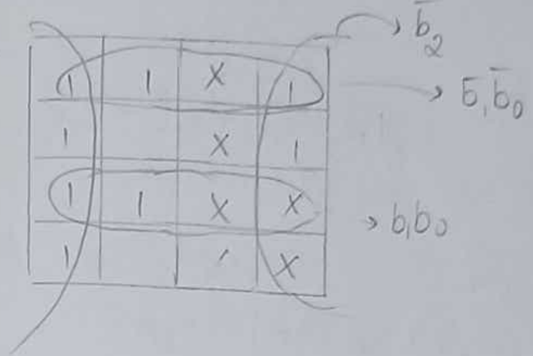
after that, Don't care

(a)



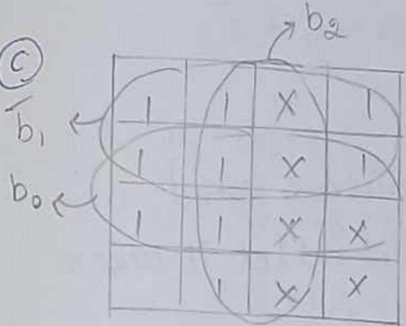
$b_1 + b_3 + b_2 b_0 + \bar{b}_0\bar{b}_2$

(b).



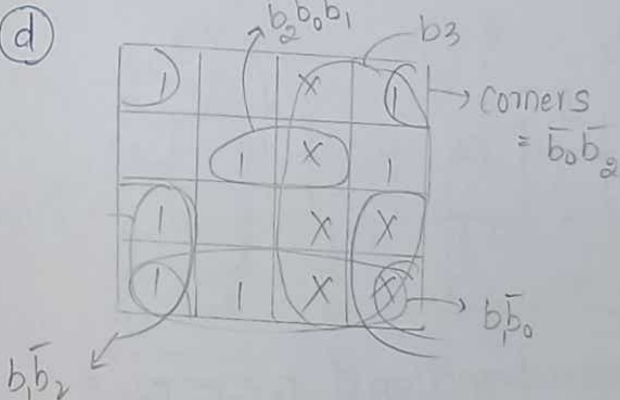
$\bar{b}_2 + b_1b_0 + \bar{b}_1\bar{b}_0$

(c)



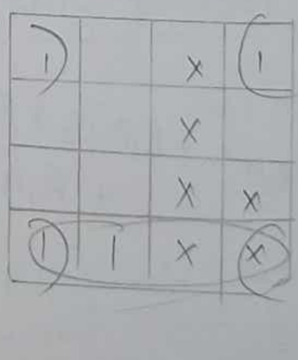
$b_0 + \bar{b}_1 + b_2$

(d)



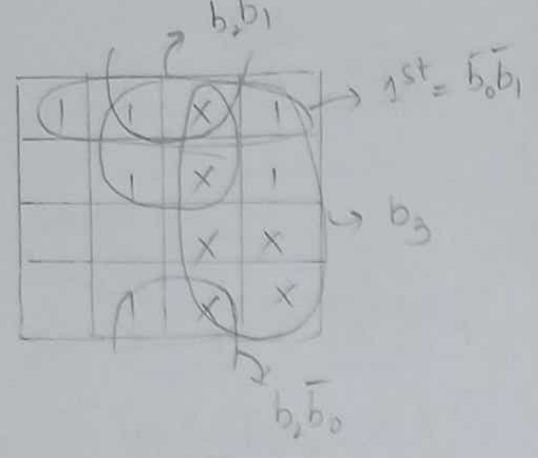
$b_3 + \bar{b}_0\bar{b}_2 + b_1\bar{b}_0 + b_2b_0\bar{b}_1 + b_1\bar{b}_2$

(e)



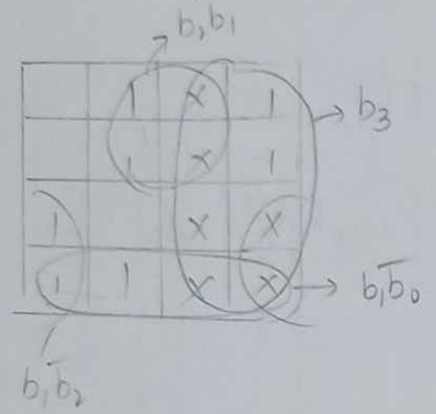
$b_1\bar{b}_0 + \bar{b}_0\bar{b}_2$

(f)



$b_3 + \bar{b}_0\bar{b}_1 + b_2\bar{b}_1 + b_2\bar{b}_0$

(g)

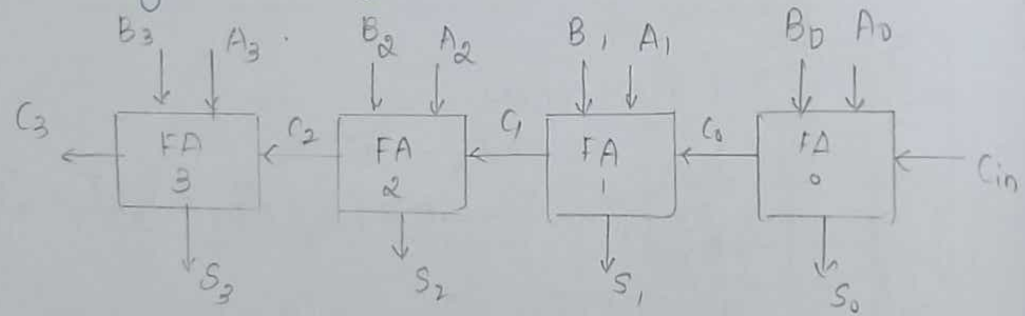


$b_3 + b_1\bar{b}_0 + b_2\bar{b}_1 + b_1\bar{b}_2$

Carry Look Ahead adder (CLA)

> CLA adder is faster in terms of operational speed

> By Full adder when we have parallel adder, it takes propagation delay to have output



At first we need to design founction of C_3, C_2, C_1, C_0 wrt C_{in},

Full adder $B_1 A_1 C_0$ and C_1

B_1	A_1	C_0	C_1
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	~~0~~ 1
1	1	1	1

$C_1 = B_1A_1C_0 + B_1A_1\bar{C}_0 + B_1\bar{A}_1C_0 + \bar{B}_1A_1C_0$

$C_1 = B_1A_1(C_0+\bar{C}_0) + C_0(B_1\bar{A}_1 + \bar{B}_1A_1)$

$= B_1A_1 + C_0(B_1 \oplus A_1)$ $\quad$ $\cancel{B_1A_1}$

$C_i = B_iA_i + C_{i-1}(B_i \oplus A_i)$

$\boxed{C_i = G_i + P_iC_{i-1}}$ General formula $G_i = A_iB_i$, $P_i = A_i \oplus B_i$

$i=0 \quad C_0 = G_0 + P_0C_{in}$

$i=1 \quad C_1 = G_1 + P_1C_0$

$= G_1 + P_1[G_0 + P_0C_{in}]$

$= G_1 + P_1G_0 + P_1P_0C_{in}$

$i=2 \quad C_2 = G_2 + P_2C_1$

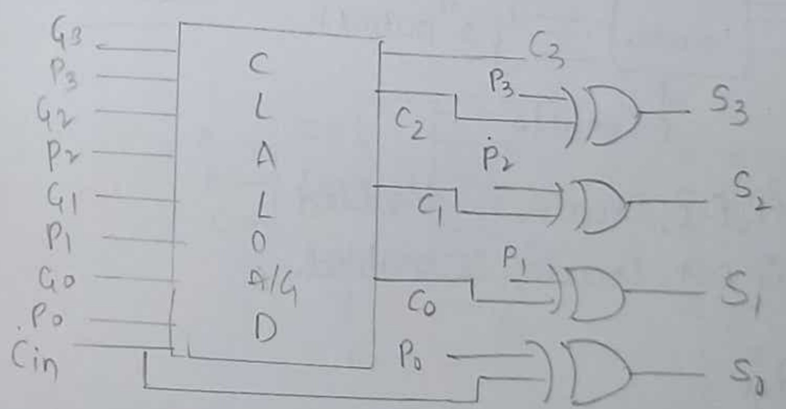
$= G_2 + P_2[G_1 + P_1G_0 + P_1P_0C_{in}]$

$= G_2 + P_2G_1 + P_2P_1G_0 + P_2P_1P_0C_{in}$

$i=3 \quad C_3 = G_3 + P_3C_2$

$= G_3 + P_3[G_2 + P_2G_1 + P_2P_1G_0 + P_2P_1P_0C_{in}]$

$= G_3 + P_3G_2 + P_3P_2G_1 + P_3P_2P_1G_0 + P_3P_2P_1P_0C_{in}$



Even parity generator / odd parity generator

i/p [EPG] o/p $\quad$ even parity generates or parity = 1 when no. of 1's at i/p is odd.

i/p [OPG] o/p $\quad$ odd parity generator = 1 when no. of 1's at input are even

even parity
↳ it makes no. of 1's = even

b_3	b_2	b_1	b_0	o/p
0	0	0	0	0
0	0	0	1	1
0	0	1	0	1
0	0	1	1	0
0	1	0	0	1
0	1	0	1	0
0	1	1	0	0
0	1	1	1	1
1	0	0	0	1
1	0	0	1	0
1	0	1	0	0
1	0	1	1	1
1	1	0	0	0
1	1	0	1	1
1	1	1	0	1
1	1	1	1	0

b_1b_0 \ b_3b_2				
	0	1		1
	1		1	
	0	1		1
	1		1	

even parity $= b_0 \oplus b_1 \oplus b_2 \oplus b_3$

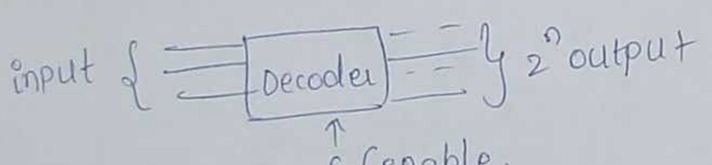
odd parity $= \overline{b_0 \oplus b_1 \oplus b_2 \oplus b_3}$

$= b_0 \odot b_1 \odot b_2 \odot b_3$

15/2/22

Decoder Basics and 2x4 Decoder

Decoder decodes n input to 2^n outputs

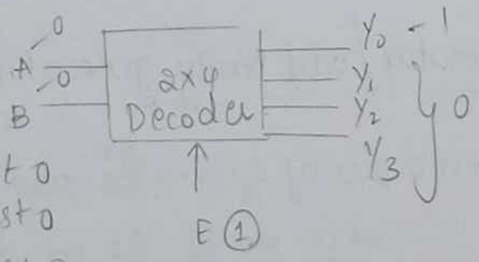


if E = 0, Decoder is disabled.

if E = 1, Decoder is enabled

2x4 Decoder

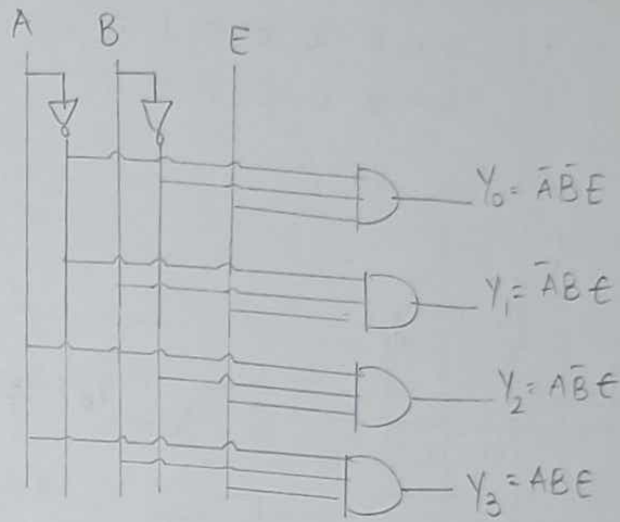
A B
0 0
0 1 → $Y_1 = 1$, rest 0
1 0 → $Y_2 = 1$, rest 0
1 1 → $Y_3 = 1$, rest 0



A	B	E	Y_0	Y_1	Y_2	Y_3
×	×	0	0	0	0	0
0	0	1	1	0	0	0
0	1	1	0	1	0	0
1	0	1	0	0	1	0
1	1	~~0~~1	0	0	0	1

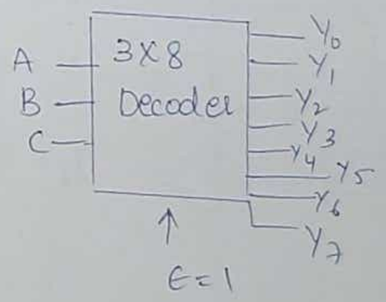
for any A, B

if E=0, no matter what is there in i/p, o/p will be zero.



* Decoder is used for chip selection, in microprocessors, in embedded systems, IC

3×8 Decoder



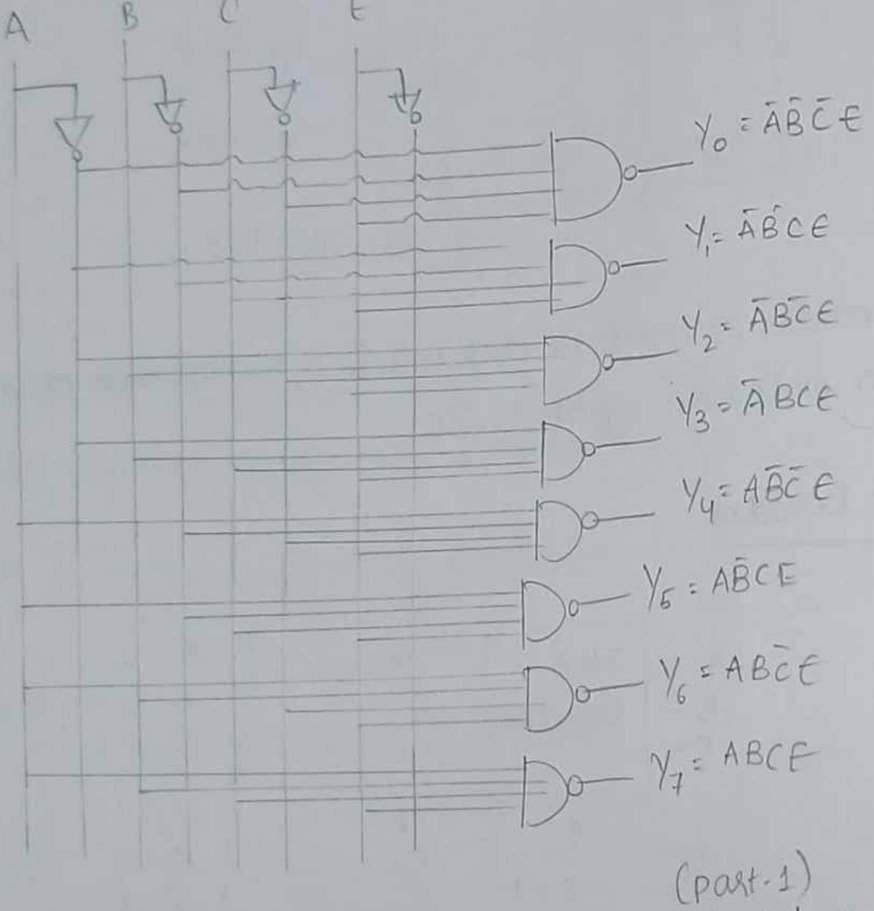
A	B	C	
0	0	0	$Y_0 = 1$
0	0	1	$Y_1 = 1$
0	1	0	$Y_2 = 1$
0	1	1	$Y_3 = 1$
1	0	0	$Y_4 = 1$
1	0	1	$Y_5 = 1$
1	1	0	$Y_6 = 1$
1	1	1	$Y_7 = 1$

E is enable terminal

E=0, Decoder is disabled

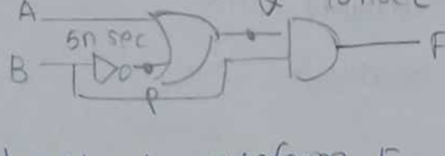
E=1, Decoder is enabled

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



(part-1)

Combinational circuit output waveform with delay at gates:



