Page No.
Date

Binary - Radix - 2
Decimal - Radix - 10
Octal - base - 8
Hexadecimal - base - 16

$(1010.010)_2 \rightarrow 1\times2^3 + 0\times2^2 + 1\times2^1 + 0\times2^0 + 0\times2^{-1} + 1\times2^{-2} + 0\times2^{-3}$

$= (10.25)_{10}$

Convert $(101.01)_2$ into decimal.

$1\times2^2 + 0\times2^1 + 1\times2^0 + 0\times2^{-1} + 1\times2^{-2}$

$\Rightarrow (5.25)_{10}$

Convert 234 into binary.

2	234	
2	117	0
2	58	1
2	29	0
2	14	1
2	7	0
2	3	1
	1	1

$\Rightarrow (11101010)_2$

Find the value of 0.634 in binary.

$.634 \times 2 = 1.268$
$.268 \times 2 = 0.536$
$0.536 \times 2 = 1.112$
$.112 \times 2 = 0.224$
$0.224 \times 2 = 0.448$

$\Rightarrow (.10100)_2$

$31 = (10010)_2$

2	37
2	18 1
2	9 0
2	4 1
2	2 0
	1 0

$0.0625 \times 2 = 0.1350$
$0.1350 \times 2 = 0.270$
$0.270 \times 2 = 0.540$
$0.540 \times 2 = 1.080$
$0.08 \times 2 = 0.16$
$\Rightarrow .0001$

$\Rightarrow (100101.0001)_2$

Convert from binary to octal :-
$(101\ 101\ 000\ 011)_2$

101 — 5 $= (5503)_8$
101 — 5
000 — 1
011 — 3

(11.010 111 1)
$\Rightarrow$ (011 . 010 111 100)$_2$
$\Rightarrow (3.274)_8$

Octal to Binary

bit

Is complement
for $00011100_{(28)_{10}} \Rightarrow 11100011$ (just reversed)

Now for 2's complement, find 1's complement first.
Add 1 to it

$$\begin{array}{r} 11100011 \\ +\quad\quad\quad 1 \\ \hline 11100100 \end{array}$$

$+45H$
$-32H$ $\Rightarrow$ $+45H + (-32H)$

$\Rightarrow$

$45H = (0100\ 0101)_2$
$-32H \Rightarrow$ 2's complement
$32 \Rightarrow 0011\ 0010$
$\Rightarrow 1100\ 1101$

$$\begin{array}{r} 11001101 \\ +\quad\quad\quad 1 \\ \hline 11001110 \end{array}$$

When we add 2 BCD numbers, we may have to go for a correction step where 6 (0110) is added to 1 nibble.

When a nibble is one of the six invalid combinations (>9) or there is a carry from the previous nibble.

23+46

```
  0010 0011
  0100 0110
  ---------
  01101001
    6    9
```

23 +48

```
  0010  0011
  0100  1000
  ----------
  0110  1011   (11)
      + 0110   (6)
  ----------
    1 0001
  ----------
    9

      71
```

28 + 39

```
  0010  1000
  0011  1001
  ----------
  0110 10001          [67]
    ↓   + 0110
  ----------
  6     0111  7
```

Page No.
Date

Binary → Grey

$b_i \rightarrow g_i$
if no. of preceding g_i is even

Binary 0110 1010
Grey 0101 1111

take xor of consecutive bits.

Grey → Binary

$b_i = g_i$
if no. of preceding g_i is odd.

Gray code - 0101 1001

binary - 0110 1110

Convert the following

$(757.25)_{10} = (\quad)_{16}$

2	757	
2	378	1
2	189	0
2	94	1
2	47	0
2	23	1
2	11	1
2	5	1
2	2	1
	1	0

$1 \times 1 + 1 \times 0 \times$

$(23.17)_{10}$

IC / chip contains a large number of such gates.

→ SSI ⇒ small scale integration
→ MSI ⇒ medium scale integration
< 100 gates / chip
→ LSI ⇒ large scale integration
< 1000 gates / chip
→ VLSI ⇒ very large scale integration
$< 10^4$ gates / chip
→ ULSI ⇒ ultra large scale integration
$> 10^8$ gates / chip.

Logic Family to Implement Gates :-

BD

Boolean Algebra

1. $F = \overline{(A+\bar{B})(C+\bar{D})}$

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

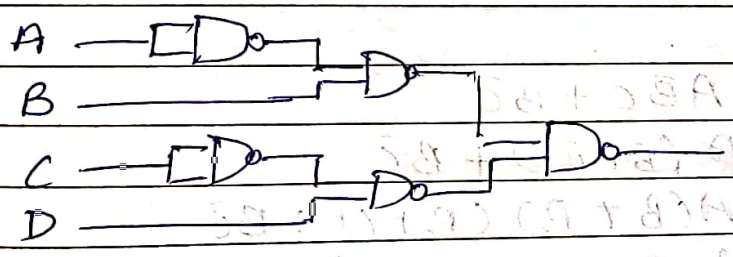
$\Rightarrow \bar{A}\cdot B + \bar{C}\cdot D$

Simplification using NAND gate →

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

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

A — [(A nand A) nand B] nand [(C nand C) nand D]



2. $F = \overline{AB} + \bar{A} + AB$

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

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

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

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

$\Rightarrow \bar{B} \quad \bar{A}+B+\bar{B} \Rightarrow \bar{A}+1 = \overline{1} = 0$

fallacy

③ $F = A[B + \bar{C}(AB + \bar{A}\bar{C})]$

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

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

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

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

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

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

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

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

$\boxed{A\bar{B}}$

④ Show that $AB + A\bar{B}C + B\bar{C}$ is equal to $AC + B\bar{C}$

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

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

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

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

$\Rightarrow AB + AC + B\bar{C}$

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

$AB(C+\bar{C}) + AC + B\bar{C}$

$ABC + AB\bar{C} + AC + B\bar{C}$

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

$\Rightarrow AC + B\bar{C}$

⑤ $A\bar{B}C + B + B\bar{D} + AB\bar{D} + \bar{A}C = B + C$

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

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

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

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

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

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

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

$\Rightarrow B + C$

⑥ Draw the logic diagram using only 2 input NAND gates to implement the following expressions

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

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

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

$\Rightarrow \overline{(\overline{AB\bar{C}\bar{D}} \cdot \overline{AB\bar{C}D}) \cdot (\overline{\bar{A}\bar{B}C\bar{D}} \cdot \overline{\bar{A}\bar{B}\bar{C}D})}$

$\Rightarrow$

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

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

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

Page No.
Date

* Q

$x^2y_2 + xy_1 - 4y = 0$

$x^2\frac{d^2y}{dx^2} + x\frac{dy}{dx} - 4y = 0$

$\Rightarrow$

SOP — Standard SOP form — Standard POS form
POS — $F(A,B,C) = \bar{A}B + \bar{B}C$ — $F(A,B,C) = (\bar{A}+\bar{B})(A+B)$
each term → minterm — each term → maxterm

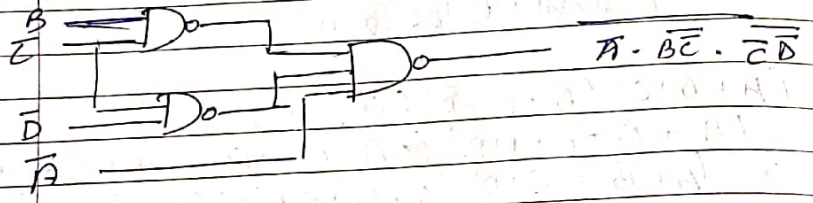
A, B, C = 1, 1, 1

$F(A,B,C) = \bar{A}B + B\bar{C}$
$= \bar{A}B(C+\bar{C}) + \bar{B}C(A+\bar{A})$
$= \bar{A}BC + \bar{A}B\bar{C} + AB\bar{C} + \bar{A}\bar{B}C$
$011 + 010 + 101 + 001$
$m_3 \quad m_2 \quad m_5 \quad m_1$

$F(A,B,C) = \sum(1,2,3,5)$

$F(A,B,C) = (\bar{A}+\bar{B})(A+B)$
$(\bar{A}+\bar{B}+C\bar{C})(A+B+C\bar{C})$
$\rightarrow (\bar{A}+\bar{B}+C)(\bar{A}+\bar{B}+\bar{C})(A+B+C)(A+B+\bar{C})$
$(1+1+0)(1+1+1)(0+0+0)(0+0+1)$
$F(A,B,C) = M_6 \quad M_7 \quad M_0 \quad M_1$

$F(A,B,C) = \prod(0,1,6,7)$



Two variable K-map

A \ B	0	1
0	00 m_0	01 m_1
1	10 m_2	11 m_3

Three variable K-map

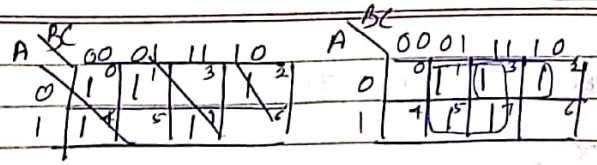
A \ BC	00	01	11	10
0	000 m_0	001 m_1	011 m_3	010 m_2
1	100 m_4	101 m_5	111 m_7	110 m_6

Simplify the boolean funcion

$F(A,B,C) = \Sigma(2,3,4,5)$

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

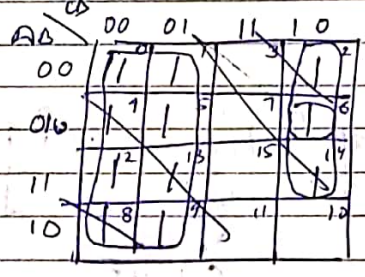
$A\bar{B} + \bar{A}B$



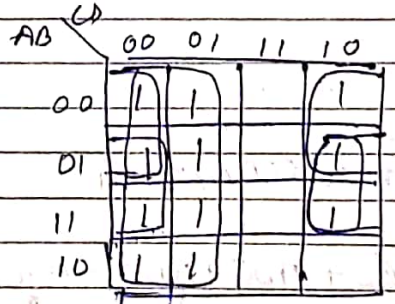
$C + \bar{A}B$

4 Variable K-Map

$F(A,B,C,D) = \Sigma m(0,1,2,4,5,6,8,9,12,13,14)$



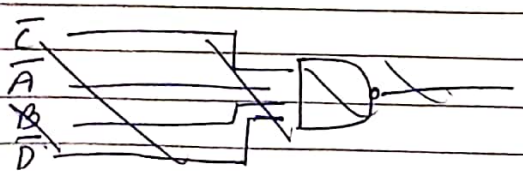
$\bar{C}\bar{C} + C\bar{D}\,A\bar{C}\bar{D} + B\bar{C}\bar{D}$



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

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

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

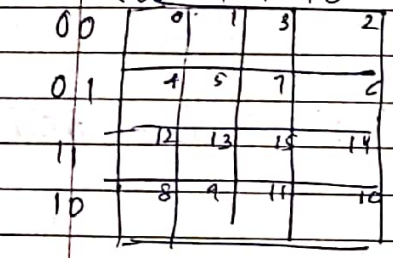


HW

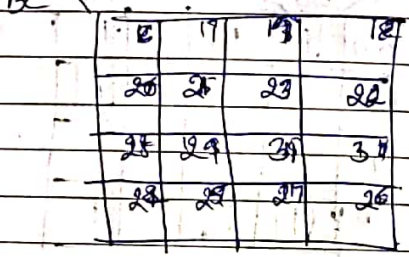
8 Implement a function using K-map
$F = \pi M(2, 8, 9, 10, 11, 14)$ and using universal NOR gate. Convert into SOP.

5- Variable K-Map

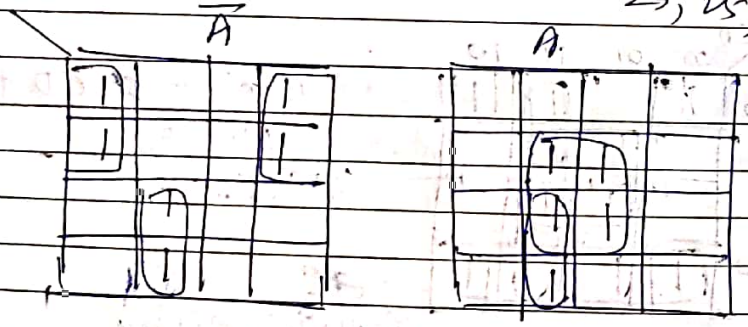
A=0



A=1



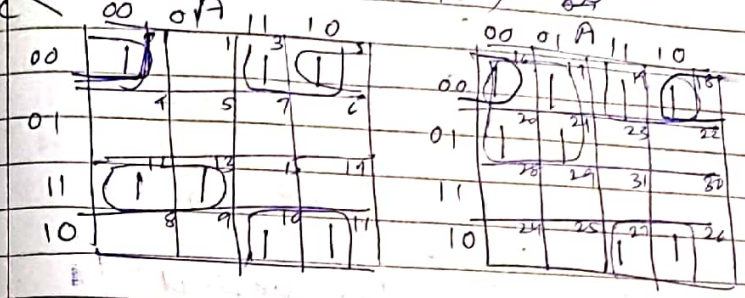
Simplify $F(A, B, C, D, E) = \Sigma m(0, 2, 4, 6, 9, 13, 21, 23, 25, 29, 31)$



$\bar{A}\bar{B}\bar{E} + \bar{A}\bar{B}\bar{D}E + ACE + AB\bar{D}E$

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

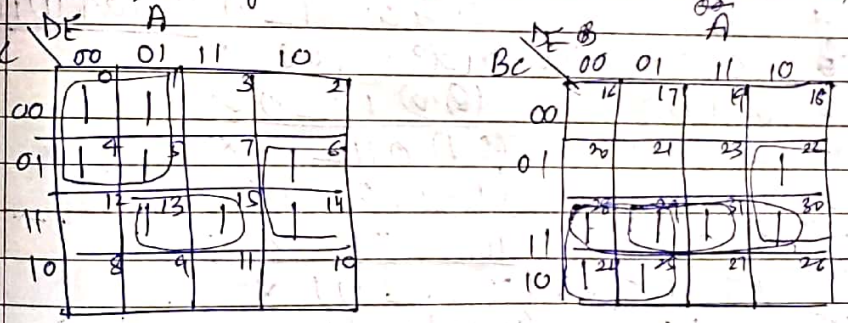
Simplify $m(0, 2, 3, 10, 11, 12, 13, 16, 17, 18, 19, 20, 21, 26, 27)$



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

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

Simplify $0, 1, 4, 5, 6, 13, 14, 15, 22, 24, 25, 28, 29, 30, 31$



$\bar{A}\bar{B}\bar{D} + A\bar{B}\bar{C}E + AB\bar{D} + \bar{B}\bar{C}\bar{E} + CDE + BCE + A\bar{B}C$

Using don't-care combinations, minimise the following & implement using universal following gate.

