

DIGITAL SYSTEM DESIGN

(CS2001)

Name : Harith Yerragolam

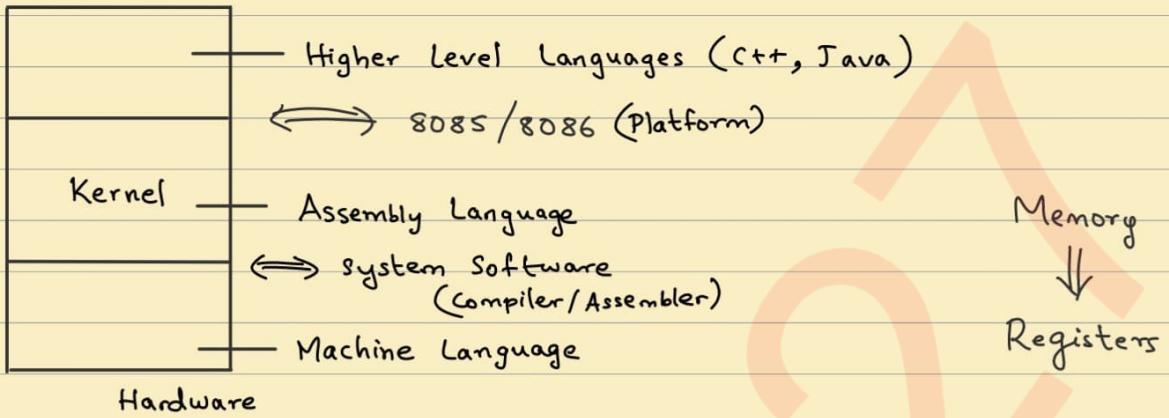
Roll No. : CS23I1027

Batch : B2

Instructor : Dr. Raghavan Sir

21/8

Application / Software



$$(142)_6 \Rightarrow 1 \times 6^2 + 4 \times 6 + 2 \times 1 \\ = \underline{\underline{62}}$$

Architecture : Van Neuman
TOC : Alan Turing
↳ Decidability, Computability

$$(43)_{10} \Rightarrow (223)_4$$

$$(15)_{10} \Rightarrow (F)_{16}$$

$$(1000)_8 \Rightarrow (512)_{10}$$

$$(1000)_2 \Rightarrow (64)_{10}$$

4	43
4	10
4	2
0	2

$$(112)_{10} = (1110000)_2$$

2	112	
2	56	0
2	28	0
2	14	0
2	7	0
2	3	1
2	1	1

27/8

$$(58)_{16} = (\)_8 = (\)_4$$

$$(131)_4 = (\)_8 = (\)_{16}$$

$$(42)_{16} = (\)_2$$

↙
₂ ④

$$\begin{aligned} 4 &\equiv 0100 \\ 2 &\equiv 0010 \end{aligned}$$

$$(42)_{16} = (01000010)_2$$

$$(31)_4 = (1101)_2$$

↙
₂²

$$(42)_8 = (100010)_2$$

\swarrow_2^3

$$(101101)_2 \equiv (\)_8 \longrightarrow (55)_8$$

\swarrow_2^3 $(101)(01)$
 $\downarrow_5 \downarrow_5$

$$(101101)_2 \equiv (\)_{16}$$

$$\equiv (2D)_{16}$$

$0010\mid 1101$
 $\downarrow_2 \downarrow_D$

$$④) (101101110)_2$$

$$101101110 \equiv (11232)_4$$

$$101101110 \equiv (556)_8$$

$$000101101110 \equiv (16E)_{16}$$

$$\rightarrow (58)_{16} = (\)_2$$

$\underbrace{01011000}_{\text{ }} \swarrow$

$$(58)_{16} \longrightarrow (\)_8$$

$\downarrow_{()_2} \quad \uparrow$
 $0\mid 011000$
 $(1\ 3\ 0)_8$

Test:
 $(58)_{16}$

$$\begin{array}{r} 4 | 88 \\ 4 | 22 \quad 0 \\ 4 | 5 \quad 2 \\ 4 | 1 \quad 1 \\ 0 \quad 1 \end{array}$$

$$(1120) \nearrow$$

$$101011000$$

$$(1\ 1\ 2\ 0)_4$$

$$16 \times 5 + 8$$

$$= \underline{\underline{(88)}}_{16}$$

$$\begin{array}{r} 1120 \\ \hline 5 \quad 8 \end{array}$$

number + 9s complement (number) = 99

$$(39)_{10} \longrightarrow (60)_9$$

$$\begin{array}{r} 99 \\ - 39 \\ \hline 60 \end{array}$$

$$60 + 1 = (61)_9$$

↪ 9s complement + 1 \Rightarrow 10s complement

$$\begin{array}{r} 42 \\ 39 \\ \hline 81 \end{array} \quad \begin{array}{r} 3 \\ 42 \\ - 39 \\ \hline 03 \end{array} \quad \equiv \quad \begin{array}{r} + 60 \\ \hline 102 \\ \downarrow +1 \\ 3 \end{array}$$

\rightarrow Binary Addition / Subtraction:

$$\begin{array}{r} 1011 \\ 0101 \\ \hline 10000 \end{array} \Rightarrow (0101)^c = 1010 \quad \begin{array}{l} \text{↪ } 1^s \text{ complement} \\ \text{↪ } 1011 \\ \text{↪ } 2^s \text{ complement} \end{array}$$

$(1)^c = 0$
 $(0)^c = 1$

$$(1+5=16)$$

$$11-5=6$$

$$\begin{array}{r} 1011 \\ 1011 \\ \hline 10110 \end{array} \quad \begin{array}{l} \text{↪ Ignore 1} \\ 0110 \rightarrow \underline{\underline{6}} \end{array}$$

$$11-5=6$$

$$\textcircled{14} \quad 1110$$

$$\textcircled{9} \quad 1001 \longrightarrow 0110$$

$$\begin{array}{r} & 1 \\ 0111 & \hline \end{array}$$

$$\begin{array}{r} 11 \\ 1110 \\ \hline 0111 \\ \underbrace{0101}_{\textcircled{5}} \end{array}$$

Carry = 1 \Rightarrow Ignore

Carry = 0 \Rightarrow Negative

$$\begin{array}{r} 1001 \\ - 14 \\ \hline 1011 \end{array} \longrightarrow 0100 \quad \begin{array}{l} \text{↪ } 1^s \text{ complement} \\ \text{↪ } 0101 \end{array}$$

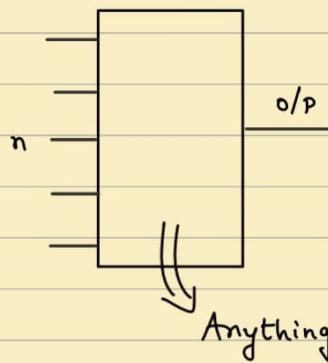
$$\max \rightarrow 2^4 = 16$$

$$11-16=-5$$

$$\textcircled{-5}$$

$\hookrightarrow 2^s \text{ complement}$

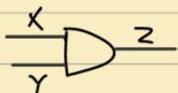
→ Gates:



3 Fundamental Gates :

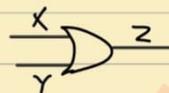
┌── AND
 └── OR
 NOT

AND



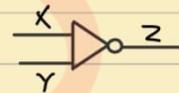
X	Y	Z
0	0	0
1	0	0
2	1	0
3	1	1

OR



X	Y	Z
0	0	0
1	0	1
2	1	0
3	1	1

NOT



X	Z
0	1
1	0

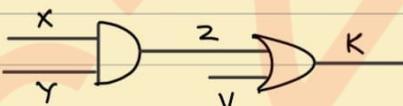
$$Z = \bar{X}$$

$$Z = X \cdot Y$$

$$\therefore Z = XY$$

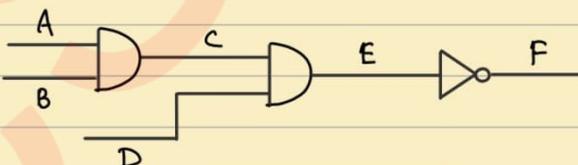
$$Z = X + Y$$

● Combination :



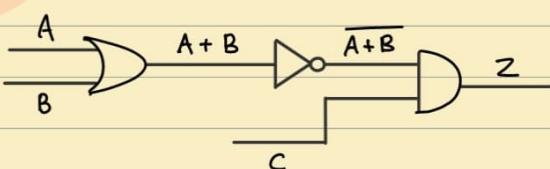
$$\begin{aligned} K &= Z + V \\ &= X \cdot Y + V \end{aligned}$$

$$\therefore K = XY + V$$



$$\begin{aligned} F &= \bar{E} \\ &= \overline{C \cdot D} \\ &= \overline{A \cdot B \cdot C} \end{aligned}$$

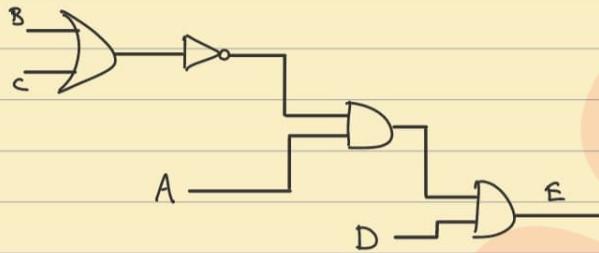
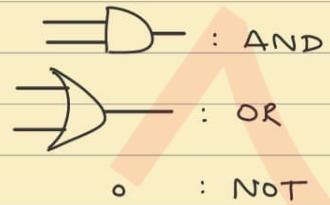
$$\therefore F = \overline{ABC}$$



$$\therefore Z = (\overline{A+B}) \cdot C$$

NOT \rightarrow more precedence, NOT > AND > OR

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



NOR : OR + NOT \rightarrow

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

NAND : AND + NOT \rightarrow

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

NOR & NAND : Universal Gates

AND, OR & NOT : Basic Gates

$$X \cdot X = X$$

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

\cdot : AND

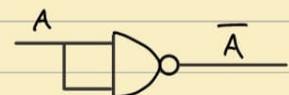
$+$: OR

$-$: NOT

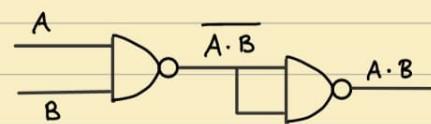
$$F(A, B) = AB + A\overline{B} = A$$

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

NOT using NAND:



AND using NANDs:



A	B	C	ABC	\bar{C}	$AB\bar{C}$	f
1	1	1	1	0	0	1
1	1	0	0	1	1	1
1	0	1	0	0	0	0
1	0	0	0	1	0	0
0	1	1	0	0	0	0
0	1	0	0	1	0	0
0	0	1	0	0	0	0
0	0	0	0	1	0	0

$$\begin{aligned}
 & AB\bar{C} + \bar{A}\bar{B}\cdot\bar{C} \\
 &= \bar{C}(AB + \bar{A}\bar{B}) \\
 &= \bar{C}
 \end{aligned}$$

George boole - Boolean Algebra

Boolean (Bool) : True or False

AND	OR	NOT
(i) •	+	-
(ii) $A \cdot I = A = I \cdot A$ $A \cdot 1 = A$ $A \cdot 0 = 0$	$A + I = A = I + A$ $A + 0 = A$ $A + I = A$	$A + \bar{A} = 1$ $A \cdot \bar{A} = 0$
$A \cdot A = A$ $A \cdot (B \cdot C) = (A \cdot B) \cdot C$ $AB = BA$ $A \cdot \bar{A} = 0$	$A + A = A$ $A + (B + C) = (A + B) + C$ $A + B = B + A$ $A + \bar{A} = 1$	

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

$$= A + A \cdot B$$

$$= A(1+B)$$

$$= A$$

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

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

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

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

$$= B + B$$

$$= B$$

$$\begin{aligned} \text{Q1)} \quad & A\bar{B} + AB + \bar{A}\bar{B} \\ & = \bar{B}(\bar{A}+A) + AB \\ & = \bar{B} + AB \end{aligned}$$

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

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

$$\begin{aligned} \text{Q2)} \quad & ABC + A\bar{B}C + A\bar{B}\bar{C} \\ & = A\bar{C}(B + \bar{B}) + A\bar{B}C \\ & = A\bar{C} + AC\bar{B} \\ & = A(C\bar{B} + \bar{C}) \end{aligned}$$

$$\begin{aligned} \text{Q3)} \quad & ABC + \bar{A}BC + AB\bar{C} \\ & = BC(A + \bar{A}) + AB\bar{C} \\ & = BC + AB\bar{C} \\ & = B(C + A\bar{C}) \end{aligned}$$

$$\begin{aligned} & A + \bar{A}B \\ & A + (1-A)B \\ & A + B - AB \\ & A(1-B) + B \\ & A\bar{B} + B \end{aligned}$$

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

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

DeMorgan's Law:

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

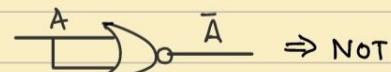
$$\overline{A \cdot B} = \bar{A} + \bar{B}$$

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

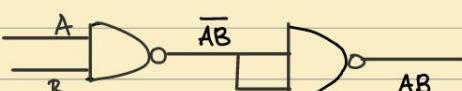
$$\Rightarrow \overline{A \cdot \bar{B}} = A + B$$

$$\begin{aligned} \text{Q4)} \quad & AB + \bar{A}B + A\bar{B} \\ & = A(B + \bar{B}) + \bar{A}B \\ & = A + \bar{A}B \\ & = A + B \end{aligned}$$

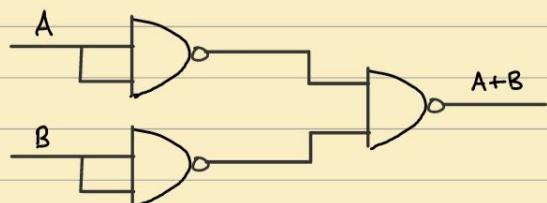
$$\begin{aligned} & ABC + AB + A\bar{C} + A\bar{B}\bar{C} \\ & = AB(C+1) + A\bar{C} + A\bar{B}\bar{C} \\ & = AB + A\bar{C} + A \cdot \bar{B}\bar{C} \\ & = A(B + \bar{C} + \bar{B}\bar{C}) \\ & = A(B + \bar{C}(1+B)) \\ & = A(B + \bar{C}) \end{aligned}$$

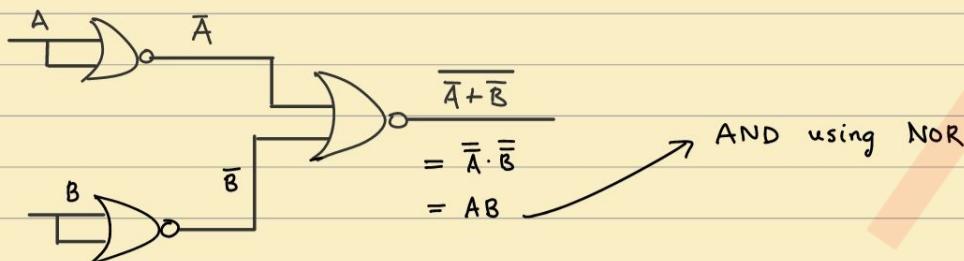


$$\overline{\overline{AB}} + \overline{\overline{A+B}} = \bar{A} + \bar{B} = A + B$$



OR using NANDs:





Q) $AB\bar{C} + A\bar{B}\bar{C} + \bar{A}\bar{B}C + ABC \longrightarrow \text{SOP (Canonical Form)}$

$$\begin{aligned} & AB\bar{C} + A\bar{B}\bar{C} + \bar{A}\bar{B}C + ABC + ABC + ABC \\ &= AB(C + \bar{C}) + AC(B + \bar{B}) + BC(A + \bar{A}) \\ &= AB + BC + CA \end{aligned}$$

Term - Standard Form (Not Unique)

→ Sum of Products (Unique) / Min. terms :

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

A	B	F
0	0	0
0	1	1
1	0	1
1	1	1

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

Look at Truth Table.
Wherever you find 1,
Write corresponding terms
in input variables.

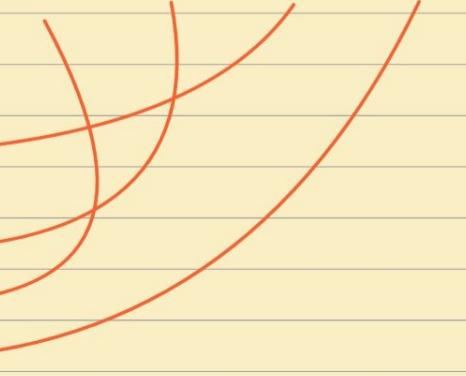
Ex. $ABC + A\bar{B}C + \bar{A}BC + ABC$

110 101 011 111

A	B	C	F
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1 ($\bar{A}BC$)
1	0	0	0
1	0	1	1 ($A\bar{B}C$)
1	1	0	1 (ABC)
1	1	1	1 (ABC)

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

110 101 011 111



Ex. $AB + \overline{A}B + A\overline{B} \Rightarrow F(A, B)$

A	B	F
0	0	0
0	1	1
1	0	1
1	1	1

$\Rightarrow A + B$

Ex. $\overline{AB} + A\overline{B}$

A	B	F
0	0	0
0	1	1
1	0	1
1	1	0

$$\begin{array}{|c|c|c|c|} \hline A & B & C & F \\ \hline 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 \\ \hline \end{array} \rightarrow \overline{A}\overline{B}C + \overline{A}B\overline{C} + \overline{A}BC + A\overline{B}\overline{C} + A\overline{B}C + A\overline{B}C + ABC$$

$$\begin{aligned} &= A(\overline{B}C + B\overline{C} + BC) \\ &= \overline{A}(\overline{B}C + B) \\ &= \overline{A}(\overline{B}C + B(1+C)) \\ &= \overline{A}(\overline{B}C + B + BC) \\ &= \overline{A}(B + C) \end{aligned}$$

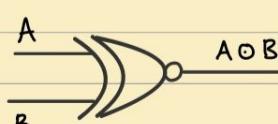
$$\begin{aligned} &\overline{AB} + \overline{AC} + A \\ &\Rightarrow \overline{AB} + \overline{AC} + A(1+B) \\ &\Rightarrow \overline{AB} + \overline{AC} + A + AB \\ &\Rightarrow B + \overline{AC} + A(1+C) \\ &\Rightarrow B + \overline{AC} + A + AC \\ &\Rightarrow A + B + C \end{aligned}$$



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

$[01, 10 \rightarrow 1]$

XOR



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

$[00, 11 \rightarrow 1]$

XNOR

$$\overline{A \oplus B} = A \circ B$$

$$\overline{A \odot B} = A \oplus B$$

Ex. $A \oplus \underbrace{B \oplus A}_{X}$

$$A \oplus X$$

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

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

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

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

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

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

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

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

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

$$= B$$

4/9

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

A	B	$A \oplus B$	$A \oplus B \oplus A$
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	1

$$A \oplus B \oplus A = B$$

- Application:

$$A \quad B \Rightarrow C = A \oplus B$$

2 GB 2 GB

2 GB

↳ Works as a Backup, i.e. if A is lost,

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

$$= A$$

$$(12.5)_{10} = (1100.1)_2$$

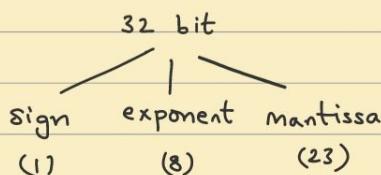
$$[0.5 \times 2 = 1]$$

∴ We get back A

IEEE 754 : Standard

↳ Floating point number - 32 bit

64 bit



For 32 bit, Bias (B) = 127

$$1100.1 = 1.1001 \times 2^3 \quad \leftarrow \text{exponent without bias}$$
$$\therefore 1024.321 = 1.024321 \times 10^3$$

$$B+e = 127 + 3 = 130 = 128 + 2$$

01000001010010

$$\text{Formula: } (-1)^s \cdot (1+m) \times 2^{e+B}$$

$$(0.125)_{10} \rightarrow ()_2$$

$$\rightarrow (0.001)_2$$

$$\left[\because \frac{1}{8} = 0.125 \right]$$

$$\begin{aligned} 0.125 \times 2 &= 0.25 \\ 0.25 \times 2 &= 0.5 \\ 0.5 \times 2 &= 1.0 \\ \Rightarrow 0.001 \end{aligned}$$

$$(0.001)_2 = 1 \times 2^{-3}$$

$$B+e = 127 + (-3)$$

$$= 124$$

$$= 128 - 4$$

Bias

↪ Negative numbers can
be stored

0.1111100000000

$$-0.125 \equiv 1.1111000000000$$

$$\begin{array}{rccccc} \text{Ex. } 0 & 0111110 & 101000 & \Rightarrow & 1.101 \times 2^e & 126 - e = 127 \\ & s & e & m & 1.101 \times 2^{-1} & \Rightarrow e = -1 \\ & & & & = (0.1101)_2 & \\ & & & & = (0.8125)_{10} & 0.0625 \\ & & & & & 250 \\ & & & & & 5 \end{array}$$

Ex. 0 10000011 110000

$$(-1)^s = 1$$

$$131 - 127 = \underline{\underline{4}}$$

$$\begin{aligned} &= 1.11 \times 2^4 \\ &= 11100 = (28.0)_{10} \end{aligned}$$

Q) -24.625

$$24.625 \equiv (11000.101)$$

$$1.1000101 \times 2^4$$

$$B + e = 127 + 4 = 131$$

$$(10000011 \ 1000101)$$

$$128 + 3$$

Q) 1 10000111 111000

$$\text{C} \rightarrow 128 + 4 + 2 + 1 \Rightarrow 127 + 8 \Rightarrow \underline{\underline{e=8}}$$

$$1.111 \times 2^8 = 111100000 \\ = (-480)_{10}$$

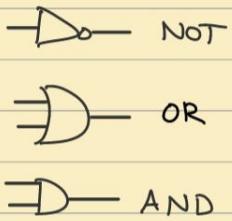
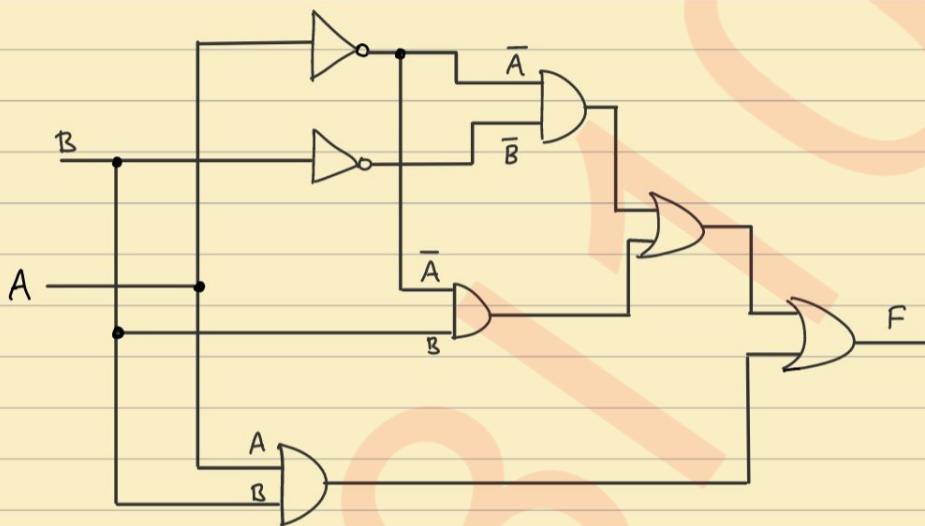
$$\left[\begin{array}{r} 2 \\ 32 \\ 64 \\ 128 \\ \hline 256 \\ 480 \end{array} \right]$$

P.T.O

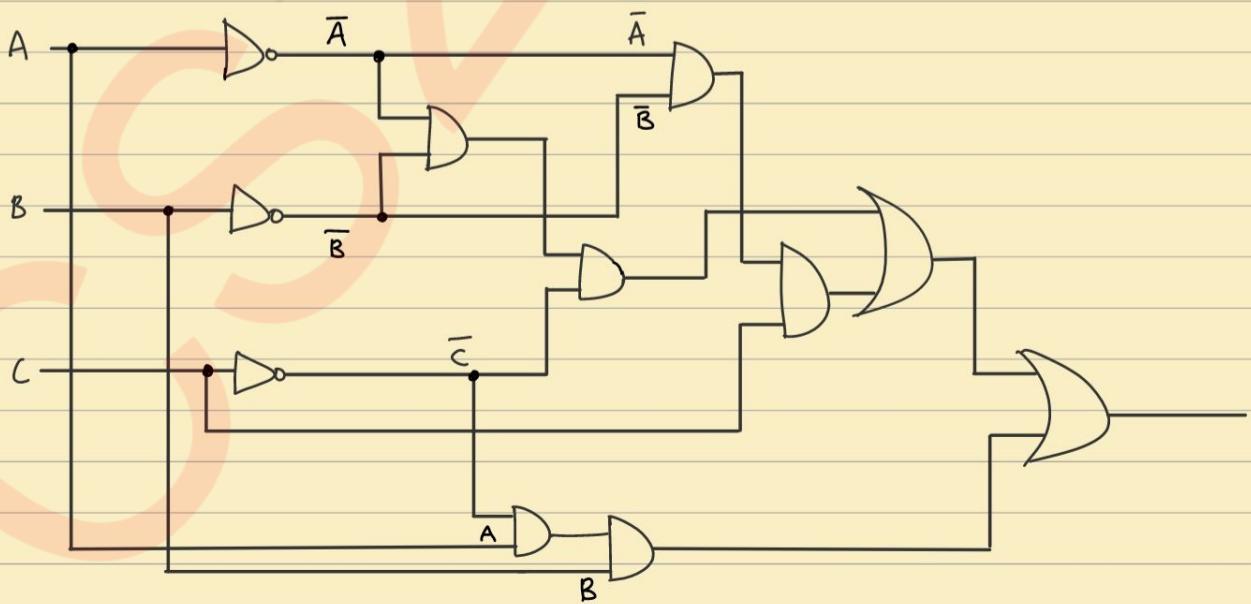
DSD Lab - I

$$\textcircled{1} \quad F(A, B) = \bar{A} \cdot \bar{B} + \bar{A} \cdot B + A \cdot B$$

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



$$\textcircled{2} \quad \bar{A}\bar{B}\bar{C} + \bar{A}\bar{B}C + AB\bar{C}$$



6/9
3 ways to represent negative nos

- (1) Signed Number
- (2) One's complement
- (3) Two's complement

(1) Signed no.

8 bit \Rightarrow 1 7
 \swarrow sign \searrow mag
+1 0 000000
-1 1 0000001

(2) 1^s complement

4 bit \rightarrow 0 0111
1 0000

Shortcome:

0 0000 - 0
1 1111 - 0

(3) 2^s complement

0 0111	1 1111	+ \rightarrow 0
$\underline{-}$	$\underline{1}$	- \rightarrow 1
1 1000	1	
$\underline{1}$	$\underline{0000}$	
1 1001		

Q) 28.875 \rightarrow float

Q) -14.8 \rightarrow float

Q) 110000110000100100 \rightarrow Decimal

Q) 0.0625 \rightarrow float

Q) 10111101100100000

32 bits
CFA12301
4 bit

$$(CFA12301)_{16} = 1100 \dots 0001$$

(1) 28.875

$$28 = 16 + 8 + 4$$

11100.111

$$\Rightarrow 1.1100111 \times 2^4$$

$$127 + 4 = 131 = 128 + 2 + 1$$

10000001

$$0.875 \times 2 = 1.75$$

$$1.75 \times 2 = 3.5$$

S = +ve \rightarrow 0

0 1000011 1100111
S e m

$$0.5 \times 2 = 1.0$$

$$\Rightarrow 0.111$$

(3) 110000110000100100

$s : 1 \text{ digit}$
 $e : 8 \text{ digits}$
 $m : 23 \text{ digits}$

$$10000110 \equiv 2 + 4 + 128$$

$s \rightarrow -ve$

$$\Rightarrow 134 - 127 = \underline{\underline{7}}$$

$$\begin{aligned} & 1.000100100 \times 2^7 \\ & = (10001001.00)_2 \\ & = 1 + 8 + 128 \\ & = -(137)_{10} \end{aligned}$$

(5) 10111101100100000

$$01111101 \rightarrow 1 + 4 + 8 + 16 + 32 + 64 \\ = 125$$

$$125 - 127 = \underline{\underline{-2}}$$

$$1.100100000 \times 2^{-2}$$

$$\begin{aligned} & = 1.001 \times 2^{-2} \\ & = (0.01001)_2 \\ & = 2^{-2} + 2^{-5} \\ & = \frac{1}{4} + \frac{1}{32} \\ & = \underline{\underline{(0.28125)_{10}}} \end{aligned}$$

$$(4) (0.0625)_{10} \equiv (0.0001)_2$$

$$(2) (-14.8)_{10}$$

$$14 = 8 + 4 + 2$$

$$(0.8)_{10} \equiv (0.1100110011001100)_2$$

$$0.8 \times 2 = 1.6$$

$$(1110.110011)_2$$

$$0.6 \times 2 = 1.2$$

$$= 1.110110011 \times 10^3$$

$$0.2 \times 2 = 0.4$$

$$127 + 3 = 130 = 128 + 2$$

$$10000010$$

$$\Rightarrow 0.11001100\dots$$

$$\therefore 110000101100110011001100110011$$

$$\swarrow s \leftarrow ve \quad \swarrow e \quad \swarrow m$$

H.W

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

$$q) \bar{A}\bar{B}\bar{C} + \bar{A}\bar{B}\bar{C} + ABC + \bar{ABC} \rightarrow \text{Solve by converting into canonical form}$$

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

$$q) 24 - 2 \text{ (2s complement)}$$

$$q) A \odot B \odot C \text{ (Truth table)}$$

$$q) (A+C) \cdot (A+B) \cdot A$$

9/9

$$\begin{aligned} A1) & A\bar{B}(C + \bar{C}) + \bar{A}\bar{B}(C + \bar{C}) + B \\ &= A\bar{B} + \bar{A}\bar{B} + B \\ &= \bar{B}(A + \bar{A}) + B \\ &= \bar{B} + B \\ &= 1 \end{aligned}$$

$$\begin{aligned} A2) & \bar{A}\bar{B}\bar{C} + \bar{A}\bar{B}\bar{C} + ABC + \bar{ABC} \\ &= \bar{A}\bar{C}(B + \bar{B}) + ABC + \bar{A} + \bar{B} + \bar{C} \\ &= \bar{A}\bar{C} + ABC + \bar{A} + \bar{B} + \bar{C} \\ &= \bar{A}(\bar{C} + 1) + ABC + \bar{B} + \bar{C} \\ &= \bar{A} + \bar{B} + \bar{C} + ABC \\ &= \bar{ABC} + ABC \\ &= 1 \end{aligned}$$

$$\begin{aligned} A3) & ABC + \bar{A}\bar{B}\bar{C} + AB\bar{C} \\ &= AB(C + \bar{C}) + \bar{A}\bar{B}\bar{C} \\ &= AB + \bar{A}\bar{B}\bar{C} \end{aligned}$$

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

$$\begin{aligned} & ABC + \bar{C}(\bar{A}\bar{B} + AB) \\ &= ABC + \bar{C}(0 + AB) \\ &= ABC + \bar{C} \cdot AB \\ &= AB(C + \bar{C}) \\ &= AB \end{aligned}$$

$$\begin{aligned} \overline{A+B} &= \overline{A+\bar{A}B} \\ \overline{A} \cdot \overline{B} &= \overline{A} \cdot \overline{A} \overline{B} \\ &= 0 \end{aligned}$$

A5)	A	B	C	$A \odot B$	$A \odot B \odot C$
	1	1	1	1	1
	1	1	0	1	0
	1	0	1	0	0
	1	0	0	0	0
	0	1	1	0	0
	0	1	0	0	0
	0	0	1	1	1
	0	0	0	1	0

$$A \odot B \odot C$$

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

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

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

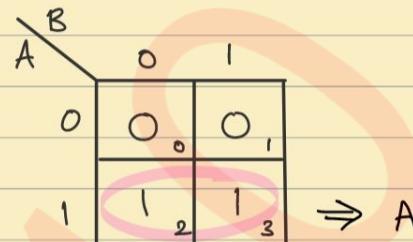
$$\Rightarrow ABC + \bar{A}\bar{B}C + (\bar{A}B \cdot (A+B)) \cdot \bar{C}$$

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

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

→ K-Maps : (Karnaugh - Maps)

A	B	$F(A \cdot B)$
0	0	0
0	1	0
1	0	1
1	1	1



0 : \bar{A}

1 : A

		BC	00	01	11	10
		A	0	1	0	0
A	B	0	0	1	0	0
		1	0	0	1	1

SOP - Min Terms

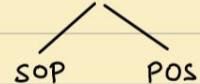
POS - Max Terms

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

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

$$\Rightarrow F(A, B, C) = \sum(m_1, m_6, m_7)$$

Canonical



Ex.

		F				
		BC	00	01	11	10
		A	0	1	0	0
0	0	1	0	1	0	0
1	0	0	4	0	5	7
0	1	0	0	1	1	1
1	1	1	6	1	7	1

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

		F				
		BC	00	01	11	10
		A	0	1	1	0
0	0	1	0	1	1	0
1	0	0	4	1	5	7
0	1	0	0	1	1	1
1	1	1	6	1	7	1

$$F = C + AB + AC$$

[Preference : More Power]

		F				
		BC	00	01	11	10
		A	0	1	0	0
0	0	1	0	1	0	0
1	0	0	4	1	5	7
0	1	0	0	1	1	1
1	1	1	6	1	7	1

$$F = \bar{A}\bar{B} + AB + \bar{B}C \quad (\text{OR})$$

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

Ex.

		F				
		00	01	11	10	
		0	0 ₀	0 ₁	1 ₃	1 ₂
		1	0 ₄	1 ₅	1 ₇	1 ₆

$$F = B + AC$$

Ex.

		F				
		00	01	11	10	
		0	0 ₀	0 ₁	1 ₃	1 ₂
		1	1 ₄	1 ₅	1 ₇	1 ₆

$$F = A + B$$

Ex.

		F				
		00	01	11	10	
		0	1 ₀	0 ₁	1 ₃	1 ₂
		1	1 ₄	1 ₅	1 ₇	1 ₆

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

Ex.

		F				
		00	01	11	10	
		0	0 ₀	0 ₁	0 ₃	0 ₂
		1	1 ₄	0 ₅	0 ₇	1 ₆

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

Ex.

		F				
		00	01	11	10	
		0	1 ₀	0 ₁	1 ₃	1 ₂
		1	1 ₄	0 ₅	0 ₇	1 ₆

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

Ex.

		F				
		00	01	11	10	
		0	1 ₀	1 ₁	0 ₃	0 ₂
		1	1 ₄	0 ₅	1 ₇	1 ₆

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

		F				
		00	01	11	10	
		0	1 ₀	1 ₁	0 ₃	1 ₂
		1	0 ₄	1 ₅	1 ₇	1 ₆

$$F = \overline{AB} + AC + \overline{BC}$$

(OR)

		F				
		00	01	11	10	
		0	1 ₀	1 ₁	0 ₃	1 ₂
		1	0 ₄	1 ₅	1 ₇	1 ₆

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

		F				
		00	01	11	10	
		0	1 ₀	0 ₁	1 ₃	0 ₂
		1	0 ₄	1 ₅	0 ₇	1 ₆

$$F = A \oplus B \oplus C$$

		F				
		00	01	11	10	
		0	1 ₀	1 ₁	1 ₃	1 ₂
		1	1 ₄	1 ₅	1 ₇	1 ₆

$$F = 1$$

10/9

Q) $\sum(m_0, m_2, m_3, m_4, m_6, m_7)$

		F				
		00	01	11	10	
		0	1 ₀	0 ₁	1 ₃	1 ₂
		1	1 ₄	0 ₅	1 ₇	1 ₆

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

Q) $\sum(m_0, m_2, m_3, m_4, m_5, m_7)$

		F				
		00	01	11	10	
		0	1 ₀	0 ₁	1 ₃	1 ₂
		1	1 ₄	1 ₅	1 ₇	0 ₆

$$F = \overline{BC} + AC + \overline{AB}$$

Q) $\sum(m_0, m_1, m_3, m_4, m_5, m_6, m_7)$

		F				
		00	01	11	10	
		0	1 ₀	0 ₁	1 ₃	1 ₂
		1	1 ₄	1 ₅	1 ₇	1 ₆

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

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

$$\sum (m_0, m_1, \dots, m_5, m_7, m_8, m_{12}, m_{13}, m_{15})$$

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

(f)

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

$$F(A, B, C, D)$$

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

(g)

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

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

If a combination of 4 reduces groups,
Then 4 is preferred.

Group \equiv Prime Implicant (PI)

Essential Group \equiv Essential Prime Implicant

Without which reduction can't happen

Non-essential / Extra group \equiv Redundant Prime Implication

Repetition

Unnecessary

Supportive \neq Redundant

S. PI

R. PI

(h)

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

$$F = B + A + D$$

8 combinations \rightarrow 3 variables reduced

4 combinations \rightarrow 2 variables reduced

2 combinations \rightarrow 1 variable reduced

H.W

$$\begin{aligned}\Sigma(m_0, m_1, m_3, m_4, m_8, m_{11}, m_{12}, m_{13}, m_{15}) \\ = \overline{CD} + \overline{ABD} + ABD + \overline{BCD}\end{aligned}$$

11/9

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

Binary	Gray Code:
00	00
01	01
10	11
11	10

Binary	Gray code:
000	000
001	001
010	011
011	010
100	110
101	111
110	101
111	100

Blueprints:

A	B	0	1
0	0	0	1
1	1	2	3

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

AB	CDE	000	001	011	010	110	111	101	100
00	0	1	3	2	6	0	5	4	
01	8	19	11	10	14	15	13	12	
11	24	25	27	26	30	31	29	28	
10	16	17	19	18	22	23	21	20	

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

$$Q) \Sigma(m_0, m_1, m_2, m_3, m_5, m_7, m_{17}, m_{19}, m_{21}, m_{22})$$

AB	CDE	000	001	011	010	110	111	101	100
00	10	11	13	12	6	07	05	4	
01	8	19	11	10	14	15	13	12	
11	24	25	27	26	30	31	29	28	
10	16	17	19	18	22	23	21	20	

Mirror elements can be combined.



Q)

AB		CDE							
		000	001	011	010	110	111	101	100
00	10	11	13	12	6	07	05	4	
01	8	19	111	10	14	15	13	12	
11	24	125	127	26	30	31	29	28	
10	16	17	119	18	22	123	121	120	

Q)

AB		CDE							
		000	001	011	010	110	111	101	100
00	10	11	13	12	6	07	05	4	
01	8	19	111	10	14	15	13	12	
11	24	125	127	26	30	31	29	28	
10	16	17	119	18	22	123	121	120	

Q)

AB		CDE							
		000	001	011	010	110	111	101	100
00	10	11	13	12	6	07	05	4	
01	8	19	011	110	14	15	13	12	
11	24	025	127	026	030	131	29	28	
10	016	017	119	018	022	023	121	020	

$$\bar{A}CDE + AC\bar{D}E \Rightarrow ADE$$

Q)

AB		CDE							
		000	001	011	010	110	111	101	100
00	10	11	13	12	6	07	05	4	
01	8	19	011	110	14	15	13	12	
11	24	025	127	126	030	131	29	28	
10	016	017	119	018	022	023	121	020	

Cannot combine / Not mirrorable

Q)

AB		CDE							
		000	001	011	010	110	111	101	100
00	10	11	13	12	6	07	05	4	
01	8	19	111	110	014	115	113	112	
11	24	125	127	126	030	131	129	128	
10	016	017	119	018	022	023	121	020	

$$\bar{B}CE + BCE = BE$$

AB		CDE							
		000	001	011	010	110	111	101	100
00	1 ₀	1 ₁	1 ₃	1 ₂	6	0 ₇	0 ₅	4	
01	8	1 ₉	0 ₁₁	1 ₁₀	14	1 ₁₅	1 ₁₃	12	
11	24	0 ₂₅	1 ₂₇	1 ₂₆	0 ₃₀	1 ₃₁	29	28	
10	1 ₁₆	0 ₁₇	0 ₁₉	1 ₁₈	1 ₂₂	0 ₂₃	0 ₂₁	1 ₂₀	

AB		CDE							
		000	001	011	010	110	111	101	100
00	1 ₀	1 ₁	1 ₃	1 ₂	6	0 ₇	0 ₅	4	
01	8	1 ₉	0 ₁₁	1 ₁₀	14	1 ₁₅	1 ₁₃	12	
11	24	0 ₂₅	1 ₂₇	1 ₂₆	0 ₃₀	1 ₃₁	29	28	
10	1 ₁₆	0 ₁₇	0 ₁₉	1 ₁₈	1 ₂₂	0 ₂₃	0 ₂₁	1 ₂₀	

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

AB		CDE							
		000	001	011	010	110	111	101	100
00	1 ₀	0 ₁	0 ₃	1 ₂	1 ₆	0 ₇	0 ₅	1 ₄	
01	8	0 ₉	0 ₁₁	1 ₁₀	1 ₁₄	1 ₁₅	1 ₁₃	12	
11	24	1 ₂₅	0 ₂₇	1 ₂₆	1 ₃₀	1 ₃₁	1 ₂₉	28	
10	1 ₁₆	0 ₁₇	0 ₁₉	1 ₁₈	1 ₂₂	0 ₂₃	0 ₂₁	1 ₂₀	

Q) $\sum(m_0, m_1, m_3, m_7, m_9, m_{13}, m_{17}, m_{19})$

AB		CDE							
		000	001	011	010	110	111	101	100
00	1 ₀	1 ₁	1 ₃	0 ₂	0 ₆	1 ₇	0 ₅	0 ₄	
01	0 ₈	1 ₉	0 ₁₁	0 ₁₀	0 ₁₄	0 ₁₅	1 ₁₃	0 ₁₂	
11	0 ₂₄	0 ₂₅	0 ₂₇	0 ₂₆	0 ₃₀	0 ₃₁	0 ₂₉	0 ₂₈	
10	0 ₁₆	1 ₁₇	1 ₁₉	0 ₁₈	0 ₂₂	0 ₂₃	0 ₂₁	0 ₂₀	

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

Gray code : (One-bit flip)

000

001

011

010

110

111

101

100

Binary :

000

001

010

011

100

101

110

111

Lab - 2

- ① $\bar{A}\bar{B}\bar{C} + \bar{A}\bar{B}C + \bar{A}B\bar{C}$
- ② $\bar{A}\bar{B}\bar{C} + ABC + AC + AB$
- ③ $A\bar{B} + \bar{A}B$ using NOR alone

①

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

③ $\bar{A}B + A\bar{B}$

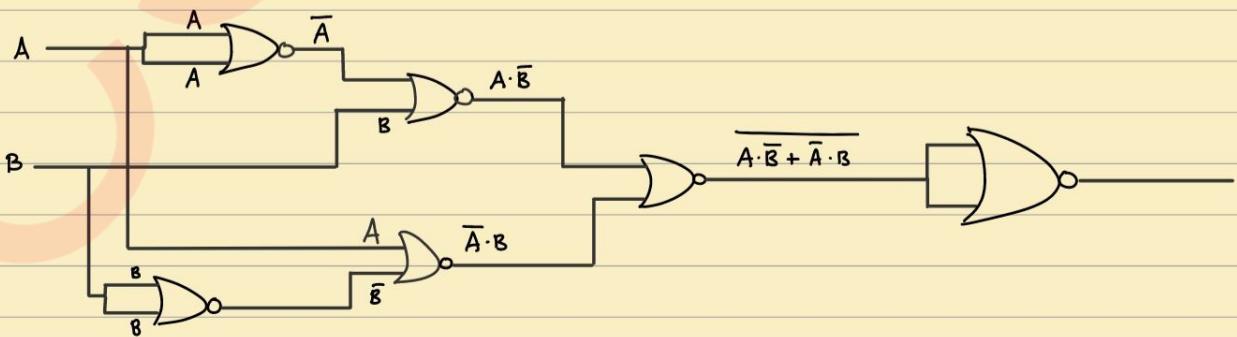
A	B	$\bar{A} + B$	$\bar{A}B$	$A\bar{B}$	F
0	0	1	0	0	0
0	1	0	1	0	1
1	0	0	0	1	1
1	1	0	0	0	0

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

$$\left[\overline{A+A} = \overline{A} \right]$$

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

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



$$g) A \oplus B \oplus C$$

Sol:

$$\begin{aligned}
 (A\bar{B} + B\bar{A}) \oplus C &= (A\bar{B} + \bar{A}B) \cdot \bar{C} + (\overline{A\bar{B} + B\bar{A}}) \cdot C \\
 &= A\bar{B}\bar{C} + \bar{A}B\bar{C} + \overline{A\bar{B}} \cdot \overline{B\bar{A}} \cdot C \\
 &= A\bar{B}\bar{C} + \bar{A}B\bar{C} + (\bar{A} + B) \cdot (A + \bar{B}) \cdot C \\
 &= A\bar{B}\bar{C} + \bar{A}B\bar{C} + (AB + \bar{A}\bar{B}) \cdot C \\
 &= A\bar{B}\bar{C} + \bar{A}B\bar{C} + ABC + \bar{A}\bar{B}C \\
 &= \bar{A}\bar{B}C + \bar{A}\bar{C}B + \bar{B}\bar{C}A + ABC \\
 &= ABC + \bar{A}\bar{B}C + \bar{A}\bar{C}B + A\bar{B}\bar{C}
 \end{aligned}$$

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

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

	111	001	010	100
BC	00	01	11	10
A	0	0 1	0 1	1
	1	1 0	1 0	0

P.T.O

Four 4 bits :

Binary code

$B_3 B_2 B_1 B_0$

0000

0001

0010

0011

0100

0101

0110

0111

1000

1001

1010

1011

1100

1101

1110

1111

Gray code:

$G_3 G_2 G_1 G_0$

0000

0001

0011

0010

0110

0111

0101

0100

1100

1101

1111

1010

1011

1001

1000

$$G_3 = B_3$$

		BC	00	01	11	10
		A	0	0	1	0
			1	1	0	1
	0		0	1	0	1
	1		1	0	1	0

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

		$B_1 B_0$	00	01	11	10
		$B_3 B_2$	00	01	11	10
			0	0	1	1
	00		0	0	1	1
	01		1	1	0	0
	11		0	0	1	1
	10		1	1	0	0

$$\begin{aligned} & \bar{B}_3 \bar{B}_2 B_1 + \bar{B}_3 B_2 \bar{B}_1 \\ & + B_3 B_2 B_1 + B_3 \bar{B}_2 \bar{B}_1 \\ & \downarrow \\ & B_1 \oplus B_2 \oplus B_3 \end{aligned}$$

		G_2	00	01	11	10
		$B_3 B_2$	00	01	11	10
			0	0	0	0
	00		0	0	0	0
	01		1	1	1	1
	11		0	0	0	0
	10		1	1	1	1

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

$B_1 B_0$	G_0			
$B_2 B_1$	00	01	11	10
00	0 8	1 9	0 13	1 12
01	0 4	1 5	0 7	1 6
11	0 12	1 13	0 15	1 14
10	0 8	1 7	0 11	1 10

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

$$G_0 = B_1 \oplus B_0$$

- Q1) $\sum (m_0, m_1, m_2, m_4, m_6, m_8, m_{12}, m_{13})$
 Q2) $\sum (m_0, m_4, m_9, m_{15}, m_{16}, m_{20}, m_{25}, m_{31})$
 Q3) $\sum (m_0, m_2, m_4, m_6, m_{16}, m_{18}, m_{20}, m_{22})$
 Q4) $\sum (m_8, m_{10}, m_{12}, m_{14}, m_{16}, m_{30})$
 Q5) $\sum (m_0, m_1, m_3, m_4, m_5, m_7, m_{16}, m_{17}, m_{19}, m_{20}, m_{21}, m_{23})$
 Q6) $\sum (m_8, m_9, m_{10}, m_{11}, m_{13}, m_{14}, m_{15}, m_{27}, m_{29})$

①

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

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

Q2)

CDE	000	001	011	010	110	111	101	100
AB	00	1 0	1 1	3 2	2 6	7 7	5 5	1 4
00	8 24	1 25	11 27	10 26	14 30	1 31	13 29	12 28
01	16 16	17 17	19 19	18 18	22 22	23 23	21 21	1 20

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

Q3)

CDE	000	001	011	010	110	111	101	100	
AB	00	1 0	1 1	3 2	1 2	1 6	7 7	5 5	1 4
00	8 24	9 25	11 27	10 26	14 30	15 31	13 29	12 28	
01	16 16	17 17	19 19	18 18	22 22	23 23	21 21	1 20	

$$\underline{\bar{B}\bar{E}}$$

Q4)

AB		CDE							
		000	001	011	010	110	111	101	100
00	0	1	3	2	6	7	5	4	
01	1	8	9	11	10	7	14	15	13
11	24	25	27	26	30	31	29	28	
10	16	17	19	18	22	23	21	20	

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

Q5)

AB		CDE								
		000	001	011	010	110	111	101	100	
00	1	0	1	3	2	6	1	7	5	4
01	8	9	11	1	10	14	15	13	12	
11	24	25	27	26	30	31	29	28		
10	16	1	17	1	19	18	22	1	23	

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

Q6)

AB		CDE							
		000	001	011	010	110	111	101	100
00	0	1	3	2	6	7	5	4	
01	1	8	1	9	1	11	1	10	1
11	24	25	1	27	26	30	31	1	29
10	16	17	19	18	22	23	21	20	

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

Q7)

AB		CDE								
		000	001	011	010	110	111	101	100	
00	0	1	1	3	2	6	1	7	5	4
01	8	1	9	11	1	10	14	15	13	12
11	24	1	25	27	26	30	1	31	29	28
10	16	1	17	1	19	18	1	22	1	23

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

P.T.O.