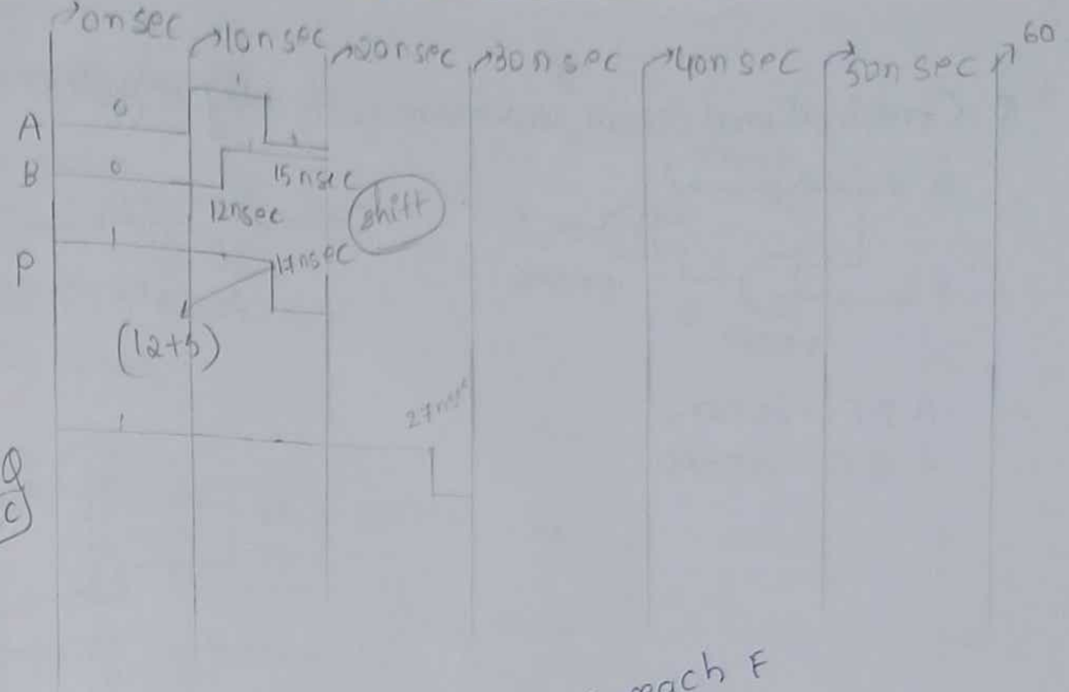
For A → delay: 20nsec

B → delay: 25nsec

1) Find output waveform F

2) Find total delay of given circuit

highest delay path is considered



$\boxed{A+P} = Q$

$\boxed{10nsec}$

Paths to reach F

A→F = 20nsec delay

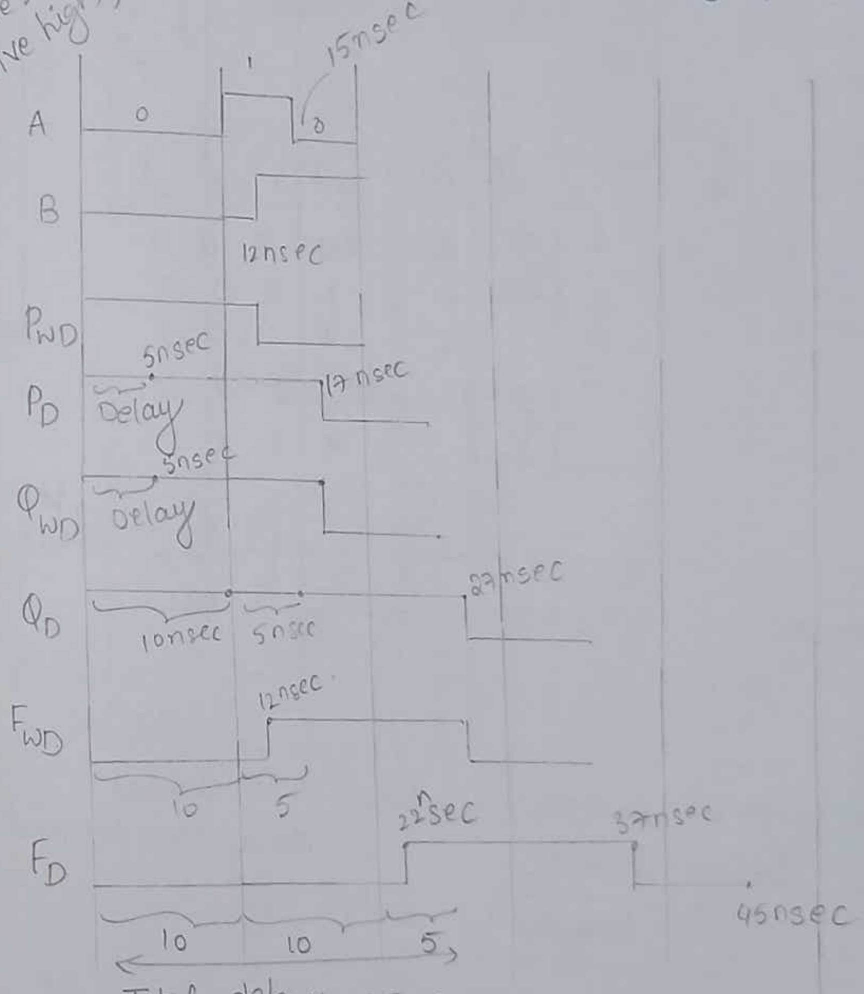
B-Q-F = 25nsec delay

B - F → 10nsec delay

21/2/22

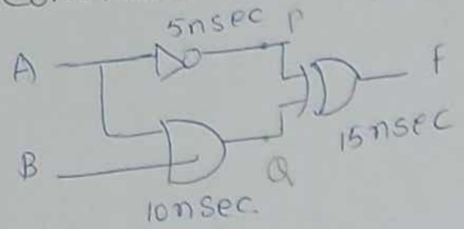
Active low,

Active high,



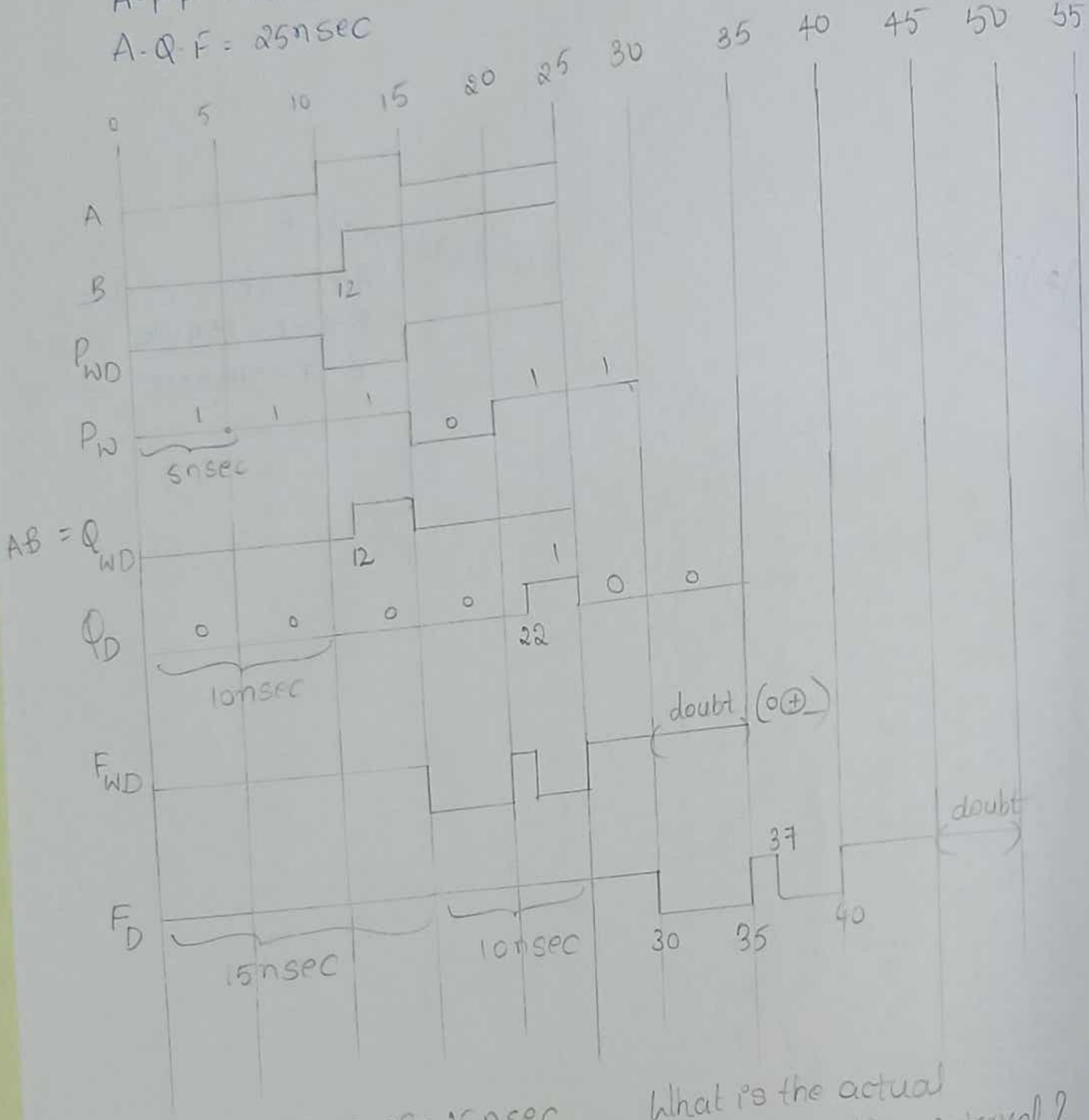
Total delay = 25 nsec.

Q. Combinational circuit o/p waveform with delay at gates



Delay = 25 nsec. ✓ or
+5
30nsec ×

A-P-F = 20nsec
A-Q-F = 25nsec



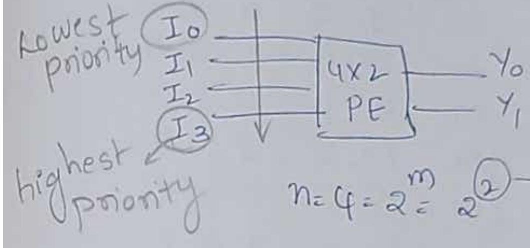
A & B are given as: 25 nsec
Delay - 25 nsec
50 nsec

What is the actual delay time interval?

22/2/22

Priority Encoder

① Basic.
② Working
③ TT (Truth Table)
④ Great



$n = 4 = 2^m = 2^{2} \rightarrow \boxed{m=2}$,

I_0
I_1
I_2
I_3 ↓ from I_0 to I_3 Priority is increasing

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

$\underline{Y_1}$

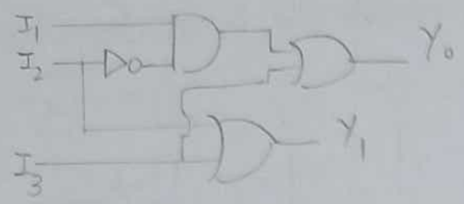
I_1I_0 \ I_3I_2	00	01 $\bar{I_3}I_2$	11 I_3I_2	10 $I_3\bar{I_2}$
00	X	1	1	1
01	0	1	1	1
11 I_1I_0	0	1	1	1
10 $I_1\bar{I_0}$	0	1	1	1

$Y_1 = I_3 + I_2$

$\underline{Y_0}$

	00	01	11	10
00	0000 X	0	1	1
01	0001 0	0	1	1
11	1	0	1	1
10	1	0	1	1

$I_3 + I_1\bar{I_2} = Y_0$



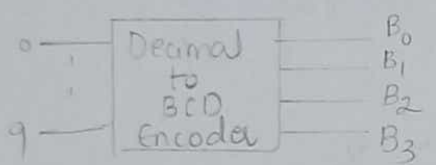
$Y_0 = I_3 + \bar{I}_2 I_1$

$Y_1 = I_3 + I_2$

Decimal to BCD encoder

Decimal [0 to 9]

BCD [0000 to 1101] encoder



I_D	B_3	B_2	B_1	B_0
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	~~0~~1	0	0	0
9	1	0	0	1

(o/p over $B_3\,B_2\,B_1\,B_0$)

$B_3 = 8 + 9$

~~$B_2 = 4 \oplus 5 \oplus 6 \oplus 7$~~

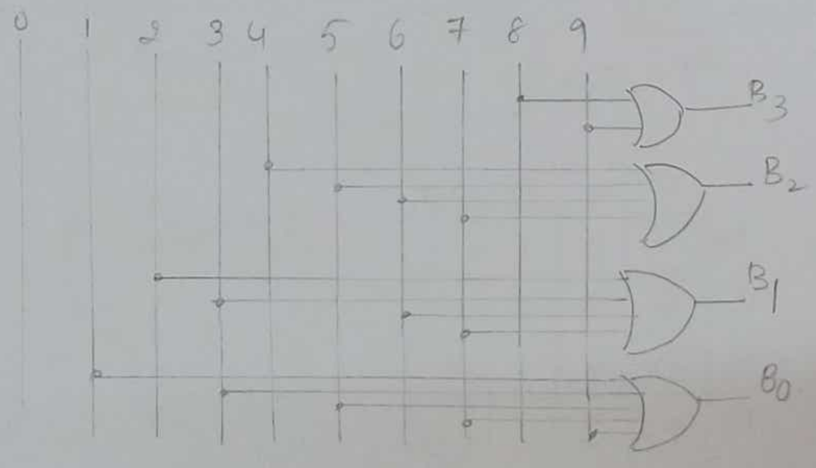
~~$B_1 = 2 \oplus 3 \oplus 6 \oplus 7$~~

~~$B_0 = 1 \oplus 3 \oplus 5 \oplus 7 \oplus 9$~~

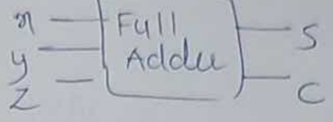
$B_2 = 4 + 5 + 6 + 7$

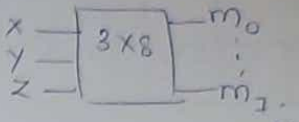
$B_1 = 2 + 3 + 6 + 7$

$B_0 = 1 + 3 + 5 + 7 + 9$.



Fulladder using 3x8 decoder



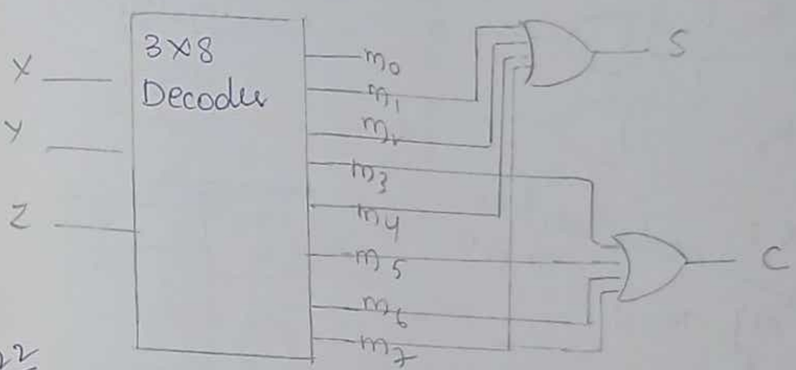


$m = 2^n = 2^3 = 8$

x	y	z	S	C	o/p
0	0	0	0	0	m_0
0	0	1	1	0	m_1
0	1	0	1	0	m_2
0	1	1	0	1	m_3
1	0	0	1	0	m_4
1	0	1	0	1	m_5
1	1	0	0	1	m_6
1	1	1	1	1	m_7

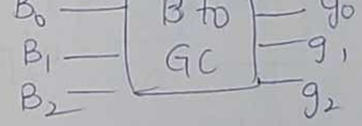
$S = m_1 + m_2 + m_4 + m_7$

$C = m_3 + m_5 + m_6 + m_7$



25/2/22

Binary to Gray code converter



Binary i/p

o/p Gray code

I/p			o/p		
B_2	B_1	B_0	G_2	G_1	G_0
0	0	0	0	0	0
0	0	1	0	0	1
0	1	0	0	1	1
0	1	1	0	1	0
1	0	0	1	1	0
1	0	1	1	1	1
1	1	0	1	0	1
1	1	1	1	0	0

G_2

$B_0 \backslash B_2B_1$	$\bar{B}_2\bar{B}_1$ 00	$\bar{B}_2B_1$ 01	B_2B_1 11	$B_2\bar{B}_1$ 10
$\bar{B}_0$ 0	0	0	1	1
B_0 1	0	0	1	1

$G_2 = B_2$

G_1

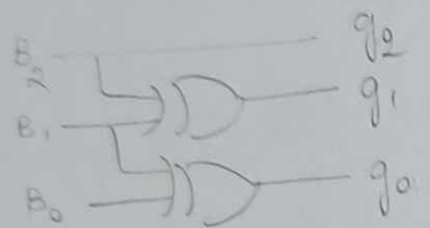
0	1	0	1
0	1	0	1

$G_1 = \bar{B}_2B_1 + B_2\bar{B}_1$
$= B_1 \oplus B_2$

G_0

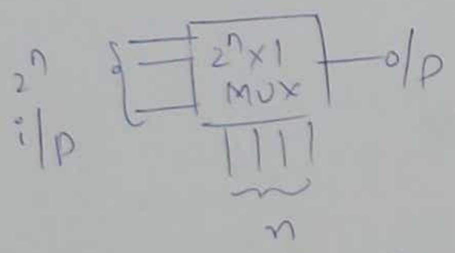
0	1	1	0
1	0	0	1

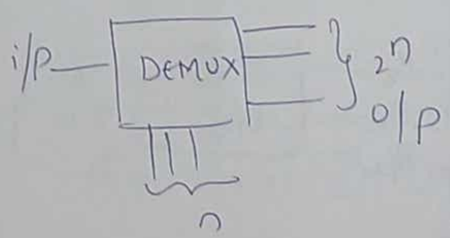
$G_0 = \bar{B}_0B_1 + B_0\bar{B}_1$
$= B_0 \oplus B_1$



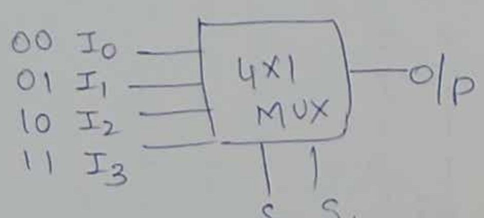
Multiplexer : → selection line or pin

$[2^n \times 1]$ MUX
↑ ↑
i/p o/p





$(4 \times 1) \rightarrow n = 2$



MSI scale
↓
Medium scale integration

① less Complex
② low cost
③ less wire/wiring
} Advantages.