$1, 5, 6, 12, 13, 14 + D(2, 4)$

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

Q-M method (tabular method)

Q Determine the prime implicant of the function $F(w,x,y,z) = \Sigma m(1,4,6,7,8,9,10,11,15)$

	(a)	(b)	(c)
1	0001 ✓	0 1 0 (1,9) _ 0 0 1	X (8,9,10,11) 1 0 _ _
4	0 1 0 0 ✓	9 1 0 0 1 (4,6) 0 1 _ 0	X (8,10,9,11) 1 0 _ _
8	1000 ✓	10 1 0 1 0 (8,9) 1 0 0 _ ✓	$w\bar{x}$
6	0 1 1 0 ✓	(8,10) 1 0 _ 0 ✓	
9	1 0 0 1 ✓	(6,7) 0 1 1 _ X	
10	1 0 1 0 ✓	(9,11) 1 0 _ 1 ✓	
7	0 1 1 1 ✓	(10,11) 1 0 1 _ ✓	
11	1 0 1 1 ✓	(7,15) _ 1 1 1 X	
15	1 1 1 1	(11,15) 1 _ 1 1 X	

X → prime implicants

Prime Implicant Table

(7,15) – 111 X

(13,15) – 11 – 1 ✓

	1	2	3	5	6	7	8	9	12	13	15
$\bar{X}\bar{Y}Z$	X							X			
$X\bar{Y}Z$						X					X
$\bar{W}Z$	X		X	X		X					
$\bar{W}Y$	X			X				X		X	
$W\bar{Y}$		X	X		X	X					
$W\bar{Y}$							X	X		X	

$\bar{W}Y + X\bar{Y}Z$

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

$F(A,B,C,D,E) = \sum m(0,1,2,8,9,15,17,21,24,25,27,31)$

	(a)	(b)	(c)
0	00000 ✓	(0,1) 0000_ ✓	(0,1,8,9) 0_00_
1	00001 ✓	(0,2) 000_0 X	(1,9,17,25) __ 001
2	00010 ✓	(0,8) 0_000 ✓	" "
8	01000 ✓	(1,9) 0_001 ✓	(8,9,24,25) _100_
9	01001 ✓	(1,17) _0001 ✓	" _100_
17	10001 ✓	(8,9) 0100_ ✓	
24	11000 ✓	(8,24) _1000 ✓	
21	10101 ✓	(9,25) _1001 ✓	(17,25) _001 ✓
25	11001 ✓	(17,21) 10_01 X	
15	01111 ✓	(24,25) 1100_ ✓	
27	11011 ✓	(25,27) 110_1 X	
31	11111 ✓	(15,31) _1111 X	
		(27,31) 11_11 X	

	0	1	2	8	9	15	17	21	24	25	27	31
$\bar{A}\bar{B}\bar{C}\bar{E}$	X	X										
$A\bar{B}\bar{D}E$						X	X					
$AB\bar{C}E$									X	X		
$BCDE$						X					X	
$ABDE$										X	X	
$\bar{A}\bar{C}\bar{D}$	X	X		X	X							
$\bar{C}\bar{D}E$		X			X		X					
$B\bar{C}\bar{D}$				X	X				X	X		

1,21 are left out

$F = \bar{A}\bar{B}\bar{C}\bar{E} + BCDE + A\bar{B}\bar{D}E + B\bar{C}\bar{D}$ ← EPIs
$+ \bar{A}\bar{C}\bar{D}/\bar{C}\bar{D}E + AB\bar{C}E/ABDE$

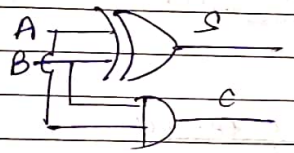
Combinational Logic Design

Half Adder

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

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

$C = A \cdot B$

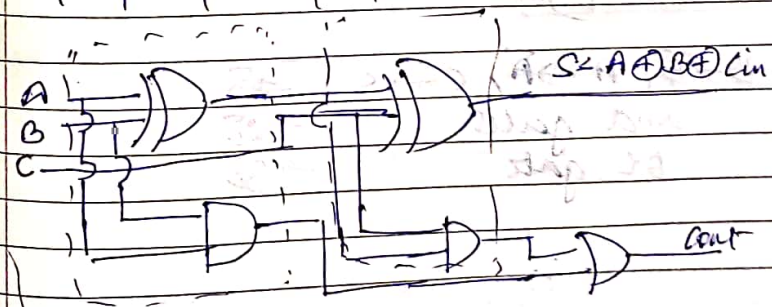


Full Adder

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

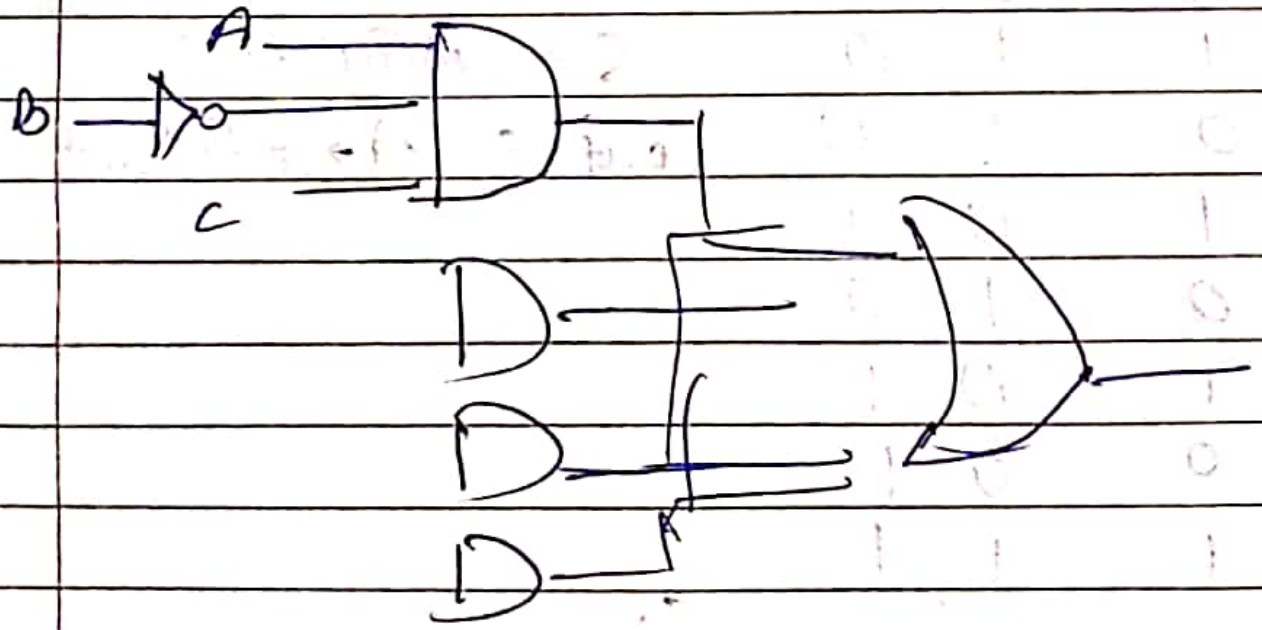
$S = A \oplus B \oplus Cin$

$Cout = AB + BCin + CinA$



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

$S = 3\delta \quad 3\delta$



$B \to \bar{B}$, $A \to \bar{A}$, $C \to \bar{C}$ $\Rightarrow \delta$

and gates $\Rightarrow 2\delta$

OR gate $\Rightarrow 3\delta$

$C_1 = P_0 C_0 + G_0$

$= A_0 B_0 + (A_0 \oplus B_0) C_0$

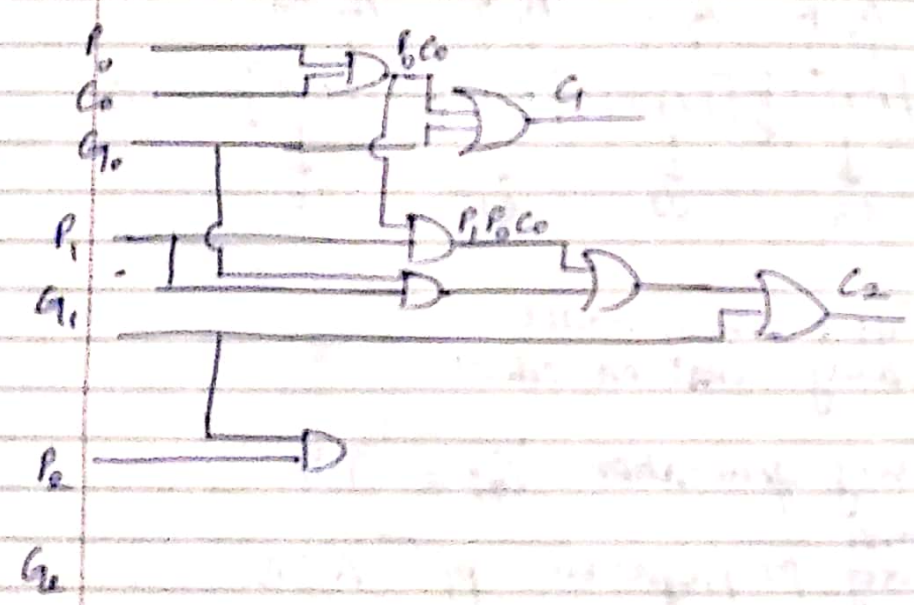
$= A_0 B_0 + B_0 C_0$

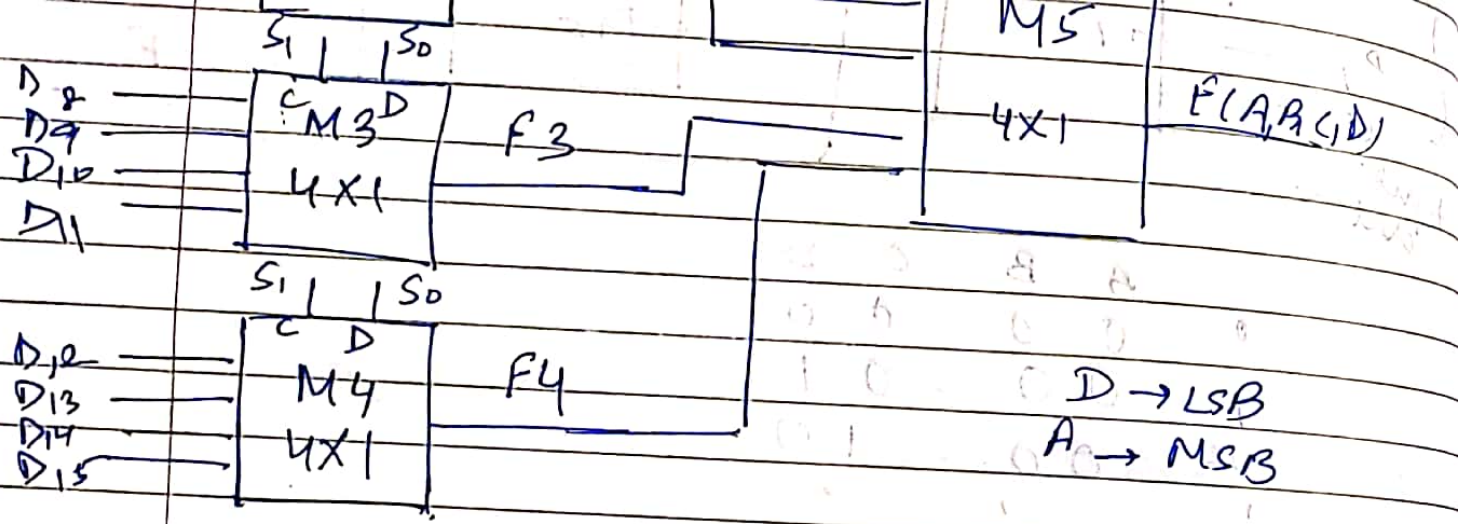
P-1 $C_2 = G_1 + P_1 C_1 = G_1 + P_1 (P_0 C_0 + G_0)$

P-2 $C_3 = G_2 + P_2 C_2$

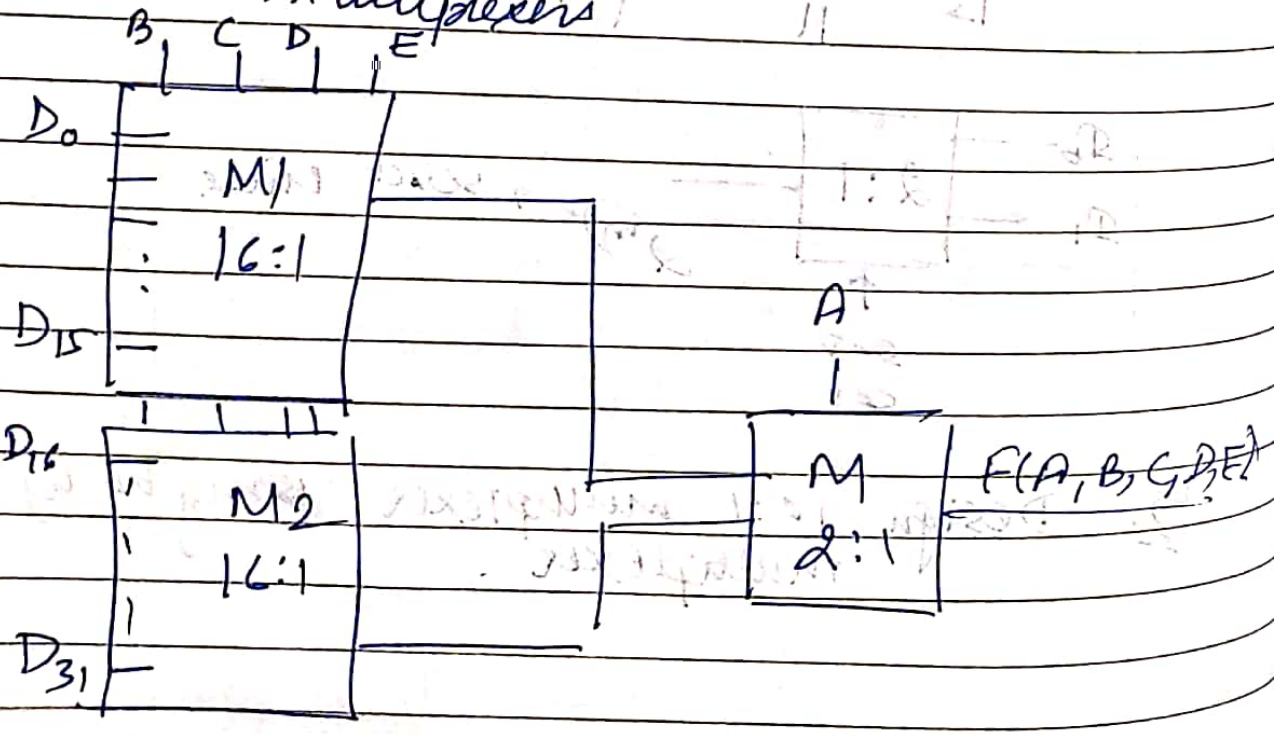
$\Rightarrow C_3 = G_2 + P_2 (G_1 + P_1 P_0 C_0 + P_1 G_0)$

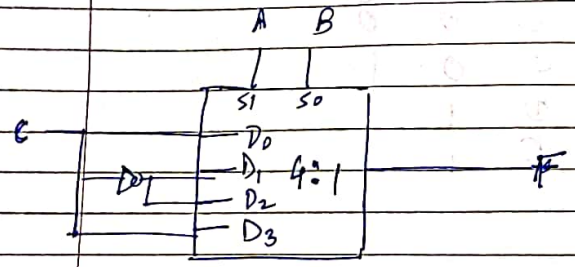
$\Rightarrow C_3 = G_2 + P_2 G_1 + P_2 P_1 P_0 C_0 + P_1 P_2 G_0$





Design 32:1 multiplexer using 2, 16:1 multiplexers and 1 2:1 multiplexers





A	B	C	f	
0	0	0	0	
0	0	1	1	F = C
0	1	0	1	F = $\bar{C}$
0	1	1	0	
1	0	0	1	F = $\bar{C}$
1	0	1	0	
1	1	0	0	
1	1	1	1	F = C

Use a multiplexer having 3 data select input to implement the logic for the function $F = \sum m(0,1,2,3,4,10,11,14,15)$

BCD Adder

A — 1001 — 9

B — 1001 — 9

1 0010 — 18

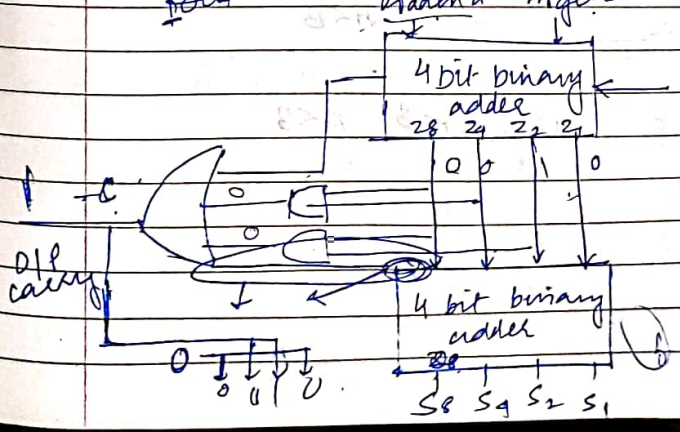
16 2

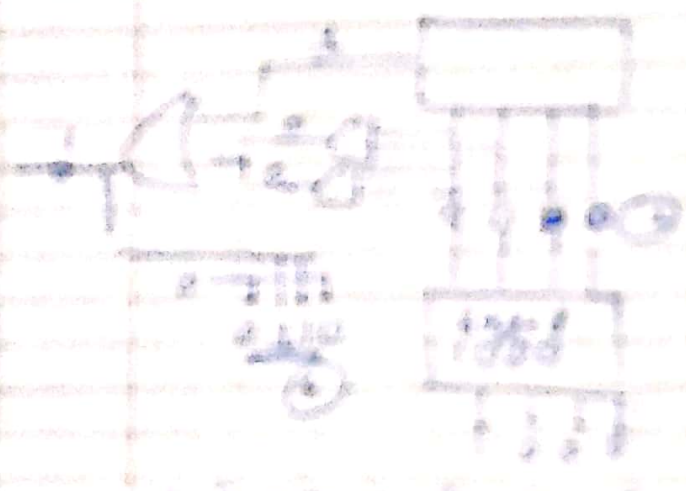
Decimal	Binary Sum					BCD Sum				
S	K	Z_8	Z_4	Z_2	Z_1	C	S_8	S_4	S_2	S_1
0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	1	0	0	0	0	1
2	0	0	0	1	0	0	0	0	1	0
3	0	0	0	1	1	0	0	0	1	1
4	0	0	1	0	0	0	0	1	0	0
5	0	0	1	0	1	0	0	1	0	1
6	0	0	1	1	0	0	0	1	1	0
7	0	0	1	1	1	0	0	1	1	1
8	0	1	0	0	0	0	1	0	0	0
9	0	1	0	0	1	0	1	0	0	1
10	0	1	0	1	0	1	0	0	0	0
11	0	1	0	1	1	1	0	0	0	1
12	0	1	1	0	0	1	0	0	1	0
13	0	1	1	0	1	1	0	0	1	1
14	0	1	1	1	0	1	0	1	0	0
15	0	1	1	1	1	1	0	1	0	1
16	1	0	0	0	0	1	0	1	1	0
17	1	0	0	0	1	1	0	1	1	1
18	1	0	0	1	0	1	1	0	0	0