Binary code

$B_3\ B_2\ B_1\ B_0$	$G_3\ G_2\ G_1\ G_0$
0 0 0 0	0 0 0 0
0 0 0 1	0 0 0 1
0 0 1 0	0 0 1 1
0 0 1 1	0 0 1 0
0 1 0 0	0 1 1 0
0 1 0 1	0 1 1 1
0 1 1 0	0 1 0 1
0 1 1 1	0 1 0 0
1 0 0 0	1 1 0 0
1 0 0 1	1 1 0 1
1 0 1 0	1 1 1 1
1 0 1 1	1 1 1 0
1 1 0 0	1 0 1 0
1 1 0 1	1 0 1 1
1 1 1 0	1 0 0 1
1 1 1 1	1 0 0 0

Gray code:

$$G_3 = B_3$$

$$G_2 = B_2 \oplus B_3$$

$$G_1 = B_1 \oplus B_2$$

$$G_0 = B_0 \oplus B_1$$

18/9

Diagram showing the mapping from Binary code to Gray code. The columns are labeled $B_3\ B_2$ and G_2 , and the rows are labeled $B_3\ B_2$. The mapping is as follows:

$B_3\ B_2$	00	01	11	10
00	0 ₀	0 ₁	0 ₃	0 ₂
01	1 ₄	1 ₅	1 ₇	1 ₆
11	0 ₂	0 ₁₃	0 ₁₅	0 ₁₄
10	1 ₈	1 ₉	1 ₁₁	1 ₁₀

$$G_2 = B_2 \overline{B_3} + B_3 \overline{B_2}$$

$$B_3 \oplus B_2$$

Diagram showing the mapping from Binary code to Gray code. The columns are labeled $B_3\ B_2$ and G_1 , and the rows are labeled $B_3\ B_2$. The mapping is as follows:

$B_3\ B_2$	00	01	11	10
00	0 ₀	0 ₁	1 ₃	1 ₂
01	1 ₄	1 ₅	0 ₇	0 ₆
11	1 ₁₂	1 ₁₃	0 ₁₅	0 ₁₄
10	0 ₈	0 ₉	1 ₁₁	1 ₁₀

$$G_1 = B_1 \overline{B_2} + B_2 \overline{B_1}$$

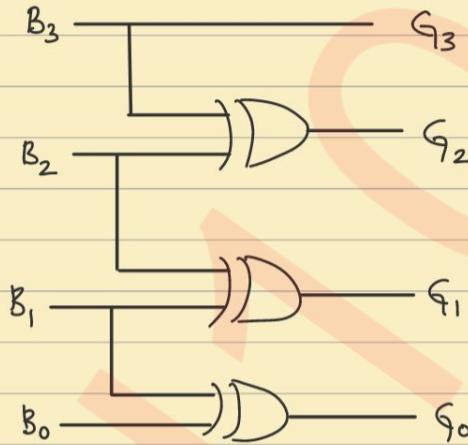
$$B_1 \oplus B_2$$

$B_3 B_2$	00	01	11	10	G_0
00	0 ₈	1 ₁	0 ₃	1 ₂	G_0
01	0 ₄	1 ₅	0 ₇	1 ₆	
11	0 ₁₂	1 ₁₃	0 ₁₅	1 ₁₄	
10	0 ₈	1 ₉	0 ₁₁	1 ₁₀	

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

$$B_1 \oplus B_0$$

Circuit:



$G_3 G_2$	00	01	11	10	B_2
00	0 ₈	0 ₁	0 ₃	0 ₂	B_2
01	1 ₄	1 ₅	1 ₇	1 ₆	
11	0 ₁₂	0 ₁₃	0 ₁₅	0 ₁₄	
10	1 ₈	1 ₉	1 ₁₁	1 ₁₀	

$$B_2 = G_3 \oplus G_2$$

$G_1 G_0$	00	01	11	10	B_1
00	0 ₈	0 ₁	1 ₃	1 ₂	B_1
01	1 ₄	1 ₅	0 ₇	0 ₆	
11	0 ₁₂	0 ₁₃	1 ₁₅	1 ₁₄	
10	1 ₈	1 ₉	0 ₁₁	0 ₁₀	

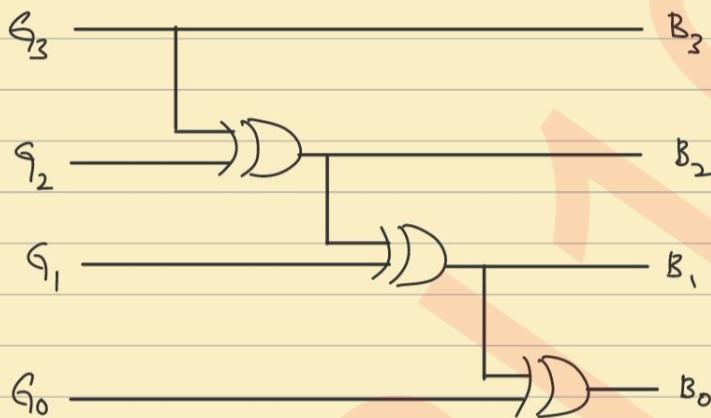
$$B_1 = \bar{G}_2 \bar{G}_3 G_1 + \bar{G}_3 G_2 \bar{G}_1 + G_3 G_2 G_1 + G_3 \bar{G}_2 \bar{G}_1$$

$$B_1 = G_3 \oplus G_2 \oplus G_1$$

		B_0				
		00	01	11	10	
G ₃ G ₂		00	0 ₈	1 ₁	0 ₃	1 ₂
		01	1 ₄	0 ₅	1 ₇	0 ₆
		11	0 ₁₂	1 ₁₃	0 ₁₅	1 ₁₄
		10	1 ₈	0 ₇	1 ₁₁	0 ₁₀

$$B_0 = G_1 \oplus G_2 \oplus G_3 \oplus G_0$$

Circuit:



$$\begin{aligned} B_3 &= G_3 \\ B_2 &= G_3 \oplus G_2 \\ B_1 &= G_3 \oplus G_2 \oplus G_1 \\ B_0 &= G_3 \oplus G_2 \oplus G_1 \oplus G_0 \end{aligned}$$

Concept

Don't care (X)

A	B	C	O/P
0	0	0	X
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

		BC				
		00	01	11	10	
A		0	X ₈	0 ₁	1 ₃	0 ₂
		1	0 ₄	0 ₅	1 ₇	1 ₆

$$BC + AB$$

Take X=0

$\because X=1$ gives extra term

X can be considered as 0 or 1 depending on the situation provided. Goal : Minimize terms.

A	B	C	O/P
0	0	0	X
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	0

BC		00	01	11	10
A		0	1	0	1
0	X	1	0	1	0
1	1	1	0	0	0
0	0	0	1	1	0

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

Take X as 1.
so that it can
be simplified.

Q) $F(A, B, C, D) = \sum(m_0, m_2, m_3, m_4, m_6, m_8, m_9, m_{12}, m_{13})$
 $+ \sum(d_1, d_5, d_{15})$

CD		00	01	11	10
AB		00	X	1	1
00	1	1	4	5	6
01	1	1	1	0	1
11	1	1	1	X	0
10	1	1	0	0	0

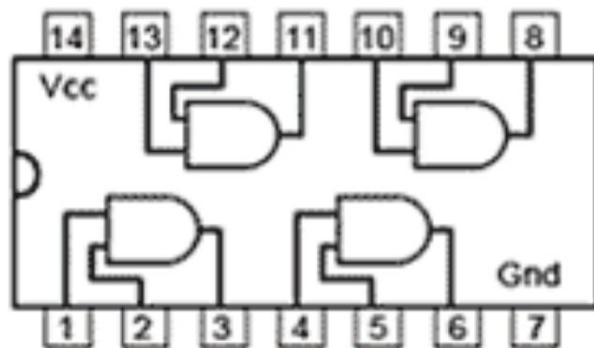
$X_1 \& X_5 = 1$
 $\therefore 8$ formation

$$X_{15} = 0$$

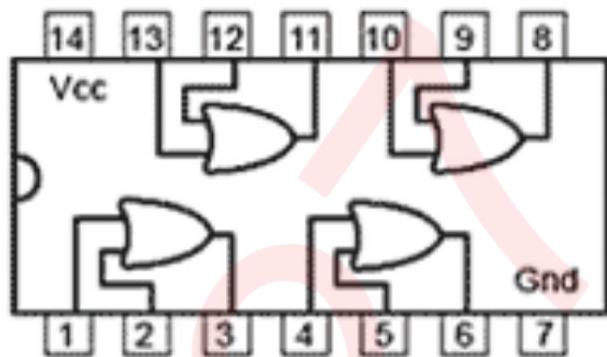
\because Extra term &
combination 1 is
already a part of 8

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

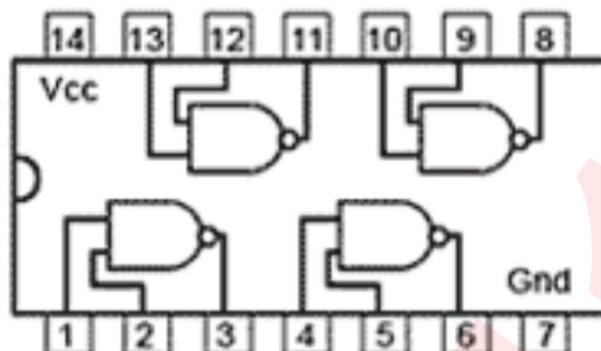
P.T.O



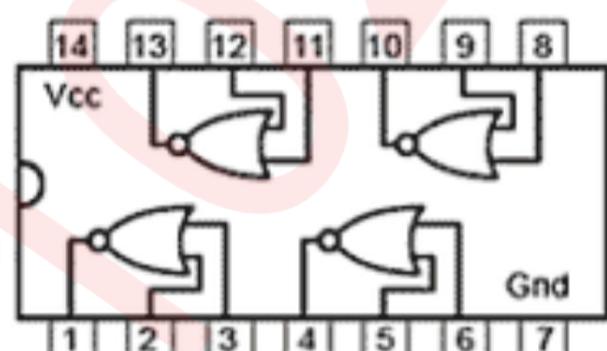
**7408 Quad 2 input
AND Gates**



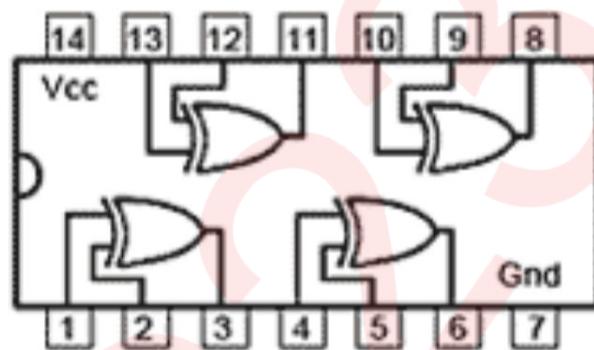
**7432 Quad 2 input
OR Gates**



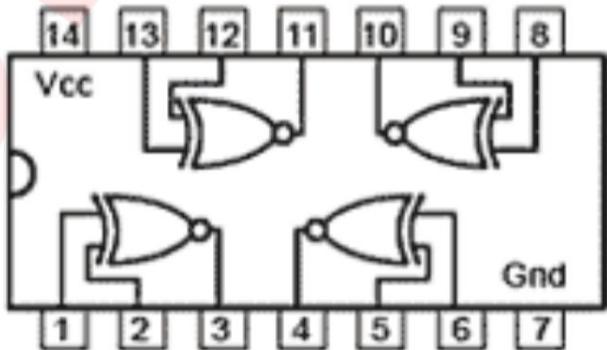
**7400 Quad 2 input
NAND Gates**



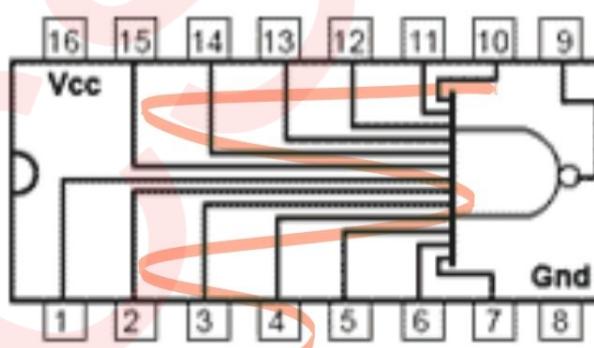
**7402 Quad 2 input
NOR Gates**



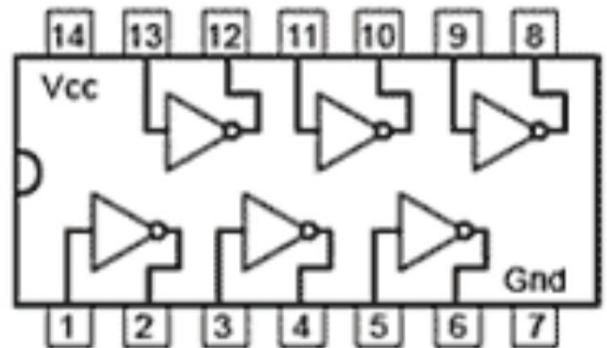
**7486 Quad 2 input
XOR Gates**



**747266 Quad 2 input
XNOR Gates**



**74133 Single 13 input
NAND Gate**



**7404 Hex NOT Gates
(Inverters)**

18/9

Lab-3

$$\textcircled{2} \quad \sum(m_0, m_2, m_4, m_6) + \sum(d_1, d_5)$$

AB
BC
F

	00	01	11	10
0	1	X	0	1
1	1	X	0	1

$$(x_1=0, x_5=0)$$

$$F = \bar{C}$$

$$\textcircled{3} \quad \sum(m_0, m_1, m_2, m_3, m_5, m_7, m_8, m_{10}, m_{14}) + \sum(d_9, d_{13})$$

AB
CD

	00	01	11	10
00	1	1	1	1
01	0	1	1	0
11	0	X	0	1
10	1	X	0	1

$$X_9 = 0$$

$$X_{13} = 0$$

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

P.T.O

19/9

BCD → Binary Coded Decimal

4	8	12
0100	1000	0001 0010
1	2	

Binary code

$B_3 \ B_2 \ B_1 \ B_0$

0000



BCD

$D_4 \ D_3 \ D_2 \ D_1 \ D_0$

0000 00

0001



0000 01

0010



0000 10

0011



0000 11

0100



0001 00

0101



0001 01

0110



0001 10

0111



0001 11

1000



0100 00

1001



0100 01

1010



1000 00 (10)

1011



1000 01 (11)

1100



1001 00 (12)

1101



1001 01 (13)

1110



1010 00 (14)

1111



1010 01 (15)

15 - 1111 (Binary)

15 - 10101 (BCD)

1 5

$$B_0 = D_0$$

$B_3 \ B_2$	$B_1 \ B_0$	00	01	11	10	D_1
00	00	0	0	1	1	1
01	01	0	0	1	1	1
11	11	1	1	0	0	0
10	10	0	0	0	0	0

$$D_1 = \overline{B}_1 \overline{B}_2 B_3 + B_1 \overline{B}_3$$

B_3B_2	B_1B_0	D_2	00	01	11	10
00			0	0	0	0
01			1	1	1	1
11			0	0	1	1
10			0	0	0	0

$$D_2 = B_2B_1 + \overline{B}_3B_2$$

B_3B_2	B_1B_0	D_3	00	01	11	10
00			0	0	0	0
01			0	0	0	0
11			0	0	0	0
10			1	1	0	0

$$D_3 = B_3\overline{B}_2\overline{B}_1$$

B_3B_2	B_1B_0	D_4	00	01	11	10
00			0	0	0	0
01			0	0	0	0
11			1	1	1	1
10			0	0	1	1

$$D_4 = B_3B_2 + B_2B_1$$

$$D_3 = B_3\overline{B}_2\overline{B}_1$$

$$D_2 = \overline{B}_3B_2 + B_2B_1$$

$$D_1 = B_3\overline{B}_2\overline{B}_1 + B_1\overline{B}_3$$

$$D_0 = B_0$$

$$D_4 = B_3 B_2 + B_2 B_1$$

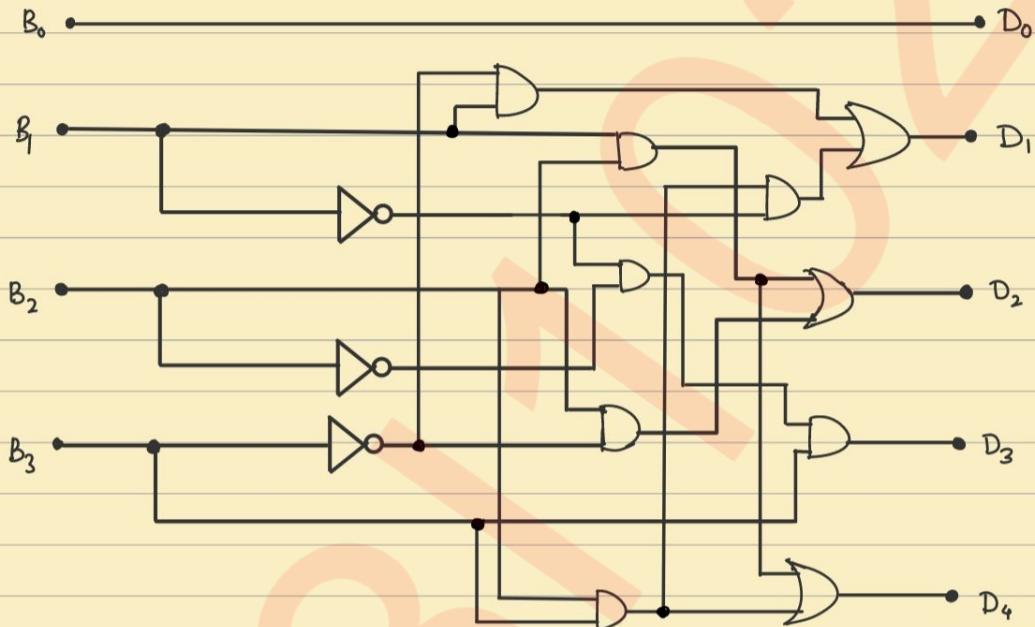
$$D_3 = B_3 \bar{B}_2 \bar{B}_1$$

$$D_2 = \bar{B}_3 B_2 + B_2 B_1$$

$$D_1 = B_3 \bar{B}_2 \bar{B}_1 + \bar{B}_1 B_3$$

$$D_0 = B_0$$

Circuit Diagram :



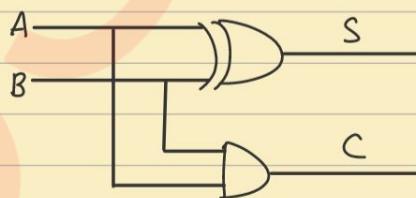
→ Addition :

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

$$S \rightarrow \text{XOR} \quad (A\bar{B} + \bar{A}B = A \oplus B)$$

$$C \rightarrow \text{AND} \quad (A \cdot B)$$

Half - Adder :



X	Y	C_{in}	S	C
1	1	1	1	1
1	1	0	0	1
1	0	1	0	1
1	0	0	1	0
0	1	1	0	1
0	1	0	1	0
0	0	1	1	0
0	0	0	0	0

But $\begin{array}{r} 1 \\ 1 \end{array}$ Half Adder can handle
 $\begin{array}{r} 1 \\ 1 \end{array}$ only 2 inputs
 $\underline{1 \ 0}$ \therefore We use Full Adder

$$S : X \oplus Y \oplus Z$$

$$C : XY + YZ + ZX$$

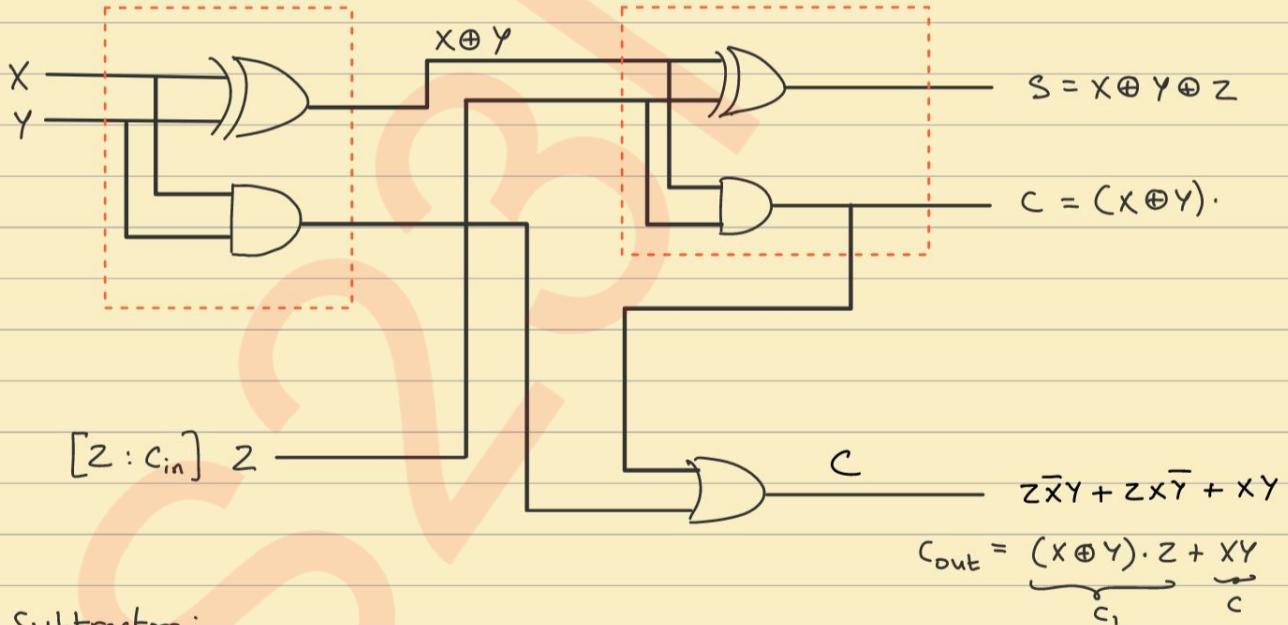
	BC	S			
A		00	01	11	10
0	0	0	1	0	1
1	1	1	0	1	0

	BC	C			
A		00	01	11	10
0	0	0	0	1	0
1	0	1	1	1	1

Without using
Half Adders

209:

Full Adders using 2 Half Adders



Subtractor:

X	Y	D	B
1	1	0	0
1	0	1	0
0	1	1	1
0	0	0	0

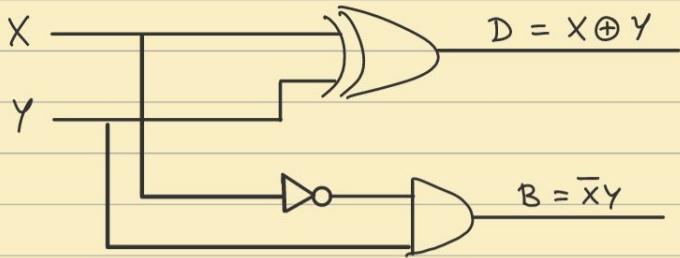
$$\begin{array}{r} -0 \\ -0 \\ \hline 00 \end{array}$$

$B \leftarrow D$

$$\begin{array}{r} -0 \\ -0 \\ \hline 01 \end{array}$$

$$D = X \oplus Y$$

$$B = \bar{X}Y$$



X	Y	Z	D	B
1	1	1	1	1
1	1	0	0	0
1	0	1	0	0
1	0	0	1	0
0	1	1	0	1
0	1	0	1	1
0	0	1	1	1
0	0	0	0	0

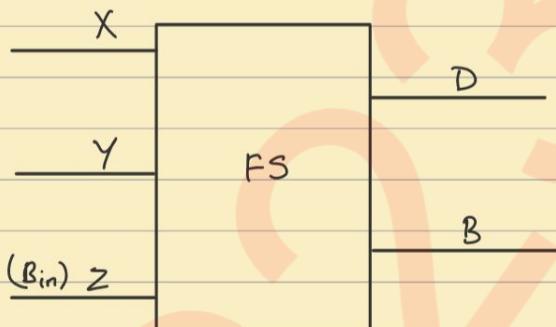
$$D = X \oplus Y \oplus Z$$

$$B = Z(X \oplus Y) + \bar{X}Y$$

B

$X \backslash YZ$	00	01	11	10
0	0 ₀	1 ₁	1 ₃	1 ₂
1	0 ₄	0 ₅	1 ₇	0 ₆

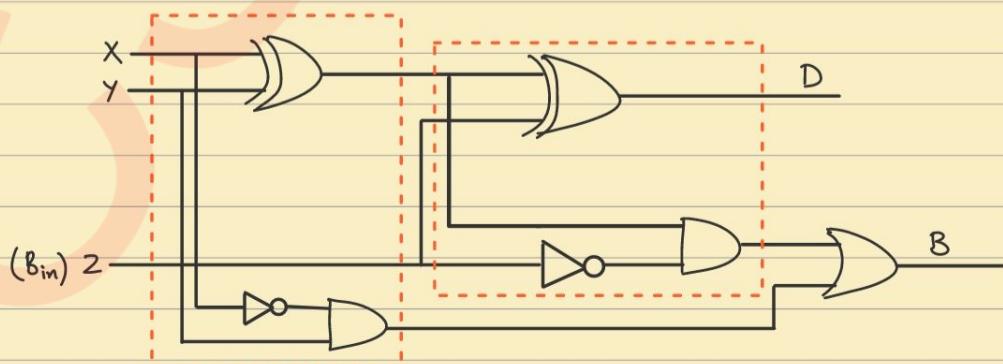
$$\bar{X}Z + \bar{X}Y + YZ$$



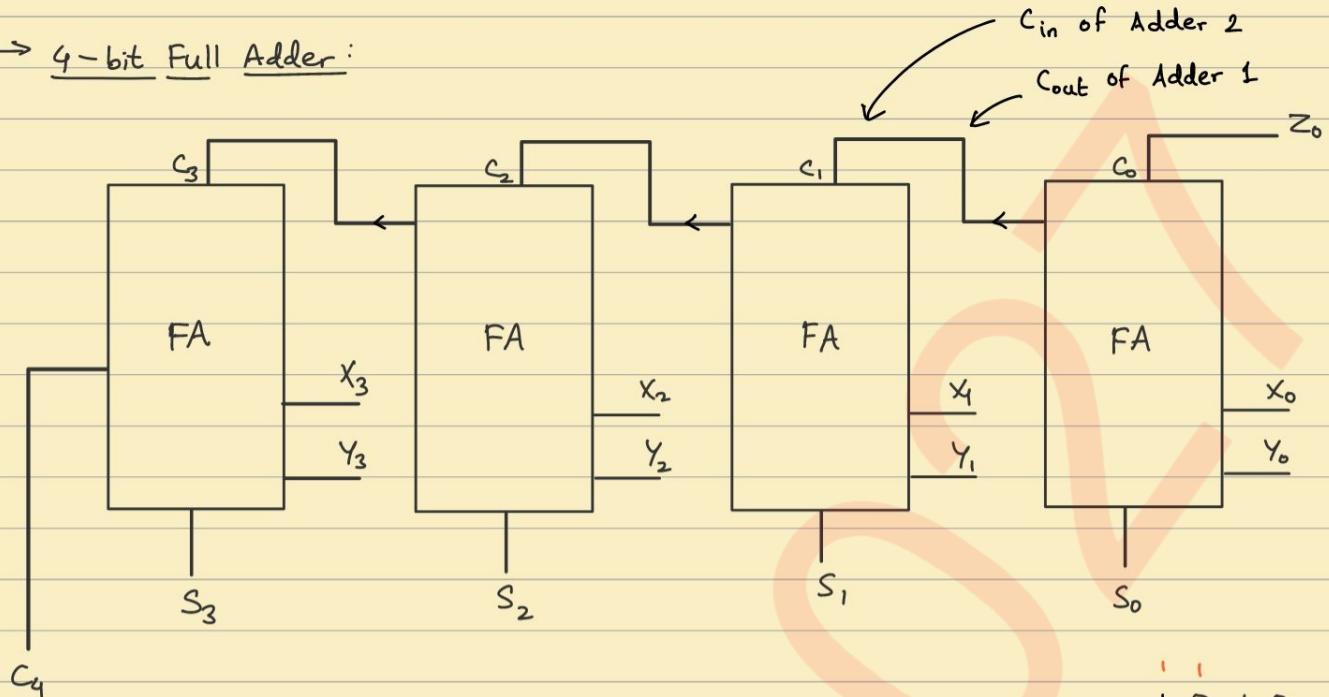
$$B_0 = X\bar{Y}$$

$$B_{out} (B_i) = (\bar{X} \oplus Y)Z + \bar{X}Y$$

Full Subtractor



→ 4-bit Full Adder:



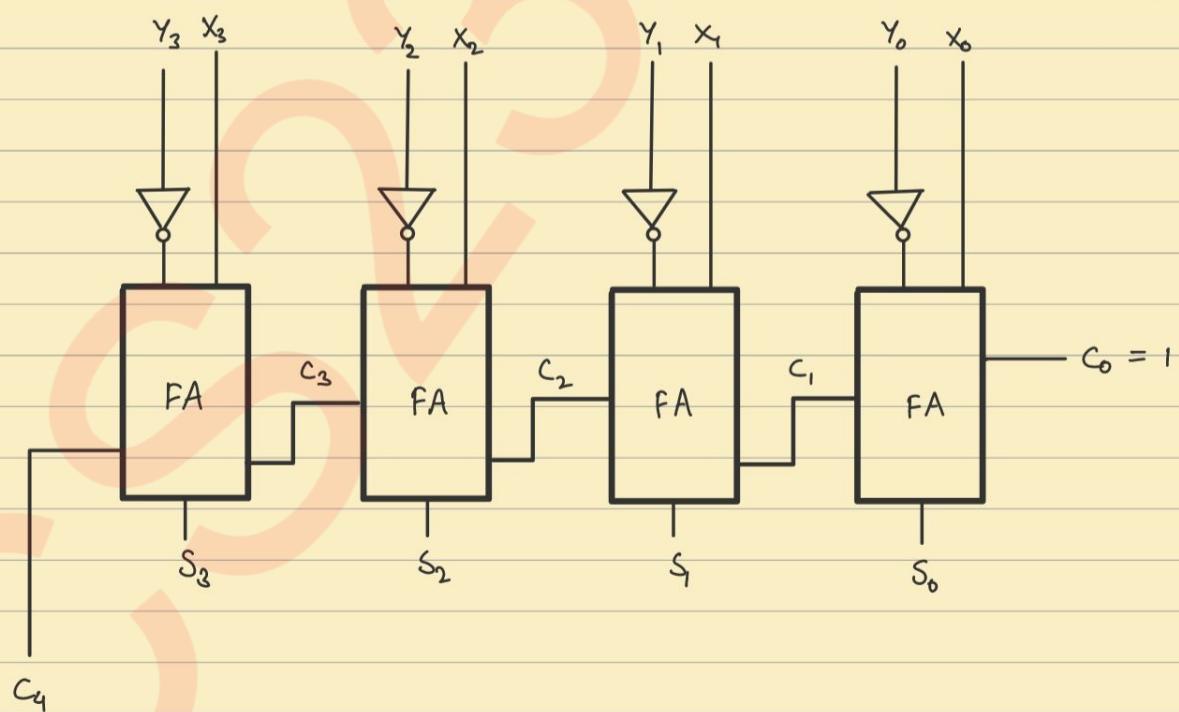
$$\begin{array}{r}
 1 \\
 1010 \\
 0111 \\
 \hline
 10001
 \end{array}$$

→ 4-bit Subtractor using Adder:

$$\begin{array}{r}
 14 \\
 - 5 \\
 \hline
 9
 \end{array}$$

14 + 2s complement of 5 = 9

$$\begin{array}{r}
 1110 \rightarrow 14 \\
 1011 \rightarrow 2s c(5) \\
 \hline
 \textcircled{1} \underbrace{1001}_{\rightarrow 9}
 \end{array}$$



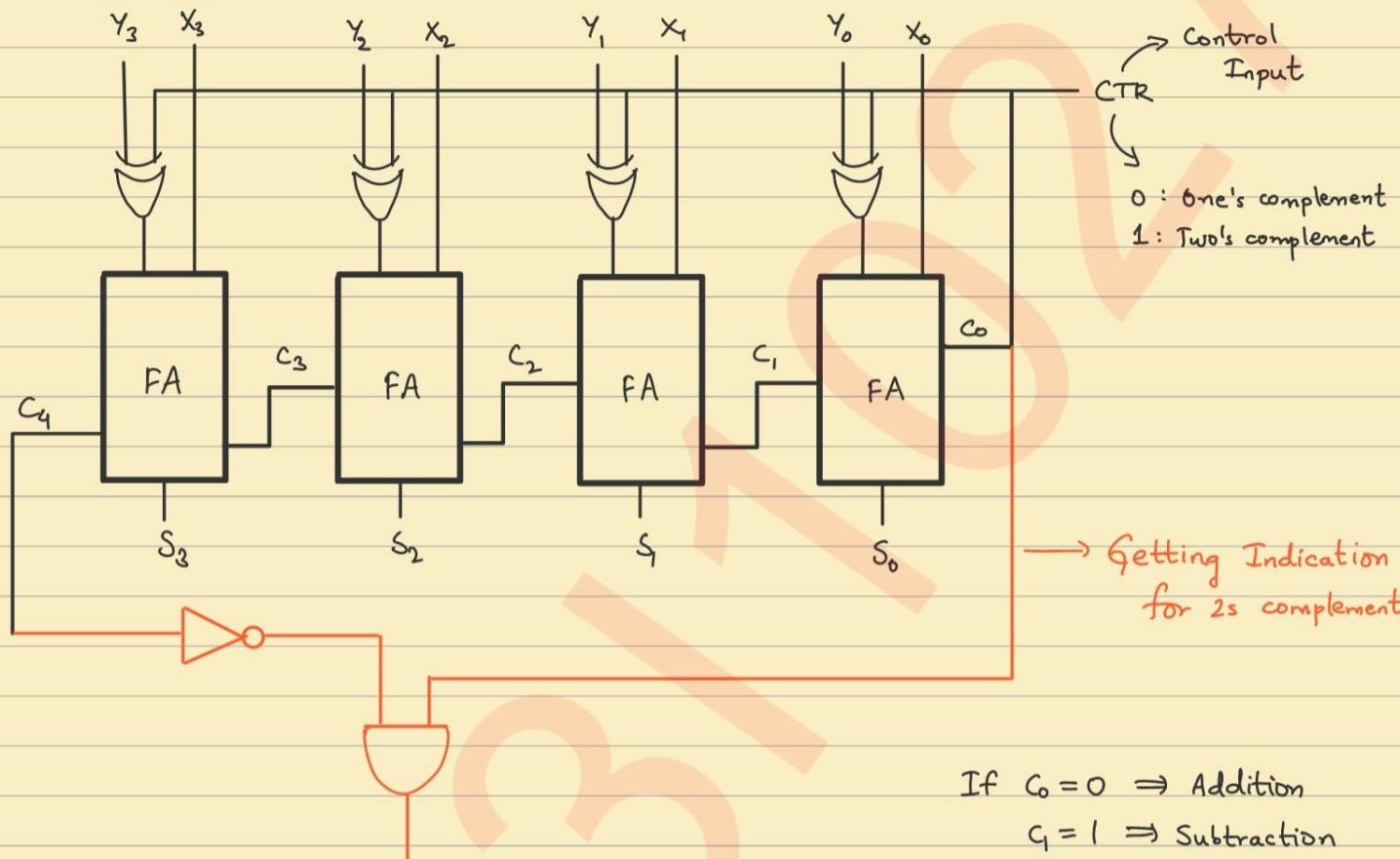
$$2s \text{ complement} = 1 + \underbrace{1s \text{ complement}}_{C_0}$$

Not Gates

$$A \oplus 0 = A$$

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

→ 4-bit Adder - Subtractor:



Note: Output carry (C_4) is ignored in subtraction

When $A < B$:

$$\begin{array}{r} 10 \\ - 12 \\ \hline \end{array} \equiv \begin{array}{r} 1010 \\ - 1100 \\ \hline \end{array}$$

$10 + 2s$ complement of 12 :
 \downarrow \curvearrowright
 1010 0100

$$\begin{array}{r} 1010 \\ 0100 \\ \hline 1110 \end{array}$$

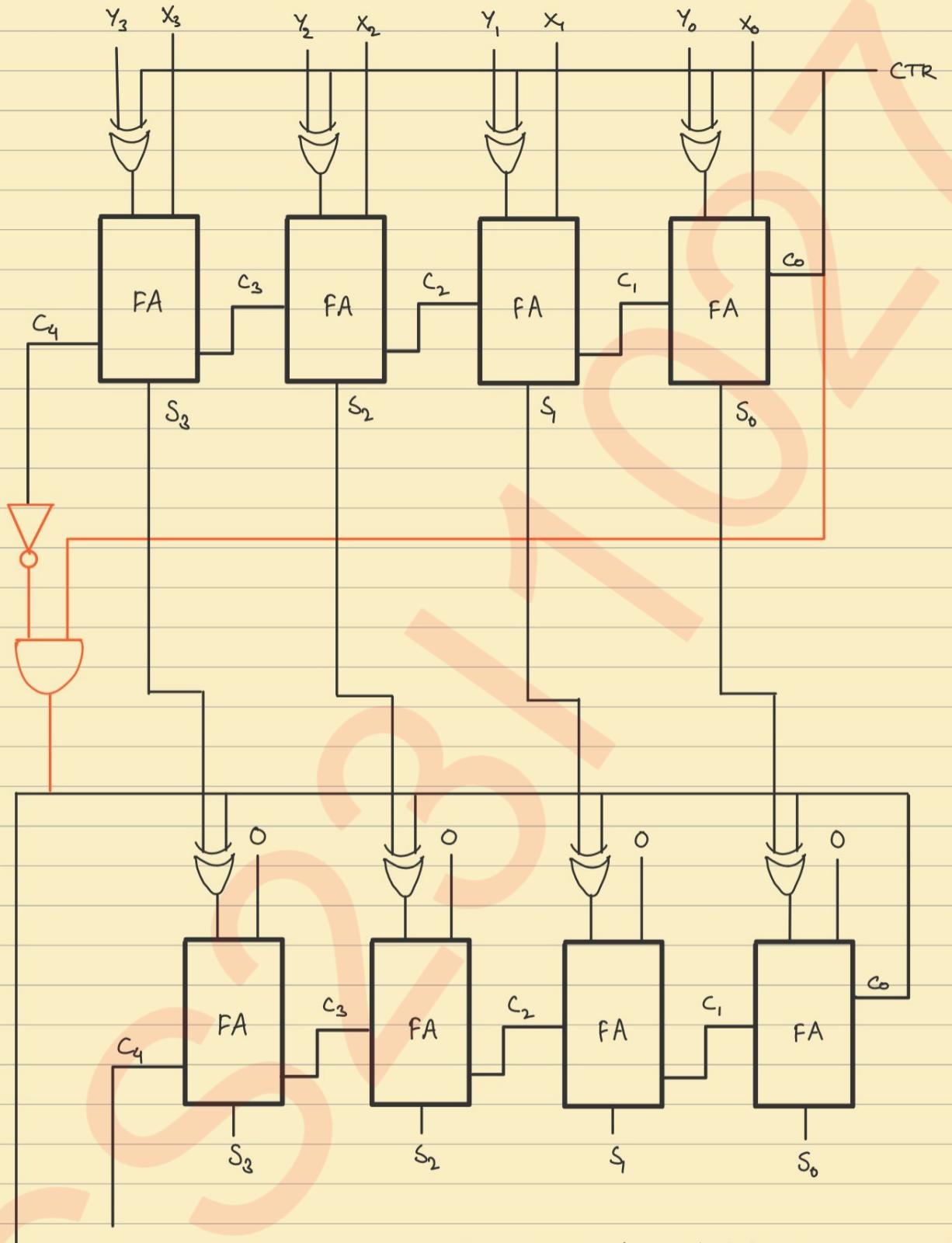
No carry

\therefore When $CTR = 1 \& C_4 = 0 \Rightarrow A < B$

But Answer displayed — $1\overbrace{110}^2$

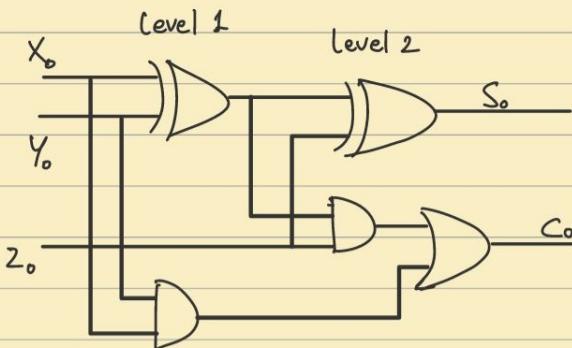
Indicating output is in Two's complement form

To convert 2s complement result into its true form,
One more adder circuit is introduced.



