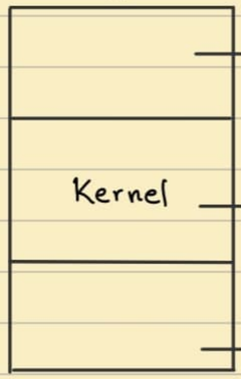
21/8

Application / Software



Higher level Languages (C++, Java)

⟺ 8085 / 8086 (Platform)

Assembly Language

⟺ System Software (Compiler / Assembler)

Machine Language

Hardware

Memory

⇓

Registers

Architechture : Van Neuman

TOC : Alan Turing

↳ Decidability, Computability

$(142)_6 \Rightarrow 1 \times 6^2 + 4 \times 6 + 2 \times 1$

$= \underline{\underline{62}}$

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

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

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

$(1000)_4 \Rightarrow (64)_{10}$

4	43	
4	10	3
4	2	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}

$4 \equiv 0100$

$2 \equiv 0010$

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

$(31)_4 = (1101)_2$

2^2

$(42)_8 = (100010)_2$

$\hookrightarrow 2^3$

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

$\hookrightarrow 2^3$

$(101)|(101)$

$\hookrightarrow 5 \quad \hookrightarrow 5$

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

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

$0010|1101$

$\hookrightarrow 2 \quad \hookrightarrow D$

Q) $(101101110)_2$

$1|01|10|11|10 \equiv (11232)_4$

$101|101|110 \equiv (556)_8$

$0001|0110|1110 \equiv (16E)_{16}$

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

$\underbrace{0101}\,\underbrace{1000}$

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

$\hookrightarrow (\ )_2 \nearrow$

$01|011|000$

$(1\ \ 3\ \ 0)_8$

Test:

$(58)_{16}$

$16 \times 5 + 8$

$= (88)_{10}$

$|01|01|10|00$

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

4	88	
4	22	0
4	5	2
4	1	1
	0	1

(1120)

$\underbrace{|11|}_{5}\underbrace{20}_{8}$

number + 9s complement (number) = 99

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

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

$60 + 1 = (61)$

↳ 9s complement + 1 ⇒ 10s complement

$$\begin{array}{r} 42 \\ 39 \\ \hline 81 \end{array} \qquad \begin{array}{r} \overset{3}{\cancel{4}}2 \\ -\ 39 \\ \hline 03 \end{array} \qquad \equiv \qquad \begin{array}{r} 42 \\ +\ 60 \\ \hline 1|02 \\ +1 \\ \hline 3 \end{array}$$

→ Binary Addition / Subtraction:

$$\begin{array}{r} 1011 \\ 0101 \\ \hline 10000 \end{array}$$

$11 + 5 = 16$

$\Rightarrow$ 1011

$(0101)^c = 1010$ → 1^s complement

$$\begin{array}{r} 1010 \\ 1 \\ \hline 1011 \end{array}$$

↳ 2^s complement

$11 - 5 = 6$

$(1)^c = 0$

$(0)^c = 1$

$$\begin{array}{r} 1011 \\ 1011 \\ \hline 10110 \end{array}$$

↳ Ignore 1

$0110 \rightarrow \underline{\underline{6}}$ $\qquad 11 - 5 = 6$

(14) 1110

(9) 1001 ⟶ 0110

$$\begin{array}{r} 0110 \\ 1 \\ \hline 0111 \end{array}$$

$$\begin{array}{r} 1110 \\ 0111 \\ \hline 1|0101 \end{array}$$

(5)

Carry = 1 ⇒ Ignore

Carry = 0 ⇒ Negative

$$\begin{array}{r} 9 \\ -14 \\ \hline 11 \end{array} \qquad \begin{array}{r} 1001 \\ 0010 \\ \hline 1011 \end{array}$$

$1011 \longrightarrow 0100$ → 1^s complement

$$\begin{array}{r} 0100 \\ 1 \\ \hline 0101 \end{array}$$

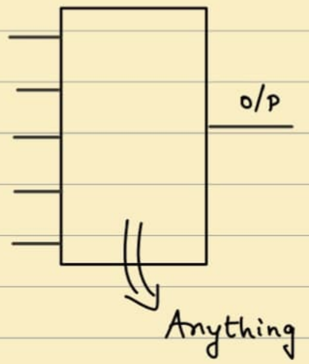
↳ 2^s complement

(−5)

max → $2^4 = 16$

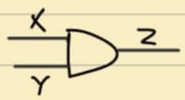
$11 - 16 = -5$

→ Gates:



3 Fundamental Gates :
- AND
- OR
- NOT

AND

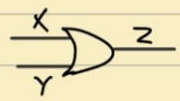


	X	Y	Z
0	0	0	0
1	0	1	0
2	1	0	0
3	1	1	1

$Z = X \cdot Y$

$\therefore \boxed{Z = XY}$

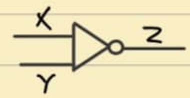
OR



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

$\boxed{Z = X + Y}$

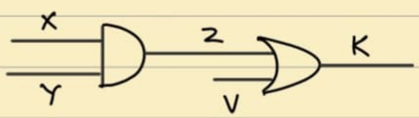
NOT



	X	Z
0	0	1
1	1	0

$\boxed{Z = \overline{X}}$

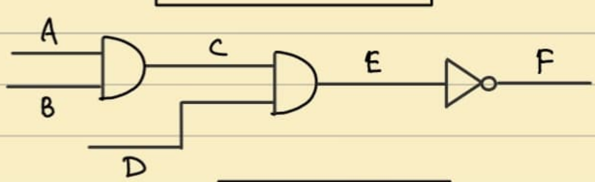
- Combination :



$K = Z + V$

$= X \cdot Y + V$

$\boxed{\therefore\ K = XY + V}$

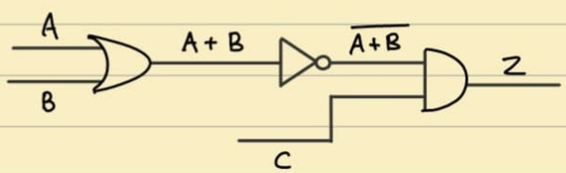


$F = \overline{E}$

$= \overline{C \cdot D}$

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

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

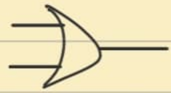


$\boxed{\therefore\ Z = (\overline{A+B}) \cdot C}$

30/8

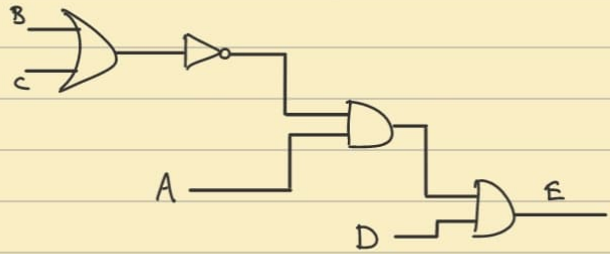
NOT $\rightarrow$ more prescedence , NOT > AND > OR


: AND


: OR

o : NOT

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



NOR : OR + NOT $\longrightarrow$ 

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

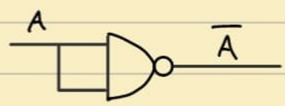
NAND : AND + NOT $\longrightarrow$ 

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

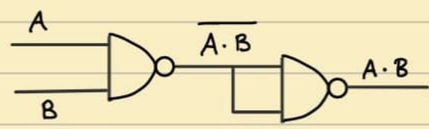
NOR & NAND : Universal Gates

AND, OR & NOT : Basic Gates

NOT using NAND :



AND using NANDs :



$X \cdot X = X$

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

• : AND

+ : OR

— : NOT

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

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

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

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

George bool – Boolean Algebra

Boolean (Bool) : True or False

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

2/9

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

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

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

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

$= \bar{B} + AB$

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

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

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

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

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

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

$A + \bar{A}B$

$A + (1-A)B$

$A + B - AB$

$A(1-B)+B$

$A\bar{B}+B$

Q) $ABC + \bar{A}BC + AB\bar{C}$

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

$= BC + AB\bar{C}$

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

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:

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

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

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

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

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

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

$= A + \bar{A}B$

$= A+B$

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

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

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

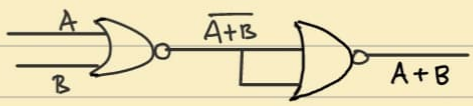
$= A(B+\bar{C}+\bar{B}\bar{C})$

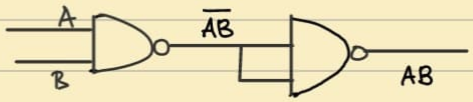
$= A(B+\bar{C}(1+B))$

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

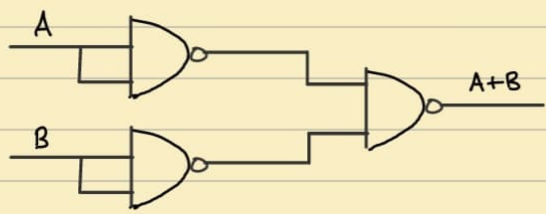
$\Rightarrow$ NOT

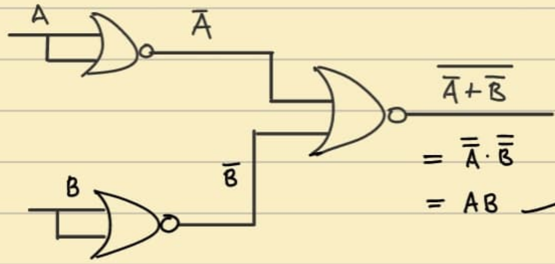
$\overline{\overline{AB}} + \overline{\bar{A}+\bar{B}} = \bar{\bar{A}} + \bar{\bar{B}} = A+B$





OR using NANDs :





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

$= AB$ → AND using NOR

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

$$AB\bar{C} + A\bar{B}C + \bar{A}BC + ABC + ABC + ABC$$
$$= AB(C+\bar{C}) + AC(B+\bar{B}) + BC(A+\bar{A})$$
$$= \underbrace{AB + BC + CA}_{\text{Term - Standard Form (Not Unique)}}$$

→ 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	$\Rightarrow \bar{A}B$
1	0	0	0	0	1	$\Rightarrow A\bar{B}$
1	1	1	0	0	1	$\Rightarrow AB$

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

Ex. $\underset{110}{AB\bar{C}} + \underset{101}{A\bar{B}C} + \underset{011}{\bar{A}BC} + \underset{111}{ABC}$

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	$(AB\bar{C})$
1	1	1	1	(ABC)

$\underset{110}{AB\bar{C}} + \underset{101}{A\bar{B}C} + \underset{011}{\bar{A}BC} + \underset{111}{ABC}$

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

$\quad 11 \quad\; 01 \quad\; 10$

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

$\Rightarrow A+B$

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

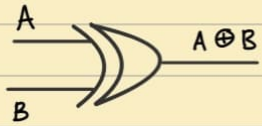
A	B	F	
0	0	0	
0	1	1	$\leftarrow \bar{A}B$
1	0	1	$\leftarrow A\bar{B}$
1	1	0	

A	B	C	F
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

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

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

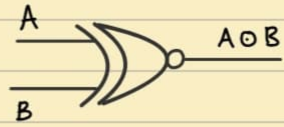
$$\bar{A}B + \bar{A}C + A$$
$$\Rightarrow \bar{A}B + \bar{A}C + A(1+B)$$
$$\Rightarrow \bar{A}B + \bar{A}C + A + AB$$
$$\Rightarrow B + \bar{A}C + A(1+C)$$
$$\Rightarrow B + \bar{A}C + A + AC$$
$$\Rightarrow A + B + C$$



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

$[01, 10 \rightarrow 1]$

<u>XOR</u>



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

$[00, 11 \rightarrow 1]$

<u>XNOR</u>

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

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

$$
\begin{aligned}
& A \oplus X \\
&= A\bar{X} + \bar{A}X \\
&= A(\overline{B \oplus A}) + \bar{A}(B \oplus A) \\
&= A(\overline{B\bar{A} + \bar{B}A}) + \bar{A}(B\bar{A} + \bar{B}A) \\
&= A(\overline{B\bar{A}} \cdot \overline{\bar{B}A}) + \bar{A}\cdot\bar{A}\cdot B + \bar{A}\cdot A \cdot B \\
&= A[(\bar{B}+A)\cdot(B+\bar{A})] + B + 0 \\
&= A(\bar{B}+A)\cdot(B+\bar{A}) \\
&= A(AB + A\bar{A} + B\bar{B} + \bar{A}\bar{B}) \\
&= A(AB + \bar{A}\bar{B}) \\
&= A\cdot A\cdot B + A\cdot\bar{A}\,\bar{B} \\
&= B
\end{aligned}
$$

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

  2 GB   2 GB   2 GB

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

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

  $= A$

  $\therefore$ We get back A

$(12.5)_{10} = (1100.1)_2$ $[0.5 \times 2 = 1]$

IEEE 754 : Standard

↪ Floating point number — 32 bit

64 bit

32 bit

sign (1)   exponent (8)   mantissa (23)

For 32 bit, Bias (B) = 127

exponent without bias

$1100.1 = 1.1001 \times 2^3$

$(\because 1024.321 = 1.024321 \times 10^3)$

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

0100000101 0010

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

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

$\longrightarrow (0.001)_2$

$\left[\because 1/8 = 0.125\right]$

$0.125 \times 2 = 0.25$
$0.25 \times 2 = 0.5$
$0.5 \times 2 = 1.0$
$\Rightarrow 0.001$

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

$B + e = 127 + (-3)$
$= 124$
$= 128 - 4$

Bias
↳ Negative numbers can be stored

0.1111100 0000000

$-0.125 \equiv$ 1.1111000000000

Ex. 0 01111110 101000 $\Rightarrow 1.101 \times 2^e$
s e m

1.101×2^{-1}
$= (0.1101)_2$
$= (0.8125)_{10}$

$126 - e = 127$
$\Rightarrow e = -1$

0.0625
250
5

Ex. 0 10000011 110000

$(-1)^0 = 1$

$131 - 127 = \underline{4}$

$= 1.11 \times 2^4$

$= 11100 = (28.0)_{10}$

Q) -24.625

$24.625 \equiv (11000.101)$

1.1000101×2^4

$B + e = 127 + 4 = 131$

$\hookrightarrow 128 + 3$

1 10000011 1000101

Q) 1 10000111 111000

$\hookrightarrow 128 + 4 + 2 + 1 \Rightarrow 127 + 8 \Rightarrow \underline{\underline{e = 8}}$

$1.111 \times 2^8 = 111100000$

$= (-480)_{10}$

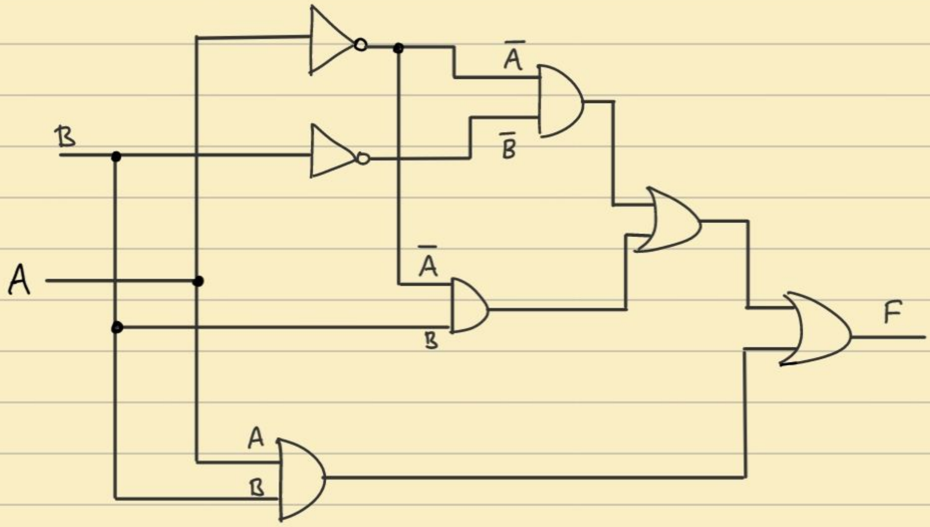
$$\left[\begin{array}{r} 32 \\ 64 \\ 128 \\ 256 \\ \hline 480 \end{array}\right]$$

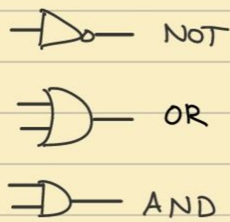
P.T.O

DSD Lab — I

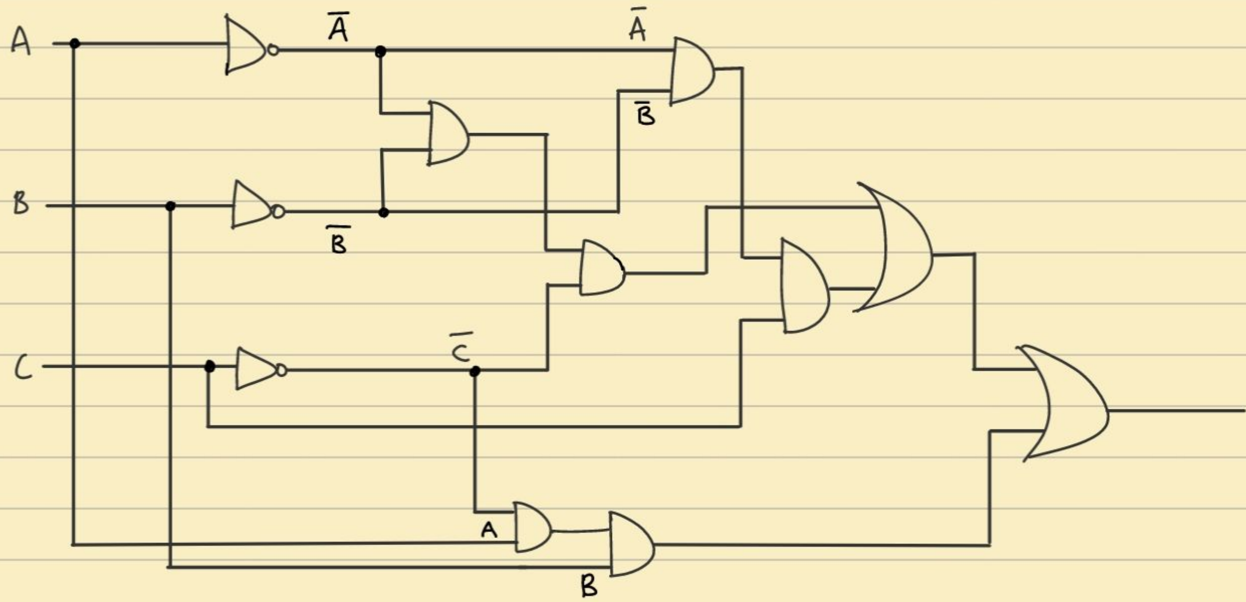
(1) $F(A, B) = \bar{A}\cdot\bar{B} + \bar{A}\cdot B + A\cdot B$

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





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



6/9

3 ways to represent negative no.s
(1) Signed Number
(2) One's complement
(3) Two's complement

(1) Signed no.

8 bit ⇒ 1 7

sign mag

+1 0 0000001

−1 1 0000001

(2) 1^s complement

4 bit → 0 0111

1 0000

Shortcome:

0 0000 − 0

1 1111 − 0

(3) 2^s complement

0 0111
1 1000
  1
---
1 1001

1111
   1
---
1 0000

+ → 0
− → 1

Q) 28.875 → Float
Q) −14.8 → Float
Q) 110000110000100100 → Decimal
Q) 0.0625 → Float
Q) 10111110110010000

$\underbrace{\text{CFA12301}}_{}$ → 32 bits
C ↳ 4 bit

$(CFA12301)_{16}$
= 1100 ... 0001

(1) 28.875 | 28 = 16 + 8 + 4
↳ 11100.111
⇒ 1.1100111×2^4

127 + 4 = 131 = 128 + 2 + 1
10000011

0.875 × 2 = 1.75
1.75 × 2 = 1.5
0.5 × 2 = 1.0
⇒ 0.111

s = +ve → 0

0 10000011 1100111
s e m

(3) $\underbrace{1}_{s}\underbrace{10000110}_{e}\underbrace{000100100}_{m}$

s : 1 digit
e : 8 digits
m : 23 digits

$10000110 \equiv 2+4+128$

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

s $\longrightarrow$ -ve

1.000100100×2^7
$= (10001001.00)_2$
$= 1+8+128$
$= -(137)_{10}$

(5) $\underbrace{1}_{s}\underbrace{01111101}_{e}\underbrace{100100000}_{m}$

$01111101 \longrightarrow 1+4+8+16+32+64$
$= 125$
$125-127 = \underline{\underline{-2}}$

$1.100100000 \times 2^{-2}$
$= 1.001 \times 2^{-2}$
$= (0.01001)_2$
$= 2^{-2}+2^{-5}$
$= \frac{1}{4}+\frac{1}{32}$

$= \underline{\underline{(0.28125)_{10}}}$

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

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

$14 = 8+4+2$

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

$0.8 \times 2 = 1.6$
$0.6 \times 2 = 1.2$
$0.2 \times 2 = 0.4$
$0.4 \times 2 = 0.8$
$\Rightarrow 0.11001100....$

$(1110.110011)_2$

$= 1.110110011 \times 10^3$

$127+3 = 130 = 128+2$
10000010

$\therefore \underbrace{1}_{s\ (-ve)}\ \underbrace{10000010}_{e}\ \underbrace{11001100110011001100 11}_{m}$

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) $\bar{A}B\bar{C} + \bar{A}\bar{B}\bar{C} + ABC + \overline{ABC}$ $\longrightarrow$ Solve by converting into canonical form ($\overline{ABC} = \bar{A}+\bar{B}+\bar{C}$)

Q) $ABC + \bar{A}\bar{B}\bar{C} + AB\bar{C}$

Q) $24 - 2$ (2s complement)

Q) $A \odot B \odot C$ (Truth table)

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

9/9

A1) $A\bar{B}(C+\bar{C}) + \bar{A}\bar{B}(C+\bar{C}) + B$

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

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

$= \bar{B} + B$

$= 1$

A2) $\bar{A}B\bar{C} + \bar{A}\bar{B}\bar{C} + ABC + \overline{ABC}$

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

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

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

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

$= \overline{ABC} + ABC$

$= 1$

A3) $ABC + \bar{A}\bar{B}\bar{C} + AB\bar{C}$

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

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

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

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

$= ABC + \bar{C}(0 + AB)$

$= ABC + \bar{C}\cdot AB$

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

$= AB$

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

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

$= 0$

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

$A \odot B \odot C$

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

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

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

$\Rightarrow ABC + \overline{A}\overline{B}C + (\overline{AB} \cdot (A+B)) \cdot \overline{C}$ As $\overline{A+B} = \overline{A} \cdot \overline{B}$

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

→ **K-Maps** : (Karnaugh - Maps)

A	B	F(A·B)
0	0	0
0	1	0
1	0	1
1	1	1

A \ B	0	1
0	0_0	0_1
1	1_2	1_3

$\Rightarrow A$

0 : $\overline{A}$

1 : A

A \ BC	00	01	11	10
0	0_0	1_1	0_3	0_2
1	0_4	0_5	1_7	1_6

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

SOP - Min Terms

POS - Max Terms

Canonical → SOP, POS

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

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

Ex.

F

A \ BC	00	01	11	10
0	1_0	1_1	0_3	0_2
1	0_4	0_5	1_7	1_6

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

F

A \ BC	00	01	11	10
0	1_0	1_1	1_3	0_2
1	0_4	1_5	1_7	1_6

$F = C + AB + AB$

[Preference : More Power]

F

A \ BC	00	01	11	10
0	1_0	1_1	0_3	0_2
1	0_4	1_5	1_7	1_6

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

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

Ex.

F

A \ BC	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

A \ BC	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

A \ BC	00	01	11	10
0	1_{0}	0_{1}	1_{3}	1_{2}
1	1_{4}	1_{5}	1_{7}	1_{6}

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

Ex.

F

A \ BC	00	01	11	10
0	0_{0}	0_{1}	0_{3}	0_{2}
1	1_{4}	0_{5}	0_{7}	1_{6}

$F = \overline{AC}$

Ex.

F

A \ BC	00	01	11	10
0	1_{0}	0_{1}	1_{3}	1_{2}
1	1_{4}	0_{5}	0_{7}	1_{6}

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

Ex.

F

A \ BC	00	01	11	10
0	1_{0}	1_{1}	0_{3}	0_{2}
1	1_{4}	0_{5}	1_{7}	1_{6}

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

F

A \ BC	00	01	11	10
0	1_0	1_1	0_3	1_2
1	0_4	1_5	1_7	1_6

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

(OR)

F

A \ BC	00	01	11	10
0	1_0	1_1	0_3	1_2
1	0_4	1_5	1_7	1_6

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

F

A \ BC	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

A \ BC	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

φ) $\Sigma(m_0, m_2, m_3, m_4, m_6, m_7)$

F

A \ BC	00	01	11	10
0	1_0	0_1	1_3	1_2
1	1_4	0_5	1_7	1_6

$F = B + \bar{C}$

φ) $\Sigma(m_0, m_2, m_3, m_4, m_5, m_7)$

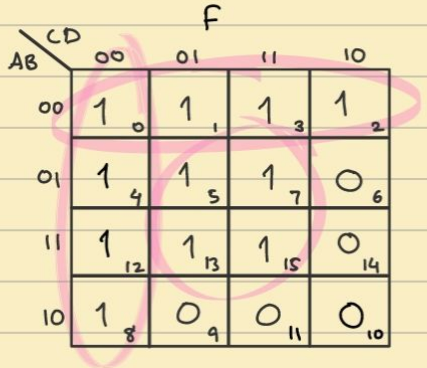
A \ BC	00	01	11	10
0	1_0	0_1	1_3	1_2
1	1_4	1_5	1_7	0_6

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

φ) $\Sigma(m_0, m_2, m_3, m_4, m_5, m_6, m_7)$

A \ BC	00	01	11	10
0	1_0	0_1	1_3	1_2
1	1_4	1_5	1_7	1_6

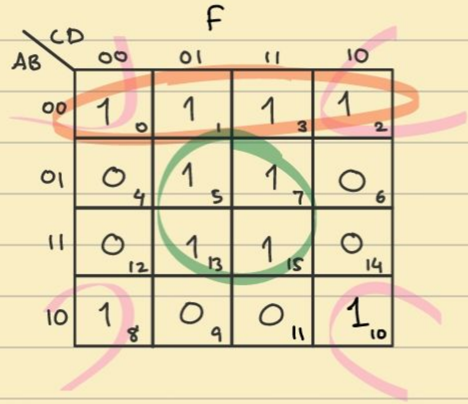
$F = A + B + \bar{C}$



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

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

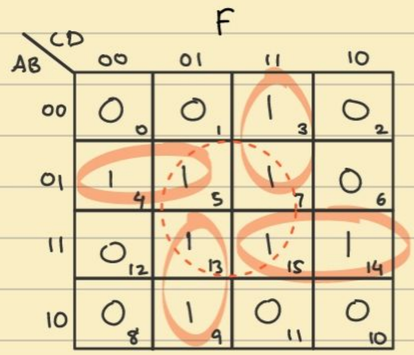
Q)



F(A, B, C, D)

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

Q)



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

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

Group ≡ Prime Implicant (PI)

Essential Group ≡ Essential Prime Implicant

↳ Without which reduction can't happen

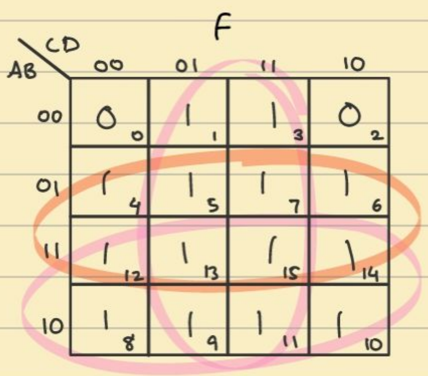
Non-essential / Extra Group ≡ Redundant Prime Implication

↳ Repitition

↳ Unnecessary

Supportive ≠ Redundant
S. PI R. PI

Q)



$F = B + A + D$

8 combinations ⟶ 3 variables reduced
4 combinations ⟶ 2 variables reduced
2 combinations ⟶ 1 variable reduced

H.W

$$\Sigma(m_0, m_1, m_3, m_4, m_8, m_{11}, m_{12}, m_{13}, m_{15})$$
$$= \overline{C}\,\overline{D} + \overline{A}\,\overline{B}D + ABD + \overline{B}CD$$

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

11/9

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	${}_{0}$	${}_{1}$
1	${}_{2}$	${}_{3}$

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

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

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

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

Q)

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}	1_{11}	${}_{10}$	${}_{14}$	${}_{15}$	${}_{13}$	${}_{12}$
11	${}_{24}$	1_{25}	1_{27}	${}_{26}$	${}_{30}$	${}_{31}$	${}_{29}$	${}_{28}$
10	${}_{16}$	1_{17}	1_{19}	${}_{18}$	${}_{22}$	1_{23}	1_{21}	${}_{20}$

Mirror elements can be combined.

Q)

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}	1_{11}	$_{10}$	$_{14}$	$_{15}$	$_{13}$	$_{12}$
11	$_{24}$	1_{25}	1_{27}	$_{26}$	$_{30}$	$_{31}$	$_{29}$	$_{28}$
10	1_{16}	1_{17}	1_{19}	$_{18}$	$_{22}$	1_{23}	1_{21}	1_{20}

Q)

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}	1_{11}	$_{10}$	$_{14}$	$_{15}$	$_{13}$	$_{12}$
11	$_{24}$	1_{25}	1_{27}	$_{26}$	$_{30}$	$_{31}$	$_{29}$	$_{28}$
10	1_{16}	0_{17}	1_{19}	$_{18}$	$_{22}$	1_{23}	0_{21}	1_{20}

Q)

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}	$_{10}$	$_{14}$	$_{15}$	$_{13}$	$_{12}$
11	$_{24}$	0_{25}	1_{27}	0_{26}	0_{30}	1_{31}	$_{29}$	$_{28}$
10	0_{16}	0_{17}	1_{19}	0_{18}	0_{22}	1_{23}	1_{21}	0_{20}

$A\bar{C}DE + ACDE$

$\Rightarrow ADE$

Q)

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}	$_{13}$	$_{12}$
11	$_{24}$	0_{25}	1_{27}	1_{26}	0_{30}	1_{31}	$_{29}$	$_{28}$
10	0_{16}	0_{17}	1_{19}	0_{18}	0_{22}	0_{23}	1_{21}	0_{20}

Cannot combine / Not mirrorable

Q)

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}	1_{11}	1_{10}	0_{14}	1_{15}	1_{13}	$_{12}$
11	$_{24}$	1_{25}	1_{27}	1_{26}	0_{30}	1_{31}	1_{29}	$_{28}$
10	0_{16}	0_{17}	1_{19}	0_{18}	0_{22}	0_{23}	1_{21}	0_{20}

$B\bar{C}E + BCE = BE$

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}	${}_{13}$	${}_{12}$
11	${}_{24}$	0_{25}	1_{27}	1_{26}	0_{30}	1_{31}	${}_{29}$	${}_{28}$
10	1_{16}	0_{17}	0_{19}	1_{18}	0_{22}	0_{23}	1_{21}	0_{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}	${}_{13}$	${}_{12}$
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}

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

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}

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

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}

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

Gray code : (One-bit flip)	Binary :
000	000
001	001
011	010
010	011
110	100
111	101
101	110
100	111

11/9

Lab - 2

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

② $\bar{A}\bar{B}C + ABC + AC + AB$

③ $A\bar{B} + \bar{A}B$ using NOR alone

①

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

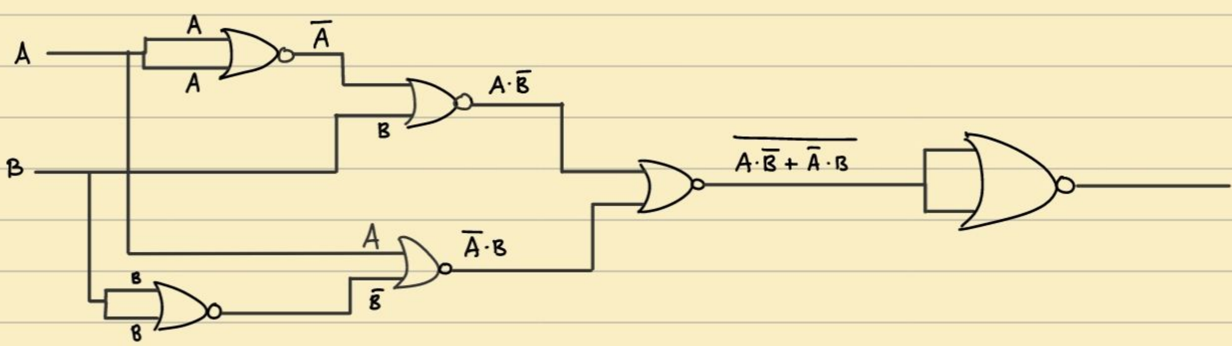
③ $\bar{A}B + A\bar{B}$

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

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

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

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



Q) $A \oplus B \oplus C$

Sol :

$$(A\bar{B}+B\bar{A}) \oplus C = (A\bar{B}+\bar{A}B)\cdot\bar{C} + \left(\overline{A\bar{B}+\bar{A}B}\right)\cdot C$$

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

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

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

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

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

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

111 001 010 100

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

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

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

P.T.O

Four 4 bits :

Binary code $B_3\ B_2\ B_1\ B_0$		Gray code : $G_3\ G_2\ G_1\ G_0$
0 0 0 0	$\longrightarrow$	0 0 0 0
0 0 0 1	$\longrightarrow$	0 0 0 1
0 0 1 0	$\longrightarrow$	0 0 1 1
0 0 1 1	$\longrightarrow$	0 0 1 0
0 1 0 0	$\longrightarrow$	0 1 1 0
0 1 0 1	$\longrightarrow$	0 1 1 1
0 1 1 0	$\longrightarrow$	0 1 0 1
0 1 1 1	$\longrightarrow$	0 1 0 0
1 0 0 0	$\longrightarrow$	1 1 0 0
1 0 0 1	$\longrightarrow$	1 1 0 1
1 0 1 0	$\longrightarrow$	1 1 1 1
1 0 1 1	$\longrightarrow$	1 1 1 0
1 1 0 0	$\longrightarrow$	1 0 1 0
1 1 0 1	$\longrightarrow$	1 0 1 1
1 1 1 0	$\longrightarrow$	1 0 0 1
1 1 1 1	$\longrightarrow$	1 0 0 0

$$\boxed{G_3 = B_3}$$

A \ BC	00	01	11	10
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_3B_2 \ B_1B_0	00	01	11	10
00	0 (0)	0 (1)	1 (3)	1 (2)
01	1 (4)	1 (5)	0 (7)	0 (6)
11	0 (12)	0 (13)	1 (15)	1 (14)
10	1 (8)	1 (7)	0 (11)	0 (10)

$\bar{B_3}\bar{B_2}B_1 + \bar{B_3}B_2\bar{B_1}$

$+ B_3B_2B_1 + B_3\bar{B_2}\bar{B}$

$\Downarrow$

$B_1 \oplus B_2 \oplus B_3$

G_2

B_3B_2 \ B_1B_0	00	01	11	10
00	0 (0)	0 (1)	0 (3)	0 (2)
01	1 (4)	1 (5)	1 (7)	1 (6)
11	0 (12)	0 (13)	0 (15)	0 (14)
10	1 (8)	1 (7)	1 (11)	1 (10)

$B_3\bar{B_2} + B_2\bar{B_3} = \boxed{B_2 \oplus B_3 = G_2}$

G_0

B_3B_2 \ B_1B_0	00	01	11	10
00	0 (0)	1 (1)	0 (3)	1 (2)
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)

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

$$\boxed{G_0 = B_1 \oplus B_0}$$

Q) $\sum(m_0, m_1, m_2, m_4, m_6, m_8, m_{12}, m_{13})$

Q) $\sum(m_0, m_4, m_9, m_{15}, m_{16}, m_{20}, m_{25}, m_{31})$

Q) $\sum(m_0, m_2, m_4, m_6, m_{16}, m_{18}, m_{20}, m_{22})$

Q) $\sum(m_8, m_{10}, m_{12}, m_{14}, m_{16}, m_{30})$

Q) $\sum(m_0, m_1, m_3, m_4, m_5, m_7, m_{16}, m_{17}, m_{19}, m_{20}, m_{21}, m_{23})$

Q) $\sum(m_8, m_9, m_{10}, m_{11}, m_{13}, m_{14}, m_{15}, m_{27}, m_{29})$

①

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

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

Q2)

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

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

Q3)

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

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

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	1_{10}	1_{14}	15	13	1_{12}
11	24	25	27	26	1_{30}	31	29	28
10	1_{16}	17	19	18	22	23	21	20

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

Q5)

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

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

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_{14}	1_{15}	1_{13}	12
11	24	25	1_{27}	26	30	31	1_{29}	28
10	16	17	19	18	22	23	21	20