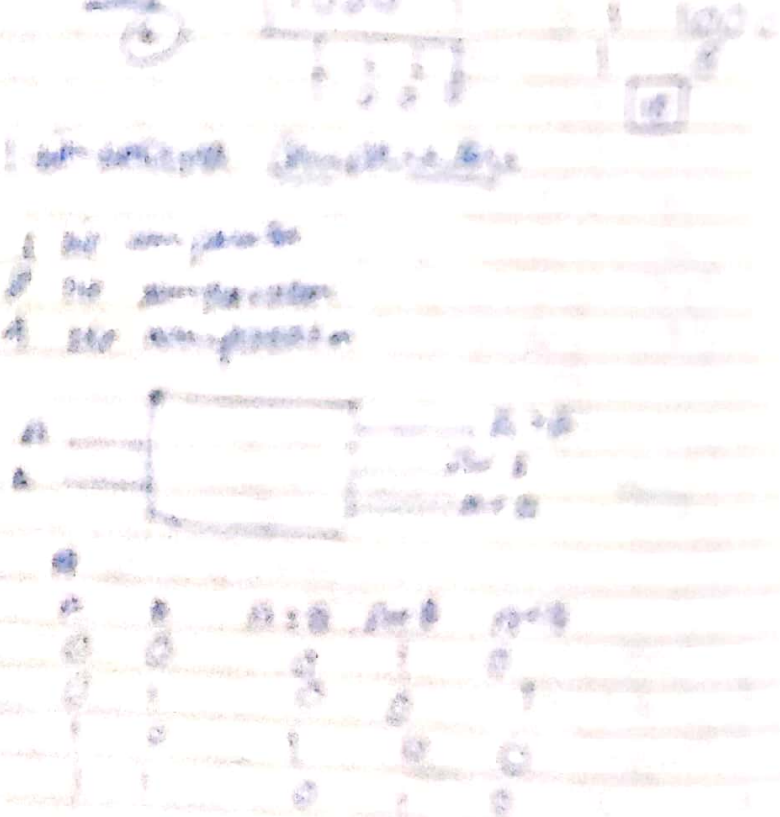
nos. greater than 9

$C = K + Z_8Z_4 + Z_8Z_2$

1000 1001C







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

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

$A = B \Rightarrow A \odot B \cdot AB + \bar{A}\bar{B}$

$A \rightarrow A_1 A_0$

$B \rightarrow B_1 B_0$

$A = A_3 A_2 A_1 A_0$

$B = B_3 B_2 B_1 B_0$

N-bit comparator

(1) 2 bit comparator

$A: A_1 A_0$

$B: B_1 B_0$

(1) $A = B$

$A_0 = B_0 = x_0 = A_0B_0 + \bar{A_0}\bar{B_0} = A_0 \odot B_0$

$A_1 = B_1 \Rightarrow A_1 \odot B_1$

$\Rightarrow \boxed{x_1 \cdot x_0}$

(2) $A > B$

$(A_1 > B_1)$ or $[(A_1 = B_1)$ and $(A_0 > B_0)]$

$A_1\bar{B_1} + [(A_1 \odot B_1) \cdot (A_0\bar{B_0})]$

$\Rightarrow \underline{A_1\bar{B_1} + x_1 A_0\bar{B_0}}$

(3) $A < B$

$\Rightarrow \underline{\bar{A_1}B_1 + x_1\bar{A_0}B_0}$

(iv) 4-bit comparator

$A : A_3 A_2 A_1 A_0$
$B : B_3 B_2 B_1 B_0$

(1) $A = B$ $\quad x_3 \cdot x_2 \cdot x_1 \cdot x_0$

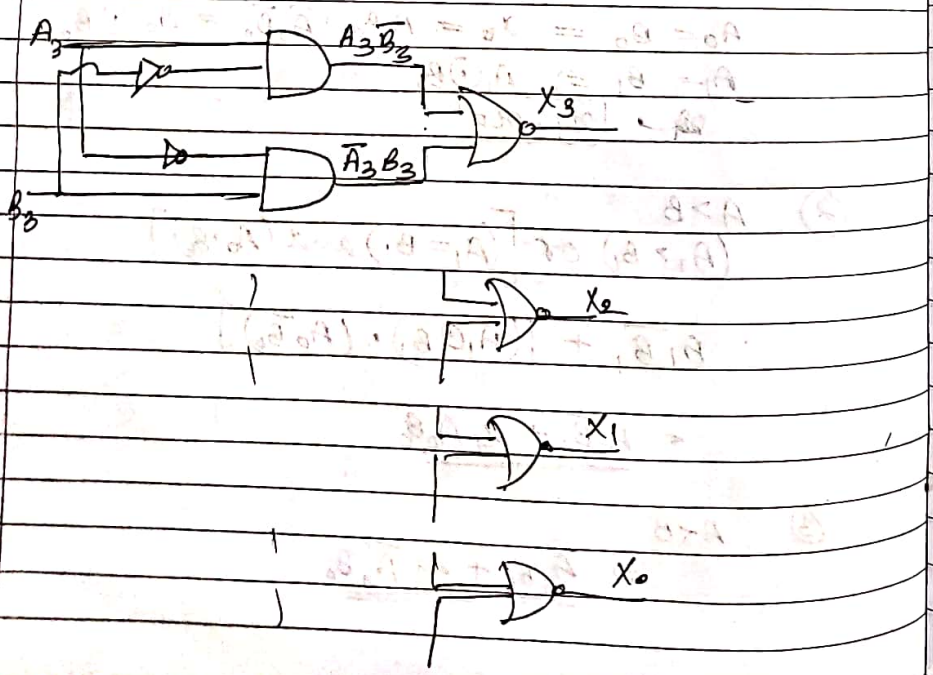
(2) $A > B$

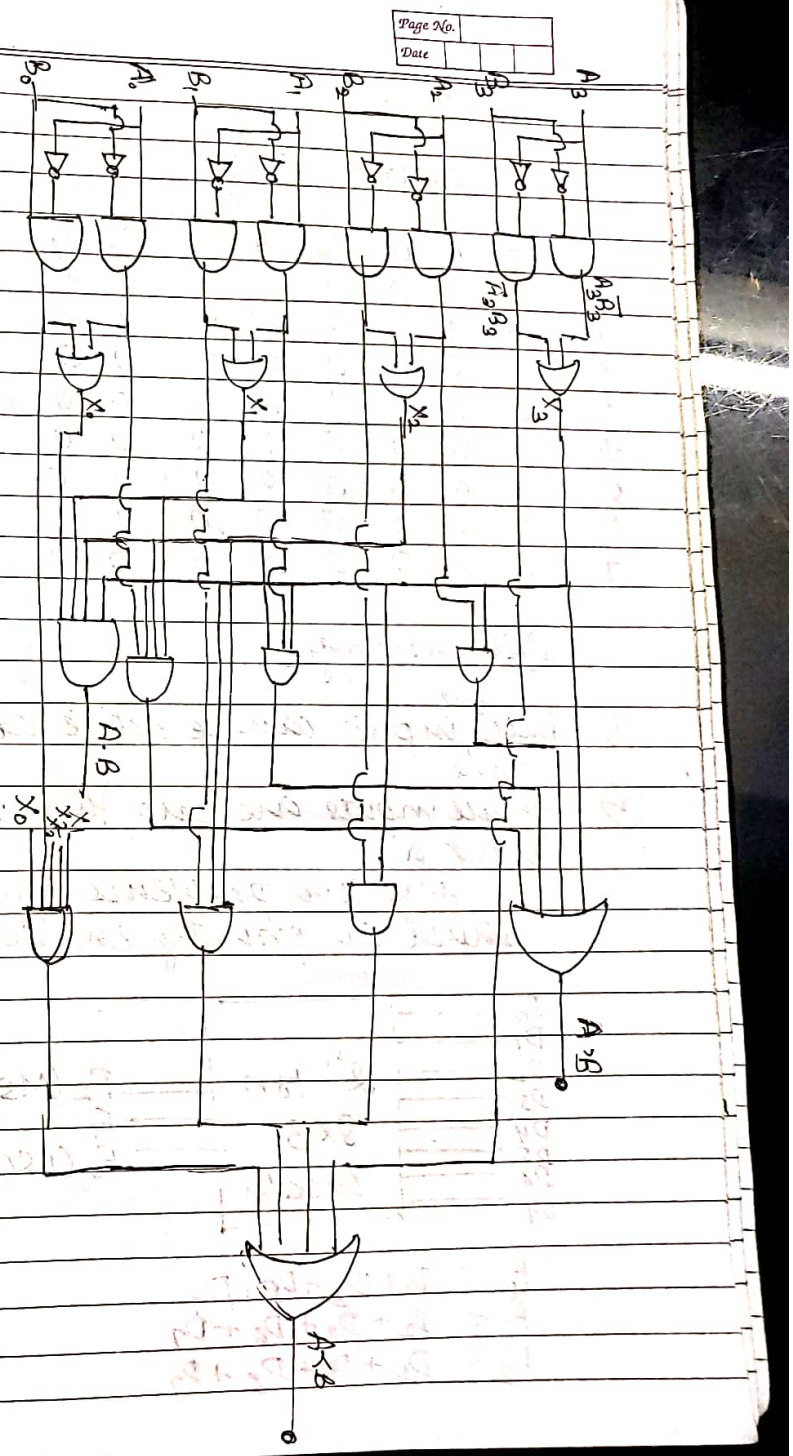
→ $(A_3 = 1$ and $B_3 = 0)$ ~~OR~~ $(A_3 = B_3$ and
→ $(A_3 = B_3$ and $A_2 = 1$ and $B_2 = 0)$
→ $A_3 = B_3$ and $A_2 = B_2$ and $A_1 = 0 (\& B_1 = 0)$
→ $(A_3 = B_3$ and $A_2 = B_2$ and $A_1 = B_1$ & $A_0 = 1 \& B_0 = 0)$

$A_3\overline{B_3} + x_3 A_2 \overline{B_2} + x_3 \cdot x_2 A_1 \overline{B_1} + x_3 \cdot x_2 x_1 A_0 \overline{B_0}$

(3) $A < B$

$\overline{A_3} B_3 + x_3 \overline{A_2} B_2 + x_3 \cdot x_2 \cdot \overline{A_1} B_1 + x_3 \cdot x_2 x_1 \overline{A_0} B_0$





Encoder

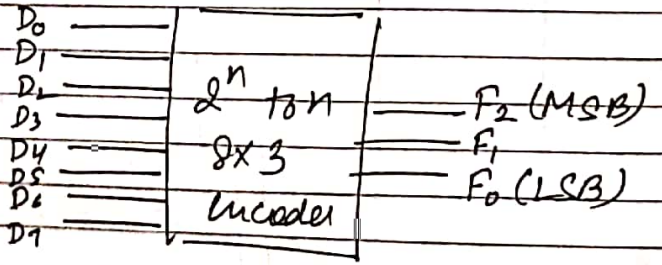
Octal to Binary

	D_0	D_1	D_2	D_3	D_4	D_5	D_6	D_7	F_2	F_1	F_0
0	1	0	0	0	0	0	0	0	0	0	0
1	0	1	0	0	0	0	0	0	0	0	1
2	0	0	1	0	0	0	0	0	0	1	0
3	0	0	0	1	0	0	0	0	0	1	1
4	0	0	0	0	1	0	0	0	1	0	0
5	0	0	0	0	0	1	0	0	1	0	1
6	0	0	0	0	0	0	1	0	1	1	0
7	0	0	0	0	0	0	0	1	1	1	1

Disadvantages

1. Only one input can be active at any given time
2. If all inputs are zero, there is no valid input

→ These two problems are mainly solved in priority encoder.



$F_0 = D_1 + D_3 + D_5 + D_7$
$F_1 = D_2 + D_3 + D_6 + D_7$
$F_2 = D_4 + D_5 + D_6 + D_7$

Priority Encoder

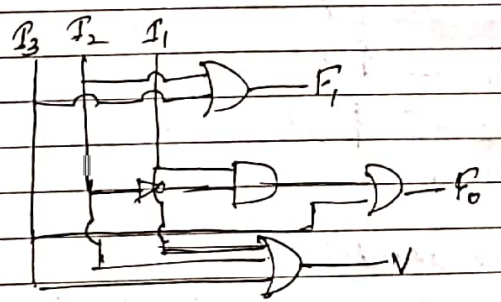
The operation of priority encoder is such that if two or more inputs are equal to 1 at the same time then the input having the highest priority will take precedence.

4 Input Priority Encoder

I_3	I_2	I_1	I_0	F_1	F_0	V
0	0	0	0	X	X	0
0	0	0	1	0	0	1
0	0	1	X	0	1	1
0	1	X	X	1	0	1
1	X	X	X	1	1	1
↓			↓			
highest priority			least priority			

V → valid bit indicator
0 → no valid bit.

$F_1 = I_2 + I_3$ $\quad F_0 = \overline{I_3} + I_1 \overline{I_2}$

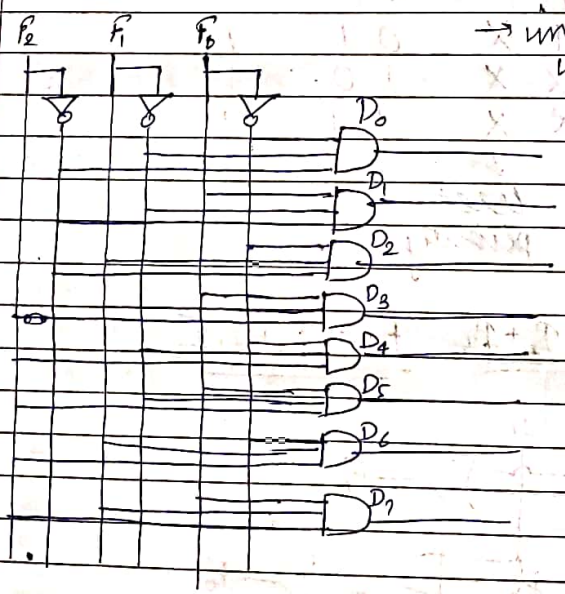


Page No.
Date

Decoders

3 to 8 line decoders

	I/P			Outputs							
	F_2	F_1	F_0	D_0	D_1	D_2	D_3	D_4	D_5	D_6	D_7
0	0	0	0	1	0	0	0	0	0	0	0
1	0	0	1	0	1	0	0	0	0	0	0
2	0	1	0	0	0	1	0	0	0	0	0
3	0	1	1	0	0	0	1	0	0	0	0
4	1	0	0	0	0	0	0	1	0	0	0
5	1	0	1	0	0	0	0	0	1	0	0
6	1	1	0	0	0	0	0	0	0	1	0
7	1	1	1	0	0	0	0	0	0	0	1



→ implemented using AND gate

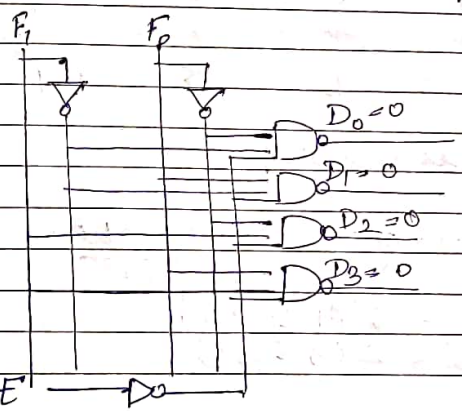
2 to 4 line decoder

Decoder may operate with complemented & uncomplemented output.

Page No.
Date

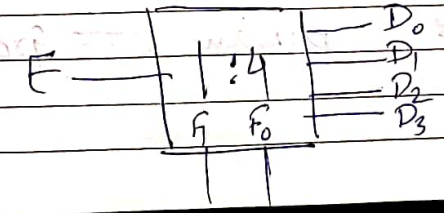
E	F_1	F_0	D_0	D_1	D_2	D_3
1	X	X	1	1	1	1
0	0	0	0	1	1	1
0	0	1	1	0	1	1
0	1	0	1	1	0	1
0	1	1	1	1	1	0

→ implemented using NAND gate



A decoder with enabled input can function as demultiplexer which is a circuit that recieves information from a single line and directs it to 2^n possible output lines.