The XOR gates calculate 1s complement

Sign



Carry Propagator

$$P_0 = X_0 \oplus Y_0 \quad | \quad S = P \oplus C$$

$$G_0 = X_0 \cdot Y_0$$

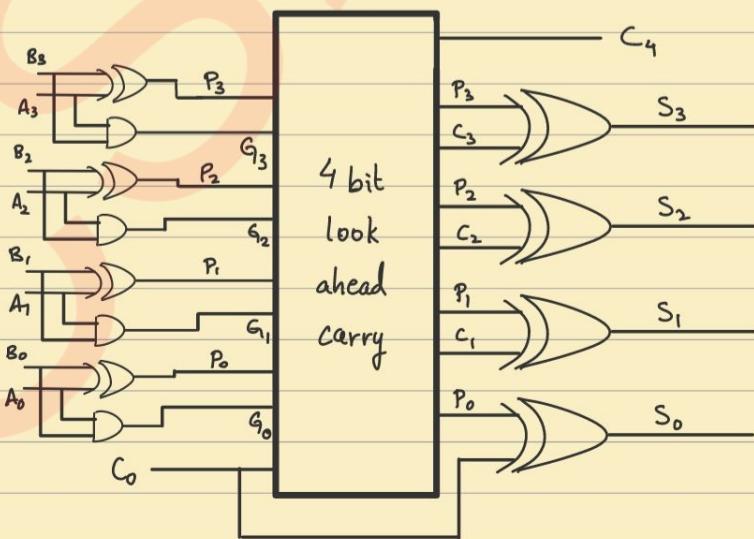
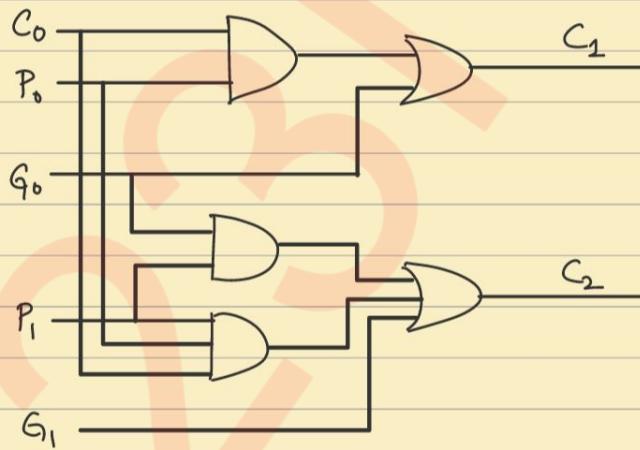
Carry Generator

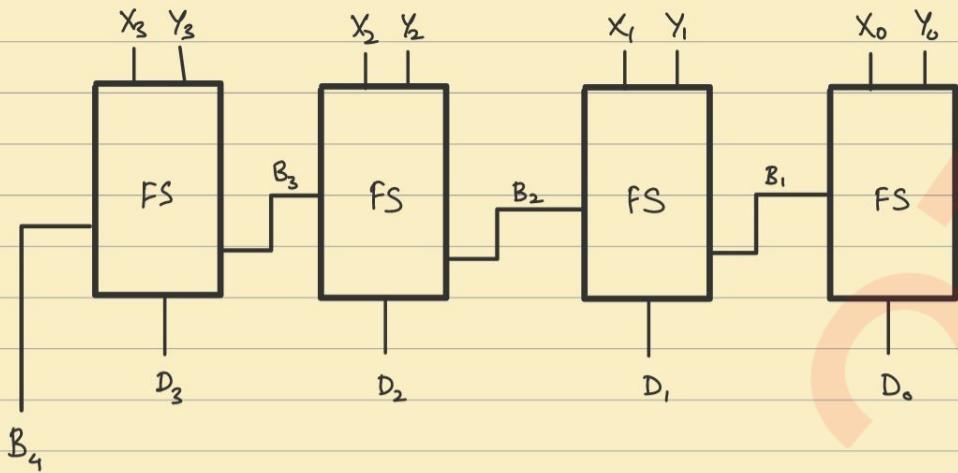
4 bit look ahead Carry Adder

$$\begin{aligned} C_1 &= G_0 + P_0 C_0 \\ C_2 &= G_1 + P_1 C_1 \\ C_3 &= G_2 + P_2 C_2 \end{aligned} \quad \left. \right\} \Rightarrow C_n = G_{n-1} + P_{n-1} \cdot C_{n-1}$$

$$\begin{aligned} C_2 &= G_1 + P_1 C_1 \\ &= G_1 + P_1 (G_0 + P_0 C_0) \\ \therefore C_2 &= G_1 + P_1 G_0 + P_0 P_1 C_0 \end{aligned}$$

$$\begin{aligned} C_3 &= G_2 + P_2 C_2 \\ &= G_2 + P_2 (G_1 + P_1 G_0 + P_0 P_1 C_0) \\ &= G_2 + P_2 G_1 + P_1 P_2 G_0 + P_0 P_1 P_2 C_0 \end{aligned}$$

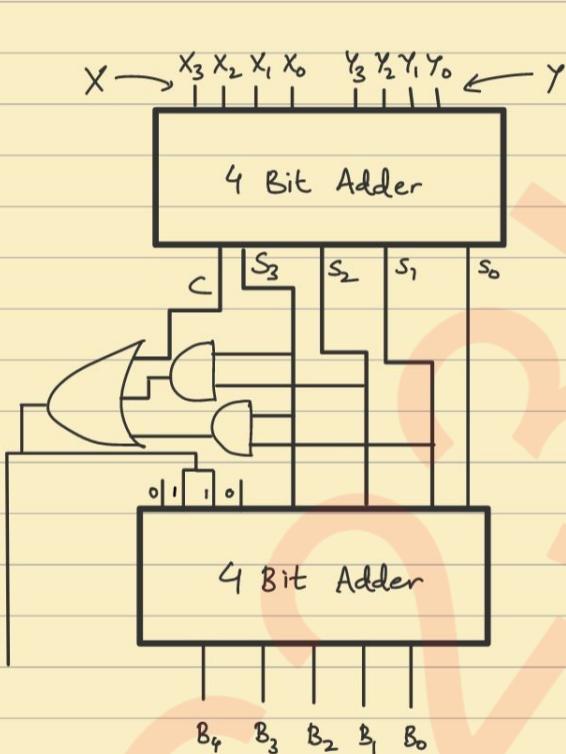




$$\rightarrow \bar{x}y + z(\bar{x} \oplus y)$$

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

$$D = x \oplus y \oplus z$$



	<u>Binary Sum</u>	<u>BCD Sum</u>
(0)	00000	00000 (0)
(1)	00001	00001 (1)
	:	:
(9)	01001	01001 (9)
(10)	01010	10000 (16)
(11)	01011	10001 (17)
(12)	01100	10010 (18)
(13)	01101	10011 (19)
(14)	01110	10100 (20)
(15)	01111	10101 (21)
(16)	10000	10110 (22)
(17)	10001	10111 (23)
(18)	10010	11000 (24)

B_4		$S_2 S_1 S_0$							
		000	001	011	010	110	111	101	100
00	00	0, 0	0, 1	0, 3	0, 2	0, 6	0, 7	0, 5	0, 4
01	01	0, 8	0, 9	1, 11	1, 10	1, 14	1, 15	1, 13	1, 12
11	X ₂₄	X ₂₅	X ₂₇	X ₂₆	X ₃₀	X ₃₁	X ₂₉	X ₂₈	
10	1, 16	1, 17	X ₁₉	1, 18	X ₂₂	X ₂₃	X ₂₁	X ₂₀	

$$B_4 = C + S_1 S_3 + S_2 S_3$$

25/9

→ Magnitude Comparator:

$$A < B : \bar{A}_0 B_0$$

$$A > B : A_0 \bar{B}_0$$

$$A = B : A_0 B_0 + \bar{A}_0 \bar{B}_0 = A_0 \odot B_0$$

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

1-bit Comparator \rightarrow 2 variables - 4 rows

2-bit Comparator \rightarrow 4 variables - 16 rows

n -bit Comparator \rightarrow 2n variables - 2^n rows

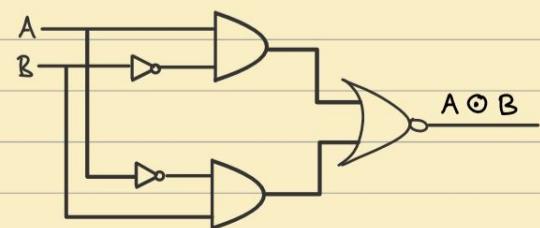
n : no. of bit of each number

Values : 2^{2n}	1 - 4
	2 - 16
	3 - 64

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

$$A \odot B = \overline{A \oplus B}$$

$$\begin{array}{c} A \bar{B} \\ \hline \overline{A \bar{B} + \bar{A} B} \end{array}$$



$$\rightarrow \begin{array}{r} 0 \\ 1 \\ 1 \\ 1 \\ - A \end{array}$$

$$\begin{array}{r} 0 \\ 0 \\ 0 \\ 1 \\ - B \end{array}$$

$$A_3 \ A_2 \ A_1 \ A_0$$

$$0 \ 1 \ 1 \ 1$$

$$B_3 \ B_2 \ B_1 \ B_0$$

$$0 \ 0 \ 0 \ 1$$

$$A_3 = B_3 \Rightarrow A_3 \bar{B}_3$$

Then, go for $A_2 \& B_2$.

Then, go for $A_1 \& B_1$

Lastly, go for $A_0 \& B_0$

1st step $\Rightarrow A = B : A \odot B$

$$A = B \Rightarrow (A_3 \odot B_3) \cdot (A_2 \odot B_2) \cdot (A_1 \odot B_1) \cdot (A_0 \odot B_0)$$

$$(A_3 \oplus B_3) \cdot (\overline{A_2} \oplus B_2) \cdot (\overline{A_1} \oplus B_1) \cdot (\overline{A_0} \oplus B_0)$$

$$\therefore A = B \Rightarrow X_3 \cdot X_2 \cdot X_1 \cdot X_0$$

$$[X = A \oplus B]$$

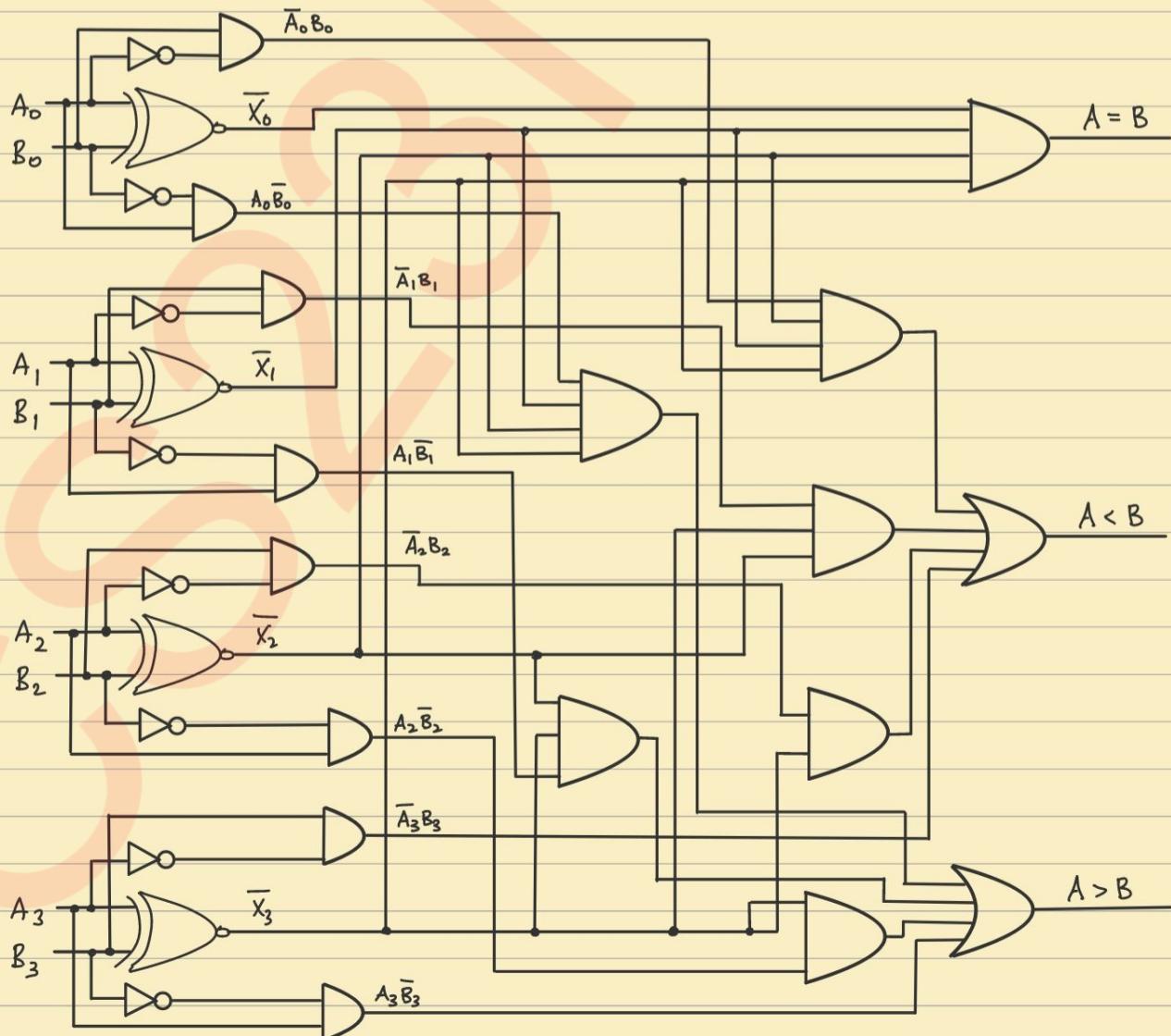
2nd Step $\Rightarrow A > B : A\bar{B}$

$$\therefore A > B \Rightarrow A_3 \bar{B}_3 + \bar{X}_3 \cdot A_2 \bar{B}_2 + \bar{X}_3 \bar{X}_2 \cdot A_1 \bar{B}_1 + \bar{X}_3 \bar{X}_2 \bar{X}_1 \cdot A_0 \bar{B}_0$$

3rd Step $\Rightarrow A < B : \bar{A}\bar{B}$

$$\therefore A < B \Rightarrow \bar{A}_3 B_3 + \bar{X}_3 \cdot \bar{A}_2 B_2 + \bar{X}_3 \bar{X}_2 \cdot \bar{A}_1 B_1 + \bar{X}_3 \bar{X}_2 \bar{X}_1 \cdot \bar{A}_0 B_0$$

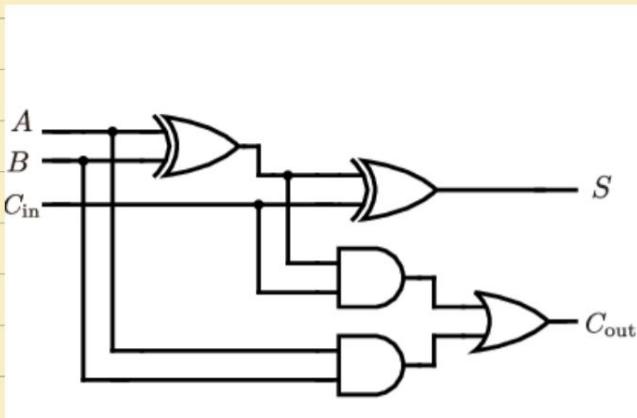
Circuit :



25/9

Lab - 4

Q1) Full - Adder



Inputs			Outputs	
A	B	C_{in}	S	C_{out}
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

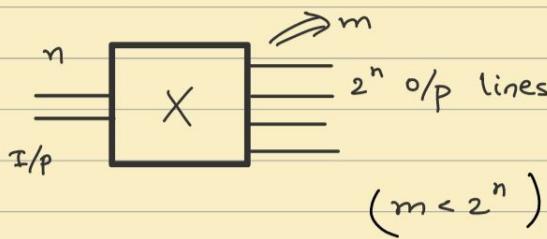
Q2) Full - Subtractor

Symbol		Truth Table				
X	Y	B_{IN}	Y	X	DIFF	B_{OUT}
0	0	0	0	0	0	0
0	0	1	1	1	1	0
0	1	0	1	0	1	1
0	1	1	0	1	0	0
1	0	0	1	0	1	1
1	0	1	0	1	0	0
1	1	0	0	0	0	1
1	1	1	1	1	1	1

P.T.O

27/9

Decoder:

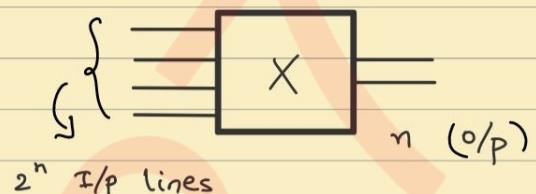


$$1 \quad 2^1 = 2$$

$$2 \quad 2^2 = 4$$

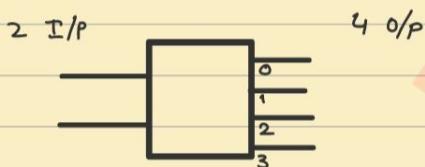
$$3 \quad 2^3 = 8$$

Encoder:



$n:m$ decoder \rightarrow Ex. Naming - 4:16 decoder

(1) 2:4 Decoder:

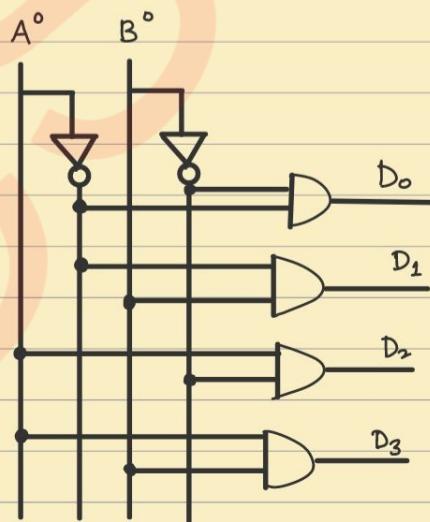


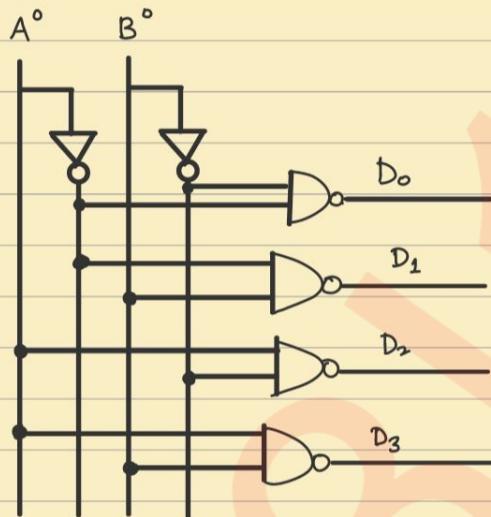
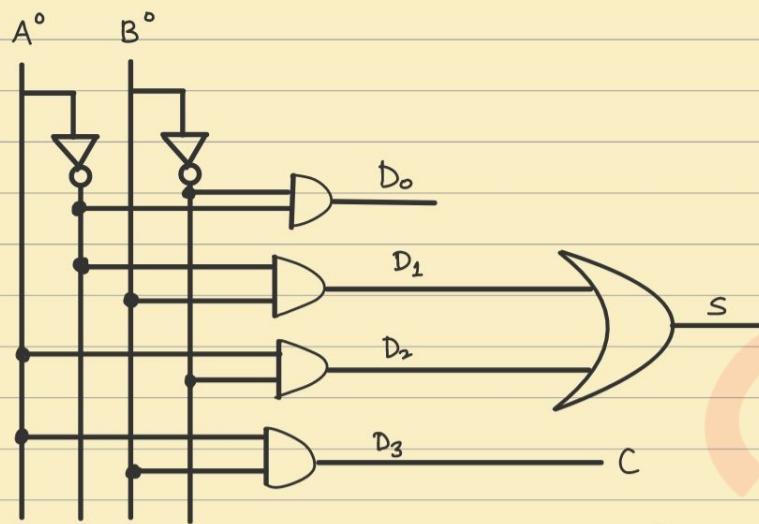
high = 1
low = 0

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

Only one output can be high for an input combination

• 2:4 Decoder:





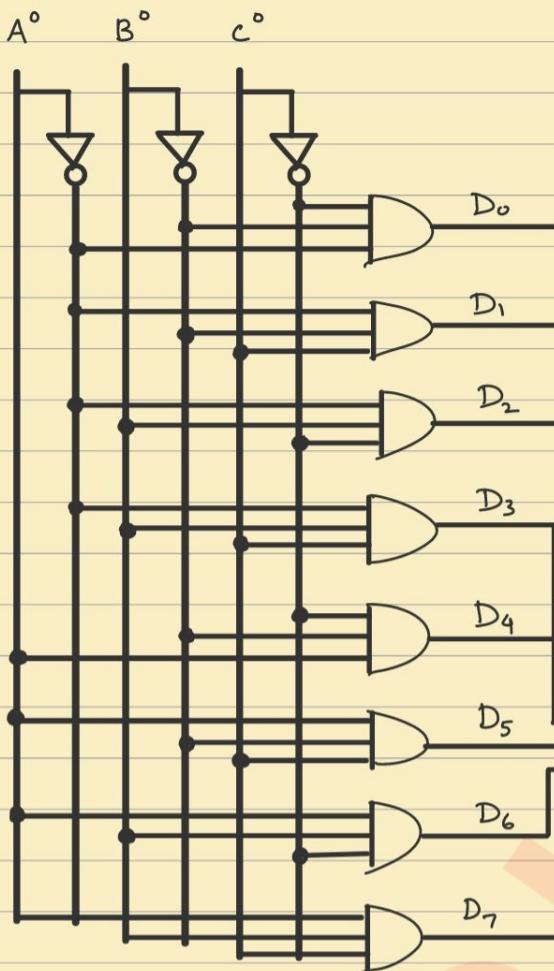
NAND is used instead of
AND :: NAND \rightarrow Cheaper

Let us look at 3:8 Decoder

\hookrightarrow Advantage : Full Adder

Decoder : Each output is attached to a bulb.

P.T.O



$$S(m_1, m_2, m_4, m_7)$$

$$C(m_3, m_5, m_6, m_7)$$

$$S = X \oplus Y \oplus Z$$

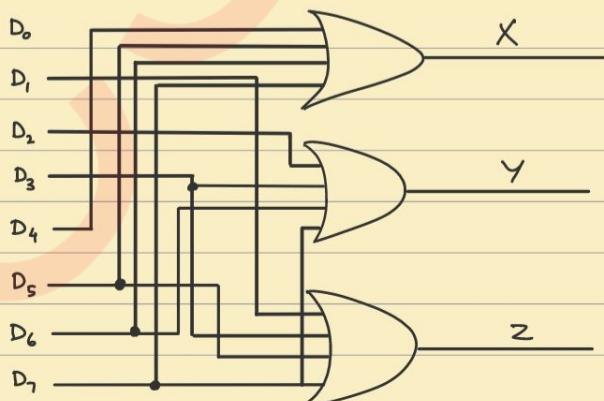
$$\left(\bar{A}\bar{B}C + \bar{A}B\bar{C} + A\bar{B}\bar{C} + ABC \right)$$

Truth table for the sum output S:

X	Y	Z	S
0	0	0	1
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	0

X	Y	Z	S	C
0	0	0	1	1
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	0	0

→ Encoder: [As an Advantage]



$$X : 4, 5, 6$$

$$Y : 2, 3, 6, 7$$

$$Z : 1, 3, 5, 7$$

$$X = D_4 + D_5 + D_6$$

$$Y = D_2 + D_3 + D_6 + D_7$$

$$Z = D_1 + D_3 + D_5 + D_7$$

28/9

Revision

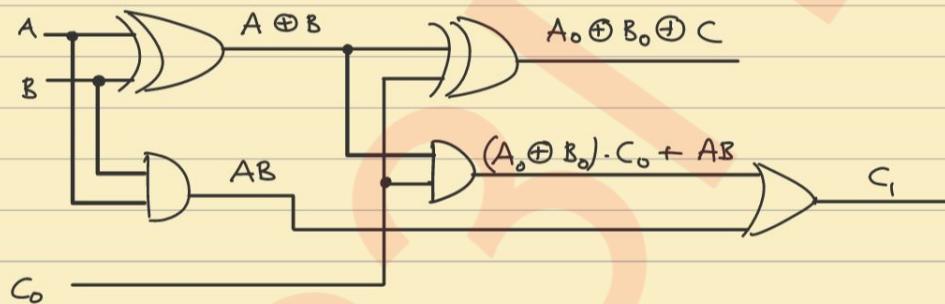
Q) $\sum(m_1, m_2, m_4, m_6, m_8, m_{17}, m_{19}, m_{25}, m_{28}, m_{30}, m_9, m_{11}, m_{23}, m_{15}, m_{29})$

		CDE	000	001	011	010	110	111	101	100
		AB	00	01	11	10	110	111	101	100
00	0	1	1	3	1	2	16	7	5	14
01	1	8	19	1	11	10	14	1	15	13
11	24	125	1	27	26	130	31	129	1	28
10	16	17	19	18	22	23	21	20		

Q)

		CDE	000	001	011	010	110	111	101	100
		AB	00	01	11	2	6	7	5	4
00	0	1	3	2	6	7	5	4		
01	8	9	11	10	14	1	15	13	12	
11	24	25	27	26	30	31	29	28		
10	16	17	19	18	22	23	21	20		

→ 4-bit look ahead carry adder:



$$\begin{aligned} C_2 &= P_1 (P_0 C_0 + G_0) + G_1 \\ &= P_0 P_1 C_0 + P_1 G_0 + G_1 \end{aligned}$$

$$C_1 = P_0 C_0 + G_0$$

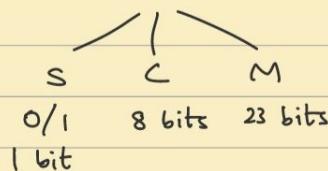
$$C_2 = P_1 C_1 + G_1$$

$$C_3 = P_2 C_2 + G_2$$

$$\begin{aligned} C_3 &= P_2 (P_0 P_1 C_0 + P_1 G_0 + P_1) + G_2 \\ &= P_0 P_1 P_2 C_0 + P_1 P_2 G_0 + P_1 P_2 + G_2 \end{aligned}$$

→ Floating Point number - IEEE 754

→ 32 bit



Ex. 0 10000010 11010000...

$(41680000)_{16}$

~~0|00 0001 0|1101000...~~

s

m

$s = 0 \Rightarrow$ Positive

$$e - D = 130 - 127 = 3$$

power = +3

1. mantissa

$$= 1.11010000... \times 2^3$$

$$= (1110.1)_2$$

$$= \underline{\underline{(14.5)}_{10}}$$

Q) ~~10111110 (101000).....~~ $\leftarrow (BF680000)_{16}$

-ve

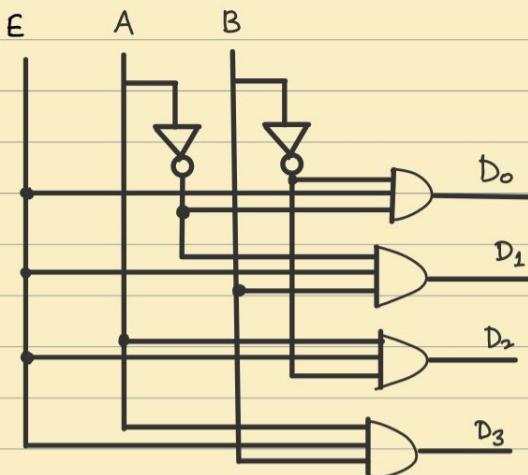
$$126 - 127 = -1$$

$$\Rightarrow -1.1101 \times 2^{-1}$$

$$\Rightarrow -0.11101$$

$$\Rightarrow (-0.90625)_{10}$$

P.T.O



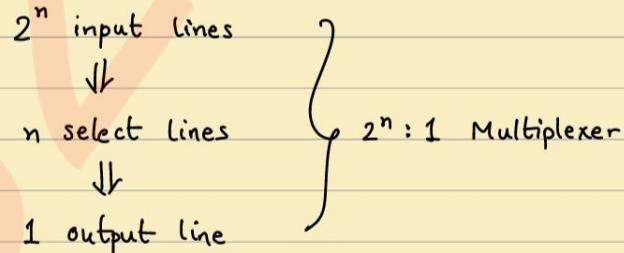
No output will be realised if Enable bit is off.

E	A	B	D ₀	D ₁	D ₂	D ₃
0	0	0	0	0	0	0
0	0	1	0	0	0	0
0	1	0	0	0	0	0
0	1	1	0	0	0	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	0	1

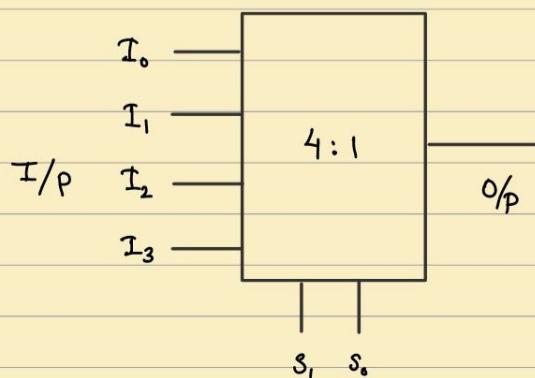
Multiplexer — Variation of this

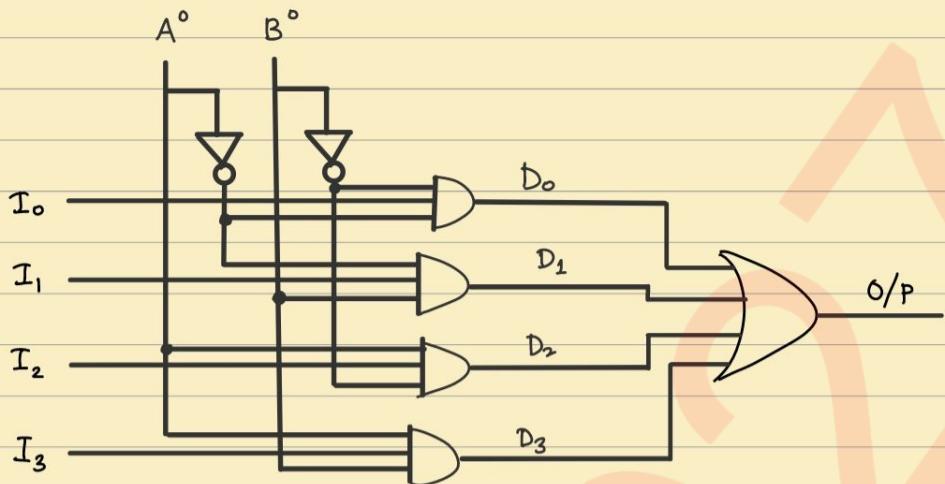
- Decoder — Input is not controlled
- Selecting one among the inputs to be passed into the circuit as the output — Multiplexer
- Multiplexer — similar to Decoder
- Ex. Railway Track — Controlling which train track to switch to
- Restricting certain access to outside world from a network. Multiplexer's logic is implemented inside a network.

- (1) Input lines
- (2) Selection/ Select lines
- (3) Output lines

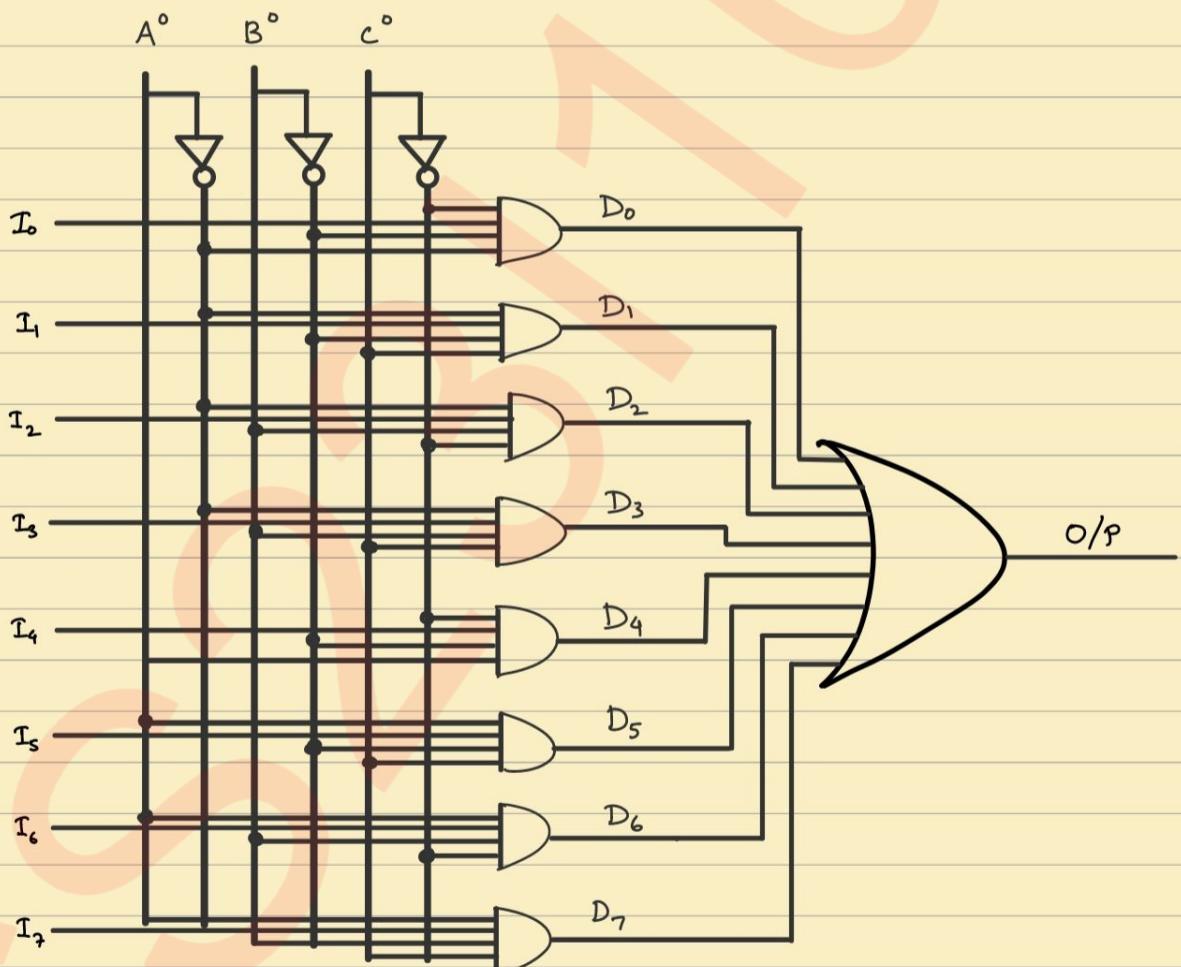


Note : Multiplexer always has only 1 output



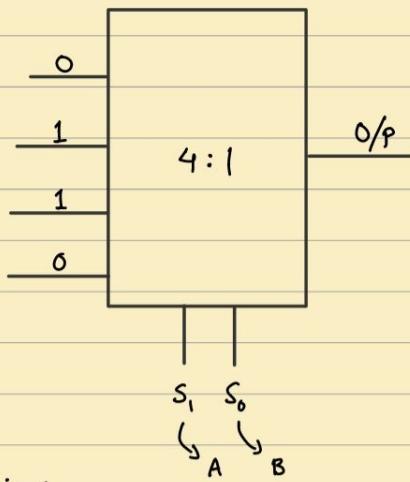


8 : 1 Multiplexer :



Realising a circuit for Half-bit Adder :

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

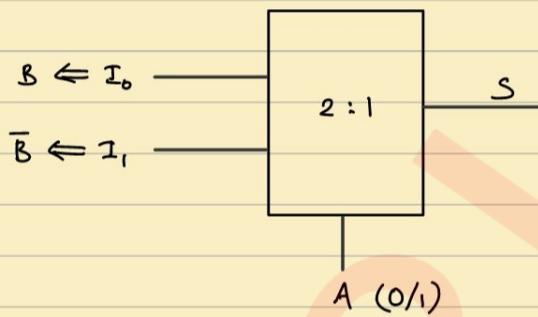


Observation:

$$\text{When } A = 0 \Rightarrow S = B$$

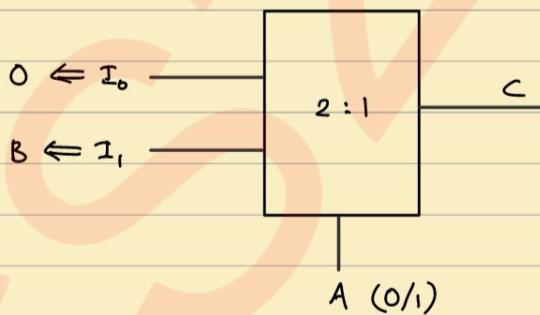
$$\text{When } A = 1 \Rightarrow S = \bar{B}$$

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



i.e. Reducing 4:1 circuit to a 2:1 Multiplexer for 2 variable function
 \therefore Multiplexer is used, Decoder cannot realise this.

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



H.W

Full Adder - Sum & Carry

9/10

$F(A, B, c) \rightarrow$ Best case of Multiplexer : 4×1
 $\because 2^3$

1 variable less $\Rightarrow 2^2 = 4$

A	B	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



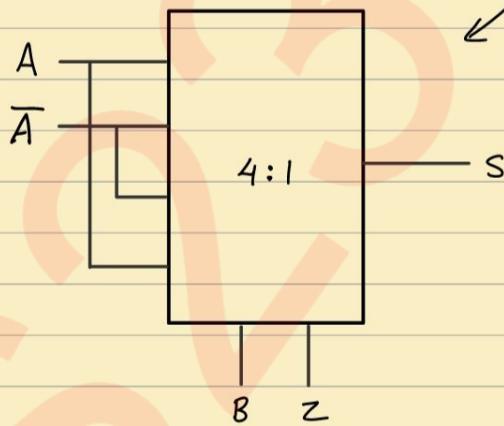
A : MSB

Z : LSB

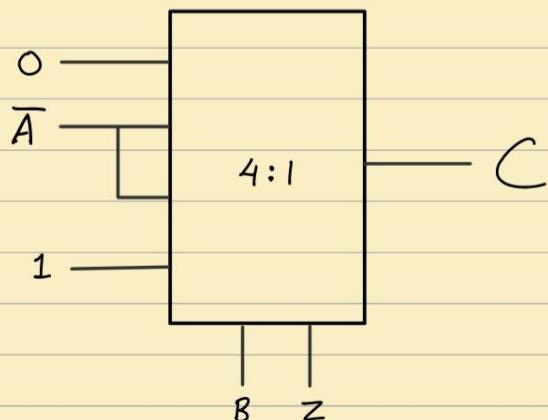
In form of a K-map.

		BZ				(S)
		00	01	10	11	
(A)	0	0 ₀	1 ₁	1 ₂	0 ₃	
	1	1 ₄	0 ₅	0 ₆	1 ₇	
		A	\bar{A}	\bar{A}	A	

wherever 1 is there,
write corresponding
to that (for 0-1 case)



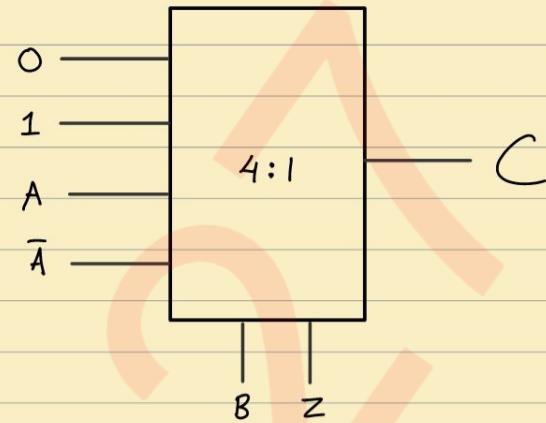
		BZ				(C)
		00	01	10	11	
(A)	0	0 ₀	0 ₁	0 ₂	1 ₃	
	1	0 ₄	1 ₅	1 ₆	1 ₇	
		0	A	A	1	



Note :
$$\begin{cases} 1 \& 1 = 1 \\ 0 \& 0 = 0 \end{cases}$$

Q) $\Sigma(1, 3, 5, 6)$ MSB

		C				
		BZ	00	01	10	11
A	0	0	1	0	1	
	1	0	1	1	0	1



→ Priority encoder → Gives priority to an input when multiple inputs are given.

Highest number, i.e. in this case, to I_3

	I_0	I_1	I_2	I_3	A	B	V
1	1	0	0	0	0	0	1
2	X	1	0	0	0	1	1
4	X	X	1	0	1	0	1
8	X	X	X	1	1	1	1
	0	0	0	0	X	X	0

$$A = I_2 + I_3 \times$$

I_0 and I_1 does not come in output ∵ Don't cares.

$$A = I_2 \cdot \bar{I}_3 + I_3$$

Similarly,

$$B = I_1 \bar{I}_2 \bar{I}_3 + I_3$$

$$V = I_0 + I_1 \bar{I}_2 \bar{I}_3 + I_3$$

→ Valid bit, Checks if output is valid for 0 values.

→ POS:

A	B	F
0	0	0
0	1	1
1	0	1
1	1	1

$$\begin{aligned}
 F(A, B) &= A\bar{B} + \bar{A}B + AB \\
 &= B(A + \bar{A}) + A\bar{B} \\
 &= B + A\bar{B} \\
 &= B + A
 \end{aligned}$$

Product of Sum - POS

↪ i.e. Max terms - Perspective

Max term - Opposite to min terms

i.e. Check for Zeros

Basic Operation : Sum

Then, Product of sums

$$0 \rightarrow A$$

$$1 \rightarrow \bar{A}$$

Check for $F(A, B) = 0$

$$\underline{(A + B)} \longrightarrow \text{Pos}$$

To convert into truth Table, $A \rightarrow 0 \& B \rightarrow 0$

While realising the expression, Write it as it is

Ex.

A	B	F
0	0	0
0	1	1
1	0	1
1	1	0

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

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

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

P.T.O

→ Hardware:

- (1) 4 bit BCD adder using Two 4-bit full adders
- (2) 2 bit Adder / Subtractor

(1)

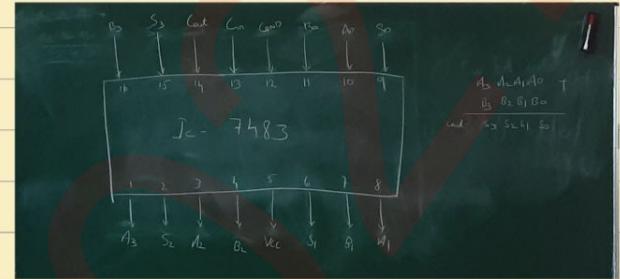
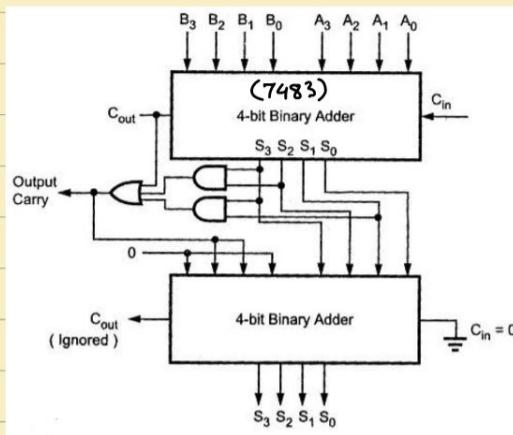
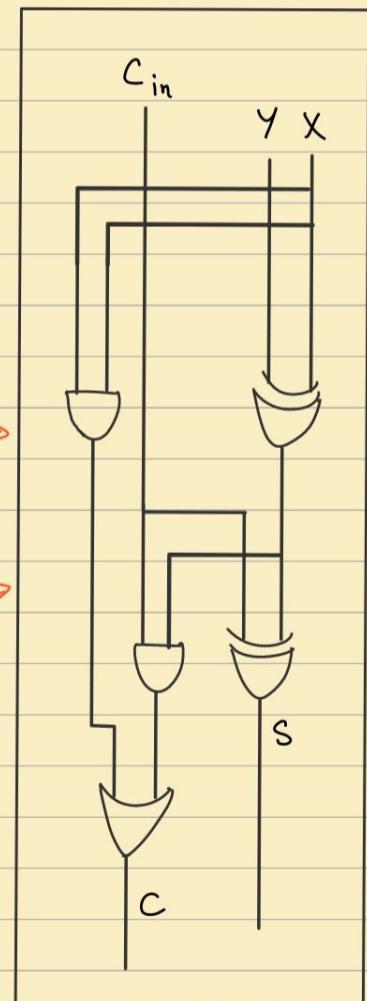
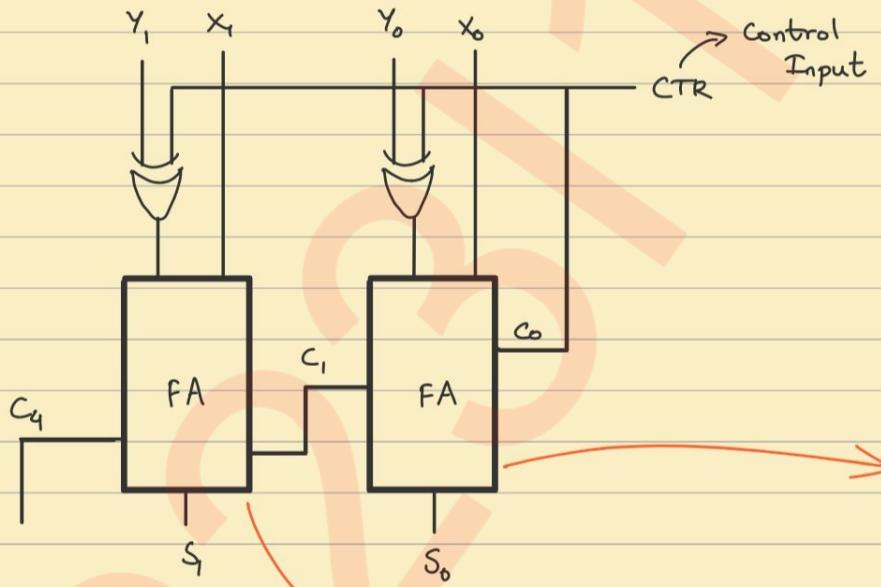


Fig. 3.32 Block diagram of BCD adder

(2)



10/10

A	B	F
0	0	0
0	1	1
1	0	1
1	1	0

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

$$POS : A + B$$

$$\overline{\overline{F}} = F$$

$$\begin{aligned} \overline{(\bar{A}\bar{B} + \bar{A}\bar{B} + AB)} &= \overline{(\bar{A}\bar{B} \cdot A\bar{B} \cdot AB)} \\ &= \overline{AB} \\ &= A + B \end{aligned}$$

Ex.

A	B	C	F
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	0

		BC		AB		
		00	01	11	10	
		0	0 ₈	1 ₁	0 ₃	1 ₂
		1	0 ₄	1 ₅	0 ₇	1 ₆

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

Application of max-terms method :

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

$$\rightarrow \text{Maxterm : } \bar{A} + \bar{B} + C + \bar{D}$$

i.e. Convenient to represent

Ex. Solve using SOP method :

$$F(A, B, C, D) = \overline{I}\overline{L}(13)$$

Where 0 is present

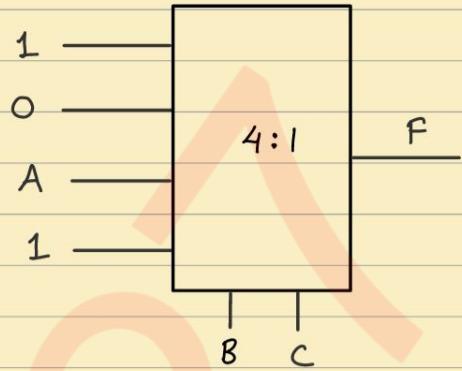
Ex. Design a multiplexer

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

Sol: $F(A, B, C) = \Sigma(0, 2, 4, 6, 7)$

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

1 0 A 1



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

High $\rightarrow 1$
Low $\rightarrow 0$

But we treat $(\bar{A} \rightarrow 1)$
 $(A \rightarrow 0)$
just temporarily

(1) 001 - $A + B + \bar{C}$

(3) 011 - $A + \bar{B} + \bar{C}$

(5) 101 - $\bar{A} + B + \bar{C}$

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

Note:

$$\Pi = \Sigma \quad (\text{Negation})$$

→ Group of circuits - Combinational circuits
 ↳ set of gates

Values are just being passed.

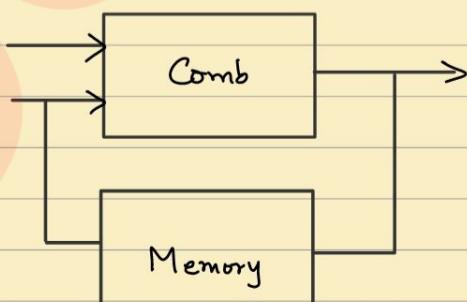
Next time, if new value is given, the old value is not retained.

Solution: Storage

 ↳ In Memory

i.e. Sequential circuit

 ↳ Store the data and give it back to the circuit.
 ↳ Biggest Invention



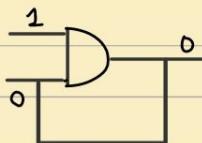
★ Flip Flop : Stores single bit

 ↳ Stores particular state

i.e. 0 or 1.

↳ Basic circuit : Latch

4 types



Looping

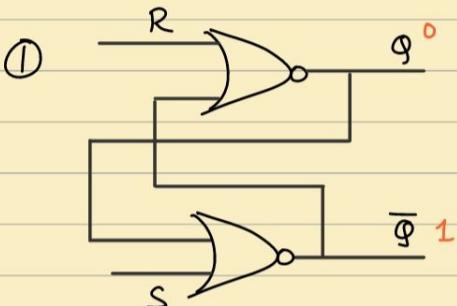
Asynchronous — Doesn't depend on time

Synchronous — Able to relate (In same state)

Time → Clock
Discrete Time events

14/10

→ RS Latch:



Previous State : $Q(t)$

Next State : $Q(t+1)$

Assuming $Q(t) = 0$ & $\bar{Q}(t) = 1$

[Trace it for minimum 1.5 cycles for result.]