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

Q7)

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

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

P.T.O

Binary code → Gray 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

$$G_3 = B_3$$
$$G_2 = B_2 \oplus B_3$$
$$G_1 = B_1 \oplus B_2$$
$$G_0 = B_0 \oplus B_1$$

18/9

G_2

$B_3B_2 \backslash B_1B_0$	00	01	11	10
00	0_{0}	0_{1}	0_{3}	0_{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}

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

G_1

$B_3B_2 \backslash B_1B_0$	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}

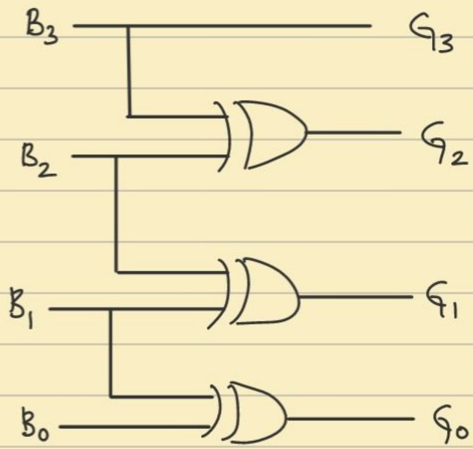
$$G_1 = B_1\overline{B_2} + B_2\overline{B_1}$$
$$B_1 \oplus B_2$$

G_0

$B_3B_2 \setminus B_1B_0$	00	01	11	10
00	0_{0}	1_{1}	0_{3}	1_{2}
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}

$$G_0 = B_1\overline{B_0} + B_0\overline{B_1}$$
$$B_1 \oplus B_0$$

Circuit:



B_2

$G_3G_2 \setminus G_1G_0$	00	01	11	10
00	0_{0}	0_{1}	0_{3}	0_{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}

$$\boxed{B_2 = G_3 \oplus G_2}$$

B_1

$G_3G_2 \setminus G_1G_0$	00	01	11	10
00	0_{0}	0_{1}	1_{3}	1_{2}
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}

$$B_1 = \overline{G_2}\,\overline{G_3}\,G_1 + \overline{G_3}\,G_2\,\overline{G_1} + G_3G_2G_1 + G_3\,\overline{G_2}\,\overline{G_1}$$

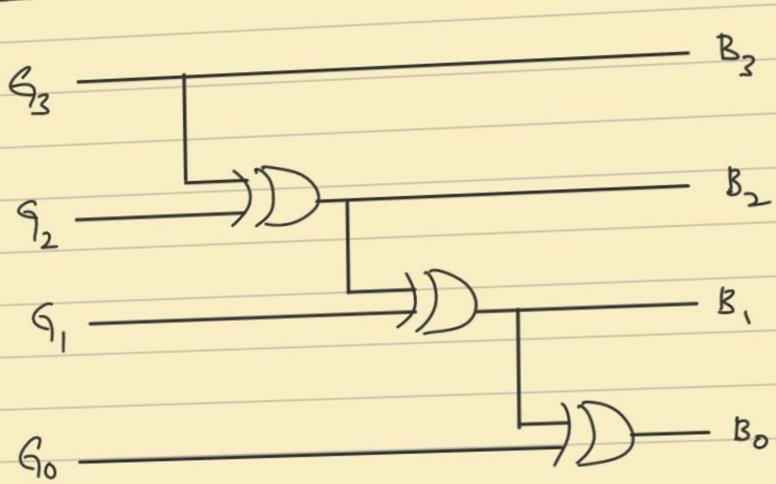
$$\boxed{B_1 = G_3 \oplus G_2 \oplus G_1}$$

B_0

G_3G_2 \ G_1G_0	00	01	11	10
00	0	1	0	1
01	1	0	1	0
11	0	1	0	1
10	1	0	1	0

$$\boxed{B_0 = G_1 \oplus G_2 \oplus G_3 \oplus G_0}$$

Circuit:



$$B_3 = G_3$$
$$B_2 = G_3 \oplus G_2$$
$$B_1 = G_3 \oplus G_2 \oplus G_1$$
$$B_0 = G_3 \oplus G_2 \oplus G_1 \oplus G_0$$

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

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

$BC + AB$

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

A \ BC	00	01	11	10
0	X_0	1_1	0_3	1_2
1	1_4	1_5	0_7	0_6

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

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

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

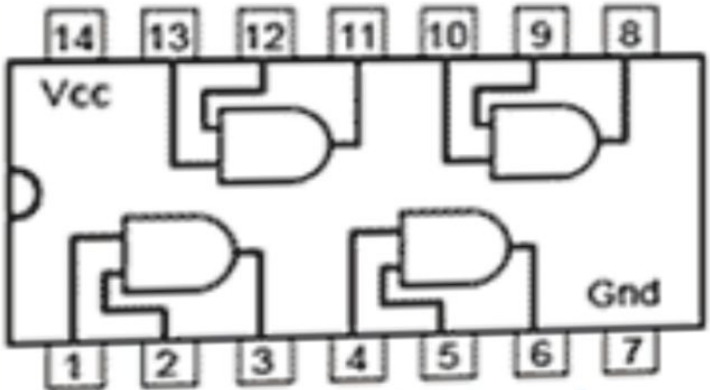
AB \ CD	00	01	11	10
00	1_0	X_1	1_3	1_2
01	1_4	X_5	0_7	1_6
11	1_{12}	1_{13}	X_{15}	0_{14}
10	1_8	1_9	0_{11}	0_{10}

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

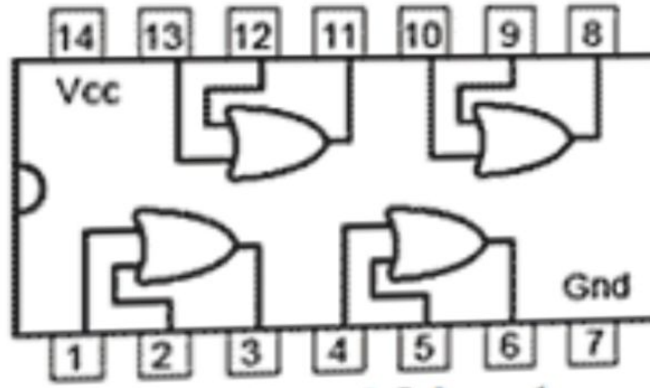
$X_{15} = 0$
$\because$ Extra term & combination 1 is already a part of 8

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

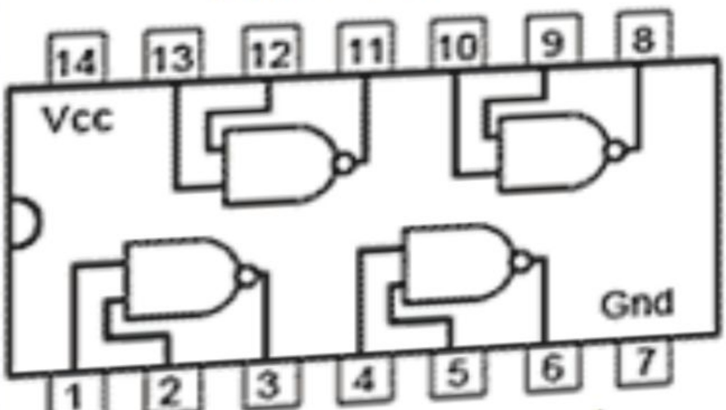
P.T.O



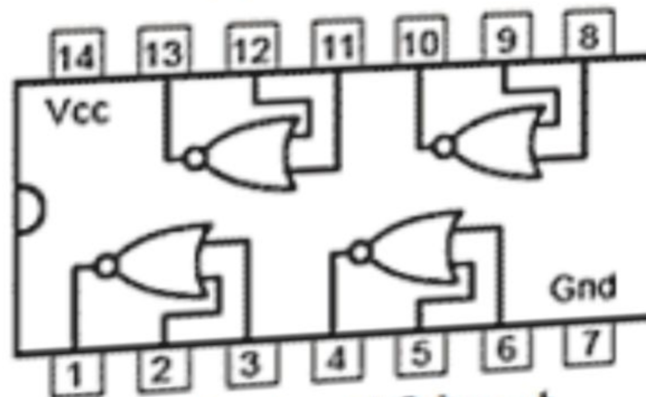
7408 Quad 2 input AND Gates



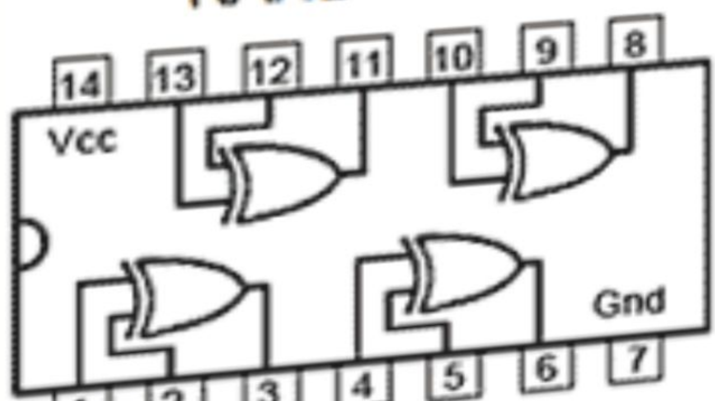
7432 Quad 2 input OR Gates



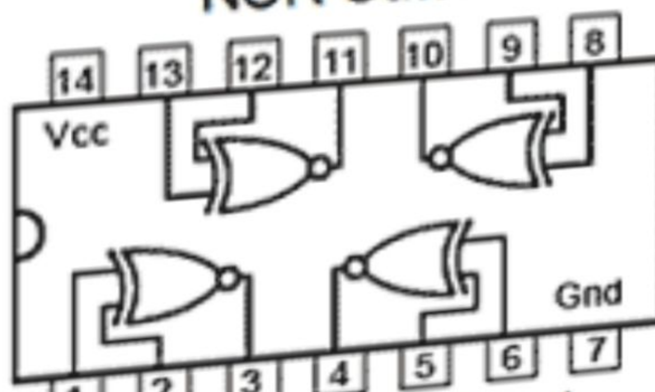
7400 Quad 2 input NAND Gates



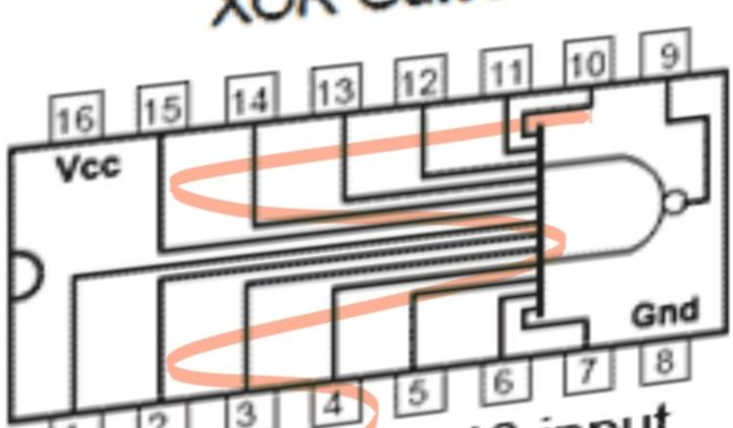
7402 Quad 2 input NOR Gates



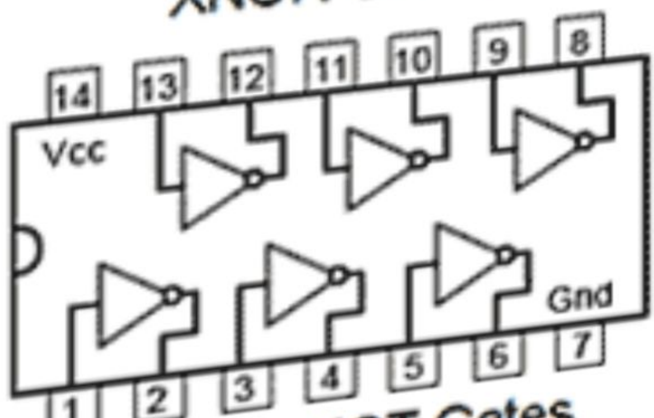
7486 Quad 2 input XOR Gates



747266 Quad 2 input XNOR Gates



74133 Single 13 input



7404 Hex NOT Gates (Inverters)

18/9

Lab-3

② $\Sigma(m_0, m_2, m_4, m_6) + \Sigma(d_1, d_5)$

F

A \ BC	00	01	11	10
0	1_{0}	X_{1}	0_{3}	1_{2}
1	1_{4}	X_{5}	0_{7}	1_{6}

$\begin{pmatrix} X_1 = 0 \\ X_5 = 0 \end{pmatrix}$

$F = \bar{C}$

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

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

$X_9 = 0$

$X_{13} = 0$

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

P.T.O

19/9

BCD → Binary Coded Decimal

4	8	12
0100	1000	0001 0010
		1 2

Binary code $B_3\ B_2\ B_1\ B_0$		BCD $D_4\ D_3\ D_2\ D_1\ D_0$	
0 0 0 0	→	0 0 0 0 0	
0 0 0 1	→	0 0 0 0 1	
0 0 1 0	→	0 0 0 1 0	
0 0 1 1	→	0 0 0 1 1	
0 1 0 0	→	0 0 1 0 0	
0 1 0 1	→	0 0 1 0 1	
0 1 1 0	→	0 0 1 1 0	
0 1 1 1	→	0 0 1 1 1	
1 0 0 0	→	0 1 0 0 0	
1 0 0 1	→	0 1 0 0 1	
1 0 1 0	→	1 0 0 0 0	(10)
1 0 1 1	→	1 0 0 0 1	(11)
1 1 0 0	→	1 0 0 1 0	(12)
1 1 0 1	→	1 0 0 1 1	(13)
1 1 1 0	→	1 0 1 0 0	(14)
1 1 1 1	→	1 0 1 0 1	(15)

15 – 1111 (Binary)

15 – $\overset{D_4}{1}$ | $010\overset{D_0}{1}$ (BCD)

1 5

$$\boxed{B_0 = D_0}$$

D_1

$B_3B_2 \backslash B_1B_0$	00	01	11	10
00	0_0	0_1	1_3	1_2
01	0_4	0_5	1_7	1_6
11	1_{12}	1_{13}	0_{15}	0_{14}
10	0_8	0_9	0_{11}	0_{10}

$$\boxed{D_1 = \overline{B}_1 B_2 B_3 + B_1 \overline{B}_3}$$

D_2

B_3B_2 \ B_1B_0	00	01	11	10
00	0 $_{0}$	0 $_{1}$	0 $_{3}$	0 $_{2}$
01	1 $_{4}$	1 $_{5}$	1 $_{7}$	1 $_{6}$
11	0 $_{12}$	0 $_{13}$	1 $_{15}$	1 $_{14}$
10	0 $_{8}$	0 $_{9}$	0 $_{11}$	0 $_{10}$

$$\boxed{D_2 = B_2B_1 + \overline{B_3}B_2}$$

D_3

B_3B_2 \ B_1B_0	00	01	11	10
00	0 $_{0}$	0 $_{1}$	0 $_{3}$	0 $_{2}$
01	0 $_{4}$	0 $_{5}$	0 $_{7}$	0 $_{6}$
11	0 $_{12}$	0 $_{13}$	0 $_{15}$	0 $_{14}$
10	1 $_{8}$	1 $_{9}$	0 $_{11}$	0 $_{10}$

$$\boxed{D_3 = B_3\overline{B_2}\,\overline{B_1}}$$

D_4

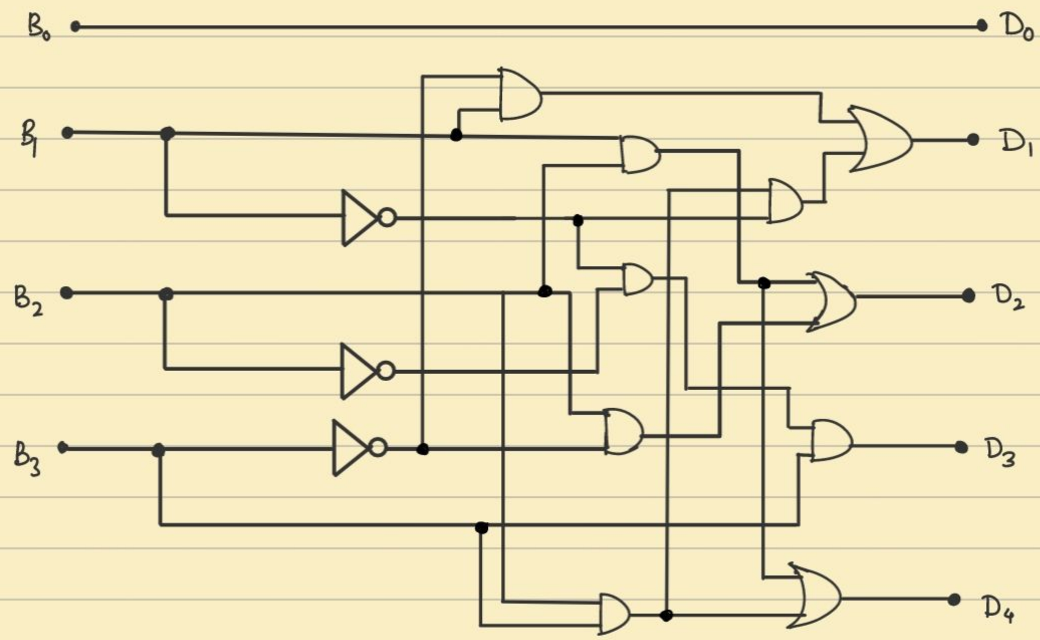
B_3B_2 \ B_1B_0	00	01	11	10
00	0 $_{0}$	0 $_{1}$	0 $_{3}$	0 $_{2}$
01	0 $_{4}$	0 $_{5}$	0 $_{7}$	0 $_{6}$
11	1 $_{12}$	1 $_{13}$	1 $_{15}$	1 $_{14}$
10	0 $_{8}$	0 $_{9}$	1 $_{11}$	1 $_{10}$

$$\boxed{D_4 = B_3B_2 + B_2B_1}$$

$$D_4 = B_3B_2 + B_2B_1$$
$$D_3 = B_3\overline{B_2}\,\overline{B_1}$$
$$D_2 = \overline{B_3}B_2 + B_2B_1$$
$$D_1 = B_3B_2\overline{B_1} + B_1\overline{B_3}$$
$$D_0 = B_0$$

$$D_4 = B_3 B_2 + B_2 B_1$$
$$D_3 = B_3 \overline{B_2}\,\overline{B_1}$$
$$D_2 = \overline{B_3} B_2 + B_2 B_1$$
$$D_1 = B_3 B_2 \overline{B_1} + B_1 \overline{B_3}$$
$$D_0 = B_0$$

Circuit Diagram :



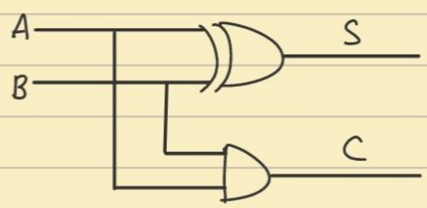
$\longrightarrow$ Addition :

Sum
Carry

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

$S \longrightarrow$ XOR $(A\overline{B} + \overline{A}B = A \oplus B)$
$C \longrightarrow$ AND $(A \cdot B)$

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
$$\begin{array}{r} {\scriptstyle 1} \\ 11 \\ +111 \\ \hline 110 \end{array}$$

Half Adder can handle only 2 inputs

∴ We use Full Adder

$S : X \oplus Y \oplus Z$

$C : XY + YZ + ZX$

↳ Without using Half Adders

S

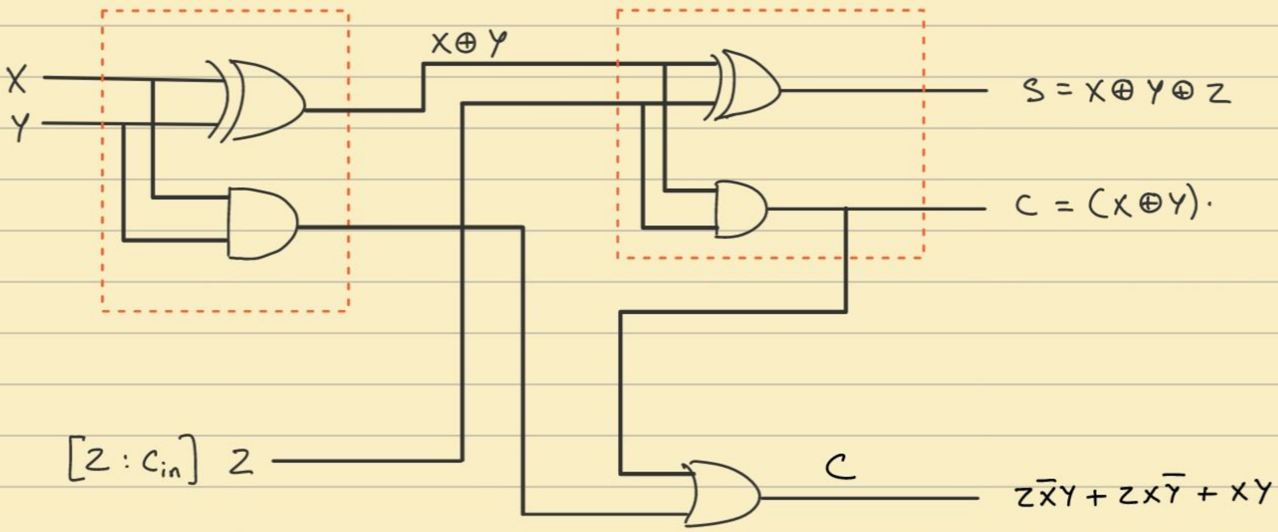
A \ BC	00	01	11	10
0	0_0	1_1	0_3	1_2
1	1_4	0_5	1_7	0_6

C

A \ BC	00	01	11	10
0	0_0	0_1	1_3	0_2
1	0_4	1_5	1_7	1_6

20/9 :

Full Adders using 2 Half Adders



$C_{out} = \underbrace{(X \oplus Y) \cdot Z}_{C_1} + \underbrace{XY}_{C}$

Subtractor :

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

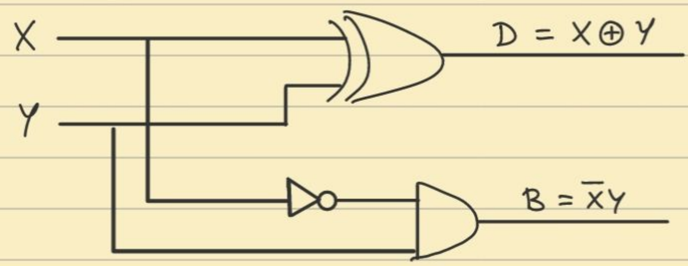
$$\begin{array}{r} 0 \\ -\,0 \\ \hline 00 \end{array}$$
B ↙ ↘ D

$$\begin{array}{r} 1 \\ -\,0 \\ \hline 01 \end{array}$$

$$\begin{array}{r} {\scriptstyle 1}0 \\ -\,1 \\ \hline 11 \end{array}$$

$D = X \oplus Y$

$B = \bar{X} Y$



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

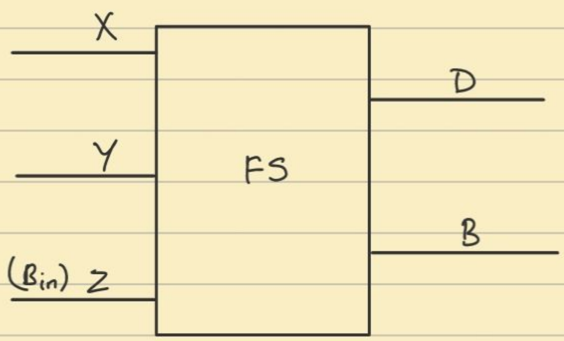
$D = X \oplus Y \oplus Z$

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

B

X \ YZ	00	01	11	10
0	0_0	1_1	1_3	1_2
1	0_4	0_5	1_7	0_6

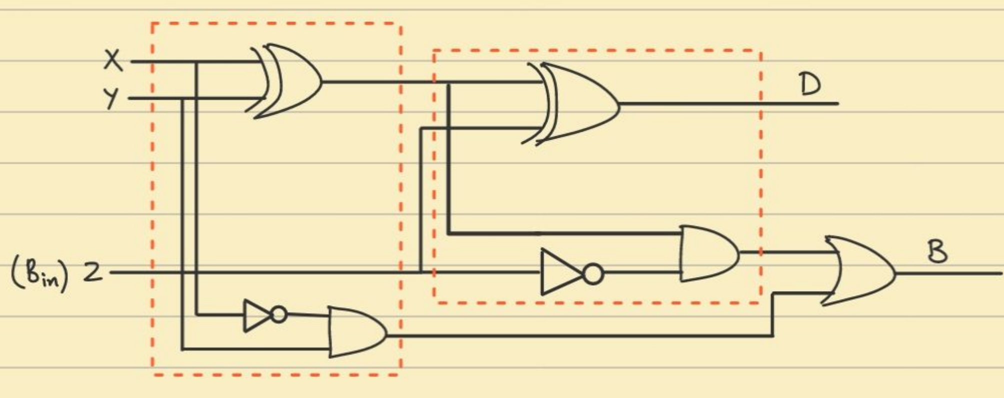
$\bar{X}Z + \bar{X}Y + YZ$



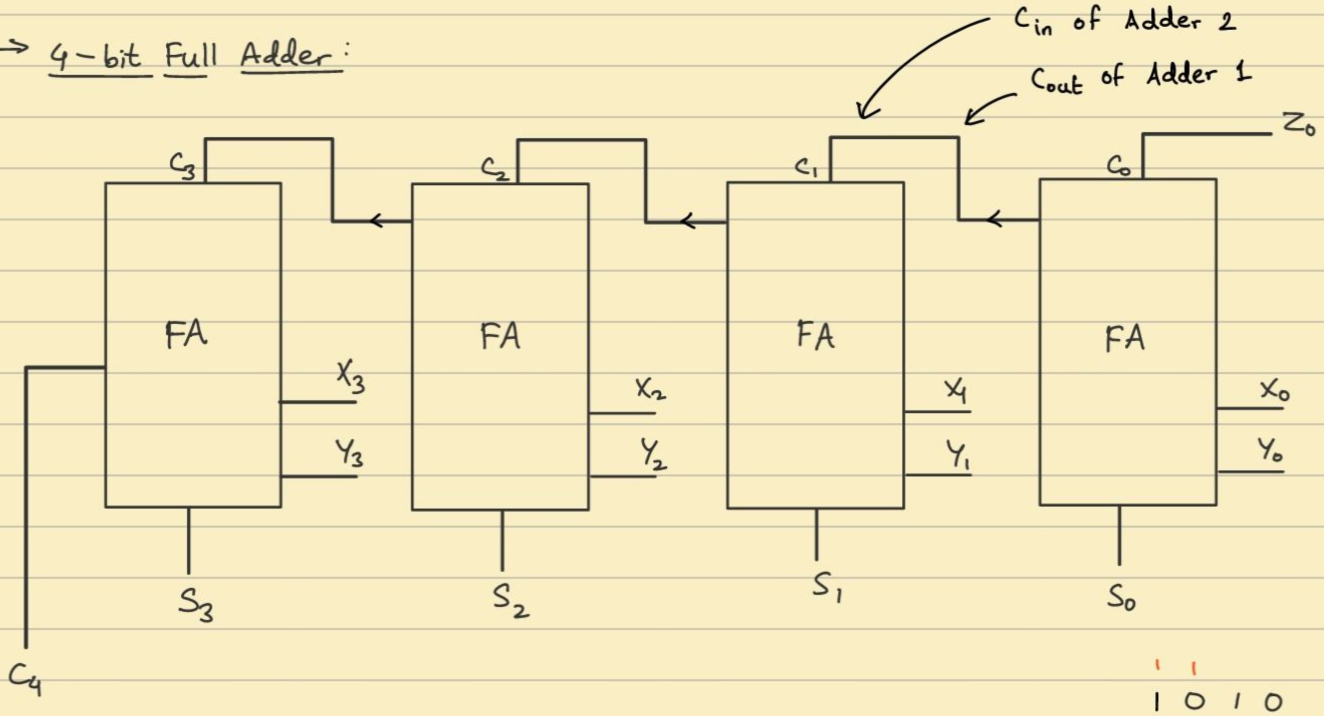
$B_0 = X\bar{Y}$

$B_{out}\ (B_1) = (\overline{X \oplus Y})Z + \bar{X}Y$

Full Subtractor



→ 4-bit Full Adder:



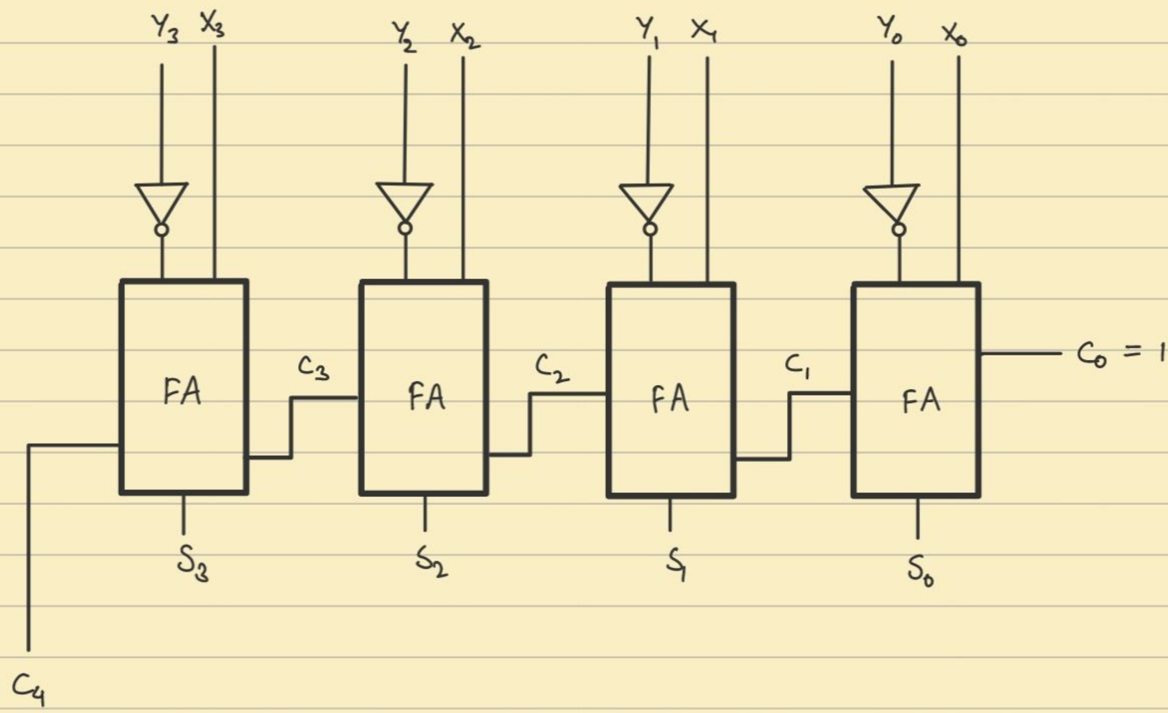
$$
\begin{array}{r}
{\scriptstyle 1\ 1\ \ \ \ } \\
1\,0\,1\,0 \\
0\,1\,1\,1 \\
\hline
1\,0\,0\,0\,1
\end{array}
$$

→ 4-bit Subtractor using Adder:

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

14 + 2s complement of 5 = 9

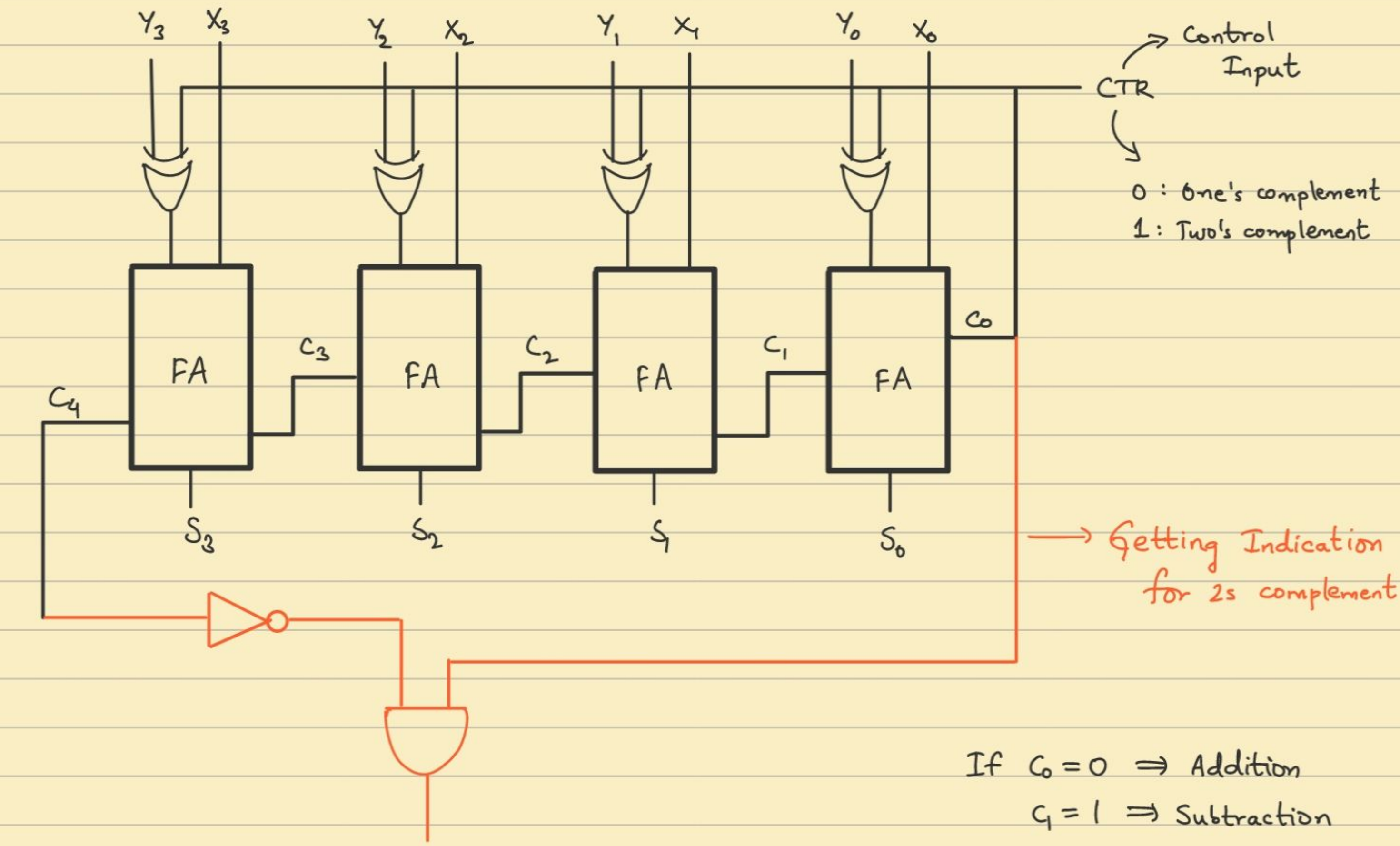
$$
\begin{array}{rl}
1\,1\,1\,0 & \rightarrow 14 \\
1\,0\,1\,1 & \rightarrow 2s\ c(5) \\
\hline
(1)\ \underbrace{1\,0\,0\,1}_{\rightarrow 9} &
\end{array}
$$



$$\text{2s complement} = \underbrace{1}_{\rightarrow C_0} + \underbrace{\text{1s complement}}_{\rightarrow \text{Not Gates}}$$

$A \oplus 0 = A$
$A \oplus 1 = \bar{A}$

→ 4-bit Adder-Subtractor :



If $C_0 = 0 \Rightarrow$ Addition
$C_0 = 1 \Rightarrow$ Subtraction

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 + $\underbrace{\text{2s complement of 12}}_{0100}$:
10 → 1010