Application: ① Data selecter
② switch
③ combinational circuit

Type :

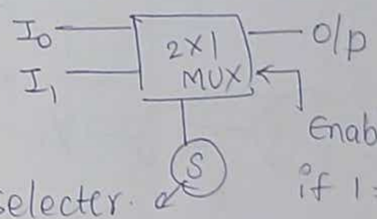
$[2^n \times 1]$ Mux

$n=1 \rightarrow [2\times1]$ Mux
$n=2 \rightarrow [4\times1]$ Mux
$n=3 \rightarrow [8\times1]$ Mux
$n=4 \rightarrow [16\times1]$ Mux
$n=5 \rightarrow [32\times1]$ Mux

2x1 Multiplexer

$[2^n \times 1]$ Mux

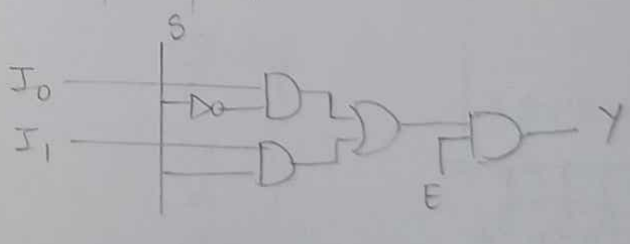
$n=1$



if 1 = work
0 = won't work

E	S	Y
0	x	0
1	0	I_0
1	1	I_1

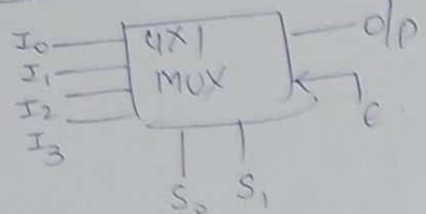
$Y = I_0\bar{S}E + I_1SE$
$= E[I_0\bar{S} + I_1S]$



4x1 MUX

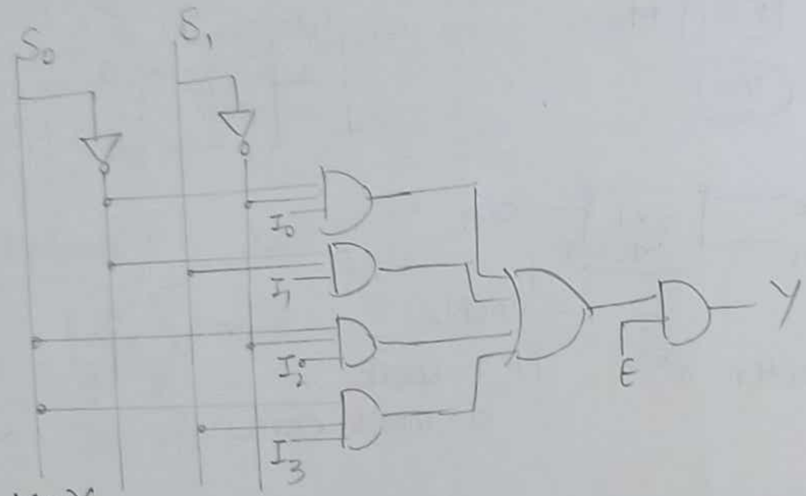
$[2^2 \times 1]$ MUX

$n=2$



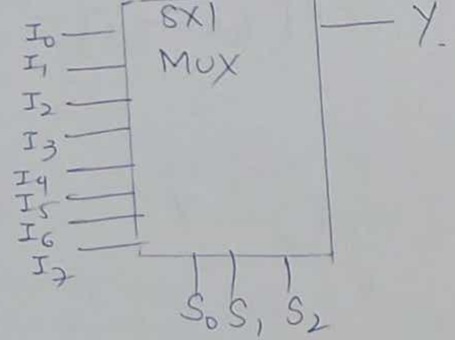
E	S_1	S_0	Y
0	x	x	0
1	0	0	I_0
1	0	1	I_1
1	1	0	I_2
1	1	1	I_3

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



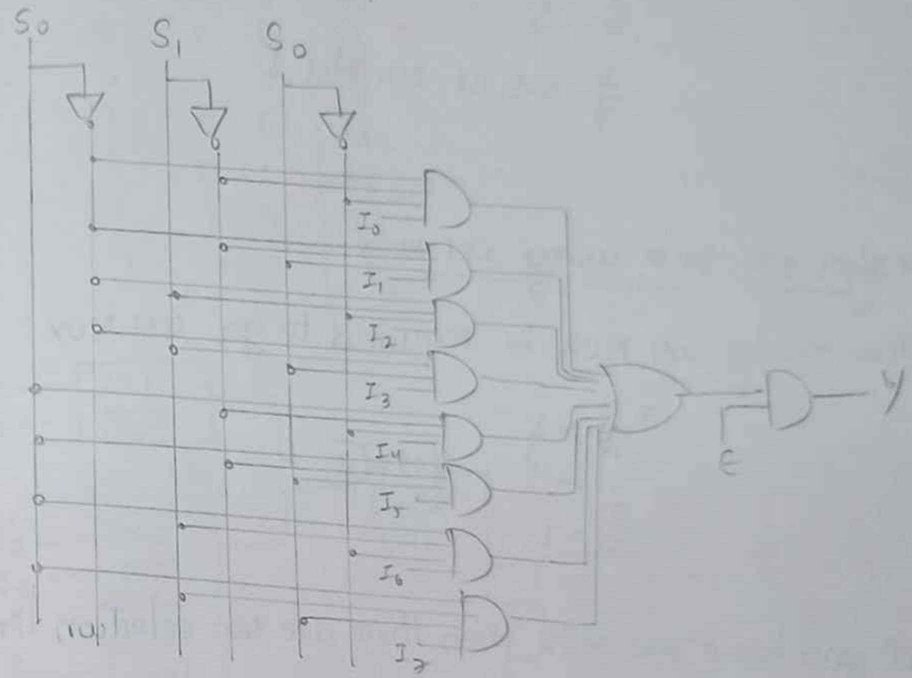
8x1 MUX

$n=3$



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

$$Y = E\left[\bar{S_2}\bar{S_1}\bar{S_0}I_0 + \bar{S_2}\bar{S_1}S_0I_1 + \bar{S_2}S_1\bar{S_0}I_2 + \bar{S_2}S_1S_0I_3 + S_2\bar{S_1}\bar{S_0}I_4 + S_2\bar{S_1}S_0I_5 + S_2S_1\bar{S_0}I_6 + S_2S_1S_0I_7\right].$$



MUX Tree

- MUX tree is used to obtain higher order mux using lower order mux

Q. How many 2x1 MUX is required to get 64x1 MUX.

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

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

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

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

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

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

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

$= \underline{63}$

Q2. How many 8×1 mux required to get 64×1 mux

$\frac{64}{8} = 8$ $\quad 8+1=9.$

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

Q3) How many 4×1 MUX required to get make 32×1 mux

$\frac{32}{4} = 8$ $\quad 8+2=10.+1=11.$

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

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

28/2/22

Design 4×1 MUX using 2×1 MUX

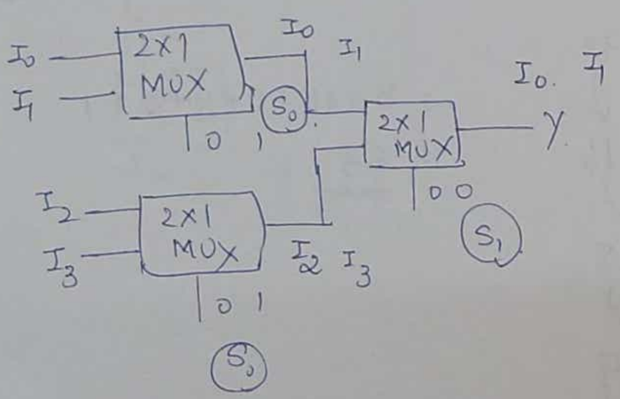
① How many 2×1 MUX is required to get 4×1 MUX

$= \frac{4}{2} = 2$ $\quad 2+1 = 3.$

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

If you have 4×1 MUX, then there are two selection line,

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



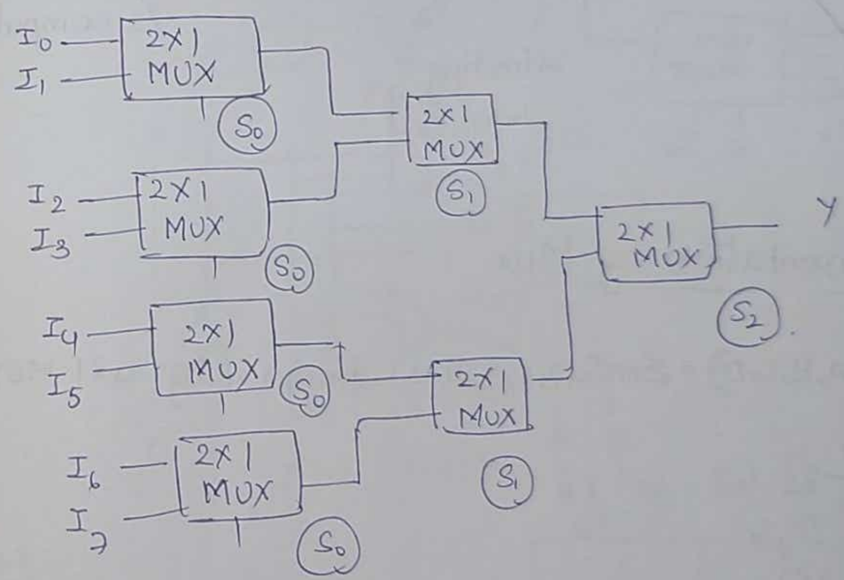
Q Design 8x1 MUX using 2x1 MUX

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

$\frac{4}{2} = 2$ $\Rightarrow 4+2+1 = 7$

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

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

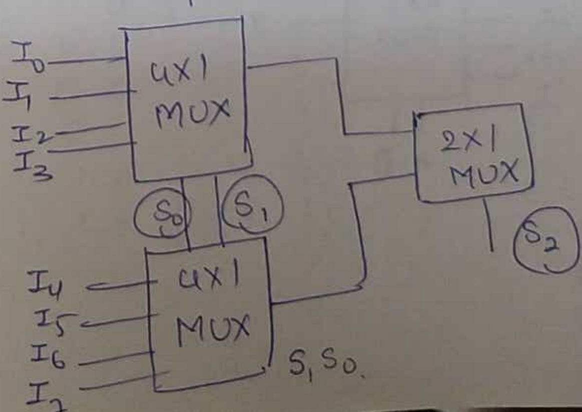


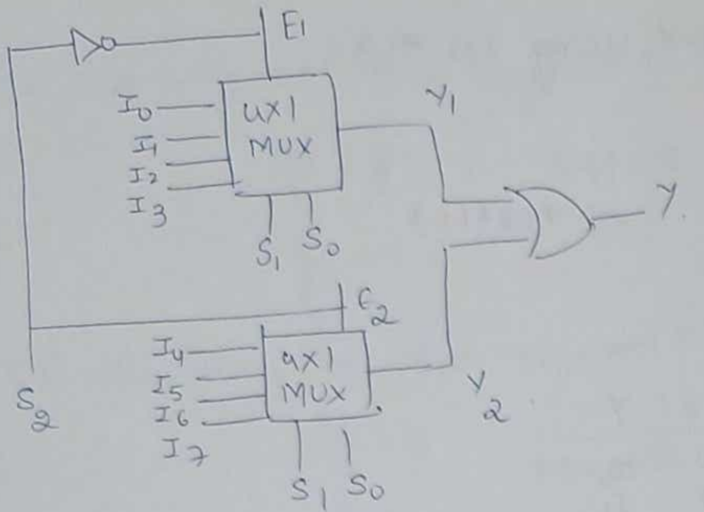
Q. 8x1 MUX using 4x1 MUX

$\frac{8}{4} = 2$ $2+1 = 3$

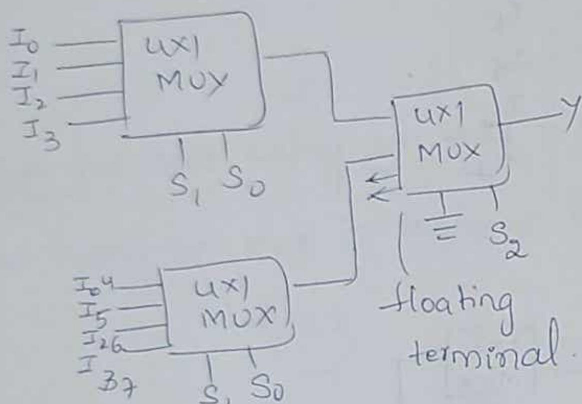
$\frac{2}{4} = 0.5$

Same truth table





X
Gate functionality is not much preferred



S_2 S_1 S_0
1 0 1

O/P = ?

I_5.

explanation is compulsory !

SOP implementation by MUX

① Function $F(A,B,C,D) = \Sigma m(0,2,6,9,11,13)$ design it by 4x1 MUX.

① Kmap

CD \ AB	$\bar A\bar B$ 00	$\bar A B$ 01	AB 11	$A\bar B$ 10
$\bar C\bar D$ 00	1			
$\bar C D$ 01			1	1
CD 11				1
$C\bar D$ 10	1	1		

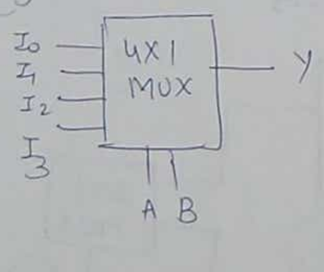
D (I_2)

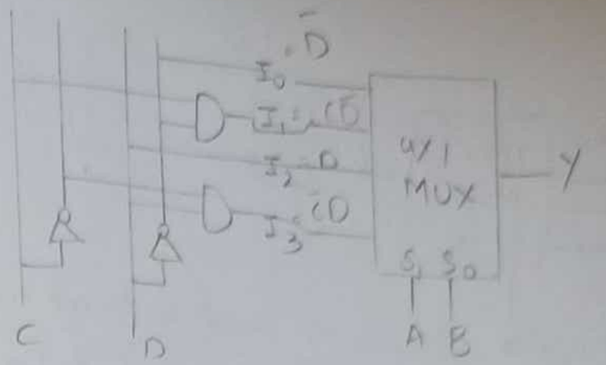
$\bar C D$ (I_3)

$\bar D$ (I_0)

$C\bar D$ (I_1)

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





② Function $F(A,B,C,D) = \Sigma m(1,4,6,8,9,13,14)$ make it by 4x1 MUX

K-map

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

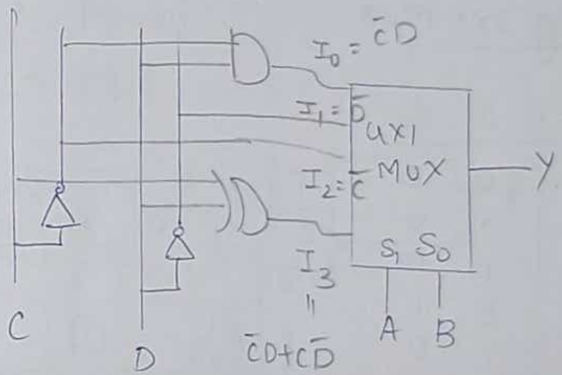
I_2 — $\bar{C}$

$\bar{C}D$ — I_0

$\bar{D}$ — I_1

I_3 — $\bar{C}D + C\bar{D}$

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



2/3/22

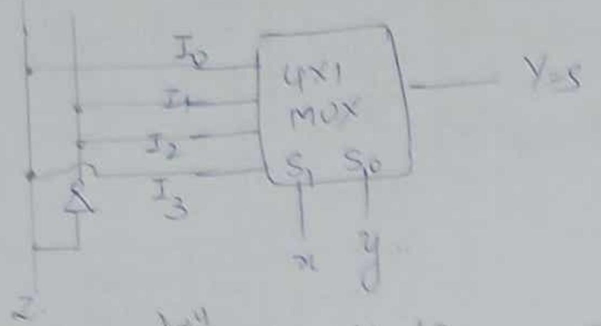
Full adder using 4x1 MUX

X	Y	Z	S	C_0
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

S

Z \ XY	00	01	11	10
0		1		1
1	1		1	

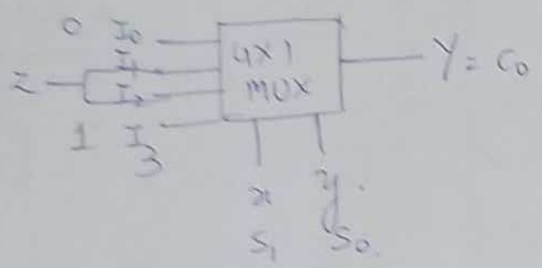
S_1	S_0	Y = S
0	0	$I_0 = Z$
0	1	$I_1 = \bar{Z}$
1	0	$I_2 = \bar{Z}$
1	1	$I_3 = Z$



Carry

$z \backslash xy$	00	01	11	10
$\bar{z}$ 0			1	
z 1		1	1	1

S_1 S_0	$Y = C_0$
0 0	$I_0 = 0$
0 1	$I_1 = z$
1 0	$I_2 = z$
1 1	$I_3 = 1$

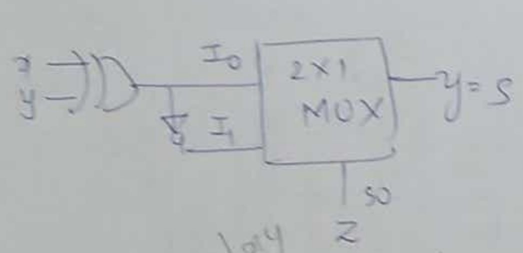


Full adder using 2x1 MUX

x	y	z	S	C_0
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

z S_0 $\backslash xy$	00	01	11	10
0		1		1
1	1		1	

S_0	$Y = S$
0	$I_0 = \bar{x}y + x\bar{y} \rightarrow x \oplus y$
1	$I_1 = \bar{x}\bar{y} + xy \rightarrow \overline{x \oplus y}$



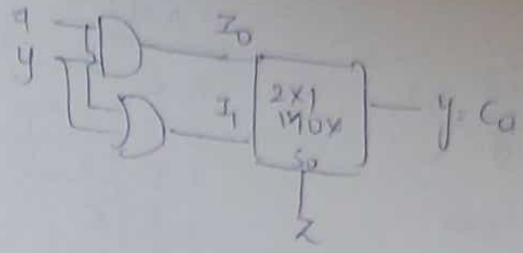
Carry

$z \backslash xy$	00	01	11	10
$\bar{z}$ 0			1	
z 1		1	1	1

S_0	$Y = SC_0$
0	$I_0 = xy$
1	$I_1 = \bar{x}y + x\bar{y} + xy$.