R	S	$Q(t+1)$	$\bar{Q}(t+1)$
0	1	1	0 ↗ → Set
1	0	0	1 ↗ → Reset
0	0	0	1 ↗ → Retain
1	1	0	0 ↗ → Indeterminate case

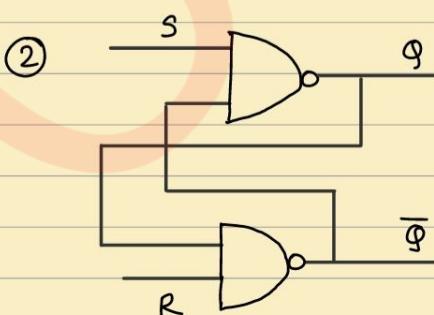
Assume $Q(t) = 1$ & $\bar{Q}(t) = 0$

R	S	$Q(t+1)$	$\bar{Q}(t+1)$
0	1	1	0 ↗ → Set
1	0	0	1 ↗ → Reset
0	0	1	0 ↗ → Retain
1	1	0	0 ↗ → Indeterminate case

R : Reset

S : Set

Flipflop → RS



SR latch

Assuming $Q(t) = 0$ & $\bar{Q}(t) = 1$

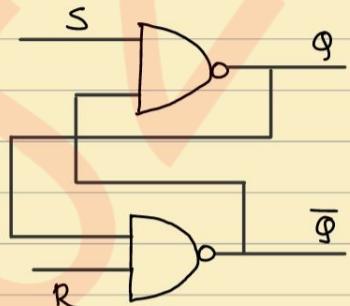
R	S	$Q(t+1)$	$\bar{Q}(t+1)$
0	1	0	1 ↴ → Reset
1	0	1	0 ↴ → Set
0	0	1	1 ↴ → Indeterminate
1	1	0	1 ↴ → Retain

Set : Making the output as 1 irrespective of previous value.

Reset : Making the output as 0 irrespective of previous value.

Assume $Q(t) = 1$ & $\bar{Q}(t) = 0$

R	S	$Q(t+1)$	$\bar{Q}(t+1)$
0	1	0	1 ↴ → Reset
1	0	1	0 ↴ → Set
0	0	1	1 ↴ → Indeterminate case
1	1	1	0 ↴ → Retain

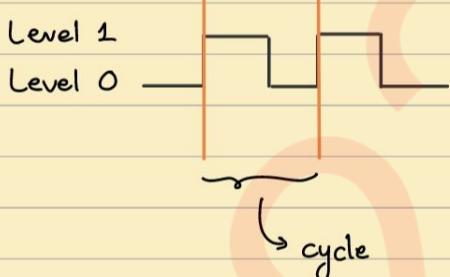


15/10

Clock:

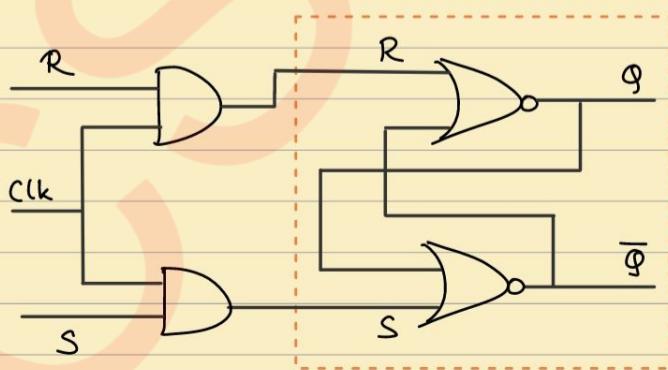
Speed : frequency \rightarrow 0 1 : Alternatively

Signal



Pulse / Band signal

• RS Flipflop



Clk \rightarrow Clock

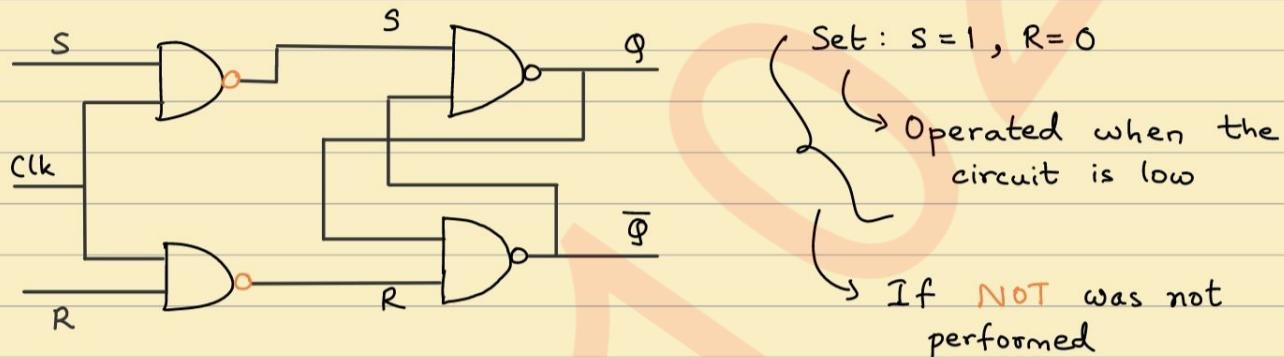
+ve clock cycle = 1
-ve clock cycle = 0

Clock = 0 \Rightarrow Output is retained

Clock = 1 \Rightarrow RS latch depending on R & S

R	S	$Q(t+i)$	$\bar{Q}(t+i)$
0	0	0	1 → Retain
0	1	1	0 → Set
1	0	0	1 → Reset
1	1	0	0 → Indeterminate case

SR flip flop:



S	R	$Q(t+i)$	$\bar{Q}(t+i)$
1	0	1	0 → Set
0	1	0	1 → Reset
0	0	1	0 → Retain
1	1	0	0 → Indeterminate case

Clock = 0 \Rightarrow Output is retained

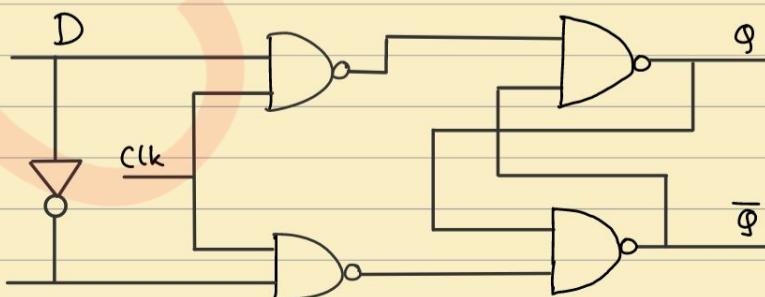
Clock = 1 \Rightarrow SR latch depending on S & R

For not retaining, We use D flip-flop.

D : Data

Connect S to R using
NOT gate

We cannot give clock input &
retain the value.



A Pulse based flip flop
Cannot retain beyond
one clock cycle

Disadvantage : Cannot retain more than 1 cycle

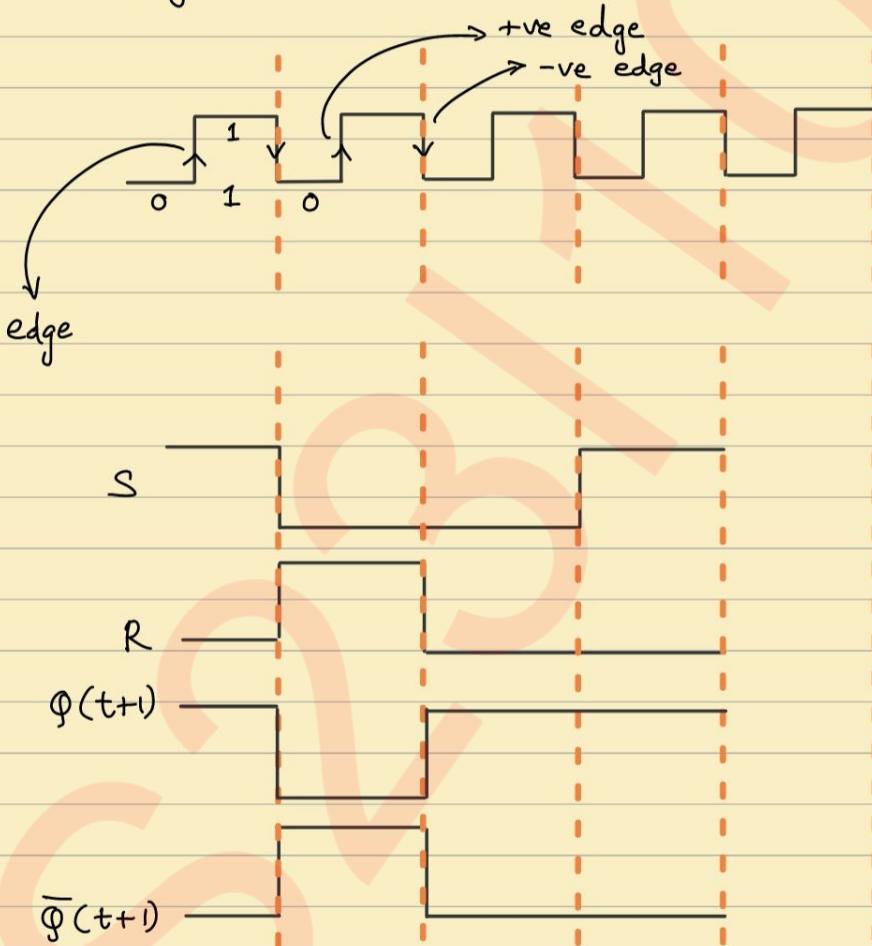
Advantage : No Indeterminate state

∴ Operation of D Flipflop is proper.

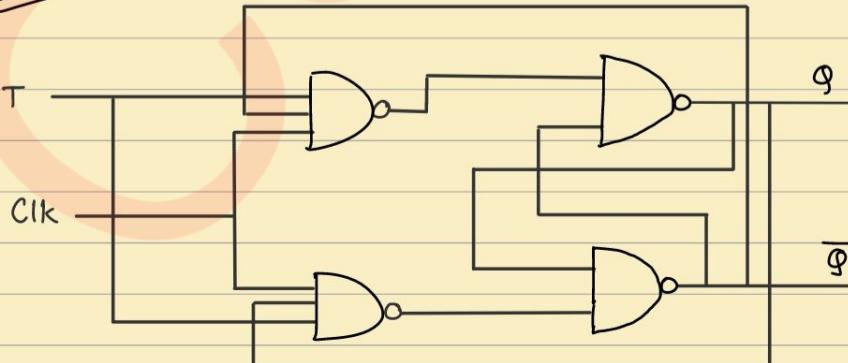
D	$\varphi(t+1)$	$\bar{\varphi}(t+1)$	Characteristic Table (AKA Truth Table)
0	0	1	
1	1	0	
X	φ_t	$\bar{\varphi}(t+1)$	

When Clock = 0 , X

Time Diagram:



18/10



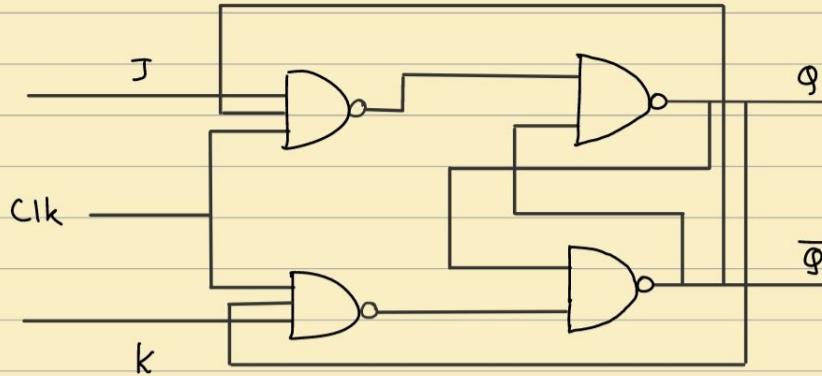
← T - flip flop

Avoids indeterminate state

T : Toggle

Flipping the bit
($1 \rightarrow 0$ & $0 \rightarrow 1$)

- JK flip flop :

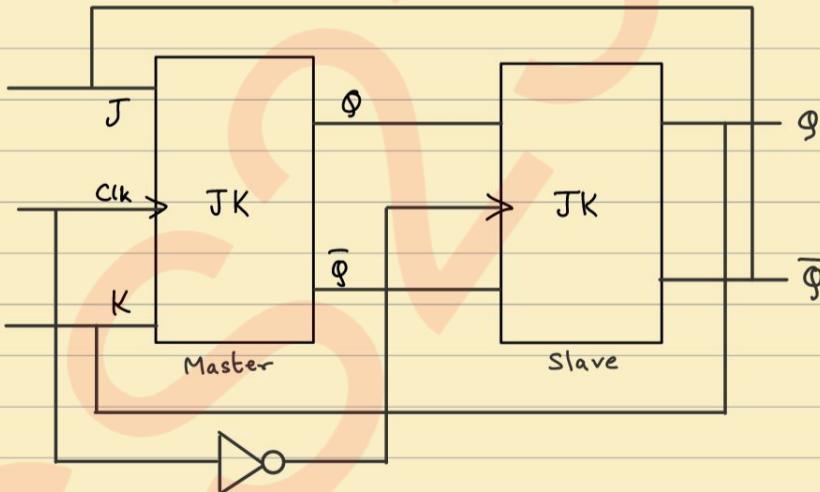


φ	J	K	$Q(t+1)$	
0	0	0	0	Retain
0	0	1	0	Reset
0	1	0	1	Set
0	1	1	1	Toggle
1	0	0	1	Retain
1	0	1	0	Reset
1	1	0	1	Set
1	1	1	0	Toggle

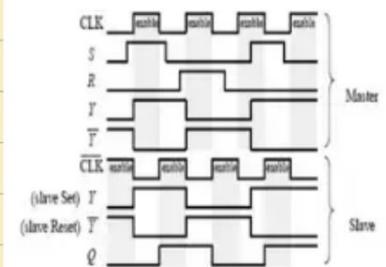
φ	JK	00	01	11	10
0	0	0	1	1	1
1	1	1	0	0	0

$$Q(t+1) = \overline{\varphi}J + \varphi\bar{K}$$

→ Master Slave Flip flop :



PRE	CIR	CLK	S	R	Q	Q	Mode
0	1	X	X	X	1	0	preset
1	0	X	X	X	0	1	cleared
0	0	X	X	X	1	1	not used (race)
1	1	Δ	0	0	0	0	hold
1	1	Δ	0	1	0	1	Reset
1	1	Δ	1	0	1	0	Set
1	1	Δ	1	1	1	1	not used (race)

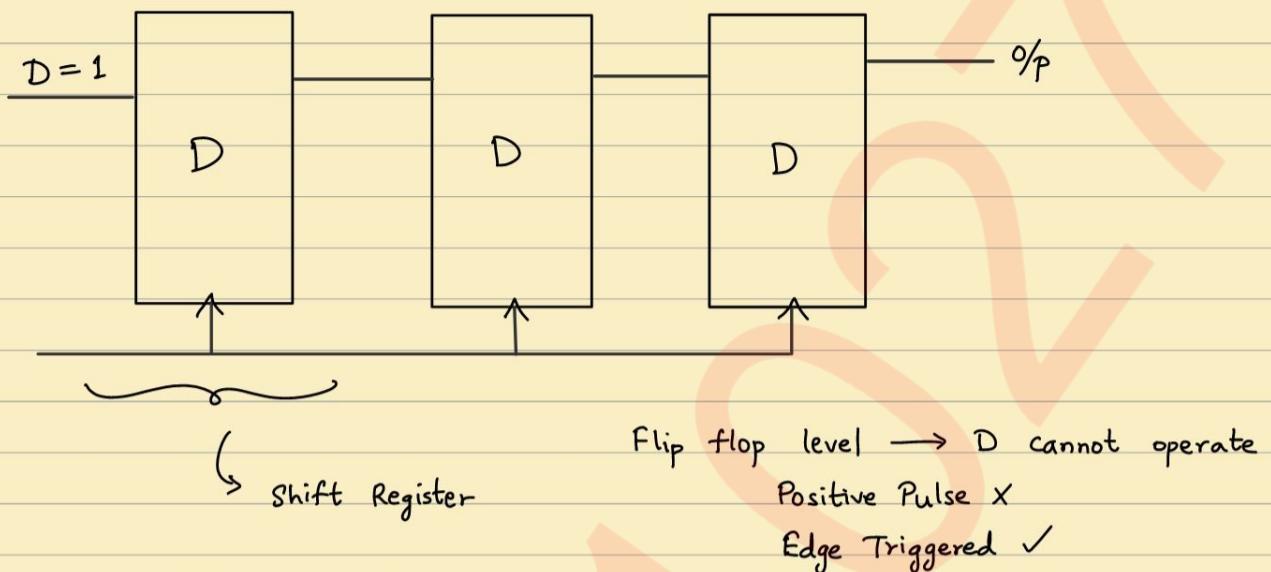


- T-flip flop :

$Q(t)$	T	$Q(t+1)$	
0	0	0	Retain
0	1	1	Toggle
1	0	1	Retain
1	1	0	Toggle

$$Q(t+1) = Q(t) \oplus T$$

→ Edge triggered Flip flop:



When edge = 1, First clock cycle
Gets enabled.

Edge becomes pulse

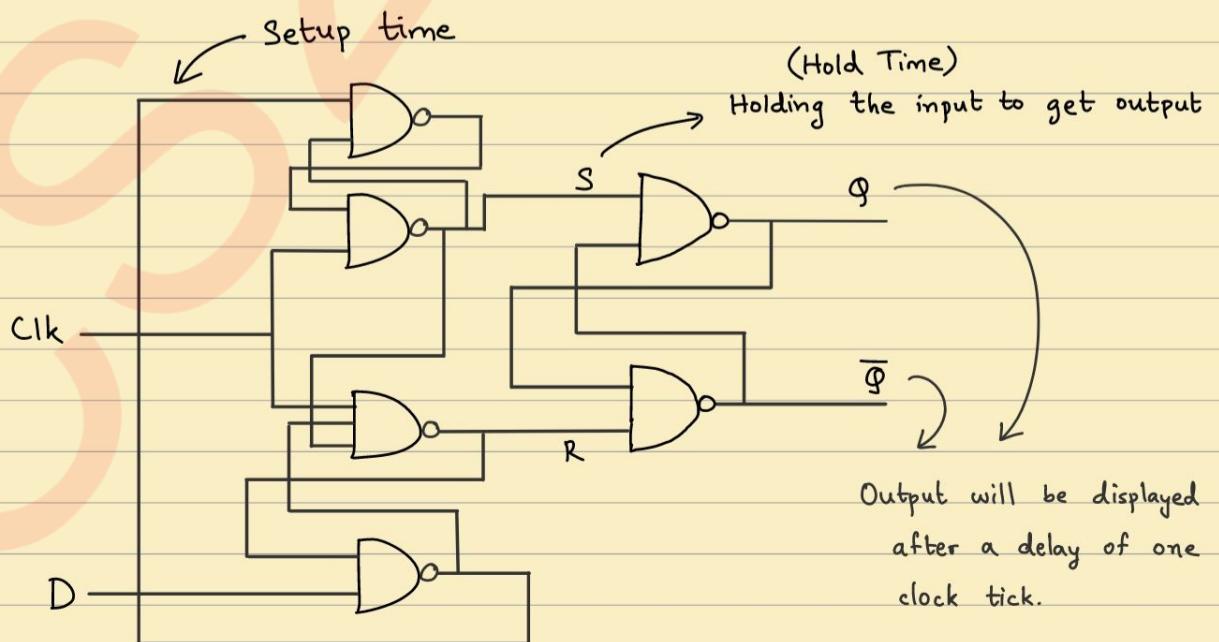
Pulse cannot change input :: Edge based Flip-flop

Negative edge cycle — Nothing happens.

D connected in series — All occur in same cycle

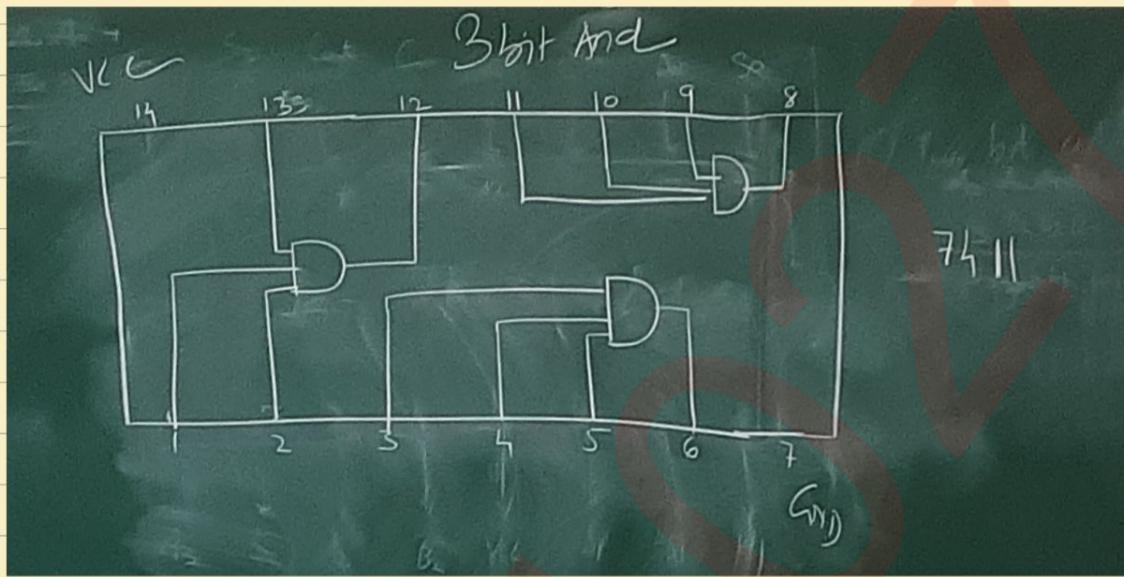
3 cycles to reach the end.

i.e. n d flip flops take n cycles to reach the end.

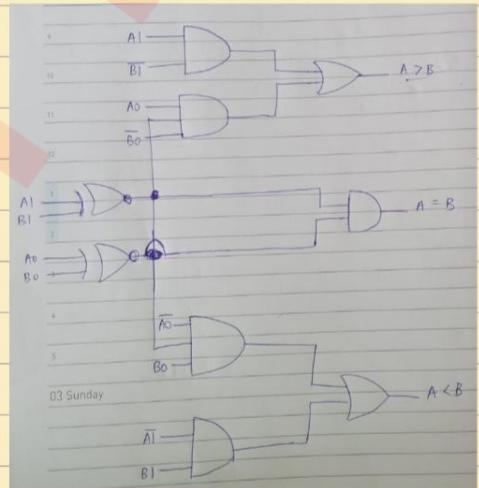


19/10

Lab - 6



→ Make a comparator circuit (2 bit)



P.T.O

21/10

Mid - Sem

Hardware :

Input : I_1, I_0 (2 bit I/P) and Select bit S_0 , to select b/w below operations :

If $S_0 = 0$:

Add with ((Last 4 digits of your roll number) % 4) to I_1, I_0

Else :

Return 2^s complement of I_1, I_0 as output

My Roll Number : CS23I1027

$$27 \% 4 = 3$$

Solution :

Method (I) - Hard Coding :

S_0	I_1	I_0	C	R_1	R_0
0	0	0	0	1	1
0	0	1	1	0	0
0	1	0	1	0	1
0	1	1	1	1	0
1	0	0	X	0	0
1	0	1	0	1	1
1	1	0	0	1	0
1	1	1	0	0	1

$I + 3$

2^s complement

S_0	I_1, I_0	C
0	00, 01, 11, 10	
1	X4, 05, 07, 06	

S_0	I_1, I_0	R_1
0	10, 01, 13, 02	
1	04, 15, 07, 16	

S_0	I_1, I_0	R_0
0	10, 01, 03, 12	
1	04, 15, 17, 06	

$$X_4 = 0$$

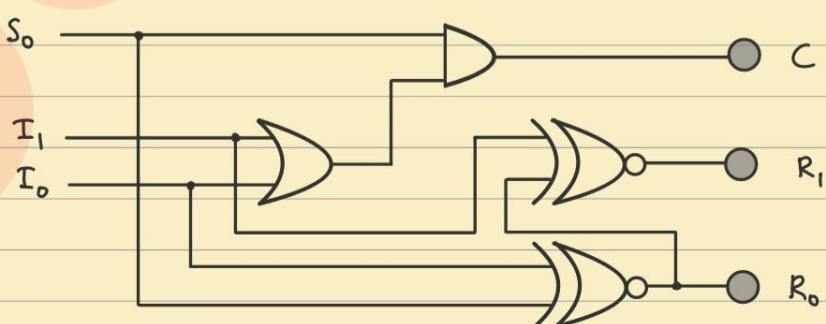
$$C = S_0(I_0 + I_1)$$

$$R_1 = S_0 \odot I_1 \odot I_0$$

$$R_0 = S_0 I_0 + \bar{S}_0 \bar{I}_0$$

$$R_0 = S_0 \odot I_0$$

Circuit Diagram :

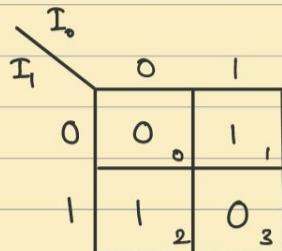


Method (II) :

I_1	I_0	I_1^c	I_0^c
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	1

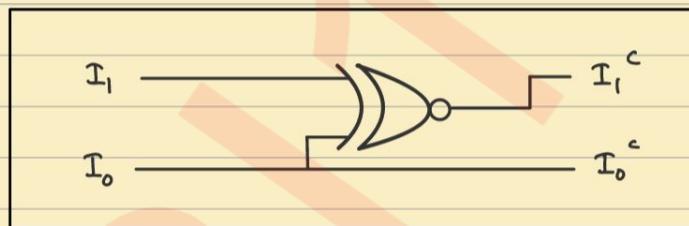
$$[C = 0]$$

$$I_0^c = I_0$$

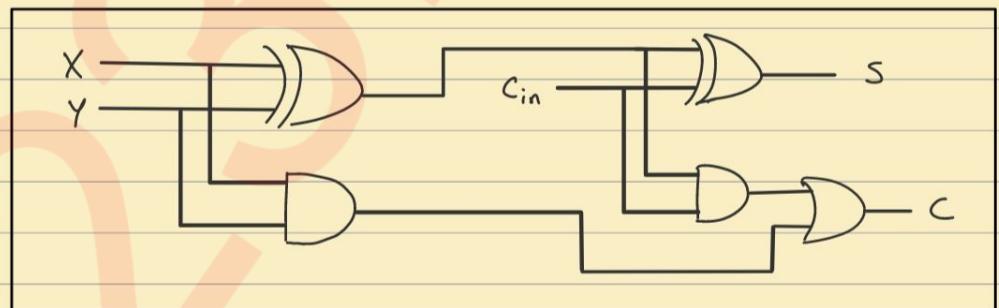


$$I_1^c = I_1 \oplus I_0$$

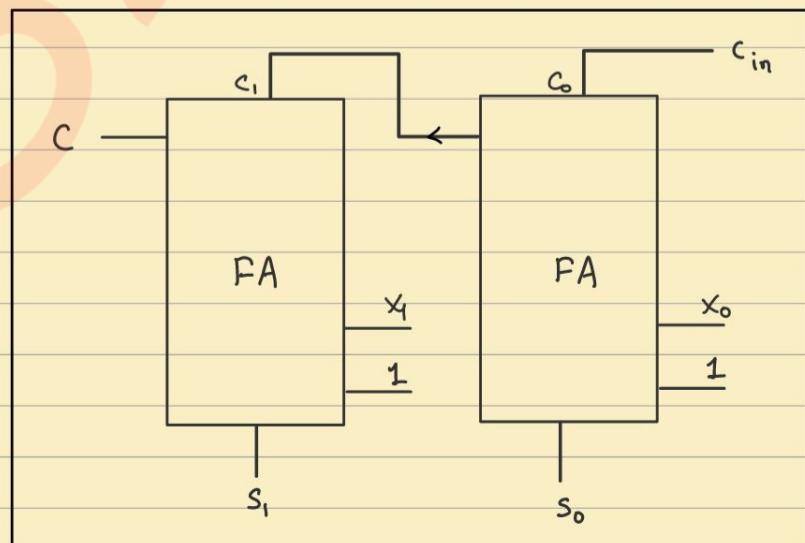
TwoC Module :



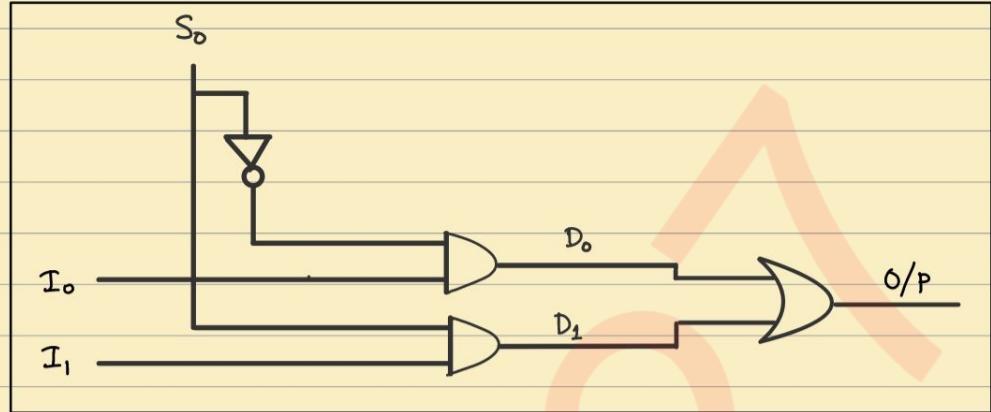
FA module :



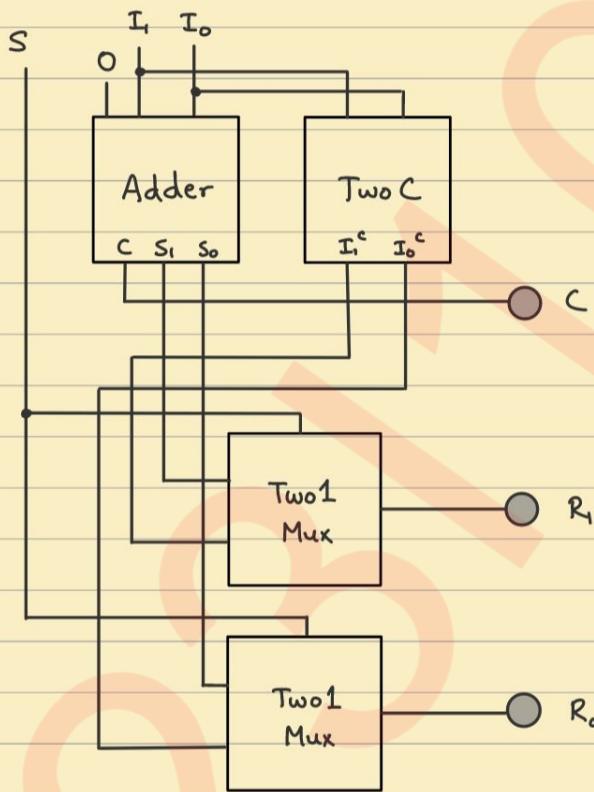
Adder Module :



Two1 Mux Module :



Main :



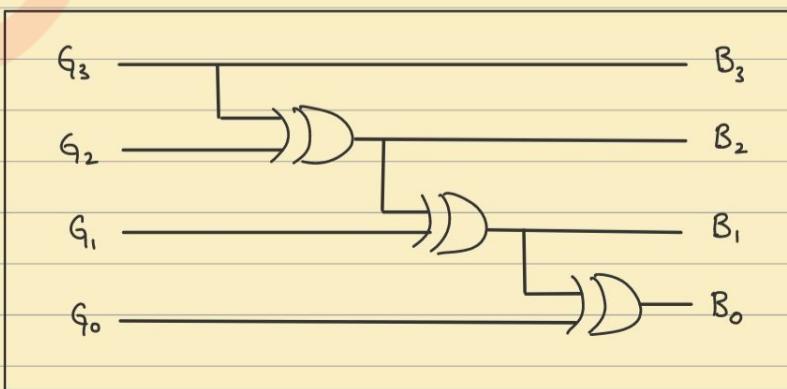
Software:

Question : Two 4-bit Grey codes, Convert them into Binary code and add them.

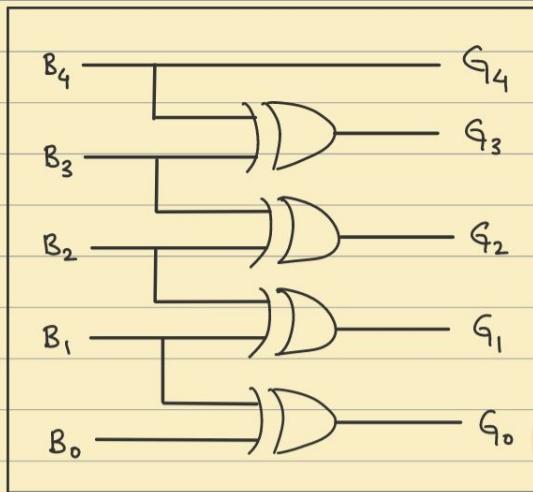
Then, convert the result back into Grey code and display as output.

Solution :

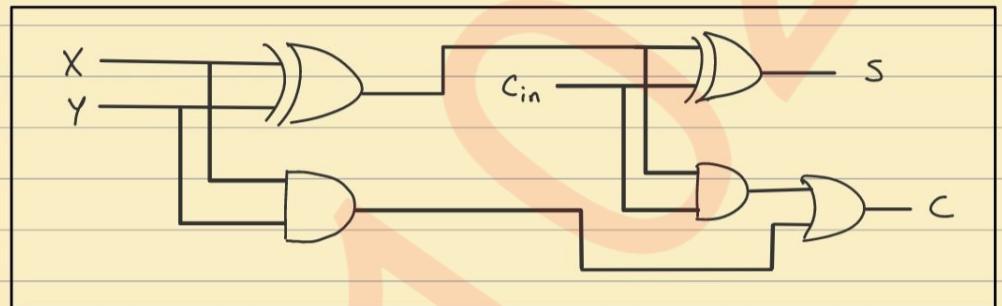
G2B Module :



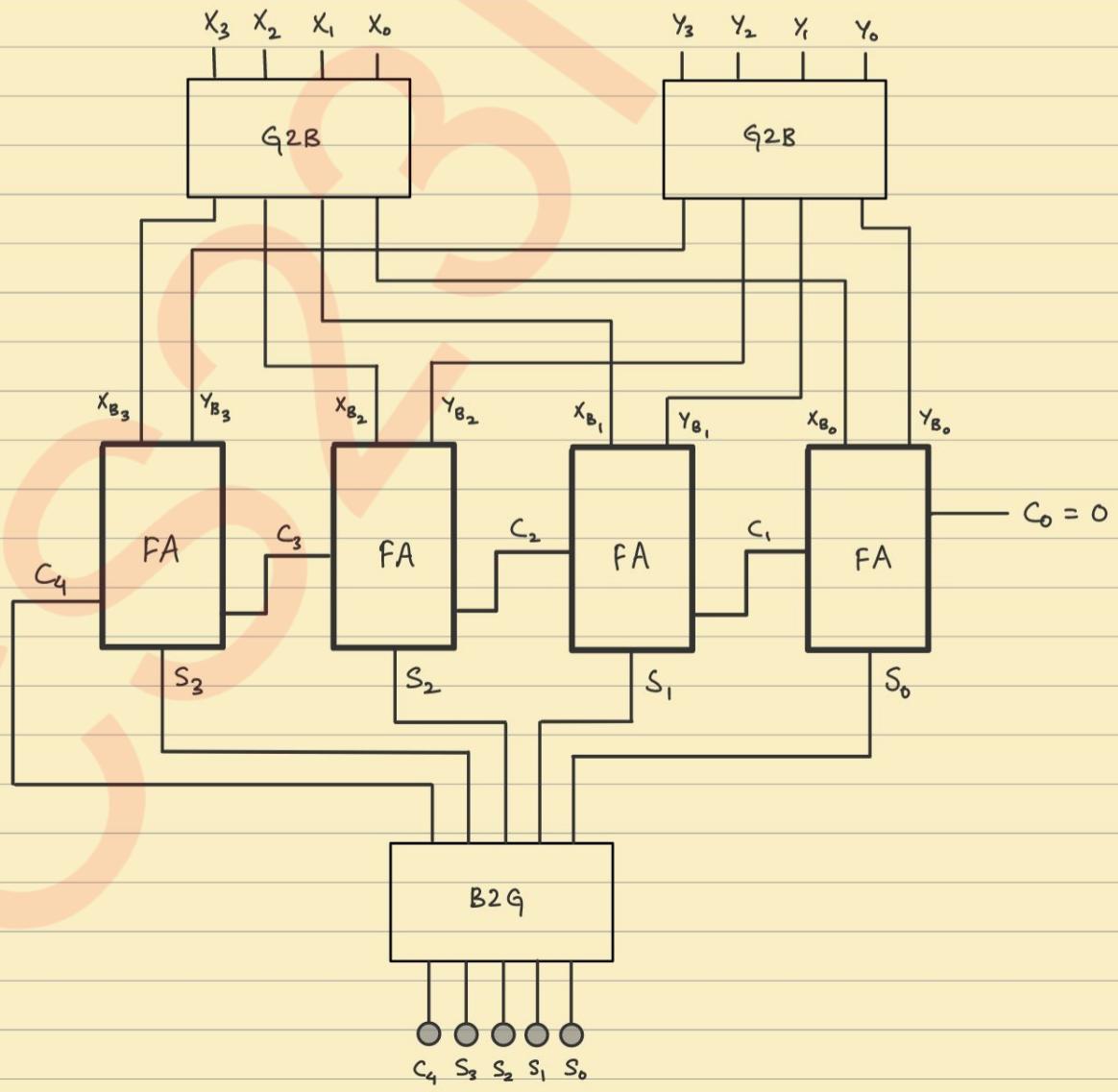
B2G Module :



FA module :

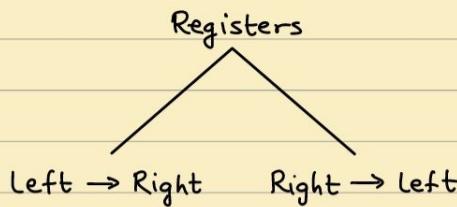


Main :



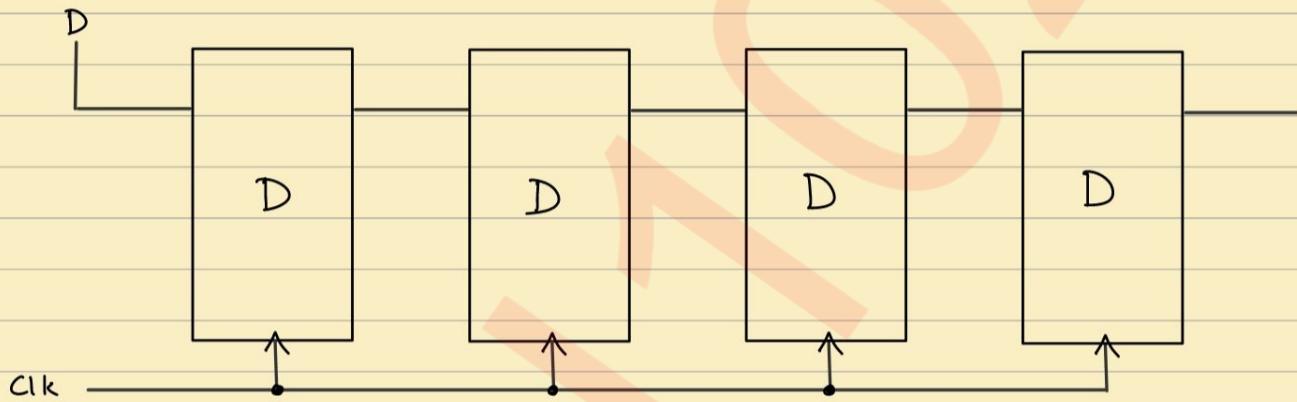
22/10

→ Serial Converters:

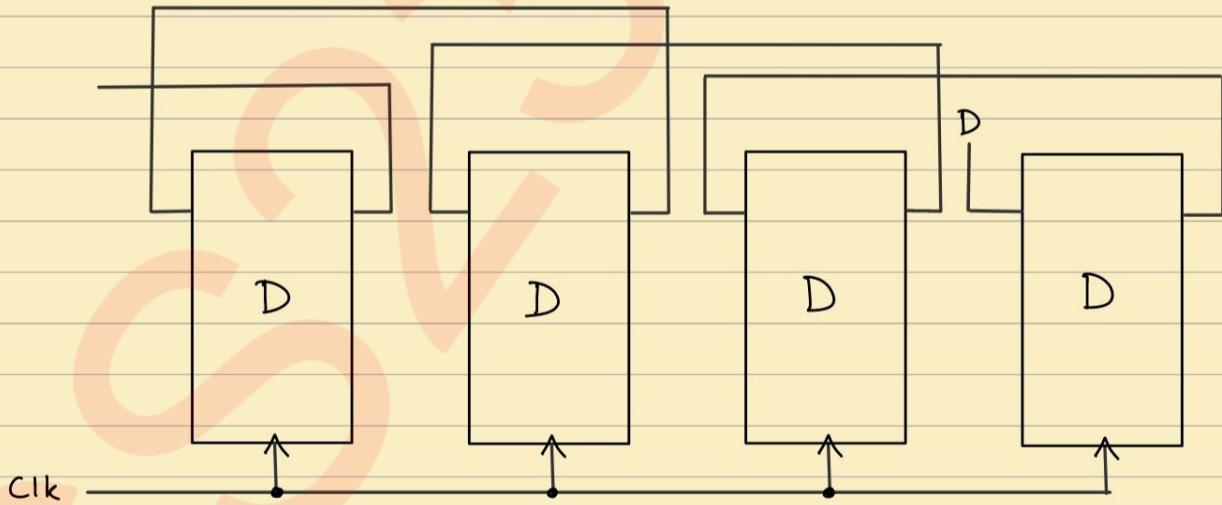


Edge Triggered - If D will take 4 cycles

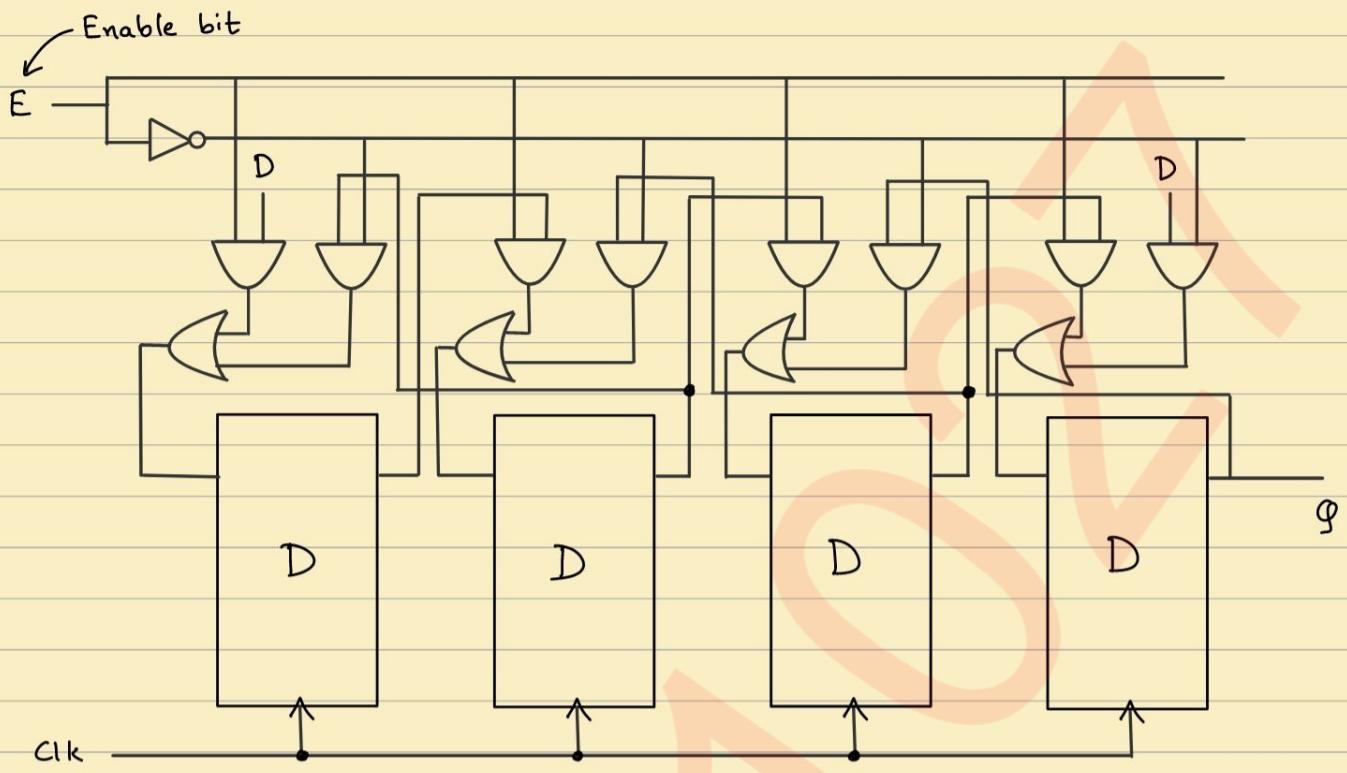
- Right Shift Register:



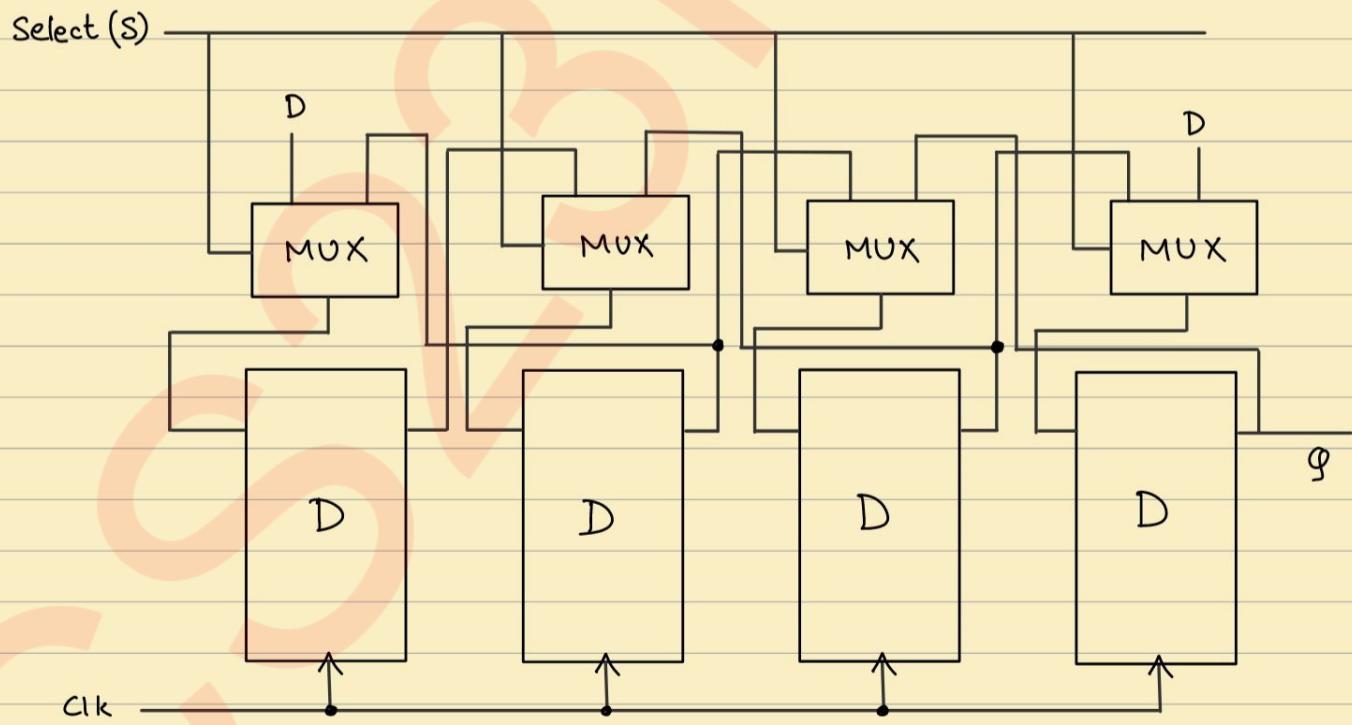
- Left Shift Register:



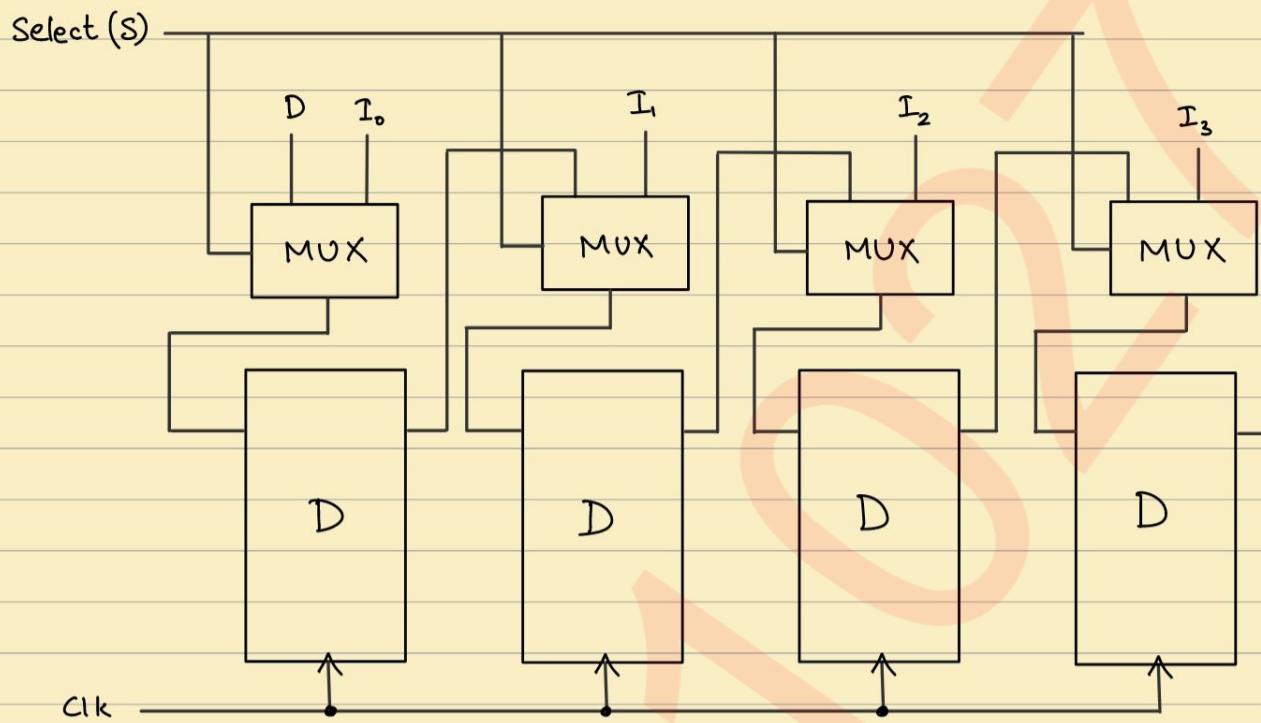
23/10



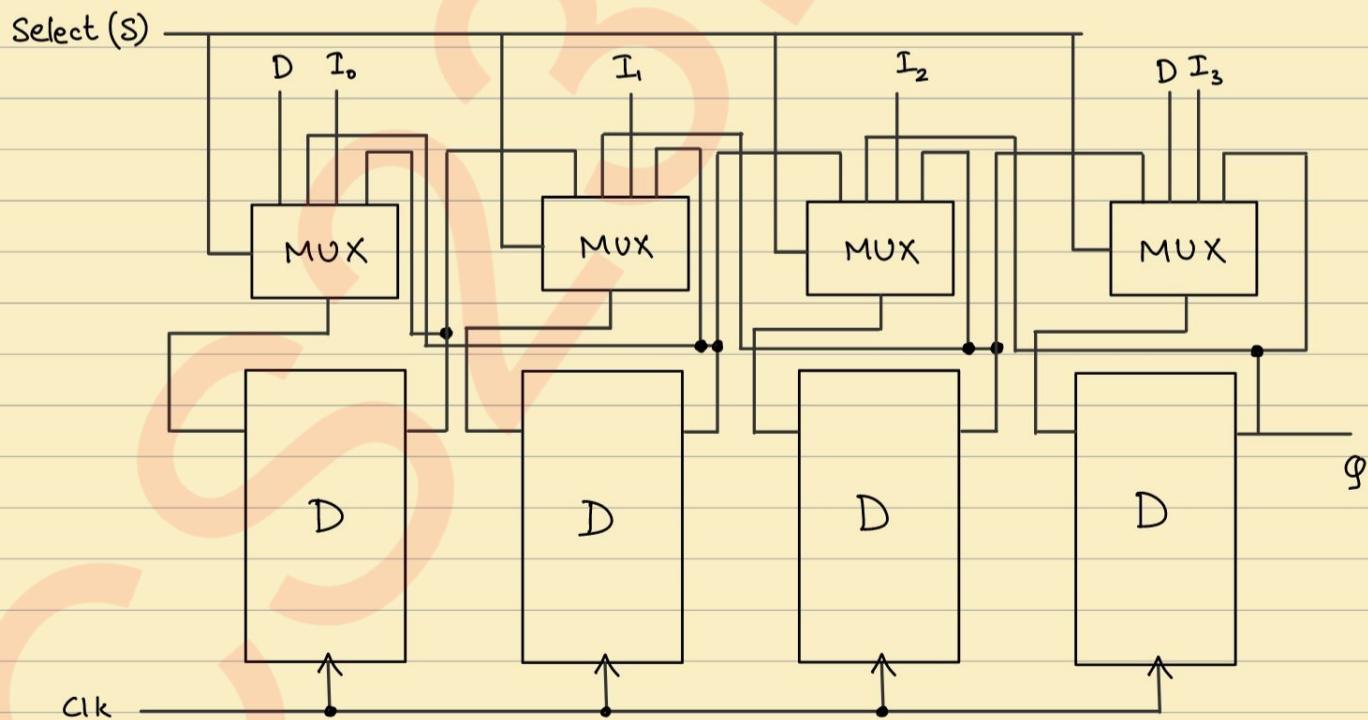
→ Parallel to Serial converter :



→ Parallel to Serial converter & Right Shift:



→ Universal Shift Register:

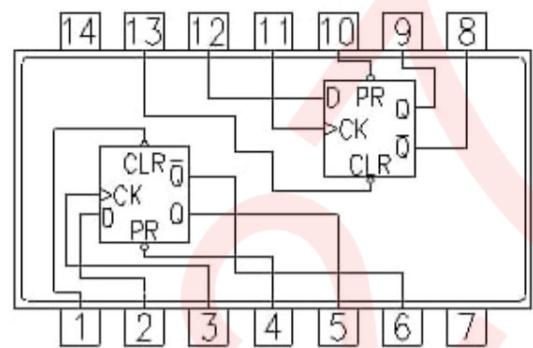
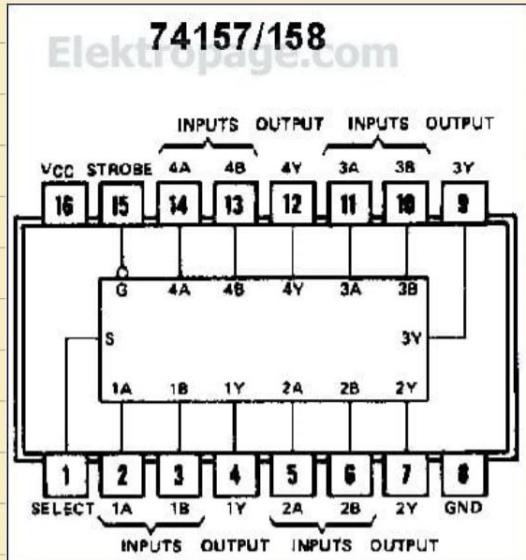


Right Shift : 00

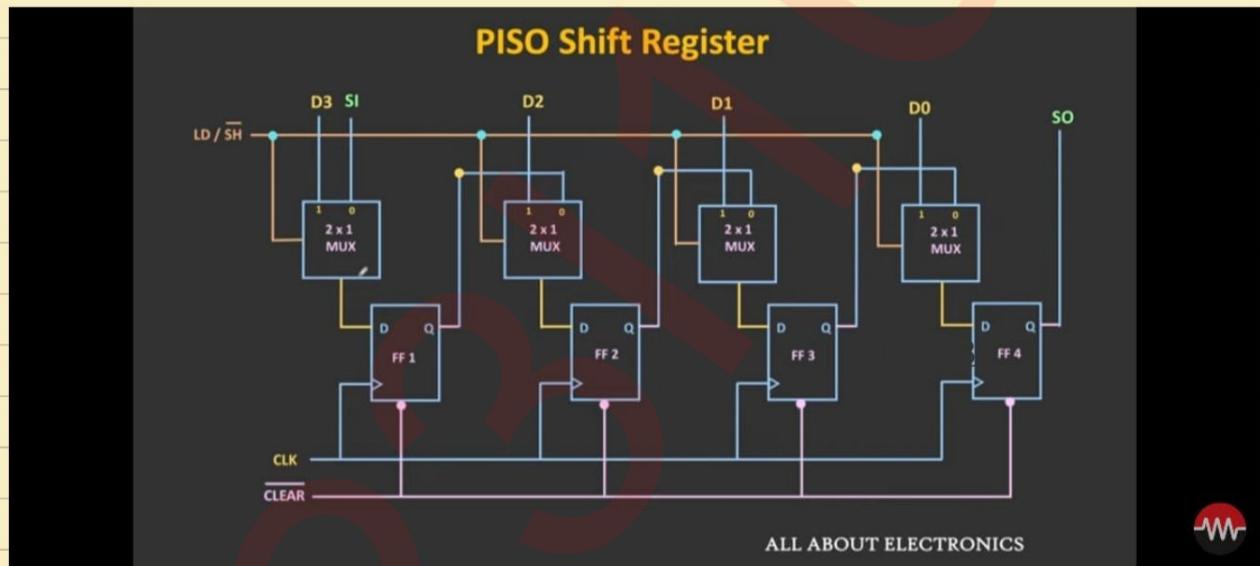
Left Shift : 01

Parallel Input : 11

Retain (Hold) : 10

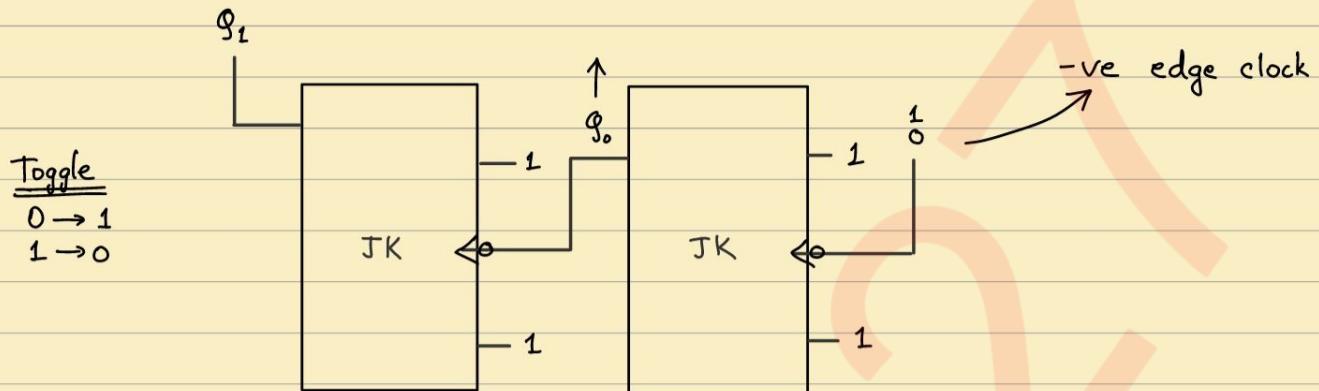


7474

Dual D Flip-Flop
with Preset and Clear

25/10

→ Asynchronous modulo-4 counter:



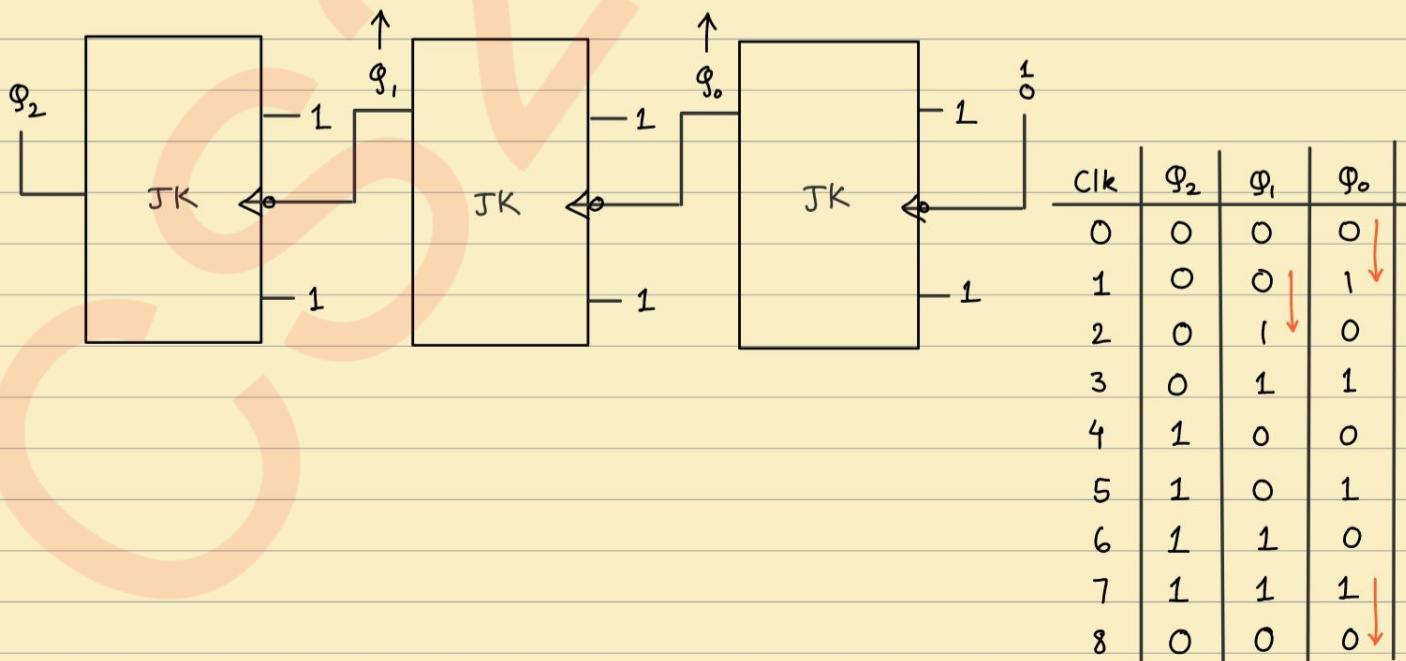
Clk	Φ_1	Φ_0	
0	0	0	+ve edge
1	0	1	-ve edge
2	1	0	+ve edge
3	1	1	-ve edge
4	0	0	

Counter

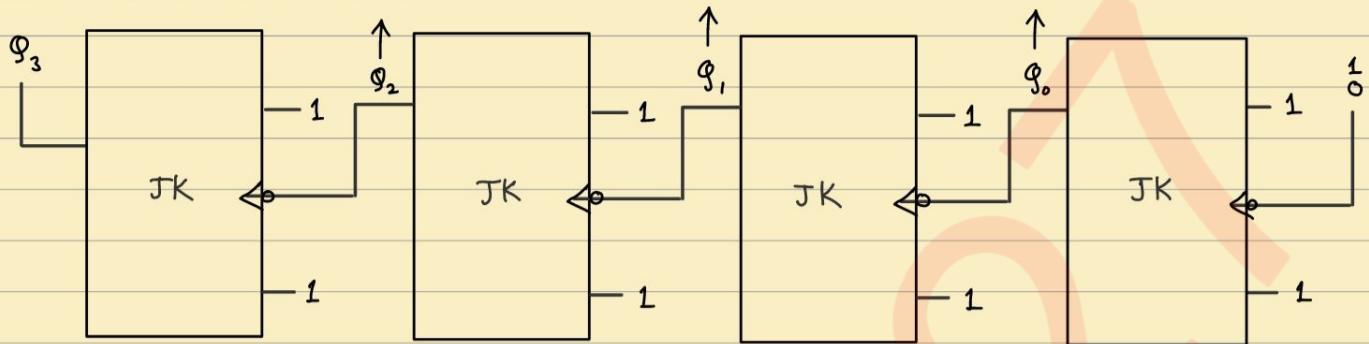
This is called Asynchronous counter ∵ it does not depend on the clock pulse but instead depends on previous output.
i.e. the edge makes the next one on or off.

This counter is called modulo 4 counter. ∵ 0-3 repeats
[modulo 10 counter : BCD]

Modulo 8 counter (Asynchronous)



Modulo 16 Counter : (Asynchronous)



Clk	Q_3	Q_2	Q_1	Q_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	0	0	0
9	1	0	0	0
10	1	0	0	1
11	1	0	1	0
12	1	0	1	1
13	1	1	0	0
14	1	1	0	1
15	1	1	1	0
16	1	1	1	1
17	1	0	0	0

→ 7 Segment display :

Display :



a	b	c	d	e	f	g
1	1	1	1	1	1	0
0	0	0	1	1	0	0

Input : 4 bits
Output : 7 bits

Homework

B_3	B_2	B_1	B_0
0	0	0	0
0	0	0	1
0	0	1	0
0	0	1	1
0	1	0	0
0	1	0	1
0	1	1	0
0	1	1	1
1	0	0	0
1	0	0	1

a	b	c	d	e	f	g
1	1	1	0	1	1	0
0	1	1	0	0	0	0
1	1	0	1	1	0	1
1	1	1	1	0	0	1
0	1	1	0	0	1	1
1	0	1	1	0	1	1
1	0	1	1	1	1	1
1	1	1	0	0	0	0
1	1	1	1	1	1	1
1	1	1	0	0	1	1

a
 f
 g
 b
 c
 e
 d

a

B_3	B_2	B_1	B_0
00	00	01	11
01	10	01	02
11	X12	X13	X15
10	18	19	X11

b

B_3	B_2	B_1	B_0
00	10	01	03
01	04	05	12
11	X12	X13	X15
10	18	09	X11

c

B_3	B_2	B_1	B_0
00	00	01	13
01	04	15	07
11	X12	X13	X15
10	18	09	X11

d

B_3	B_2	B_1	B_0
00	10	11	13
01	14	15	17
11	X12	X13	X15
10	18	19	X11

e

B_3	B_2	B_1	B_0
00	10	11	13
01	14	05	17
11	X12	X13	X15
10	18	19	X11

f

B_3	B_2	B_1	B_0
00	10	01	13
01	04	15	17
11	X12	X13	X15
10	18	19	X11

g

B_3	B_2	B_1	B_0
00	00	01	13
01	14	15	16
11	X12	X13	X15
10	18	19	X11

$$a = B_0 + B_1 \bar{B}_2 + B_1 B_3 + \bar{B}_2 \bar{B}_3$$

$$b = \bar{B}_1 \bar{B}_3 + B_2 \bar{B}_3$$

$$c = B_0 \bar{B}_3 + B_2 \bar{B}_3 + \bar{B}_1 B_2 + B_1 \bar{B}_2 B_3$$

$$d = B_1 + \bar{B}_2 + B_3$$

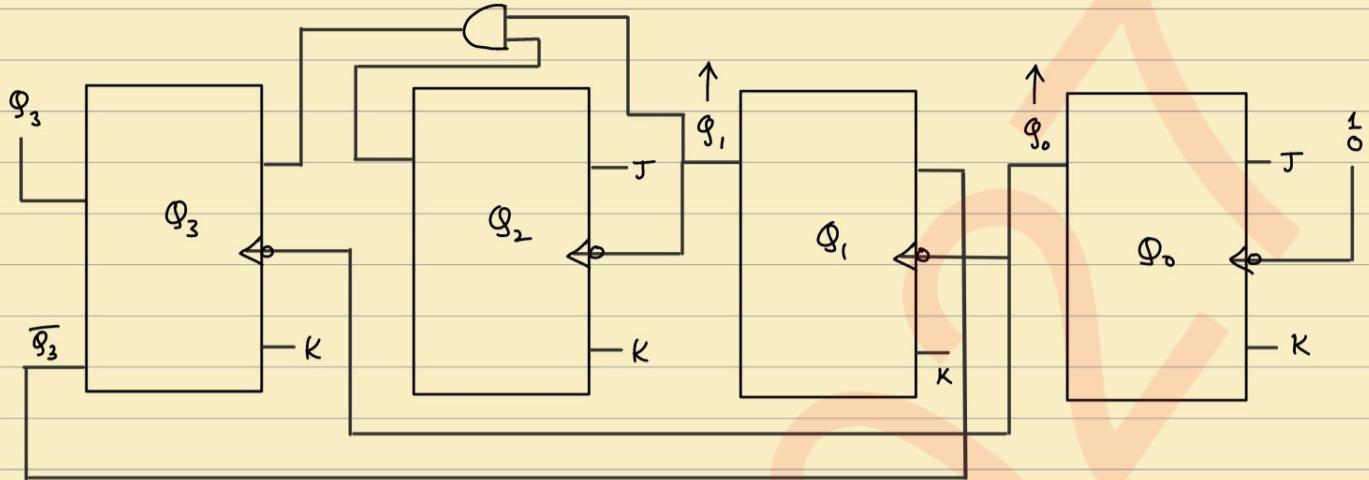
$$e = B_0 + \bar{B}_1 + B_2 B_3 + \bar{B}_2 \bar{B}_3$$

$$f = B_0 + B_2 + B_1 B_3 + \bar{B}_1 \bar{B}_3$$

$$g = B_0 + B_2 \bar{B}_3 + B_1 \bar{B}_2 + \bar{B}_0 \bar{B}_1 B_2$$

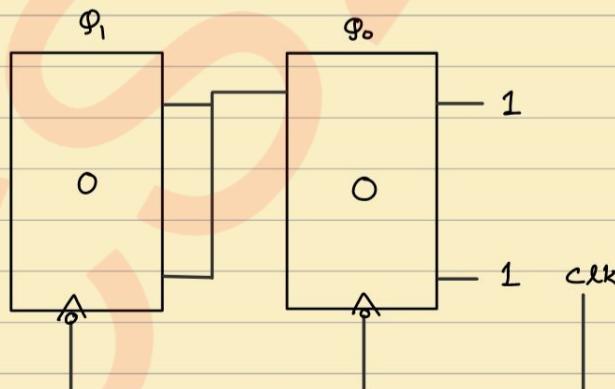
25/10

→ Ripple counter: - mod 10



Q_3	Q_2	Q_1	Q_0
0	0	0	0
0	0	0	1
0	0	1	0
0	0	1	1
0	1	0	0
0	1	0	1
0	1	1	0
0	1	1	1
1	0	0	0
1	0	0	1
0	0	0	0

→ Synchronous Mod-4 Up counter:

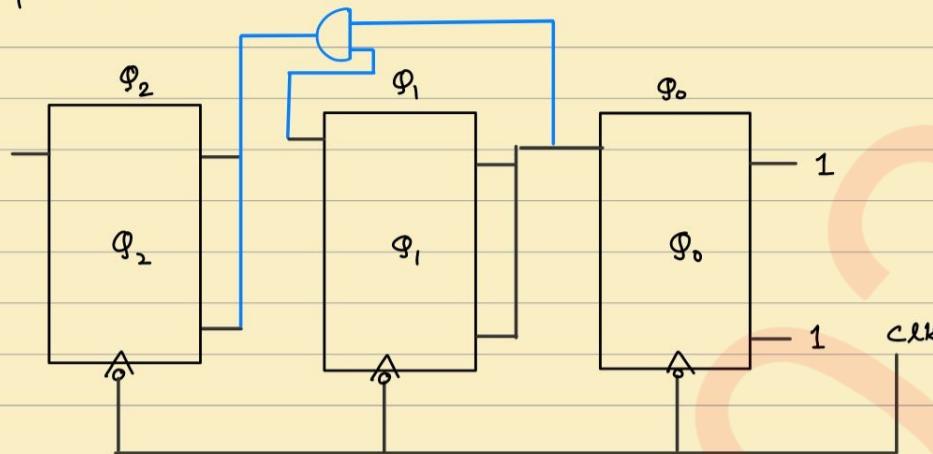


Depends on the previous output & not on the clock.
 $\therefore 1 \ 1$

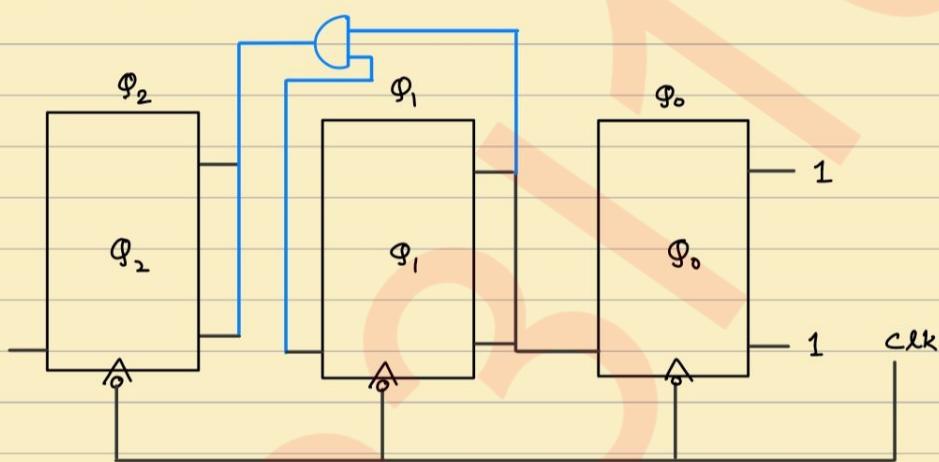
Q_1	Q_0
0	0
0	1
1	0
1	1

→ Synchronous mod-8 Counter:

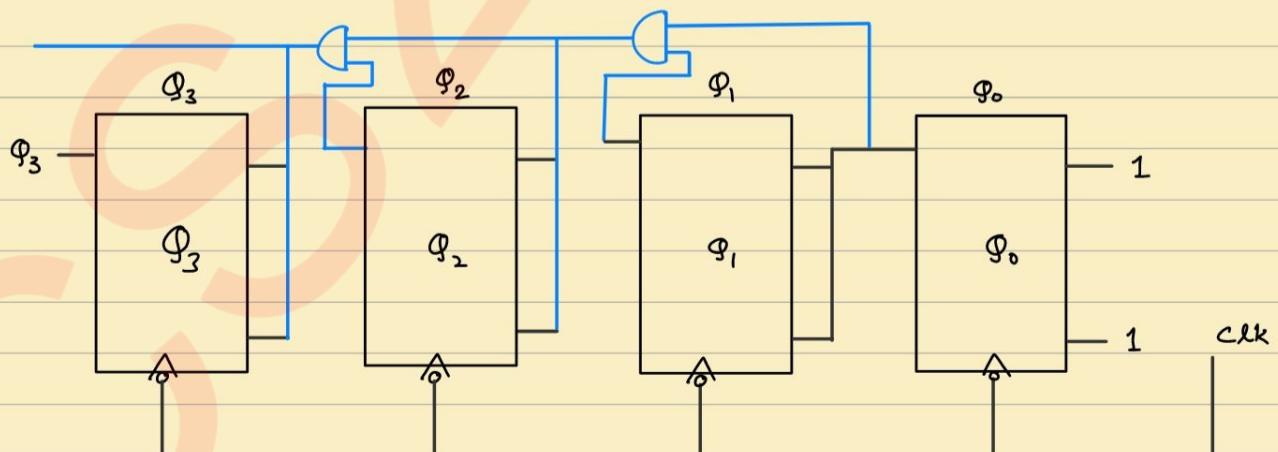
- Up Counter:



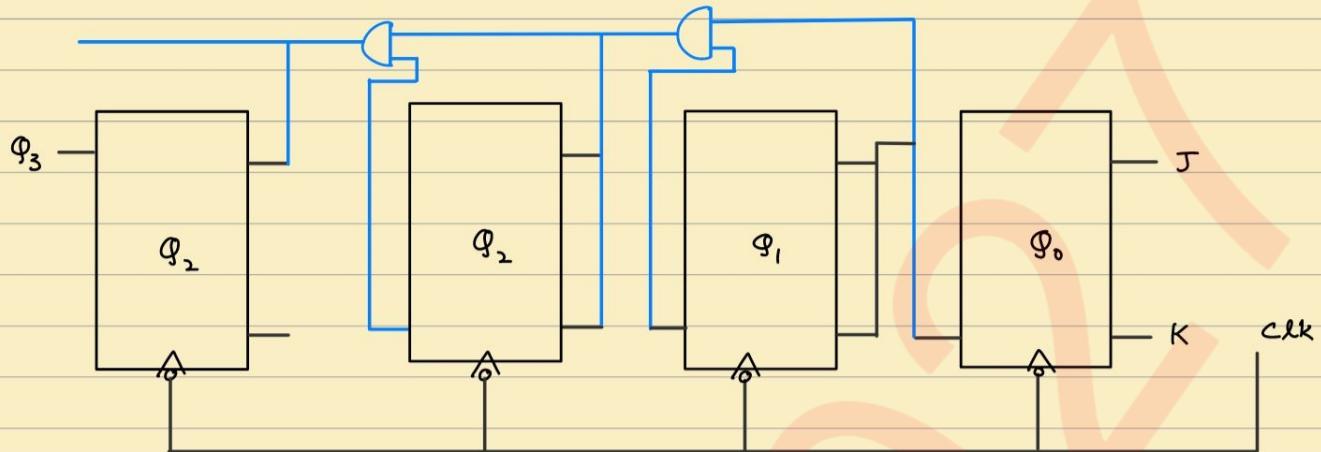
- Down counter:



→ Synchronous mod-16 Up counter:



- Synchronous mod-16 Down counter:



Output is taken from Q .

But Circuit operates on \bar{Q} .

Up - Counter :

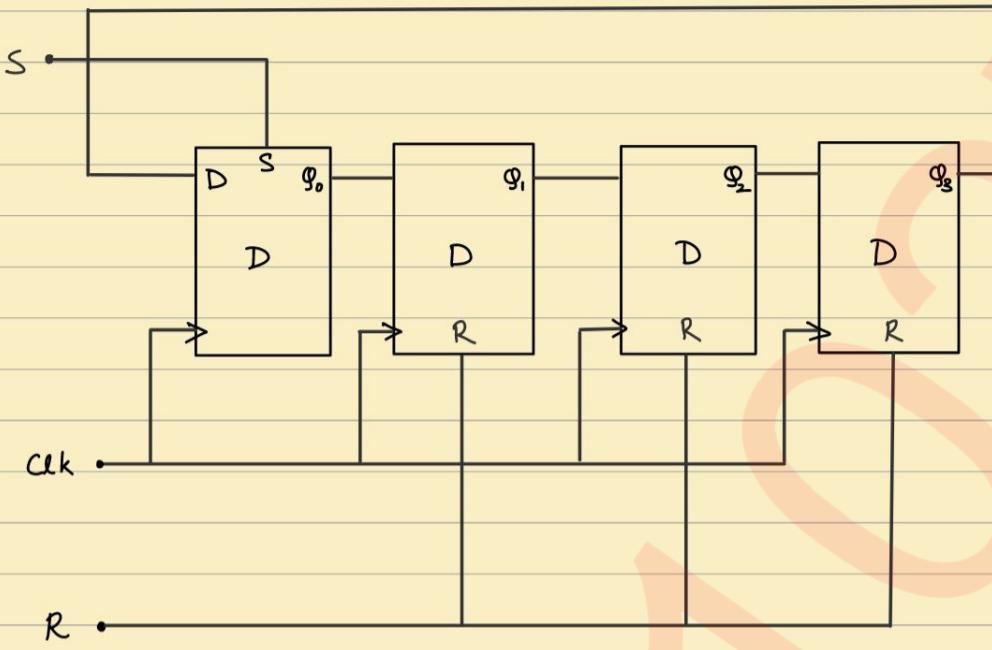
Q_3	Q_2	Q_1	Q_0
0	0	0	0
0	0	0	1
0	0	1	0
0	0	1	1
0	1	0	0
0	1	0	1
0	1	1	0
0	1	1	1
1	0	0	0
1	0	0	1
1	0	1	0
1	0	1	1
1	1	0	0
1	1	0	1
1	1	1	0
1	1	1	1

Down - counter:

Q_3	Q_2	Q_1	Q_0
1	1	1	1
1	1	1	0
1	1	0	1
1	1	0	0
1	0	1	1
1	0	1	0
1	0	0	1
1	0	0	0
0	1	1	1
0	1	1	0
0	1	0	1
0	1	0	0
0	0	1	1
0	0	1	0
0	0	0	1
0	0	0	0

28/10

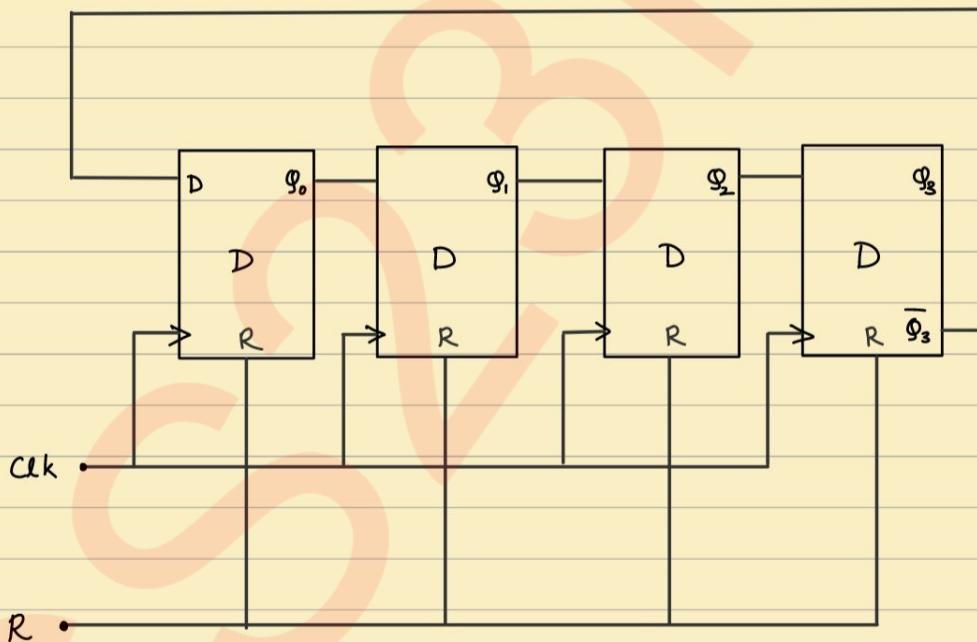
→ Ring Counter



	Q_3	Q_2	Q_1	Q_0
(S:0)	0	0	0	0
(S:1)	1	0	0	0
	0	1	0	0
	0	0	1	0
	0	0	0	1
	1	0	0	0
	0	1	0	0
	0	0	1	0
	0	0	0	1
⋮	⋮	⋮	⋮	⋮

No. of States
= No. of flipflops
= 4

→ Modification: (Variation)



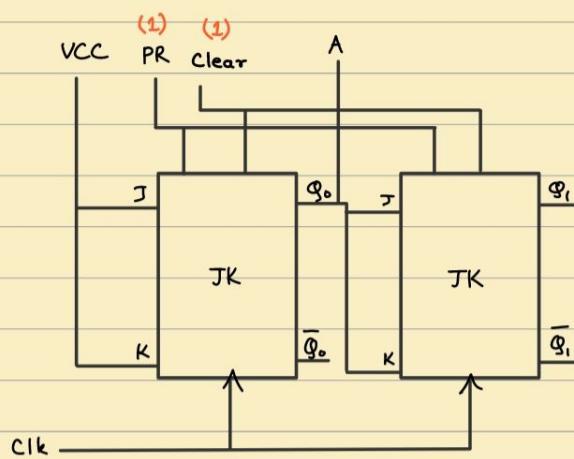
	Q_3	Q_2	Q_1	Q_0
	0	0	0	0
	1	0	0	0
	1	1	0	0
	1	1	1	0
	1	1	1	1
	0	1	1	1
	0	0	1	1
	0	0	0	1
	0	0	0	0
⋮	⋮	⋮	⋮	⋮

AKA Twisted Tail Counter
Connecting \bar{Q} instead of Q

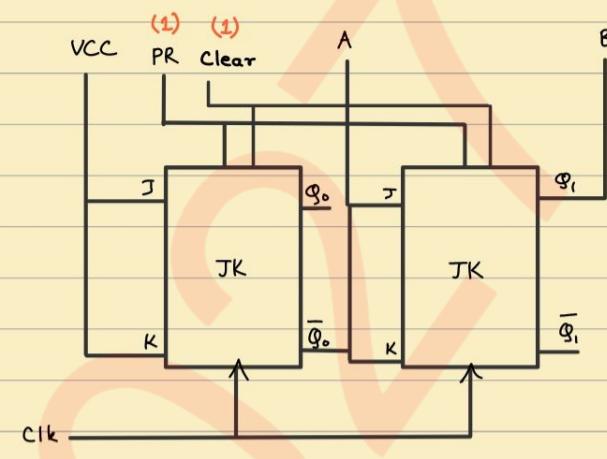
AKA Johnson Counter

Advantage: No. of states are more, i.e. $2 \times (\text{no. of flip-flops})$
if no. of states = k, no. of states = $2k$

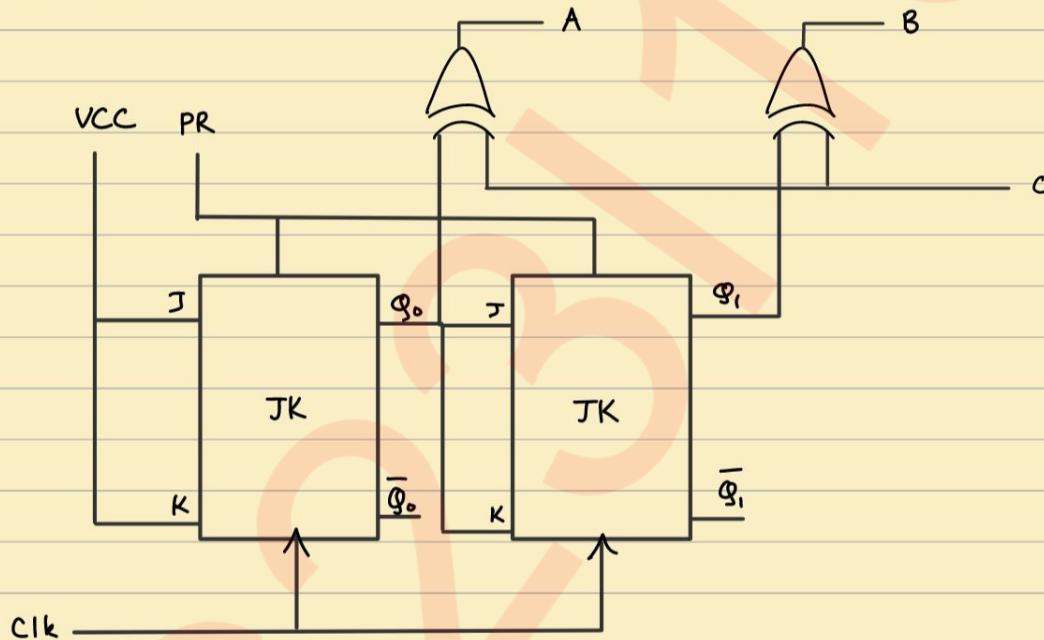
Q1) Up - Counter:



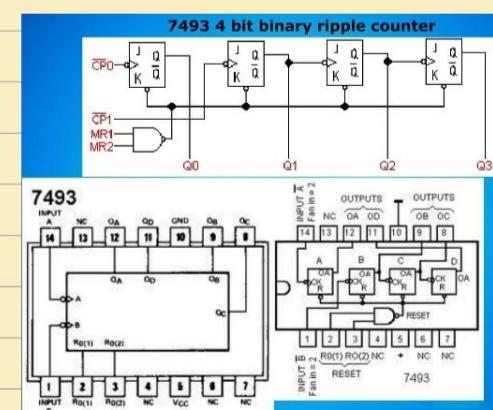
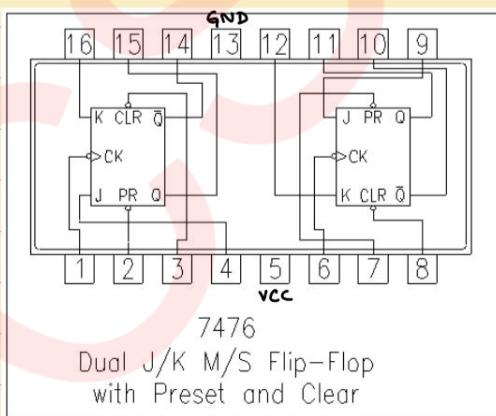
Q2) Down - Counter:



Q3) Combined:



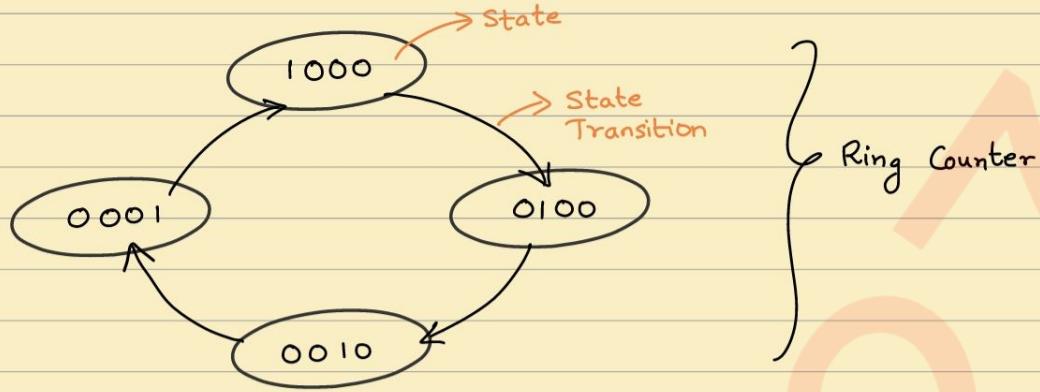
Q4) 6-bit counter using 2-bit counter + 4-bit ripple counter



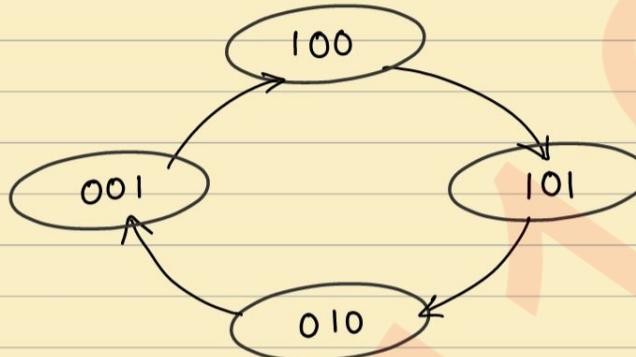
Connect
 ① Q1 to Clock A
 ② QA to Clock B

{ Output: φ1 φ2 φA φB φC φD

29/10



Example :



I	O
100	101
101	010
010	001
001	100

$$Q_t = 0, Q_{t+1} = 1$$

$$Q_t = 1, Q_{t+1} = 0$$

Excitation tables :

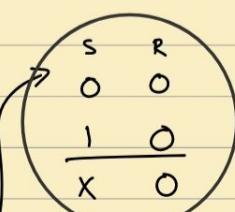
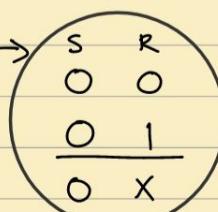
DFF:	Q_t	Q_{t+1}	D
	0	0	0
	0	1	1
	1	0	0
	1	1	1

$$D = Q_{t+1}$$

TFF:	Q_t	Q_{t+1}	T
	0	0	0
	0	1	1
	1	0	1
	1	1	0

$$T = Q_t \oplus Q_{t+1}$$

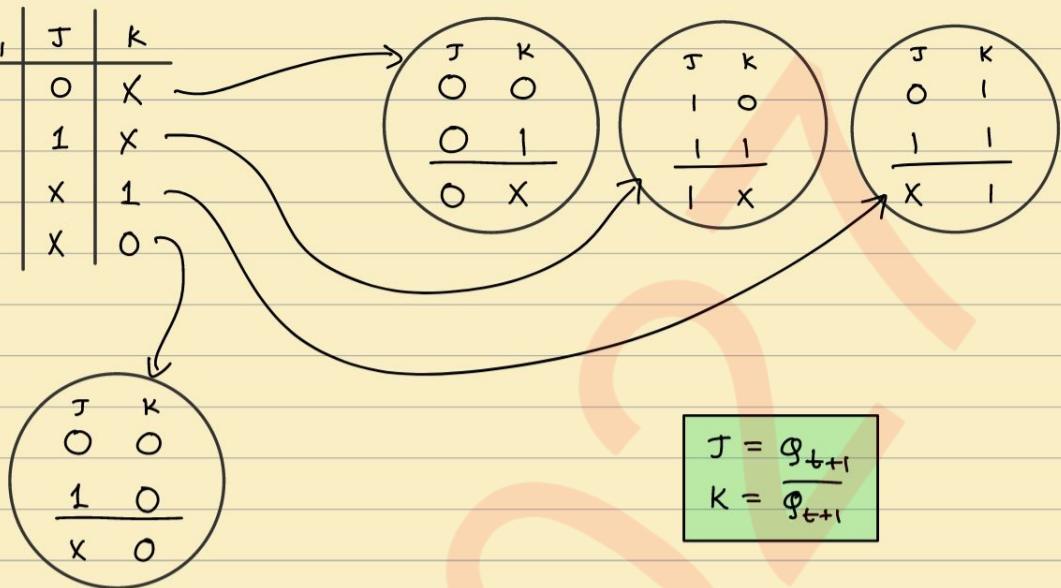
SR:	Q_t	Q_{t+1}	S	R
	0	0	0	X
	0	1	1	0
	1	0	0	1
	1	1	X	0



$$S = Q_{t+1}$$

$$R = \overline{Q_{t+1}}$$

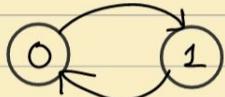
JK :	Q_t	Q_{t+1}	J	K
0	0	0	X	
0	1	1	X	
1	0	X	1	
1	1	X	0	