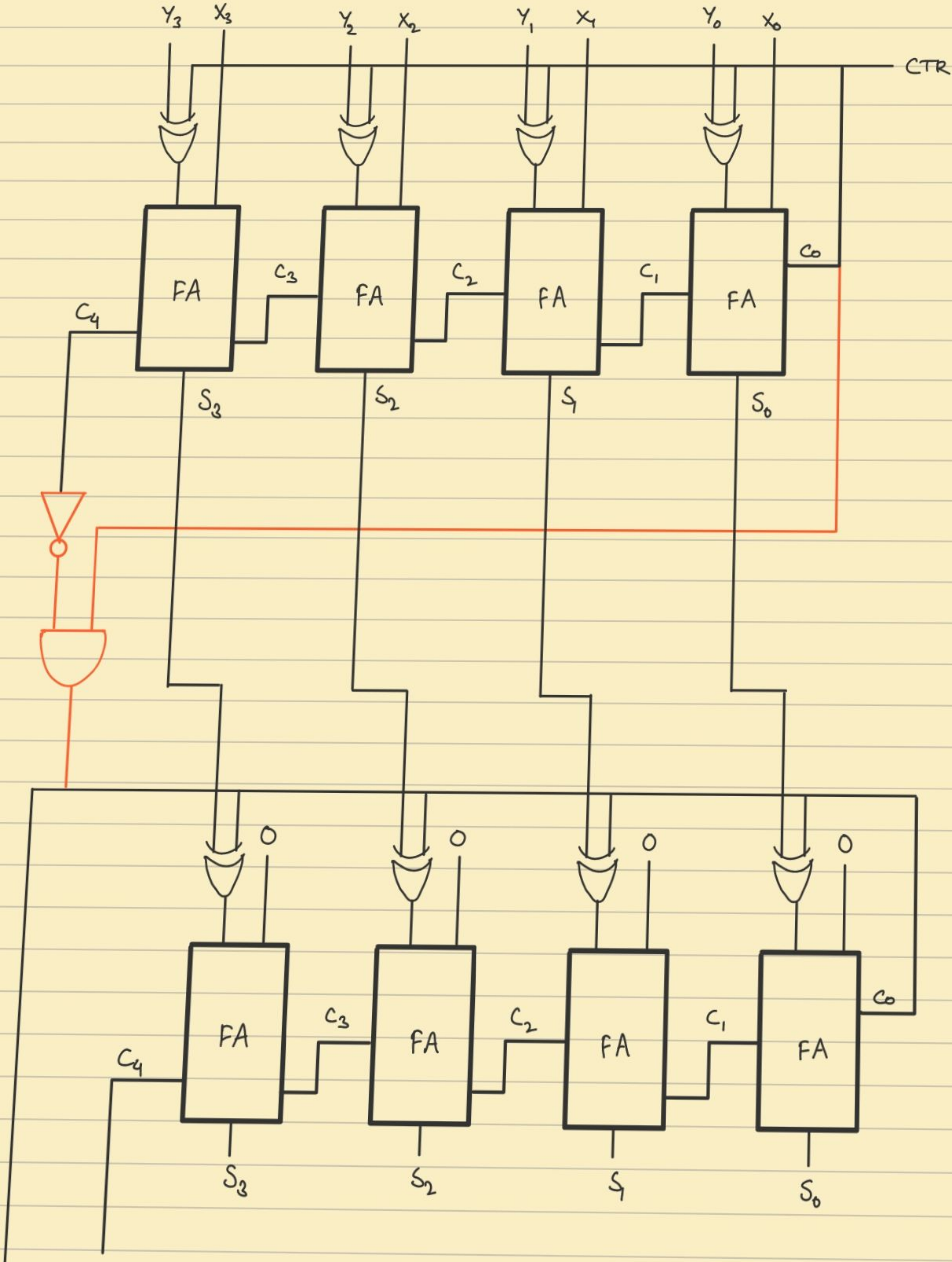
$$\begin{array}{r} 1010 \\ 0100 \\ \hline 1110 \end{array}$$

No carry → Indicating output is in Two-s complement form

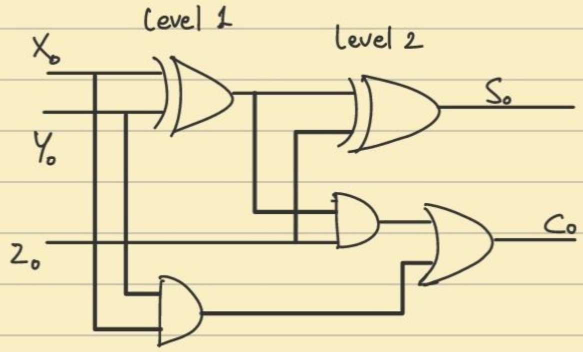
∴ When CTR = 1 & $C_4 = 0 \Rightarrow A < B$

But Answer displayed — 11110

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



The XOR gates calculate 1s complement



Carry Propagator

$$P_0 = X_0 \oplus Y_0 \qquad S = P \oplus C$$
$$G_0 = X_0 Y_0$$

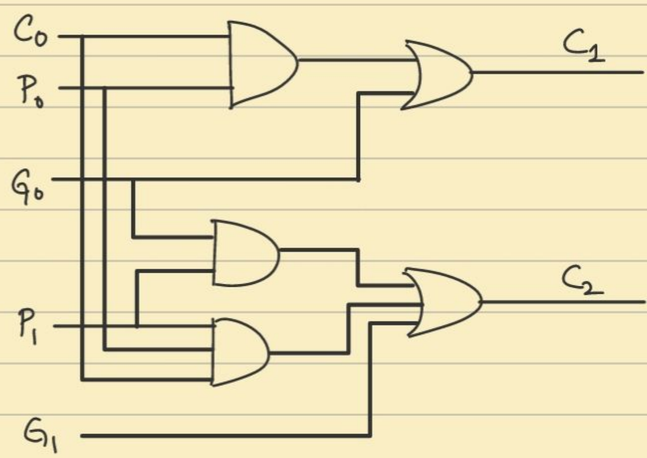
Carry Generator

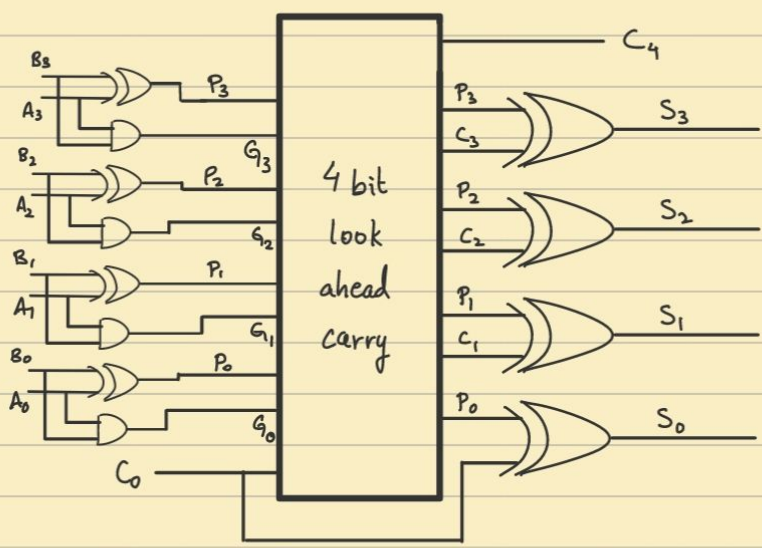
4 bit look ahead Carry Adder

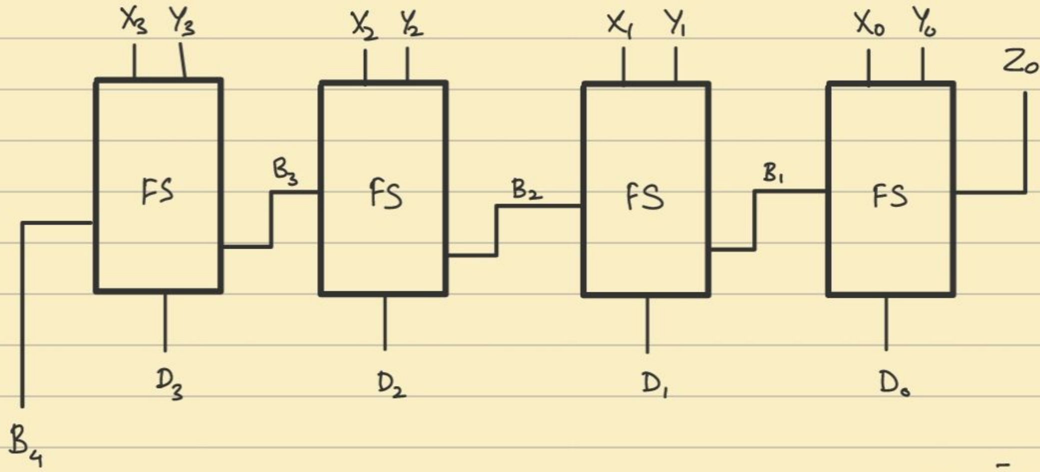
$$\left.\begin{aligned} C_1 &= G_0 + P_0 C_0 \\ C_2 &= G_1 + P_1 C_1 \\ C_3 &= G_2 + P_2 C_2 \end{aligned}\right\} \Rightarrow \boxed{C_n = G_{n-1} + P_{n-1} \cdot C_{n-1}}$$

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

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



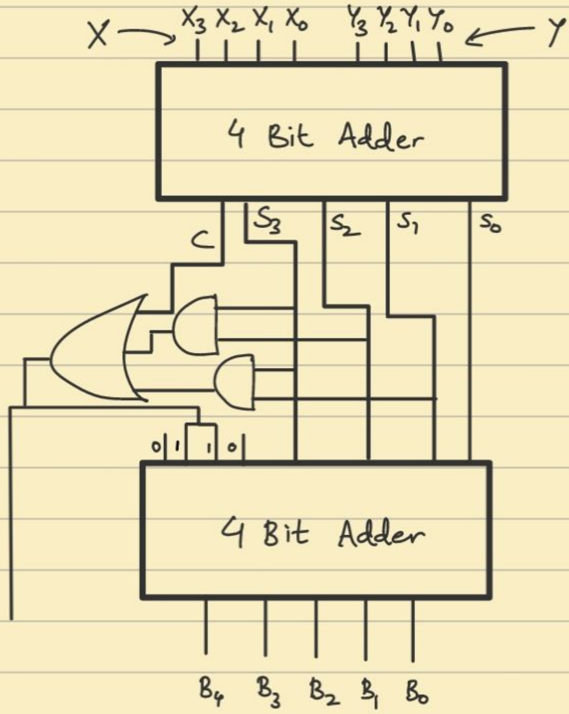




$\bar{X}Y + Z(\overline{X \oplus Y})$

$B = \bar{X}Y + \bar{X}Z + ZY$

$D = X \oplus Y \oplus Z$



	Binary Sum	BCD Sum	
	$CS_3S_2S_1S_0$	$B_4B_3B_2B_1B_0$	
(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

CS_3 \ $S_2S_1S_0$	000	001	011	010	110	111	101	100
00	0_0	0_1	0_3	0_2	0_6	0_7	0_5	0_4
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}

$\boxed{B_4 = C + S_1S_3 + S_2S_3}$

25/9

→ **Magnitude Comparator:**

$A < B : \overline{A_0} B_0$
$A > B : A_0 \overline{B_0}$
$A = B : A_0 B_0 + \overline{A_0}\,\overline{B_0} = A_0 \odot B_0$

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 Comperator → 2 variables – 4 rows
2-bit Comperator → 4 variables – 16 rows
n-bit Comperator → 2n variables – 2^{2n} rows

n : no. of bit of each number

Values : 2^{2n}

1	4
2	16
3	64

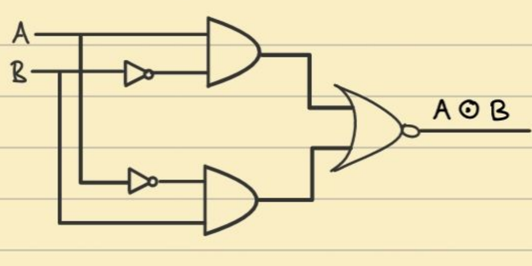
B_1	B_0	A_1	A_0	$A=B$	$A<B$	$A>B$
0	0	0	0	1	0	0
0	0	0	1	0	0	1
0	0	1	0	0	1	1
0	0	1	1	0	0	1
0	1	0	0	0	1	0
0	1	0	1	1	0	0
0	1	1	0	0	0	1
0	1	1	1	0	0	1
1	0	0	0	0	1	0
1	0	0	1	0	1	0
1	0	1	0	1	0	0
1	0	1	1	0	0	1
1	1	0	0	0	1	0
1	1	0	1	0	1	0
1	1	1	0	0	1	0
1	1	1	1	1	0	0

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

$A\overline{B}$ $\quad$ $\overline{A}B$

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

⇓



$\longrightarrow$ $0111 - A$

$\underline{0001 - B}$

$A_3\ A_2\ A_1\ A_0$

$0\ 1\ 1\ 1$

$B_3\ B_2\ B_1\ B_0$

$0\ 0\ 0\ 1$

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

Then, go for A_2 & B_2.

Then, go for A_1 & B_1

Lastly, go for A_0 & B_0

1^{st} step $\Longrightarrow A = B : A \odot B$

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

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

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

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

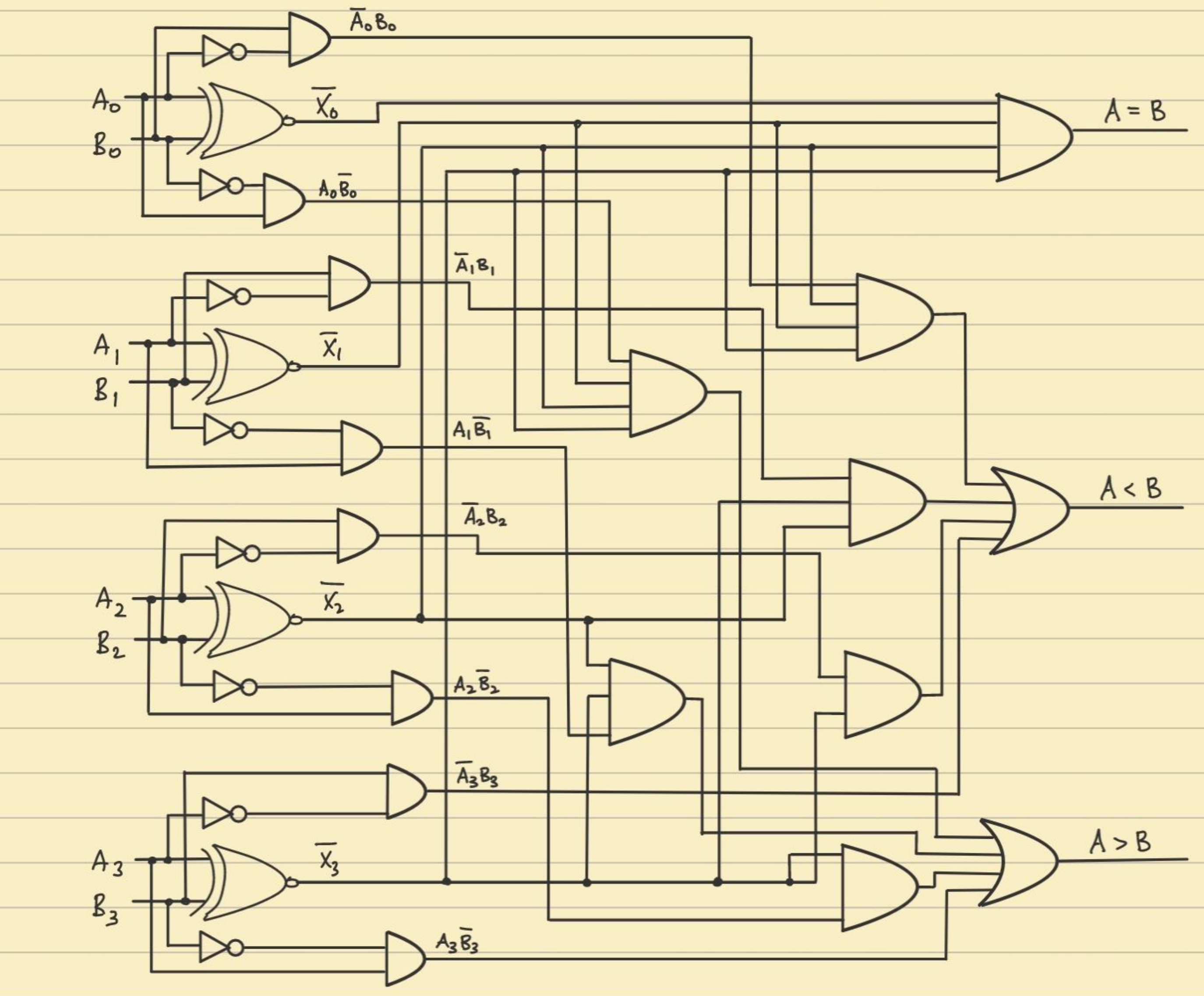
2^{nd} step $\Longrightarrow A > B : A\overline{B}$

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

3^{rd} Step $\Longrightarrow A < B : \overline{A}B$

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

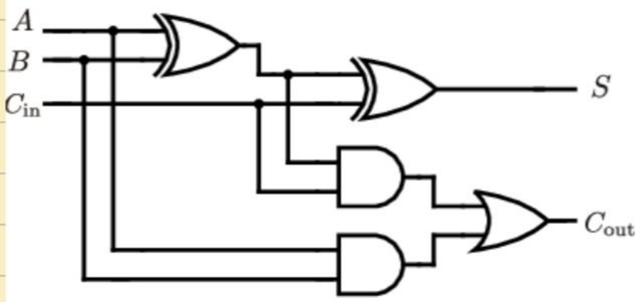
Circuit :



25/9

Lab-4

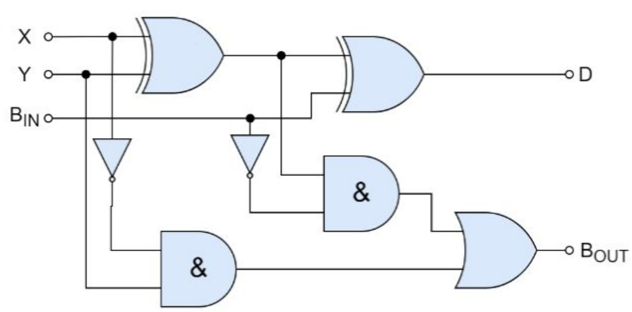
Q1) Full-Adder



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



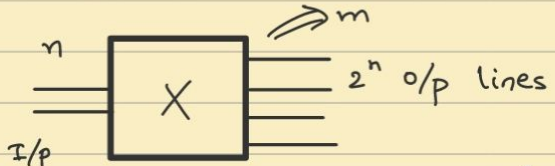
Truth Table

B_{IN}	Y	X	DIFF	B_{OUT}
0	0	0	0	0
0	0	1	1	0
0	1	0	1	1
0	1	1	0	0
1	0	0	1	1
1	0	1	0	0
1	1	0	0	1
1	1	1	1	1

P.T.O

27/9

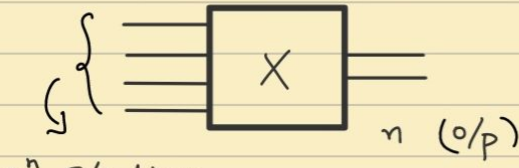
Decoder:



$(m < 2^n)$

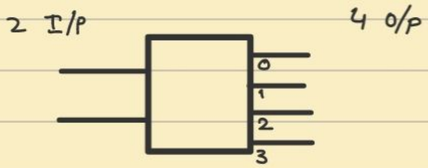
1	$2^1 = 2$
2	$2^2 = 4$
3	$2^3 = 8$

Encoder:



n:m decoder → Ex. Naming – 4:16 decoder

(1) 2:4 Decoder:

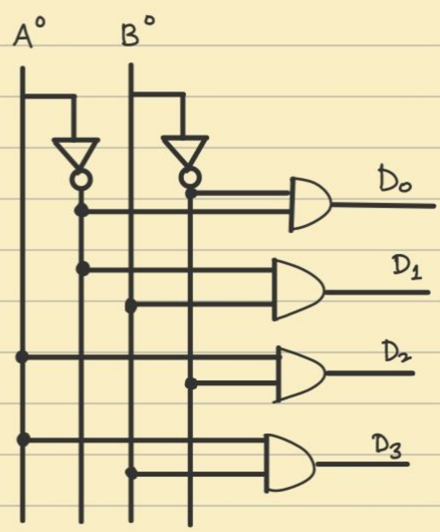


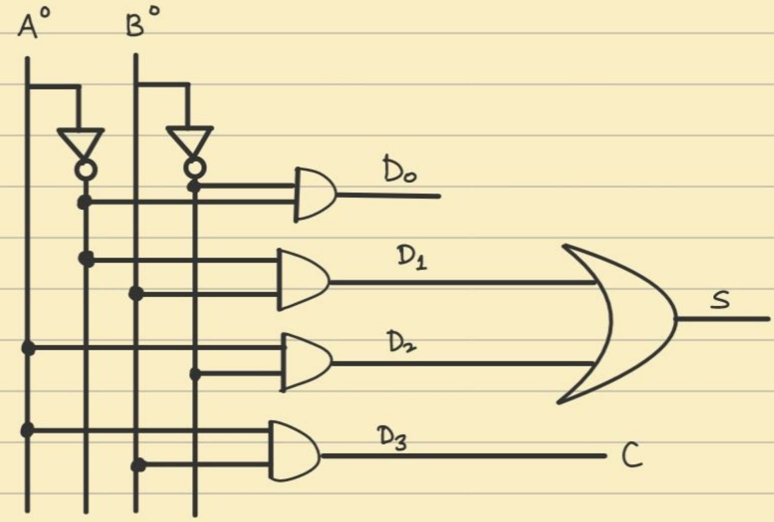
high = 1
low = 0

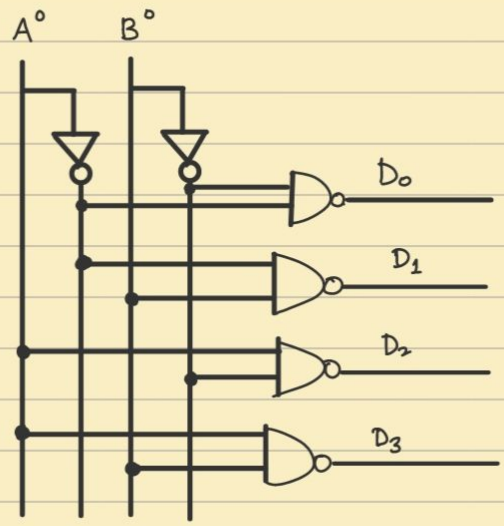
A	B	D_0	D_1	D_2	D_3
0	0	1	0	0	0
0	1	0	1	0	0
1	0	0	0	1	0
1	1	0	0	0	1

Only one output can be high for an input combination

- 2:4 Decoder:







NAND is used instead of AND $\because$ NAND $\rightarrow$ Cheaper

Let us look at 3:8 Decoder

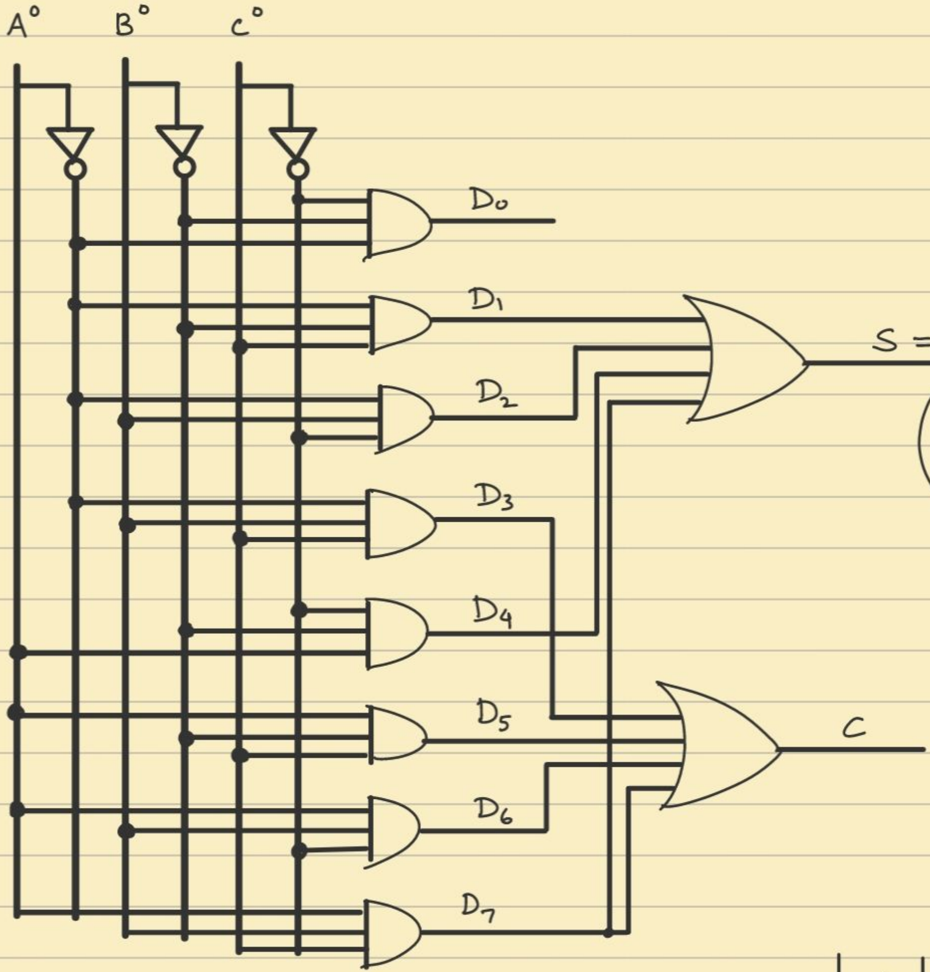
↳ Advantage : Full Adder

Decoder : Each output is attached to a bulb.

P.T.O

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

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



$S = X \oplus Y \oplus Z$

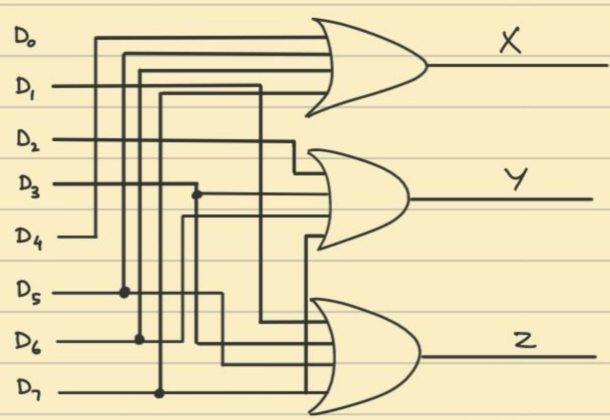
$\left(\bar{A}\bar{B}C + \bar{A}B\bar{C} + A\bar{B}\bar{C} + ABC\right)$

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

S

X \ YZ	00	01	11	10
0	0 (0)	1 (1)	0 (3)	1 (2)
1	1 (4)	0 (5)	1 (7)	0 (6)

→ 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

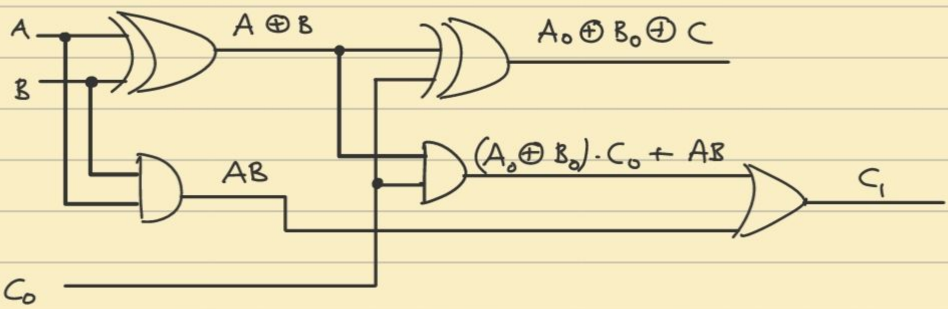
Q) $\sum(m_1, m_2, m_4, m_6, m_8, m_{17}, m_{19}, m_{25}, m_{28}, m_{30}, m_9, m_{11}, m_{27}, m_{15}, m_{29})$

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

Q)

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

→ <u>4-bit look ahead carry adder</u>:



$$C_1 = P_0 C_0 + G_0$$
$$C_2 = P_1 C_1 + G_1$$
$$C_3 = P_2 C_2 + G_2$$

$$C_2 = P_1 (P_0 C_0 + G_0) + G_1$$
$$= P_0 P_1 C_0 + P_1 G_0 + G_1$$

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

→ Floating Point number — IEEE 754

↪ 32 bit

S	C	M
0/1	8 bits	23 bits
1 bit		

Ex. 0 10000010 11010000...

$(41680000)_{16}$

$$\underbrace{0}_{s}|\overbrace{10000010}|\underbrace{1101000\ldots}_{m}$$

$S = 0 \Rightarrow$ Positive

$e - D = 130 - 127 = 3$

power $= +3$

$1.\text{mantissa}$

$= 1.11010000\ldots \times 2^3$

$= (1110.1)_2$

$= (14.5)_{10}$

g) $\underbrace{1}_{-ve}\underbrace{01111110}_{e}\underbrace{1101000}_{m}\ldots\ldots \leftarrow (BF680000)_{16}$

$126 - 127 = -1$

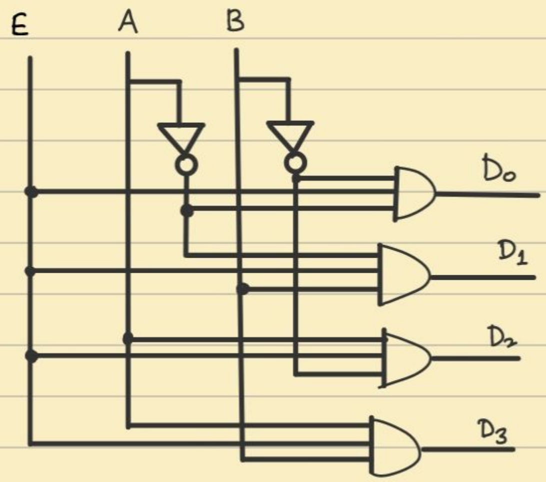
$\Rightarrow -1.1101 \times 2^{-1}$

$\Rightarrow -0.11101$

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

P.T.O

8/10



No output will be realised if Enable bit is off.

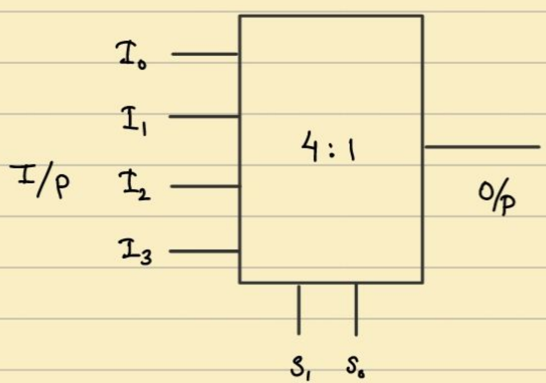
E	A	B	D_0	D_1	D_2	D_3
0	0	0	0	0	0	0
0	0	1	0	0	0	0
0	1	0	0	0	0	0
0	1	1	0	0	0	0
1	0	0	1	0	0	0
1	0	1	0	1	0	0
1	1	0	0	0	1	0
1	1	1	0	0	0	1

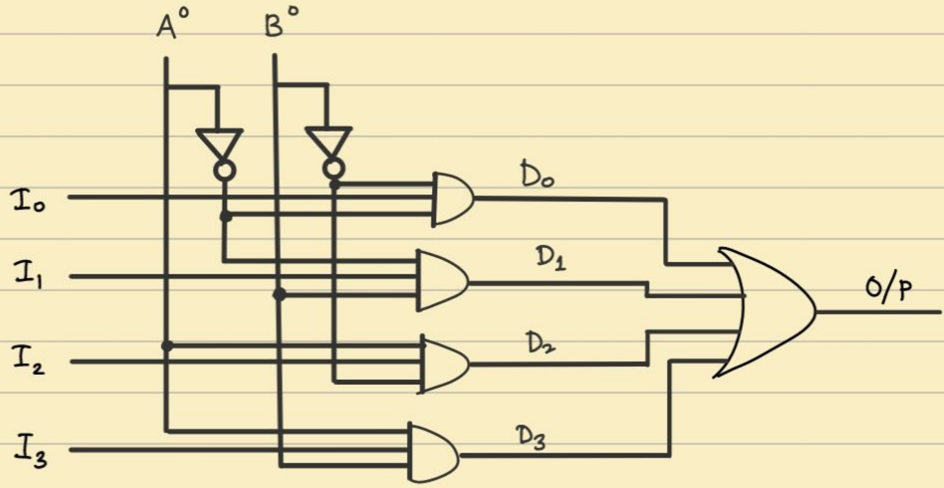
Multiplexer – Variation of this

- 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

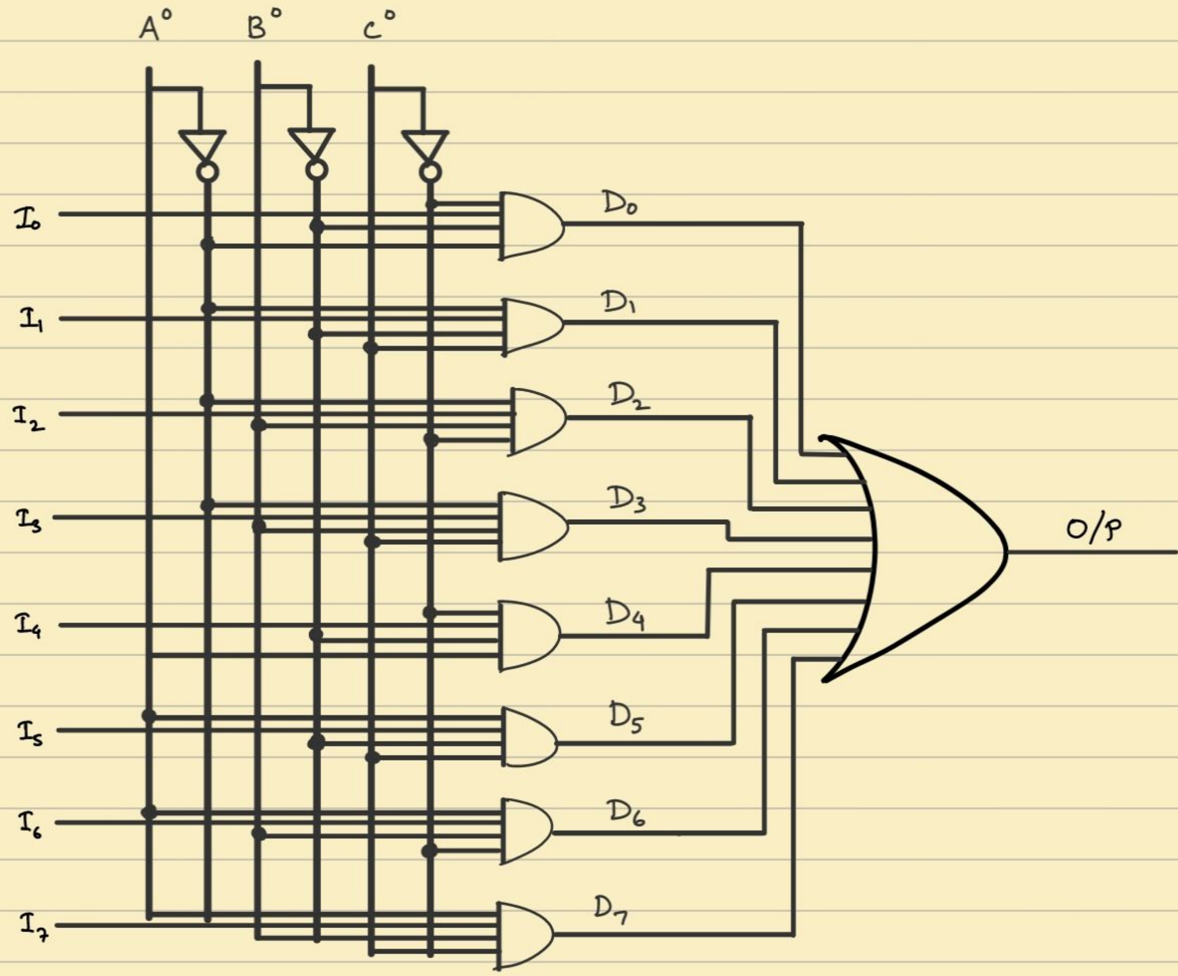
$$\left.\begin{array}{c} 2^n \text{ input lines} \\ \Downarrow \\ n \text{ select lines} \\ \Downarrow \\ 1 \text{ output line} \end{array}\right\} 2^n : 1 \text{ Multiplexer}$$

Note : Multiplexer always has only 1 output



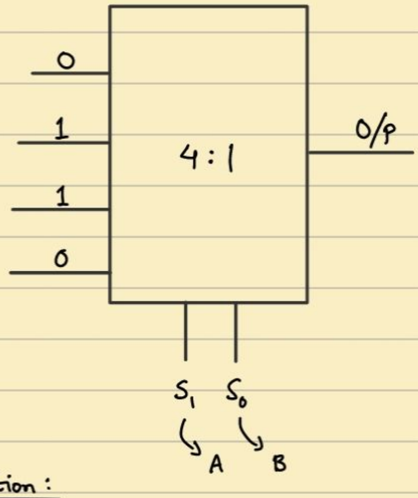


8 : 1 Multiplexer :



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

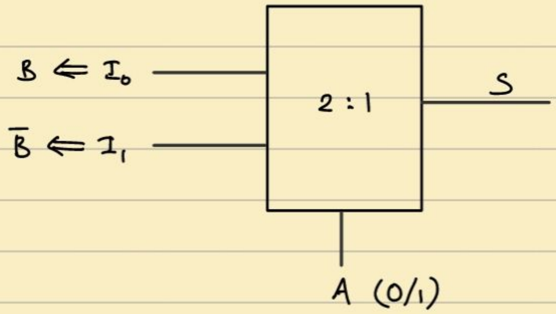


<u>Observation :</u>

When $A = 0 \Rightarrow S = B$

When $A = 1 \Rightarrow S = \overline{B}$

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

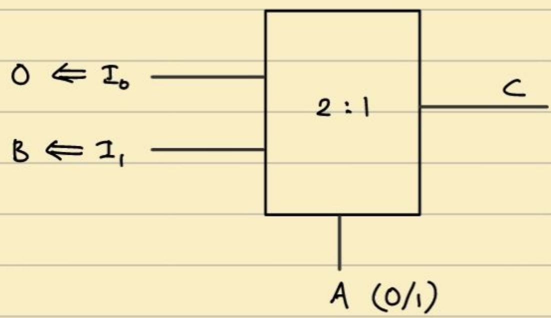
$\therefore$ Multiplexer is used, Decoder cannot realise this.