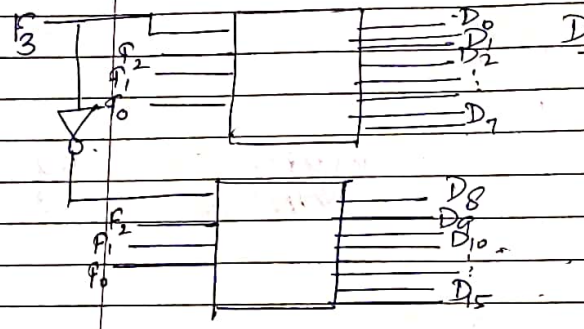
Because decoder and demultiplexer operations are obtained from the same circuit a decoder with an enabled input is also referred as decoder/demultiplexer.



Page No.
Date

2 to 4

Realise a 4 2:16 line decoder with enabled input using 2 3:8 line decoders.



Decimal	F_3	F_2	F_1	F_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
⋮	⋮	⋮	⋮	⋮
15	1	1	1	1

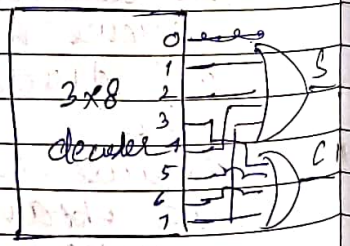
Here, F_3 acts as enabler.

Combinational Logic Implementation

~~SC(x,y,z)~~ $S(x,y,z) = \Sigma(1,2,4,7)$

$C(x,y,z) = \Sigma(3,5,6,7)$

MSB X	Y	LSB Z	S	C
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1



Code Converters

Binary ⟷ Grey code

Page No.
Date

Design 4-bit binary to gc converter

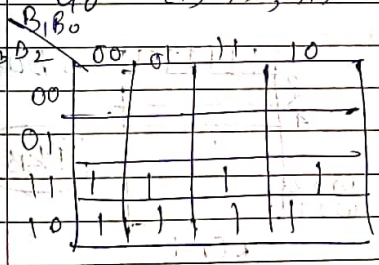
	Binary I/P B_3	B_2	B_1	B_0	GC O/P G_3	G_2	G_1	G_0
0	0	0	0	0	0	0	0	0
1	0	0	0	1	0	0	0	1
2	0	0	1	0	0	0	1	1
3	0	0	1	1				
15	1	1	1	1	1	0	0	0

$G_3 = \{8,9,10,11,12,13,14,15\}$

$G_2 = \{4,5,6,7,8,9,10,11\}$

$G_1 = \{2,3,4,5,10,11,12,13\}$

$G_0 = \{1,2,5,6,9,10,13,14\}$

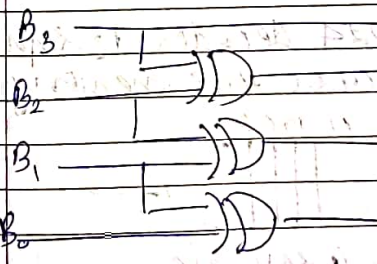


$G_3 = B_3$

$G_2 = \bar{B}_3 B_2 + B_3 \bar{B}_2$

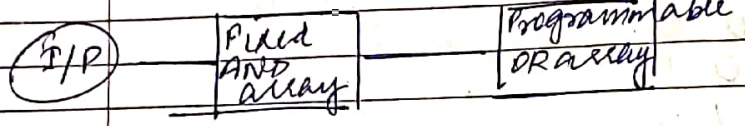
$G_1 = B_2 \bar{B}_1 + \bar{B}_2 B_1$

$G_0 = \bar{B}_1 B_0 + B_1 \bar{B}_0$

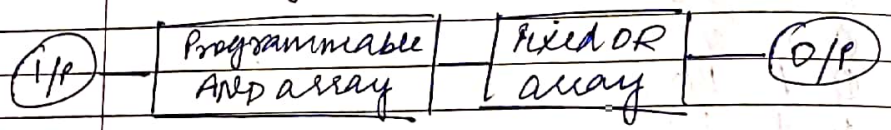


PLD's - Programmable Logic Devices

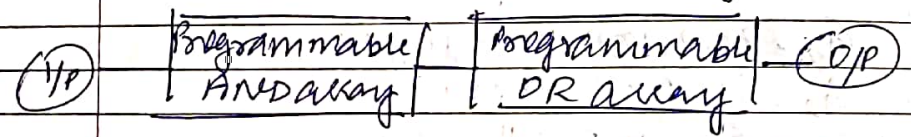
(a) PROM (Programmable Read Only Memory)



(b) PAL (Programmable Array Logic)



(c) PLA (Programmable Logic Array)



1 KB = 1024 bytes = 2^{10}

2 KB = 2^{11}

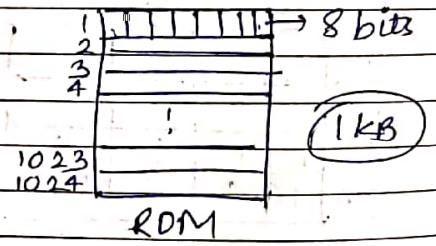
4 KB = 2^{12}

8 KB = 2^{13}

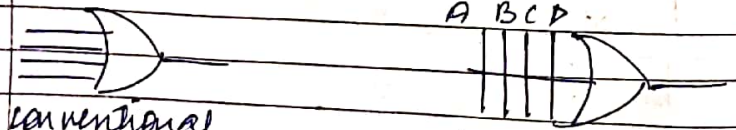
!

1 MB = 2^{20}

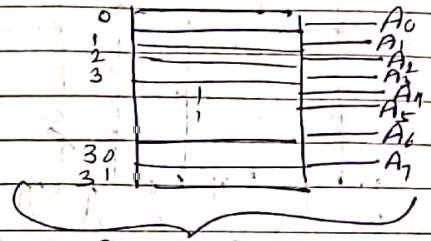
1 GB = 2^{30}



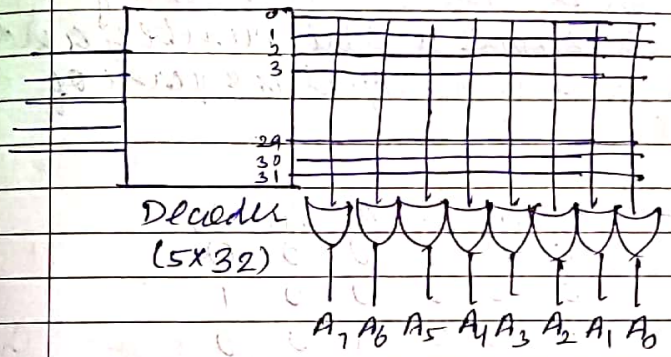
10 : 1024 decoder is required to access this memory.



To mark inputs, put a cross.



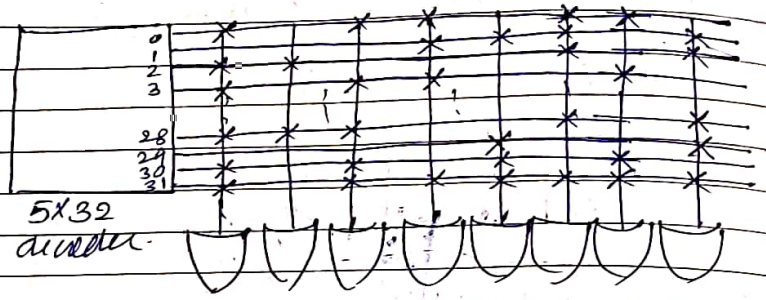
32 x 8 ROM.



ROM truth table

	Input					Output							
	I_4	I_3	I_2	I_1	I_0	A_7	A_6	A_5	A_4	A_3	A_2	A_1	A_0
0	0	0	0	0	0	1	0	1	1	0	1	1	0
1	0	0	0	0	1	0	0	0	1	1	1	0	1
2	0	0	0	1	0	1	1	0	0	0	1	0	1
3	0	0	0	1	1	1	0	1	1	0	0	1	0
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
28	1	1	1	0	0	1	1	1	0	0	1	0	1
29	1	1	1	0	1	0	0	0	0	1	0	0	1
30	1	1	1	1	0	1	0	1	0	1	0	1	0
31	1	1	1	1	1	1	0	1	1	1	1	1	1

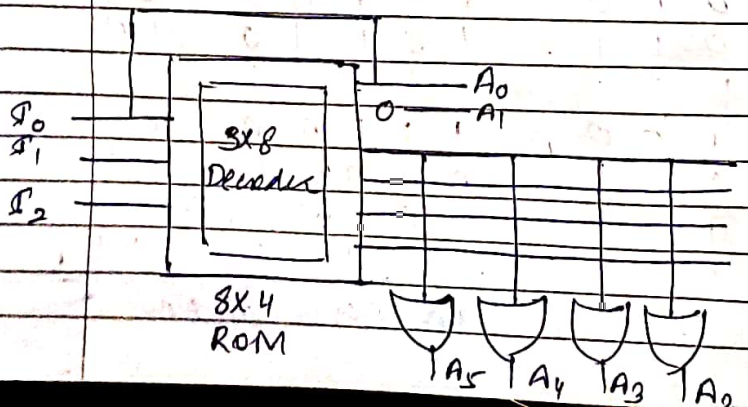
random data



Design a combinational circuit using a ROM which accepts a 3 bit number and outputs a binary no. equal to square of input no.

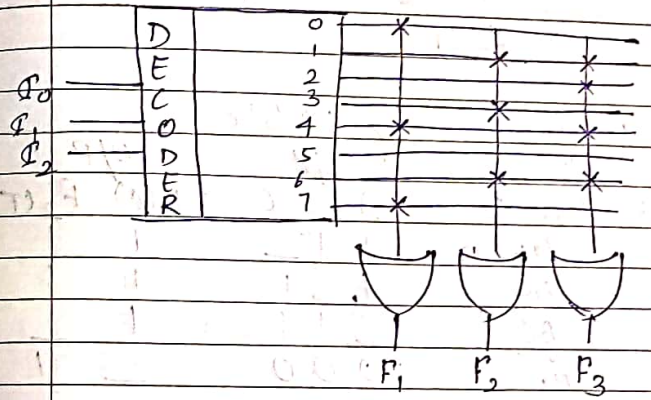
	I_2	I_1	I_0	A_5	A_4	A_3	A_2	A_1	A_0
0	0	0	0	0	0	0	0	0	0
1	0	0	1	0	0	0	0	0	1
2	0	1	0	0	0	0	1	0	0
3	0	1	1	0	0	1	0	0	1
4	1	0	0	0	1	0	0	0	0
5	1	0	1	0	1	1	0	0	1
6	1	1	0	1	0	0	1	0	0
7	1	1	1	1	1	0	0	0	1

$A_0 = I_0$ $\quad$ $A_1 = 0$



Realise the following functions using PROM of 8x3 –

$F_1 = \Sigma m(0,4,7)$ $\quad$ $F_2 = \Sigma m(1,3,6)$ $\quad$ $F_3 = \Sigma m(1,2,4,6)$



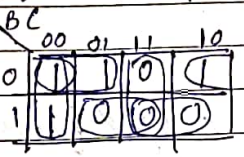
H.W. Design a combinational ckt using PROM which accepts a 3 bit binary no. and generates its equivalent excess 3 code

PLA

Ex: $F_1(ABC) = \Sigma(0,1,2,4)$

$F_2(ABC) = \Sigma(0,5,6,7)$

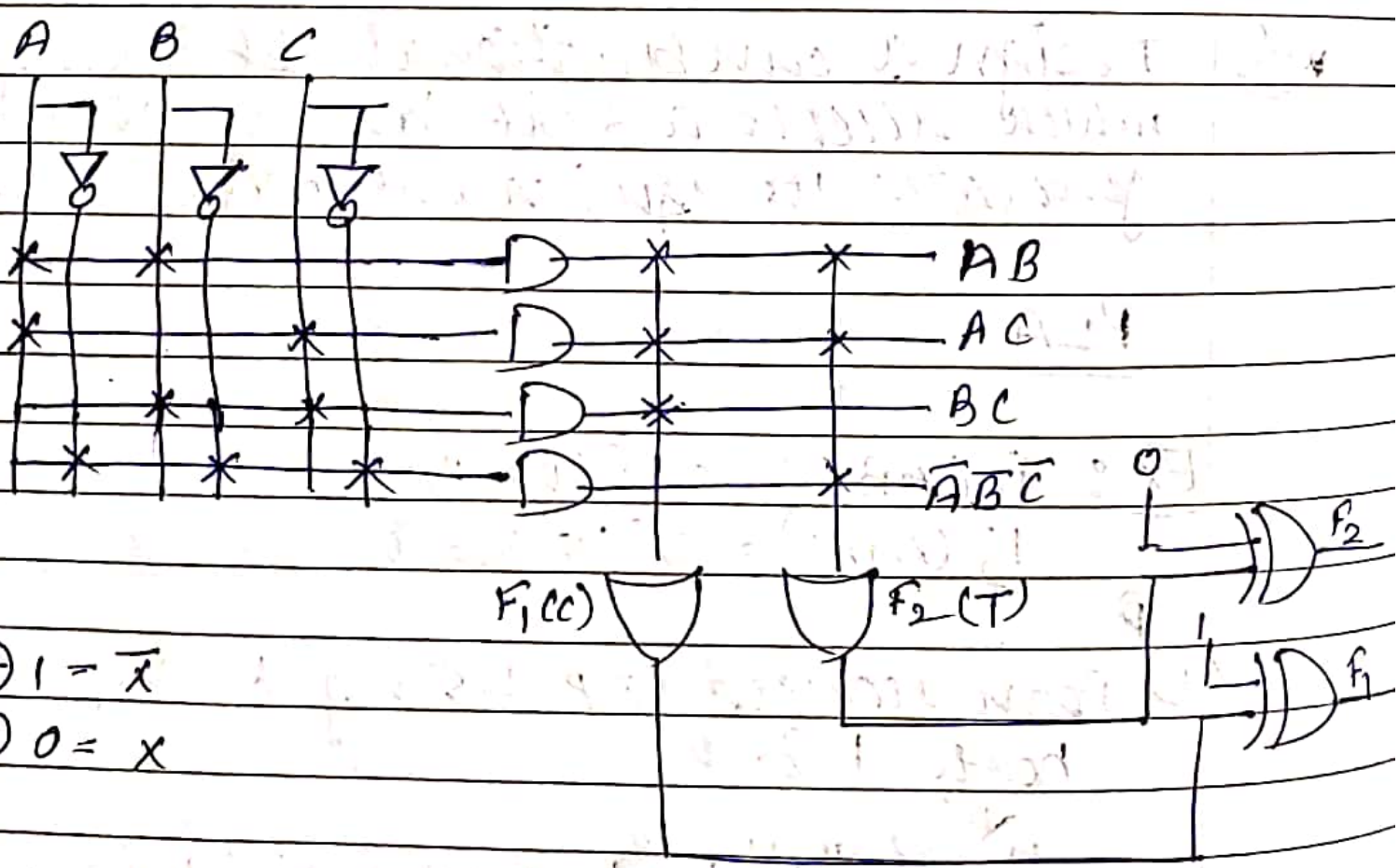
(1) Form reduced exp. using Kmap for both 1 & 0



$F_1(T) = \bar{A}\bar{C} + \bar{A}\bar{B} + \bar{B}\bar{C}$

$F_1(C) = AB + AC + BC$

2	AC	1	—	1	1
3	BC	—	1	1	—
4	$\bar{A}\bar{B}\bar{C}$	0	0	0	—



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

$X \oplus 0 = X$

Implement with added function

$F_3 = \Sigma(0,3,5,7)$

PAL (Programmable Array logic)

Implement the following boolean function using PAL with 4 inputs and 3 o/p's AND-OR structure and write PAL programming table.