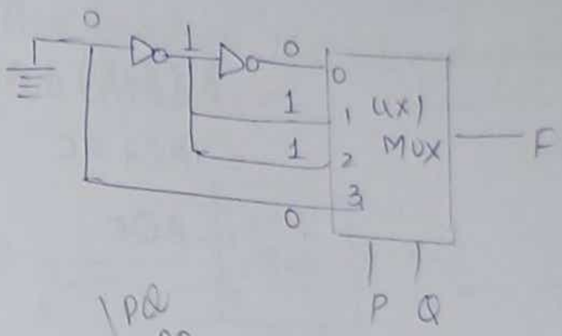
$= y(\bar{x}+x) + x\bar{y}$

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



Boolean function from Multiplexer circuit

Q1)

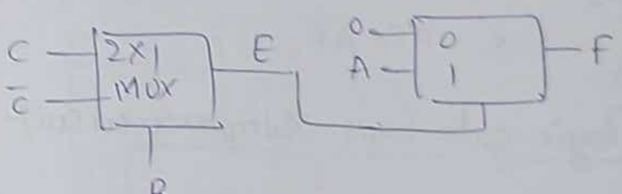


PQ	F
00	0
01	1
10	1
11	0

$R \backslash PQ$	00	01	11	10
0	1		1	
1		1		1

$$F = \bar{P}\bar{Q}\cdot 0 + \bar{P}Q\cdot 1 + PQ\cdot 0 + P\bar{Q}\cdot 1$$
$$= \bar{P}Q + P\bar{Q}$$
$$= P \oplus Q$$

Q2)



E = ?
F = ?

if $B = 0$, $E = C$
if $B = 1$, $E = \bar{C}$

B	E
0	C
1	$\bar{C}$

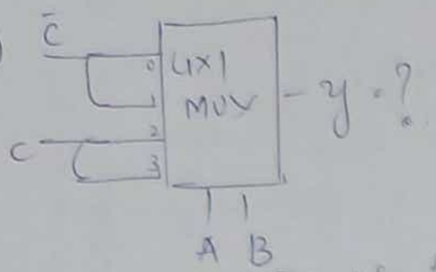
$$E = \bar{B}C + B\bar{C}$$
$$= B \oplus C$$

E	F
0	0
1	A

$$F = \bar{E}\cdot 0 + EA$$
$$= E\cdot A$$
$$= (B\oplus C)A = A\bar{B}C + AB\bar{C}$$

if $\bar{A}$, $F = E\bar{A}$
$$= (B\oplus C)\bar{A}$$
$$= \bar{A}\bar{B}C + \bar{A}B\bar{C}$$

③



$y = ?$

A B	y
0 0	$\bar{C}$
0 1	$\bar{C}$
1 0	C
1 1	C

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

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

$= A \odot C$

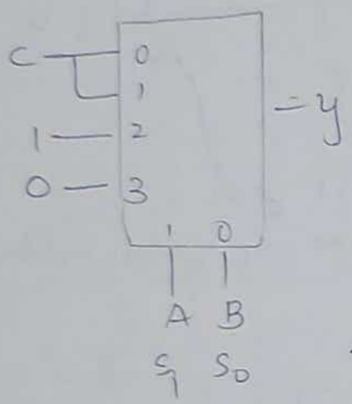
$= \bar{A}\bar{B}\bar{C} + \bar{A}B\bar{C} + A\bar{B}C + ABC$

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

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

$= A \odot C$

④



S_1 S_0

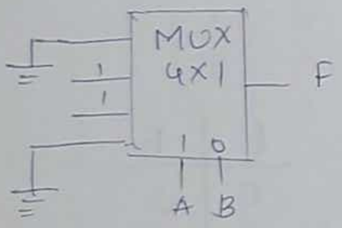
A B	y
0 0	C
0 1	C
1 0	1
1 1	0

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

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

Identification of logic gate from Multiplexer circuit

①



$F = ?$

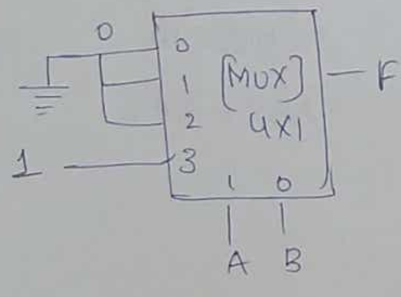
A B	y
0 0	0
0 1	1
1 0	1
1 1	0.

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

$F = A \oplus B$

"XOR gate"

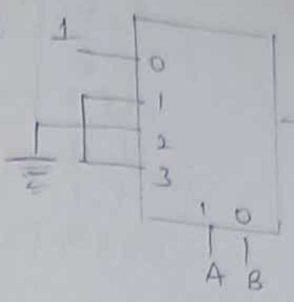
②



$F = AB$

"AND gate"

③



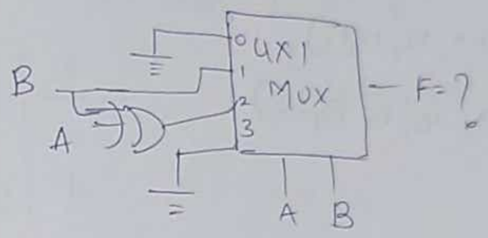
$F = \bar{A}\bar{B}$

$= \overline{A+B}$

xor "NOR Gate"

Examples based on Multiplexer, combinational ckt in digital electronics :-

①



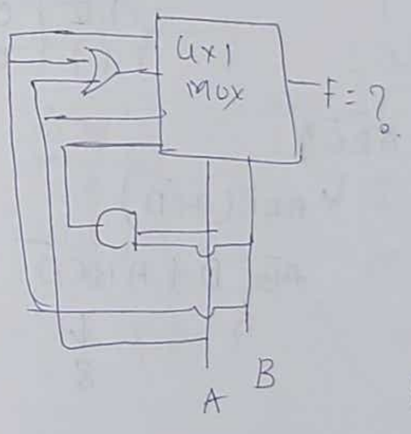
A B	F
0 0	0
0 1	B
1 0	$A \oplus B$
1 1	0

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

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

$F = A \oplus B$

②.



A B	F
0 0	B
0 1	A+B
1 0	A
1 1	A·B

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

$= 0 + \bar{A}B + 0 + A\bar{B} + AB$

$= B + A\bar{B}$

$= B + A$

Summation of mean term

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

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

3 2 5 4 7 6

$= \Sigma m($

Summation of mean term

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

$= \Pi m(0)$

A	B	mean
0	0	$\bar{A}\bar{B} - m_0$
0	1	$\bar{A}B - m_1$
1	0	$A\bar{B} - m_2$
1	1	$AB - m_3$

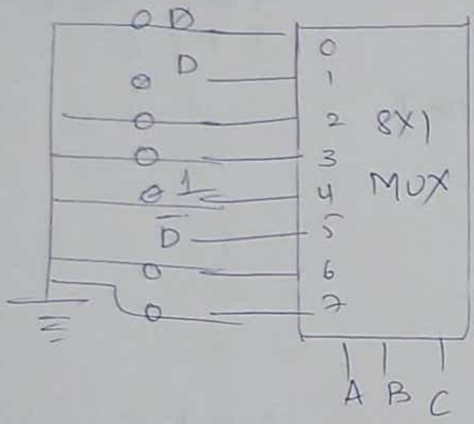
(3) Function F(A,B,C,D) can be expanded as

a) $\Sigma m(3,8,9,10)$

b) $\Sigma m(3,8,10,4)$

c) $\Pi(0,1,2,4,5,6,7,11,12,13,15)$

d) $\Pi(0,1,2,4,5,6,7,10,12,13,15)$.



A	B	C	F
0	0	0	0
0	0	1	D
0	1	0	0
0	1	1	0
1	0	0	~~0~~ 1
1	0	1	$\bar{D}$
1	1	0	0
1	1	1	0

$F = \bar{A}\bar{B}CD + A\bar{B}C\bar{D} + A\bar{B}\bar{C}1.$

$= \Sigma m(3,10,8,9)$

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

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

9   8

$= \Sigma m(3,8,9,10)$.

(4) Find the correct statement

A. Multiplexer.

a) select one of many input & transmit on single o/p

b) Routs the data from single i/p to many o/p.

c) It converts parallel data to serial data

d) It is a combinational ckt.

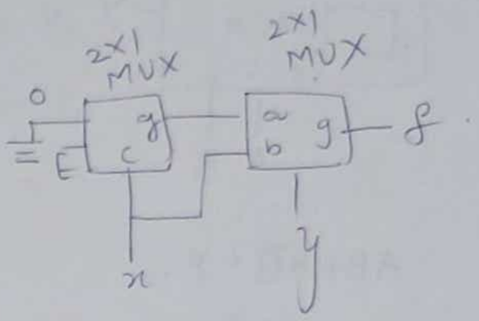
(5) If function $g = ac + b\bar{c}$ the find the o/p f

a) $x \oplus y$

b) $x\,E + x\bar{y}$

c) $y\bar{x}E + x\bar{y}$

d) None of these



according to g function

x	a
~~1~~ 0	0
0	E

$a = \bar{x}E$ $\quad c = x.$

$b = x$

$g = x\,E\,0 + \bar{x}E$

$g = ac + b\bar{c}$

$= \cancel{xEx + x\bar{x}}$

$= \cancel{xE}$

y	f
~~1~~ 0	a
0	b.

$f = yb + \cancel{\bar{y}a}$

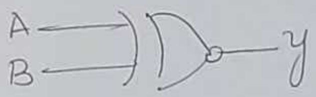
$= \cancel{yx + \bar{y}xE}$

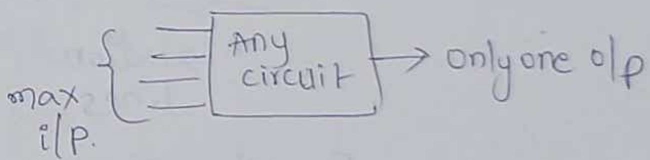
$= ya + \bar{y}b$

$= y\bar{x}E + \bar{y}x$.

(1) XNOR

Coincidence logic





A	B	y
0	0	~~0~~ 1
0	1	0
1	0	0
1	1	~~0~~ 1

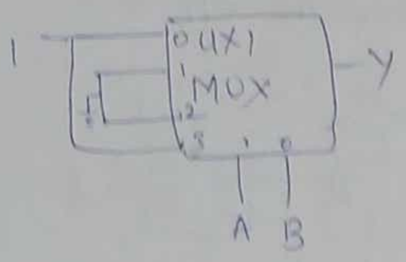
(1) Input too rich ≥ 2, o/p = 1

(2) if all i/p = 0 o/p = 1

(4) if all i/p = 1 o/p = 1

XNOR

②.



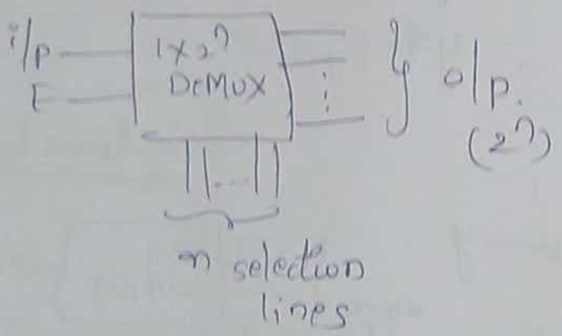
$AB + \bar{A}\bar{B} = Y$

XNOR

"Coincidence logic"

Demultiplexer and 1 to 2 DeMUX

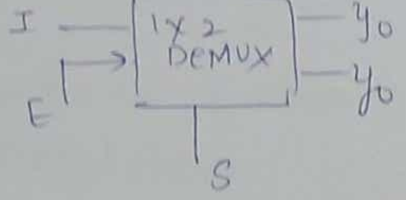
- It is having reverse operation to that of MUX
- Basic form of DEMUX $[1\times2^n]$ DeMux
- Demux is series to parallel converter.



E=0, then o/p → 0

E=1, operation will be there in Demux.

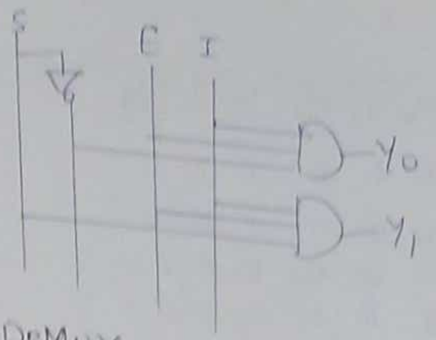
1 to 2 DEMUX



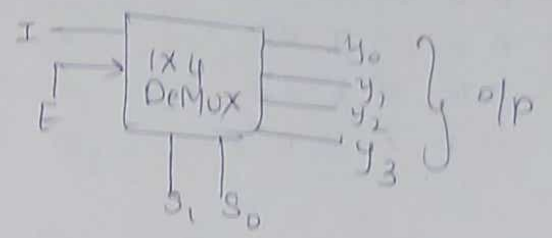
E	S	Y_0	Y_1
0	X	0	0
1	0	I	0
1	1	0	I

$Y_0 = IE\bar{S}$

$Y_1 = IES$.