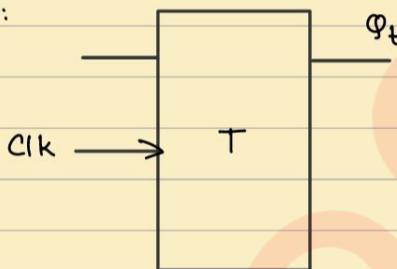
$$J = Q_{t+1}$$

$$K = \overline{Q_{t+1}}$$

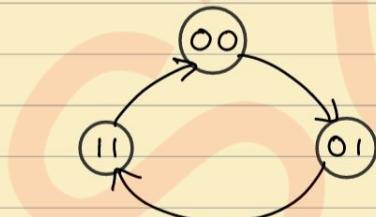
Q)



Sol:



Q)



<u>Current</u>		<u>Next</u>			
Q_A	Q_B	Q_{A+1}	Q_{B+1}	T_A	T_B
0	0	0	1	1	0
0	1	1	1	1	0
1	1	0	0	0	1

$$T_A = Q_B$$

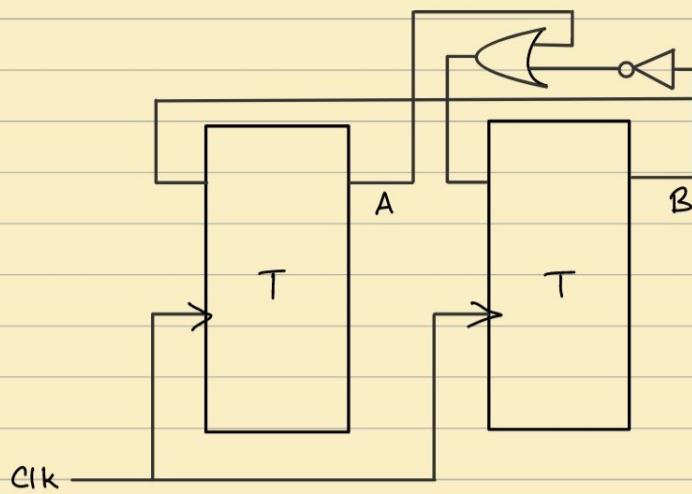
$$T_B = Q_A \odot Q_B$$

Q_A	Q_B	T_A	Q_{A+1}	Q_{B+1}
0	0	0	0	1
1	X	1	1	1

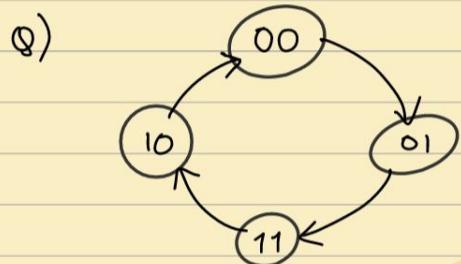
Q_A	Q_B	T_B	Q_{A+1}	Q_{B+1}
0	1	0	1	0
1	X	1	1	1

$$T_B = \overline{Q_B} + Q_A$$

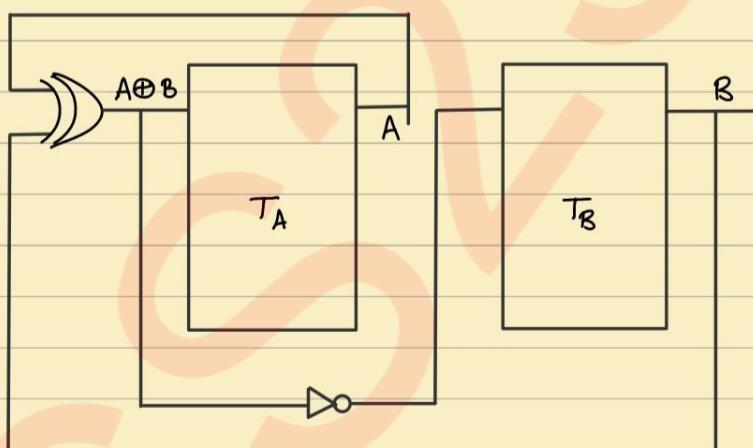
Circuit Diagram:



Q/11



Sol:

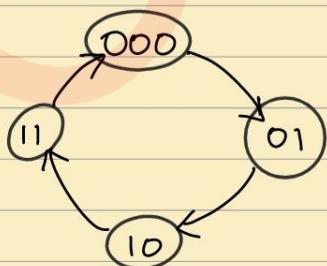


Prev		Next			
A	B	A	B	T _A	T _B
0	0	0	1	0	1
0	1	1	1	1	0
1	1	1	0	0	1
1	0	0	0	1	0

$$T_A = A \oplus B$$

$$T_B = A \circ B$$

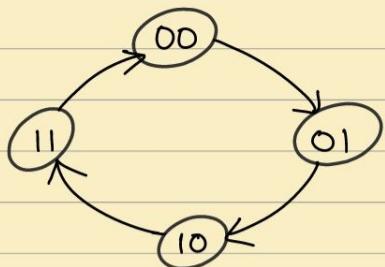
Q)



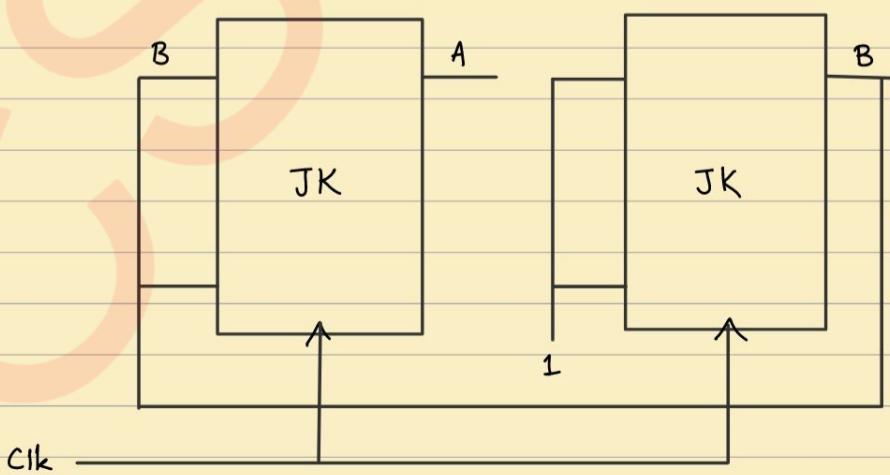
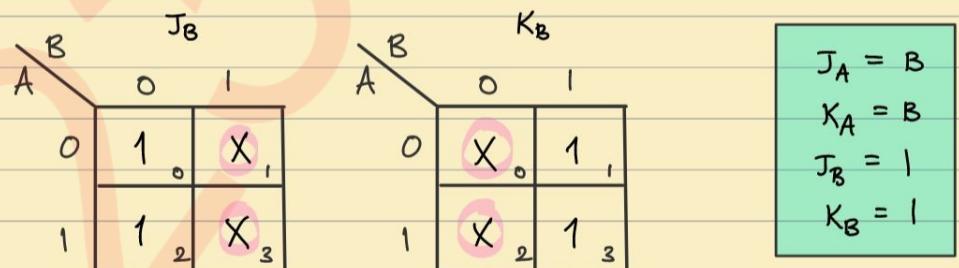
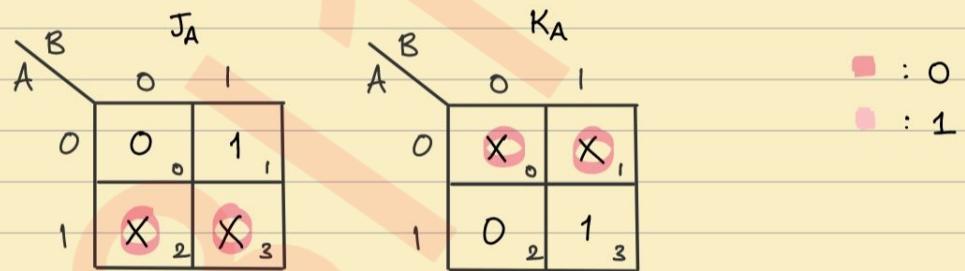
Not Valid

But 3 flip-flops used.
∴ 3 max bits exist

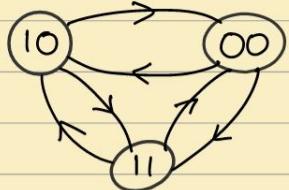
Q)

Sol:

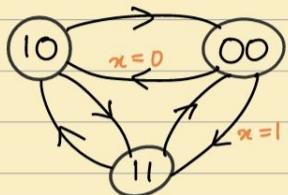
Prev		Next	J_A	K_A	J_B	K_B
A	B	A	B			
0	0	0	1	0	X	1
0	1	1	0	1	X	X
1	0	1	1	X	0	1
1	1	0	0	X	1	X



Note:



Cannot know which transition to go from.
Hence, an actual input ' x ' is taken.



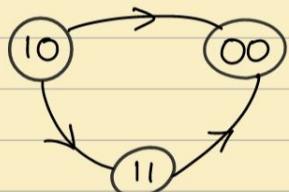
$x = 0, 1$ (i/p)

Note:



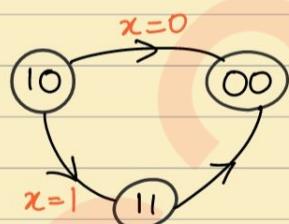
Doesn't make sense

But,



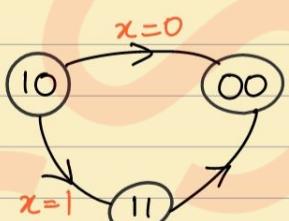
Not deterministic, i.e. no one specific path exists.
Hence, we add input variable,

To differentiate multiple possibilities. (More than 1 transition)



' x ' is taken to choose which transition to undergo.

(Q)



Sol:

x is not required :: It only just decides the transition

x	Prev		Next			
	A	B	A	B	T_A	T_B
0	1	0	0	0	1	0
0	1	0	0	1	1	1
X	0	1	0	0	0	1
X	0	0	X	X	X	X
X	1	1	X	X	X	X

No. of flip-flops
= No. of bits
= 2

This is an example of Finite Set Machine (FSM)

		T _A				
		AB	00	01	11	10
x	0	X ₀	O ₁	X ₃	1 ₂	
	1	X ₄	O ₅	X ₇	1 ₆	

		T _B				
		AB	00	01	11	10
x	0	X ₀	1 ₁	X ₃	O ₂	
	1	X ₄	1 ₅	X ₇	O ₆	

Selective Prime Implicants

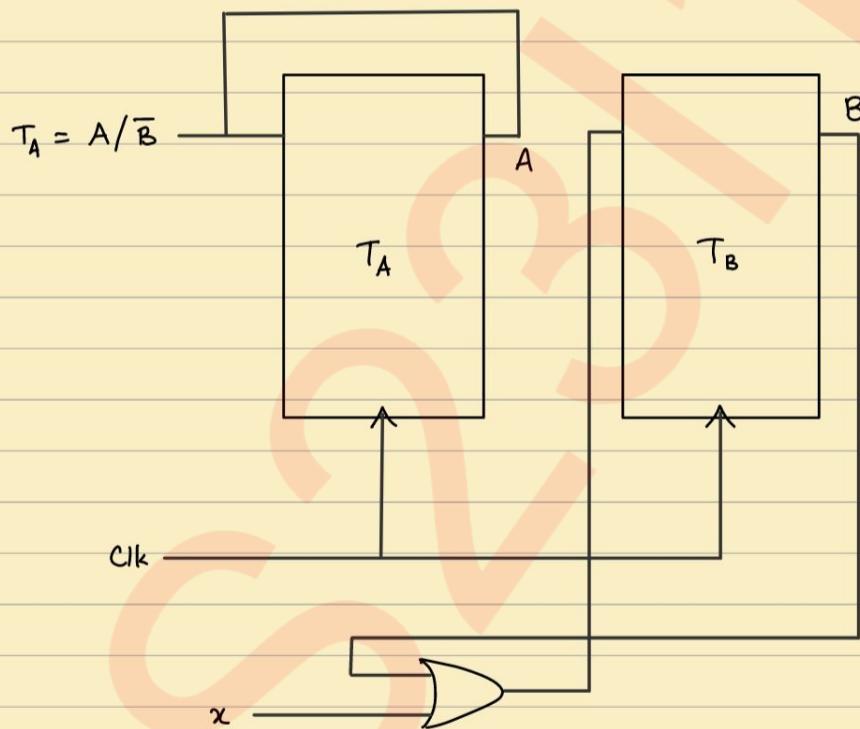
i.e. Has 2 answers & both are perfectly valid

i.e. $\{(X_0, X_4) = 0 \text{ or } (X_3, X_7) = 0\}$

$$T_A = A \text{ (or) } \bar{B}$$

$$T_B = x + B \text{ (or) } x + \bar{A}$$

Circuit Diagram:



Cameo:



6/11

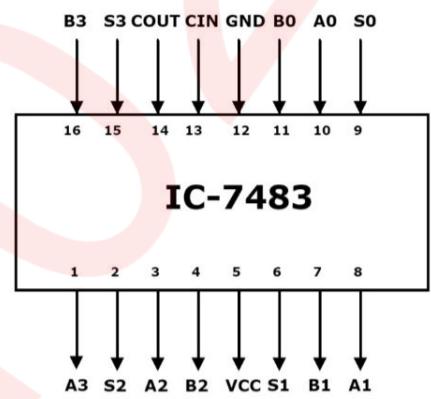
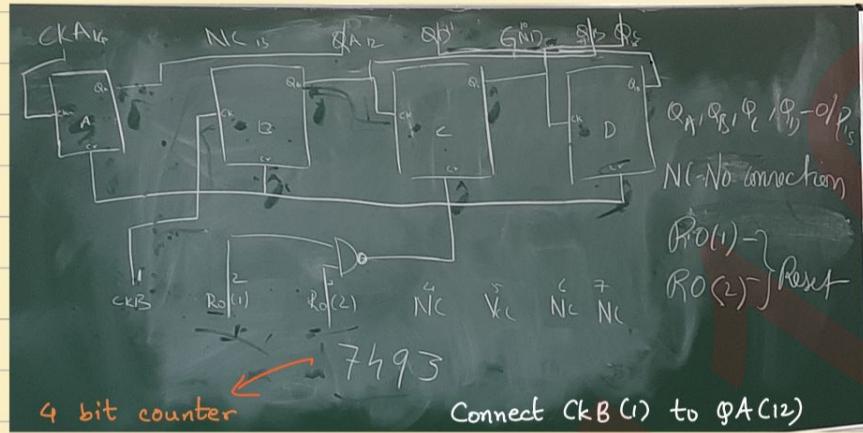
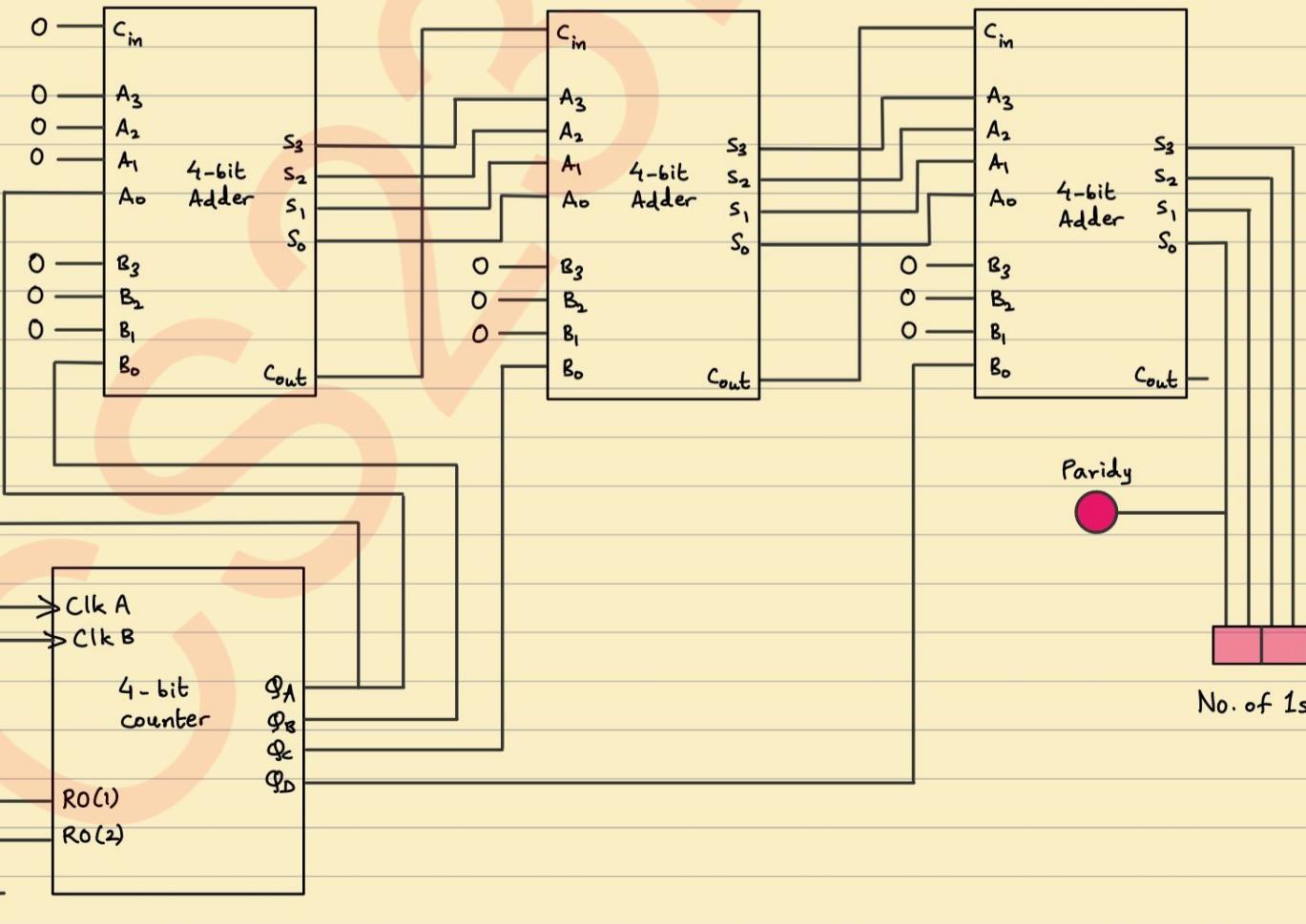
Lab - 9Hardware

Q1) Given 4 bits input, Check Parity :

i.e. If no. of 1s is odd \Rightarrow Output is 1

If no. of 1s is even \Rightarrow Output is 0.

Generate 0000 to 1111 (Set of Inputs using Counter)
and count the number of 1s of the counter's output

Circuit:

Q2) JK flipflop using NAND gates

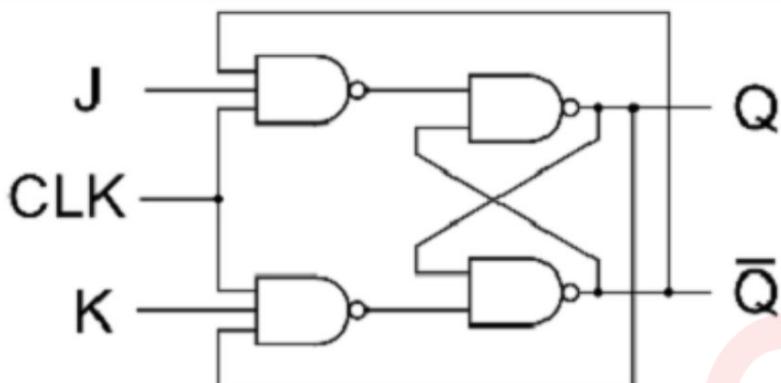


Figure 4: JK Flip Flop

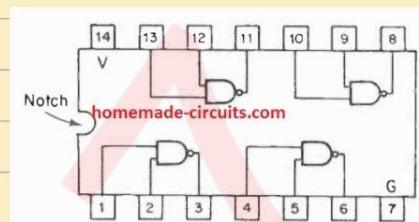
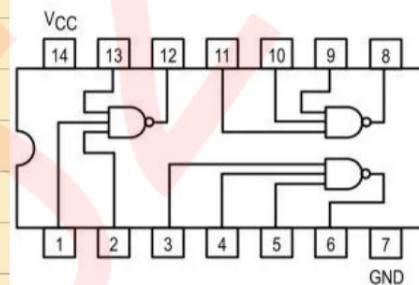


Figure #1. 7400 IC logic diagram.



7410 IC Logic Diagram

Software

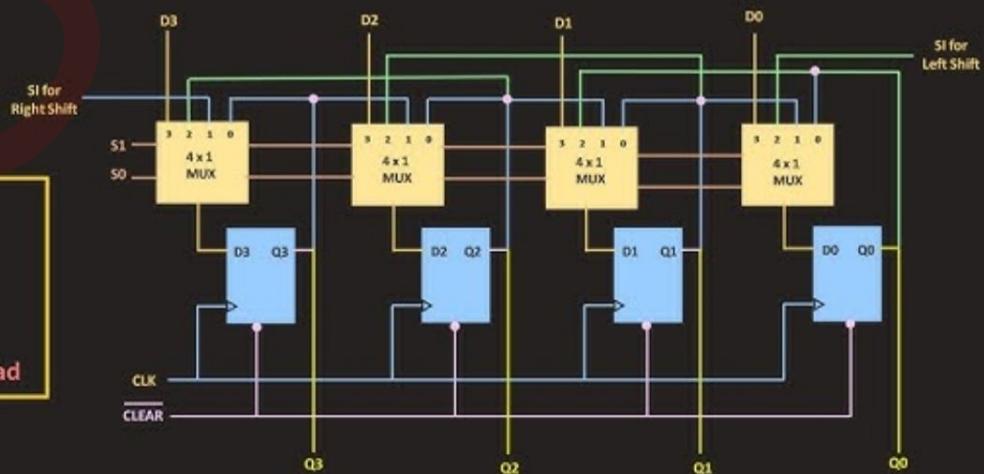
Q) 4 bit Universal Shift Register

- (i) Left Shift
- (ii) Right Shift
- (iii) Parallel Input (Using MUX)
- (iv) Retain Value

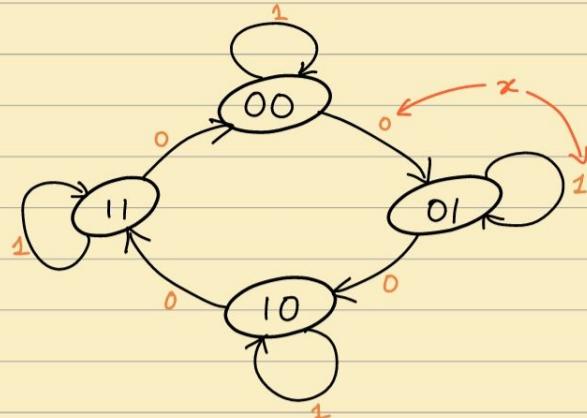
Circuit:

Universal Shift Register

S1 S0	
0 0	Hold
0 1	Shift Right
1 0	Shift Left
1 1	Parallel Load



8/11



x	Prev		Next		T_A	T_B
	A	B	A	B		
0	0	0	0	1	0	1
0	0	1	1	0	1	1
0	1	0	1	1	0	1
0	1	1	0	0	1	1
1	0	0	0	0	0	0
1	0	1	0	1	0	0
1	1	0	1	0	0	0
1	1	1	1	1	0	0

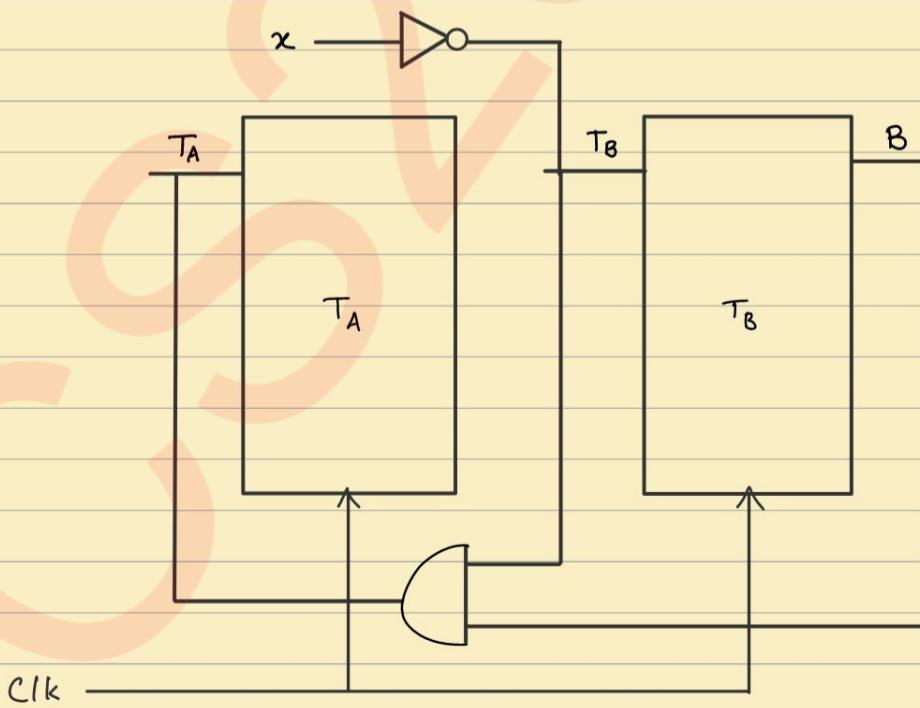
x	T_A			
	00	01	11	10
0	0	1	1	0
1	0	0	0	0

$$T_A = \bar{x}B$$

x	T_B			
	00	01	11	10
0	1	1	1	1
1	0	0	0	0

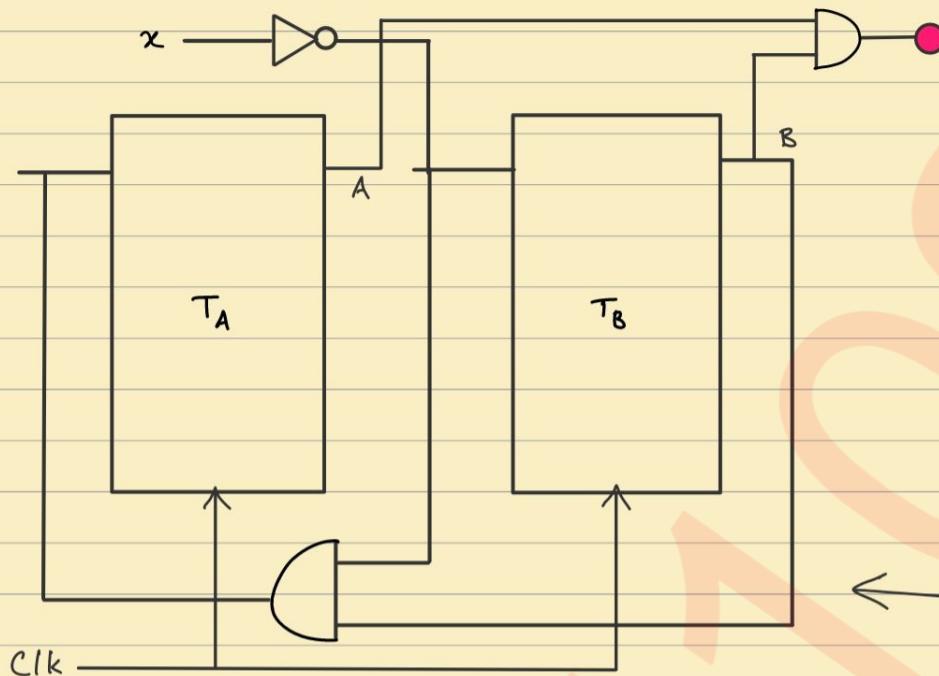
$$T_B = \bar{x}$$

- Circuit Diagram:



Whenever you reach 11 state, circuit should notify, Hence

Circuit :



This circuit notifies but the problem is, this counts cycles, i.e. the LED will be on the whole time.

Ex. if a stopwatch is stopped at a particular time, the counter will not stop.

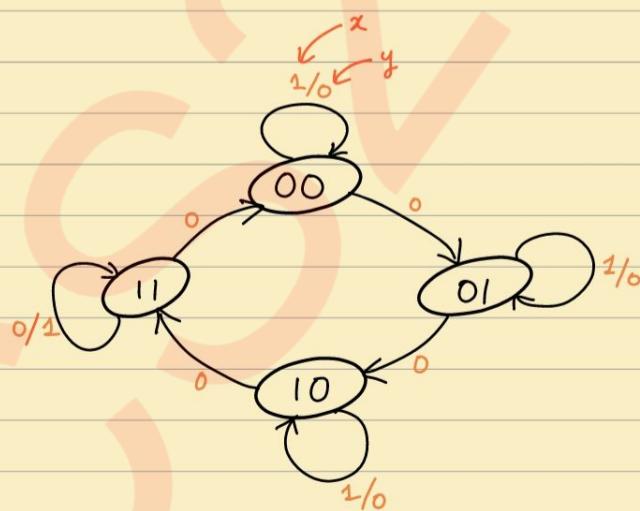
Hence, Mealy is used.

Based on present state, we'll have output : Moore Machine

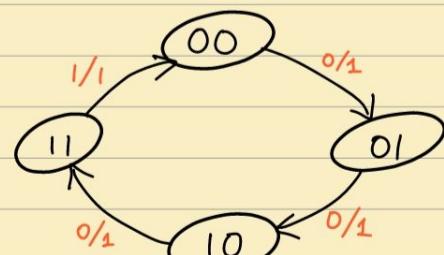
Based on present state + input, we'll have output : Mealy Machine

FSM

- Moore Machine (y : no. of states)
- Mealy Machine (y : no. of transitions)

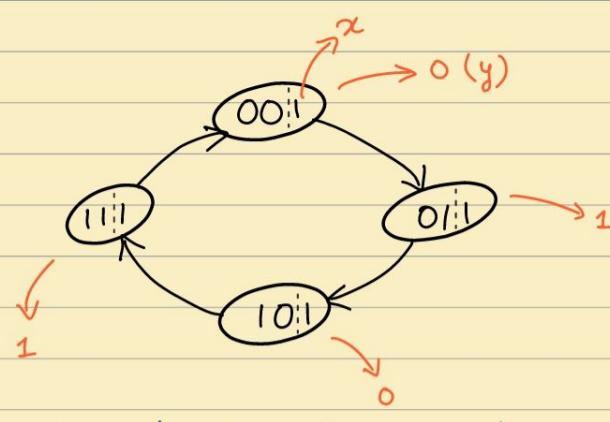


Moore Machine



Mealy Machine

9/11



* Alan Turing

Moore Machine
H.W : Moore's Law

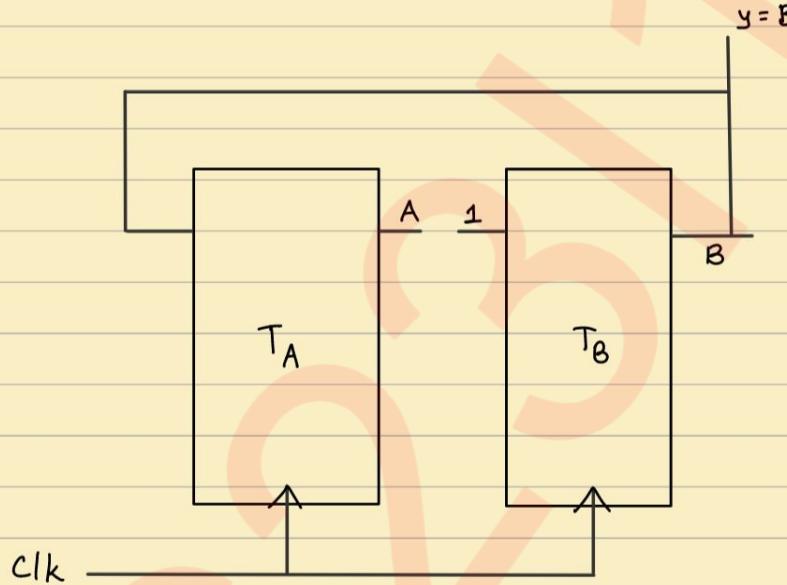
* Von Neumann - Computer Architecture

Prev		Next		T_A	T_B	y
A	B	A	B			
0	0	0	1	0	1	0
0	1	1	0	1	1	1
1	0	1	1	0	1	0
1	1	0	0	1	1	1

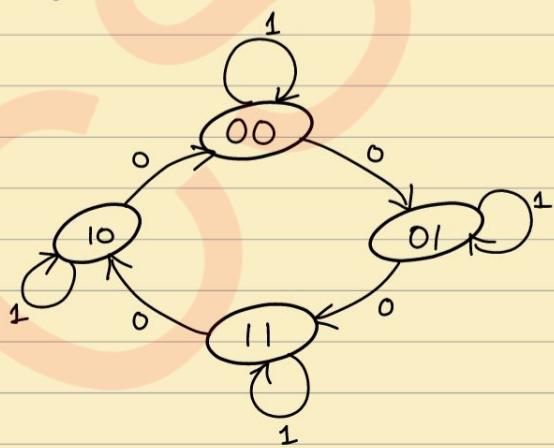
T_A		
A	0	1
0	0	1
1	0	1

$$T_A = B$$

- Circuit Diagram:

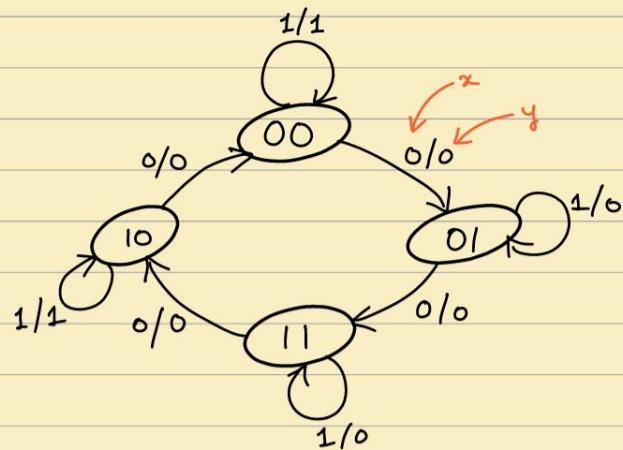


- Mealy Machine:



Even + Stay = 1
↳ Transition
↳ State

Mealy Machine primarily depends on the transition.



Output is valid until the next state :: Transition is also taken into consideration.

x	A	B	A	B	T _A	T _B	y
0 0	0	0	0	1	0	1	0
0 0	0	1	1	1	1	0	0
0 1	0	0	0	0	1	0	0
0 1	1	1	1	0	0	1	0
1 0	0	0	0	0	0	0	1
1 0	1	0	1	0	0	0	0
1 1	0	1	0	0	0	0	1
1 1	1	1	1	1	0	0	0

AB	00	01	11	10	T _A
0	0	1	0	1	0
1	0	0	0	1	1

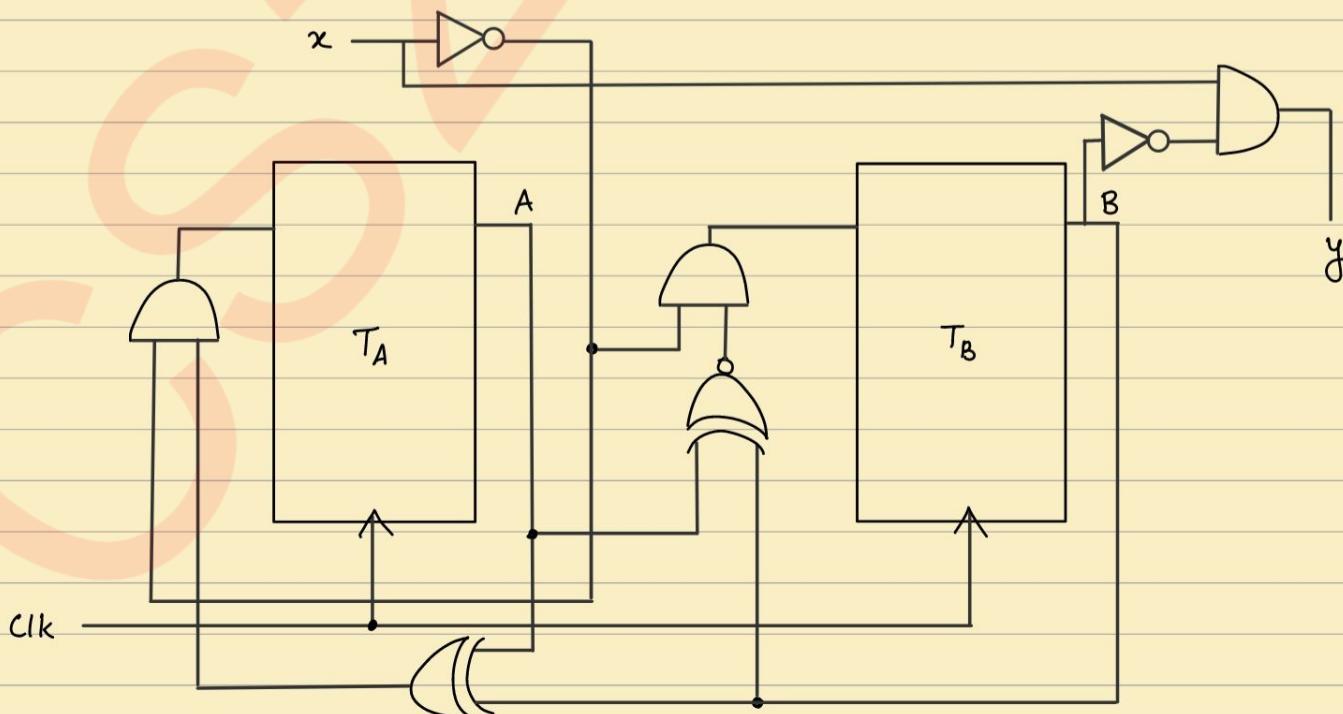
AB	00	01	11	10	T _B
0	1	0	1	0	1
1	0	0	0	1	0

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

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

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

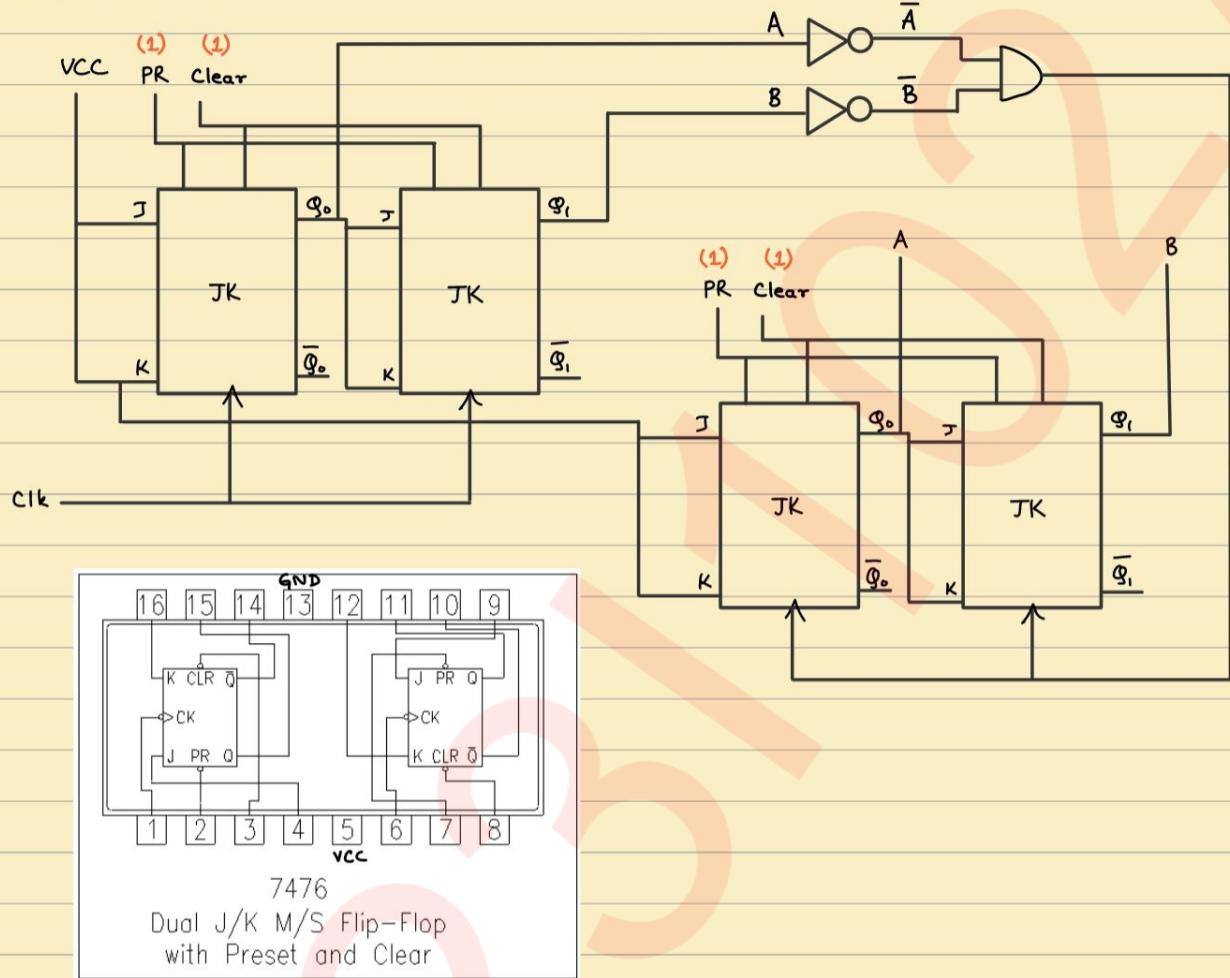
- Circuit Diagram:



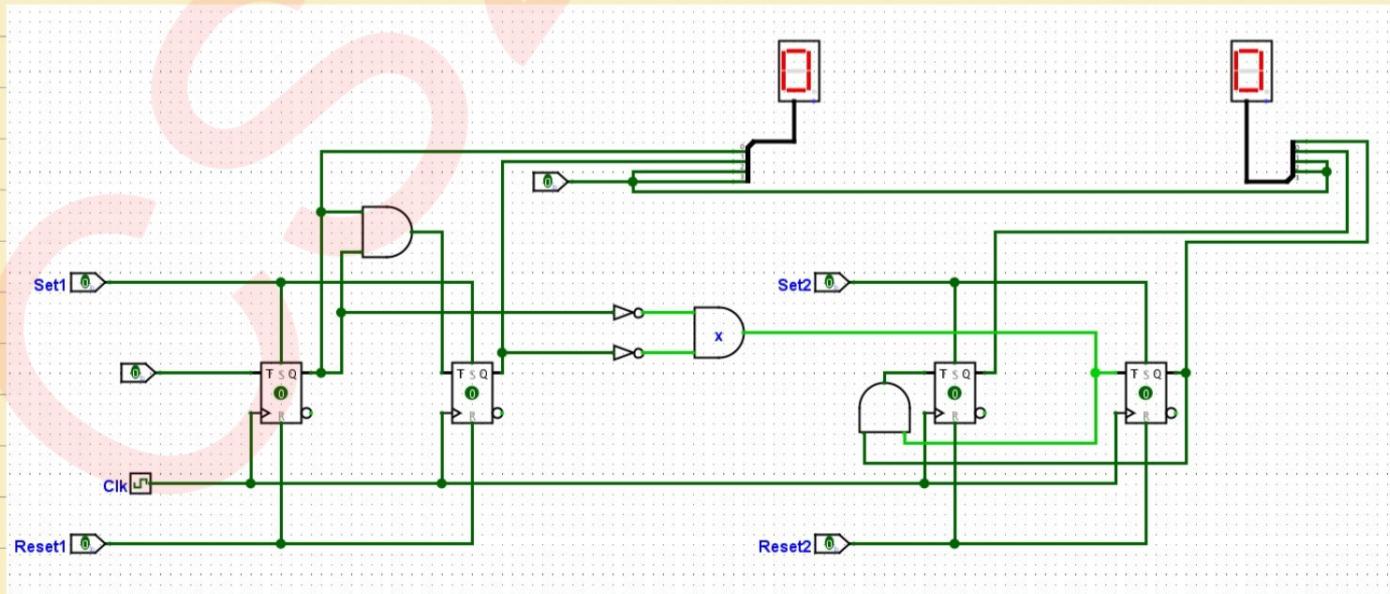
Hardware

Q) Make a 2 bit counter, count the no. of cycles i.e. no. of times 00 gets displayed

Asynchronous :



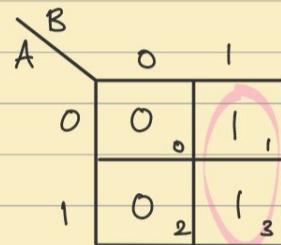
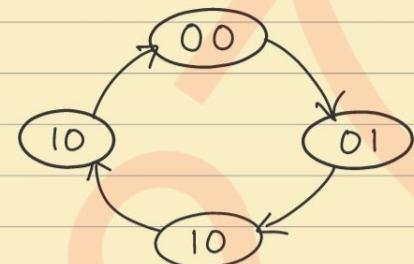
Synchronous :



Explanation:

①

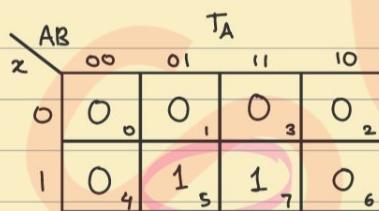
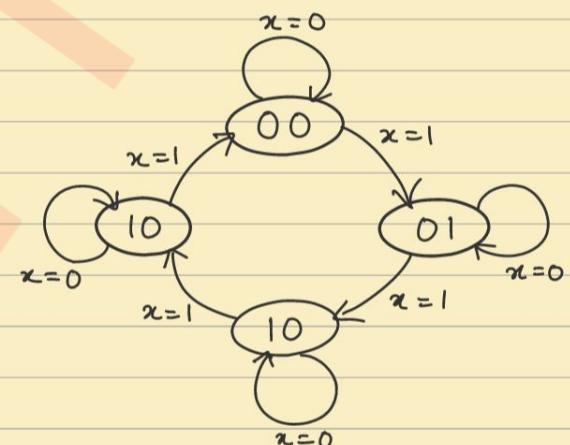
Prev		Next		T_A	T_B
A	B	A	B		
0	0	0	1	0	1
0	1	1	0	1	1
1	0	1	1	0	1
1	1	0	0	1	1