$F_1(ABCD) = \Sigma m(2,12,13)$ $F_2(ABCD) = \Sigma m(7,8,9,10,11,12,13,14,15)$

$F_3(ABCD) = \Sigma m(0,1,3,4,5,6,7,8,10,11,15)$

$F_4(ABCD) = \Sigma m(1,2,8,12,13)$

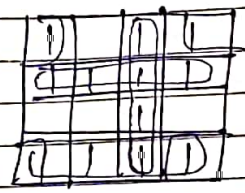
AB \ CD	00	01	11	10
00		1		1
01				
11				
10	1	0		

$F_1 = \bar{A}\bar{B}C\bar{D} + AB\bar{C}$

$F_4 = \bar{A}\bar{B}C\bar{D} + AB\bar{C} + \bar{A}\bar{B}\bar{C}D + A\bar{C}\bar{D} = F_1 + \bar{A}\bar{B}\bar{C}D + A\bar{C}\bar{D}$

$F_2 = A + BCD$

$F_3 = \bar{A}B + CD + \bar{B}\bar{D}$



PAL Programming table

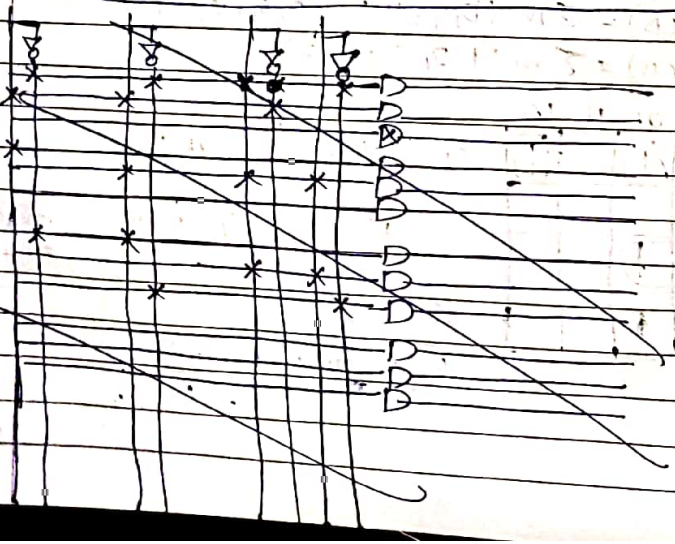
Product term	A	B	C	D	F1	F2	F3	F4
	AND i/p				o/p			
1	0	0	1	0	—			
2	1	1	0	—	—			
3								
4	1	—	—	—				
5	—	1	1	1				
6								
7	0	1	—	—				
8	—	1	1	—				
9	—	0	0	—				
10	—	—	—	1				
11	0	0	0	1	—			
12	1	—	0	0	—			

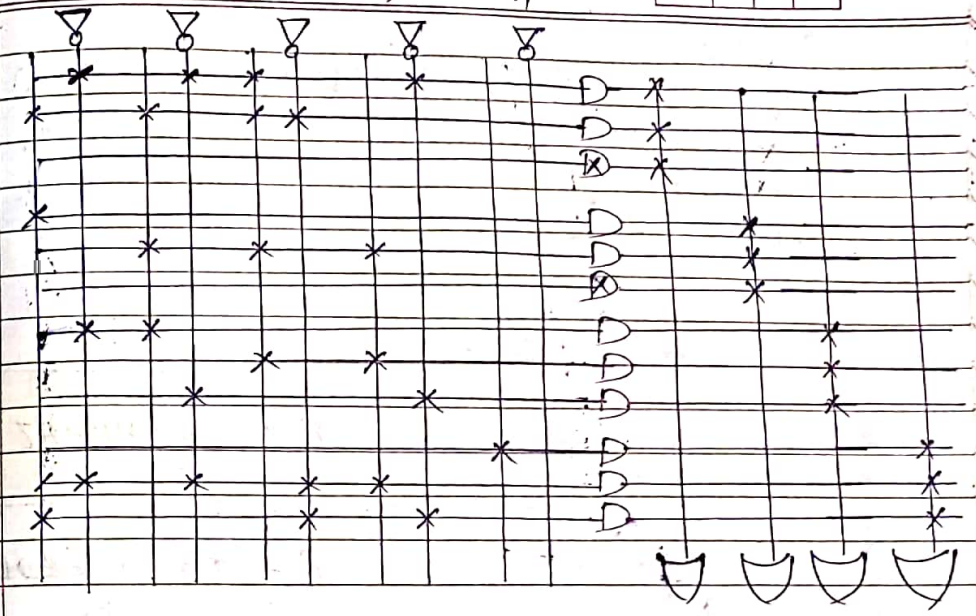
$F_1 = \bar{A}\bar{B}C\bar{D} + AB\bar{C}$

$F_2 = A + BCD$

$F_3 = \bar{A}B + CD + B\bar{D}$

$F_4 = F_1 + \bar{A}B\bar{C}D + A\bar{C}\bar{D}$



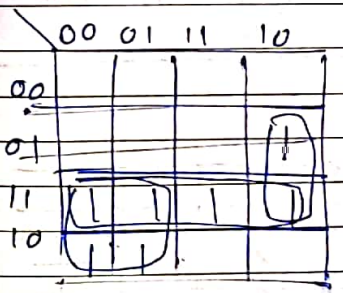


$F_1(ABCD) = \Sigma m(6,8,9,12-15)$

$F_2(ABCD) = \Sigma m(1,4-7,10-13)$

$F_3(ABCD) = \Sigma m(4-7,10-11)$

$F_4(ABCD) = \Sigma m(4-7,9-15)$

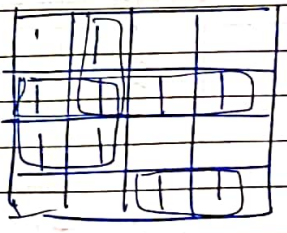


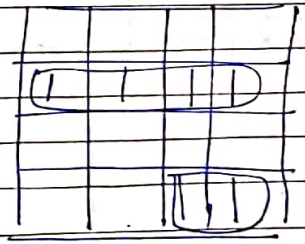
$F_1 = B\bar{C}\bar{D} + AB + A\bar{C}$

$F_2 = \bar{A}\bar{B}C + \bar{A}B + B\bar{C} + AB\bar{C}$

$F_3 = \bar{A}B + A\bar{B}C$ → $F_3 + \bar{A}\bar{B}C + B\bar{C}$

$F_4 = B + AD + AC$

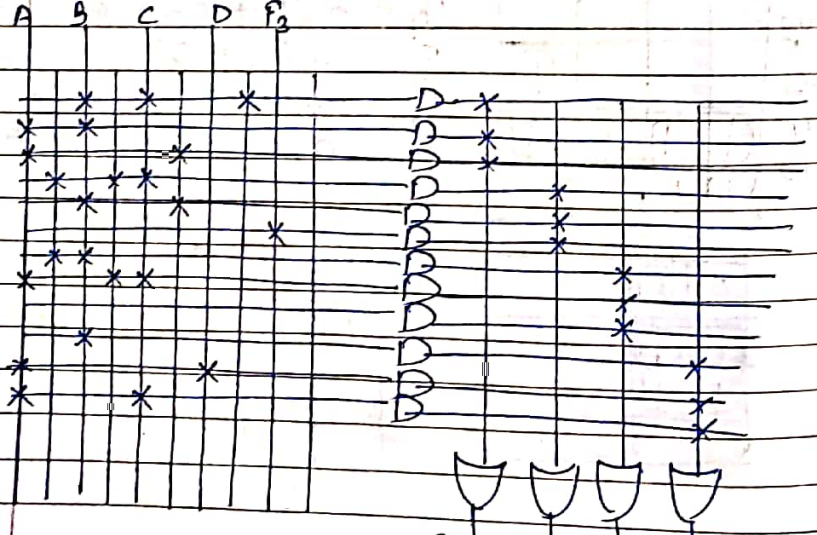




	00	01	11	10
00	·	·	·	
01	1	1	1	1
11	1	1	1	1
10	·	1	1	1

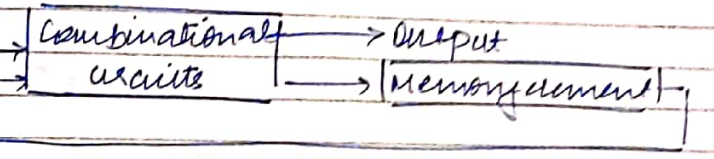
$F_4 = B + AD + \bar{A}C$

Product Term	I/P A	B	C	D	O/P F_3
1	–	1	0	–	
2	1	1	–	–	$F_1 = B\bar{C}\bar{D} + AB + A\bar{C}$
3	1	–	0	–	
4	–	–	–	1	
5	0	0	1	–	$F_2 = \bar{A}\bar{B}C + \bar{A}B + B\bar{C} + F_3$
6	–	1	0	–	
7	0	1	–	–	
8	1	0	1	–	$F_3 = \bar{A}B + A\bar{B}C$
9	–	–	–	–	
10	–	1	–	–	
11	1	–	–	1	$F_4 = B + AD + \bar{A}C$
12	1	–	–	1	



Unit-3

Synchronous Sequential logic

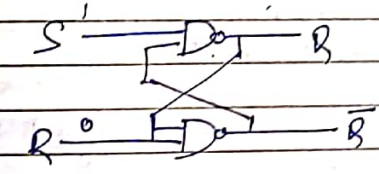


Combinational → current output depends upon current input.

Sequential → depends upon previous output.

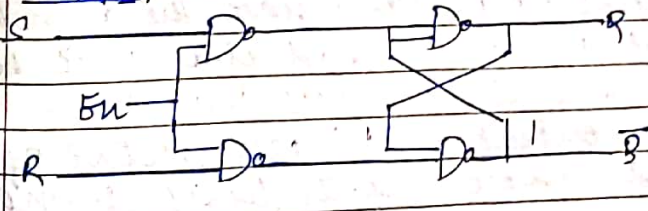
	Latch	Flipflop
(1)	level triggered	E-triggered
(2)		

S-R Latch with NAND gate



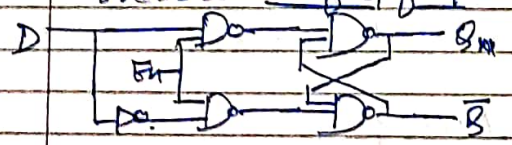
S	R	Q_{n+0}	Q_{n+1}	State
0	0	0	X	invalid
0	0	1	X	invalid
0	1	0	1	reset
0	1	1	1	reset
1	0	0	0	reset
1	0	1	0	reset
1	1	0	0	memory
1	1	1	1	memory (NC) no change

Flipflop



Clock	S	R	Q_n	Q_{n+1}	State
1	0	0	0	0	NO change
1	0	0	1	1	"
1	0	1	0	0	Reset
1	0	1	1	0	
1	1	0	0	1	Set
1	1	0	1	1	
1	1	1	0	x	Invalid
1	1	1	1	x	
0	x	x	0		
0	x	x	1		

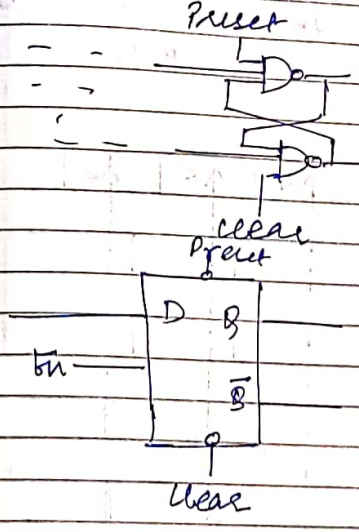
Enable D-flipflop



Enable	D	Q_n	Q_{n+1}	State
0	x	0	0	NC
0	x	1	1	(no change)
1	0	0	0	Reset
1	0	1	0	"
1	1	0	1	Set
1	1	1	1	"

Preset - in order to device the flipflop to an initial set or clear state the asynchronous input that is preset and clear will be used. Activating preset terminal makes output $Q = 1$. irrespective of D and enable inputs. (asynchronous) Similarly activating clear terminal makes

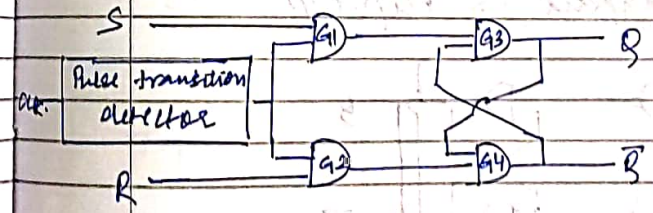
$Q = D$.



Activating = inputting 0.
[preset = 0 ⇒ Q_n = 1
clear = 0 ⇒ Q_n = 0.
whatever be the values of clock & synchronous inputs.

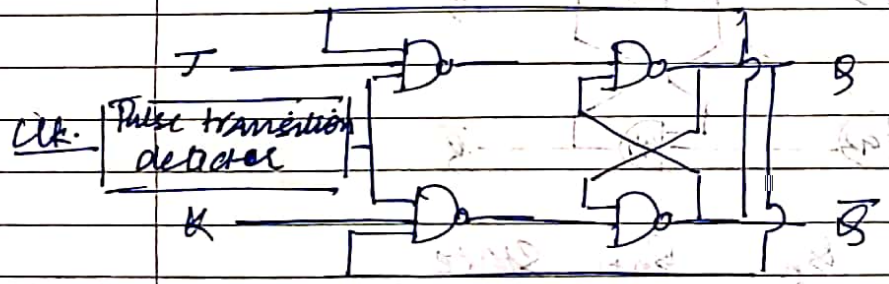
⇒ Logic Symbol.

Edge-triggered SR flipflop



S	R	Q_n	Q_{n+1}	State
0	0	0	0	
0	0	1	1	NC
0	1	0	0	Reset
0	1	1	0	
1	0	0	1	Set
1	0	1	1	
1	1	0	x	Indetermi-
1	1	1	x	nate

Edge Triggered JK Flipflop

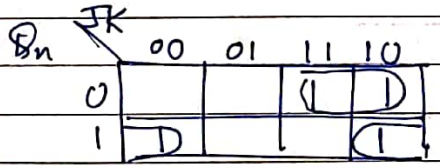


J	K	Q_n	Q_{n+1}	State
0	0	0	0	NC
0	0	1	1	
0	1	0	0	Reset
0	1	1	0	
1	0	0	1	Set
1	0	1	1	
1	1	0	1	Toggle
1	1	1	0	

Excitation Table and Characteristic Equation Of flip flop

D flip flop A table which lists the present state, next state and the excitation of a flip flop is called its excitation table.

Present State	I/P	Next State
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



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

Characteristic Table Eqn.

D flip flop –

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

$Q_{n+1} = D$

Drawbacks → width of pulse cannot be controlled.

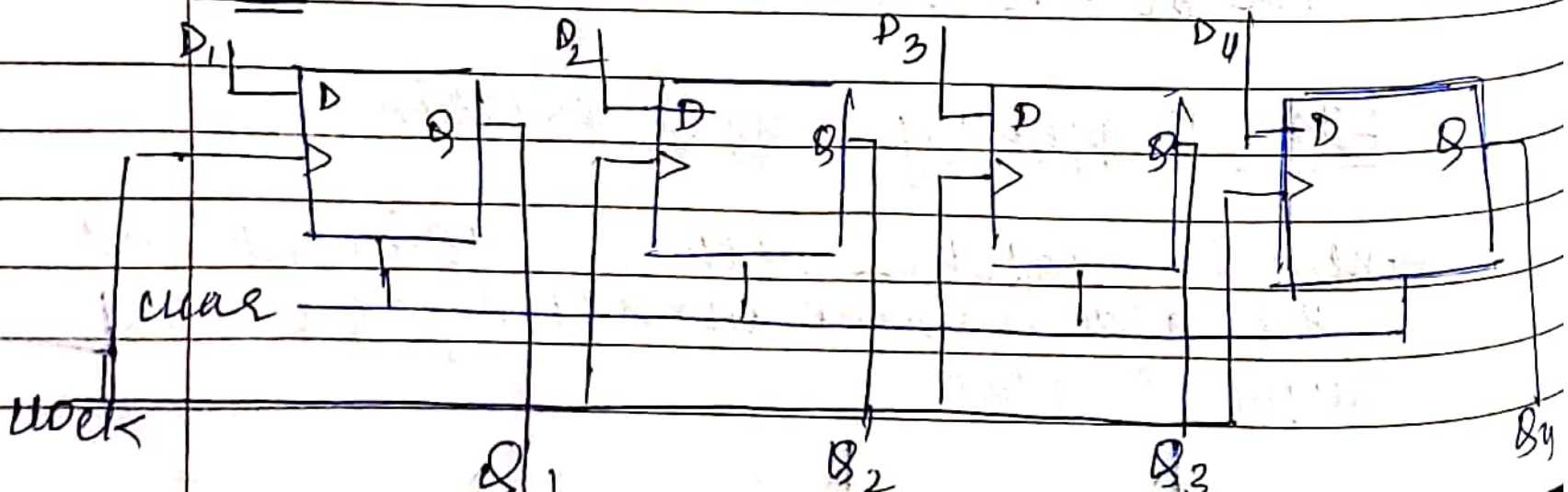
Registers

→ group of flipflops with common clock & input used for storing binary data.