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

<u>Observation :</u>

When $A = 0 \Rightarrow C = 0$

When $A = 1 \Rightarrow C = B$



<u>H.W</u>

Full Adder – Sum & Carry

<u>9/10</u>

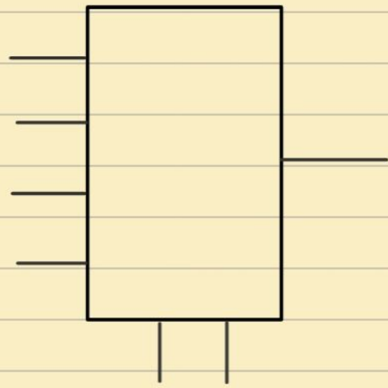
$F(A, B, C) \longrightarrow$ Best case of Multiplexer : 4×1

$\because 2^3$

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

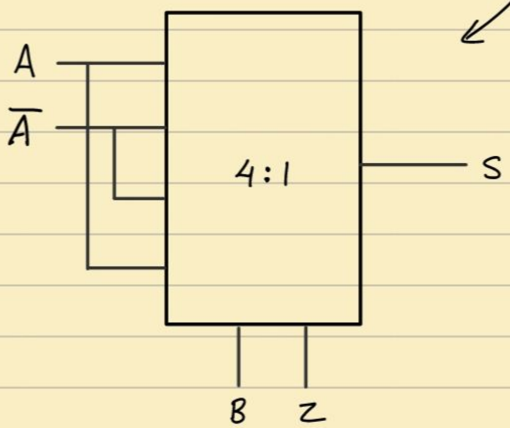


In form of a K-map.

S

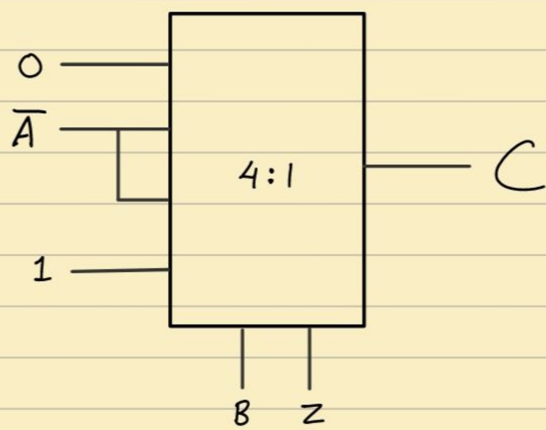
A \ BZ	00	01	10	11
0	0 (0)	1 (1)	1 (2)	0 (3)
1	1 (4)	0 (5)	0 (6)	1 (7)
	A	$\bar{A}$	$\bar{A}$	A

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



C

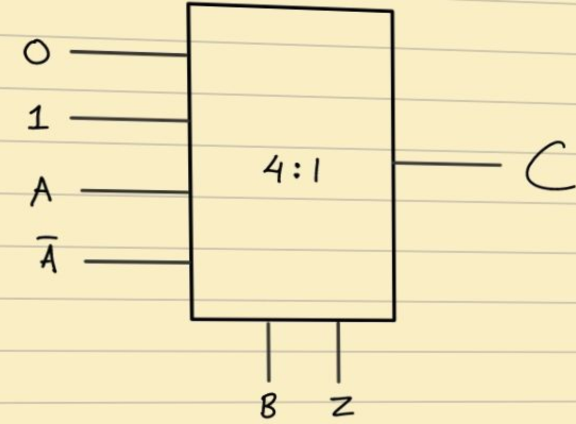
A \ BZ	00	01	10	11
0	0 (0)	0 (1)	0 (2)	1 (3)
1	0 (4)	1 (5)	1 (6)	1 (7)
	0	A	A	1



Note : $\left\{ \begin{array}{l} 1 \ \& \ 1 = 1 \\ 0 \ \& \ 0 = 0 \end{array} \right\}$

Ⓒ

A \ BZ	00	01	10	11
0	0 [0]	1 [1]	0 [2]	1 [3]
1	0 [4]	1 [5]	1 [6]	0 [7]
	0	1	A	$\bar{A}$



→ 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

$A = I_2 + I_3$ ✗

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

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

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

$= B + A\bar{B}$

$= B + A$

Product of Sum — POS — Perspective

↳ i.e. Max terms

i.e. Check for Zeroes

Basic Operation : Sum

Then, Product of sums

$0 \longrightarrow A$

$1 \longrightarrow \overline{A}$

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

$\underline{\underline{(A + B)}} \longrightarrow$ POS

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

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

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

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

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

P.T.O

→ Hardware :

(1) 4 bit BCD adder using Two 4-bit full adders

(2) 2 bit Adder / Subtractor

(1)

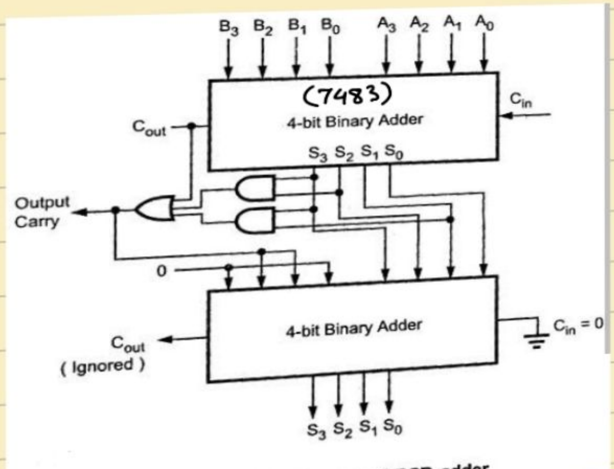
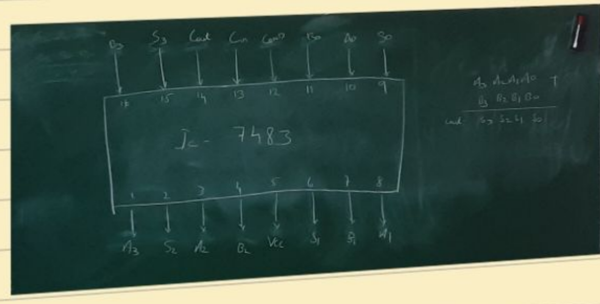
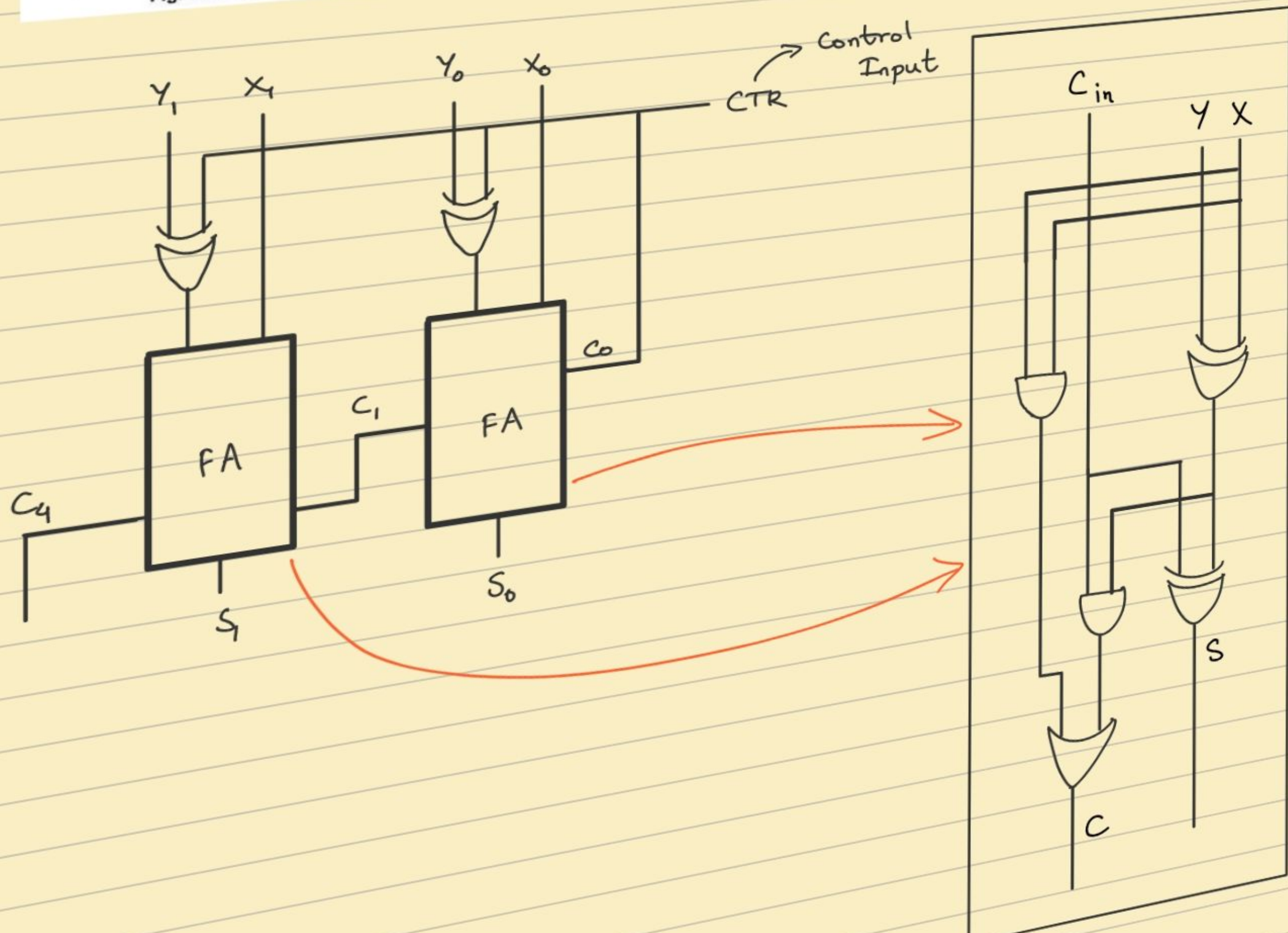


Fig. 3.32 Block diagram of BCD adder



(2)



14/10

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 $I_1 I_0$

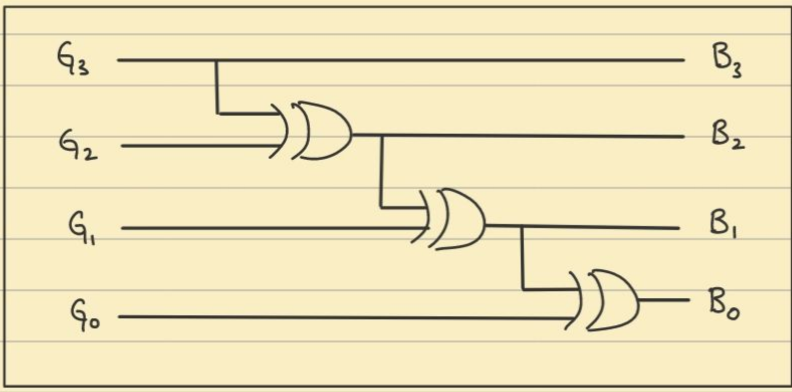
Else :

Return 2^s complement of $I_1 I_0$ as output

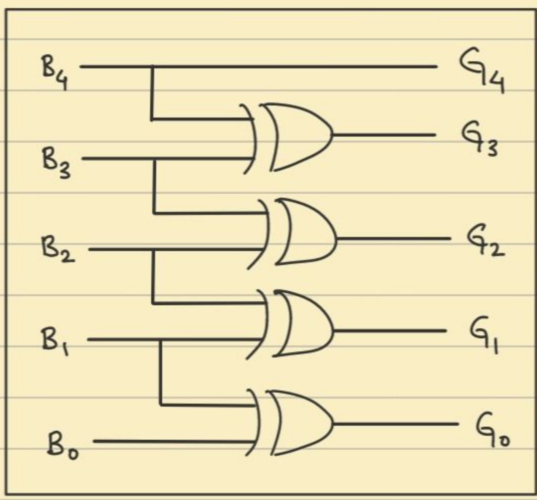
Software:

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.

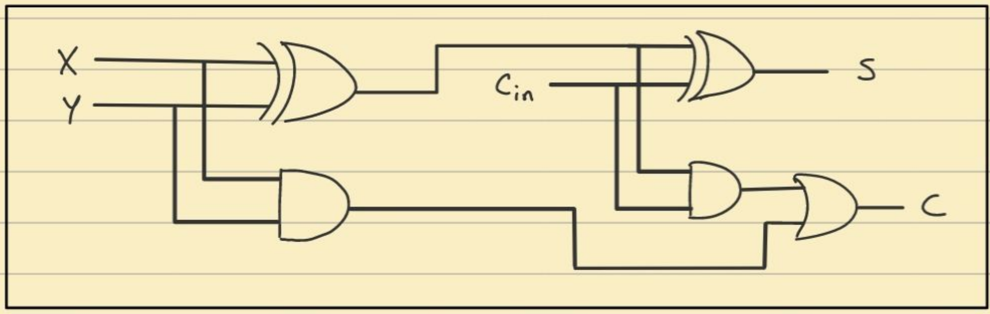
Solution :



G2B Module

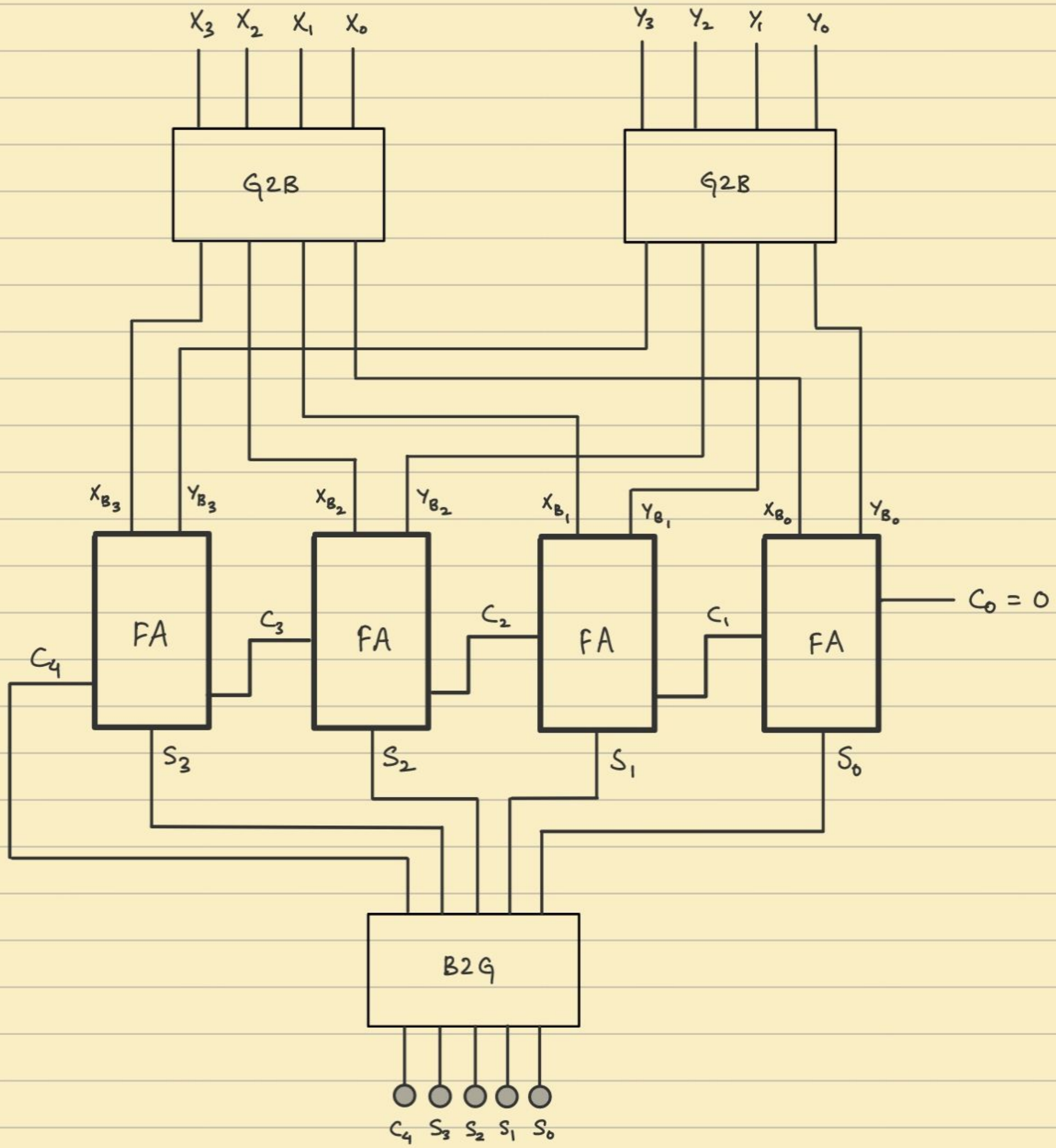


B2G Module



FA module

Main :



10/10

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

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

POS : $A+B$

$$\bar{\bar{F}} = F$$

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

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

A \ BC	00	01	11	10
0	0_{0}	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 \ CD	00	01	11	10
00	1_{0}	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}

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

i.e. Convenient to represent

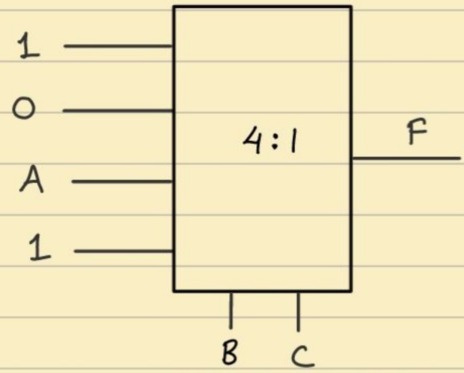
Ex. Solve using SOP method :

$$F(A,B,C,D) = \Pi(13)$$

$\hookrightarrow$ Where 0 is present

Ex. Design a multiplexer

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



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

F

A \ BC	00	01	11	10
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 $\longrightarrow$ 1
Low $\longrightarrow$ 0

But we treat $\begin{pmatrix}\bar{A} \to 1 \\ A \to 0\end{pmatrix}$ just temporarily

(1) 001 — $A+B+\bar{C}$
(3) 011 — $A+\bar{B}+\bar{C}$
(5) 101 — $\bar{A}+B+\bar{C}$

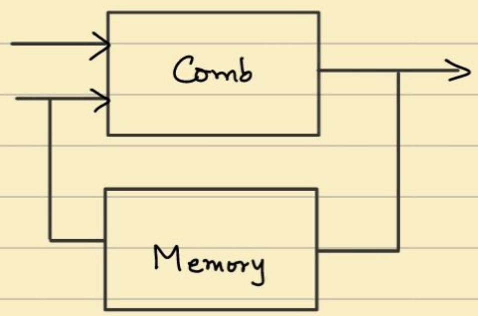
$$\therefore F = (A+B+\bar{C})\cdot(A+\bar{B}+\bar{C})\cdot(\bar{A}+B+\bar{C})$$

Note:

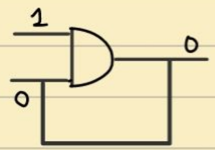
$\Pi = \overline{\Sigma}$ (Negation)

$\longrightarrow$ 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

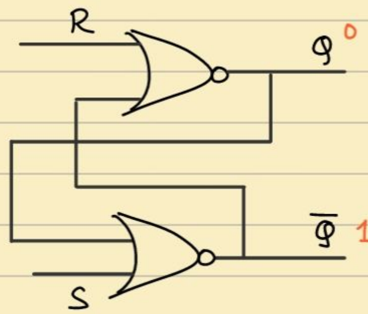


Looping

Asynchronous – Doesn't depend on time
Synchronous – Able to relate (In same state)
→ Time ⟶ Clock
→ Discrete Time events

14/10

→ **RS Latch :**

① 

Previous State : $Q(t)$
Next State : $Q(t+1)$

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

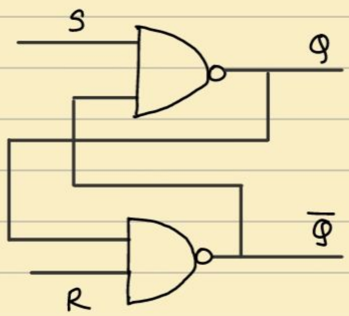
[Trace it for minimum 1.5 cycles for result.]

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

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

R	S	$Q(t+1)$	$\overline{Q}(t+1)$	
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

② 

SR latch

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

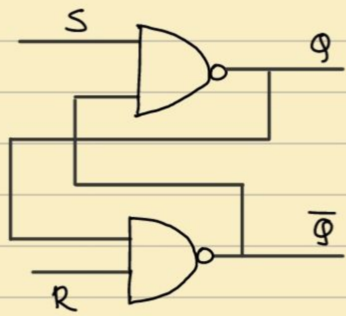
R	S	$Q(t+1)$	$\overline{Q}(t+1)$	
0	1	0	1	} → Reset
1	0	1	0	} → Set
0	0	1	1	} → Indeterminate
1	1	0	1	} → Retain

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

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

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

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

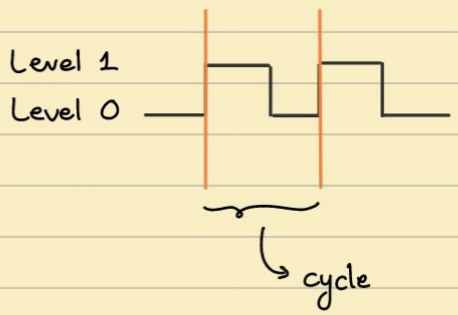


15/10

Clock :

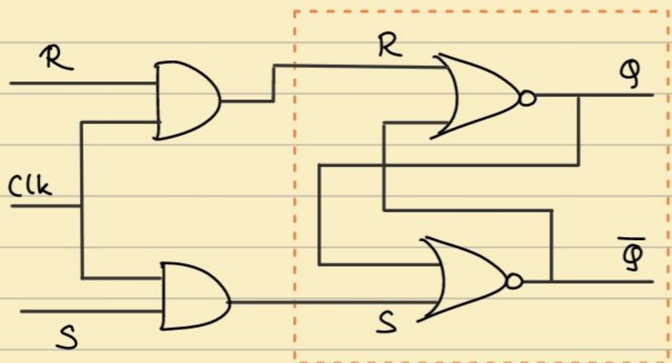
Speed : frequency → 0 1 : Alternatively

Signal



Pulse / Band signal

- **RS Flipflop**



Clk → Clock

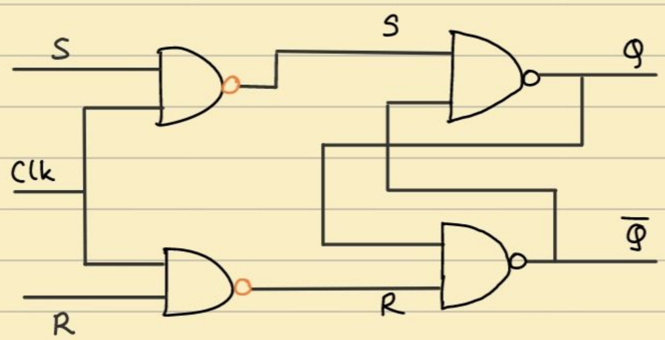
+ve clock cycle = 1
−ve clock cycle = 0

Clock = 0 ⇒ Output is retained

Clock = 1 ⇒ RS latch depending on R & S

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

SR flipflop :



Set : S = 1 , R = 0

→ Operated when the circuit is low

→ If NOT was not performed

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

Clock = 0 ⇒ Output is retained

Clock = 1 ⇒ SR Latch depending on S & R

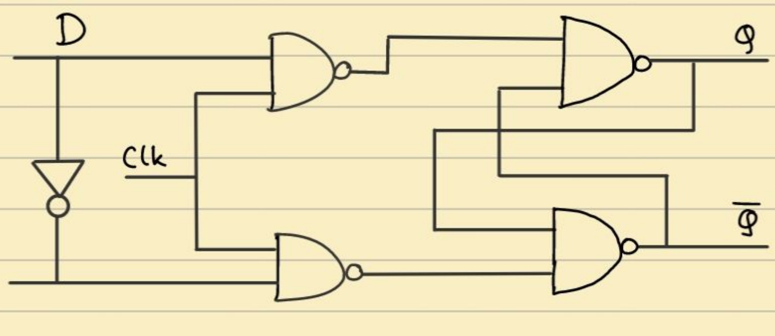
For not retaining, We use D flip-flop.

→ We cannot give clock input & retain the value.

D : Data

↓

Connect S to R using NOT gate



A Pulse based flipflop Cannot retain beyond one clock cycle

Disadvantage : Cannot retain more than 1 cycle

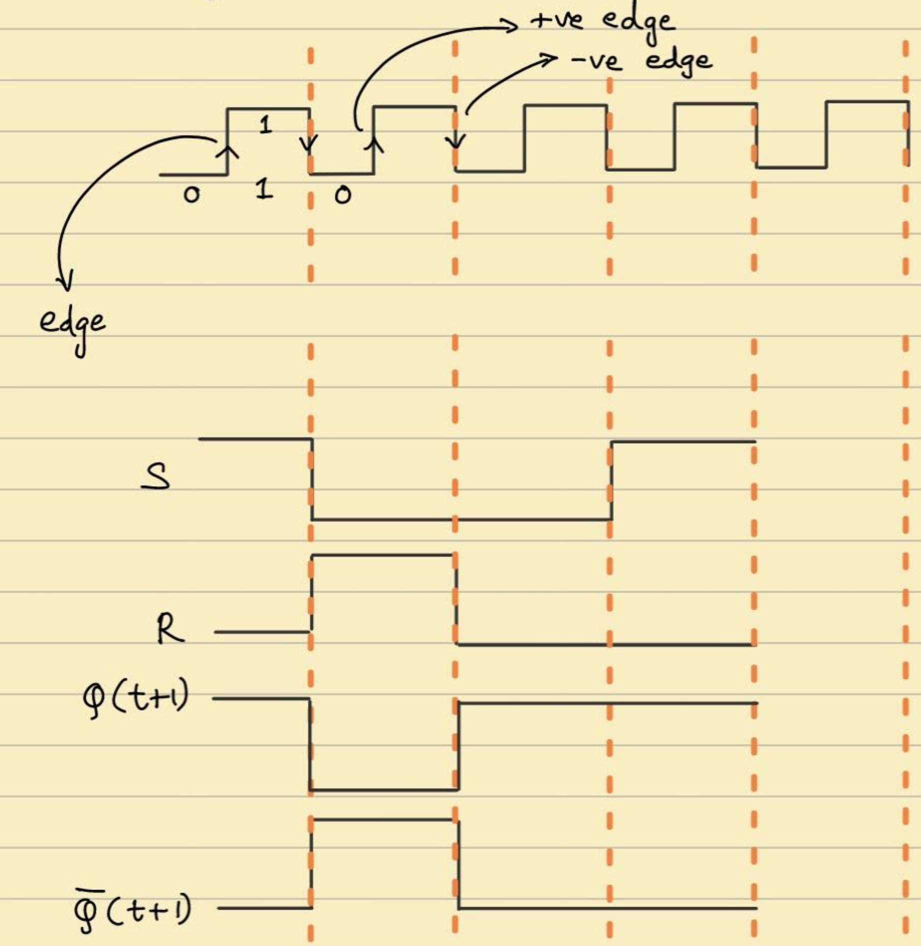
Advantage : No Indeterminate state

∴ Operation of D Flipflop is proper.

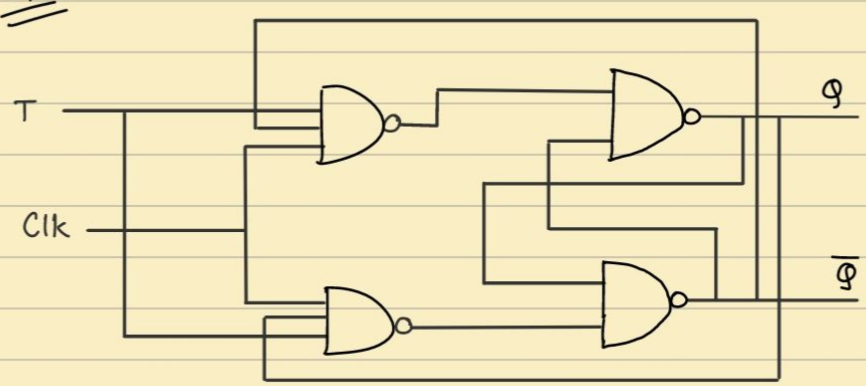
	D	$Q(t+1)$	$\bar{Q}(t+1)$
	0	0	1
	1	1	0
	X	Q_t	$Q(t+1)$
When Clock = 0 ,	X	Q_t	$Q(t+1)$

Charecteristic Table (AKA Truth Table)

Time Diagram :



18/10

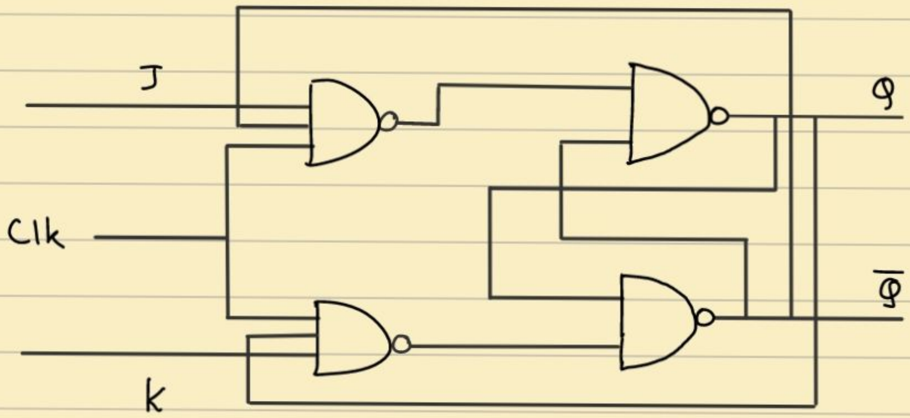


← T - flip flop

Avoids indeterminate state

T : Toggle

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

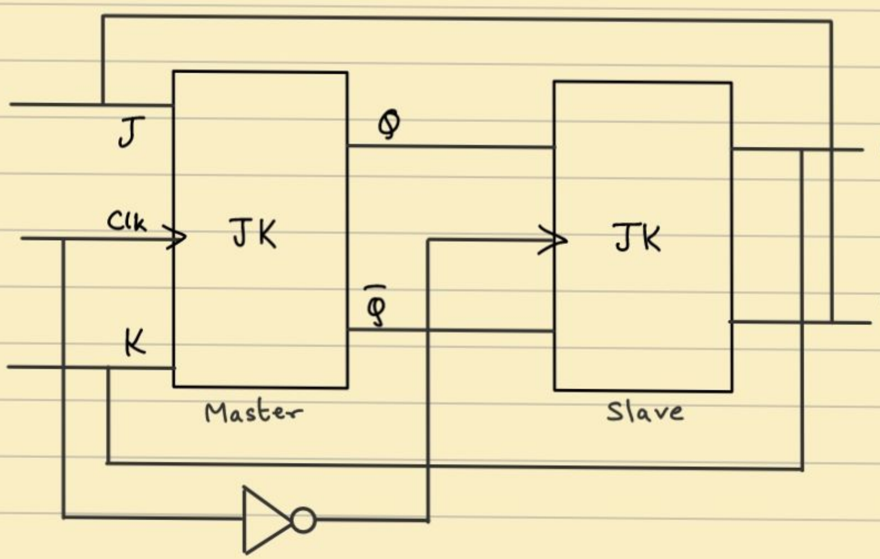


Q	J	K	$Q(t+1)$	
0	0	0	0	Retain
0	0	1	0	Reset
0	1	0	1	Set
0	1	1	1	Toggle
1	0	0	1	Retain
1	0	1	0	Reset
1	1	0	1	Set
1	1	1	0	Toggle

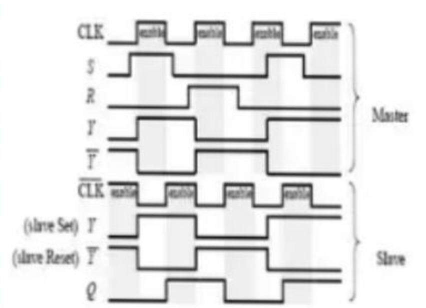
Q \ JK	00	01	11	10
0	0 (0)	0 (1)	1 (3)	1 (2)
1	1 (4)	0 (5)	0 (7)	1 (6)

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

→ <u>Master Slave Flip flop</u> :

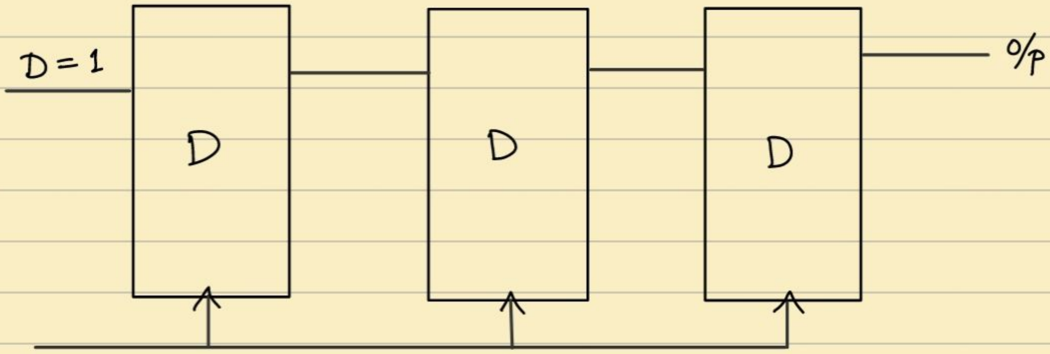


$\overline{PRE}$	$\overline{CLR}$	CLK	S	R	Q	$\overline{Q}$	Mode
0	1	X	X	X	1	0	preset
1	0	X	X	X	0	1	cleared
0	0	X	X	X	1	1	not used (race)
1	1	⎍	0	0	0	0	hold
1	1	⎍	0	1	0	1	Reset
1	1	⎍	1	0	1	0	Set
1	1	⎍	1	1	1	1	not used (race)



• T-flip flop :

→ Edge triggered Flip flop :



Shift Register

Flip flop level → D cannot operate

Positive Pulse ✗

Edge Triggered ✓

When edge = 1 , First clock cycle
Gets enabled.
Edge becomes pulse
Pulse cannot change input ∵ Edge based Flip-flop

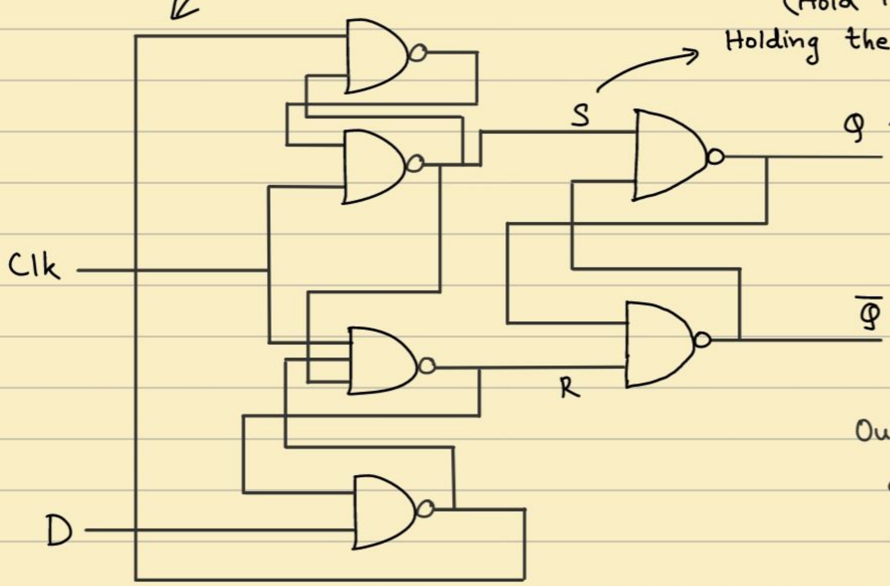
Negative edge cycle — Nothing happens.