1 to 4 DeMux



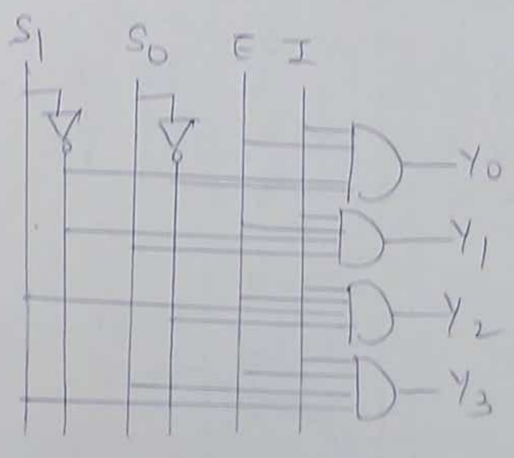
E	S_1	S_0	Y_0	Y_1	Y_2	Y_3
0	X	X	0	0	0	0
1	0	0	I	0	0	0
1	0	1	0	I	0	0
1	1	0	0	0	I	0
1	1	1	0	0	0	I

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

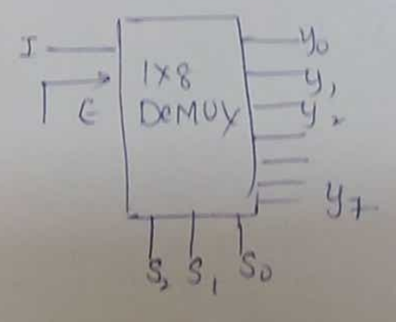
$Y_1 = IE\bar{S}_1S_0$

$Y_2 = IES_1\bar{S}_0$

$Y_3 = IES_1S_0$



1X8 DeMux



E	S_2	S_1	S_0	y_0	y_1	y_2	y_3	y_4	y_5	y_6	y_7
0	x	x	x	0	0	0	0	0	0	0	0
1	0	0	0	I	0	0	0	0	0	0	0
1	0	0	1	0	I	0	0	0	0	0	0
1	0	1	0	0	0	I	0	0	0	0	0
1	0	1	1	0	0	0	I	0	0	0	0
1	1	0	0	0	0	0	0	I	0	0	0
1	1	0	1	0	0	0	0	0	I	0	0
1	1	1	0	0	0	0	0	0	0	I	0
1	1	1	1	0	0	0	0	0	0	0	I

$y_0 = I E \bar{S}_2 \bar{S}_1 \bar{S}_0$

$y_1 = S I E \bar{S}_2 \bar{S}_1 S_0$

$y_2 = I E \bar{S}_2 S_1 \bar{S}_0$

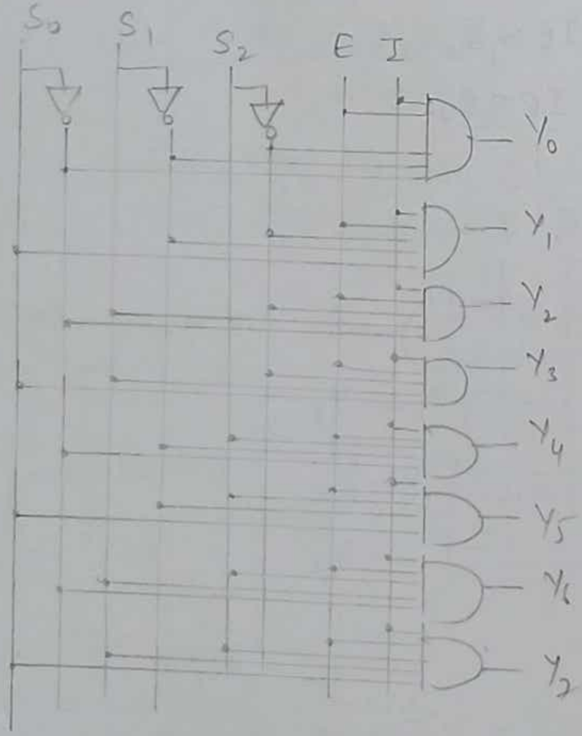
$y_3 = I E \bar{S}_2 S_1 S_0$

$y_4 = I E S_2 \bar{S}_1 \bar{S}_0$

$y_5 = I E S_2 \bar{S}_1 S_0$

$y_6 = I E S_2 S_1 \bar{S}_0$

$y_7 = I E S_2 S_1 S_0$



Full Subtractor using 1x8 DeMux :

	A	B	B_{in}	D	B_o
m_0	0	0	0	0	0
m_1	0	0	1	1	1
m_2	0	1	0	1	1
m_3	0	1	1	0	1
m_4	1	0	0	1	0
m_5	1	0	1	0	0
m_6	1	1	0	0	0
m_7	1	1	1	1	1

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

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

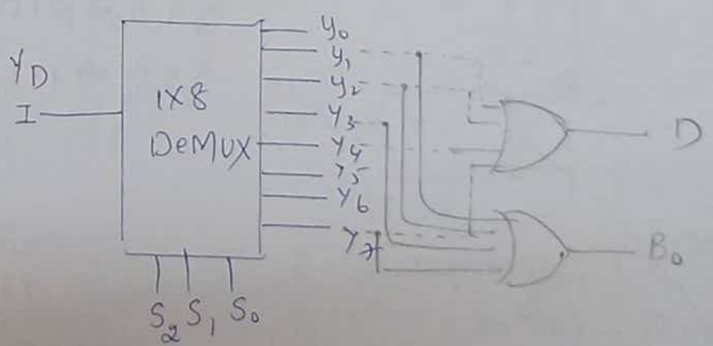
1:8 Demux

i/p o/p

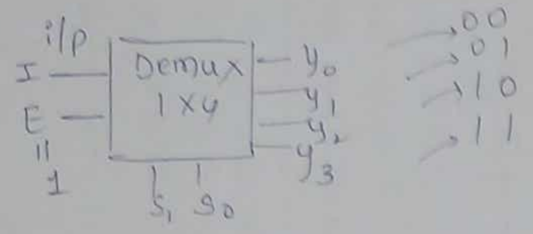
$[1 \times 2^{3}]$ DeMux

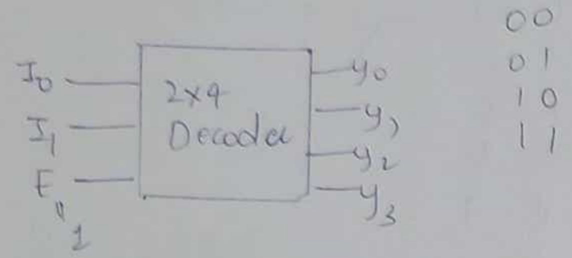
$n = 3$, S.L

S_2	S_1	S_0	Y_0	Y_1	Y_2	Y_3	Y_4	Y_5	Y_6	Y_7
0	0	0	1	0	0	0	0	0	0	0
0	0	1	0	1	0	0	0	0	0	0
0	1	0	0	0	1	0	0	0	0	0
0	1	1	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



DeMux as Decoder





i/p I = 1

A B

$S_1 S_0 = I_0 I_1$

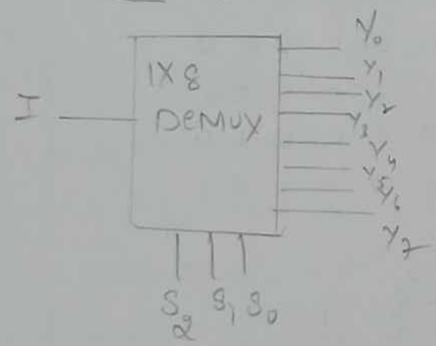
then Demux = Decoder

Implementation of Boolean expression using demux.

- $F(A,B,C,D) = \Sigma m(1,3,6,7)$ by 1X8 Demux.

DeMux = $[1 \times 2^n]$

$n = 3$.



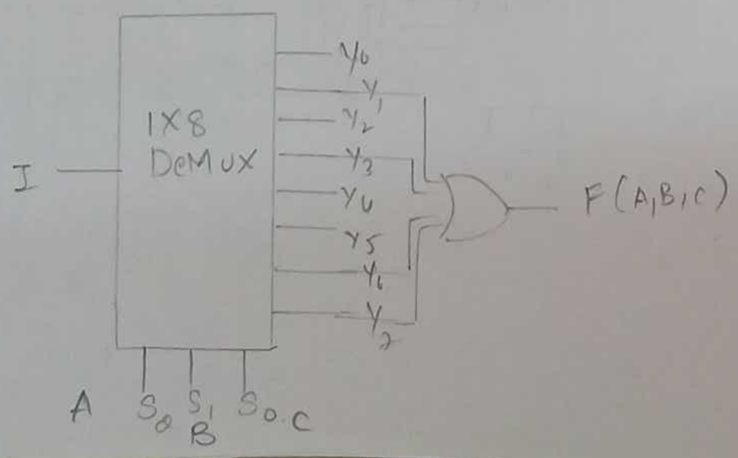
if I = 1

$S_2 S_1 S_0 \Rightarrow 001$ $Y_1 = 1$

$S_2 S_1 S_0 \Rightarrow 011$ $Y_3 = 1$

$S_2 S_1 S_0 \Rightarrow 1010$ $Y_6 = 1$

$S_2 S_1 S_0 \Rightarrow 111$ $Y_7 = 1$.



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

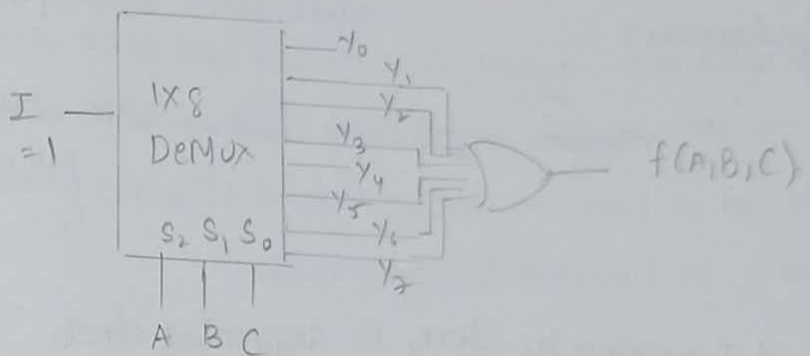
	C \ AB	01	01	11	10
$\bar{C}$	0		1	1	
C	1	1	1	1	1

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

1X8 DeMux

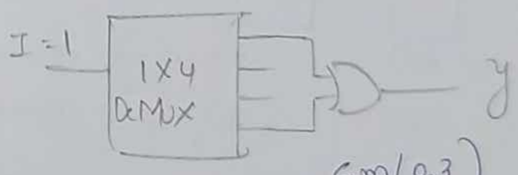
↳ $[1 \times 2^{n}]$ Demux

$n = 3$



Boolean expression of function using DeMux

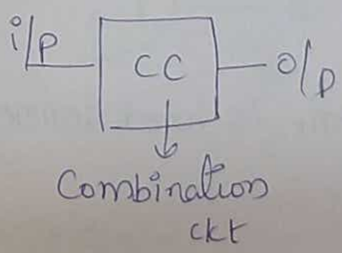
$y = AB + \bar{A}\bar{B}$ (1X4)

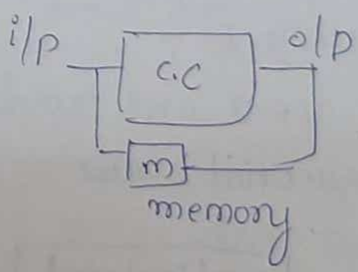


$\Sigma m(0,3)$

Sequential circuit

- Sequential ckt is a combination of circuit and memory.

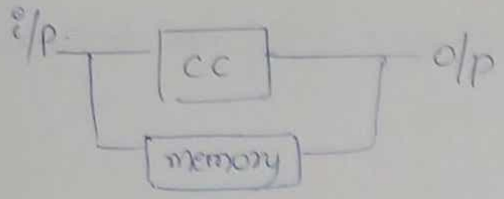




→ S.C. (sequential ckt)

it depends on present i/p & past o/p.

- In sequential ckt, present o/p is depending on present i/p & past o/p.



Example of sequential circuits : Flip flop, Counter, Register → all are useful for storage.

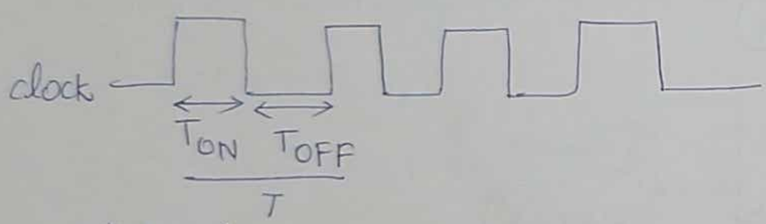
classification of sequential ckt.

1. Synchronous SC → All memory element will be given same clock pulse.
2. Asynchronous SC → memory element are working with diff clock pulse.

4/3/22

clock and Triggering by clock in sequential clock

- clock is signal, used in digital ckt



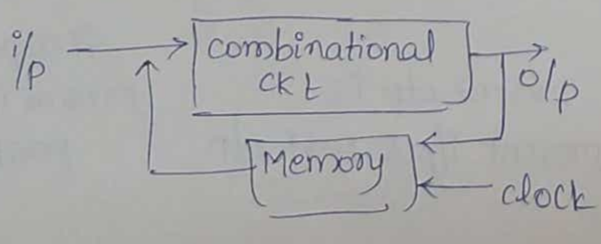
Digital

Duty cycle $D = \frac{T_{ON}}{T}$, usually 0.5

$= \frac{T/2}{T} = 1/2 = 0.5$

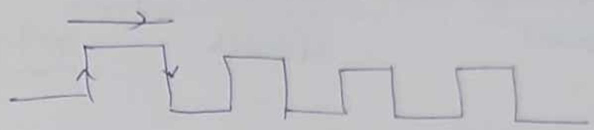
Frequency $F = \frac{1}{T}$

Operational speed and transition of state is ~~fixed~~ define by clock in sequential circuit



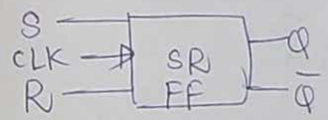
- o/p of circuit is defined.
- o/p state will change wrt clock

- There are two types of clock trigering
1) Edge trigger clock — +ve ET, -ve ET
2) Level trigger clock — +ve LT, -ve LT



- Clock raise from low to high, trigger - +ve Edge trigger ET
- clock ~~to~~ falls from high to low, trigger - -ve ET
- During clock is high triggered referred to be - +ve LT
  low triggered referred to -ve LT

SR Flipflops

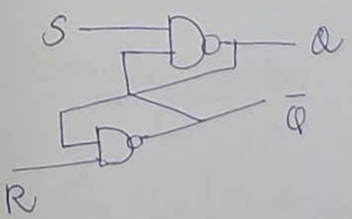
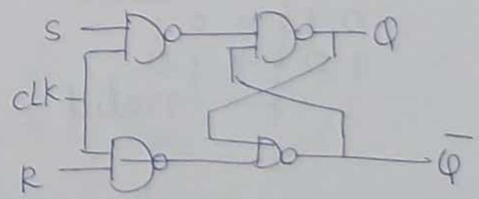


S → set
R → Reset

CLK —▷ +ve ET
CLK —o▷ -ve ET

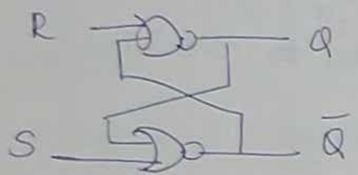
CLk —— +ve LT
CLk ——o -ve LT

Latch & Flip flop differeces

Latch	Flip flop
	
- It doesnot have clock in its internal circuit	- It has clock in its with its internal ckt
- It checks inputs Continously and changes output correspondingly	- It checks i/p continuously and changes o/p with respect to clock signal.

Latch	FF
Latch is sensitive to i/p signal.	FF is sensitive to i/p and clock signals
- Follows level triggering	Follows edge triggering.
- It cannot ~~u~~ be used as register	- It can be used as register
- It is ~~lo~~ Asynchronous ckt	- It ~~can~~ is synchronous ckt
- It is less complex and need less power	- It is more complex and need more power
- Faster	- slower

SR latch using NOR gate



NOR
R S
0 0

memory

$S=1, R=0 \quad Q=1 \quad \bar{Q}=0$

$S=0, R=0 \quad Q=1 \quad \bar{Q}=0 \;/\; Q=0, \bar{Q}=1$

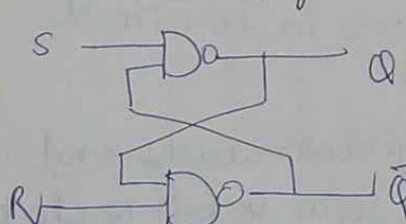
$S=0, R=1 \quad Q=0 \quad \bar{Q}=1$

Condition is violating.

If $S=1, R=1 \quad Q=0, \bar{Q}=0$ ✗ [Not possible] or invalid.

S R	Q $\bar{Q}$
0 0	memory
0 1	0 1
1 0	1 0.
1 1	invalid

SR latch using NAND gate



$S=1, R=0, \quad Q=0, \bar{Q}=1$

$S=1, R=1, \quad Q=0, \bar{Q}=1 \;/\; Q=1, \bar{Q}=0.$

memory

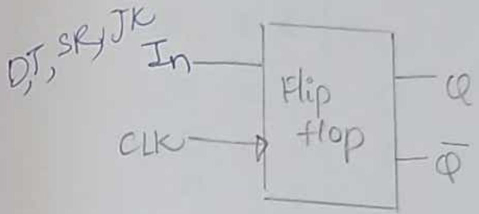
$S=0, R=1 \quad Q=1, \bar{Q}=0.$

not possible ← $S=0, R=0 \quad Q=1, \bar{Q}=1$

S	R	Q $\bar{Q}$
0	0	Invalid
0	1	1 0
1	0	0 1
1	1	Memory.

15/3/22

Truth table, characteristics tablo & Excitation table of Flipflop



i/p
- Set-Reset (SR) flip flop or Latch
- Jk flipflop
- D (Data or Delay) flip flop
- T(Toggle) flip flop

Q and $\bar{Q}$ will be not change for all flipflop but diff FF i/p terminal will change

Truth Table

CLK	In	Q	$\bar{Q}$

CLK	In	Q_{n+1}

next state

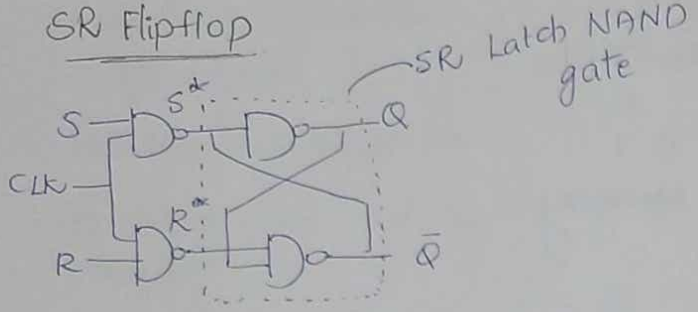
Characteristics table:

Q_n	S	R	Q_{n+1}

Exitation table:

Q_n	Q_{n+1}	S	R

SR Flipflop



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

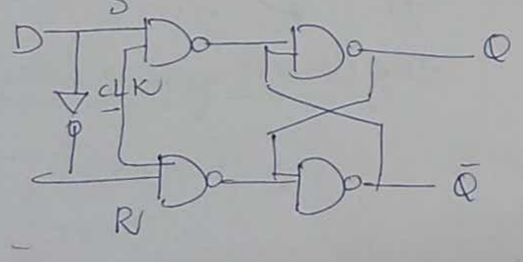
$R^* = \overline{R \cdot CLK} = \bar{R} + \overline{CLK}$

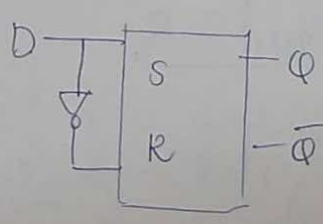
S^*	R^*	Q	$\bar{Q}$
0	0	Invalid	
0	1	1	0
1	0	0	1
1	1	Memory	

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	Invalid	

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	Invalid.