On basis of config-

1. Parallel In Parallel Out — PIPO
2. Serial in serial out — SISO ⇒ shift-register
3. & PISO
4. SI PO — n bit register - n flipflops. can store n bit word

PIPO →



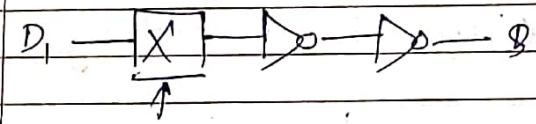
CMOS Logic (Dynamic)

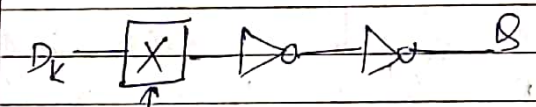
→ info. is not stored in flipflops, but as charge on tiny capacitors.

0 → 1 charging capacitor → data will be stored for ms.
1 → 0 discharging capacitor

for dynamic gates
(not, and, or, etc.)

hence they need to be refreshed continuously.

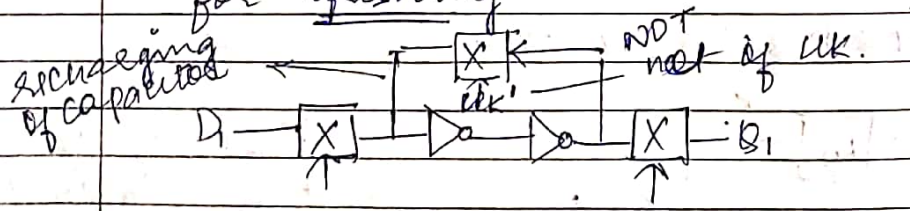




X — CMOS transmission gate.

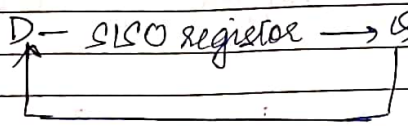
clock/load.
= 1 → 0.

for refreshing



refresh is carried out in every clock period when clk = 0

Ring counter

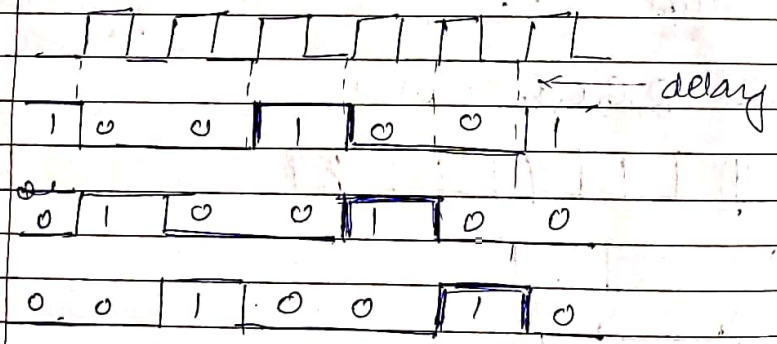


for a k bit ring counter, contents get repeated after k clocks (modulo k)

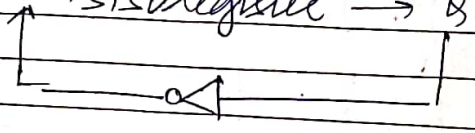
4 | 1000
0100
0010
0001
1000

modulo-4

3 bit ring counter $Q_1Q_2Q_3 = 100$



Johnson counter



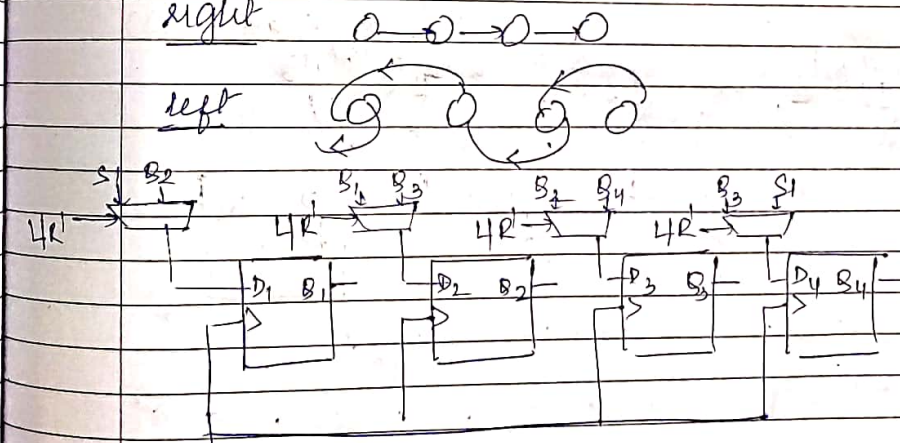
Repetition after 2k clocks. (mod-2k)

clocks	1	2	3	4	5	6	7	8	9
Q_1	0	1	1	1	1	0	0	0	0
Q_2	0	0	1	1	1	1	0	0	0
Q_3	0	0	0	1	1	1	1	0	0
Q_4	0	0	0	0	1	1	1	1	0

8 unique patterns

Bidirectional shift register

Mux'es are used to feed appropriate signals to the D inputs

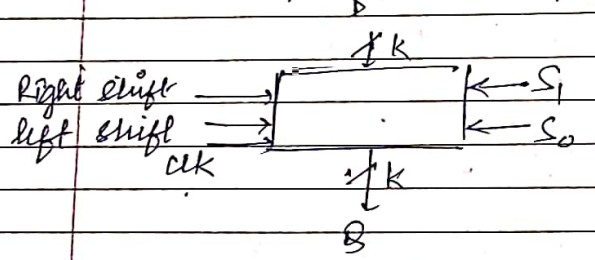


right shift $4R' = 0$

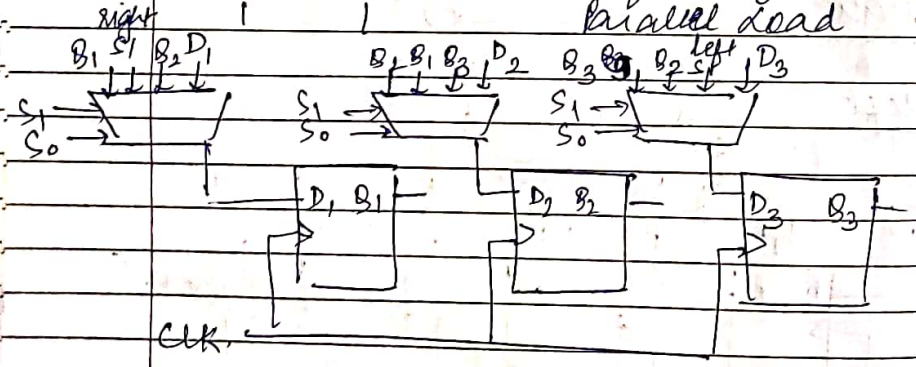
left shift $4R' = 1$

universal shift register

- bidirectional shift register + series/parallel input
- output : series/parallel



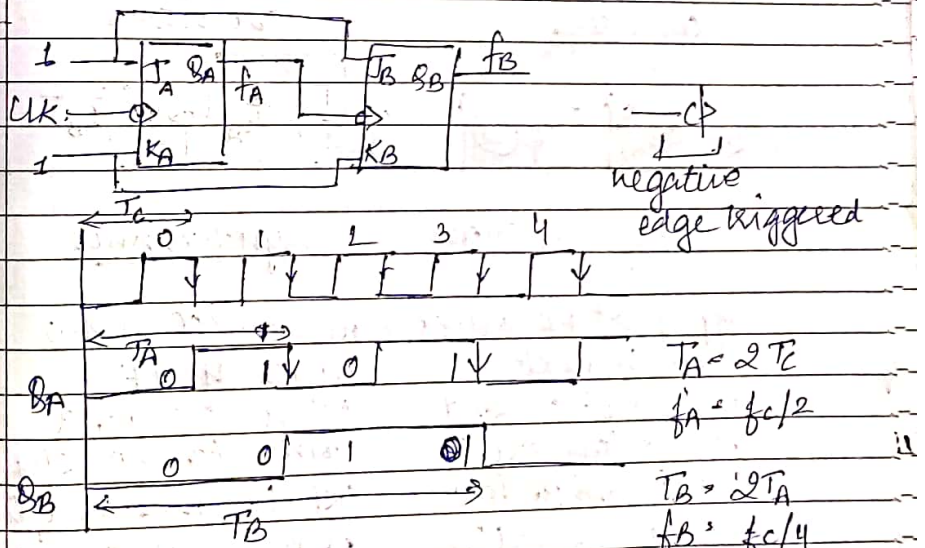
Control Input		Action
0	0	No change
0	1	Right shift
1	0	Left shift
1	1	Parallel load



Synchronous Sequential ckt	Asynchronous Sequential ckt.
- clocked FF acts as memory element	- unclocked FF/time delay element is used as memory element.
- slower	- faster
- status of memory element is affected only at active edge of clock	- status changes as soon as input is changed
- flipflops are used	- latches are used

Counters

- sequential ckt that counts (1 2 3 4 . .)



$T_A = 2T_C$

$f_A = f_C/2$

$T_B = 2T_A$

$f_B = f_C/4$

n no. of FF ⇒ J,K = 1, -ve edge triggered

- frequency is divided by 2^n

CLK	Q_B	Q_A
0	0	0
1	0	1
2	1	0
3	1	1

repeats after 3

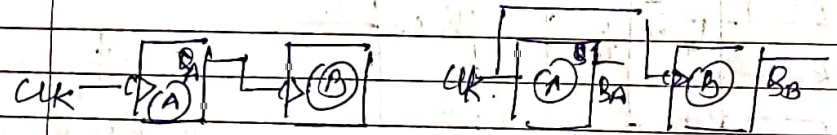
- we have to check how many clock pulses have elapsed after 0.

0 to 15 - $2^4 = 16$ · 4 flipflops

Types of counters

Ripple/asynchronous counters

synchronous counters

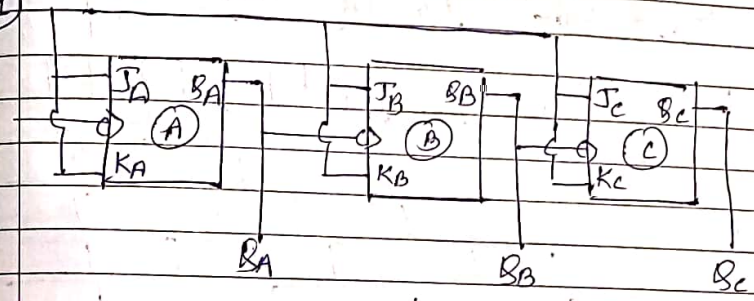


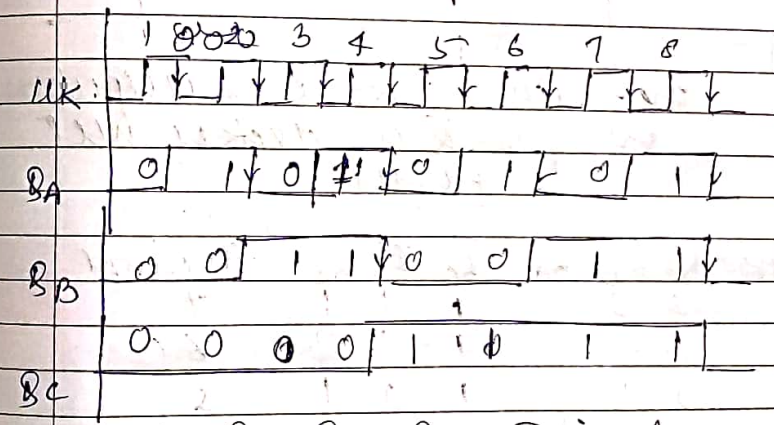
Asynchronous	Synchronous
o/p of 1st FF drives clock of next FF are not clocked simultaneously	no connection b/w clocks of FF FF are clocked simultaneously
ckt. is simple for more no. of states	ckt. becomes complicated as no. of states increases
speed is slow	speed is high

Refresh -- (1) up counters 0-1-2-3-...
(2) down counters 7-6-5-...-0
(3) up/down "

3 bit async. up counters

(1)



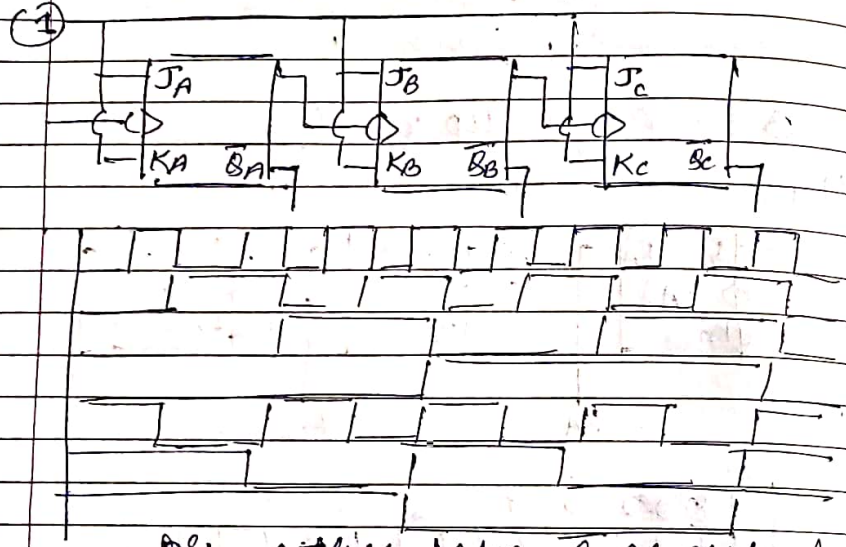


CLOCK	Q_C	Q_B	Q_A	Decimal
1	0	0	0	0
2	0	0	1	1
3	0	1	0	2
4	0	1	1	3
5	1	0	0	4
6	1	0	1	5
7	1	1	0	6
8	1	1	1	7

$2^3 = 8$ states

max. count $= 2^n - 1 = 2^3 - 1 = \boxed{7}$

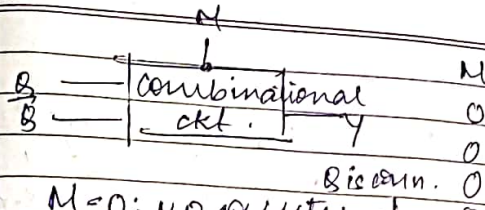
3 bit down counter (async)



DS - either take Q as output OR put $\bar{Q}$ as clock of next flipflop.

clk.	Q_C	Q_B	Q_A	$\bar{Q}_C$	$\bar{Q}_B$	$\bar{Q}_A$	Decimal
	0	0	0	↑	↑	↑	7
	0	0	1	↑	↑	0	6
	0	1	0	↑	0	↑	5
	0	1	1	↑	0	0	4
	1	0	0	0	↑	↑	3
	1	0	1	0	↑	0	2
	1	1	0	0	0	↑	1
	1	1	1	0	0	0	0

up/down counter



B is comm.

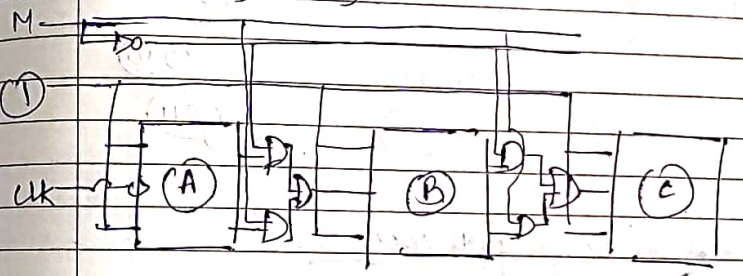
M = 0: up counter ↓

M = 1: down counter ↓

$\bar{Q}$ is connected

M	Q	$\bar{Q}$	Y
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

$Y = M\bar{Q} + M\bar{Q}$

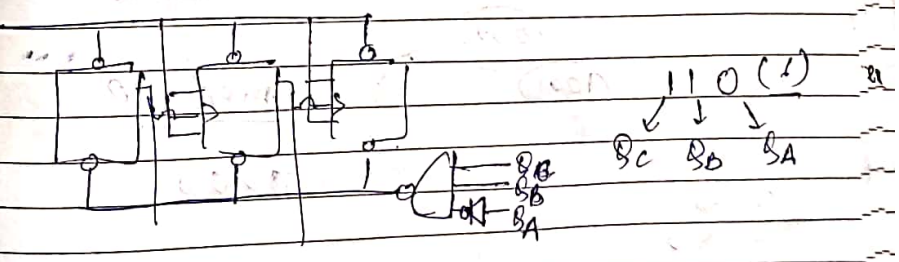


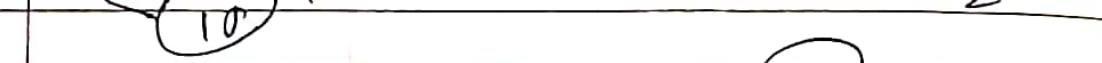
2 bit ripple counter = mod-4.

n " " n = mod-2^n : state

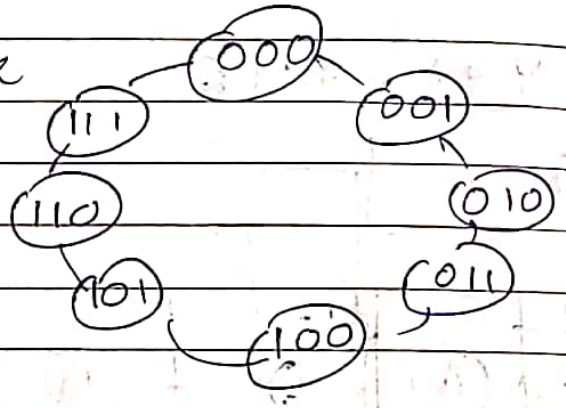
mod-6 using mod-8 counter

- reset when counter reaches 5





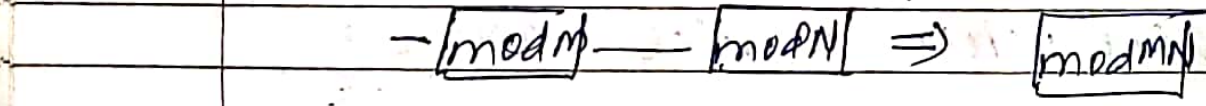
3 bit up counter

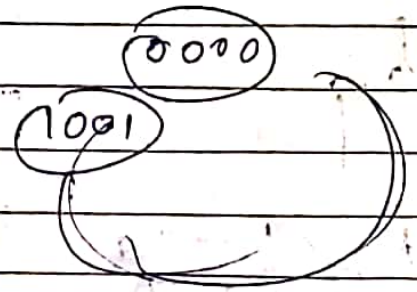


Decade (BCD) ripple counter

-ve edge :- $\overline{Q}$: clock : UP
$\overline{B}$: clock : down.