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.

Setup time

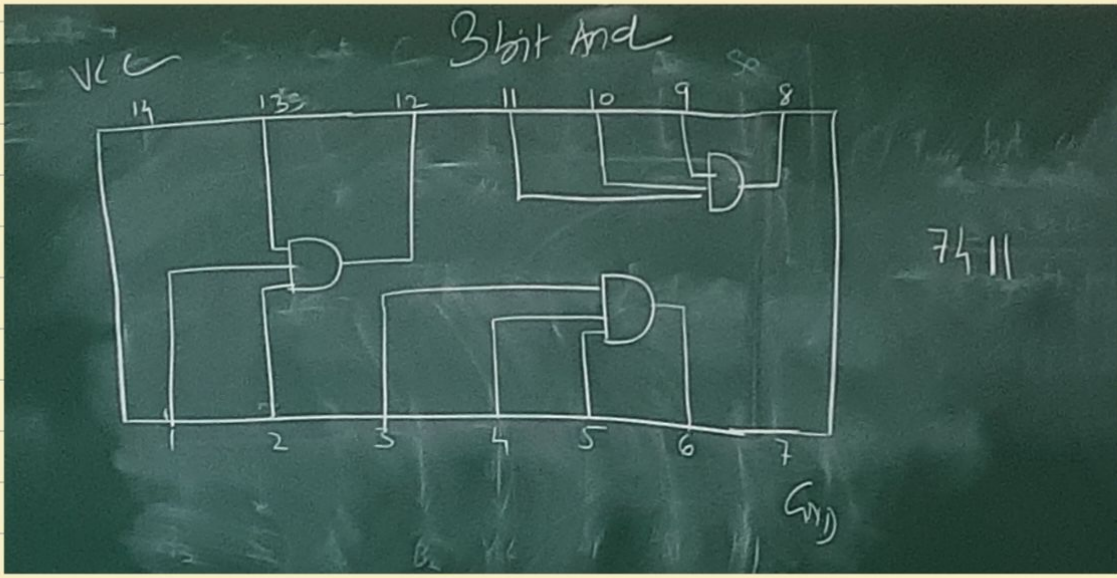
(Hold Time)
Holding the input to get output



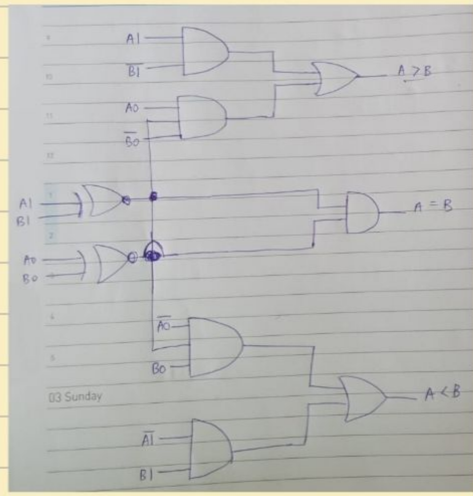
Output will be displayed after a delay of one clock tick.

19/10

Lab - 6



Make a comperator circuit (2 bit)



P.T.O

22/10

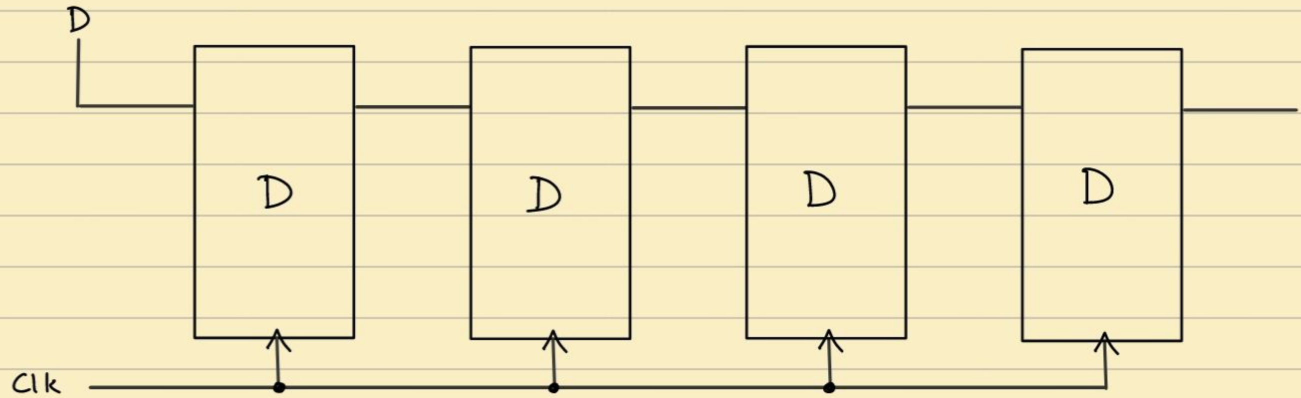
→ **Serial Converters:**

Registers

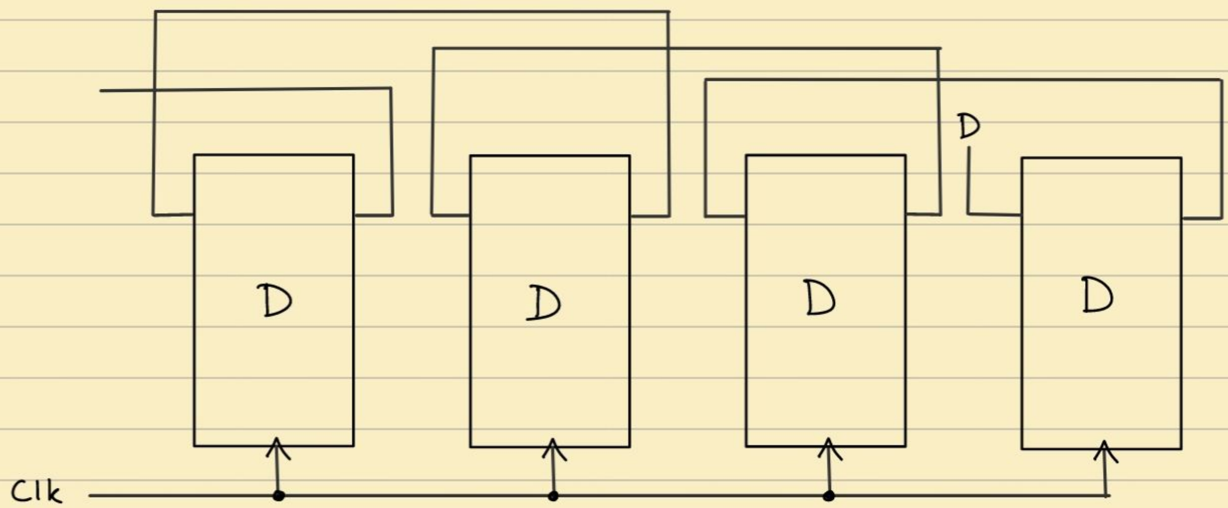
Left → Right   Right → Left

Edge Triggered - If D will take 4 cycles

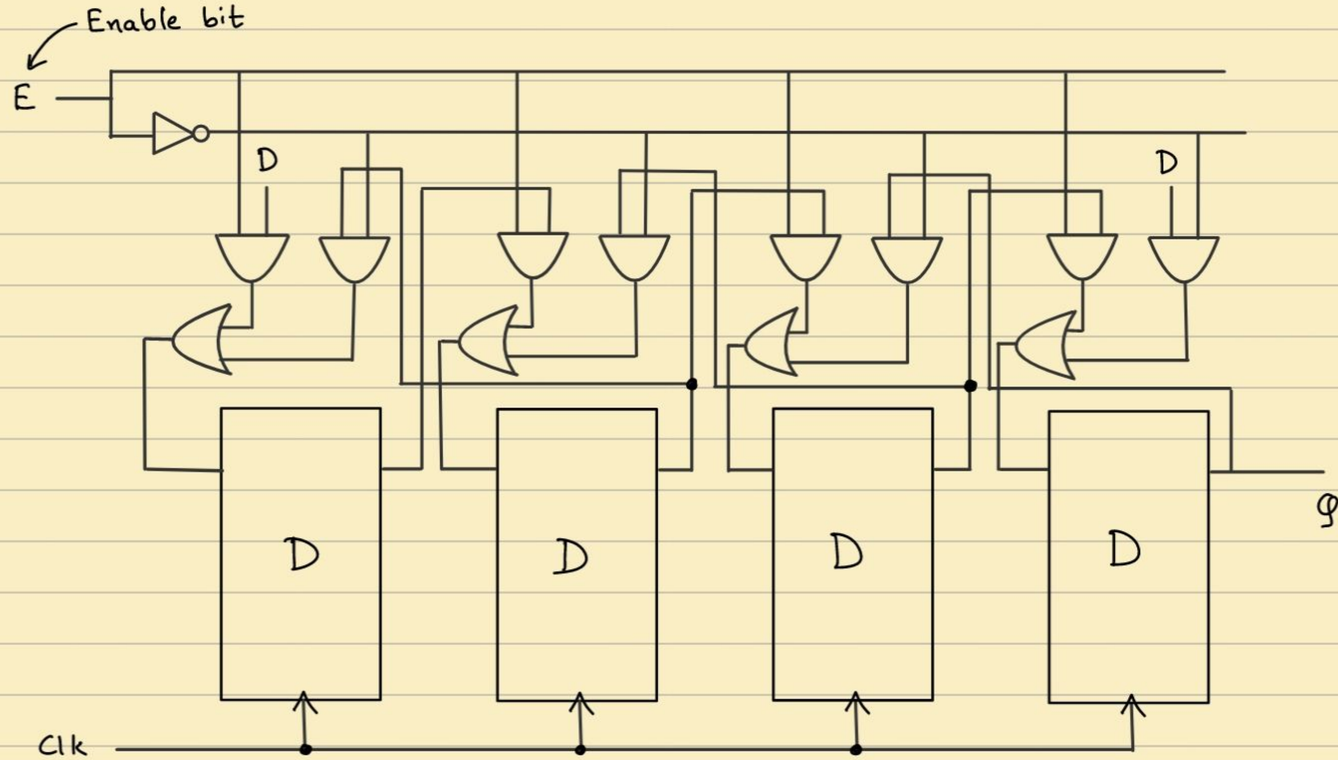
- **Right Shift Register:**



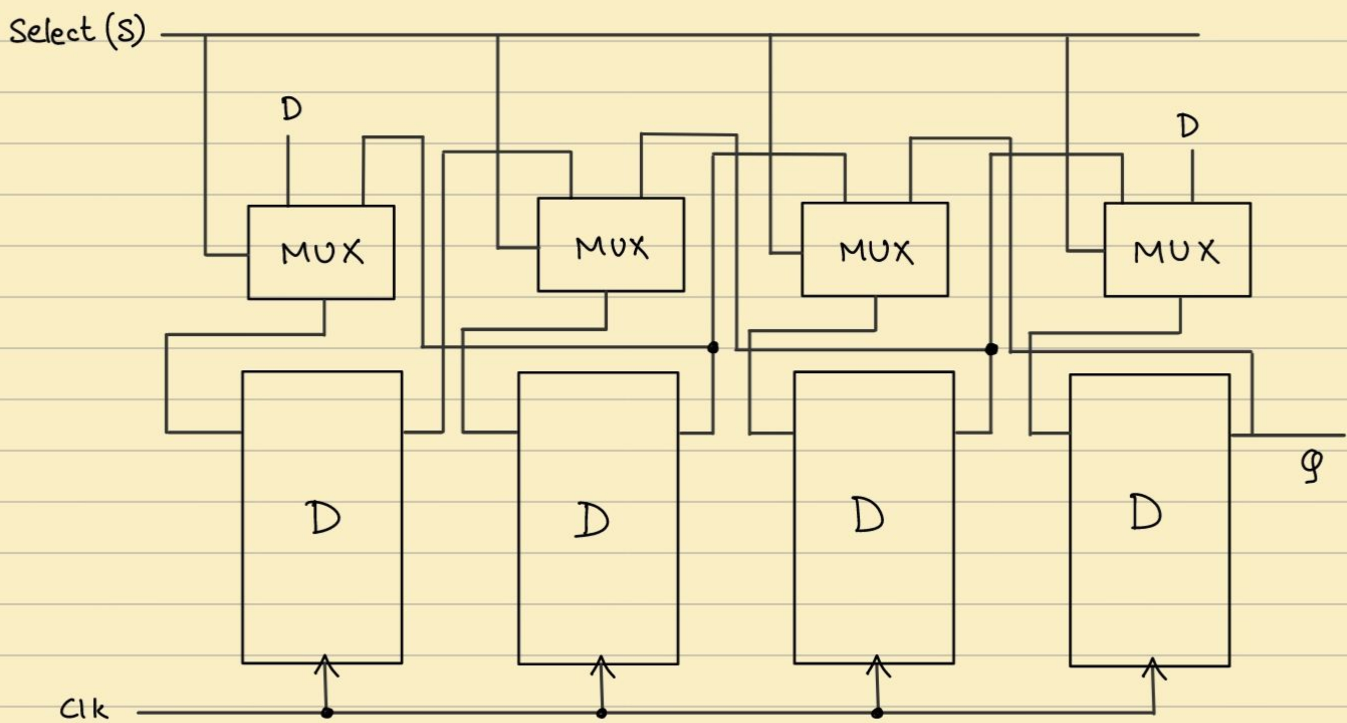
- **Left Shift Register:**



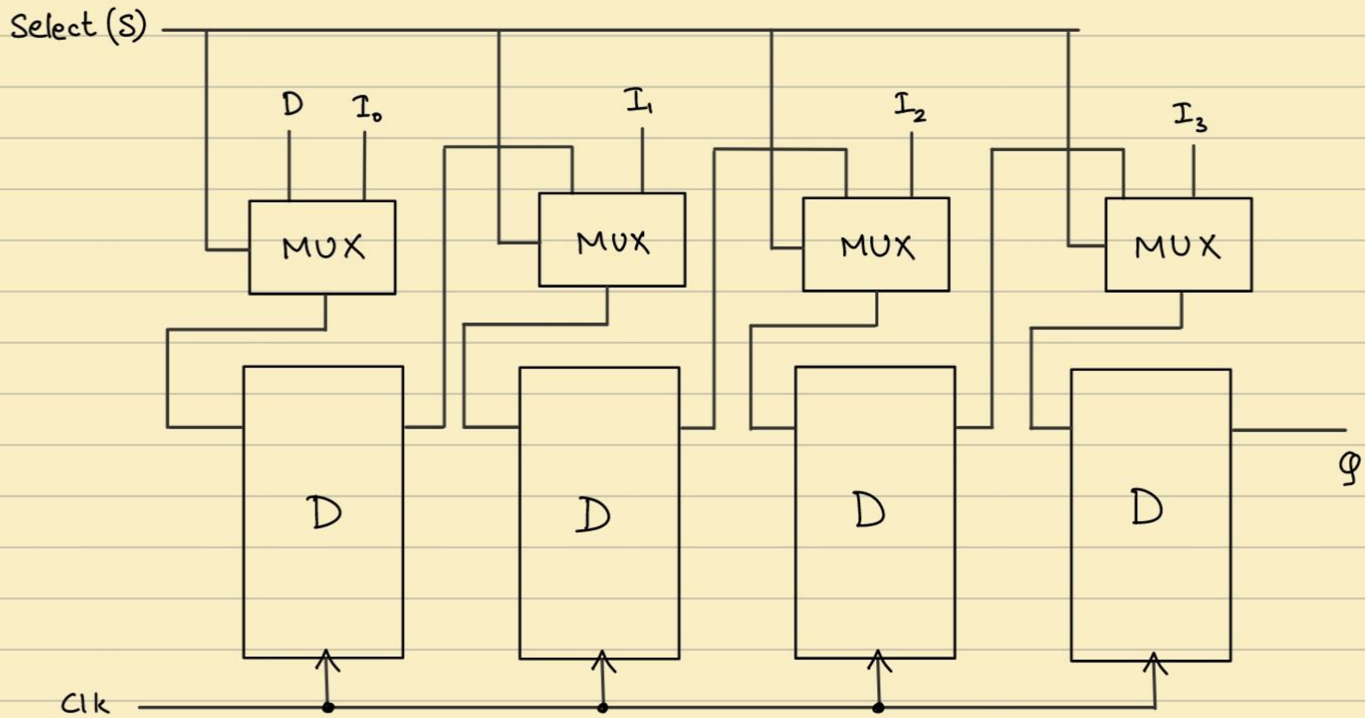
23/10



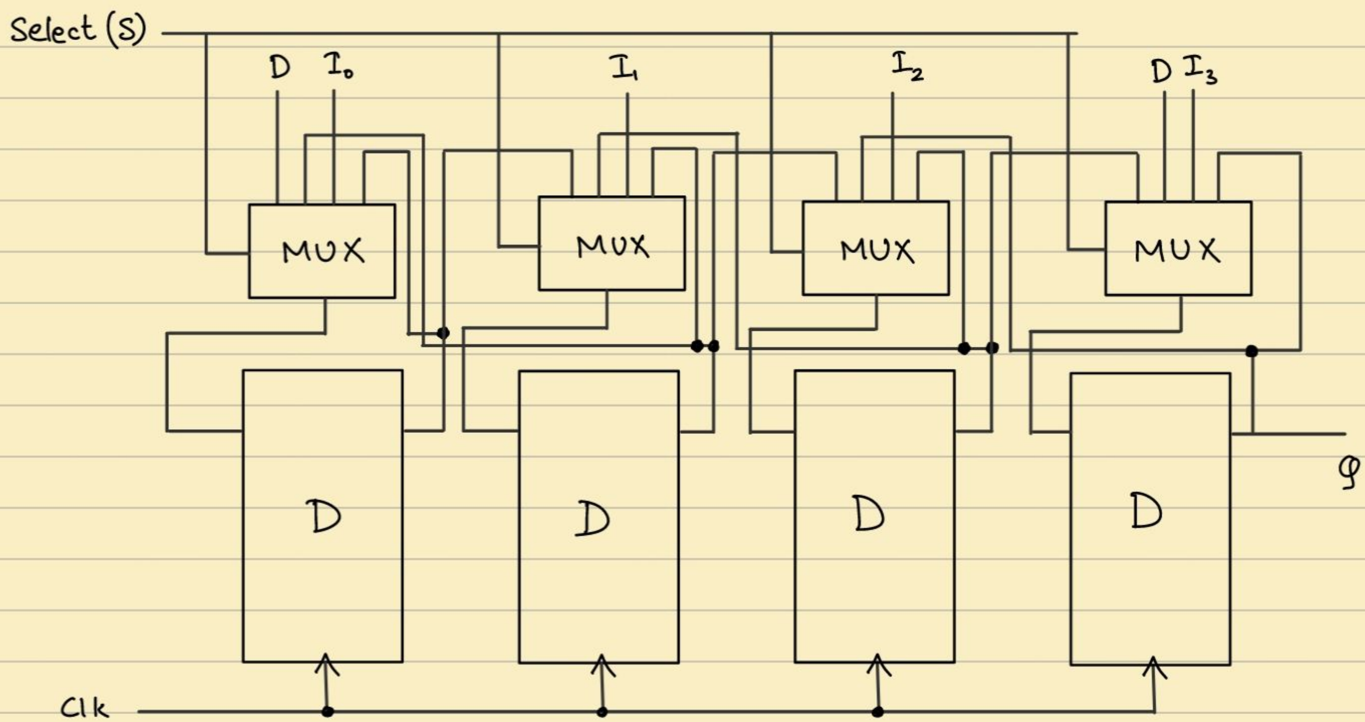
→ Parallel to Serial converter :



→ Parallel to Serial converter & Right Shift:



→ Universal Shift Register:

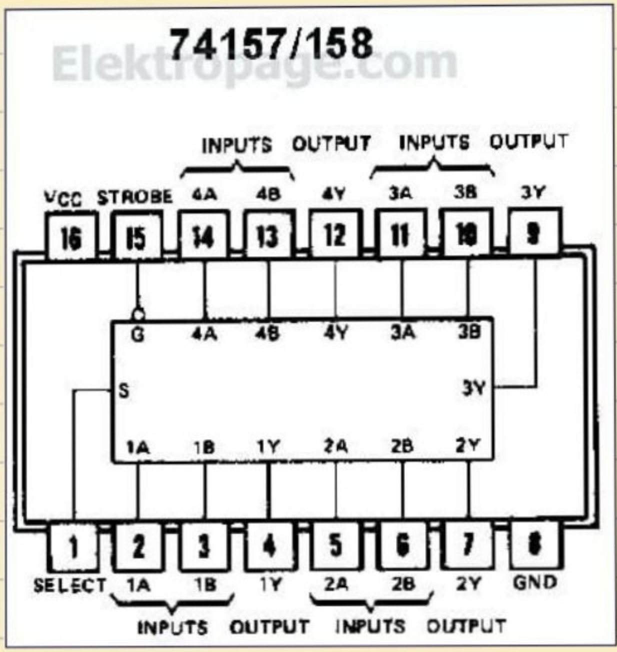


Right Shift : 00

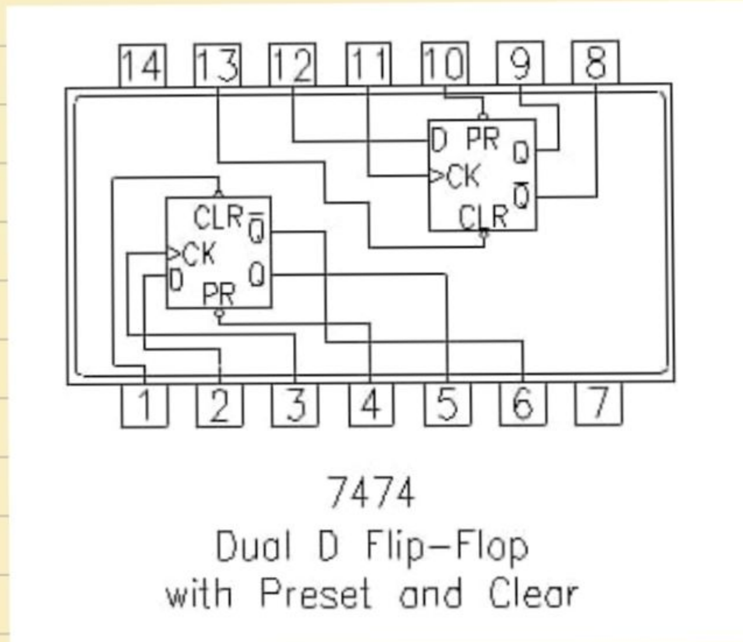
Left Shift : 01

Parallel Input : 11

Retain (Hold) : 10

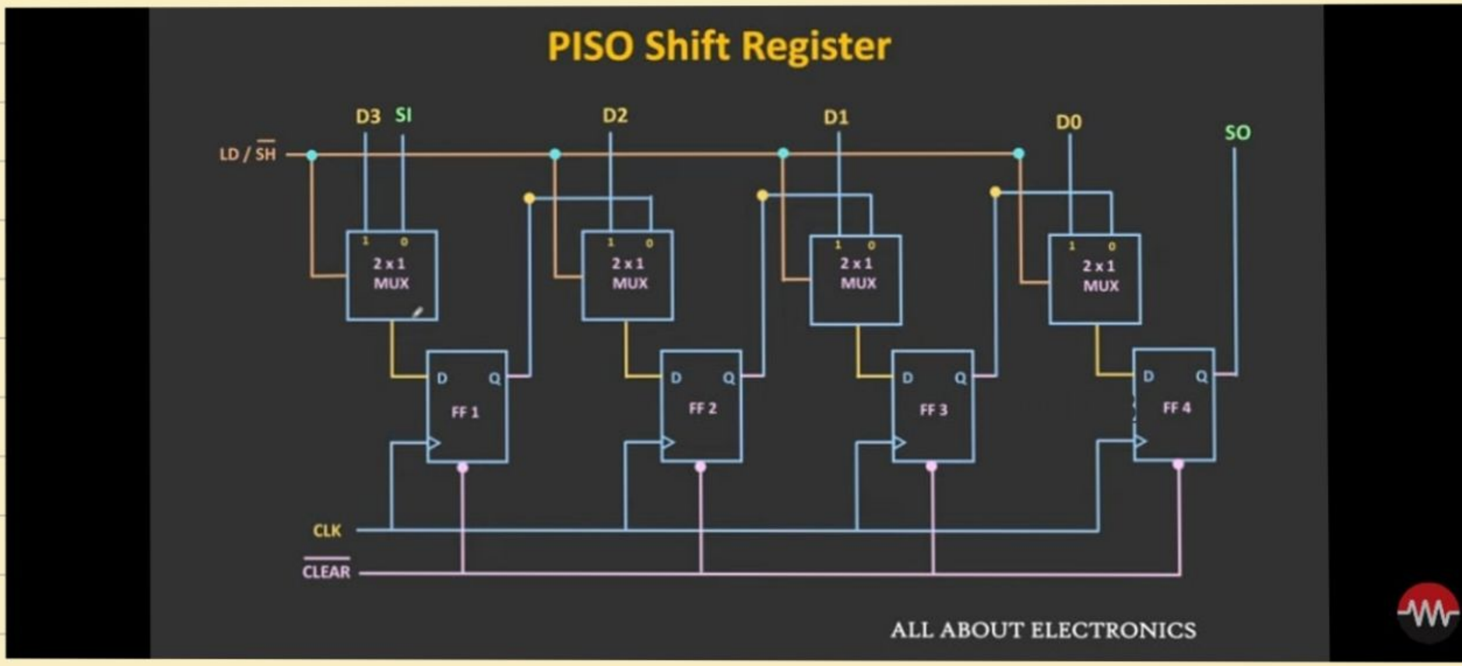


74157/158
Elektropage.com
INPUTS OUTPUT INPUTS OUTPUT
VCC STROBE 4A 4B 4Y 3A 3B 3Y
16 15 14 13 12 11 10 9
G 4A 4B 4Y 3A 3B
S 3Y
1A 1B 1Y 2A 2B 2Y
1 2 3 4 5 6 7 8
SELECT 1A 1B 1Y 2A 2B 2Y GND
INPUTS OUTPUT INPUTS OUTPUT



14 13 12 11 10 9 8
D PR Q
>CK $\overline{Q}$
CLR
CLR $\overline{Q}$
>CK
D Q
PR
1 2 3 4 5 6 7
7474
Dual D Flip-Flop
with Preset and Clear

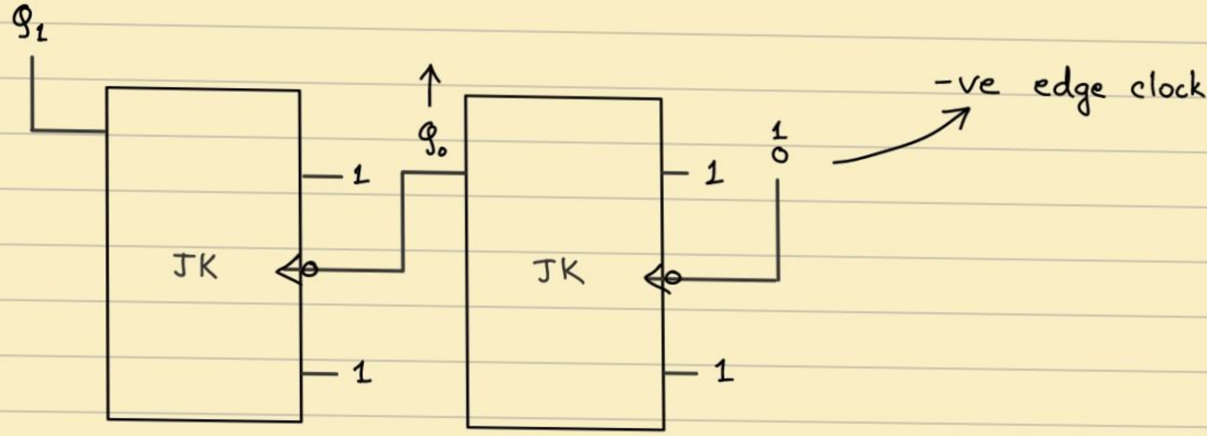
9)



PISO Shift Register
D3 SI
D2
D1
D0
SO
LD / $\overline{SH}$
1 0
2 x 1
MUX
D Q
FF 1
FF 2
FF 3
FF 4
CLK
$\overline{CLEAR}$
ALL ABOUT ELECTRONICS

Toggle

$0 \rightarrow 1$
$1 \rightarrow 0$



Clk	Q_1	Q_0
0	0	0
1	0	1
2	1	0
3	1	1
4	0	0

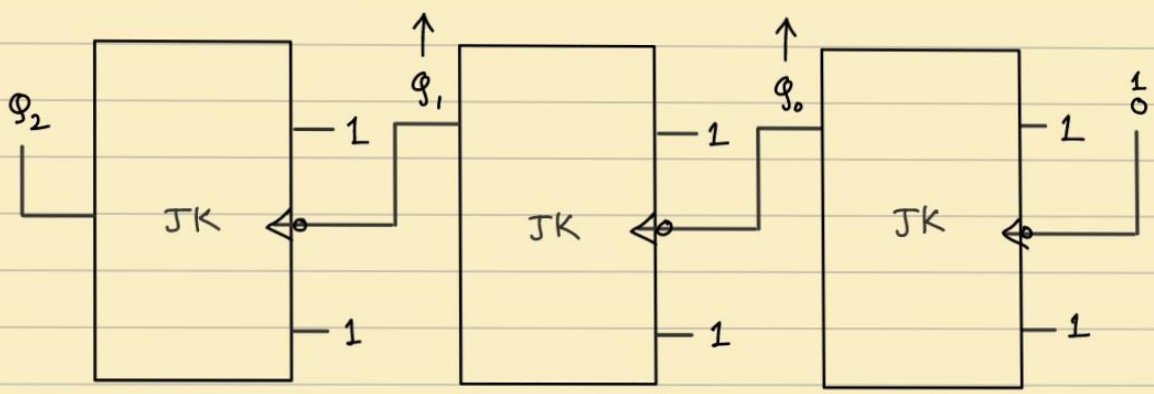
+ve edge
-ve edge
+ve edge
-ve edge

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.

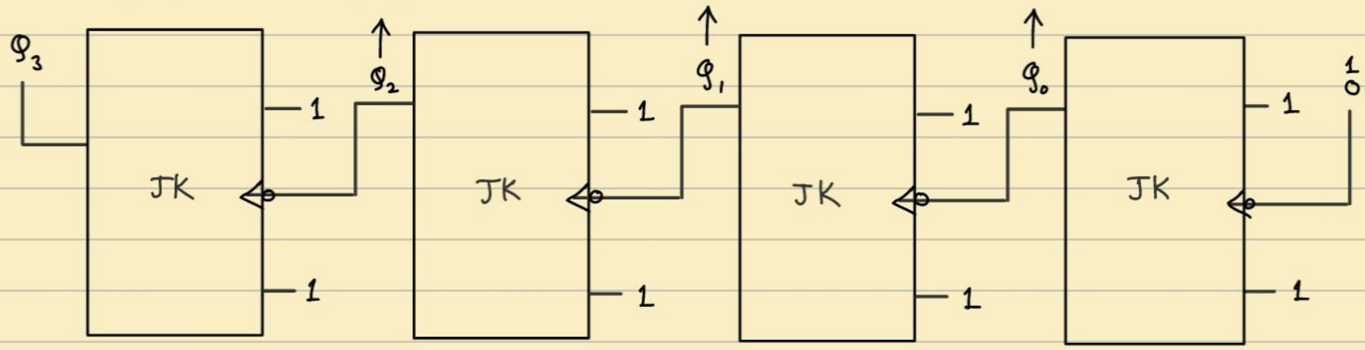
This counter is called modulo 4 counter. ∵ 0 – 3 repeats
[modulo 10 counter : BCD]

Modulo 8 counter (Asynchronous)



Clk	Q_2	Q_1	Q_0
0	0	0	0
1	0	0	1
2	0	1	0
3	0	1	1

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 :**

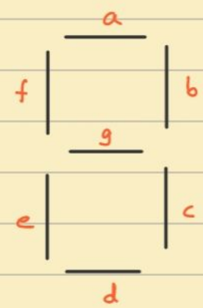
Display :

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



a

B_0B_1 \ B_2B_3	00	01	11	10
00	1_{0}	0_{1}	0_{3}	0_{2}
01	1_{4}	1_{5}	0_{7}	1_{6}
11	X_{12}	X_{13}	X_{15}	X_{14}
10	1_{8}	1_{9}	X_{11}	X_{10}

b

B_0B_1 \ B_2B_3	00	01	11	10
00	1_{0}	0_{1}	0_{3}	1_{2}
01	0_{4}	0_{5}	0_{7}	1_{6}
11	X_{12}	X_{13}	X_{15}	X_{14}
10	1_{8}	0_{9}	X_{11}	X_{10}

c

B_0B_1 \ B_2B_3	00	01	11	10
00	0_{0}	0_{1}	1_{3}	1_{2}
01	0_{4}	1_{5}	0_{7}	1_{6}
11	X_{12}	X_{13}	X_{15}	X_{14}
10	1_{8}	0_{9}	X_{11}	X_{10}

d

B_0B_1 \ B_2B_3	00	01	11	10
00	1_{0}	1_{1}	1_{3}	0_{2}
01	1_{4}	1_{5}	1_{7}	1_{6}
11	X_{12}	X_{13}	X_{15}	X_{14}
10	1_{8}	1_{9}	X_{11}	X_{10}

e

B_0B_1 \ B_2B_3	00	01	11	10
00	1_{0}	1_{1}	1_{3}	1_{2}
01	1_{4}	0_{5}	1_{7}	0_{6}
11	X_{12}	X_{13}	X_{15}	X_{14}
10	1_{8}	1_{9}	X_{11}	X_{10}

f

B_0B_1 \ B_2B_3	00	01	11	10
00	1_{0}	0_{1}	1_{3}	1_{2}
01	0_{4}	1_{5}	1_{7}	1_{6}
11	X_{12}	X_{13}	X_{15}	X_{14}
10	1_{8}	1_{9}	X_{11}	X_{10}

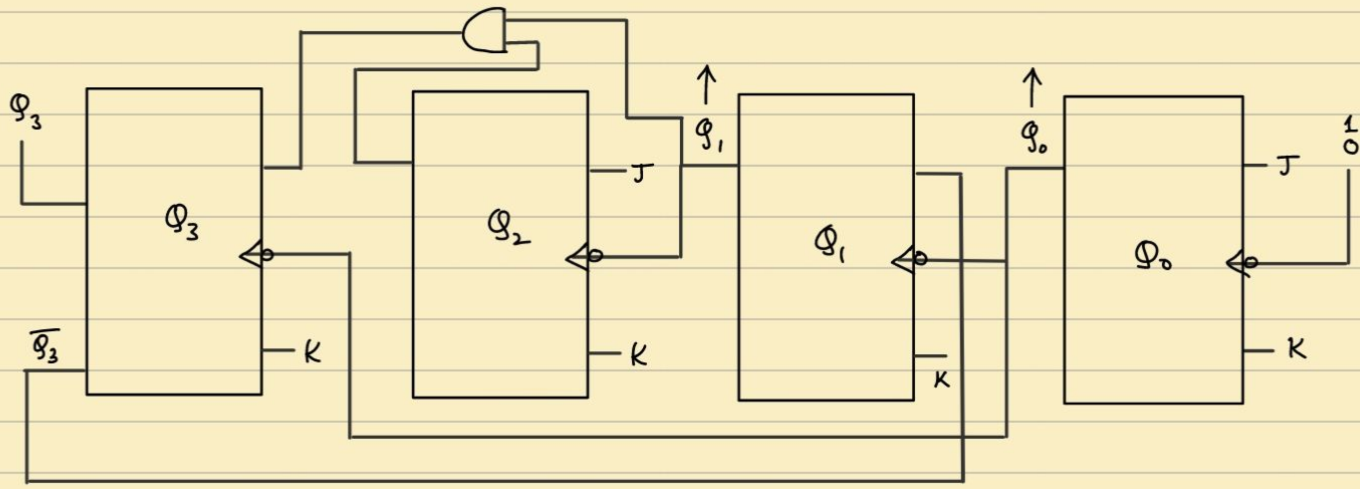
g

B_0B_1 \ B_2B_3	00	01	11	10
00	0_{0}	0_{1}	1_{3}	1_{2}
01	1_{4}	1_{5}	0_{7}	1_{6}
11	X_{12}	X_{13}	X_{15}	X_{14}
10	1_{8}	1_{9}	X_{11}	X_{10}

$$a = B_0 + B_1\overline{B_2} + B_1\overline{B_3} + \overline{B_2}\,\overline{B_3}$$
$$b = \overline{B_1}\,\overline{B_3} + B_2\overline{B_3}$$
$$c = B_0\overline{B_3} + B_2\overline{B_3} + \overline{B_1}B_2 + B_1\overline{B_2}B_3$$
$$d = B_1 + \overline{B_2} + B_3$$
$$e = B_0 + \overline{B_1} + B_2B_3 + \overline{B_2}\,\overline{B_3}$$
$$f = B_0 + B_2 + B_1B_3 + \overline{B_1}\,\overline{B_3}$$
$$g = B_0 + B_2\overline{B_3} + B_1\overline{B_2} + \overline{B_0}\,\overline{B_1}B_2$$