D-Flip flop (Data Flipflop)





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	Invalid

SR - characteristics table

Q_n	S	R	Q_{n+1}
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	Invalid
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	Invalid.

Q_{n+1}

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

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

Excitation table

→ write by charateristic table.

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

D- characteristic table:-

CLK	D	Q_{n+1}
0	X	Q_n
1	0	0
1	1	1

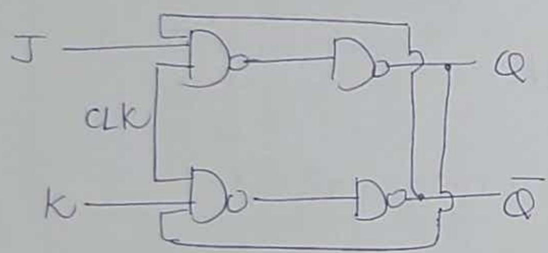
Q_n	D	Q_{n+1}
0	0	0
0	1	1
1	0	0
1	1	1

$Q_{n+1} = D$

Excitation table:-

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

JK Flipflop



CLK = 1, J = 1, K = 1

$Q = 0, \bar{Q} = 1$

CLK	S	R	Q	$\bar{Q}$
0	X	X	M	
1	0	0	M	
1	0	1	0	1
1	1	0	1	0
1	1	1	Invalid.	

CLK	J	K	Q	$\bar{Q}$
0	X	X	M	
1	0	0	M	
1	0	1	0	1
1	1	0	1	0
1	1	1	M	

CLK	J	K	Q_{n+1}
0	X	X	Q_n
1	0	0	Q_n
1	0	1	0
1	1	0	1
1	1	1	$\bar{Q}_n$

Characteristics Table :

Q_n	J	K	Q_{n+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

Q_n \ JK	$\bar{J}\bar{K}$	$\bar{J}K$	JK	$J\bar{K}$
$\bar{Q}_n$	0	0	1	1
Q_n	1	0	0	1

$$Q_{n+1} = J\bar{Q}_n + \bar{K}Q_n$$

Exitation table

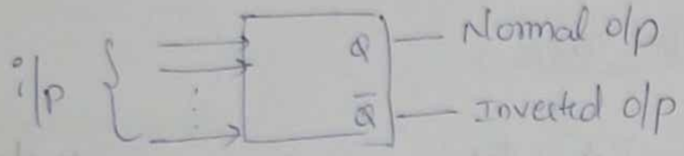
Q_n	Q_{n+1}	J	K
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

16/3/22

Flip flops (F/F)

- FF is a memory element which is capable of storing one bit of information and it is used in clocked sequential circuits
- A Flip flop has two outputs, one for normal value and other for complement value of the bit stored in it.

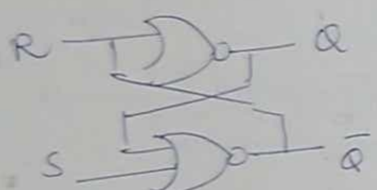
- A F/F can maintain a binary state indefinitely (as long as power is delivered to the circuit) until directed by an i/p signal to switch states
- A F/F is also known as bistable multivibrator



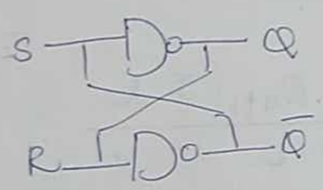
- F/F dependency is how i/p and clock Pulse can cause transition btw two states
- SR, JK, D and T (F/F) Delay and Time

Basic Flipflop ckt or Latch

- A basic F/F ckt can be constructed from two NAND gate or two NOR gate



F/F w NOR



F/F NAND.

S	R	Q	$\bar{Q}$
1	0	1	0
0	1	0	1
0	0	Memory	
1	1	Invalid	

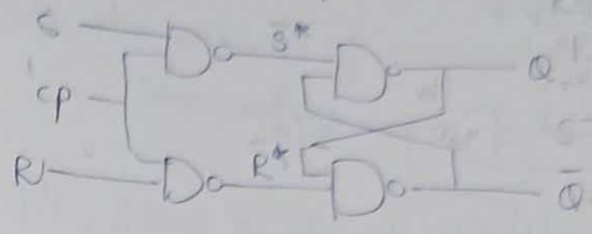
S	R	Q	$\bar{Q}$
0	0	Invalid	
0	1	1	0
1	0	0	1
1	1	Memory	

Synchronous Sequential circuit :

Clocked SR Flipflop

- The clocked SR flipflop consist of the basic NAND latch and two other NAND gates to provide clock pulse (CP)

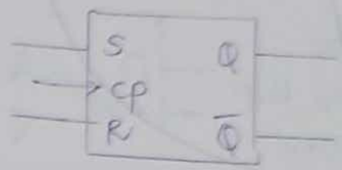
- CP is used for the synchronization & act as an additional control input



$S^* = \overline{S \cdot CP}$

$R^* = \overline{R \cdot CP}$

Graphical symbol of SR F/F:



Characteristic table

CP = 1

Q_n	S	R	Q_{n+1}
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	Invalid (X)
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	Invalid (X)

don't care

S	R	Q_{n+1}
0	0	Memory - Q_n
0	1	0
1	0	1
1	1	Invalid

Excitation table

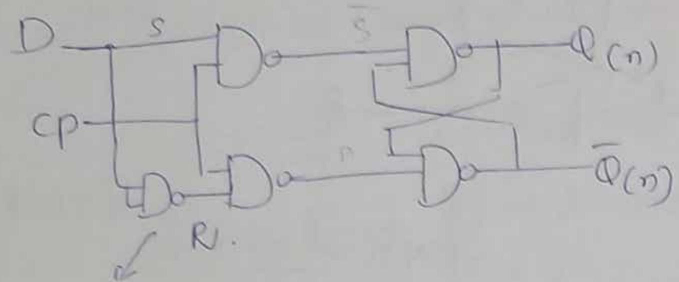
Q_{n+1} k-map

Q \ SR	$\overline{S}\overline{R}$	$\overline{S}R$	SR	$S\overline{R}$
$\overline{Q}$	0	0	X	1
Q	1	0	X	1

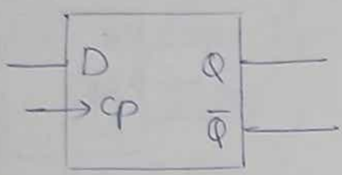
$$Q_{n+1} = S + Q\overline{R}$$

D-Flip flop :- Delay F/F.

Clocked SR F·F



we can take inverter also, but we have to take diff IC.



Characteristics table

$Q(n)$	D	$Q_{(n+1)}$
0	0	0
0	1	1
1	0	0
1	1	1

$Q_{n+1} = D$.

Characteristic Equation

Q \ D	0	1
0	0	1
1	0	1

$Q_{n+1} = D$.

Excitation Table :-

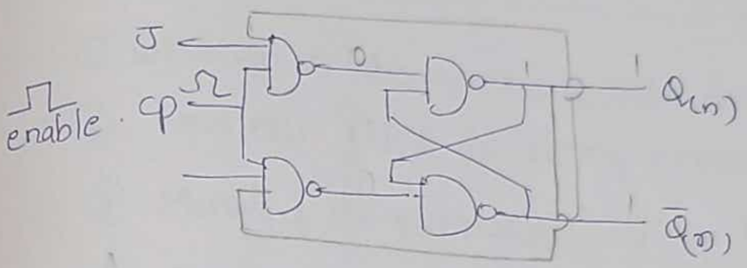
D	$Q_{(n+1)}$
0	0
1	1

Whatever delay is given o/p will be the same.

JK F/F ~ Jack Kilby

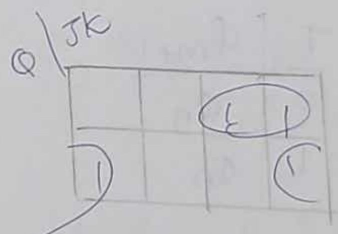
- JK F/F is refined form of SR F/F.
- It solve the problem of intermediate state when both i/p as 1.

• JK F/F, J and K behave like i/p S and R to set and reset the F/F.



Q_n	J	K	Q_{n+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

k-map

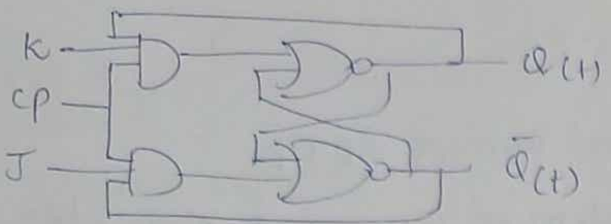


$$Q_{n+1} = J\bar{Q}_n + \bar{K}Q_n$$

Excitation table:

J	K	Q_{n+1}
0	0	Q_n
0	1	0
1	0	1
1	1	$\bar{Q}_n$

- If the clock pulse duration is more than the propagation delay of the flipflop o/p changes many times which causes Unstability in the o/p. This is known as Race around condition.

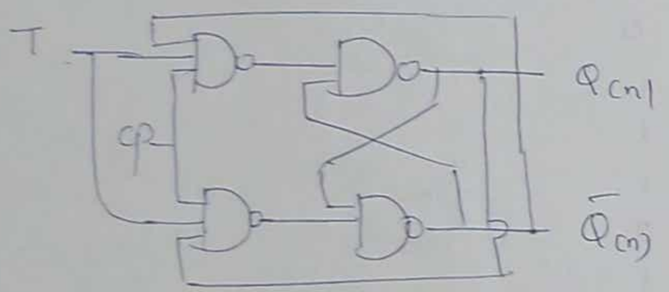


JK F/F NOR

J K	Q_{n+1}
0 0	Q_n
0 1	0
1 0	1
1 1	$\bar{Q}_n$

T-Flip Flop :-

- The T(F/F) is a single i/p version of JK(F/F) as shown in Figure



characteristics table:

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

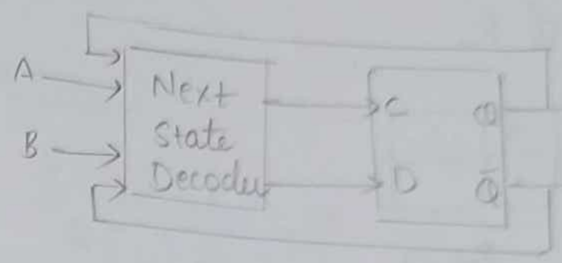
T	$Q_{(n+1)}$
0	Q_n
1	$\bar{Q}_n$

Q_n \ T	$\bar{T}$	T
$\bar{Q}_n$	0	1
Q_n	1	0

$$Q_{(n+1)} = Q\bar{T} + \bar{Q}T$$
$$= Q \oplus T$$

Realisation of one F/F to using other F/F

- It is possible to implement a flipflop circuit using any other flipflop



1. write down characteristic table for x F/F
2. Find out Y F/F i/p using excitation table by x F/F
3. Minimize the expression using K-map
4. Draw the circuit diagram

Ex) Realise D F/F using SR F/F

Step-1

Q(t)	D	$Q_{(n+1)}$
0	0	0
0	1	1
1	0	0
1	1	1

Step-2

Q_n	Q_{n+1}	S	R
0	0	0	x
0	1	1	0
1	0	0	1
1	1	x	0.

Excitation table to find Flip flop

Q_n	D	Q_{n+1}	S	R
0	0	0	0	x
0	1	1	1	0
1	0	0	0	1
1	1	1	x	0

Step-3: K Map of S

	$\bar{D}$	D
$\bar{Q}_n$		1
Q_n		x

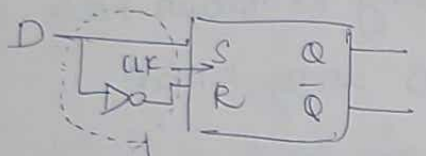
$S = D$

k-map of R

x	
1	

$R = \bar{D}$

Step-4:-



Next state Decoder

② Realise JK(F/F) using SR.

Step-1

Q_n	J	K	Q_{n+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

Step-2:

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

Flip flop i/p using Excitation table,

Q_n	J	K	Q_{n+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

Step-3

K-map:

S

Q \ JK	$\bar{J}\bar{K}$	$\bar{J}K$	JK	$J\bar{K}$
0	0	0	1	1
1	x	0	0	x

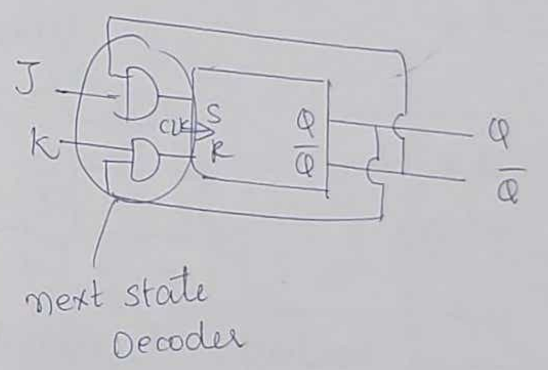
$S = J\bar{Q}$

R

x	x		
	1	1	

$R = KQ$

Step 4:



31/3/22

SR Flipflop

1/4/22.

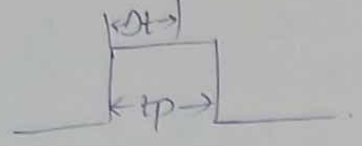
JK F/F :- $J=1, K=1, Q=0$

tp is clock pulse (width)

Δt is the propagation delay through the NAND gate

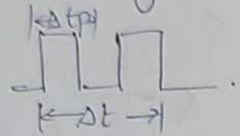
- when clock pulse is applied ~~to~~ e, within Δt, $Q=1$

and if $Q=1, J=1, K=1 \Rightarrow Q=0$

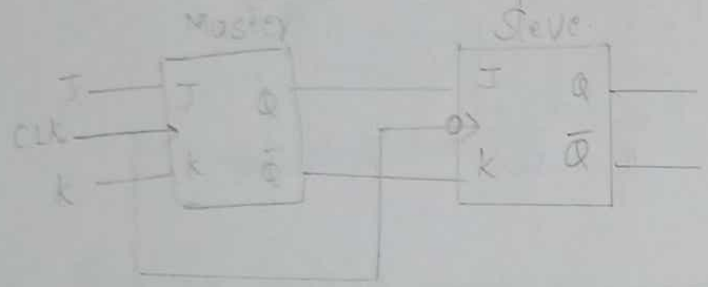


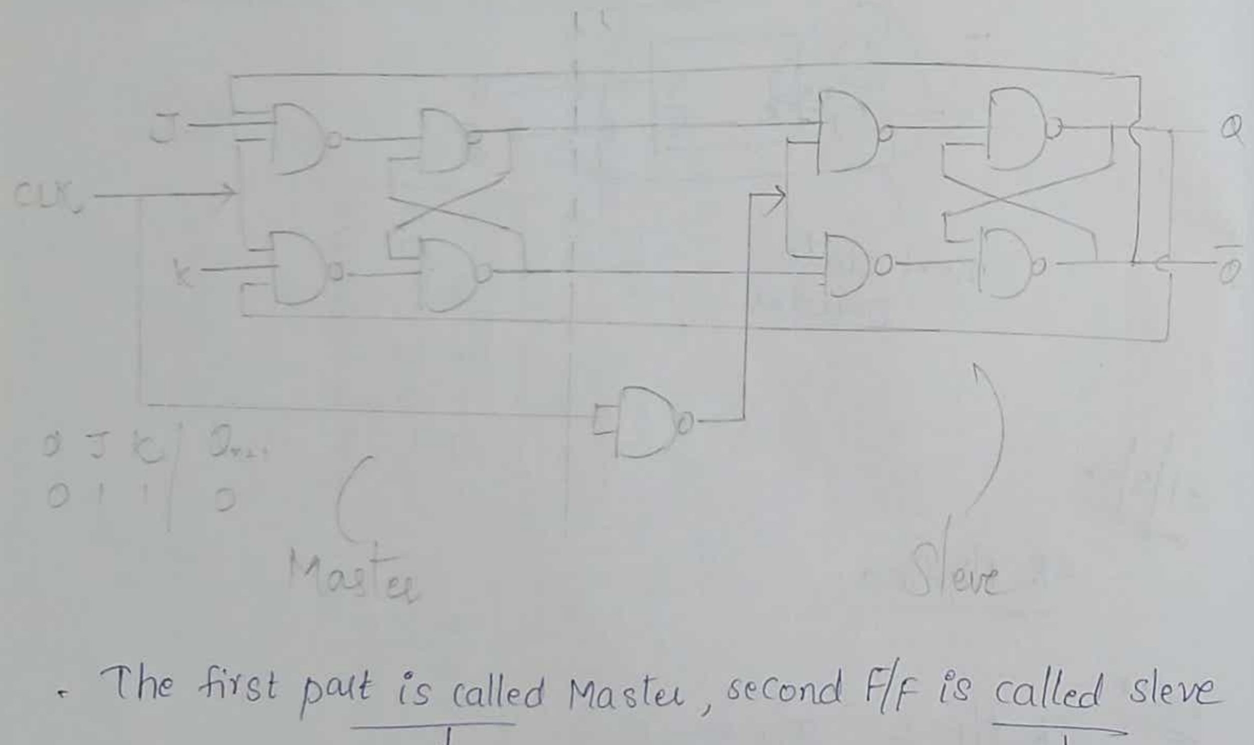
- ~~The~~ Hence the o/p will oscillate back between 0 & 1 in t_p duration of the clock pulse width.