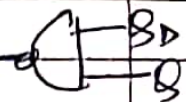
+ve edge :- Q : clock : down
$\overline{Q}$: clock : UP.



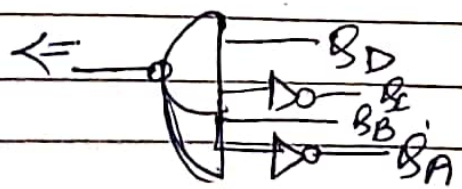


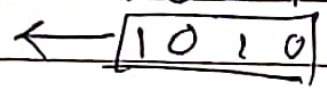
mod - 10

0000



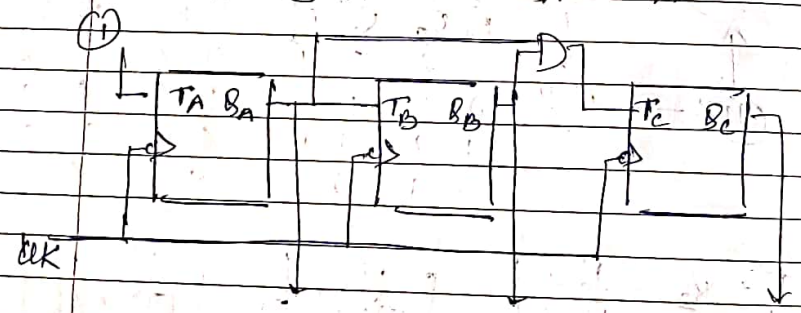
1001





P.S.			N.S.					
Q_C	Q_B	Q_A	Q_C^*	Q_B^*	Q_A^*	T_C	T_B	T_A
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	0	0	0	0	1
1	1	1	0	0	0	1	1	1

$T_C = Q_B Q_A \quad T_B = Q_A \quad T_A = 1$



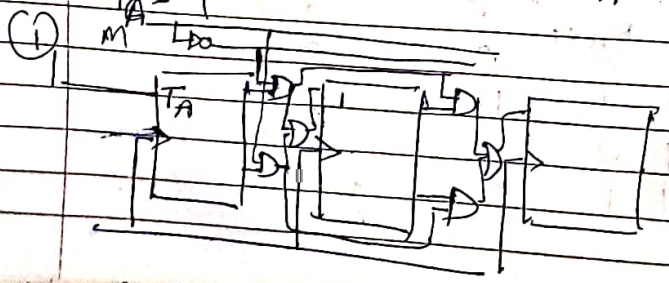
3 bit up/down sync. counter.

M = 0 :- UP M = 1 : down

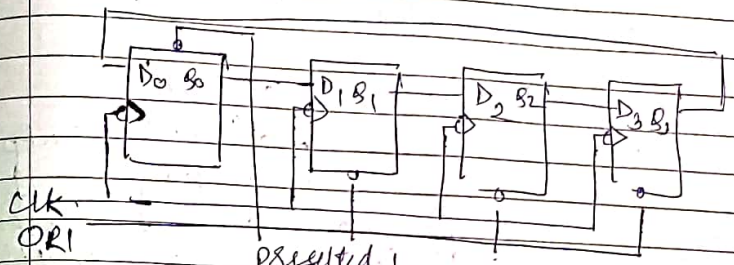
$T_C = \bar{M} Q_B Q_A + M \bar{Q}_B \bar{Q}_A$

$T_B = \bar{M} Q_A + M \bar{Q}_A = M \oplus Q_A$

$T_A = 1$



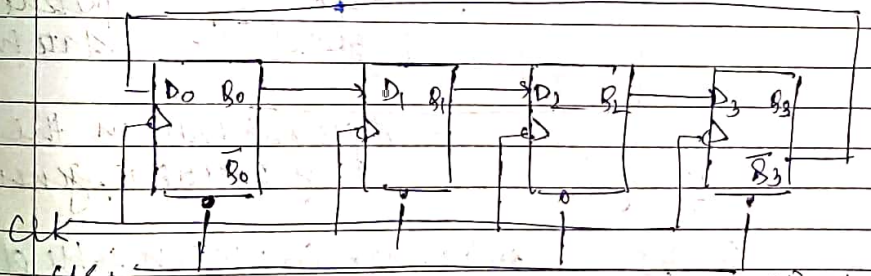
Ring counter



preset

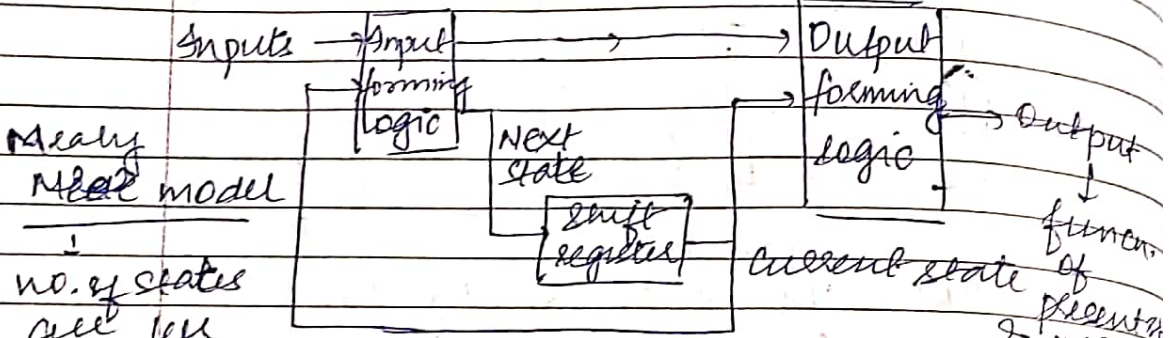
ORI	CLK	Q_0	Q_1	Q_2	Q_3
	↑	1	0	0	0
1	↓	0	1	0	0
1	↓	0	0	1	0
1	↓	0	0	0	1
1	↓	1	0	0	0

Johnson counter



no. of states = 2 × no. of flipflops

CLR	CLK	Q_3	Q_2	Q_1	Q_0
1	↑	0	0	0	0
0	↓	1	0	0	0
0	↓	1	1	0	0
0	↓	1	1	1	0
0	↓	1	1	1	1
0	↓	0	1	1	1
0	↓	0	0	1	1
0	↓	0	0	0	1
0	↓	0	0	0	0

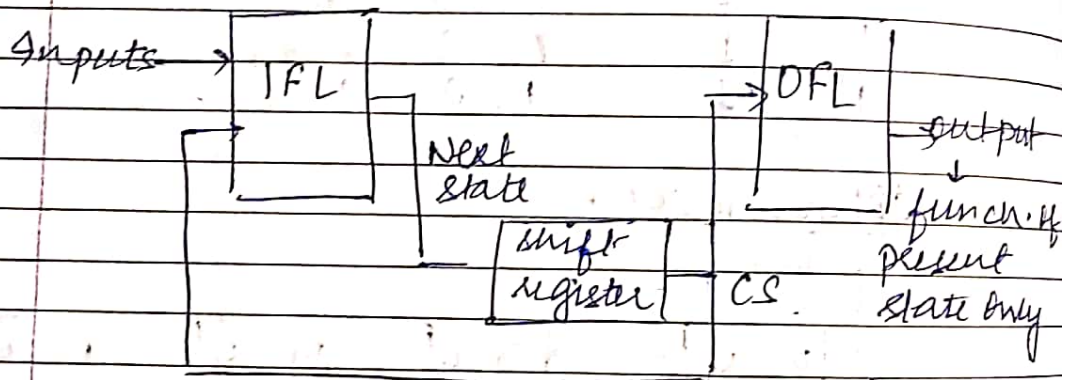


Mealy model

↓

no. of states are less

Moore model

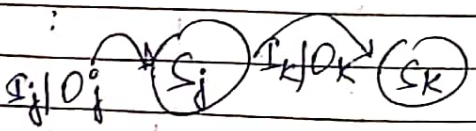


- different only in the way output is generated

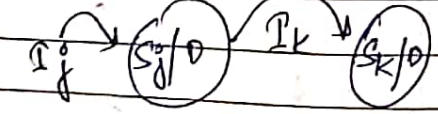
Analysis

3 steps : 1. logic eqn (excitation table etc.)
2. State table - CS, NS.
3. State diagram - pictorial representation of logic ckt.

Mealy model :



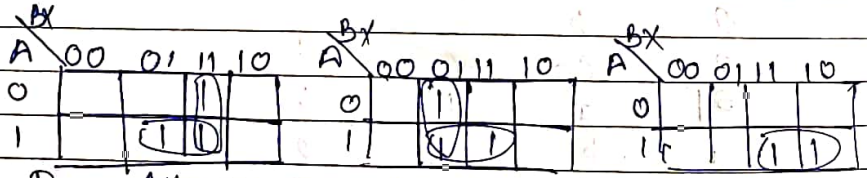
Moore model :



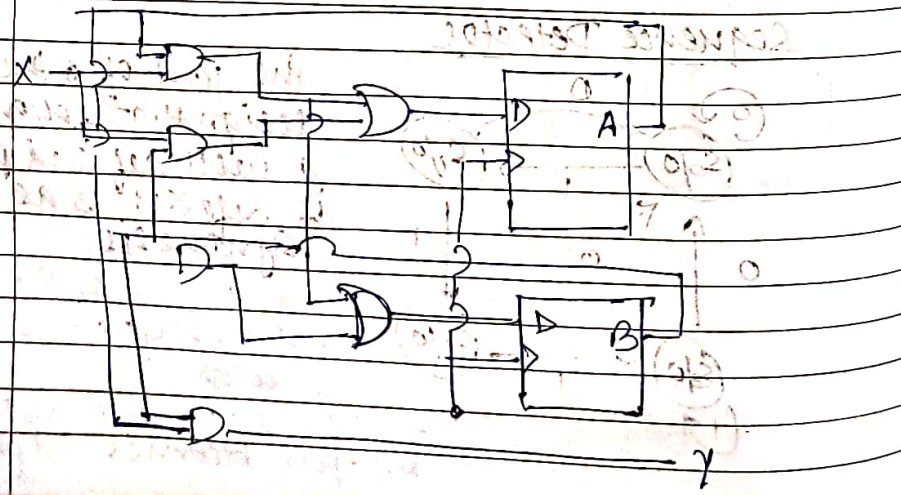
→ detects sequence of 3 1's (3 or more) in a string of bits

Synthesis using D flipflops

	P.S. A	P.S. B	Input X	N.S. A	N.S. B	Output Y
0	0	0	0	0	0	0
1	0	0	1	0	1	0
2	0	1	0	0	0	0
3	0	1	1	1	0	0
4	1	0	0	0	0	0
5	1	0	1	1	1	0
6	1	1	0	0	0	1
7	1	1	1	1	1	1

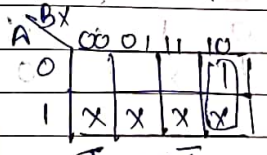


$D_A = AX + BX$ $D_B = AX + \bar{B}X$ $Y = AB$

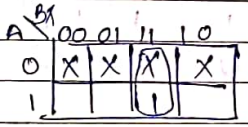


Synthesis using JK Flipflop

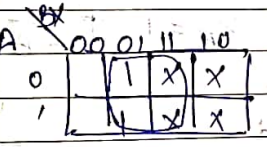
P.S. A	P.S. B	Input X	N.S. A	N.S. B	J_A	K_A	J_B	K_B
0	0	0	0	0	0	X	0	X
0	0	1	0	1	0	X	1	X
0	1	0	1	0	1	X	X	1
0	1	1	0	1	0	X	X	0
1	0	0	1	0	X	0	0	X
1	0	1	1	1	X	0	1	X
1	1	0	1	1	X	0	X	0
1	1	1	0	0	X	1	X	1



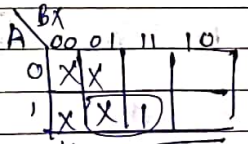
$J_A = B\bar{X}$



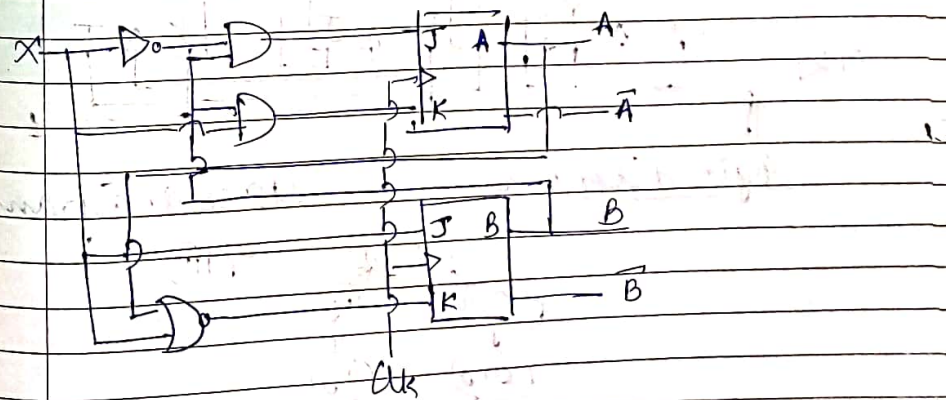
$K_A = BX$



$J_B = X$



$K_B = A \odot X$



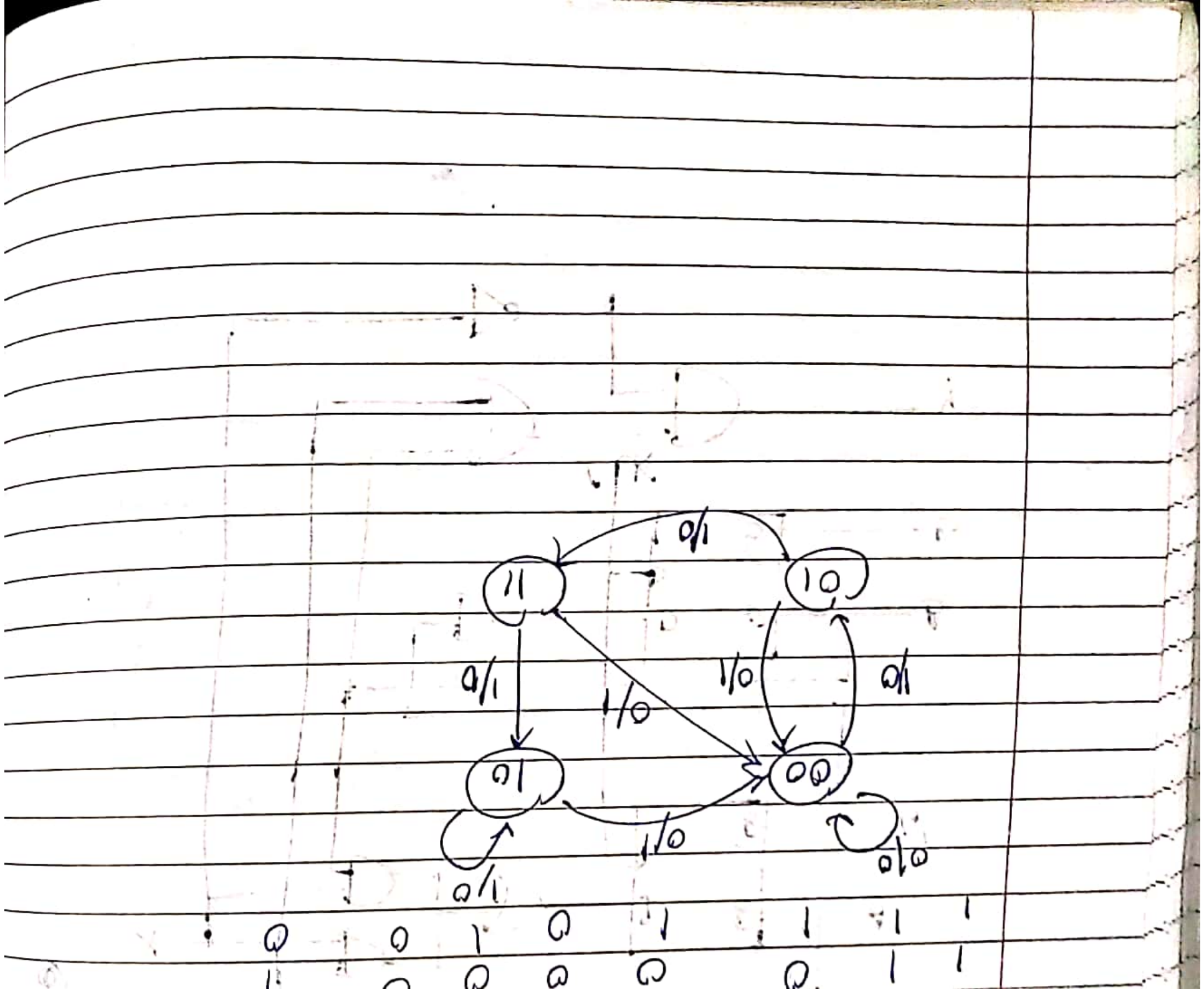
Page No.
Date

Counter with non binary sequence

A counter with n flipflops may have a binary sequence of less than 2^n states

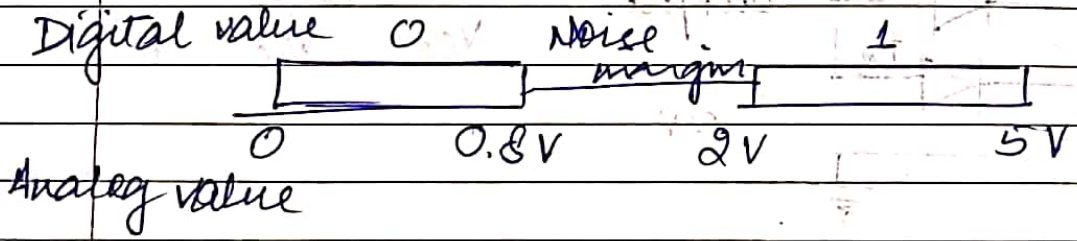
	P.S.			N.S.			Flipflop					
	A	B	C	A	B	C	J_A	K_A	J_B	K_B	J_C	K_C
0	0	0	0	0	0	1	0	X	0	X	1	X
1	0	0	1	0	1	0	0	X	1	X	X	1
2	0	1	0	0	1	1	1	X	X	1	0	X
4	1	0	0	0	1	0	X	0	0	X	1	X
5	1	0	1	1	1	0	X	0	1	X	X	1
6	1	1	0	0	0	0	X	1	X	1	0	X

3 & 7 → don't care

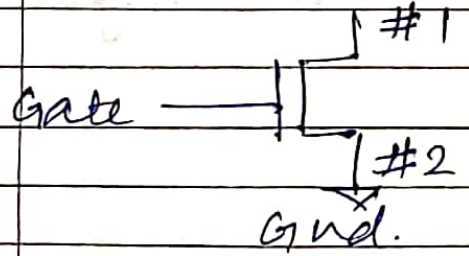


is treated as logic 0, another range is treated as logic 1

For reliable opn. there must be considerable gap b/w the ranges.
↳ noise margin



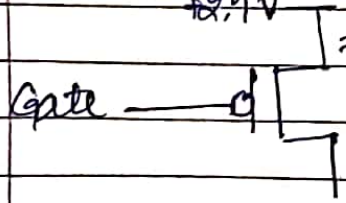
n type MOS transistor



+ve voltage :- switch closed

-ve voltage :- switch open

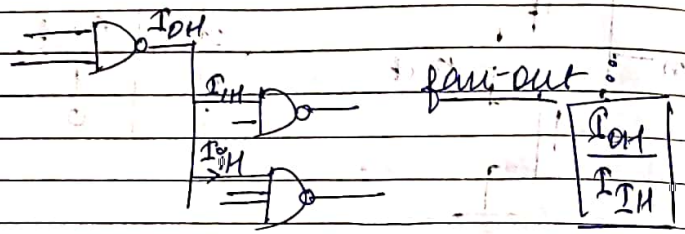
p type MOS transistor



+ve voltage :- switch open

-ve voltage :- switch closed

- no. of inputs that can be connected to the output of a gate



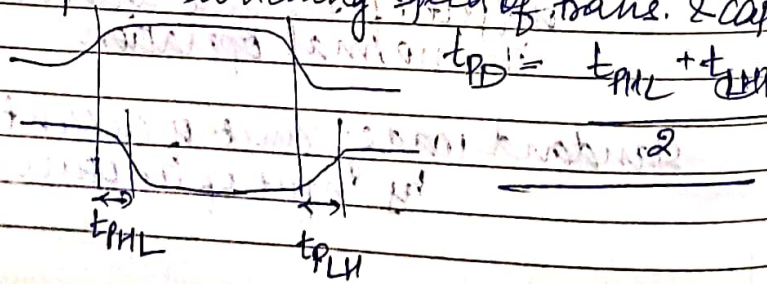
fan-out : $\frac{I_{OH}}{I_{IH}}$

Power dissipation - (mW)
↳ amt of power needed by the gate

$P = V_{CC} \times I_{CC}$
V_{CC} → voltage supply
I_{CC} → current drawn from gate.

$I_{CC(avg.)} = \frac{I_{CC \text{ at high voltage}} + I_{CC \text{ low voltage}}}{2}$

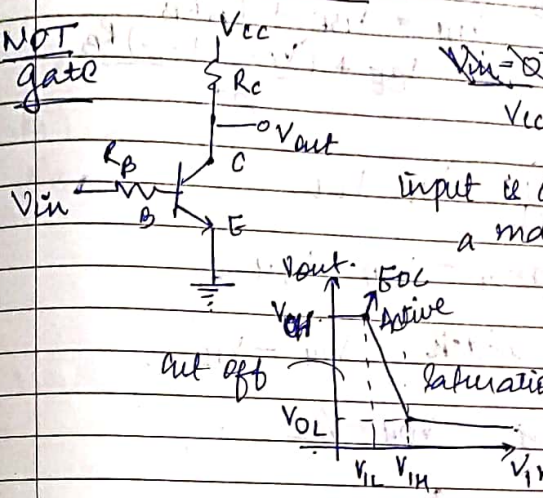
Propagation delay -
avg transition delay for signal to propagate from input to output when signal changes its value
↳ due to finite switching speed of trans. & capacitance



$t_{PD} = \frac{t_{PHL} + t_{PLH}}{2}$

Transistor as switch

NOT gate



$V_{in} = 0$:
$V_{CC} - I_C R_C = V_{out} = V_{CC}$
input is considered low to a max. point V_{IL}

$V_{in} = 1 \Rightarrow V_{IH} \Leftarrow$ min. voltage treated as logic high
$V_{out} = V_{OL}$

Cut off Base current $I_B = 0$ till $V_B - V_{in} < V_{IL}$
$V_{out} = V_{OH}$ over entire range.

Break point / edge of cutoff (EOC), point where $V_{in} = V_{IL}$, $V_{out} = V_{OH}$

Active $I_B = \frac{V_B - V_{out}}{R_B}$ $I_B = \frac{V_{in} - V_{BE}}{R_B}$

$I_C = \beta I_B$

$V_{out} = V_{CC} - \frac{\beta R_C (V_{in} - V_{BE})}{R_B}$

Saturation region -
Edge of saturation - $V_{in} = V_{IH}$, $V_{out} = V_{OL}$

$I_C = \frac{V_{CC} - V_{CE}}{R_C}$

$V_{IH} = V_{BE} + I_B R_B$

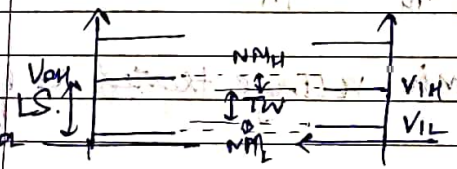
$= V_{BE} + \frac{(V_{CC} - V_{CE}) R_B}{\beta R_C}$

Noise margin)$_L$ $= V_{IL} - V_{OL}$

Noise margin)$_H$ $= V_{OH} - V_{IH}$

Transition width $= V_{IH} - V_{IL}$

Logic swing $= V_{OH} - V_{OL}$



Fan-out $= I_{C(drives)} / I_{B(load)}$

Static Power dissipation
- when transistor is either on/off
- depends on current drawn

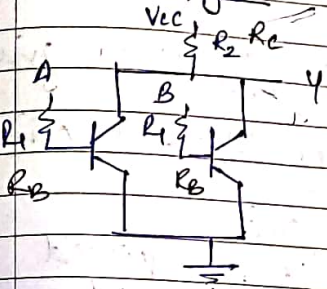
$E = P_0 \times t_{PD}$

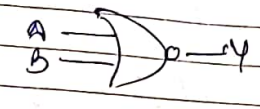
figure of merit — the lower, the better

- significant in switching of BJT

Dynamic Power dissipation
- depends on switching speed
- when transistor switches
- significant in switching of CMOS trans.

RTL NOR gate





For 3 input, another input will come in parallel between A & B.

more no. of inputs - more delay.

Fan-in: max. no. of inputs a gate can handle.

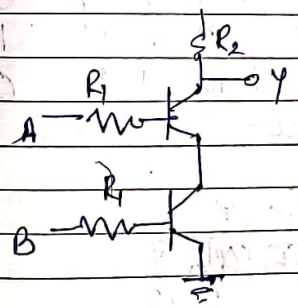
A	B	Y
0	0	1
0	1	0
1	0	0
1	1	0

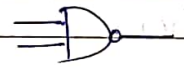
A=0=B ⇒ transistor is off

in all other cases, at least 1 trans. works ⇒ earthed.

= 0

RTL NAND gate





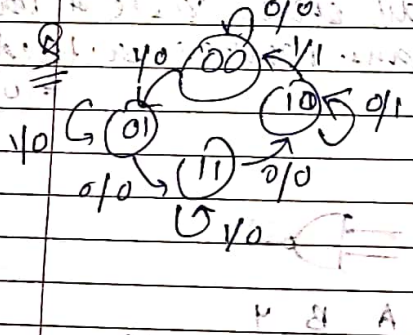
A	B	Y
0	0	1
0	1	1
1	0	1
1	1	0

If any one is off, ckt off ⇒ conduction doesn't occur

If both are on ⇒ earthed

Clocked Sequential Circuits

Steps – (1) State/timing diagram is given
(2) obtain state table
(3) reduce states by state reduction method.
(4) do state assignment
(5) determine no. of flipflops required
(6) decide type of flipflop required
(7) derive circuit excitation eqn.
(8) obtain expression for output & input
(9) implement circuit



P.S.		N.S.				Output	
		x=0		x=1			
B_A	B_B	B_A^+	B_B^+	B_A^+	B_B^+	x=0	x=1
0	0	0	0	0	1	0	0
0	1	1	1	0	1	0	0
1	0	1	0	0	0	1	1
1	1	0	0	0	1	0	0

CKt. Excitation Table –

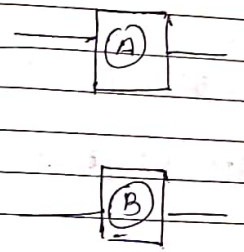
P.S.			N.S.		FF. I/P		
B_A	B_B	x	B_A^+	B_B^+	T_A	T_B	y
0	0	0	0	0	0	0	0
0	0	1	0	1	0	1	0
0	1	0	1	1	1	0	0
0	1	1	0	1	0	0	0
1	0	0	1	0	0	0	1
1	0	1	0	0	1	0	1
1	1	0	1	0	0	1	0
1	1	1	1	1	0	0	0

Using T flipflop

$T_A = B_A B_B \bar{x} + B_A \bar{B}_B x$

$T_B = \bar{B}_A B_B x + B_A B_B \bar{x}$

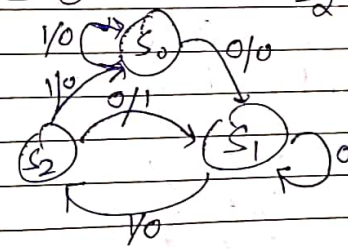
$y = B_A B_B \bar{x} + B_A \bar{B}_B x = B_A \bar{B}_B$



Sequence detector – 010

State diagram

S_0 – reset (powerup)
S_1 – 0 S_2 – 01



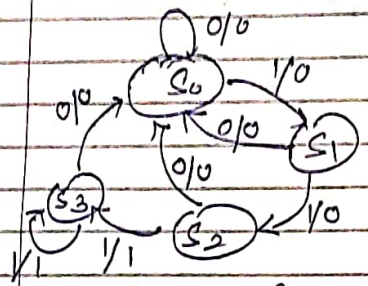
$S_0 = 00$
$S_1 = 01$
$S_2 = 10$
binary assignment

3 or more consecutive 1's

eg – x = 0011101111 10
y = 0000100011 10 (overlapping)
y = 0000010001 00 (no overlapping)

S_0 = Reset S_1 = 1 S_2 = 1/1

	Has	Awaits
S_0 = Reset	--	1 1 1 . . .
S_1 = 1	1	1 1 . . .
S_2 = 11	11	1 - . . .
S_3 = 111	111	- - - -



$S_0 = 00$
$S_1 = 01$
$S_2 = 10$
$S_3 = 11$

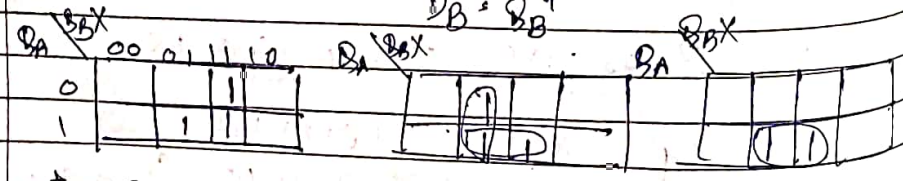
State table

P.S. Q_A	Q_B	x	Q_A^+	Q_B^+	y
0	0	0	0	0	0
0	0	1	0	1	0
0	1	0	0	0	0
0	1	1	1	0	0
1	0	0	0	0	0
1	0	1	1	1	1
1	1	0	0	0	0
1	1	1	1	1	1

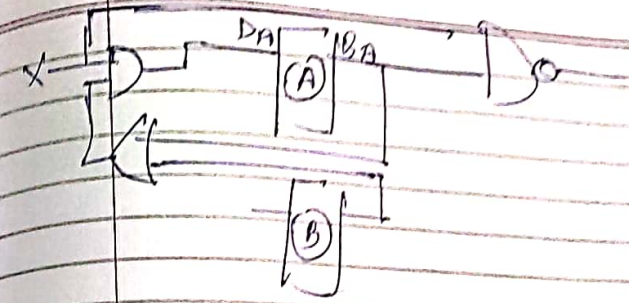
D flipflop - N.S. D

$D_A = Q_A^+$

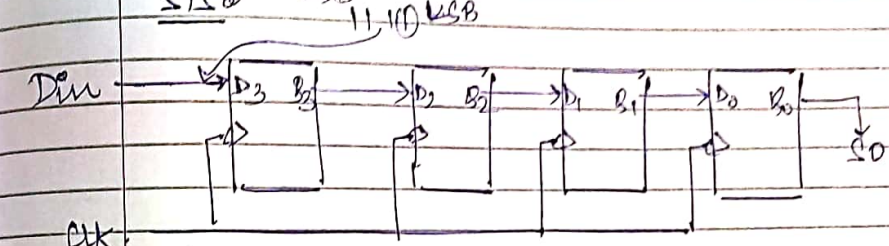
$D_B = Q_B^+$



$Q_A^+ = Q_B X + Q_A X$ $Q_B^+ = \overline{Q_B} X + Q_A X$ $Y = Q_A X$



Shift register



Clk.	Q_3	Q_2	Q_1	Q_0
	0	0	0	0
	1	0	0	0
	1	1	0	0
	1	1	1	0
	1	1	1	1