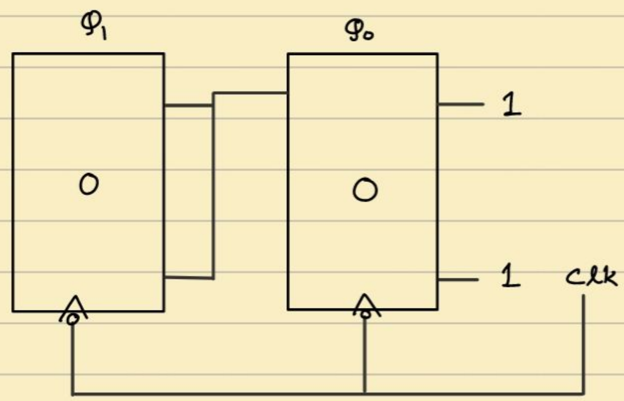
25/10

→ **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:**

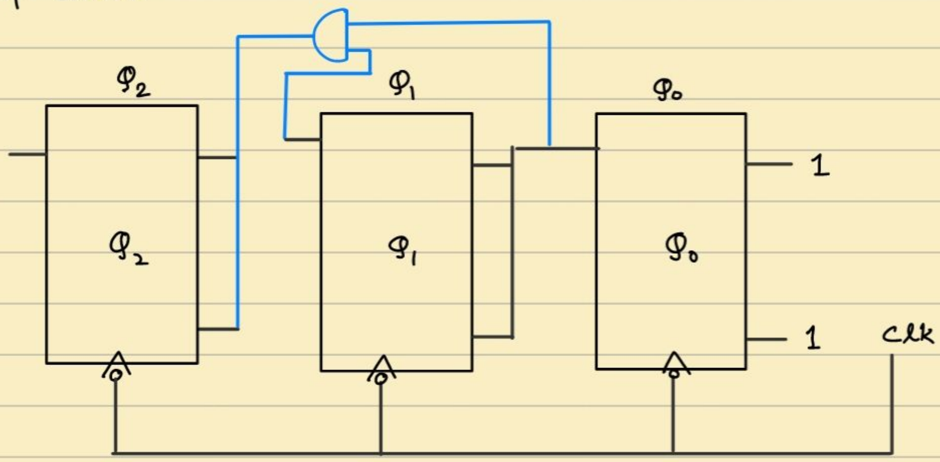


Depends on the previous output & not on the clock.
∴ 1 1

Q_1	Q_0
0	0
0	1
1	0
1	1

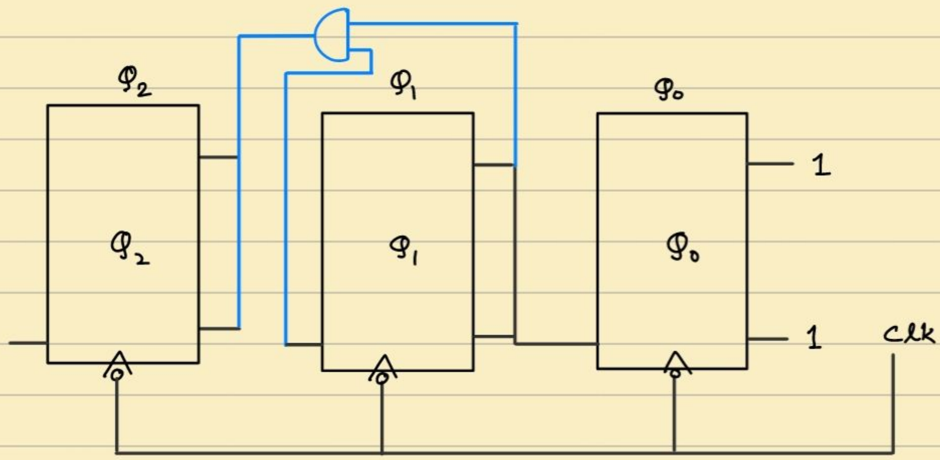
→ Synchronous mod-8 Counter:

- Up Counter:



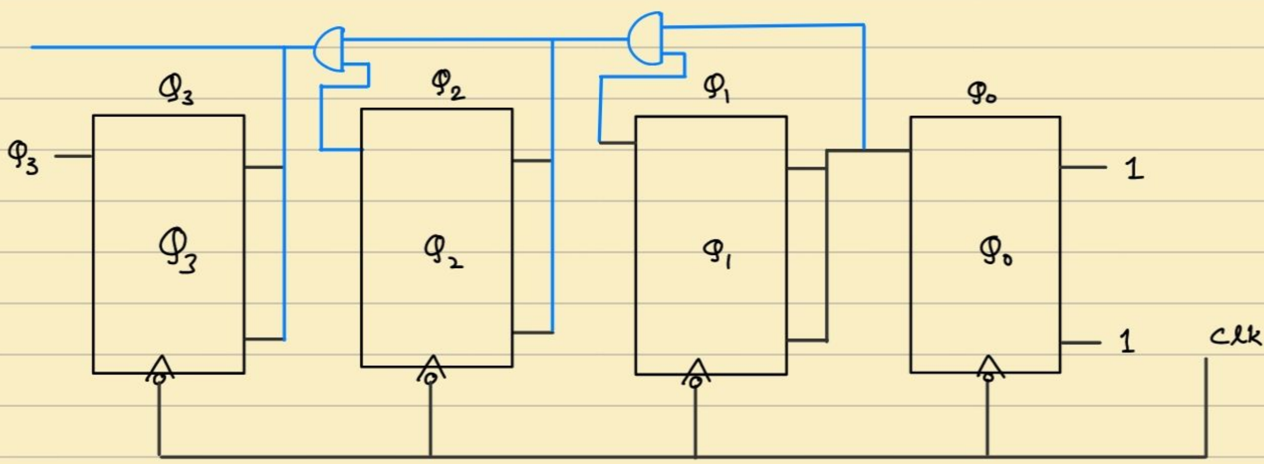
Q_2	Q_1	Q_0
0	0	0
0	0	1
0	1	0
0	1	1
1	0	0
1	0	1
1	1	0
1	1	1

- Down counter:

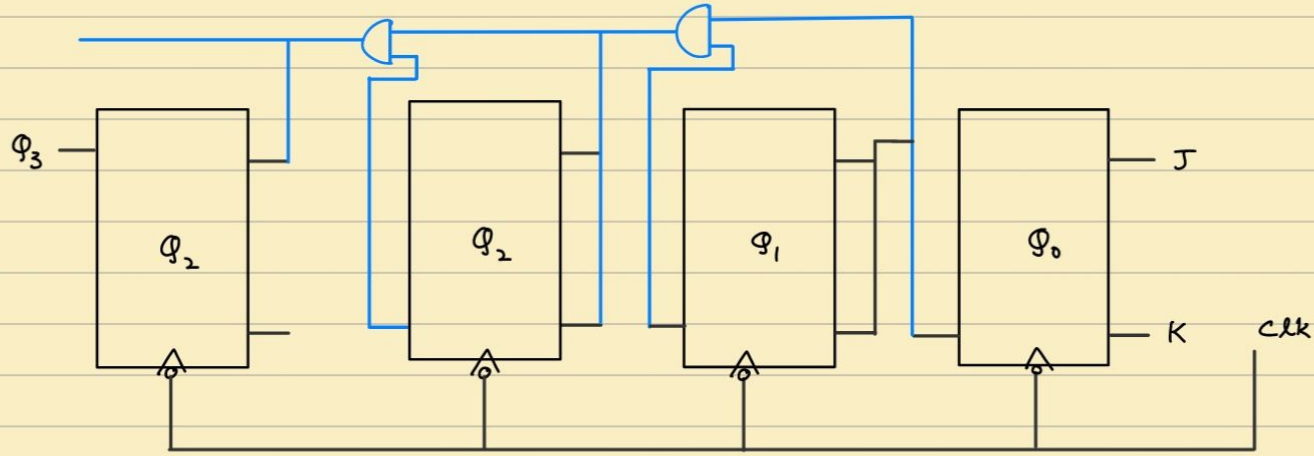


Q_2	Q_1	Q_0
1	1	1
1	1	0
1	0	1
1	0	0
0	1	1
0	1	0
0	0	1
0	0	0

→ Synchronous mod-16 Up counter:



- Synchronous mod-16 Down counter:



Output is taken from Q.
But Circuit operates on $\overline{Q}$.

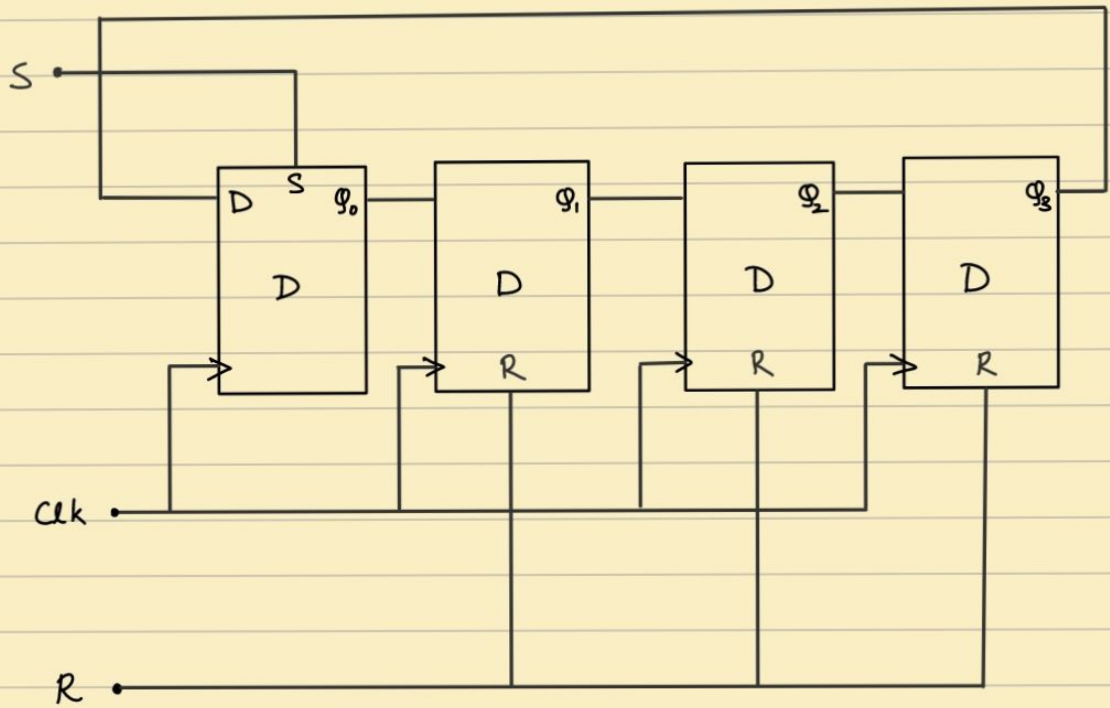
Up-Counter:

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

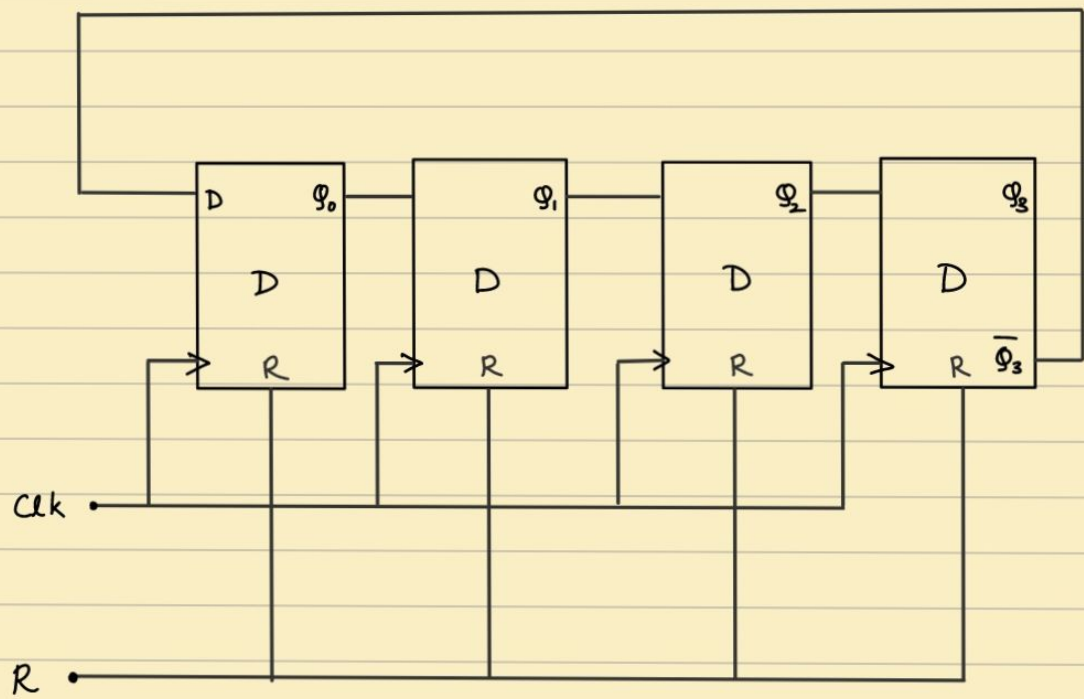
→ Ring Counter



	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)

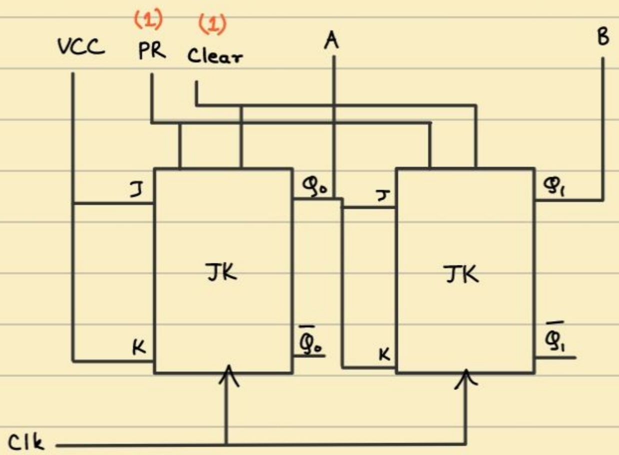


Q_3	Q_2	Q_1	Q_0
0	0	0	0
1	0	0	0
1	1	0	0
1	1	1	0
1	1	1	1
0	1	1	1
0	0	1	1
0	0	0	1
0	0	0	0
⋮	⋮	⋮	⋮

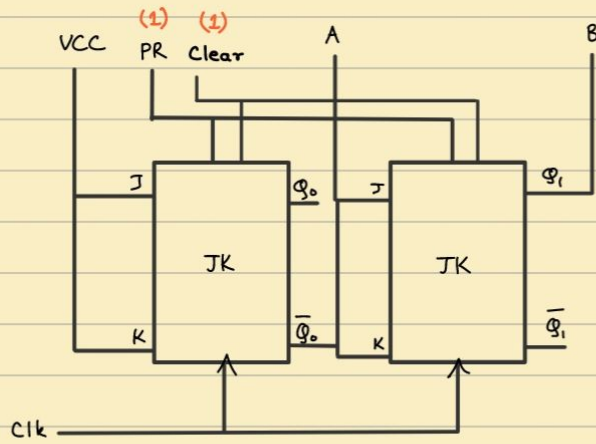
28/10

Lab - 8

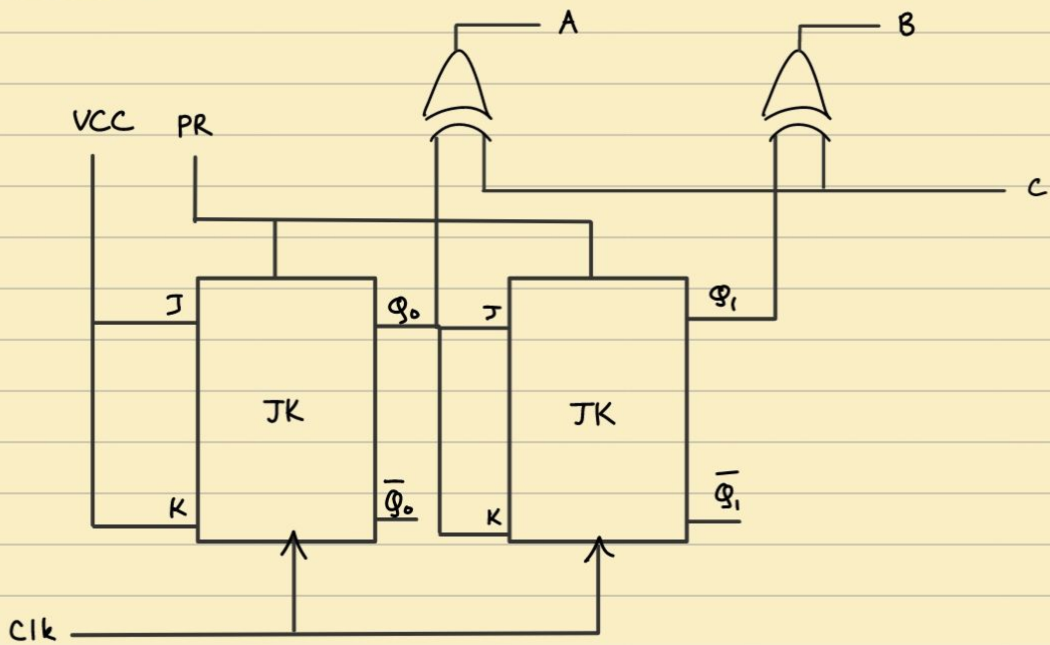
Q1) Up - Counter :



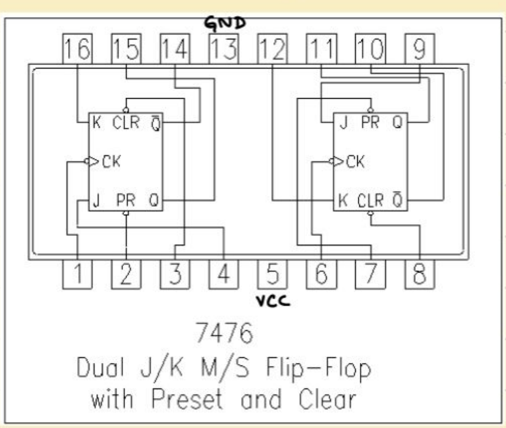
Q2) Down - Counter :

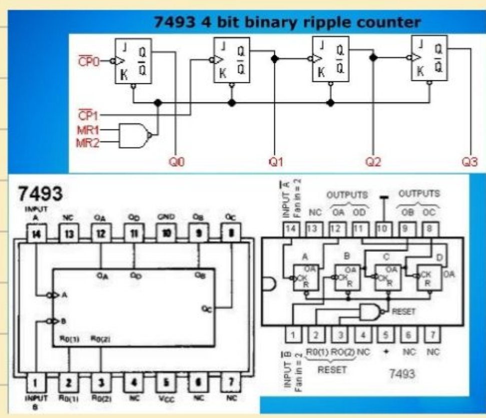


Q3) Combined :



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



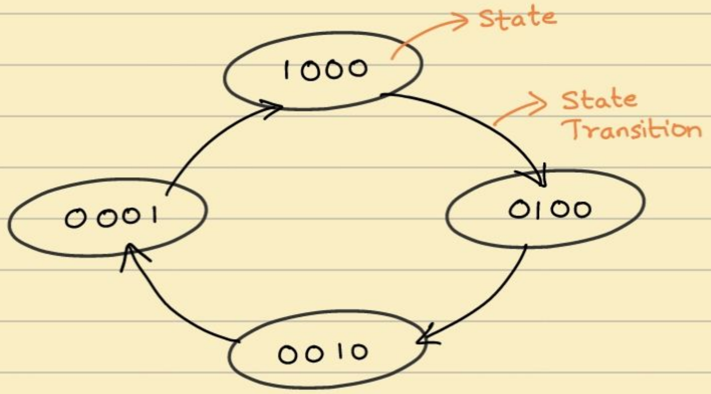


NC : Not connect

Connect ① Q1 to Clock A
② QA to Clock B

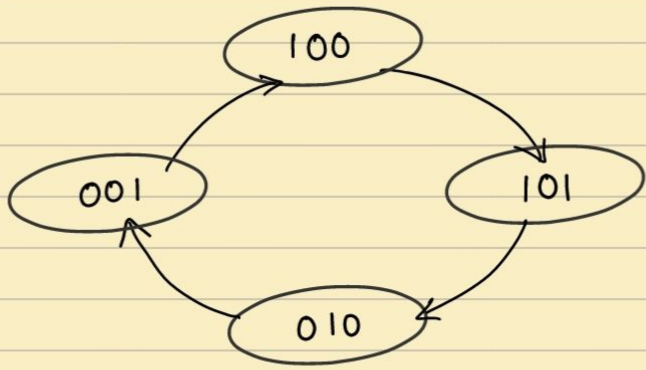
Output : Q1 Q2 QA QB QC QD

29/10



Ring Counter

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

$\boxed{D = q_{t+1}}$

TFF :

q_t	q_{t+1}	T
0	0	0
0	1	1
1	0	1
1	1	0

$\boxed{T = q_t \oplus q_{t+1}}$

SR :

q_t	q_{t+1}	S	R
0	0	0	X
0	1	1	0
1	0	0	1
1	1	X	0

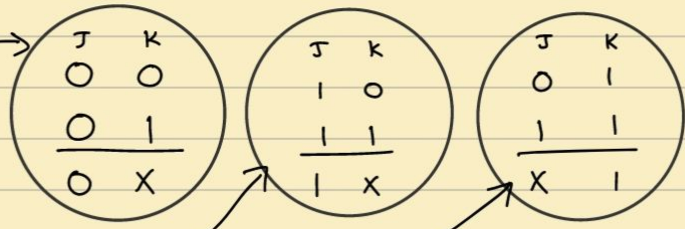
S	R
0	0
0	1
0	X

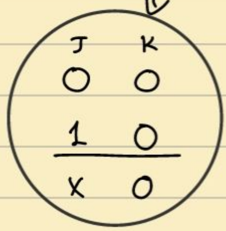
S	R
0	0
1	0
X	0

$$\boxed{\begin{aligned} S &= q_{t+1} \\ R &= \overline{q_{t+1}} \end{aligned}}$$

JK :

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

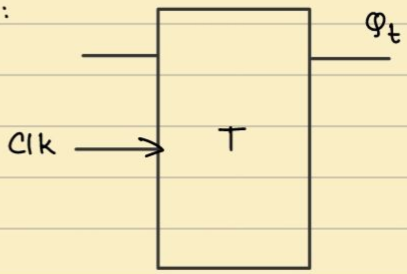




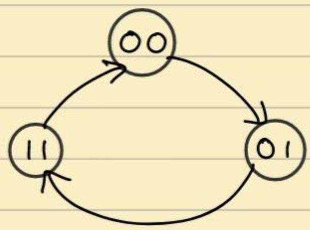
$$\boxed{\begin{aligned} J &= Q_{t+1} \\ K &= \overline{Q_{t+1}} \end{aligned}}$$

Q)

<u>Sol</u>:



Q)



Current		Next			
Q_A	Q_B	Q_{A+1}	Q_{B+1}	T_A	T_B
0	0	0	1	0	1
0	1	1	1	1	0
1	1	0	0	1	1

$T_A = Q_B$

$T_B = Q_A \odot Q_B$

T_A

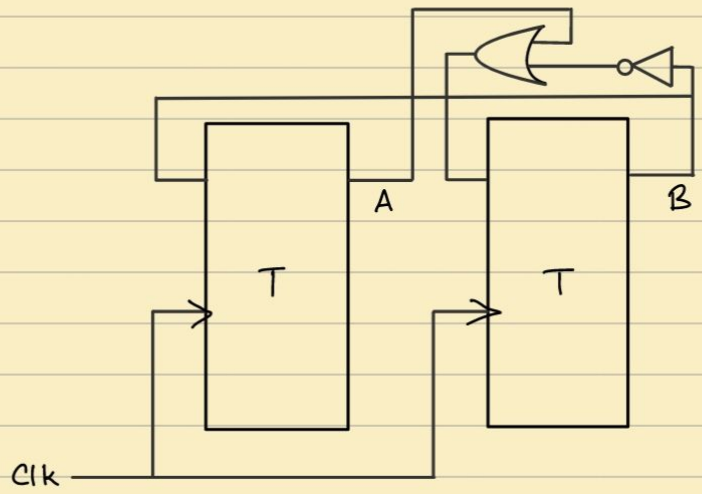
$Q_A \backslash Q_B$	0	1
0	0 (0)	1 (1)
1	X (2)	1 (3)

T_B

$Q_A \backslash Q_B$	0	1
0	1 (0)	0 (1)
1	X (2)	1 (3)

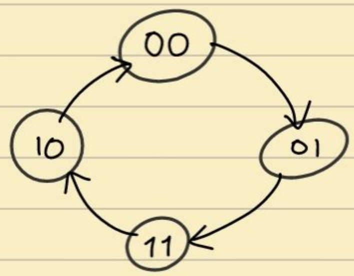
$T_B = \overline{Q_B} + Q_A$

Circuit Diagram:

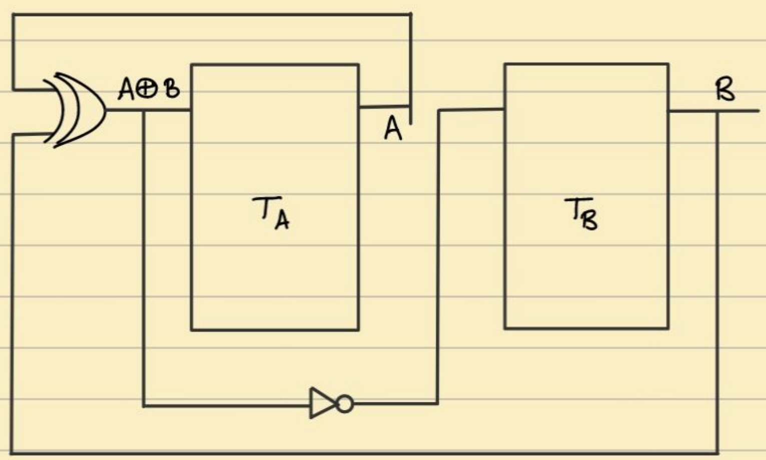


6/11

Q)



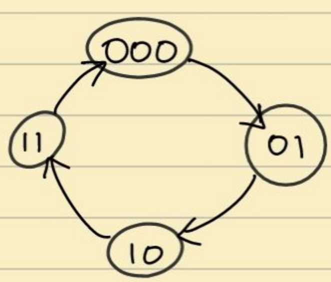
Sol:



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

$$\boxed{\begin{aligned} T_A &= A \oplus B \\ T_B &= A \odot B \end{aligned}}$$

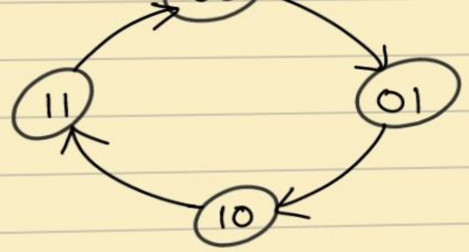
Q)



Not Valid

But 3 flip-flops used.

∵ 3 max bits exist



Sol :

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

J_A

A \ B	0	1
0	0 ₀	1 ₁
1	X ₂	X ₃

K_A

A \ B	0	1
0	X ₀	X ₁
1	0 ₂	1 ₃

: 0

: 1

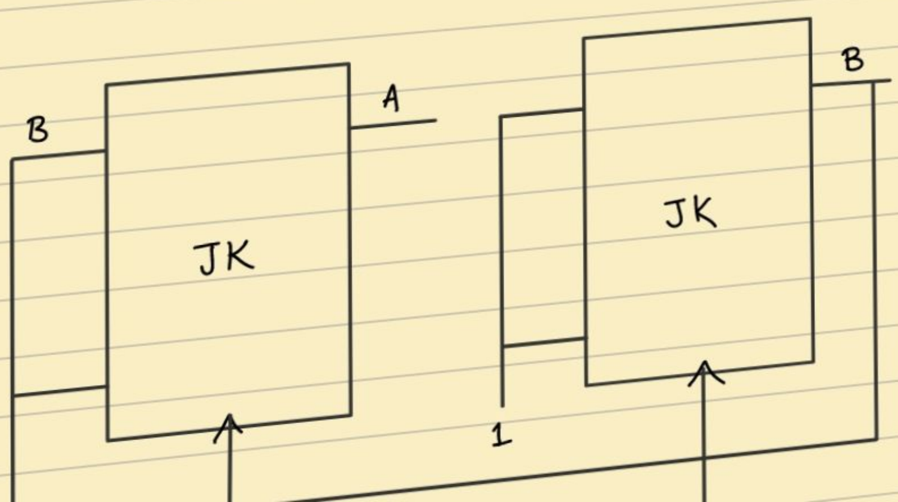
J_B

A \ B	0	1
0	1 ₀	X ₁
1	1 ₂	X ₃

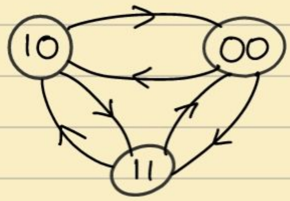
K_B

A \ B	0	1
0	X ₀	1 ₁
1	X ₂	1 ₃

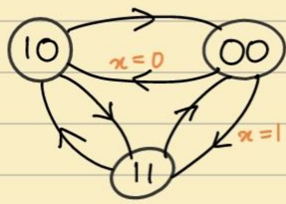
$J_A = B$
$K_A = B$
$J_B = 1$
$K_B = 1$



Note:

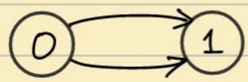


Cannot know which transition to go from.
Hence, an actual input 'x' is taken.



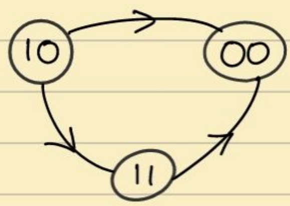
$x = 0, 1$ (i/p)

Note:



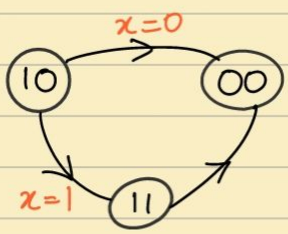
Doesn't make sense

But,



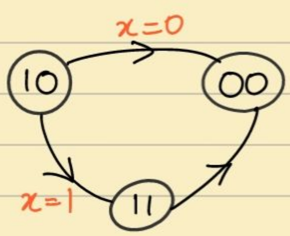
Not deterministic, i.e. no one specific path exists.
Hence, we add input variable

To differentiate multiple possibilities. (More than 1 transition)



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

Q)



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

Sol:

x is not required ∵ It only just decides the transition

Prev			Next			
x	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

This is an example of Finite Set Machine (FSM)

T_A

$x \backslash AB$	00	01	11	10
0	X_0	0_1	X_3	1_2
1	X_4	0_5	X_7	1_6

T_B

$x \backslash AB$	00	01	11	10
0	X_0	1_1	X_3	0_2
1	X_4	1_5	X_7	0_6

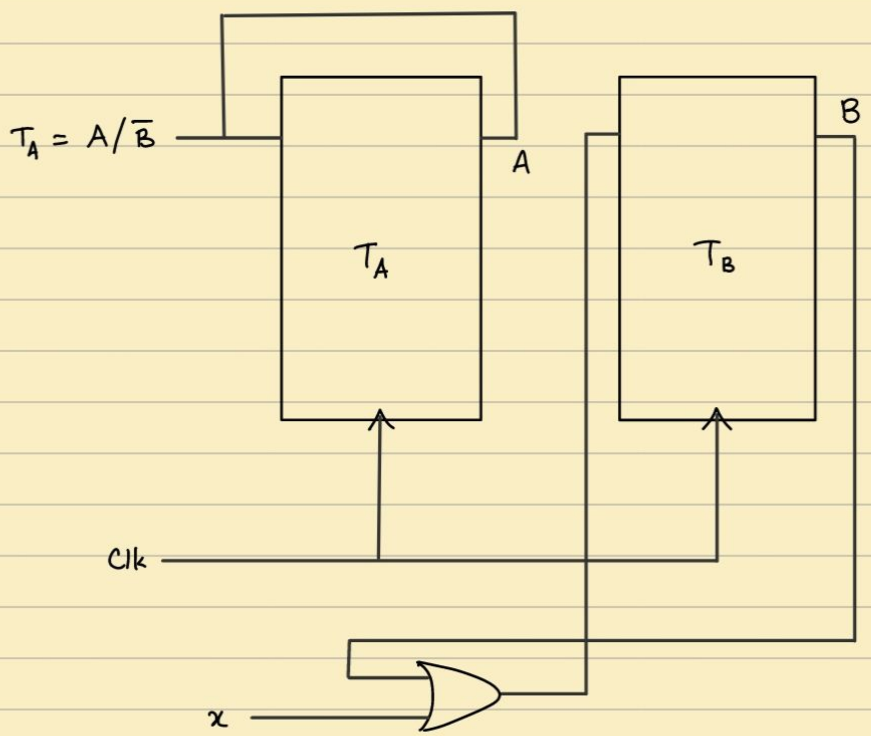
Selective Prime Implicants

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

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

$$\boxed{\begin{aligned} T_A &= A \text{ (or) } \overline{B} \\ T_B &= x + B \text{ (or) } x + \overline{A} \end{aligned}}$$

Circuit Diagram:



Cameo:

Lab - 4

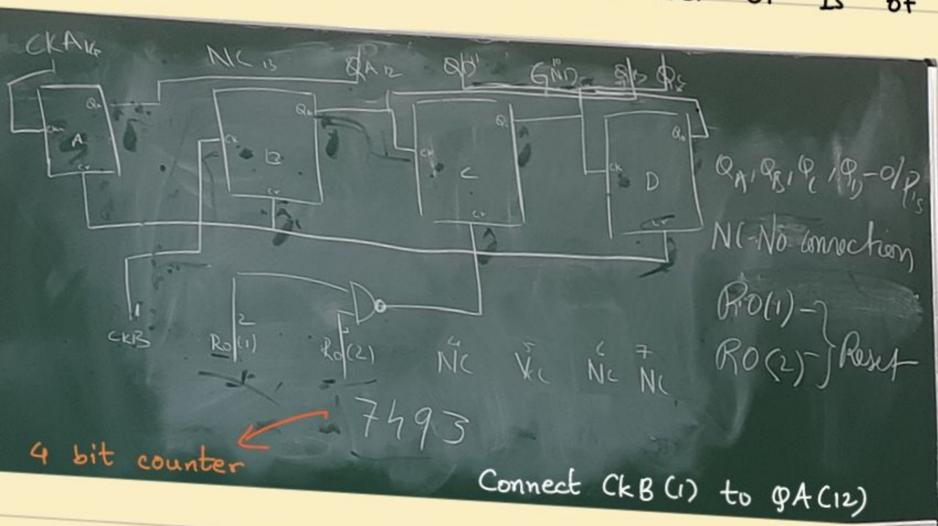
Hardware

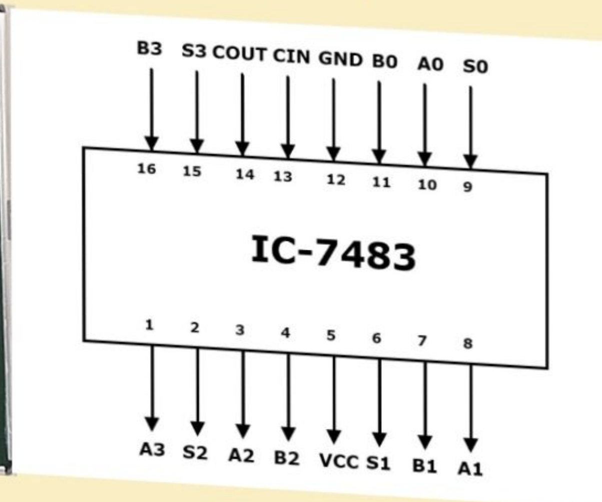
Q1) Given 4 bits input, Check Paridy :

i.e. If no. of 1s is odd ⟹ Output is 1

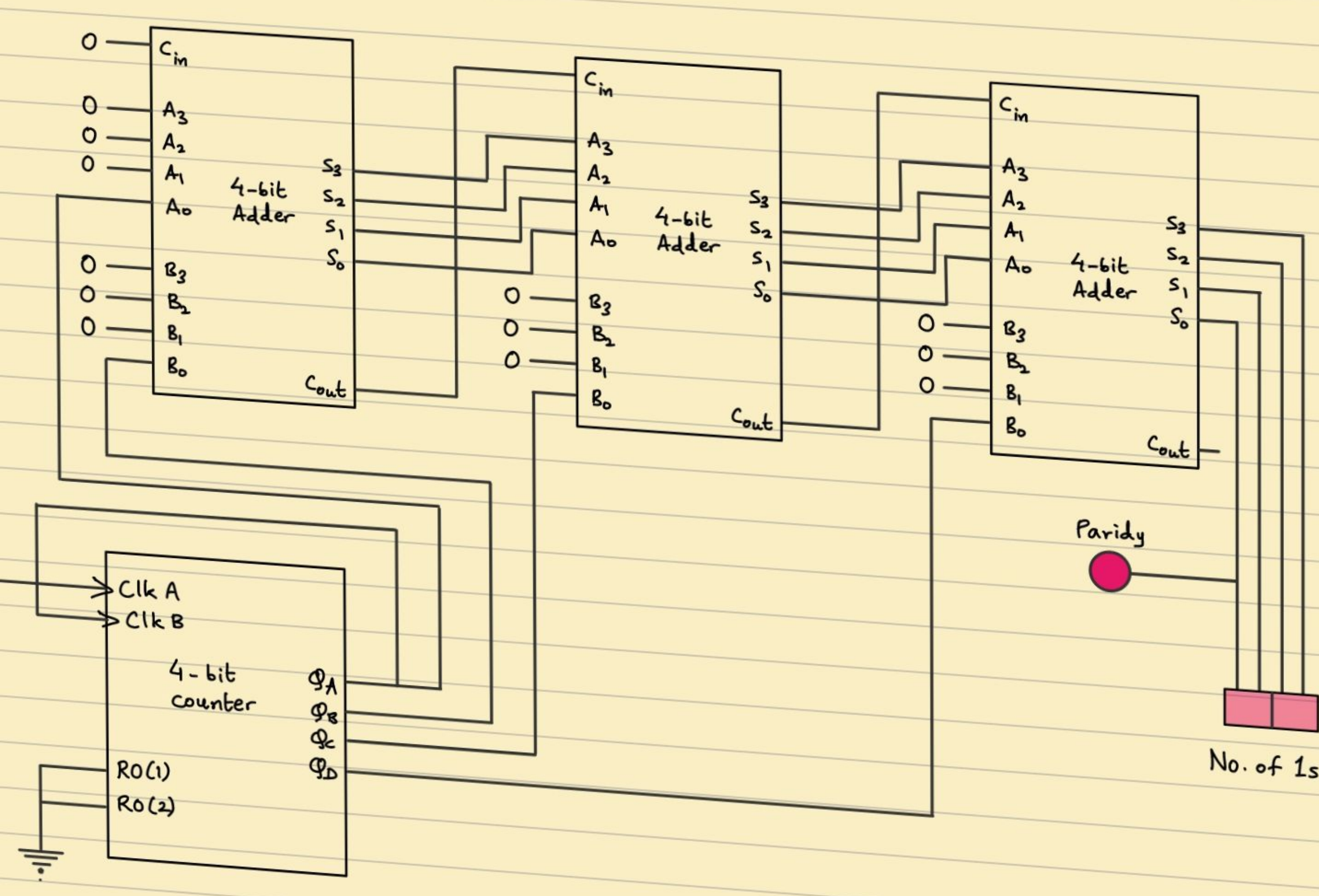
If no. of 1s is even ⟹ Output is 0.

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





Circuit:



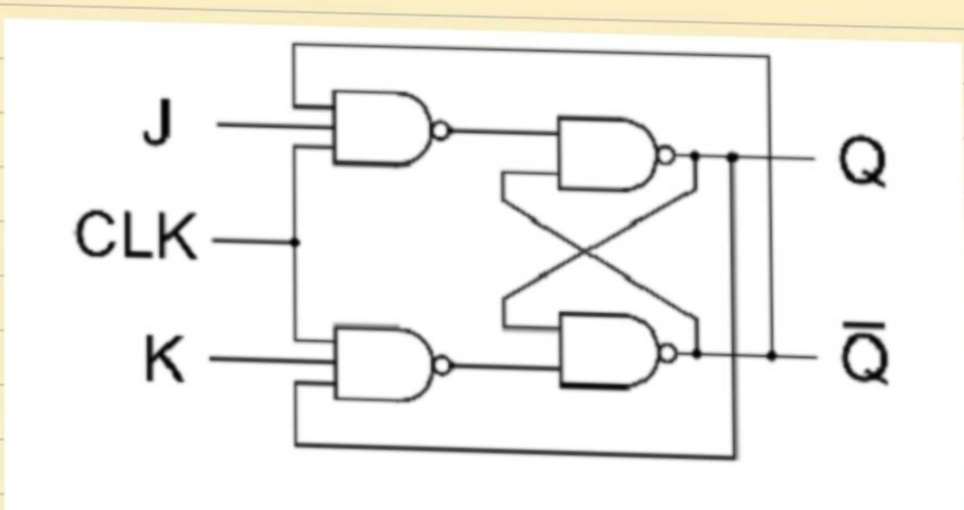
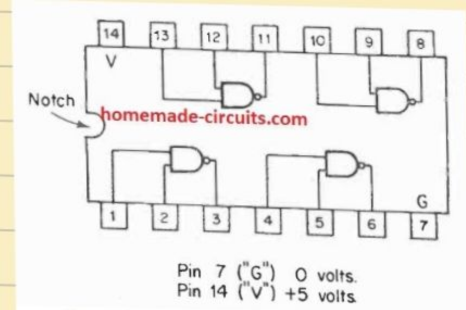
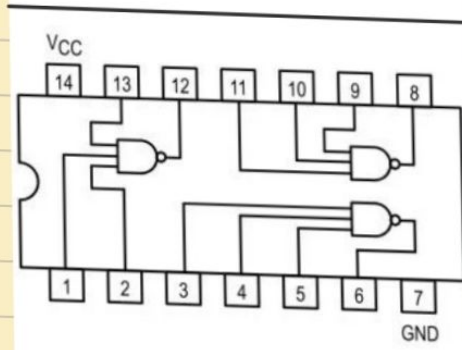


Figure 4: JK Flip Flop



Fugure #1. 7400 IC logic diagram.



7410 IC Logic Diagram

<u>Software</u>

Q) 4 bit Universal Shift Register

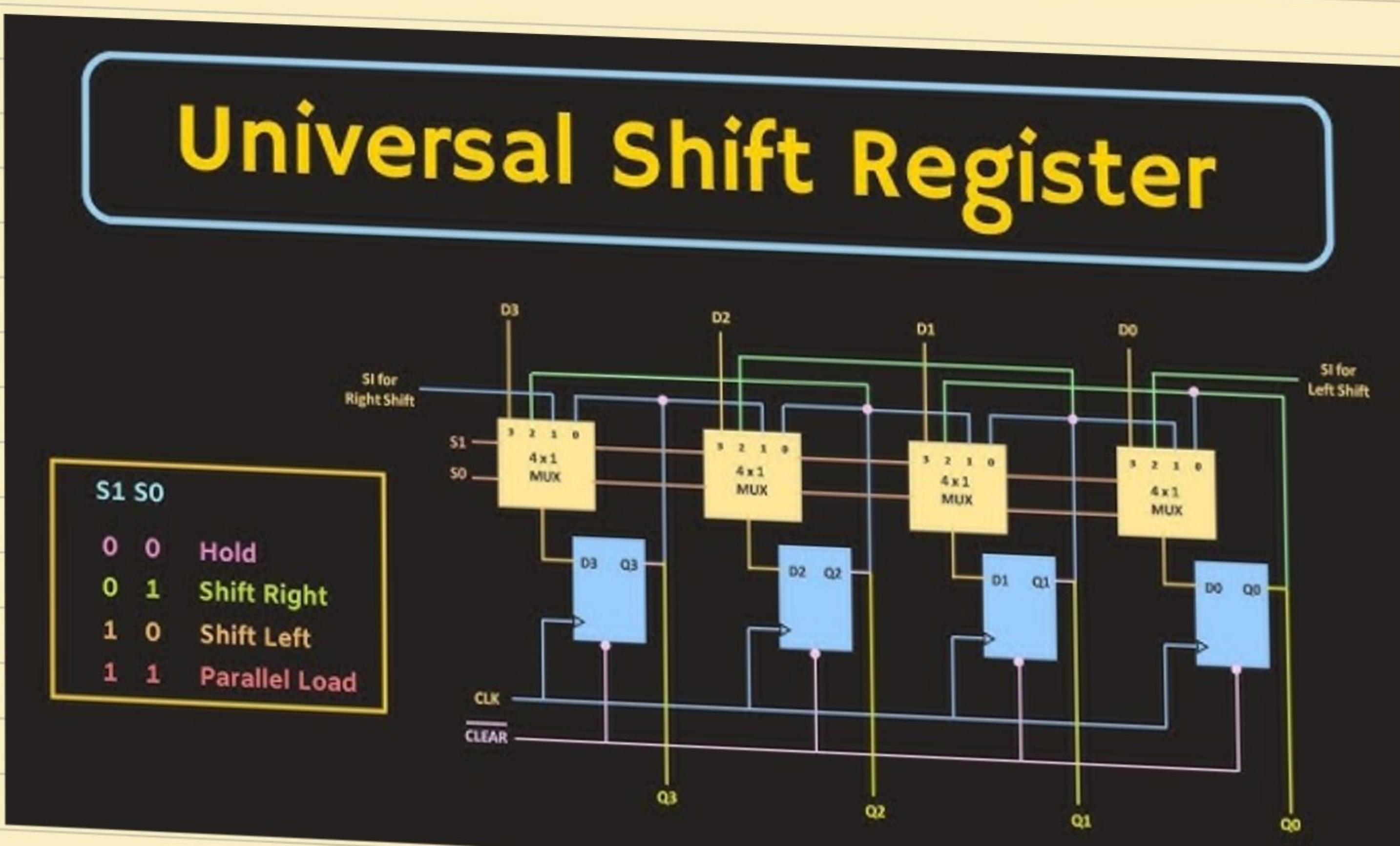
(i) Left Shift

(ii) Right Shift

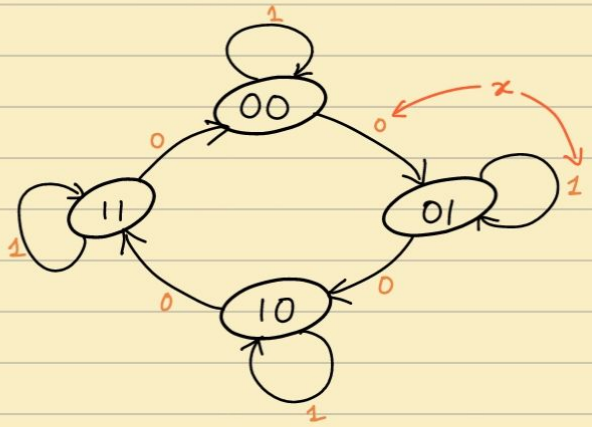
(iii) Parallel Input (Using MUX)

(iv) Retain Value

<u>Circuit</u>:



8/11



	Prev		Next			
x	A	B	A	B	T_A	T_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

T_A

$x \backslash AB$	00	01	11	10
0	0_0	1_1	1_3	0_2
1	0_4	0_5	0_7	0_6

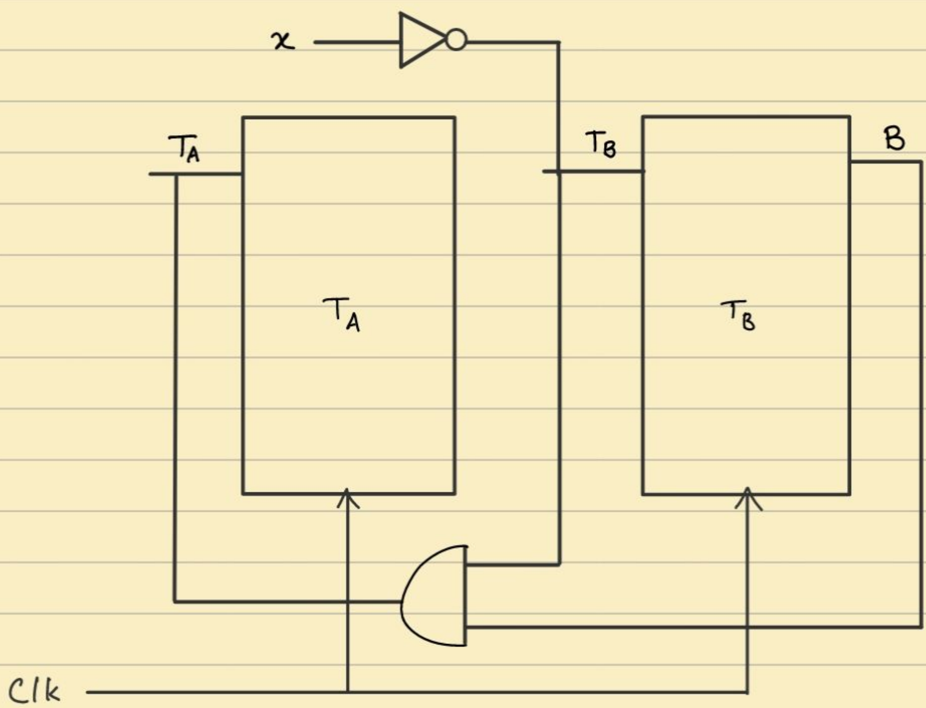
$$\boxed{T_A = \bar{x}B}$$

T_B

$x \backslash AB$	00	01	11	10
0	1_0	1_1	1_3	1_2
1	0_4	0_5	0_7	0_6

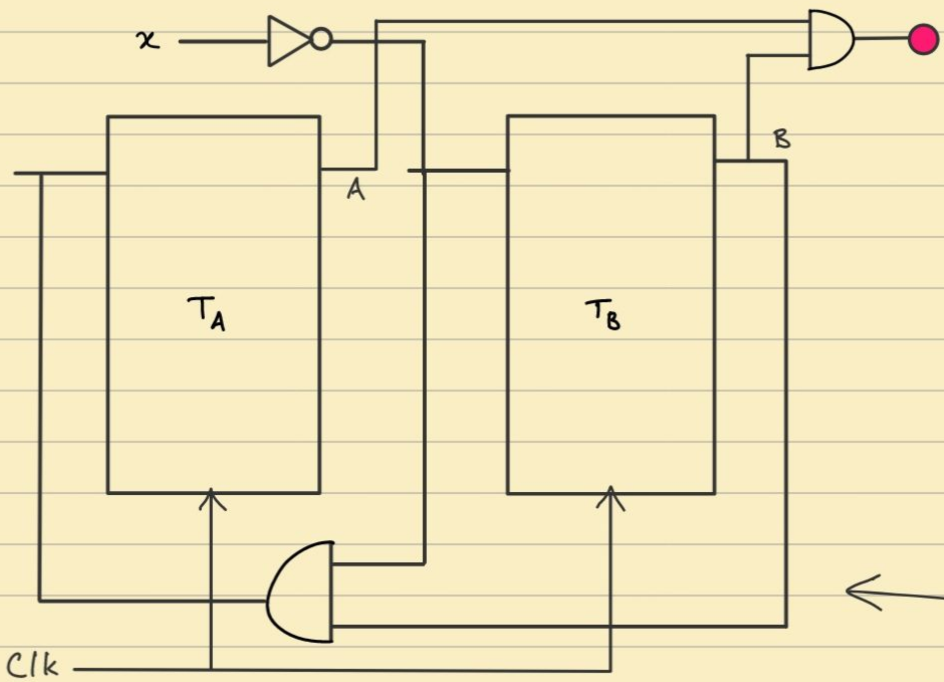
$$\boxed{T_B = \bar{x}}$$

- 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.

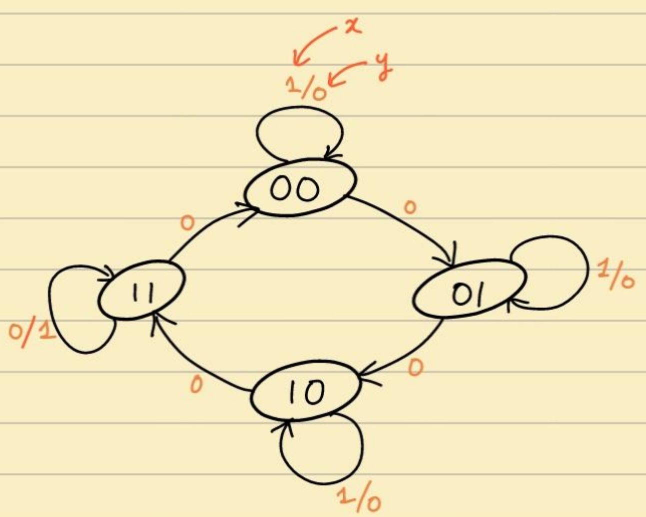
Ex. if a stopwatch is stopped at a particular time, the counter will not stop.

Hence, Mealy is used.

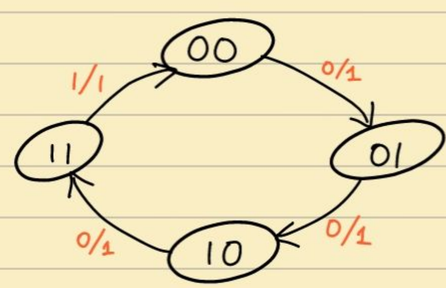
Based on present state, we'll have output : Moore Machine

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

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

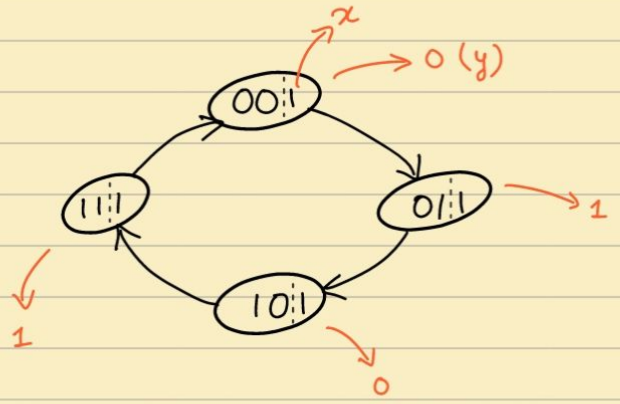


Moore Machine



Mealy Machine

9/11



* Alan Turing

Moore Machine

H.W : Moore's Law

* Von Neumon - Computer Architechture

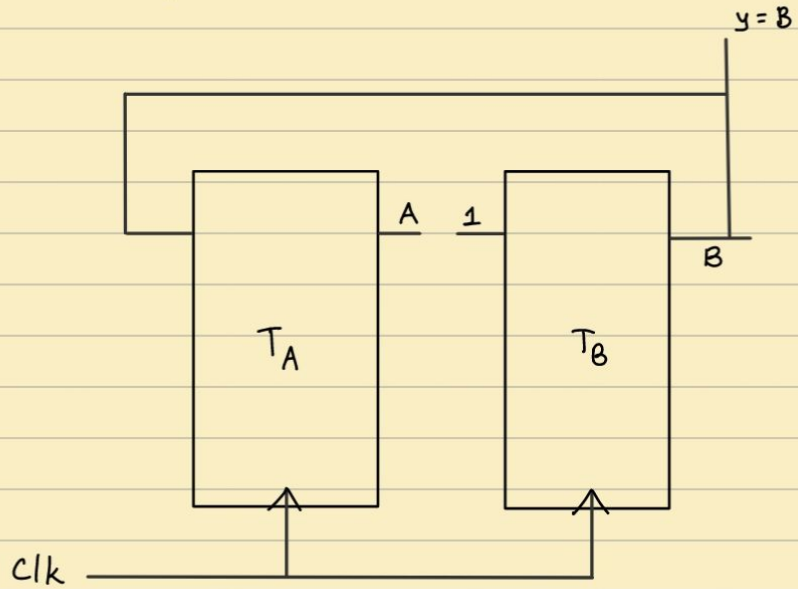
Prev		Next				
A	B	A	B	T_A	T_B	y
0	0	0	1	0	1	0
0	1	1	0	1	1	1
1	0	1	1	0	1	0
1	1	0	0	1	1	1

T_A

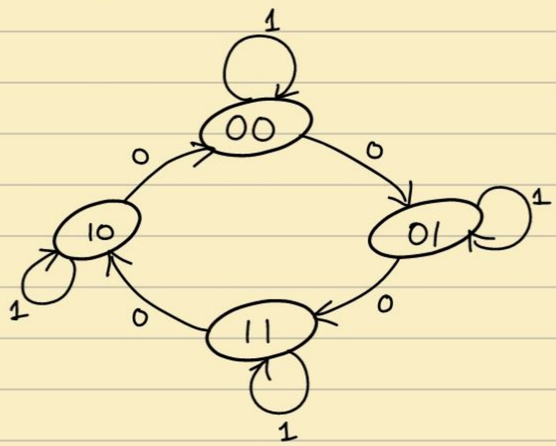
A \ B	0	1
0	0 (0)	1 (1)
1	0 (2)	1 (3)

$\boxed{T_A = B}$

- Circuit Diagram :



- Mealy Machine :

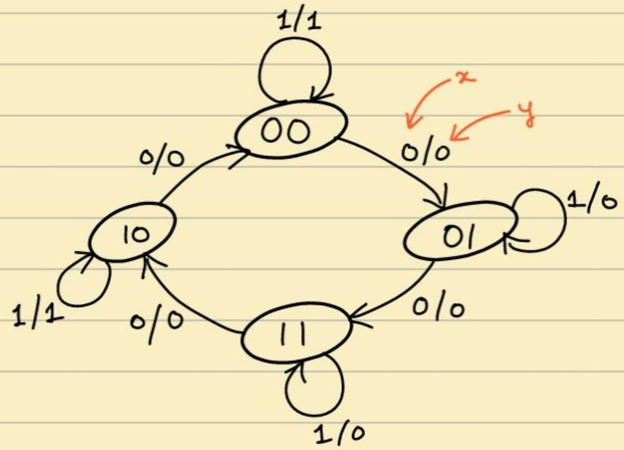


Even + Stay = 1

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

T_A

$x \backslash AB$	00	01	11	10
0	0_{0}	1_{1}	0_{3}	1_{2}
1	0_{4}	0_{5}	0_{7}	0_{6}

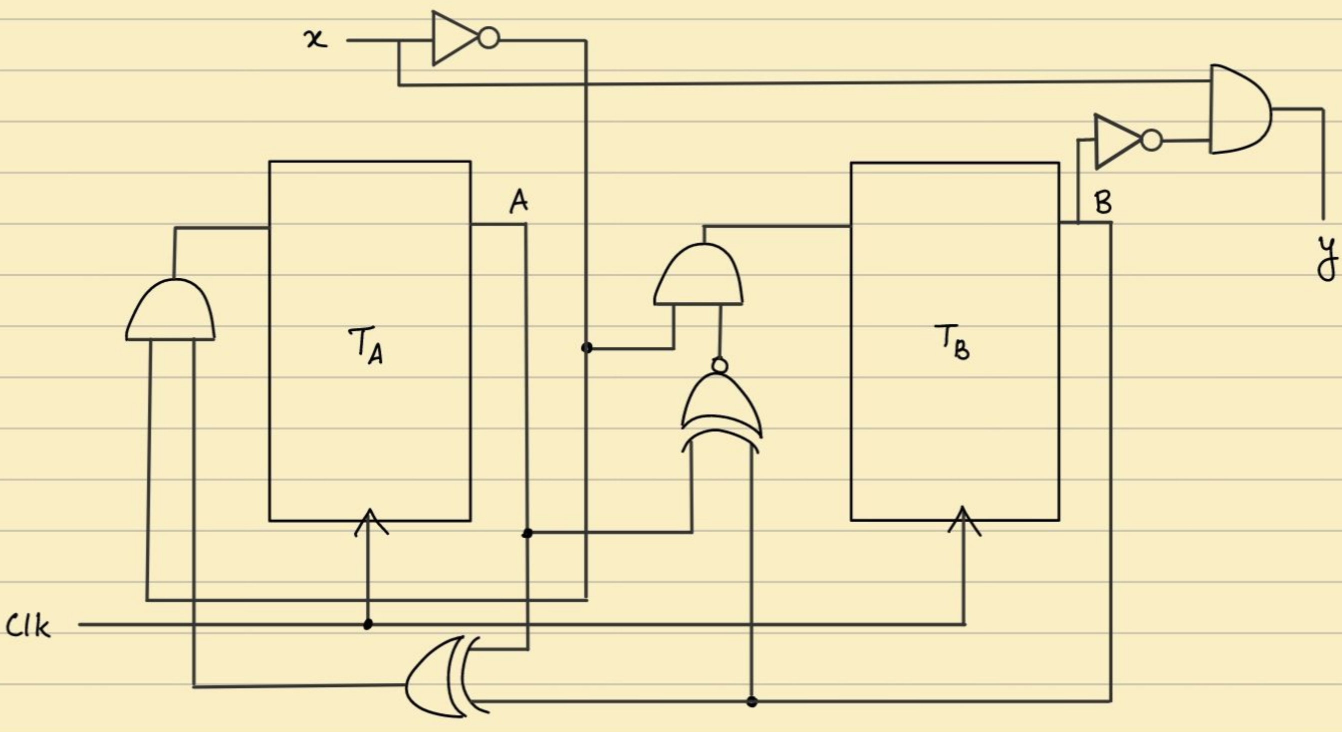
T_B

$x \backslash AB$	00	01	11	10
0	1_{0}	0_{1}	1_{3}	0_{2}
1	0_{4}	0_{5}	0_{7}	0_{6}

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

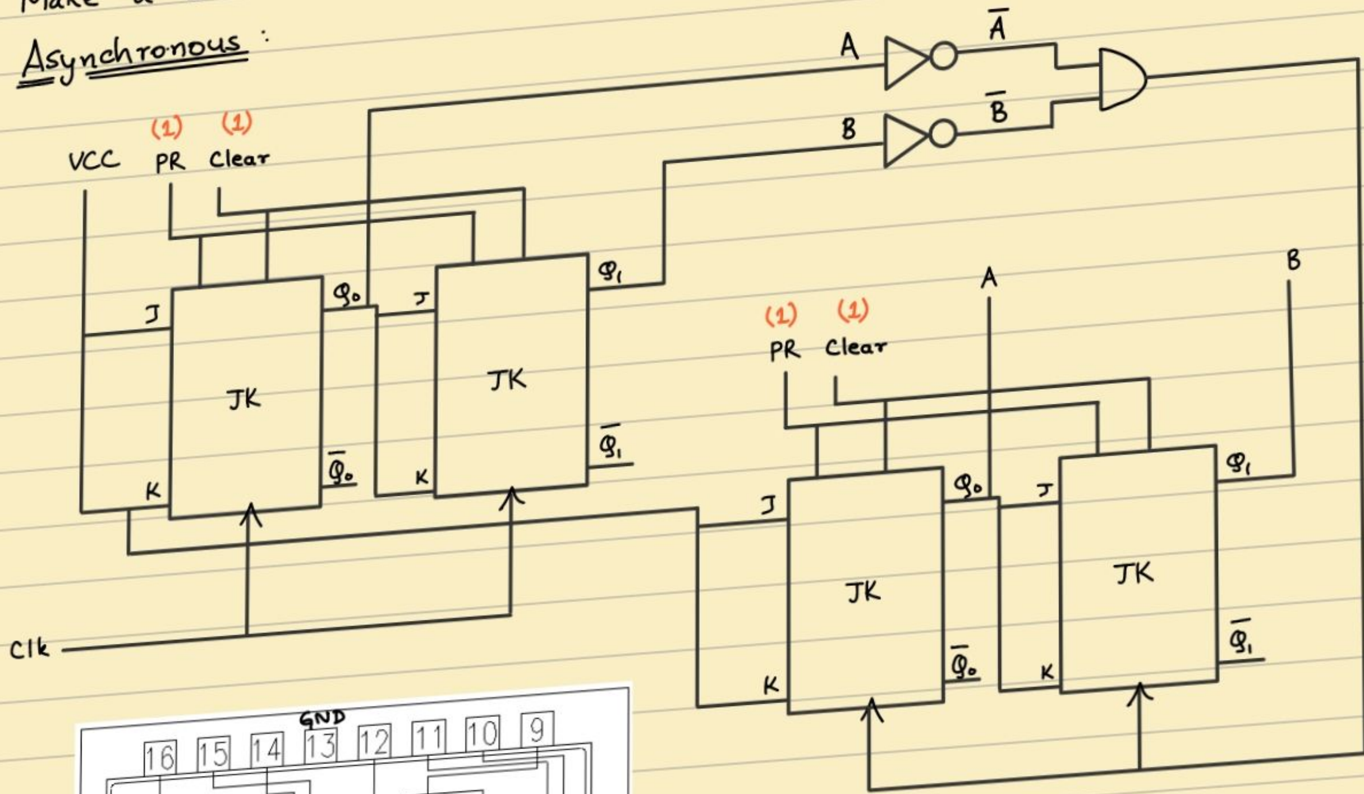
$$y = x\bar{B}$$

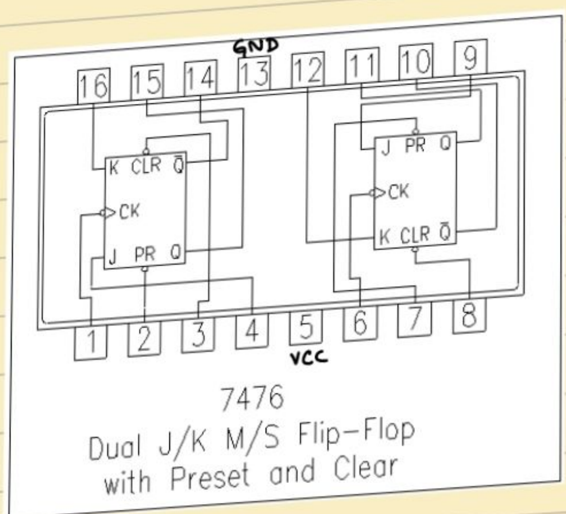
- <u>Circuit Diagram</u>:



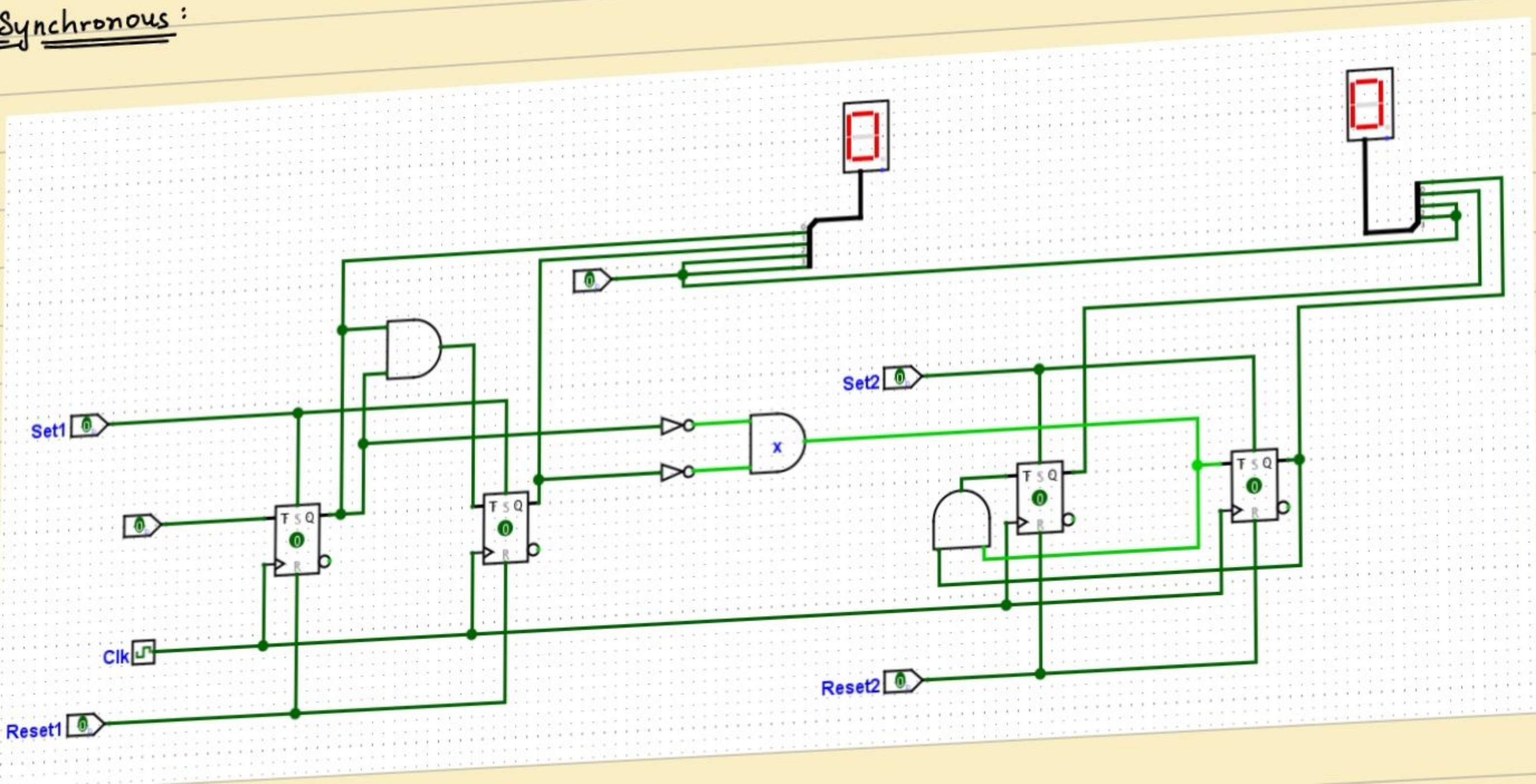
) Make a 2 bit counter, count the no. of

Asynchronous :





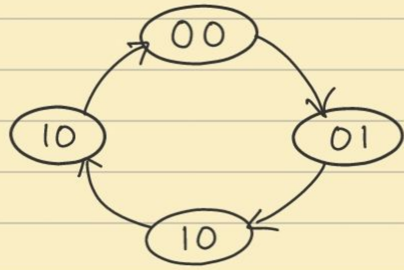
Synchronous :



Explanation:

(1)

Prev		Next			
A	B	A	B	T_A	T_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

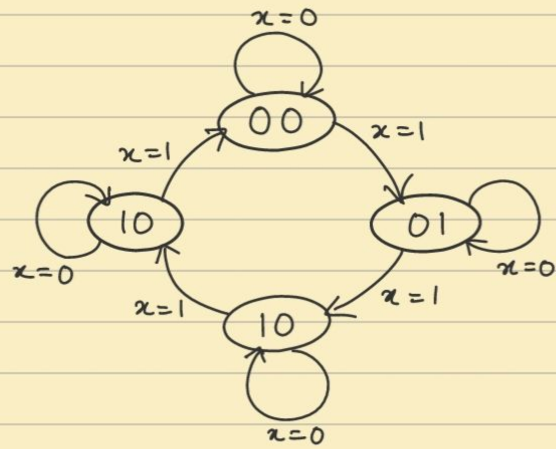


A \ B	0	1
0	0_{0}	1_{1}
1	0_{2}	1_{3}

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

(2)

	Prev		Next			
x	A	B	A	B	T_A	T_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

x \ AB	00	01	11	10
0	0_{0}	0_{1}	0_{3}	0_{2}
1	0_{4}	1_{5}	1_{7}	0_{6}

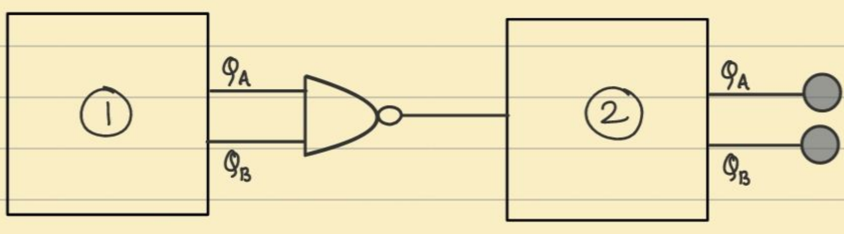
$\boxed{T_A = xB}$

T_B

x \ AB	00	01	11	10
0	0_{0}	X_{1}	X_{3}	X_{2}
1	1_{4}	1_{5}	1_{7}	1_{6}

$\boxed{T_B = x}$

Circuit Diagram:



11/11

Assignment - I

CS23I1027

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

a

B_0B_1 \ B_2B_3	00	01	11	10
00	1_{0}	1_{1}	1_{3}	1_{2}
01	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	0_{0}	0_{1}	1_{3}	1_{2}
01	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	0_{0}	1_{1}	1_{3}	0_{2}
01	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	1_{0}	1_{1}	1_{3}	1_{2}
01	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	1_{0}	0_{1}	0_{3}	1_{2}
01	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	1_{0}	1_{1}	0_{3}	0_{2}
01	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	0_{0}	1_{1}	1_{3}	1_{2}
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}