→ if t_p is less than Δt, if we dividing this, unstability part becomes stable.
($t_p < \Delta t$)



- Another way to remove Race Around condition is your Master sleve.





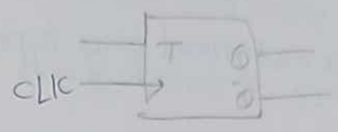
- The first part is called Master, second F/F is called sleve
  - Master: driven by +ve clock cycle
  - sleve: driven by negative clock cycle.
- During +ve clock, Master F/F gives intermediate o/p. but sleve doesn't respond.

- During -ve clock, sleve F/F activate and it pop copy the previous o/p of Master and produces final O/P.

T- F/F

- Single i/p version of JK F/F

T- Toggle F/F



Why T F/F is required?

Q(t)	T	Q(T+1)
0	0	0
0	1	1
1	0	1
1	1	0

Q(t+1)

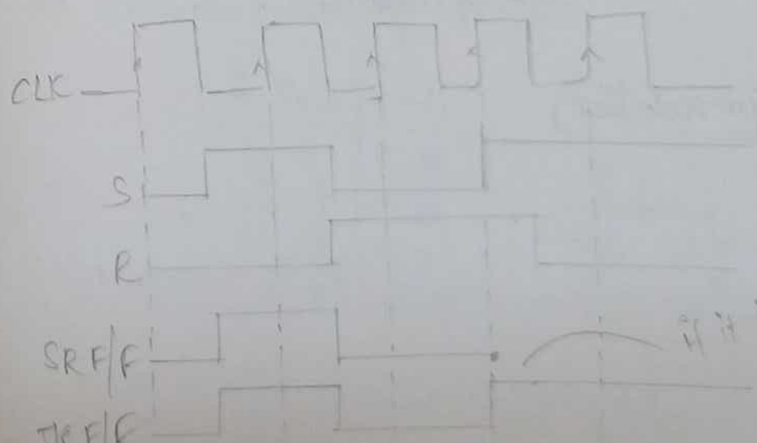
T \ Q	$\bar{Q}$	Q
$\bar{T}$	0	1
T	1	0

$$= Q\bar{T} + \bar{Q}T$$
$$= Q \oplus T$$

Why toggling is Important?

- For interchanging your output

Flip flop waveform



based on excitation table

if it invalid, stop there

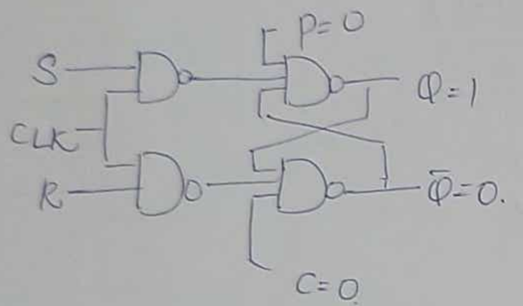
5/4/22

Present and clear input

- Preset is used to make output $Q=1$
- Clear is used to make output $Q=0$

· Preset and clear input does not need any type of synchronization

· Preset and clear i/p gives predefined o/p without bothering about other i/p.



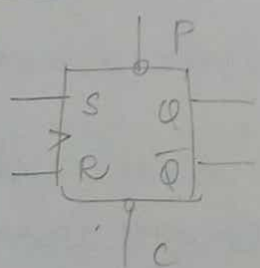
if $p=0 \Rightarrow Q=1, \bar{Q}=0$

if $C=0 \Rightarrow Q=0, \bar{Q}=1$

What will happen to Q_{n+1} ?

Preset	Clear	Q_n
0	0	Not functioning
0	1	1
1	0	0
~~0~~ 1	1	F/F operation

Graphics Representation



($P=0$, $C=0$, then it is functioning)

- If your preset = 0, what will happen to your present & next o/p ?

Steps of F/F conversion

Step 1: Note the available F/F and required F/F

Step 2: Write characteristic table of required F/F

Step 3: Write Excitation table of available F/F

Step 4: Solve Boolean expression

Step 5: Draw circuit

Q1. SR F/F to D F/F

1) available F/F = SR
Required F/F = D

2) Characteristic table of D F/F :-

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

3) Excitation table of SR F/F :

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

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

4) K-Map :-

S: rows D, columns Q_n

	0	0
	1	X

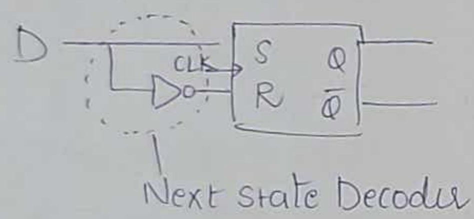
$S = D$

R: rows D, columns Q_n

	X	1
	0	0

$R = \overline{D}$

5) Circuit:



③ SR to T F/F

1) Available - SR
Required - T

2) characteristic table

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

3) Excitation table

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

Q_n	T	Q_{n+1}	S	R
0	0	0	0	x
0	1	1	1	0
1	0	1	x	0
1	1	0	0	1

4) K-Map:

S:

$T \backslash Q_n$	0	1
0	0	x
1	1	0

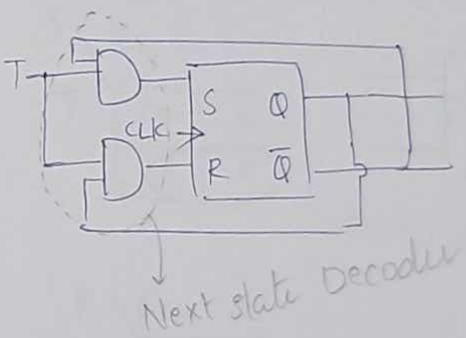
$S = \bar{Q}_n T$

R:

$T \backslash Q_n$	0	1
0	x	0
1	0	1

$R = Q_n T$

5) Circuit Diagram



(4) JK to D F/F

1) Available - JK
Required - D

2) Characteristic table

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

3) Excitation table of JK

Q_n	Q_{n+1}	J	K
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

Q_n	D	Q_{n+1}	J	K
0	0	0	0	X
0	1	1	1	X
1	0	0	X	1
1	1	1	X	0

4) k-Map

J: D \ Q_n

	0	X
	1	X

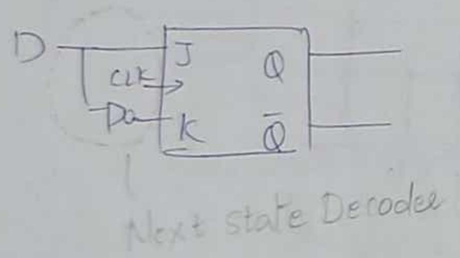
$J = D$

K: D \ Q_n

	X	1
	X	0

~~J~~ $K = \overline{D}$

5) Circuit Diagram



7/4/22

① Jk to T conversion

1) Available Jk
Required T

2) Characteristic Table

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

3) Excitation table of JK

Q_n	Q_{n+1}	J	K
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

Q	T	Q_{n+1}	J	K
0	0	0	0	X
0	1	1	1	X
1	0	1	X	0
1	1	0	X	1

4) K-Map

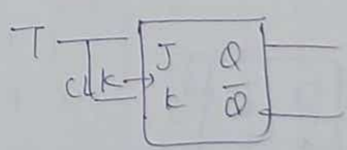
J: T \ Q_n	0	1
0	0	X
1	1	X

J = T

K: T \ Q_n	0	1
0	X	0
1	X	1

K = T

5) Circuit diagram



② D to T F/F

1) Available D
   Required T

2) characteristic Table

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

3) Excitation table of D

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

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

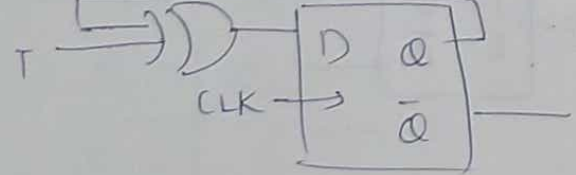
4) K-Map

D: T \ Q_n		
	0	1
	1	0

$D = Q_n \oplus T$

$= T\bar{Q}_n + Q_n\bar{T}$

5) Circuit Diagram



③ D to JK

1) Available D
   Required JK

2) Characteristic table JK

Q	J	K	Q_{n+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

3) Excitation table of D

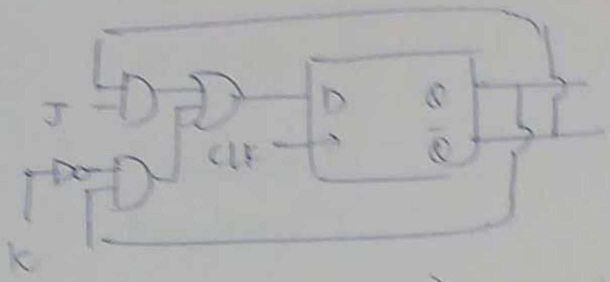
Q_n	Q_{n+1}	D
0	0	0
0	1	1
1	0	0
1	1	1

Q_n	J	K	Q_{n+1}	D
0	0	0	0	0
0	0	1	0	0
0	1	0	1	1
0	1	1	1	1
1	0	0	1	1
1	0	1	0	0
1	1	0	1	1
1	1	1	0	0

4) K-Map

D: K \ Q_nJ	$\bar{Q}_n\bar{J}$	$\bar{Q}_nJ$	Q_nJ	$Q_n\bar{J}$
$\bar{K}$	0	1	1	1
K	0	1	0	0

$D = \cancel{J\bar{K}} + Q_n\bar{K} + \bar{Q}_nJ$



circuit diagram

④ T to D.

1) Available T
Required D

2) characteristic of D

Q	D	Q_{n+1}
0	0	0
0	1	1
1	0	0
1	1	1

3) Excitation table of T

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

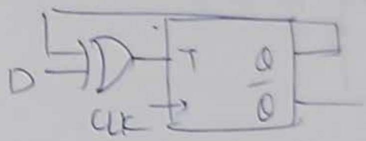
Q	D	Q_{n+1}	T
0	0	0	0
0	1	1	1
1	0	0	1
1	1	1	0

4) K-Map

D \ Q	$\bar{Q}$	Q
$\bar{D}$	0	1
D	1	0

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

5) ckt diagram $= Q \oplus D$



⑤ T to JK

1) Available T
Required JK

2) Characteristic of JK

Q	J	K	Q_{n+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

3) Excitation table of T

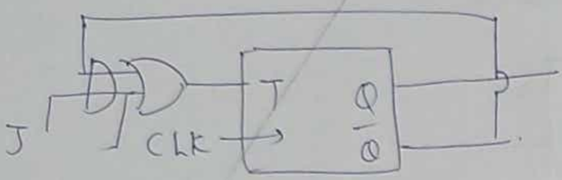
Q	Q_{n+1}	T
0	0	0
0	1	1
1	0	1
1	1	0

Q	J	K	Q_{n+1}	T
0	0	0	0	0
0	0	1	0	0
0	1	0	1	1
0	1	1	1	1
1	0	0	1	0
1	0	1	0	1
1	1	0	1	0
1	1	1	0	1

4) K-Map

T (Q_nJ \ K)		$\bar{Q}_nJ$	Q_nJ	$Q_n\bar{J}$
$\bar{K}$	0	1	0	0
K	0	1	1	1

$T = \bar{Q}_nJ + JK + Q_nK$

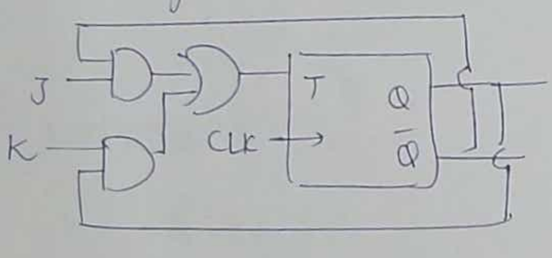


4) K-Map

T (Q_nJ \ K)				
$\bar{K}$	0	1	0	0
K	0	1	1	1

$T = \bar{Q}_nJ + Q_nK$

5) circuit diagram



Note:

① A new F/F is having behaviour as described below. It has 2 i/p x and y when both i/p are same, they are 1, 1. This F/F is going to set else F/F represents results. If i/p are diff and they are 0 and 1, the F/F complements itself. Otherwise it is going to retain the last state.

Which of the following expression in the characteristics expression of new F/F.

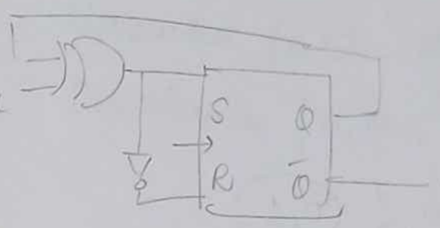
A) $XQ + Y\bar{Q}$

B) $X\bar{Q} + YQ$

C) $X\bar{Q} + Y\bar{Q}$

D) None of the above

(2) Make expression of Q for given ckt 1.



a) XQ

b) $X\bar{Q}$

c) $X \oplus Q$

d) $X \odot Q$.

$S = x \oplus Q$

$R = x \odot Q$

Q	x	Q_{n+1}
0	0	0
0	1	1
1	0	1
1	1	0

Q_{n+1}

$x \backslash Q$		
	0	1
	1	0

$Q_{n+1} = x \oplus Q$.

Sol: (1)

x	y	Q_{n+1}
1	1	1 (set)
0	0	0 (reset)
0	1	$\bar{Q}$
1	0	Q

Q	x	y	Q_{n+1}
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

k-map

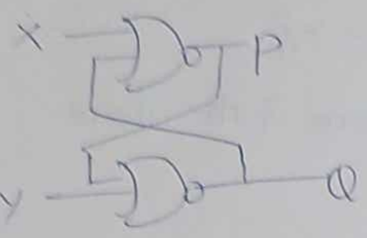
Q_{n+1}

$y \backslash Qx$				
	0	0	1	0
	1	1	1	0

$x\bar{y} + Qx = Q_{n+1}$

(3) If i/p x,y changes from 0,1 to 0,0. output PQ will change from 1,0 to

	x, y	P Q
if	0,1	$\bar{Q}$, 0

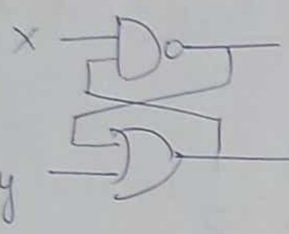


$P = \bar{Q} = 1$

$\bar{Q} = 1 \Rightarrow Q = 0$

x, y	P Q
0 0	$\bar{Q}$ $\bar{P}$ = 1 0

(4) 01 to 11, o/p 1,1 to ? 0,1



$P = \overline{xQ}$

$Q = y + P$.

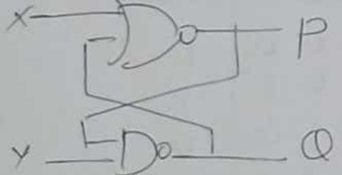
$x, y = 0,1 \qquad P = \overline{xQ}, \quad Q = y + P$

$\qquad 1 = 1 \qquad Q = 1 + P \qquad Q = 1, \; P = 1$

$x, y = 1,1 \qquad P = \overline{1 \cdot 1} = 0 \qquad Q = y + P$

$\qquad\qquad = 1 + 1 = 1$

(5)



11 to 00 then ?

$\boxed{01 \text{ to } 01}$

$P = \overline{x + Q} \qquad x, y = 1, 1 \qquad x, y = 0, 0$

$Q = \overline{yP} \qquad P = 0 \qquad P = \bar{Q} = 0$

$\qquad Q = \bar{P} = 1 \qquad Q = 1$

HW

Q. D latch added functionality

Counter

- In digital electronics, counter is used to count pulses
- It is used as frequency divider

Two types of counters,

UpCounter [0,1,2,..N] - Ex:- Bank machine

DownCounter [N, N-1, ...1,0] - Ex:- sports (foot ball)

classification of counter

1. Synchronous counter - memory elements join in same clock pulse.
2. Asynchronous counter

Synchronous conter :

All memory elements are provided with same clock pulse

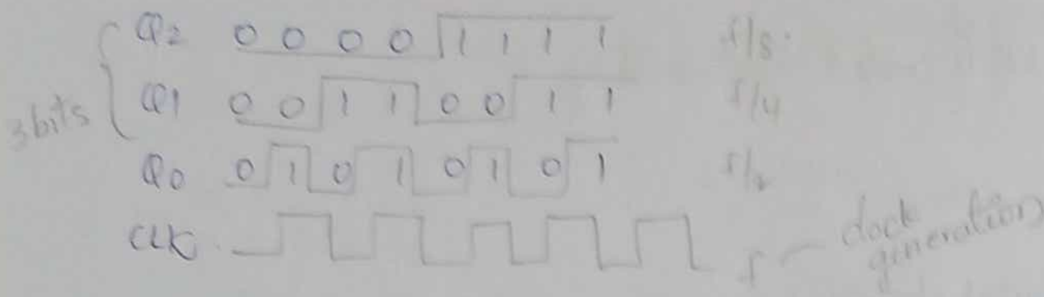
Asynchronous counter :

All memory elements are connected with diff clock signals.