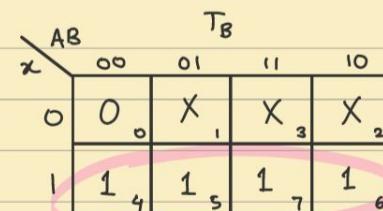
$$\begin{aligned}T_A &= B \\T_B &= 1\end{aligned}$$

②

x	Prev		Next		T_A	T_B
	A	B	A	B		
0	0	0	0	0	0	0
1	0	0	0	1	0	1
1	0	1	1	0	1	1
1	1	0	1	1	0	1
1	1	1	0	0	1	1
0	0	1	0	1	0	0
0	1	0	1	0	0	0
0	1	1	1	1	0	0

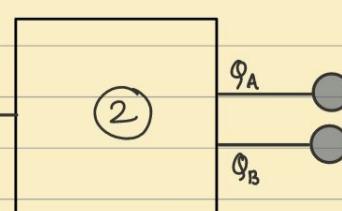
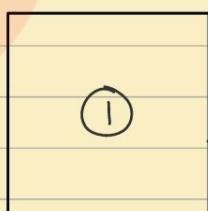


$$T_A = xB$$



$$T_B = x$$

Circuit Diagram:



11/11

Assignment - I

B_3	B_2	B_1	B_0	a	b	c	d	e	f	g
0	0	0	0	1	0	0	1	1	1	0
0	0	0	1	1	0	1	1	0	1	1
0	0	1	0	1	1	0	1	1	0	1
0	0	1	1	1	1	1	1	0	0	1
0	1	0	0	0	1	1	0	0	0	0
0	1	0	1	0	1	1	0	0	0	0
0	1	1	0	1	1	1	1	1	1	0
0	1	1	1	1	1	0	1	1	0	1
1	0	0	0	1	1	1	0	0	0	0

CS23I1027



a

B_0B_1	B_2B_3	00	01	11	10
00	00	1 ₀	1 ₁	1 ₃	1 ₂
01	04	0 ₄	0 ₅	1 ₇	1 ₆
11	X ₁₂	X ₁₃	X ₁₅	X ₁₄	
10	1 ₈	X ₉	X ₁₁	X ₁₀	

b

B_0B_1	B_2B_3	00	01	11	10
00	00	0 ₀	0 ₁	1 ₃	1 ₂
01	14	1 ₄	1 ₅	1 ₇	1 ₆
11	X ₁₂	X ₁₃	X ₁₅	X ₁₄	
10	1 ₈	X ₉	X ₁₁	X ₁₀	

c

B_0B_1	B_2B_3	00	01	11	10
00	00	0 ₀	1 ₁	1 ₃	0 ₂
01	14	1 ₄	1 ₅	0 ₇	1 ₆
11	X ₁₂	X ₁₃	X ₁₅	X ₁₄	
10	1 ₈	X ₉	X ₁₁	X ₁₀	

d

B_0B_1	B_2B_3	00	01	11	10
00	00	1 ₀	1 ₁	1 ₃	1 ₂
01	04	0 ₄	0 ₅	1 ₇	1 ₆
11	X ₁₂	X ₁₃	X ₁₅	X ₁₄	
10	0 ₈	X ₉	X ₁₁	X ₁₀	

e

B_0B_1	B_2B_3	00	01	11	10
00	00	1 ₀	0 ₁	0 ₃	1 ₂
01	04	0 ₄	0 ₅	1 ₇	1 ₆
11	X ₁₂	X ₁₃	X ₁₅	X ₁₄	
10	0 ₈	X ₉	X ₁₁	X ₁₀	

f

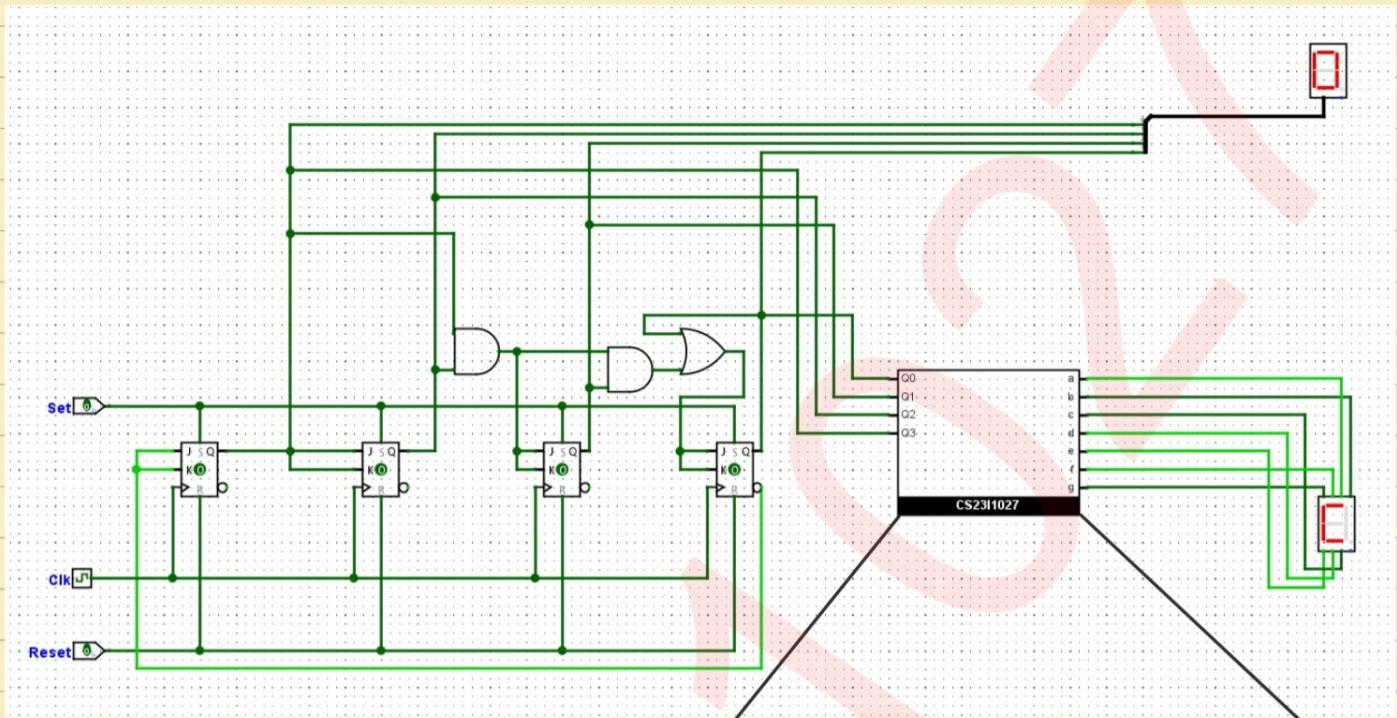
B_0B_1	B_2B_3	00	01	11	10
00	00	1 ₀	1 ₁	0 ₃	0 ₂
01	04	0 ₄	0 ₅	0 ₇	1 ₆
11	X ₁₂	X ₁₃	X ₁₅	X ₁₄	
10	0 ₈	X ₉	X ₁₁	X ₁₀	

g

B_0B_1	B_2B_3	00	01	11	10
00	00	0 ₀	1 ₁	1 ₃	1 ₂
01	04	0 ₄	0 ₅	1 ₇	0 ₆
11	X ₁₂	X ₁₃	X ₁₅	X ₁₄	
10	0 ₈	X ₉	X ₁₁	X ₁₀	

$a = \overline{B}_1 + B_2$ $b = B_0 + B_1 + B_2$ $c = B_0 + B_1\overline{B}_2 + B_1 \oplus B_3$ $d = \overline{B}_0\overline{B}_1 + B_2$ $e = B_1B_2 + \overline{B}_0\overline{B}_1\overline{B}_3$ $f = \overline{B}_0\overline{B}_1\overline{B}_2 + B_1B_2\overline{B}_3$ $g = \overline{B}_1B_2 + \overline{B}_1B_3 + B_2B_3$

Circuit:

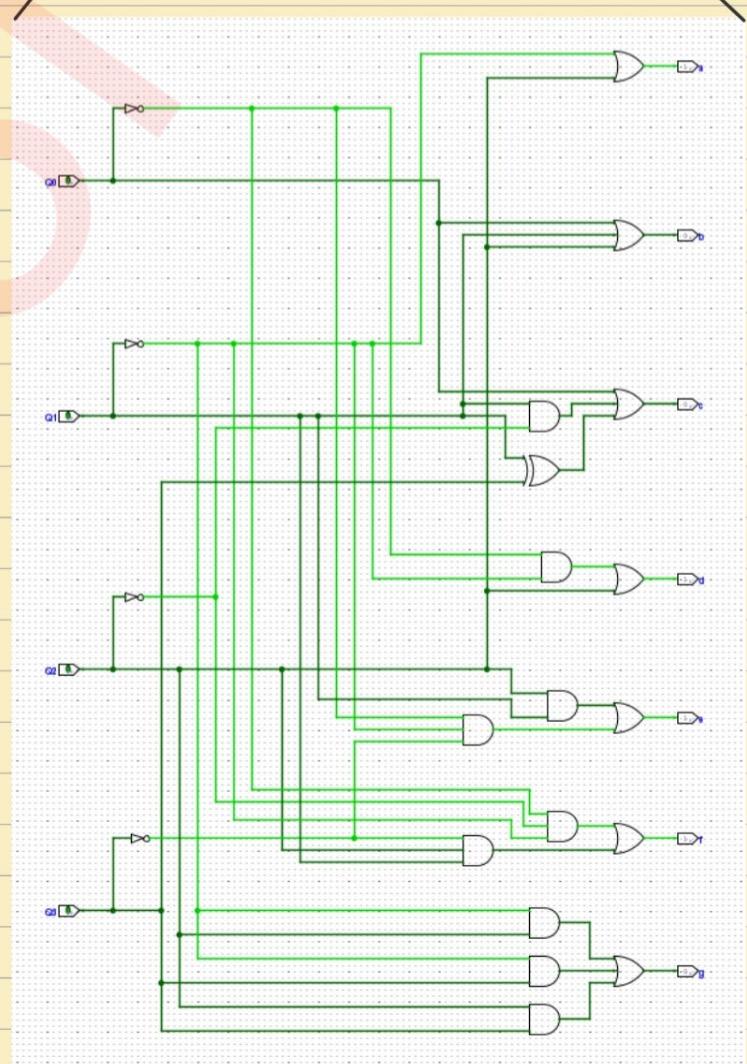


C S 2 3 T 1 0 2 7
① ② ③ ④ ⑤ ⑥ ⑦ ⑧

Use Modulo-9 counter

Explained in
the next page
(Sequential Component)

Combinational part
already derived in
the previous page



Current				Next				T_A	T_B	T_C	T_D
A	B	C	D	A	B	C	D				
0	0	0	0	0	0	0	0	1	0	0	0
0	0	0	1	0	0	1	0	0	0	1	1
0	0	1	0	0	0	1	1	0	0	0	1
0	0	1	1	0	1	0	0	0	1	1	1
0	1	0	0	0	1	0	1	0	0	0	1
0	1	0	1	0	1	1	0	0	0	1	1
0	1	1	0	0	1	1	1	0	0	0	1
0	1	1	1	0	0	0	0	1	1	1	1
1	0	0	0	0	0	0	0	1	0	0	1

$$T_D = 1$$

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

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

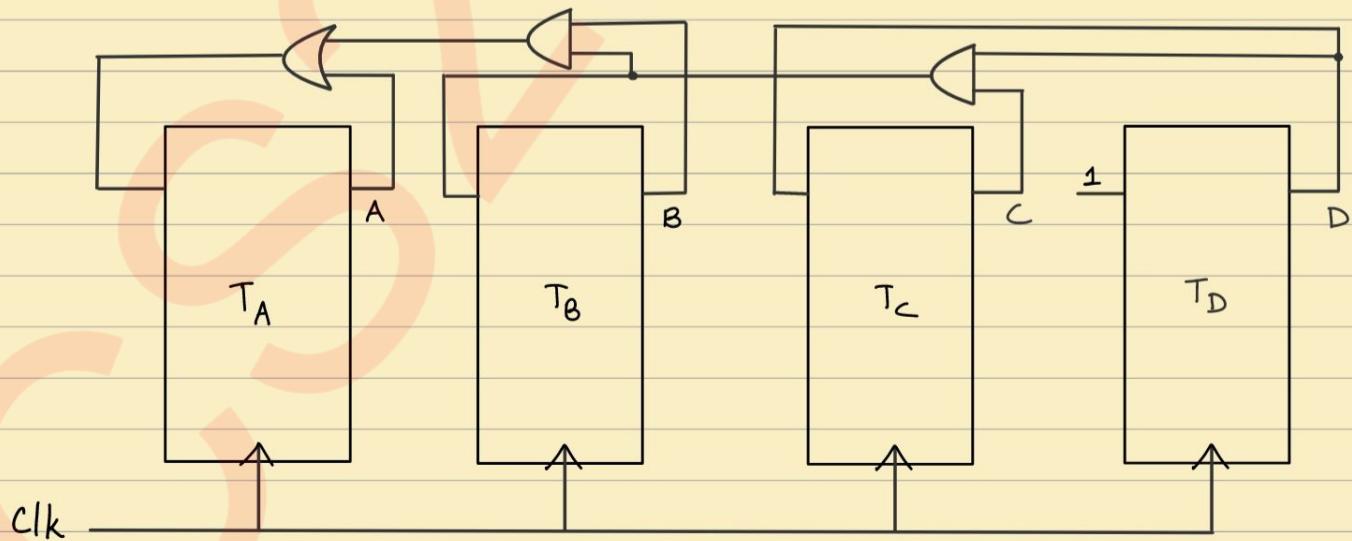
		CD				
		AB	00	01	11	10
	00		0 ₀	1 ₁	1 ₃	0 ₂
01	01		0 ₄	1 ₅	1 ₇	0 ₆
11	X ₁₂	X ₁₃	X ₁₅	X ₁₄		
10	0 ₈	X ₉	X ₁₁	X ₁₀		

$$T_A = A + BCD$$

$$T_B = CD$$

$$T_C = D$$

Circuit :



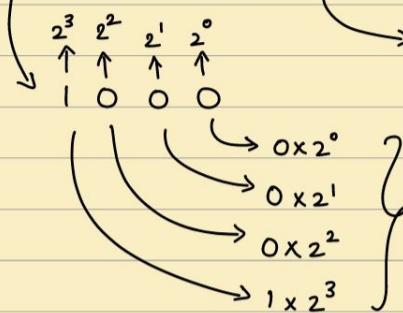
Codes

Weighted

Ex. BCD

Non-weighted

Ex. Gray



0011 to Binary : 2
(No weights)

Weights are assigned to bits, i.e. Positional weights
AKA 8421 BCD

Note : 2421 BCD exists

↳ Weighted but weights are different

* Excess 3 code:

↪ Non weighted

Ex. Excess 3 code of 1 is 4.

$$\begin{array}{r} 0001 \\ 0011 \\ \hline 0100 \end{array}$$

- Excess 3 of 18 :

• Addition :

+3 +3

1 8

0001 1000

Ex. Addition

$$\begin{array}{r} 43 \\ 82 \\ \hline ? \end{array} \quad \begin{array}{l} \text{Ans}_{BCD} = 125 \\ \text{Ans}_{E3} = 158 \end{array}$$

$$\begin{array}{r}
 \underline{\text{BCD}} \\
 + 0100 \\
 \hline
 1000
 \end{array}
 \quad
 \begin{array}{r}
 \underline{\text{BCD}} \\
 + 0011 \\
 \hline
 0010
 \end{array}$$

$$\begin{array}{r}
 \downarrow \\
 1100
 \end{array}
 \quad
 \begin{array}{r}
 \downarrow \\
 0101
 \end{array}$$

$$\begin{array}{r}
 (+6) \quad 0110 \\
 \hline
 0010
 \end{array}
 \quad
 \begin{array}{r}
 0110 \\
 \hline
 0101 \quad (+6)
 \end{array}$$

$$\begin{array}{r}
 \textcircled{T} \\
 \downarrow +3 \\
 \underline{\text{E3}} \\
 0101
 \end{array}
 \quad
 \begin{array}{r}
 \textcircled{O} \\
 \downarrow -3 \\
 \underline{\text{E3}} \\
 1000
 \end{array}
 \Rightarrow 125$$

$$\begin{array}{r}
 + 18 \\
 + 13 \\
 \hline
 0001 \\
 0001 \\
 \hline
 0011
 \end{array}
 \quad
 \begin{array}{r}
 1000 \\
 0011 \\
 \hline
 1011
 \end{array}$$

Ans (BCD) : 31

Ans (E3) : 64

$\downarrow \text{BCD} \rightarrow \text{Excess 3}$ $\downarrow \text{BCD} \rightarrow \text{Excess 3}$

$$\begin{array}{r}
 0011 \\
 0110 \quad (+6) \\
 \hline
 1001
 \end{array}
 \quad
 \begin{array}{r}
 1011 \\
 0110 \quad (+6) \\
 \hline
 0001
 \end{array}$$
 $\therefore \text{Carry} \rightarrow +3$

$$\begin{array}{r}
 + 0011 \\
 \hline
 0100
 \end{array}
 \quad
 \textcircled{4} \quad (E3)$$

$\textcircled{6} \rightarrow (E3)$

$\text{BCD} : 4 - 3 = 1$

$\text{BCD} : 6 - 3 = \underline{\underline{3}}$

$\therefore \underline{\underline{31}}$

$[1101 : 2^5 \text{ complement of } 3]$

[1101 : 2^s complement of 3]

Carry = 0 \Rightarrow Subtract 3

Carry = 1 \Rightarrow Add 3

→ Converter:

I/P → BCD				Excess - 3				$E_0 = \overline{B}_0$
B_3	B_2	B_1	B_0	E_3	E_2	E_1	E_0	
0	0	0	0	0	0	1	1	→ 0
0	0	0	1	0	1	0	0	→ 1
0	0	1	0	0	1	0	1	→ 2
0	0	1	1	0	1	1	0	→ 3
0	1	0	0	0	1	1	1	→ 4
0	1	0	1	1	0	0	0	→ 5
0	1	1	0	1	0	0	1	→ 6
0	1	1	1	1	0	1	0	→ 7
1	0	0	0	1	0	1	1	→ 8
1	0	0	1	1	1	0	0	→ 9

This is Self Complementary, i.e.
whatever numbers add upto 9
is BCD, their Excess 3 codes
are bitwise complements to
each other.

		E_2						E_1					
		$B_3 B_2$	00	01	11	10			$B_3 B_2$	00	01	11	10
			00	0 ₀	1 ₁	1 ₃	1 ₂			1 ₀	0 ₁	1 ₃	0 ₂
			01	1 ₄	0 ₅	0 ₇	0 ₆			1 ₄	0 ₅	1 ₇	0 ₆
			11	X ₁₂	X ₁₃	X ₁₅	X ₁₄			X ₁₂	X ₁₃	X ₁₅	X ₁₄
			10	0 ₈	1 ₉	X ₁₁	X ₁₀			1 ₈	0 ₉	X ₁₁	X ₁₀

		E_0						E_3					
		$B_3 B_2$	00	01	11	10			$B_3 B_2$	00	01	11	10
			00	1 ₀	0 ₁	0 ₃	1 ₂			0 ₀	0 ₁	0 ₃	0 ₂
			01	1 ₄	0 ₅	0 ₇	1 ₆			0 ₄	1 ₅	1 ₇	1 ₆
			11	X ₁₂	X ₁₃	X ₁₅	X ₁₄			X ₁₂	X ₁₃	X ₁₅	X ₁₄
			10	1 ₈	0 ₉	X ₁₁	X ₁₀			1 ₈	1 ₉	X ₁₁	X ₁₀

Note:

Every Algo is Sequential Nature.

$$E_0 = \overline{B}_0$$

$$E_1 = B_0 B_1 + \overline{B}_0 \overline{B}_1$$

$$E_2 = \overline{B}_0 \overline{B}_1 B_2 + B_0 \overline{B}_2 + B_1 \overline{B}_2$$

$$E_3 = B_3 + B_0 B_2 + B_1 B_2$$

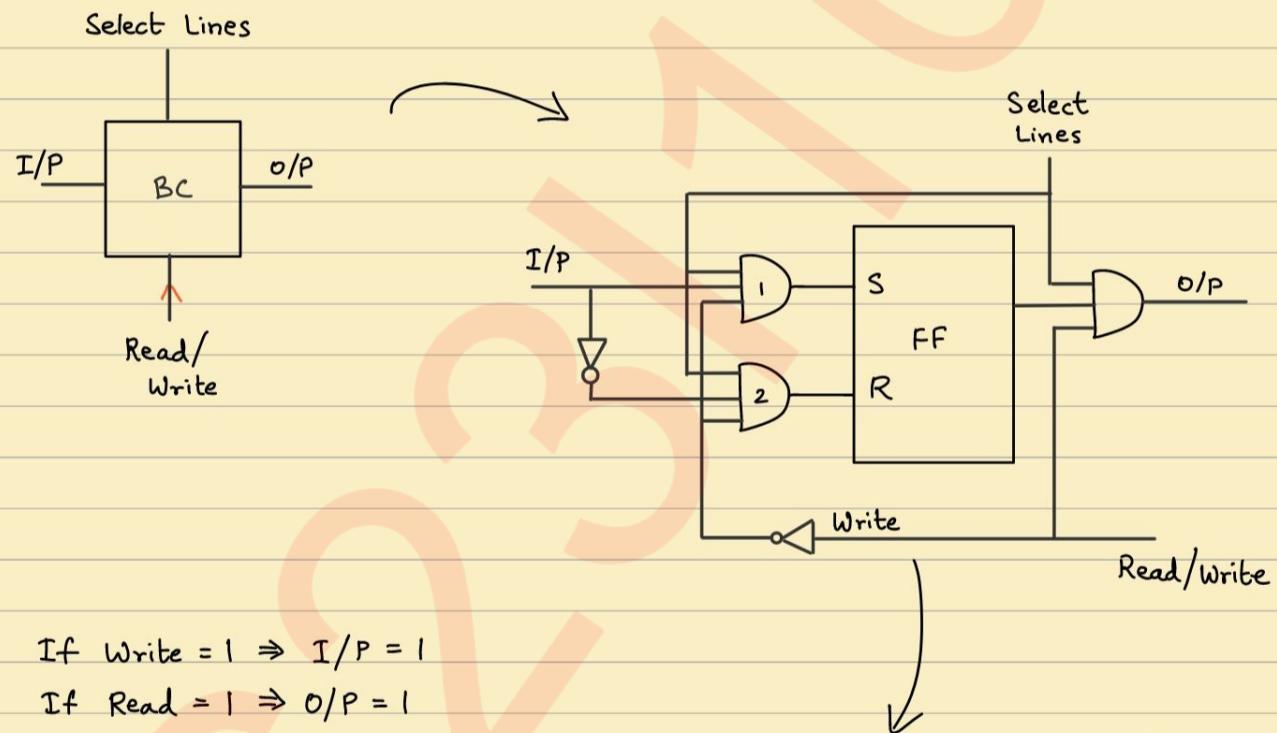
13/11

RAM - Random Access Memory \longrightarrow Volatile : Once powered off, the data is supposed to be erased

Non - Volatile



\rightarrow Basic Cell / Binary Cell:



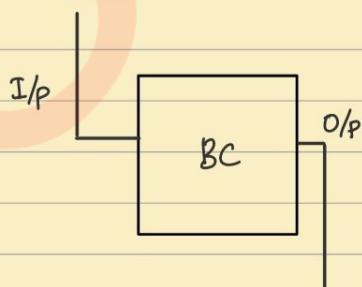
If Write = 1 \Rightarrow I/P = 1

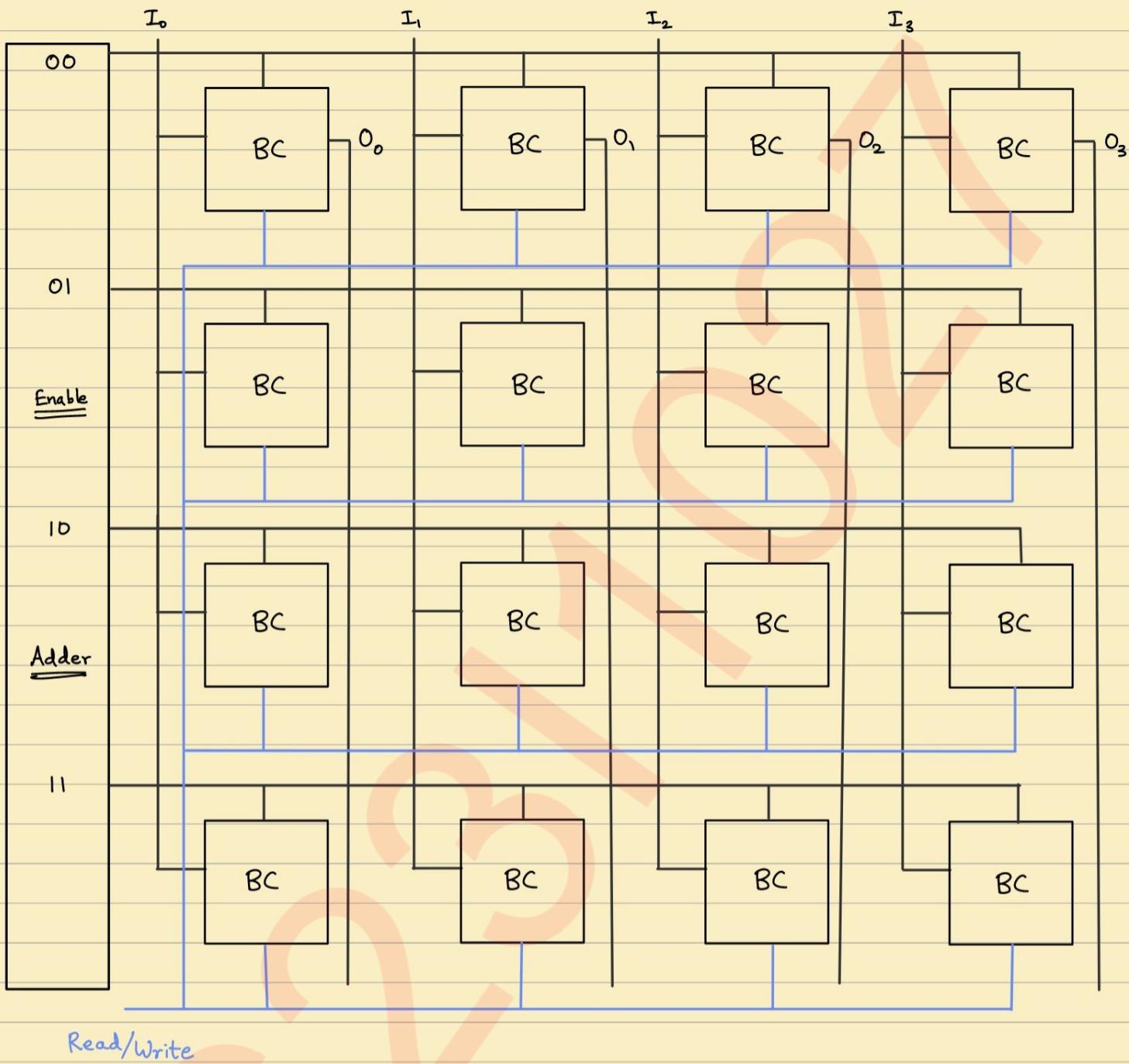
If Read = 1 \Rightarrow O/P = 1

Mentioned as a
Latch but acts
as a flip flop.

0 = Write
1 = Read

Smallest Addressable value of unit - Word
 \hookrightarrow 4 bit





At one time, only one binary cell will be enabled (in any column)

k : No. of bits in select lines

Address

w : word size

$$\text{no. of words} = 2^k$$

$$\text{Size} : 2^k \times w$$

Hardware:

Q1) Make a counter in $1 \rightarrow 2 \rightarrow 3 \rightarrow 2 \rightarrow 1 \rightarrow 2 \dots$ manner.

Ask user for 2 bits

When counter = 1 \Rightarrow Add State (+1)

When counter = 2 \Rightarrow Subtract State (-2)

When counter = 3 \Rightarrow Bitwise XOR

Solution:

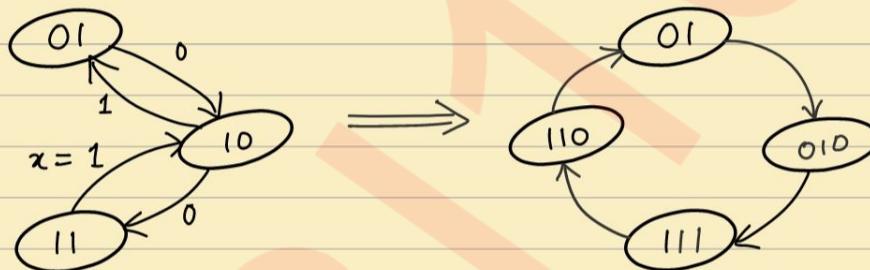
(i) Sequential Logic

$x \rightarrow$ Manual

To automate x , Take x as another flip flop.

Q : Indirectly converting Mealy Machine to Moore Machine

i.e.



Current			Next			T_A	T_B	T_C
A	B	C	A	B	C			
0	0	1	0	1	0	0	1	1
0	1	0	0	1	1	0	0	1
0	1	1	1	1	0	1	0	1
1	1	0	0	0	1	1	1	1

A	BC			T_A
	00	01	11	
0	X ₀	0 ₁	1 ₃	0 ₂
1	X ₄	X ₅	X ₇	1 ₆

$$T_A = BC + A$$

A	BC			T_B
	00	01	11	
0	X ₀	1 ₁	0 ₃	0 ₂
1	X ₄	X ₅	X ₇	1 ₆

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

A	BC			T_C
	00	01	11	
0	X ₀	1 ₁	1 ₃	1 ₂
1	X ₄	X ₅	X ₇	1 ₆

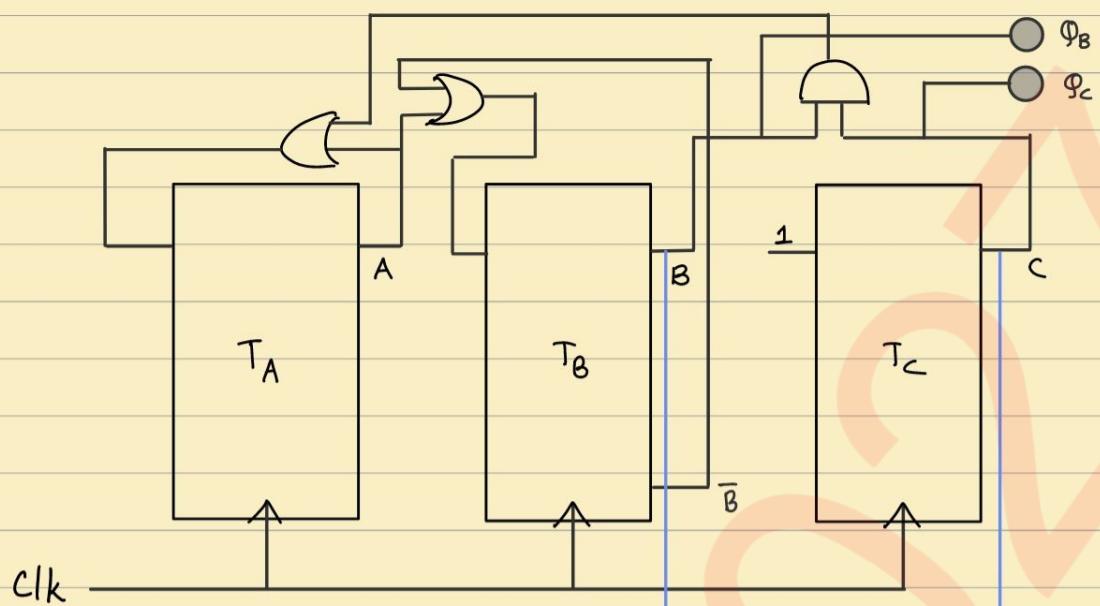
$$T_C = 1$$

Advantage of Converting : Automation

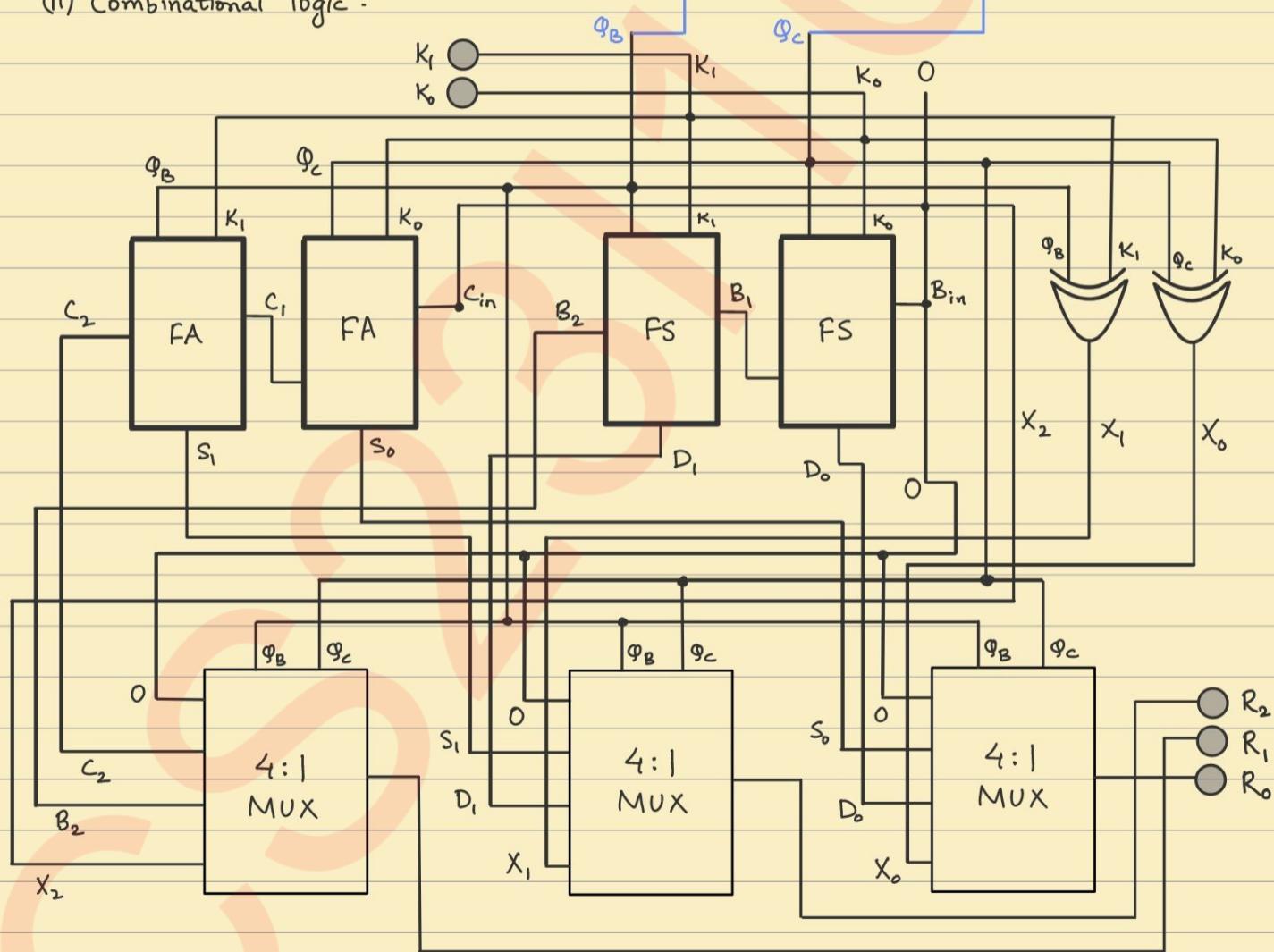
Disadvantage : No. of states increase tremendously &
Inputs change

Circuit Diagram for counter:

$$\begin{aligned} T_A &= Q_A + Q_B \cdot Q_C \\ T_B &= Q_A + \overline{Q}_B \\ T_C &= 1 \end{aligned}$$

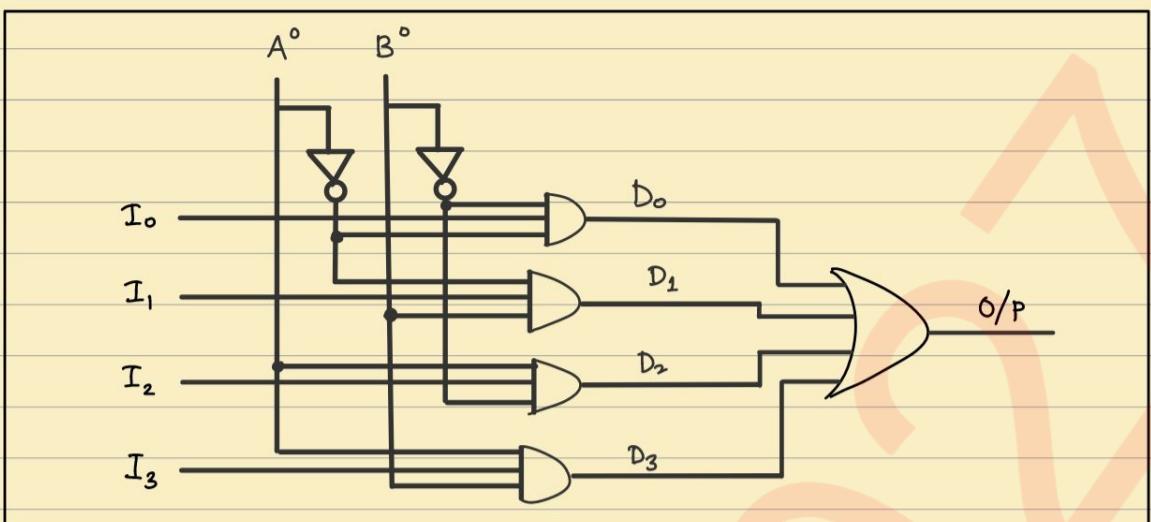


(ii) Combinational logic :

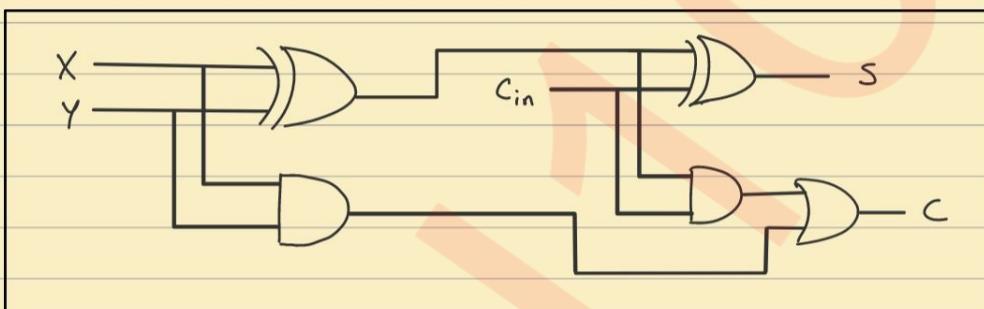


MUX {

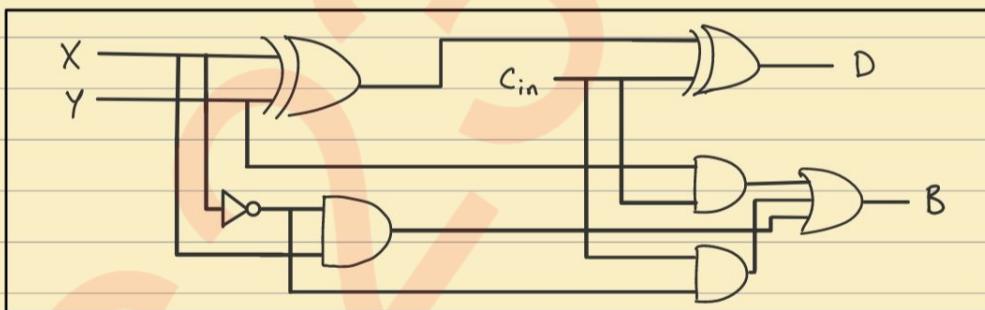
- 00 - Nothing (I_0)
- 01 - Add (I_1)
- 10 - Subtract (I_2)
- 11 - XOR (I_3)



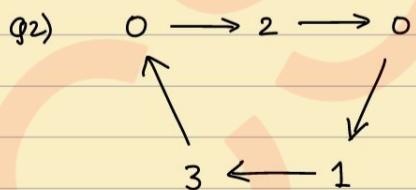
4:1 MUX module



FA module



FS module



K: User Input - $K_1 K_0$
 Low to High: Add K
 High to Low: Subtract K

Solution:

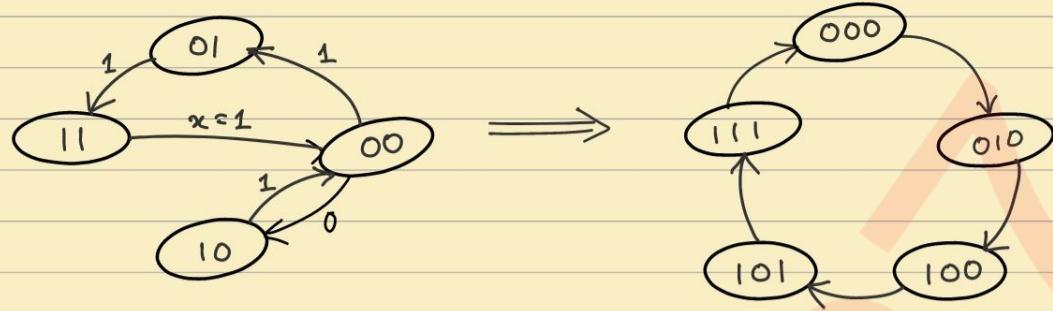
Method - (I) :

$x \rightarrow$ Manual

To automate x , Take x as another Flip flop.

Q : Indirectly converting Mealy Machine to Moore Machine

i.e.



Current			Next			T_A	T_B	T_C
A	B	C	A	B	C			
0	0	0	0	1	0	0	1	0
0	1	0	1	0	0	1	1	0
1	0	0	1	0	1	0	0	1
1	0	1	1	1	1	0	1	0
1	1	1	0	0	0	1	1	1

A	BC			T_A				
	00	01	11	10	00	01	11	10
0	0	X	X	1	0	0	1	0
1	0	0	1	X	1	0	0	1

$$T_A = B$$

A	BC			T_B				
	00	01	11	10	00	01	11	10
0	1	X	X	1	0	1	0	1
1	0	1	1	X	1	0	1	X

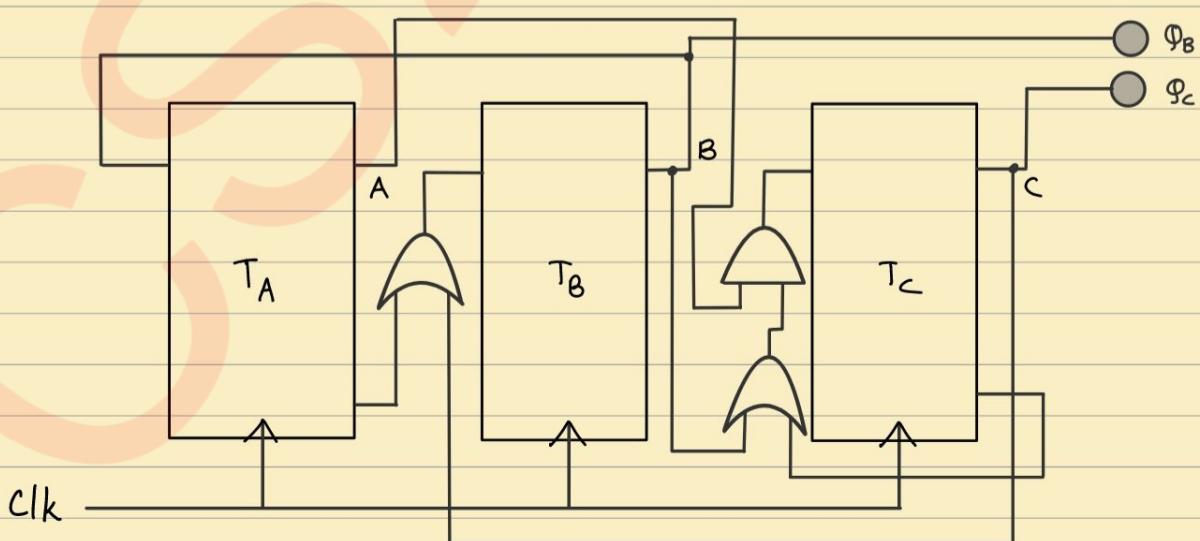
$$T_B = \bar{A} + C$$

A	BC			T_C				
	00	01	11	10	00	01	11	10
0	0	X	X	0	1	0	1	0
1	1	0	1	X	1	0	1	X

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

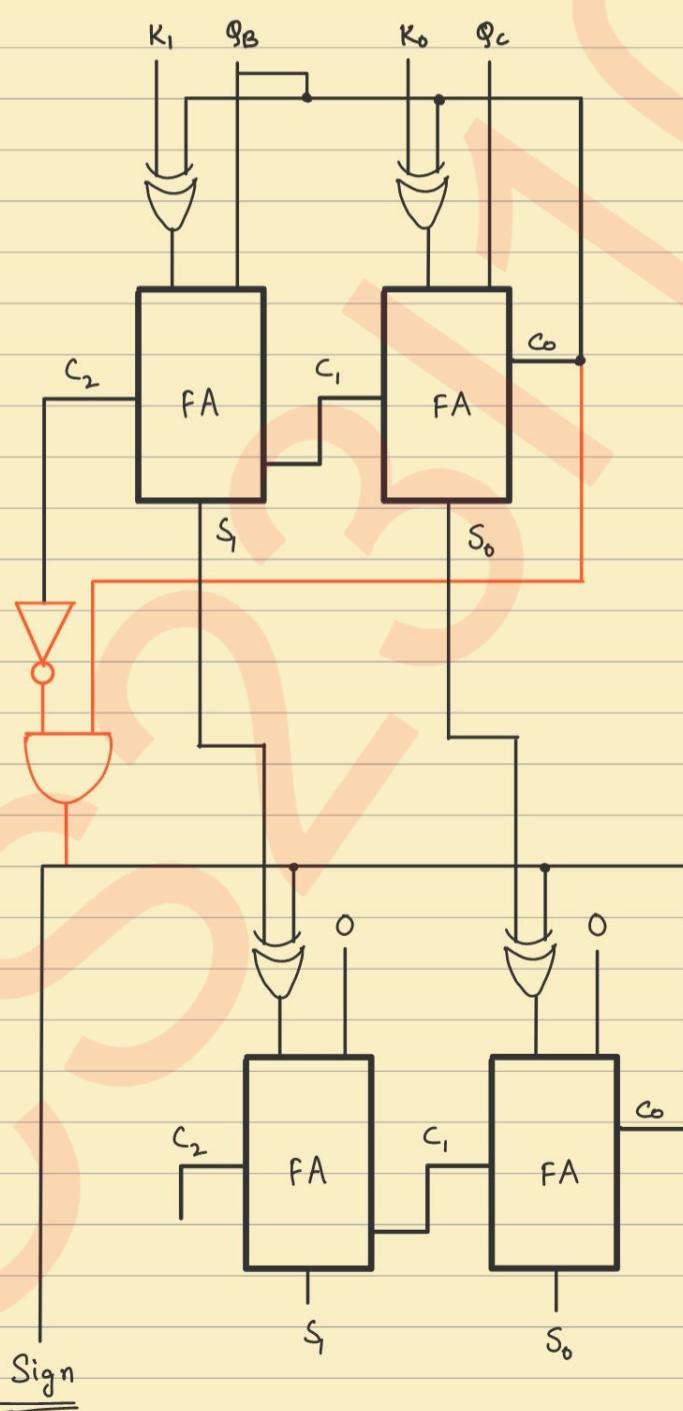
$$\begin{aligned} T_A &= \Phi_B \\ T_B &= \overline{\Phi_A} + \Phi_C \\ T_C &= \Phi_A (\Phi_B + \overline{\Phi_C}) \end{aligned}$$

Circuit Diagram for counter:



Combinational component :

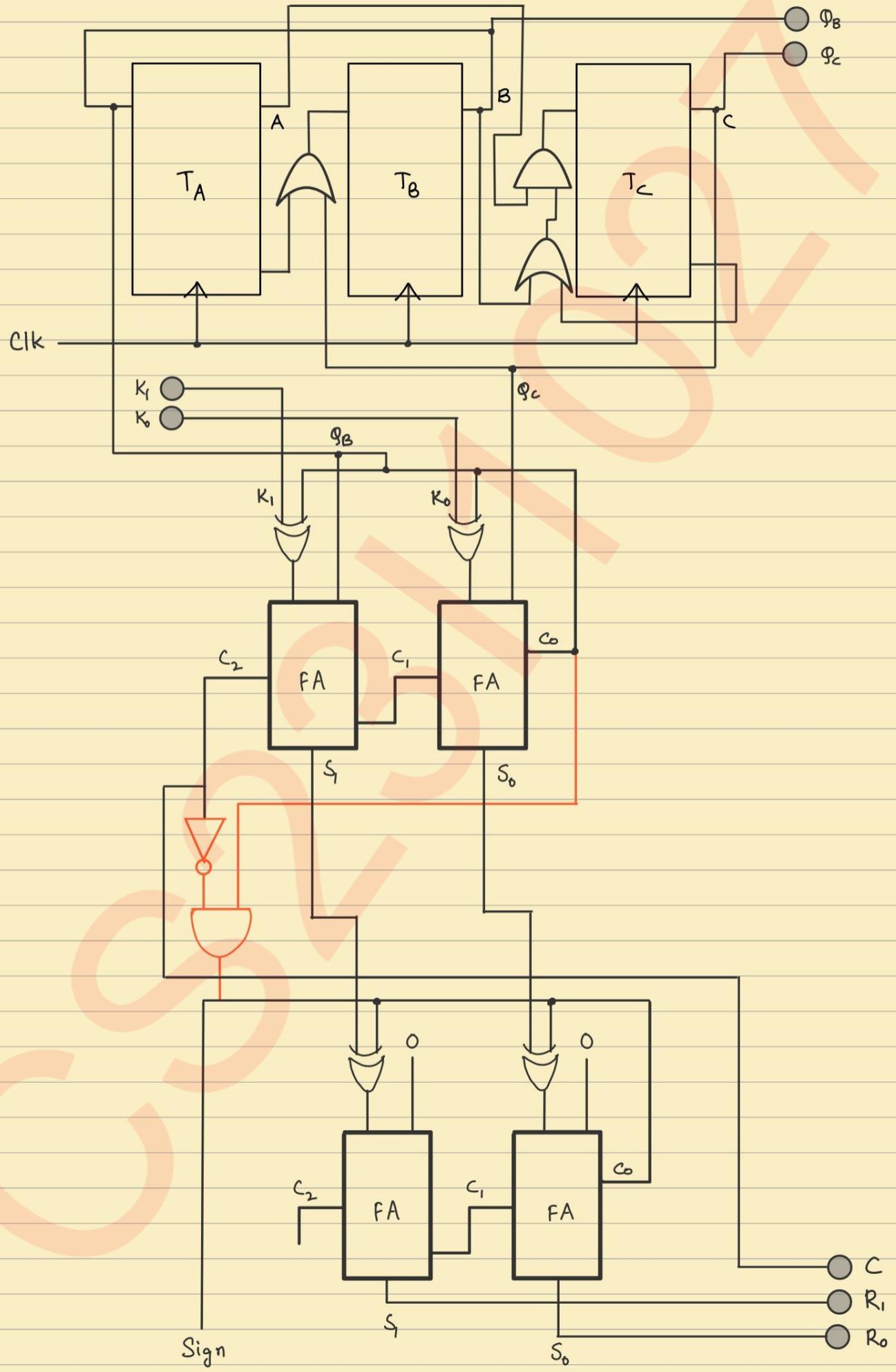
Current			Next			Result	CTR
A	B	C	A	B	C		
0	0	0	0	1	0	$(BC) + K$	0
0	1	0	1	0	0	$(BC) - K$	1
1	0	0	1	0	1	$(BC) + K$	0
1	0	1	1	1	1	$(BC) + K$	0
1	1	1	0	0	0	$(BC) - K$	1



A	CTR				
	BC	00	01	11	10
0	0	X	X	1	2
1	0	0	1	X	6

$$CTR = B = T_A$$

Linking Sequential & Combinational components :



Software

(Q1) 4 Circuits : Down Counter - 4 bit

Excess 3 converter , whose input is taken from the current output
(consider BCD Input) - 4 bit

Ring Counter - 4 bit

Retain Value - 4 bit

Do not use extra flip-flops , i.e. use only 4 flip flops.

Any one will be activated at a time using switch.

(Q2) 4 Circuits : Up Counter - 4 bit

Excess 3 converter , whose input is taken from the current output
(consider BCD Input) - 4 bit

Johnson Counter - 4 bit

Retain Value - 4 bit

Do not use extra flip-flops , i.e. use only 4 flip flops.

Any one will be activated at a time using switch.

P.T.O

Down Counter:

Current				Next							
Q_3	Q_2	Q_1	Q_0	Q_3	Q_2	Q_1	Q_0	D_3	D_2	D_1	D_0
1	1	1	1	1	1	1	0	1	1	1	0
1	1	1	0	1	1	0	1	1	0	0	1
1	1	0	1	1	1	0	0	1	1	0	0
1	1	0	0	1	0	1	1	1	0	1	1
1	0	1	1	1	0	1	0	1	0	1	0
1	0	1	0	1	0	0	1	1	0	0	1
1	0	0	1	1	0	0	0	1	0	0	0
1	0	0	0	0	1	1	1	0	1	1	1
0	1	1	1	0	1	1	0	0	1	1	0
0	1	1	0	0	1	0	1	0	1	0	1
0	1	0	1	0	1	0	0	0	1	0	0
0	1	0	0	0	0	1	1	0	0	1	1
0	0	1	1	0	0	1	0	0	0	1	0
0	0	1	0	0	0	0	1	0	0	0	1
0	0	0	1	0	0	0	0	0	0	0	0
0	0	0	0	1	1	1	1	1	1	1	1

Q_t	Q_{t+1}	D
0	0	0
0	1	1
1	0	0
1	1	1

$$D = Q_{t+1}$$

$$D_0 = \bar{Q}_0$$

		D_2						D_1					
		$Q_3 Q_0$	00	01	11	10			$Q_3 Q_0$	00	01	11	10
$Q_3 Q_2$	$Q_2 Q_1$	00	1 ₀	1 ₁	0 ₃	1 ₂			00	1 ₀	0 ₁	1 ₃	0 ₂
00	00	00	1 ₀	1 ₁	0 ₃	1 ₂	01	01	00	1 ₀	0 ₁	1 ₃	0 ₂
01	01	01	0 ₄	0 ₅	1 ₇	0 ₆	11	11	10	1 ₄	0 ₅	1 ₇	0 ₆
11	11	11	0 ₁₂	0 ₁₃	1 ₁₅	0 ₁₄	10	10	10	1 ₈	0 ₉	1 ₁₁	0 ₁₀
10	10	10	1 ₈	1 ₉	0 ₁₁	1 ₁₀							

$$D_2 = \bar{Q}_2(\bar{Q}_0 + \bar{Q}_1) + Q_0 Q_1 Q_2$$

$$D_1 = \bar{Q}_1 \bar{Q}_0 + Q_0 Q_1$$

		D_3						
		$Q_3 Q_0$	00	01	11	10		
$Q_3 Q_2$	$Q_2 Q_1$	00	1 ₀	1 ₁	1 ₃	1 ₂		
00	00	00	1 ₀	1 ₁	1 ₃	1 ₂	01	01
01	01	01	1 ₄	1 ₅	0 ₇	1 ₆	11	11
11	11	11	0 ₁₂	0 ₁₃	1 ₁₅	0 ₁₄	10	10
10	10	10	0 ₈	0 ₉	0 ₁₁	0 ₁₀		

$$D_3 = \bar{Q}_3(\bar{Q}_0 + \bar{Q}_1 + \bar{Q}_2) + Q_0 Q_1 Q_2 Q_3$$

$$\begin{aligned} D_0 &= \bar{Q}_0 \\ D_1 &= \bar{Q}_1 \bar{Q}_0 + Q_0 Q_1 \\ D_2 &= \bar{Q}_2(\bar{Q}_0 + \bar{Q}_1) + Q_0 Q_1 Q_2 \\ D_3 &= \bar{Q}_3(\bar{Q}_0 + \bar{Q}_1 + \bar{Q}_2) + Q_0 Q_1 Q_2 Q_3 \end{aligned}$$

Excess - 3 :

B_3	B_2	B_1	B_0	E_3	E_2	E_1	E_0
0	0	0	0	0	0	1	1
0	0	0	1	0	1	0	0
0	0	1	0	0	1	0	1
0	0	1	1	0	1	1	0
0	1	0	0	0	1	1	1
0	1	0	1	1	0	0	0
0	1	1	0	1	0	0	1
0	1	1	1	1	0	1	0
1	0	0	0	1	0	1	1
1	0	0	1	1	1	0	0

$$E_0 = \overline{B_0}$$

$$E_1 = B_0 B_1 + \overline{B_0} \overline{B_1}$$

$$E_2 = \overline{B_0} \overline{B_1} B_2 + B_0 \overline{B}_2 + B_1 \overline{B}_2$$

$$E_3 = B_3 + B_0 B_2 + B_1 B_2$$

		E_2				
		00	01	11	10	
$B_3 B_2$	$B_1 B_0$	00	0	1	1	1
00	00	0	1	1	1	1
01	01	1	0	0	0	0
11	11	X	X	X	X	X
10	10	0	1	X	X	X

		E_1				
		00	01	11	10	
$B_3 B_2$	$B_1 B_0$	00	1	0	1	0
00	00	1	0	1	0	0
01	01	1	0	1	0	0
11	11	X	X	X	X	X
10	10	1	0	X	X	X

		E_0				
		00	01	11	10	
$B_3 B_2$	$B_1 B_0$	00	1	0	0	1
00	00	1	0	0	1	1
01	01	1	0	0	1	0
11	11	X	X	X	X	X
10	10	1	0	X	X	X

		E_3				
		00	01	11	10	
$B_3 B_2$	$B_1 B_0$	00	0	0	0	0
00	00	0	0	0	0	0
01	01	0	1	1	1	0
11	11	X	X	X	X	X
10	10	1	1	X	X	X

Retain :

Current				Next							
Q_3	Q_2	Q_1	Q_0	Q_3	Q_2	Q_1	Q_0	D_3	D_2	D_1	D_0
1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	0	1	1	1	0	1	1	1	0

$$D_0 = Q_0$$

$$D_1 = Q_1$$

$$D_2 = Q_2$$

$$D_3 = Q_3$$

Ring Counter:

Current				Next							
Q_3	Q_2	Q_1	Q_0	Q_3	Q_2	Q_1	Q_0	D_3	D_2	D_1	D_0
1	0	0	0	0	1	0	0	0	1	0	0
0	1	0	0	0	0	1	0	0	0	1	0
0	0	1	0	0	0	0	1	0	0	0	1
0	0	0	1	1	0	0	0	1	0	0	0

Q_t	Q_{t+1}	D
0	0	0
0	1	1
1	0	0
1	1	1

$$D = Q_{t+1}$$

D_3

$Q_3 Q_2$	00	01	11	10
$Q_1 Q_0$	X	1	X ₃	0 ₂
00
01	0 ₄	X ₅	X ₇	X ₆
11	X ₁₂	X ₁₃	X ₁₅	X ₁₄
10	0 ₈	X ₉	X ₁₁	X ₁₀

D_2

$Q_3 Q_2$	00	01	11	10
$Q_1 Q_0$	X	0	X ₃	0 ₂
00
01	0 ₄	X ₅	X ₇	X ₆
11	X ₁₂	X ₁₃	X ₁₅	X ₁₄
10	1 ₈	X ₉	X ₁₁	X ₁₀

$$D_3 = Q_0$$

$$D_2 = Q_3$$

D_1

$Q_3 Q_2$	00	01	11	10
$Q_1 Q_0$	X	0	X ₃	0 ₂
00
01	1 ₄	X ₅	X ₇	X ₆
11	X ₁₂	X ₁₃	X ₁₅	X ₁₄
10	0 ₈	X ₉	X ₁₁	X ₁₀

D_0

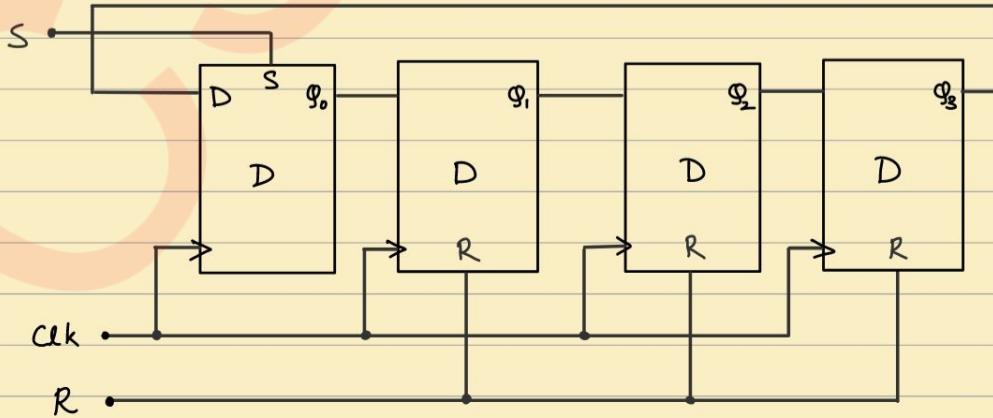
$Q_3 Q_2$	00	01	11	10
$Q_1 Q_0$	X	0	X ₃	1 ₂
00
01	0 ₄	X ₅	X ₇	X ₆
11	X ₁₂	X ₁₃	X ₁₅	X ₁₄
10	0 ₈	X ₉	X ₁₁	X ₁₀

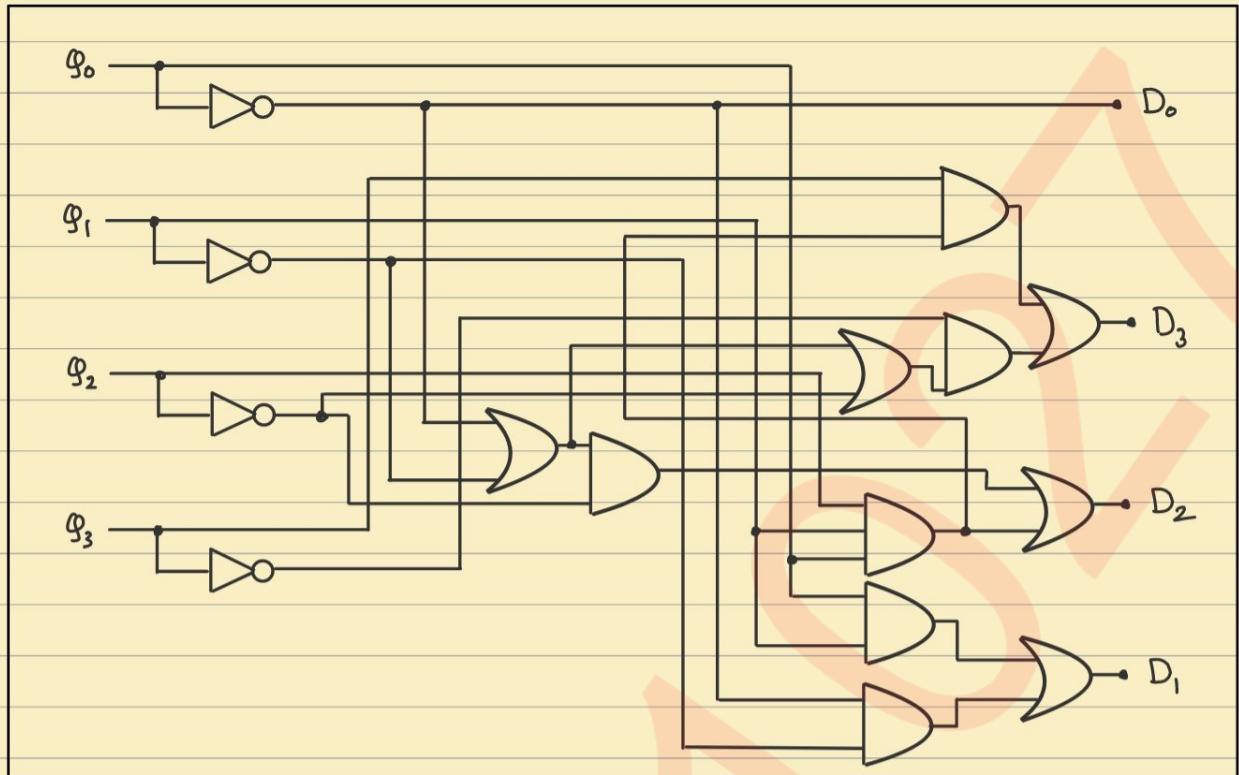
$$D_1 = Q_2$$

$$D_0 = Q_1$$

Ha @ 5Ri #1 Th & 0

$$\begin{aligned} D_0 &= Q_1 \\ D_1 &= Q_2 \\ D_2 &= Q_3 \\ D_3 &= Q_0 \end{aligned}$$





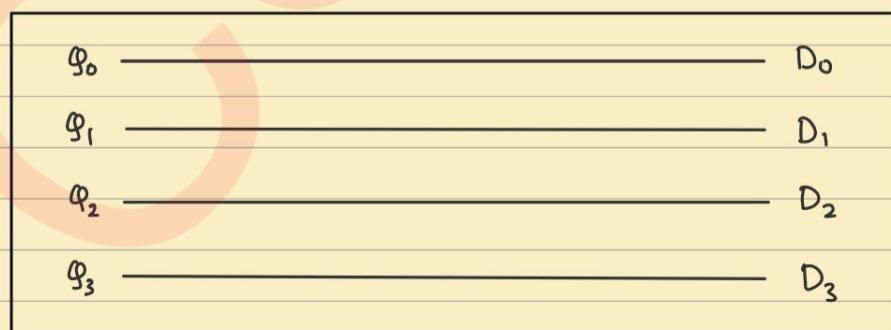
Down - Counter Module :
(DC)

$$\begin{aligned}
 D_0 &= \bar{Q}_0 \\
 D_1 &= \bar{Q}_1 \bar{Q}_0 + Q_0 Q_1 \\
 D_2 &= \bar{Q}_2 (\bar{Q}_0 + \bar{Q}_1) + Q_0 Q_1 Q_2 \\
 D_3 &= \bar{Q}_3 (\bar{Q}_0 + \bar{Q}_1 + \bar{Q}_2) + Q_0 Q_1 Q_2 Q_3
 \end{aligned}$$



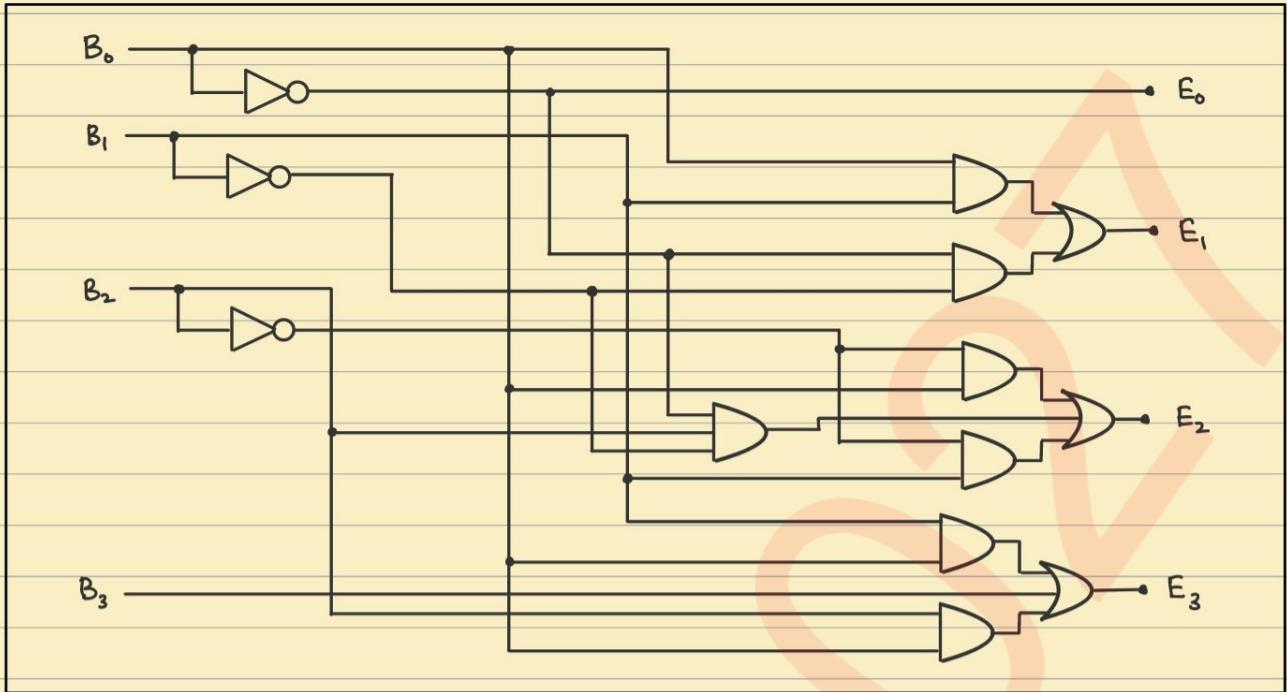
$$\begin{aligned}
 D_3 &= Q_0 \\
 D_2 &= Q_3 \\
 D_1 &= Q_2 \\
 D_0 &= Q_1
 \end{aligned}$$

Ring Counter (RC) Module



$$\begin{aligned}
 D_0 &= Q_0 \\
 D_1 &= Q_1 \\
 D_2 &= Q_2 \\
 D_3 &= Q_3
 \end{aligned}$$

Retain (R) Module



Excess-3 Module :
(E3)

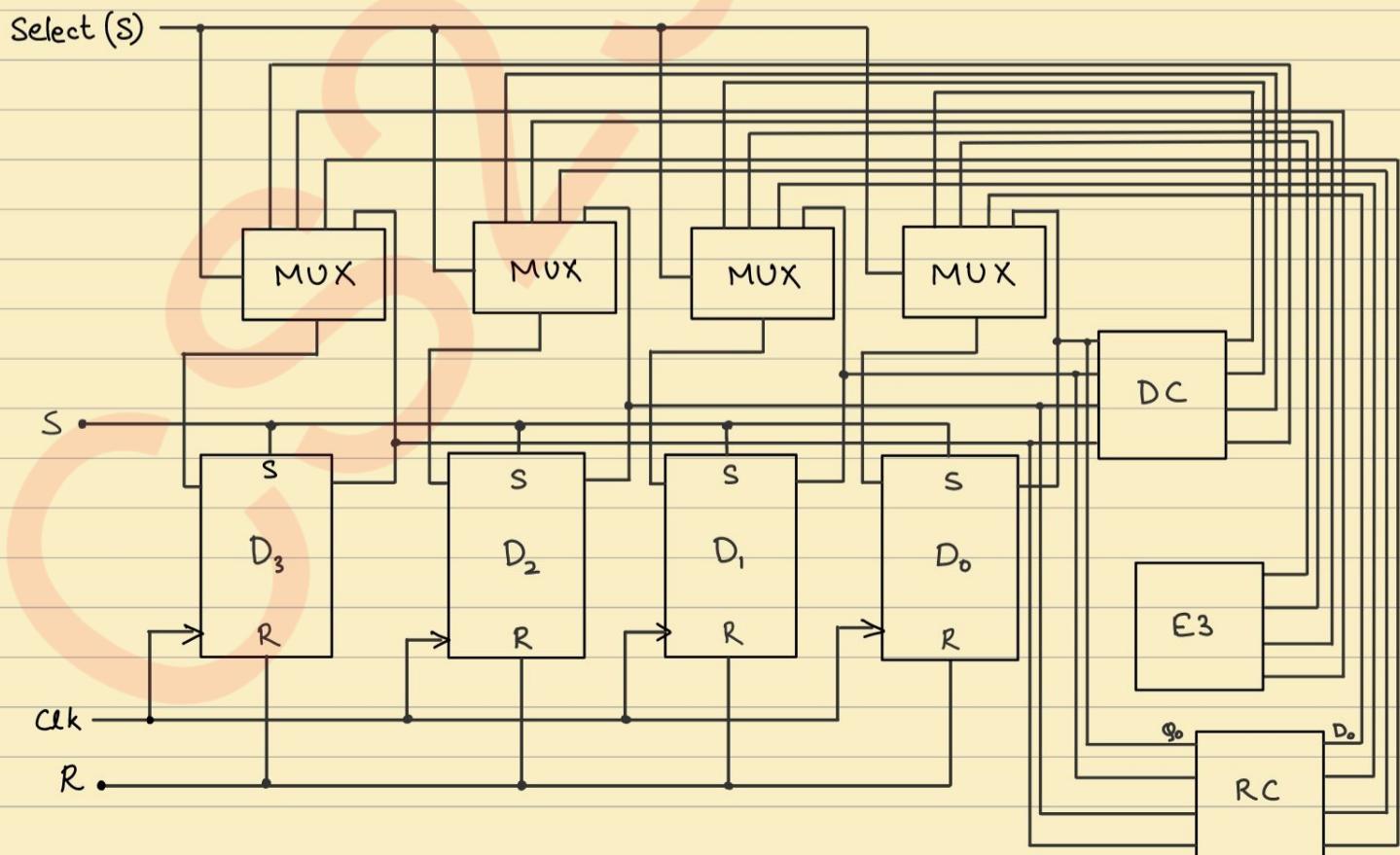
$$E_0 = \overline{B}_0$$

$$E_1 = B_0 \overline{B}_1 + \overline{B}_0 \overline{B}_1$$

$$E_2 = \overline{B}_0 \overline{B}_1 B_2 + B_0 \overline{B}_2 + B_1 \overline{B}_2$$

$$E_3 = B_3 + B_0 B_2 + B_1 B_2$$

Circuit Diagram : (Hint)



Up Counter:

Current				Next							
Q_3	Q_2	Q_1	Q_0	Q_3	Q_2	Q_1	Q_0	D_3	D_2	D_1	D_0
0	0	0	0	0	0	0	1	0	0	0	1
0	0	0	1	0	0	1	0	0	0	1	0
0	0	1	0	0	0	1	1	0	0	1	1
0	0	1	1	0	1	0	0	0	1	0	0
0	1	0	0	0	1	0	1	0	1	0	1
0	1	0	1	0	1	1	0	0	1	1	0
0	1	1	0	0	1	1	1	0	1	1	1
0	1	1	1	1	0	0	0	1	0	0	0
1	0	0	0	1	0	0	1	1	0	0	1
1	0	0	1	1	0	1	0	1	0	1	0
1	0	1	0	1	0	1	1	0	1	1	1
1	0	1	1	1	1	0	0	1	0	0	0
1	1	0	0	1	1	0	1	1	1	0	1
1	1	0	1	1	1	1	0	1	1	1	1
1	1	1	0	1	1	1	1	0	0	0	0

Q_t	Q_{t+1}	D
0	0	0
0	1	1
1	0	0
1	1	1

$$D = Q_{t+1}$$

$$D_0 = \bar{Q}_0$$

		D_2				
		$Q_3 Q_2$	00	01	11	10
$Q_1 Q_0$		00	0 ₀	0 ₁	1 ₃	0 ₂
01		04	15	07	16	
11		112	113	015	114	
10		08	09	111	010	

		D_3				
		$Q_3 Q_2$	00	01	11	10
$Q_1 Q_0$		00	0 ₀	0 ₁	0 ₃	0 ₂
01		04	05	17	06	
11		112	113	015	114	
10		18	19	111	110	

$$D_2 = Q_2(\bar{Q}_0 + \bar{Q}_1) + Q_0 Q_1 \bar{Q}_2$$

$$D_3 = Q_3(Q_0 + Q_1 + Q_2) + Q_0 Q_1 Q_2 \bar{Q}_3$$

		D_1				
		$Q_3 Q_2$	00	01	11	10
$Q_1 Q_0$		00	0 ₀	1 ₁	0 ₃	1 ₂
01		04	15	07	16	
11		112	113	015	114	
10		08	19	011	110	

$$D_1 = Q_1 \bar{Q}_0 + Q_0 \bar{Q}_1$$

$$D_0 = \bar{Q}_0$$

$$D_1 = Q_1 \bar{Q}_0 + Q_0 \bar{Q}_1$$

$$D_2 = Q_2(\bar{Q}_0 + \bar{Q}_1) + Q_0 Q_1 \bar{Q}_2$$

$$D_3 = Q_3(Q_0 + Q_1 + Q_2) + Q_0 Q_1 Q_2 \bar{Q}_3$$

Excess - 3 :

B_3	B_2	B_1	B_0	E_3	E_2	E_1	E_0
0	0	0	0	0	0	1	1
0	0	0	1	0	1	0	0
0	0	1	0	0	1	0	1
0	0	1	1	0	1	1	0
0	1	0	0	0	1	1	1
0	1	0	1	1	0	0	0
0	1	1	0	1	0	0	1
0	1	1	1	1	0	1	0
1	0	0	0	1	0	1	1
1	0	0	1	1	1	0	0

$$E_0 = \overline{B_0}$$

$$E_1 = B_0 B_1 + \overline{B_0} \overline{B_1}$$

$$E_2 = \overline{B_0} \overline{B_1} B_2 + B_0 \overline{B_2} + B_1 \overline{B_2}$$

$$E_3 = B_3 + B_0 B_2 + B_1 B_2$$

Johnson Counter :

Current				Next				D			
Q_3	Q_2	Q_1	Q_0	Q_3	Q_2	Q_1	Q_0	D_3	D_2	D_1	D_0
0	0	0	0	1	0	0	0	1	0	0	0
1	0	0	0	1	1	0	0	1	1	0	0
1	1	0	0	1	1	1	0	1	1	1	0
1	1	1	0	1	1	1	1	1	1	1	1
1	1	1	1	0	1	1	1	0	1	1	1
0	1	1	1	0	0	1	1	0	0	1	1
0	0	1	1	0	0	0	1	0	0	0	1
0	0	0	1	0	0	0	0	0	0	0	0

Q_t	Q_{t+1}	D
0	0	0
0	1	1
1	0	0
1	1	1

$$D = Q_{t+1}$$

$$D_3 = \overline{Q_0}$$

$$D_2 = Q_3$$

$$D_1 = Q_2$$

$$D_0 = Q_1$$

Retain :

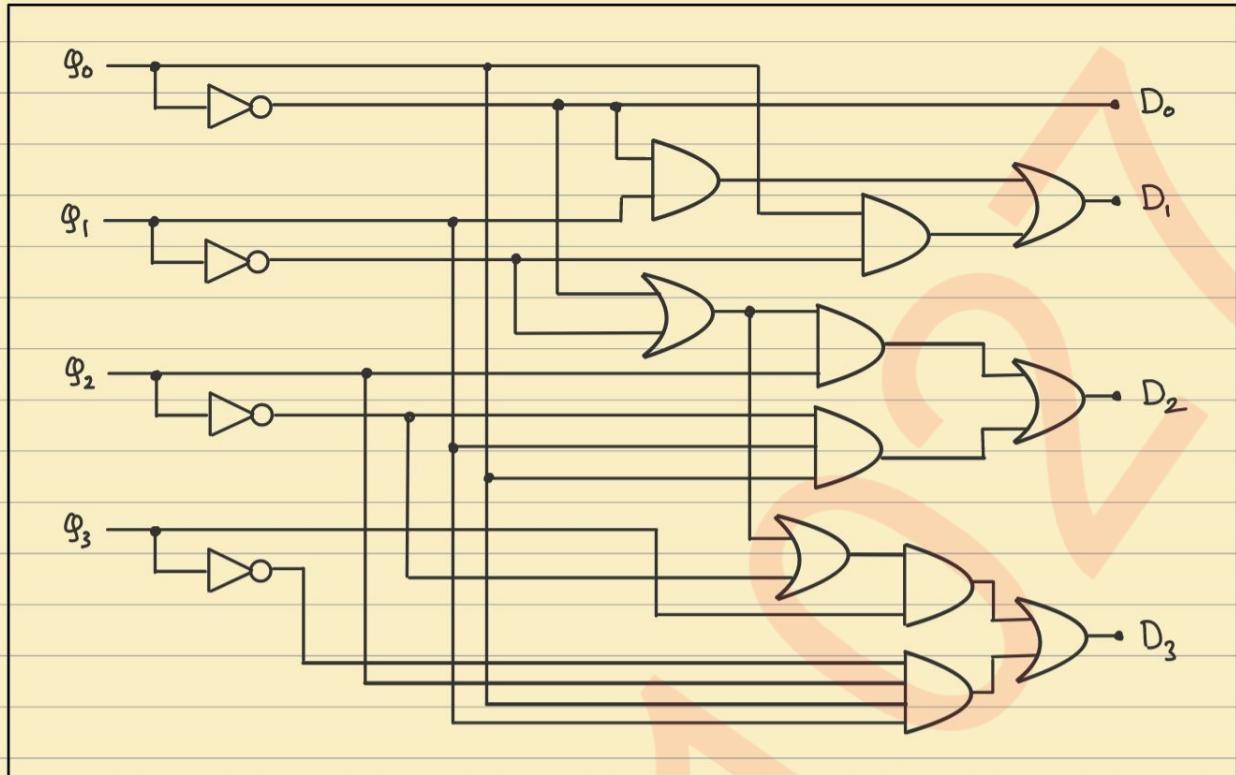
Current				Next				D			
Q_3	Q_2	Q_1	Q_0	Q_3	Q_2	Q_1	Q_0	D_3	D_2	D_1	D_0
1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	0	1	1	1	0	1	1	1	0

$$D_0 = Q_0$$

$$D_1 = Q_1$$

$$D_2 = Q_2$$

$$D_3 = Q_3$$



Up-Counter Module :
(UC)

$$\begin{aligned}
 D_0 &= \bar{Q}_0 \\
 D_1 &= Q_1 \bar{Q}_0 + \bar{Q}_0 \bar{Q}_1 \\
 D_2 &= Q_2 (\bar{Q}_0 + \bar{Q}_1) + Q_0 Q_1 \bar{Q}_2 \\
 D_3 &= Q_3 (\bar{Q}_0 + \bar{Q}_1 + \bar{Q}_2) + Q_0 Q_1 Q_2 \bar{Q}_3
 \end{aligned}$$



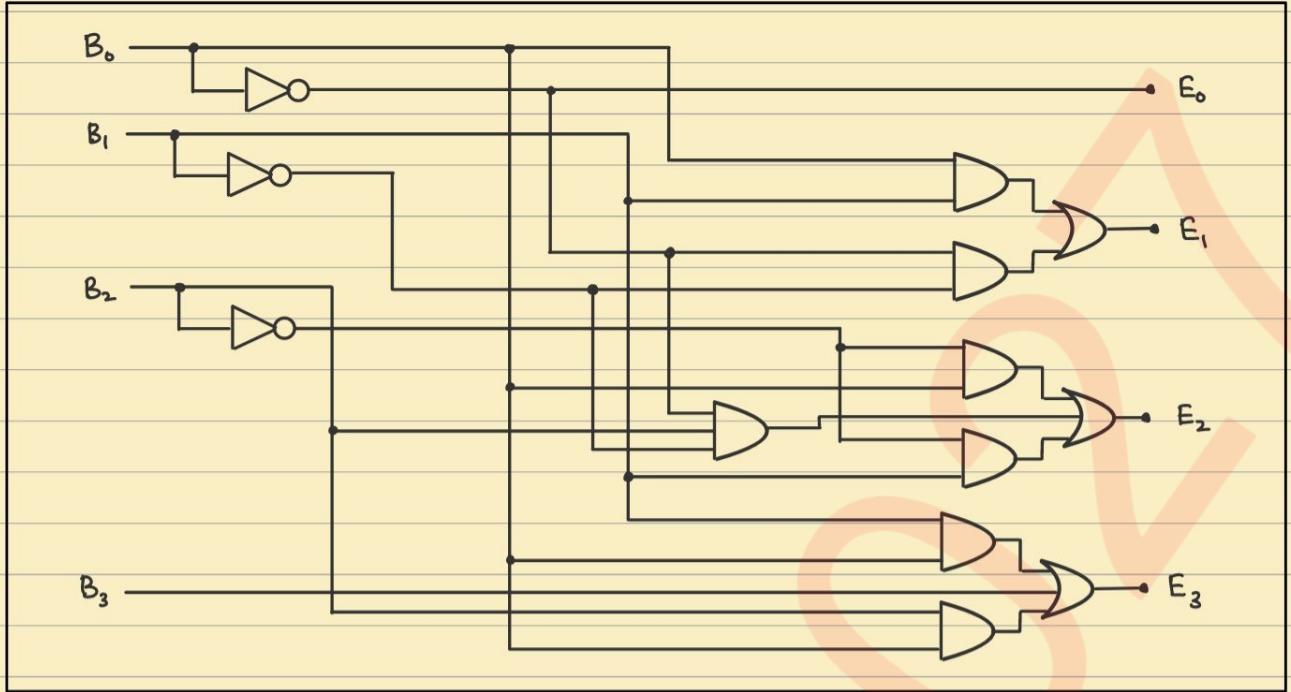
Johnson Counter (JC) Module

$$\begin{aligned}
 D_3 &= \bar{Q}_0 \\
 D_2 &= Q_3 \\
 D_1 &= Q_2 \\
 D_0 &= Q_1
 \end{aligned}$$



$$\begin{aligned}
 D_0 &= Q_0 \\
 D_1 &= Q_1 \\
 D_2 &= Q_2 \\
 D_3 &= Q_3
 \end{aligned}$$

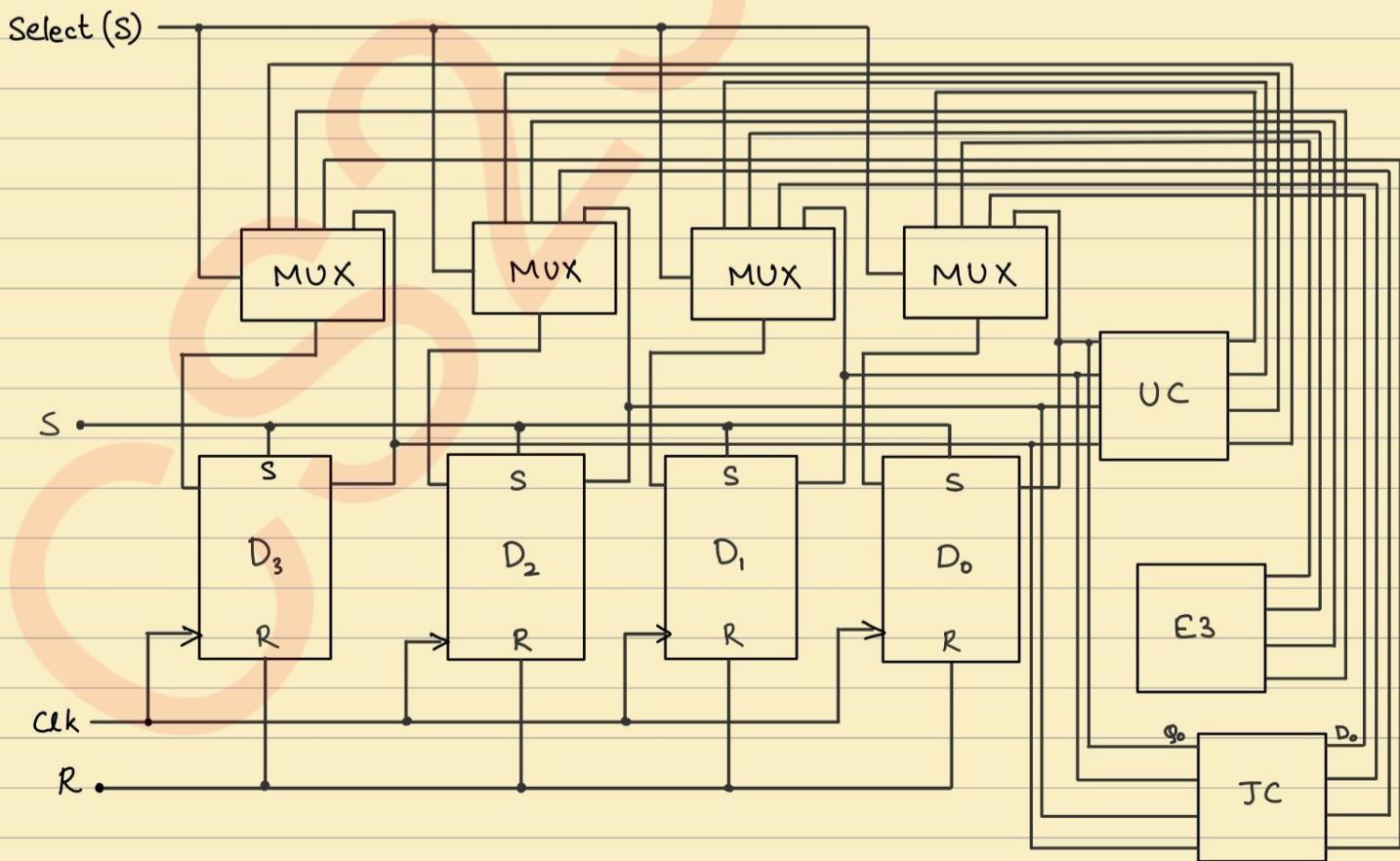
Retain (R) Module



→ Excess-3 Module :
(E3)

$$\begin{aligned}
 E_0 &= \overline{B}_0 \\
 E_1 &= B_0 B_1 + \overline{B}_0 \overline{B}_1 \\
 E_2 &= \overline{B}_0 \overline{B}_1 B_2 + B_0 \overline{B}_2 + B_1 \overline{B}_2 \\
 E_3 &= B_3 + B_0 B_2 + B_1 B_2
 \end{aligned}$$

Circuit Diagram : (Hint)



S - RAM : Static RAM

[COA course]

Basic Storage Medium

Word

Byte

 k : bits for address 2^k : values can be generated

↳ No. of words / address locations

Size of each word changes over time.

Ex. size \rightarrow 4 bytes, i.e. $w=4$

∴ 32 bits

$$k = 10$$

Each address will have

$$2^{10} \times 4 = 4096 \text{ bytes}$$

$$= 4 \text{ kilobytes}$$

$$\Rightarrow \text{Total Memory} = 4 \text{ kilobytes}$$

[1 kilobyte : 1024 bytes]

Q) Word Addressable system, word size = 2

Total Memory capacity = 32 kilobytes

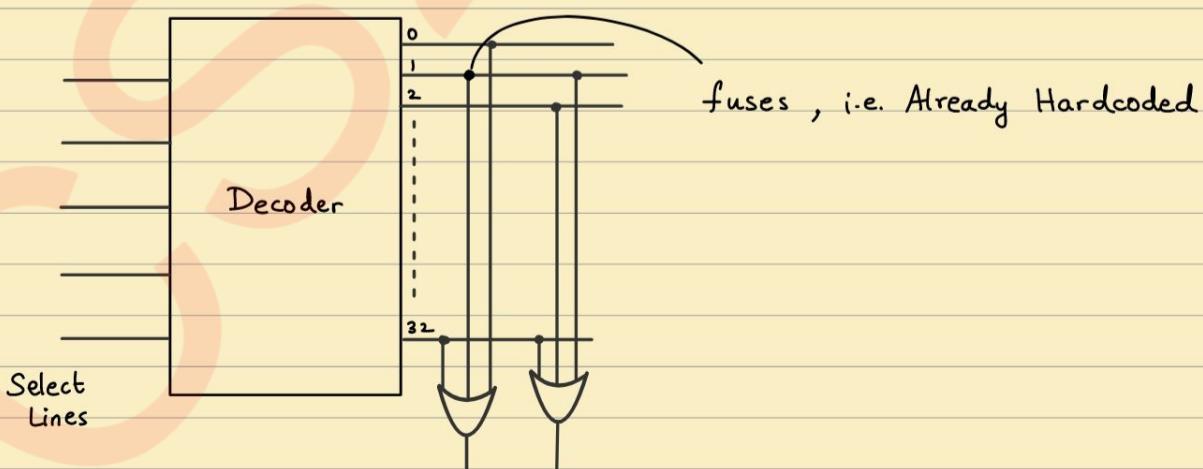
Sol: In bytes, $32 \times 2^{10} = 2 \times 2^k$ [$w=2$]

$$\Rightarrow 2^5 \times 2^{10} = 2^{k+1}$$

$$\Rightarrow 2^{k+1} = 2^5$$

$$k = 4$$

∴ 14 bits

$$\begin{cases} \text{CN course} - 10^3 [1000] \\ \text{COA course} - 2^{10} [1024] \end{cases}$$


ROM : electric circuits operating on diode

↳ Cannot Modify Structure, Read-Only

Advantage : Fast

Disadvantage : Costly

[OS Course]

BIOS → ROM is used

↳ Booting

OS cannot be stored in RAM, since
On Switch on → Deleted, (Non-volatile)
Hence RAM is not used.

Why Read-only, Why not Write?

↳ It should not change. Start-up should not be modified.
i.e. Content changes → Corrupted → Cannot be reused.

Types of ROM : Erasable ROMs, Normal ROMs
(EPROM)

ROM does not use flip-flops.

↳ Given small voltage, whatever is stored will execute.

RAM : Sequential Circuit

ROM : Combinational Circuit (No memory unit)

_____ X _____