$a = \overline{B_1} + B_2$

$b = B_0 + B_1 + B_2$

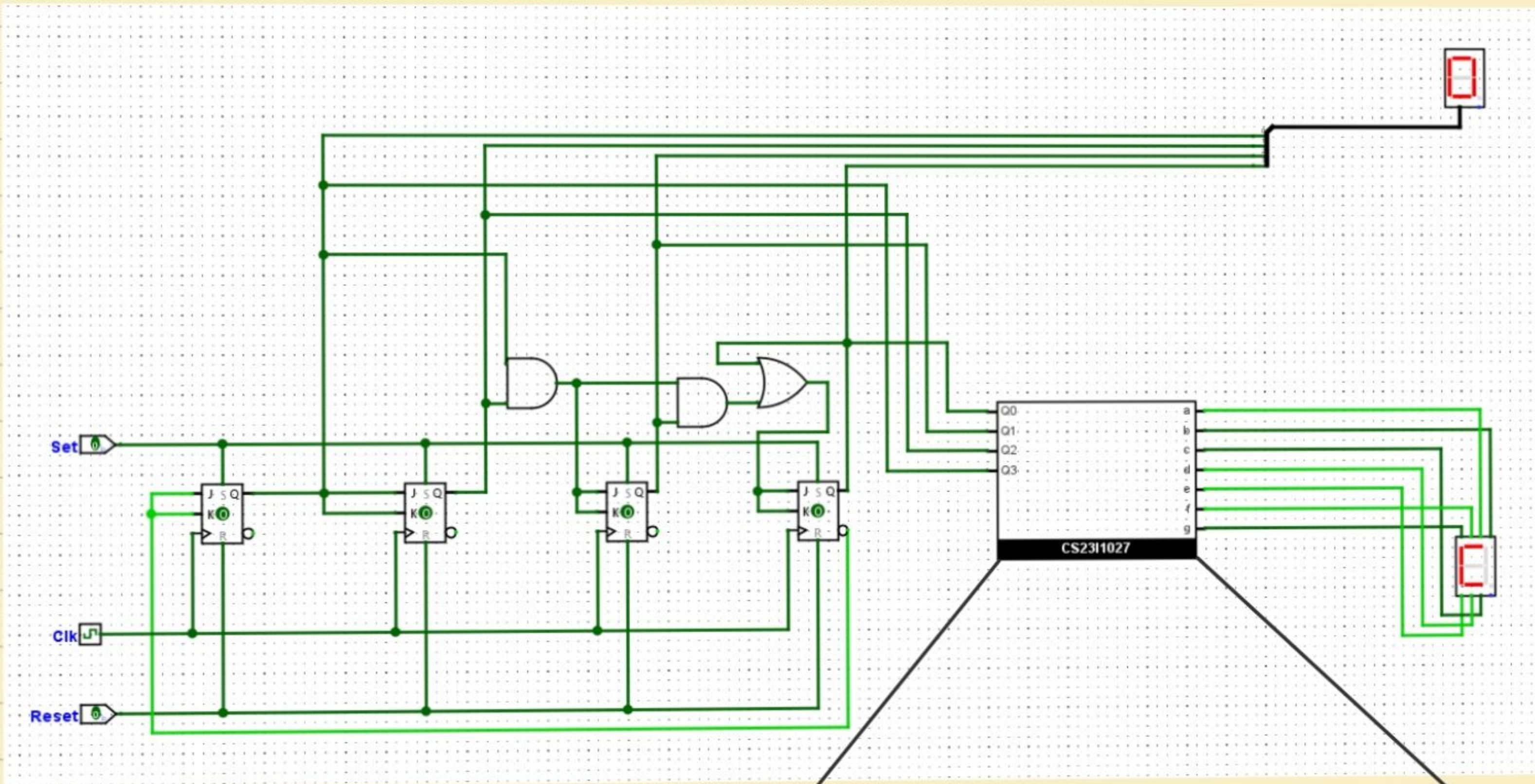
$c = B_0 + B_1\overline{B_2} + B_1 \oplus B_3$

$d = \overline{B_0}\,\overline{B_1} + B_2$

$e = B_1B_2 + \overline{B_0}\,\overline{B_1}\,\overline{B_3}$

$f = \overline{B_0}\,\overline{B_1}\,\overline{B_2} + B_1B_2\overline{B_3}$

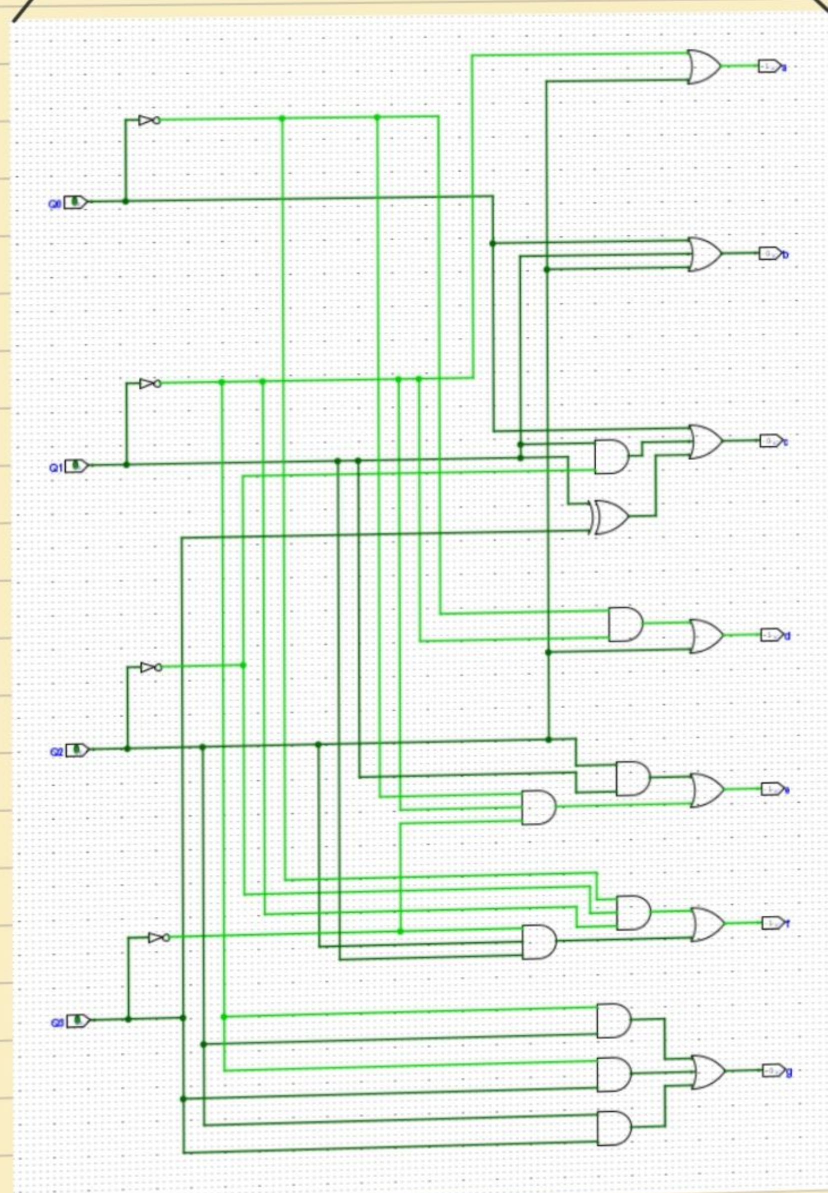
$g = \overline{B_1}B_2 + \overline{B_1}B_3 + B_2B_3$



C S 2 3 I 1 0 2 7
0 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							
A	B	C	D	A	B	C	D	T_A	T_B	T_C	T_D
0	0	0	0	0	0	0	1	0	0	0	1
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	1	0	0	0	1	1	1	1
1	0	0	0	0	0	0	0	1	0	0	1

$\boxed{T_D = 1}$

T_A

AB \ CD	00	01	11	10
00	0_{0}	0_{1}	0_{3}	0_{2}
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}

$\boxed{T_A = A + BCD}$

T_B

AB \ CD	00	01	11	10
00	0_{0}	0_{1}	1_{3}	0_{2}
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}

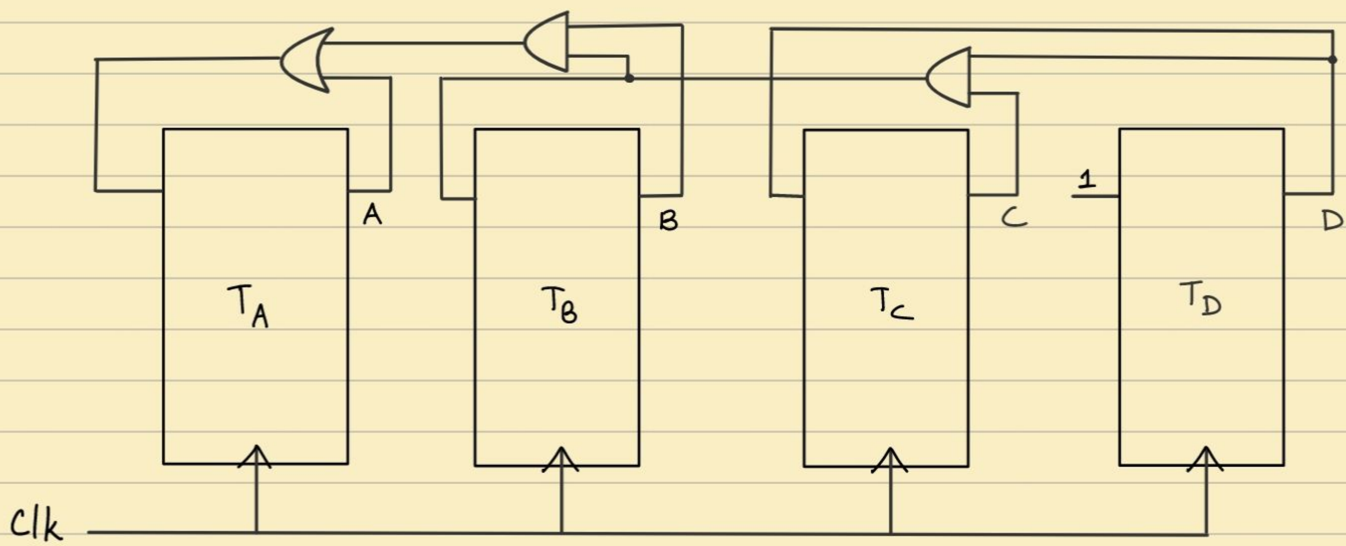
$\boxed{T_B = CD}$

T_C

AB \ CD	00	01	11	10
00	0_{0}	1_{1}	1_{3}	0_{2}
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}

$\boxed{T_C = D}$

Circuit :



11/11

Codes

- Weighted — Ex. BCD
- Non-weighted — Ex. Gray

$$\begin{array}{cccc} 2^3 & 2^2 & 2^1 & 2^0 \\ \uparrow & \uparrow & \uparrow & \uparrow \\ 1 & 0 & 0 & 0 \end{array}$$

0×2^0
0×2^1
0×2^2
1×2^3

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

0011 to Binary : 2
(No weights)

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 :

+3 +3
1 8

0001 1000

Ex. Addition

$$\begin{array}{r} 43 \\ 82 \\ \hline ? \end{array}$$

$Ans_{BCD} = 125$
$Ans_{E3} = 158$

BCD

$$\begin{array}{r} 0100 \\ +\ 1000 \\ \hline 1100 \\ (+6)\ 0110 \\ \hline 1\,0010 \end{array} \qquad \begin{array}{r} 0011 \\ +\ 0010 \\ \hline 0101 \\ 0110\ (+6) \\ \hline 0\,1011 \end{array}$$

$\Downarrow +3$ $\qquad$ $\Downarrow -3$

E3 0101 $\qquad$ E3 1000 ⇒ 1 2 5 (BCD)

BCD
(5) − 3 = 2 $\qquad$ (8) − 3 = 5

- Addition :

$$\begin{array}{r} 18 \\ +\ 13 \\ \hline \end{array}$$

Ans (BCD) : 31
Ans (E3) : 64

$$\begin{array}{r} 0001 \\ 0001 \\ \hline 0011 \end{array} \qquad \begin{array}{r} 1000 \\ 0011 \\ \hline 1011 \end{array}$$

⇓ BCD → Excess 3 $\qquad$ ⇓ BCD → Excess 3

$$\begin{array}{r} 0011 \\ 0110\ (+6) \\ \hline 0\,1001 \\ +\ 1101 \\ \hline 1\,0110 \end{array} \qquad \begin{array}{r} 1011 \\ 0110\ (+6) \\ \hline 1\,0001 \\ +\ 0011 \\ \hline 0100 \end{array}$$

∵ Carry → +3

0110 → 6 → (E3)
0100 ← 4 → (E3)

BCD : 6 − 3 = 3
BCD : 4 − 3 = 1
∴ 31

[1101 : 2's complement of 3]

Carry = 0 ⇒ Subtract 3	
Carry = 1 ⇒ Add 3	

→ Converter :

I/p → BCD, 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	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

$\boxed{E_0 = \overline{B_0}}$

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

B_3B_2 \ B_1B_0	00	01	11	10
00	0_{0}	1_{1}	1_{3}	1_{2}
01	1_{4}	0_{5}	0_{7}	0_{6}
11	X_{12}	X_{13}	X_{15}	X_{14}
10	0_{8}	1_{9}	X_{11}	X_{10}

E_1

B_3B_2 \ B_1B_0	00	01	11	10
00	0_{0}	1_{1}	1_{3}	0_{2}
01	1_{4}	0_{5}	1_{7}	0_{6}
11	X_{12}	X_{13}	X_{15}	X_{14}
10	1_{8}	0_{9}	X_{11}	X_{10}

E_0

B_3B_2 \ B_1B_0	00	01	11	10
00	1_{0}	0_{1}	0_{3}	1_{2}
01	1_{4}	0_{5}	0_{7}	1_{6}
11	X_{12}	X_{13}	X_{15}	X_{14}
10	1_{8}	0_{9}	X_{11}	X_{10}

E_3

B_3B_2 \ B_1B_0	00	01	11	10
00	0_{0}	0_{1}	0_{3}	0_{2}
01	0_{4}	1_{5}	1_{7}	1_{6}
11	X_{12}	X_{13}	X_{15}	X_{14}
10	1_{8}	1_{9}	X_{11}	X_{10}

Note:

Every Algo is Sequential Nature.

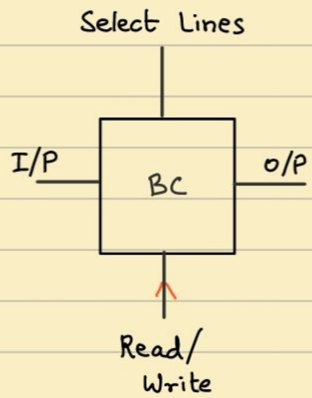
$$E_0 = \overline{B_0}$$
$$E_1 = B_0B_1 + \overline{B_0}B_3 + \overline{B_0}\,\overline{B_1}B_2 + B_0\overline{B_2}\,\overline{B_3}$$
$$E_2 = \overline{B_0}\,\overline{B_1}B_2 + B_0\overline{B_2} + B_1\overline{B_2}$$
$$E_3 = B_3 + B_0B_2 + B_1B_2$$

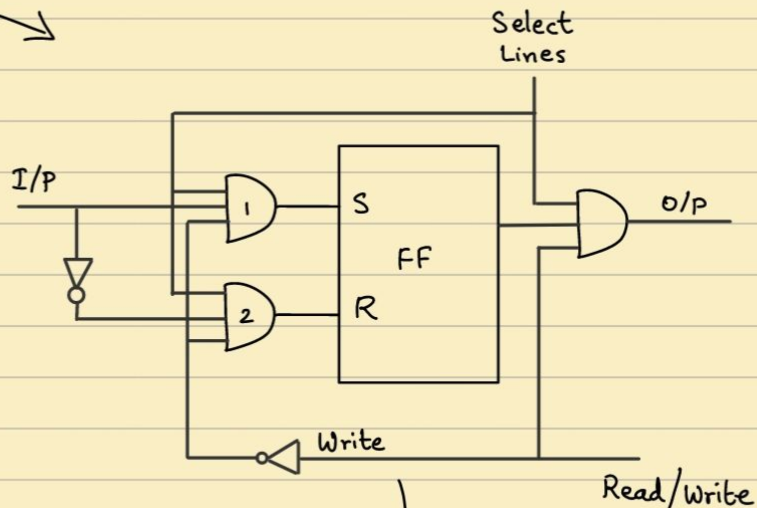
13/11

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

Non - Volatile
- ROM
- HDD (Hard Disk Drive)
  - SSD (Solid State Device)
  - Flash Drives

→ **Basic Cell : / Binary Cell :**



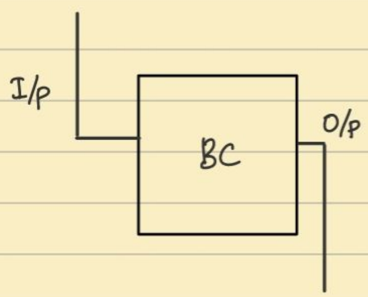


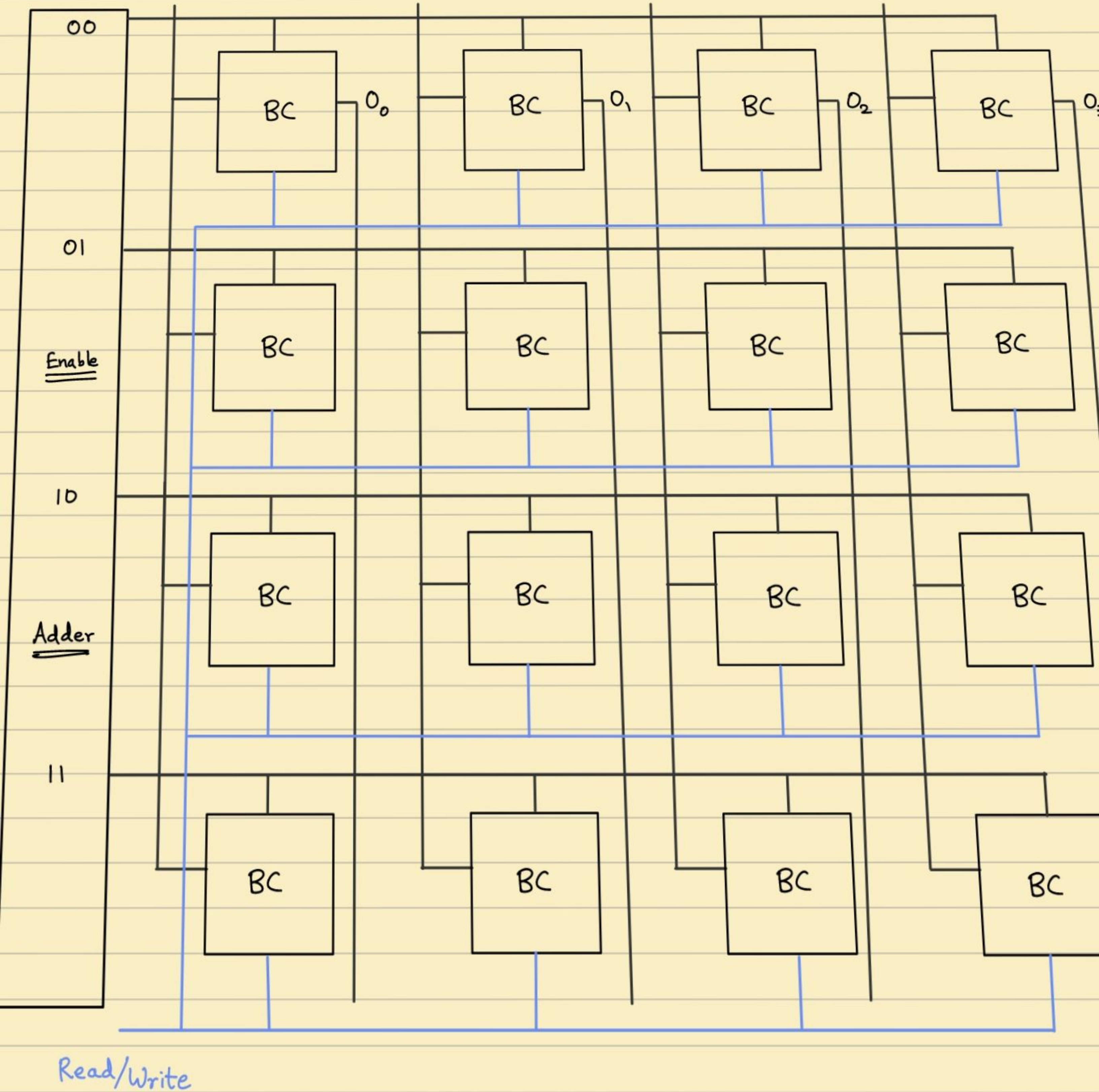
If Write = 1 ⇒ I/P = 1
If Read = 1 ⇒ O/P = 1

Mentioned as a Latch but acts as a flip flop.

0 = Write
1 = Read

Smallest Addressable value of unit - Word
↳ 4 bit




00
01
Enable
10
Adder
11
BC
O_0
O_1
O_2
O_3
Read/Write

Hardware:

Q1) Make a counter in $1 \to 2 \to 3 \to 2 \to 1 \to 2 \ldots$ 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:

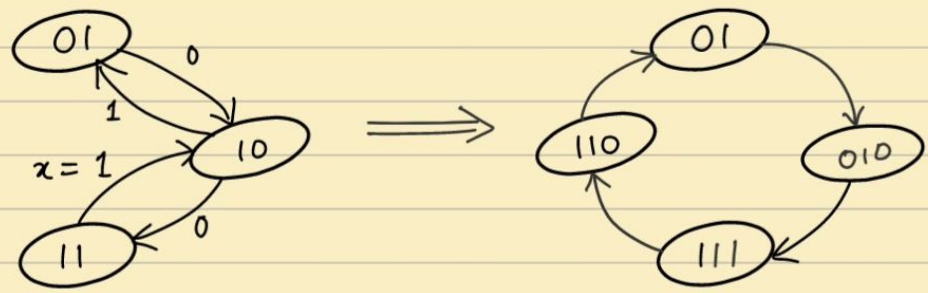
(i) Sequential Logic

$x \longrightarrow$ Manual

To automate x, Take x as another Flip flop.

Q : Indirectly converting Mealy Machine to Moore Machine

i.e.



Current			Next					
A	B	C	A	B	C	T_A	T_B	T_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

T_A

A \ BC	00	01	11	10
0	X_0	0_1	1_3	0_2
1	X_4	X_5	X_7	1_6

$\boxed{T_A = BC + A}$

T_B

A \ BC	00	01	11	10
0	X_0	1_1	0_3	0_2
1	X_4	X_5	X_7	1_6

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

T_C

A \ BC	00	01	11	10
0	X_0	1_1	1_3	1_2
1	X_4	X_5	X_7	1_6

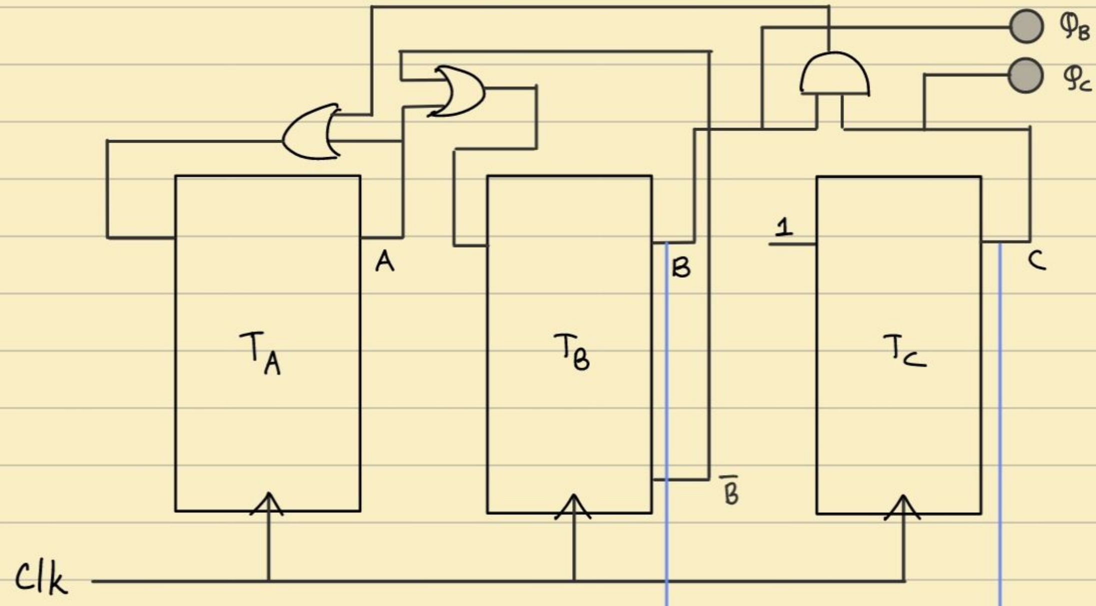
$\boxed{T_C = 1}$

Advantage of Converting : Automation

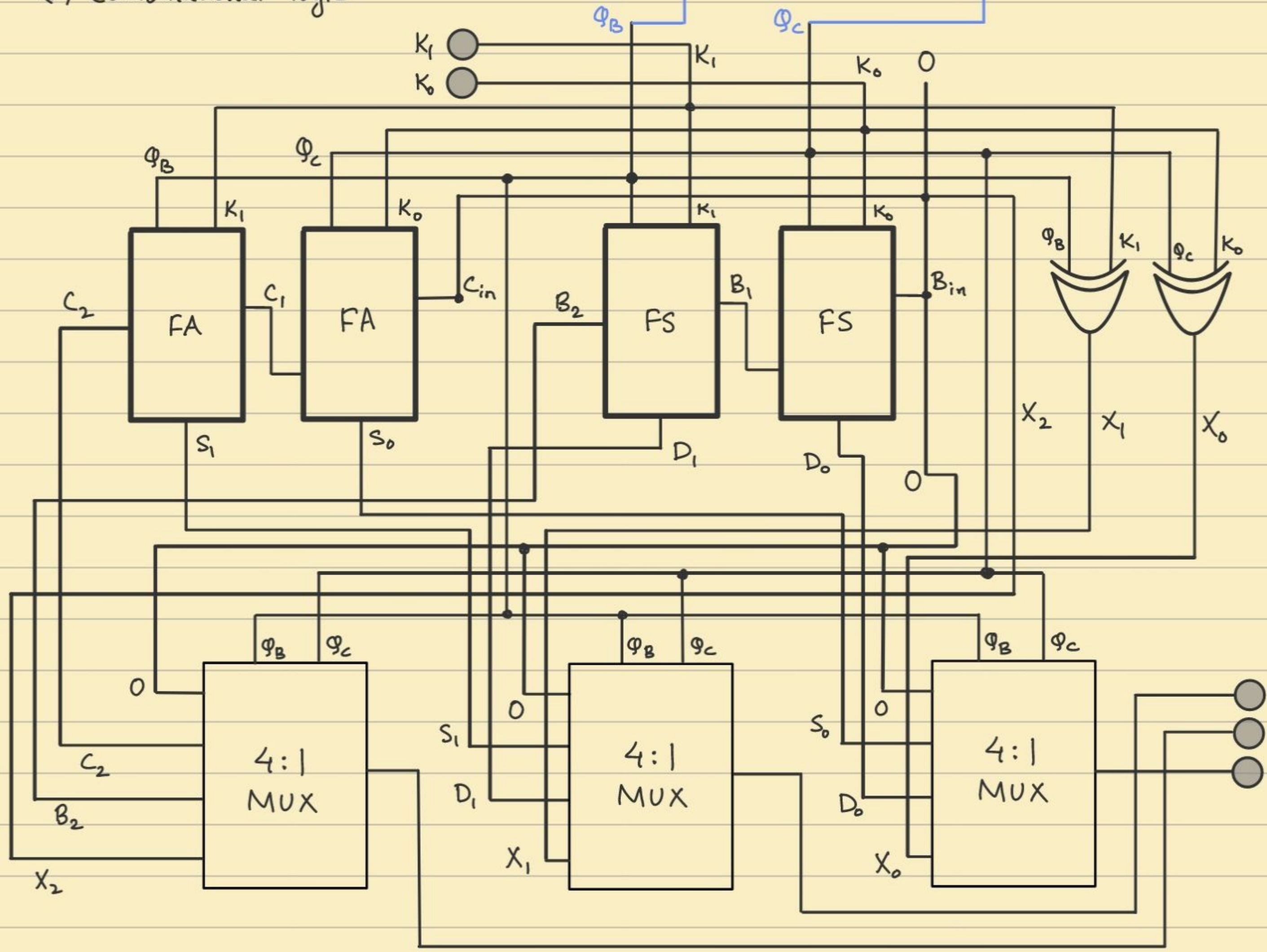
Disadvantage : No. of states increase tremendously & Inputs change

$$T_A = Q_A + Q_B \cdot Q_C$$
$$T_B = Q_A + \overline{Q_B}$$
$$T_C = 1$$

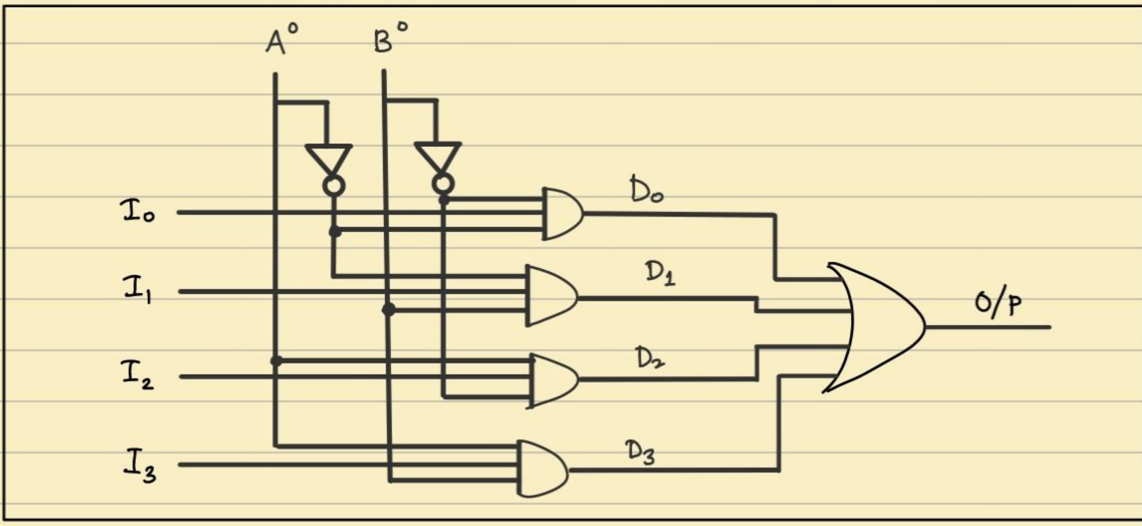
Circuit Diagram for counter:



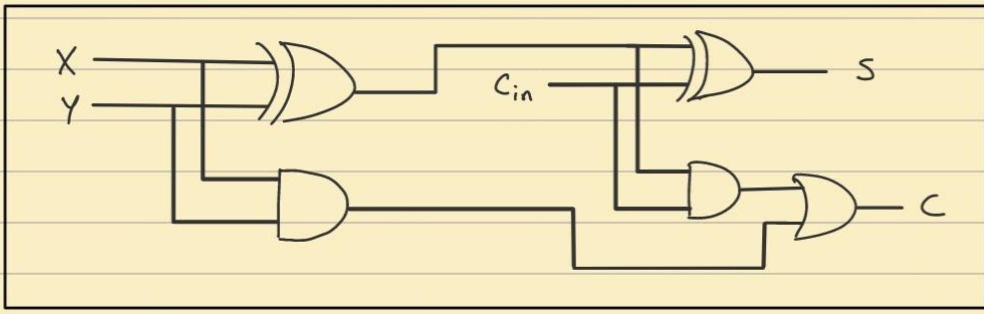
(ii) Combinational logic :



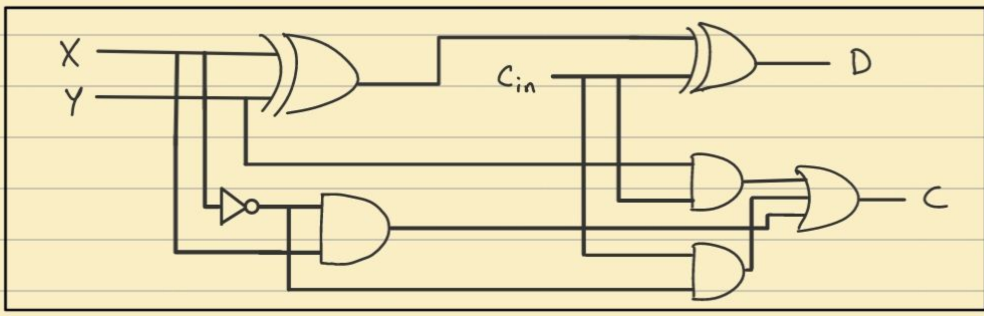
MUX $\begin{cases} 00 - \text{Nothing} & (I_0) \\ 01 - \text{Add} & (I_1) \\ 10 - \text{Subtract} & (I_2) \\ 11 - \text{XOR} & (I_3) \end{cases}$



4:1 MUX module

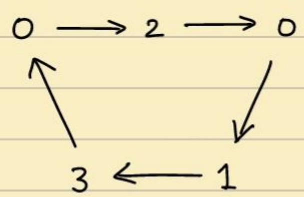


FA module



FS module

Q2)



K : User Input – $K_1 K_0$

Low to High : Add K

High to Low : Subtract K

Software

Q1) 4 Circuits : Down Counter - 4 bit
Ring Counter - 4 bit
Retain Value - 4 bit
Excess 3 converter, whose input is taken from the current output (consider BCD Input) - 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.

Q1) 4 Circuits : Up Counter - 4 bit
Johnson Counter - 4 bit
Retain Value - 4 bit
Excess 3 converter, whose input is taken from the current output (consider BCD Input) - 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.

20/11

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 ⟶ 4 bytes , i.e. $\omega = 4$
$\therefore$ 32 bits

$k = 10$

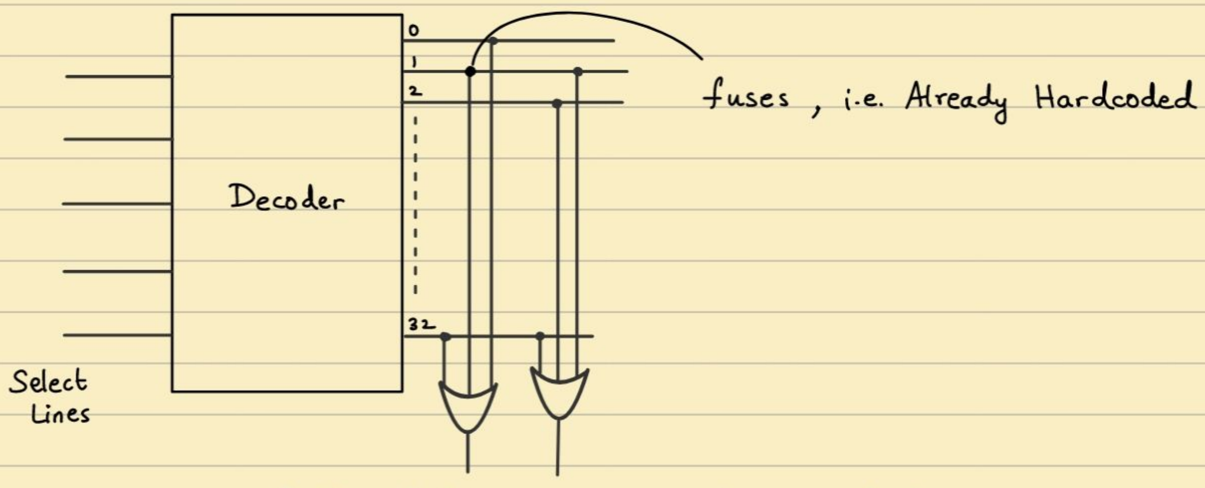
Each address will have
$2^{10} \times 4 = 4096$ bytes [1 kilobyte : 1024 bytes]
$= 4$ kilobytes
$\Rightarrow$ Total Memory $= 4$ kilobytes

Q) Word Addressable system , word size $= 2$
Total Memory capacity $= 32$ kilobytes

Sol: In bytes, $32 \times 2^{10} = 2 \times 2^k$ [$\omega = 2$]
$\Rightarrow 2^5 \times 2^{10} = 2^{k+1}$
$\Rightarrow 2^{k+1} = 2^{15}$
$\boxed{k = 14}$

$\therefore$ 14 bits

(CN course – 10^3 [1000]
COA course – 2^{10} [1024])



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
(EP ROM)

ROM does not use flip-flops.

↪ Given small voltage, Whatever is stored will execute.

RAM : Sequential Circuit
ROM : Combinational Circuit (No memory unit)

——— X ———