- For n bit counter, Max Count = 2^n

$n = 3$ bit, Max count = 8.

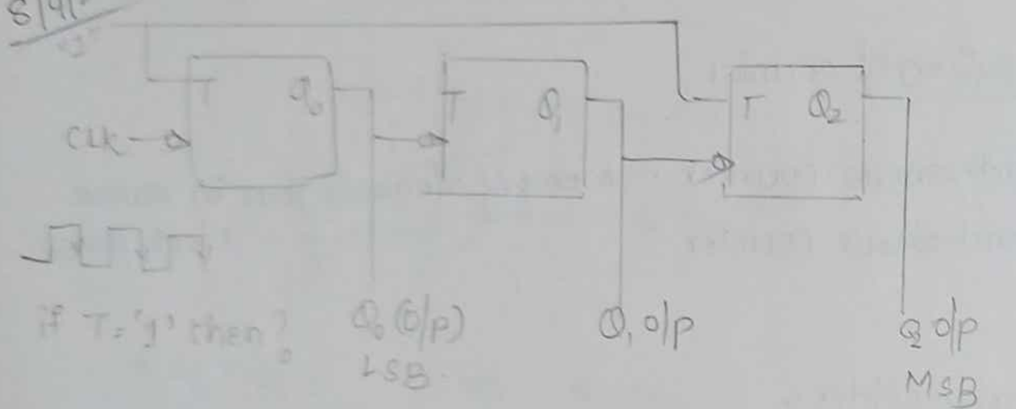
up counter ↓		down counter ↑
	000	
	001	
	010	
	011	
	100	
	101	
	110	
	111	

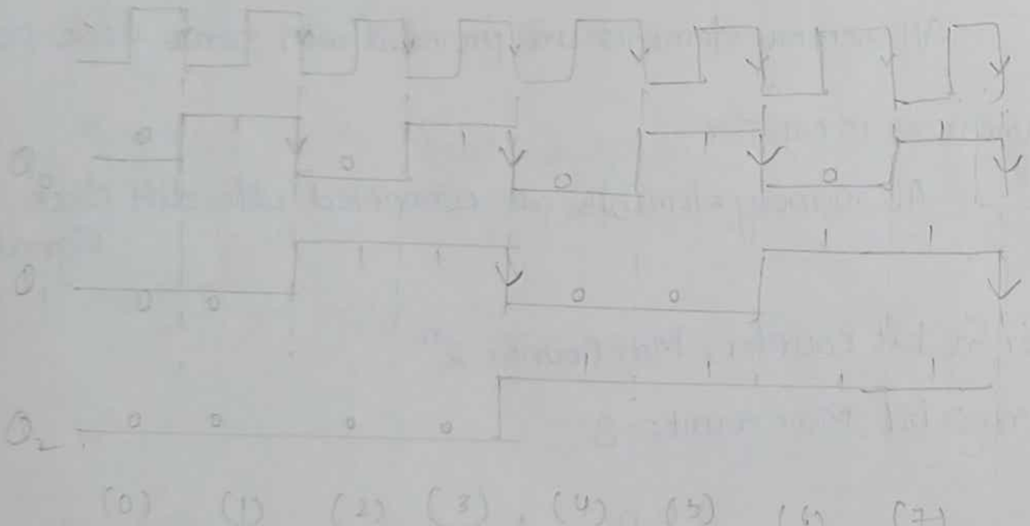


Asynchrounous (Ripple) up Counter :

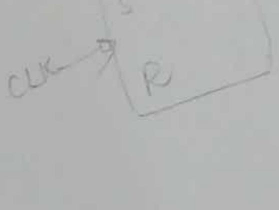
- Here we don't provide same clock signal to all memory elements

8/4/22

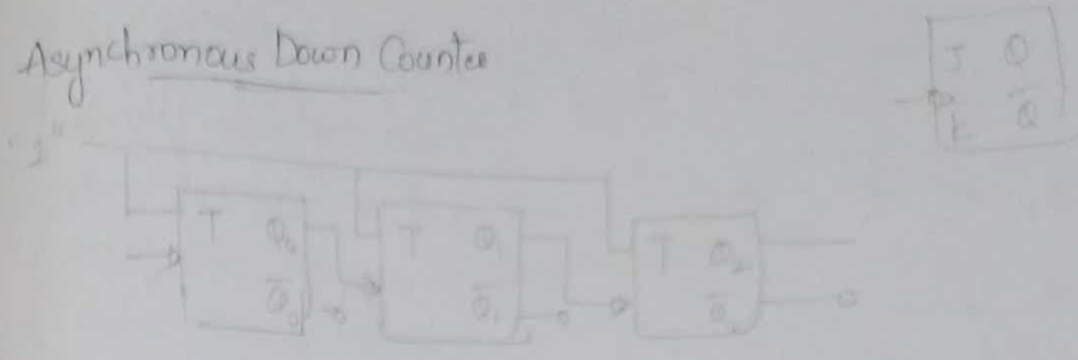


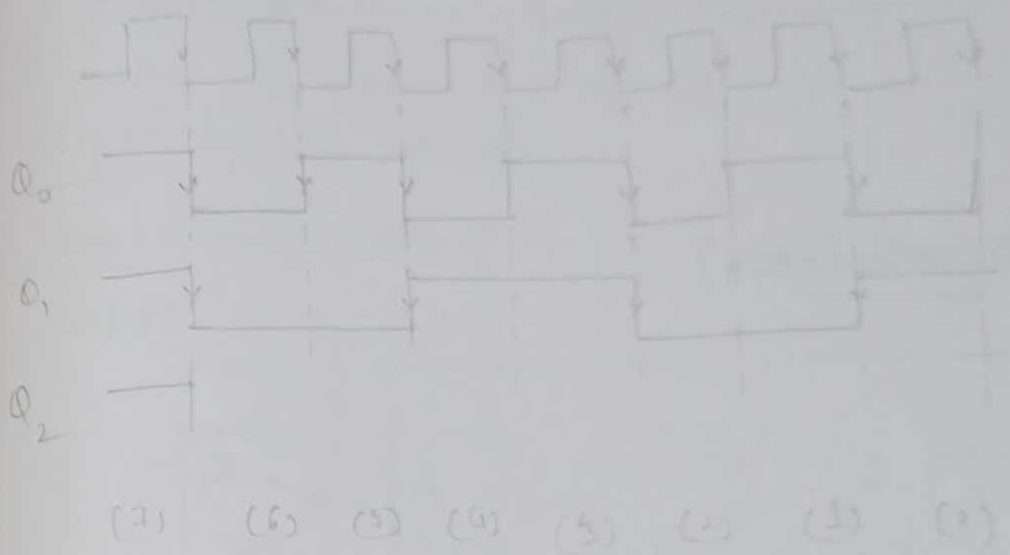


(0 to 7) counter → up counter



Asynchronous Down Counter





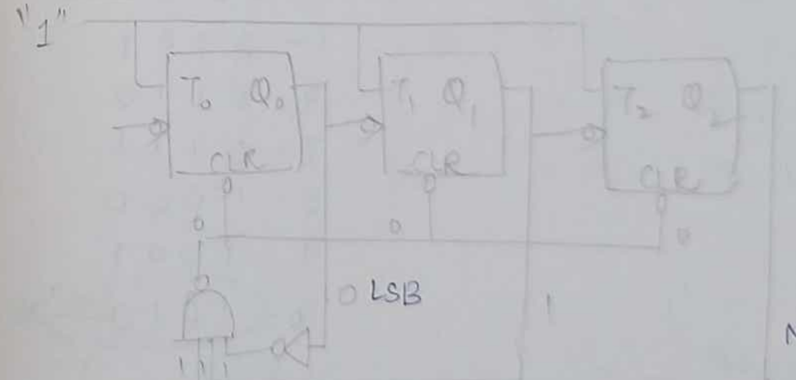
Module counter by Asynchrounous counter

(2^n)

- By 2 bit counter at max we have mod4 counter
- By 3 bits counter at max we have 8 counter
- By 4 bits Counter at max we have 16 counter (2^n)

CLR → clear active low

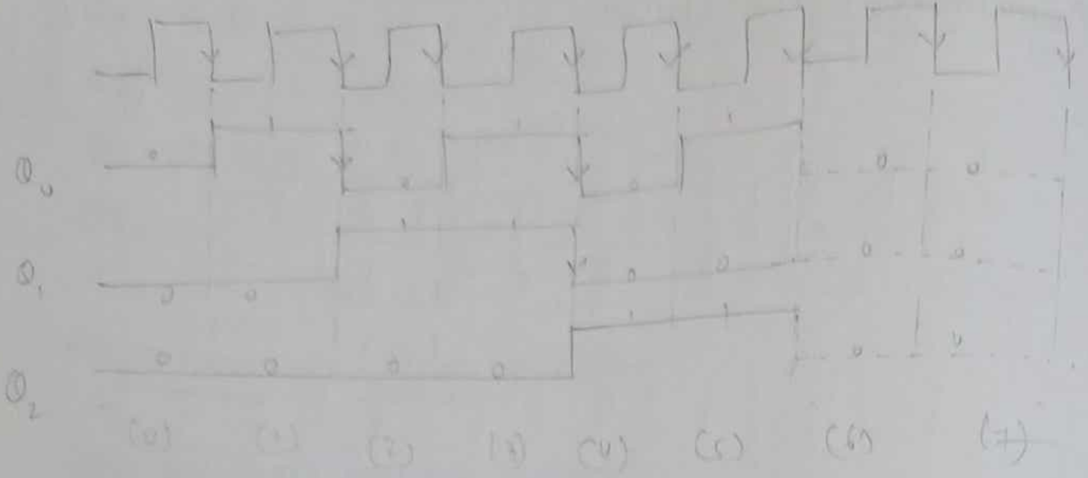
Module 6 Counter :



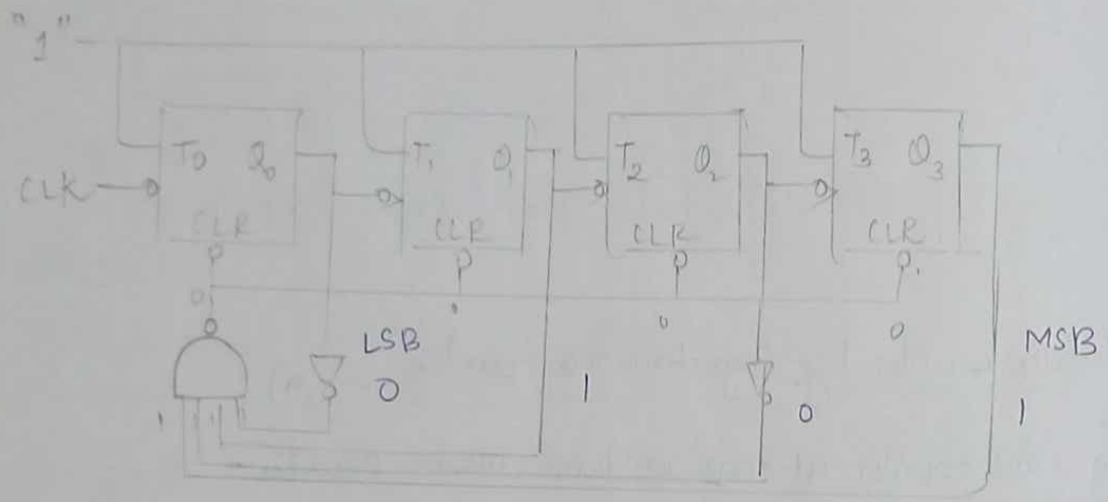
0	0	0	0
1	0	0	1
2	0	1	0
3	0	1	1
4	1	0	0
5	1	0	1
6	1	1	0

MSB

it won't work
it will again go to 0,0,0



BCD counter (0 to 9)



BCD (0 to 9)

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 - X

2-bit Synchronous Counter by JK F/F

- Identify no. of bits and F/F
- Write Excitation table of F/F
- Make state diagram & table
- Solve Boolean Expression
- Make Circuit

"Steps"

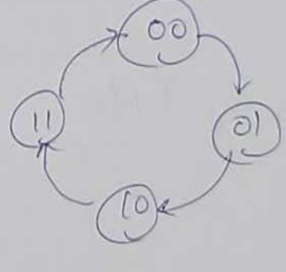
Step 1: $n = 2$ bits F/F = JK

Step 2: Excitation table of JK

Step 3:

Q_n	Q_{n+1}	J	K
0	0	0	x
0	1	1	x
1	0	x	1
1	1	x	0

Step 3: State diagram



Q_1 Q_0	Q_1^+ Q_0^+	J_1 K_1	J_0 K_0
0 0	0 1	0 X	1 X
0 1	1 0	1 X	X 1
1 0	1 1	X D	~~X~~ ~~0~~ 1 X
1 1	0 0	X ~~0~~ 1	X 1

Step 4:

WRONG!!

J_1, Q_1, Q_1^+

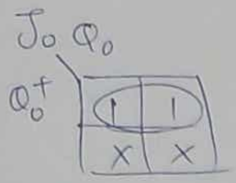
0	X
1	X

$J_1 = Q_1^+$

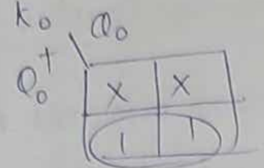
K_1

X	01
X	D

$K_1 = \overline{Q_1^+}$

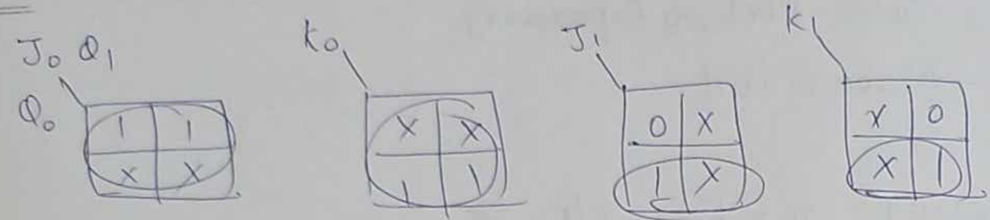


$\overline{Q_0^+} = J_0$



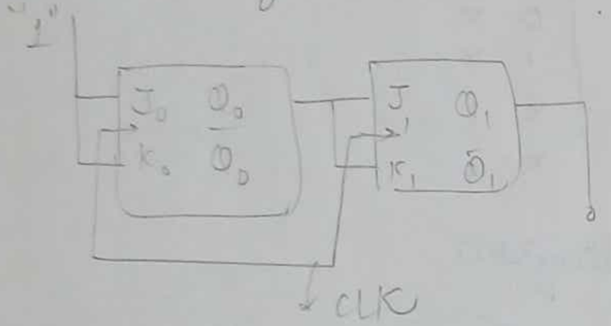
$K_0 = Q_0^+$

Step 4:



$J_0 = 1$ $K_0 = 1$ $J_1 = Q_0$ $K_1 = Q_0$

Step 5: Circuit diagram



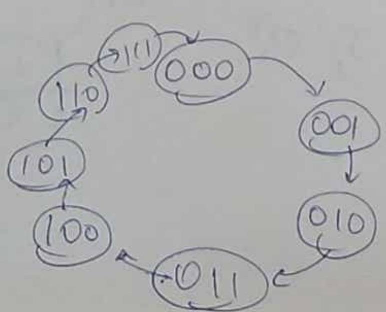
3 bits synchronous Counter by T F/F

Step 1: $n = 3$ bits
F/F = T

Step 2: Excitation table

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

Step 3:



state diagram

State Table:

$Q_2\ Q_1\ Q_0$	$Q_2^+\ Q_1^+\ Q_0^+$	T_2	T_1	T_0
0 0 0	0 0 1	0	0	1
0 0 1	0 1 0	0	1	1
0 1 0	0 1 1	0	0	1
0 1 1	1 0 0	1	1	1
1 0 0	1 0 1	0	0	1
1 0 1	1 1 0	0	1	1
1 1 0	1 1 1	0	0	1
1 1 1	0 0 0	1	1	1

Step 4:

T_2

$Q_0 \backslash Q_2Q_1$	$\bar{Q}_2\bar{Q}_1$	$\bar{Q}_2Q_1$	Q_2Q_1	$Q_2\bar{Q}_1$
$\bar{Q}_0$	0	0	0	0
Q_0	0	1	1	0

$T_2 = Q_1Q_0$

T_1

0	0	0	0
1	1	1	1

$T_1 = Q_0$

T_0

1	1	1	1
1	1	1	1

$T_0 = 1$

Step 5: