

02-09-2024

Advanced Digital Design.

Vaibhav Tarale

- Basic understanding
- Application
- Apply in practical environment.

(Assignment)

Q Why digital design?

- Different blocks in uC
- Digital 1 (or) High [$< +5V$ (Important)] $\rightarrow V_{dd} = V_{cc}$
- Digital 0 (or) Low $0V \rightarrow V_{ss}$ (cmos logic)

Basics of digital circuit

Objectives of design

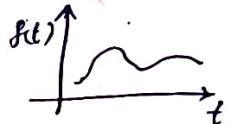
- Power \downarrow
- Area \downarrow
- Speed \uparrow

Important objectives during design

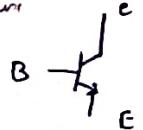
- Area should be minimum, power should be minimum & speed should be maximum

Soc will have digital & analog section & our objective are following.

Digital - 0 or 1 [Discrete]
Analog - $f(t)$ [Function of time]



$f(t) \rightarrow$ (TTL \rightarrow BJT)
(open collector) cannot pull high
 V_{ee}
ECL \rightarrow P↑
BICMOS \rightarrow more



1. Use of basic design concepts to design logic
2. Optimization of digital logic to have min area, min power & max speed
3. To design efficient finite state machine (FSM)
4. Understand the processor architecture requirements & design efficient architecture & micro architecture
5. Timing analysis of digital circuit

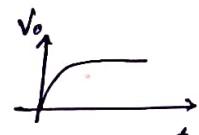
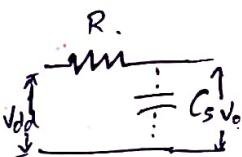
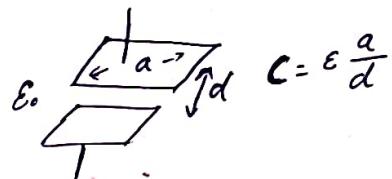
→ CMOS - NAND Explained using diagram

Ideal $\Rightarrow A$

$\gamma = A$

→ Stray capacitance } charging
→ wire resistance } discharging

\Rightarrow Low power filter



Energy in capacitors

$$E = \int P dt$$

$$\Rightarrow E = \frac{1}{2} CV^2 (J)$$

$$Pt = \frac{1}{2} CV^2$$

$$\Rightarrow P = \frac{1}{2} CV^2 f$$

Reducing power

→ Reducing Voltage level $V = 5V$

by using different
fab node
(lower nm, lower V)

3.3V

2.8V

2.4V

2.0V

1.8V

1.5V

1.2V

1.0V

0.8V

0.6V

↓ low voltage

Reduce power.

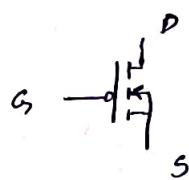
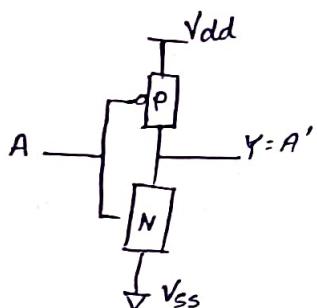
→ multiple clock & multiple power voltage

180 nm → 5 nm [Reduce capacitance]

∴ → cannot be reduced (reduce performance)

$$A \rightarrow Y = \bar{A}$$

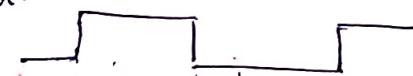
A	Y
0	1
1	0



$$\tau = RC_s$$

$$V_c = V_f (V_i - V_f) e^{-t/\tau}$$

$$\Rightarrow V_c = V_{dd} (1 - e^{-t/\tau})$$



practical



propagation delay

t_{pd} (LOW → HIGH
HIGH → LOW)

$t_{pd LH}$

$t_{pd HL}$

!!!

→ To be discussed
during timing analysis.

$$t_{pd} = \frac{t_{pd LH} + t_{pd HL}}{2}$$

practically every wire will have resistance R & as there is formation of stray capacitance at the o/p of a gate, it will take some time for charging & discharging.

Properties of capacitor

- Energy storage element

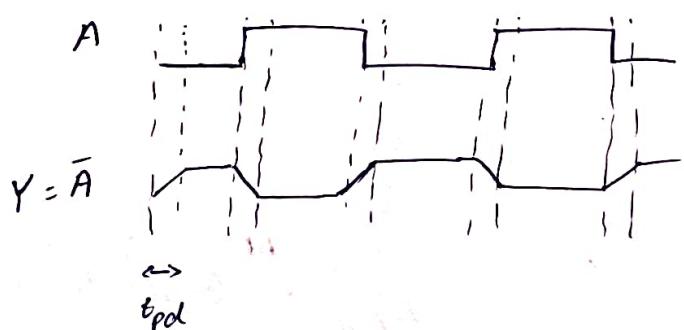
- $E = \frac{1}{2} CV^2$

- $V_c = V_{dd} (1 - e^{-t/\tau})$

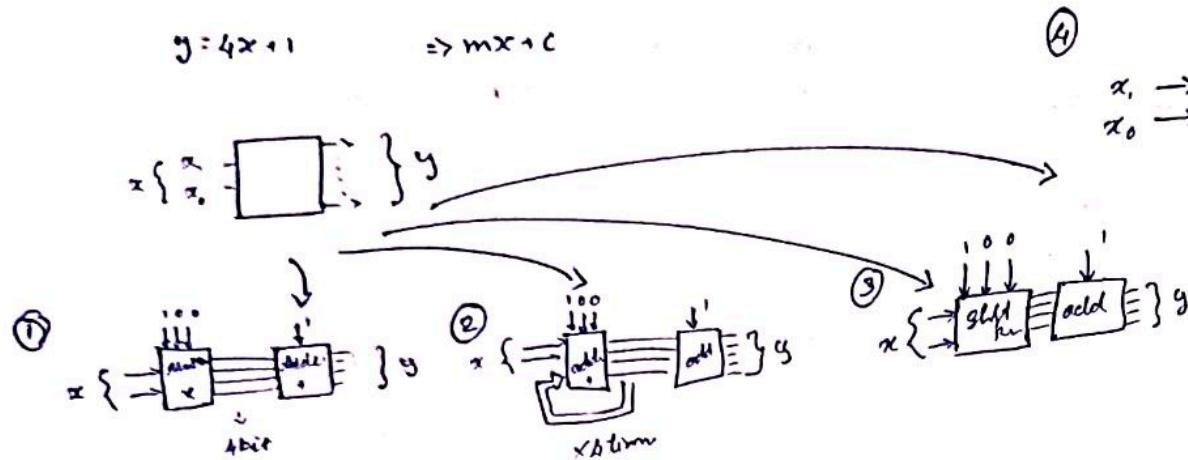
voltage across capacitor cannot change instantaneously it will take finite amount of time to charging & discharging. Due to that every logic gate have propagation delay

t_{pd} (propagation delay)

minimum amount of time required to have valid stable o/p after change of i/p is called t_{pd}



Area - minimum



Design minimised
based on
truth table
realisation.

Multiplication Binary 2,4,8 [shift digit (shift register)]

Area optimization - consider above example develop a logic for straight line equation $y = 4x + 1$ where x is 2-bit binary number. Following are various architectures to design logic. but we need to select architecture which has min area, min power & max speed.

Basics of digital circuit

Logic gates

NOT

$$A \rightarrow D \rightarrow Y$$

A	Y = \bar{A}
0	1
1	0

AND

$$A \otimes B \rightarrow D \rightarrow Y$$

AB	Y = A + B
00	0
01	1
10	1
11	1

AB	Y = AB
00	0
01	0
10	0
11	1

OR

$$A \oplus B \rightarrow D \rightarrow Y$$

AB	Y = A + B
00	0
01	1
10	1
11	1

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

NAND

$$\overline{A} \otimes \overline{B} \rightarrow D \rightarrow Y$$

AB	Y = $\overline{A+B}$
00	1
01	0
10	0
11	0

NOR

$$\overline{A} \oplus \overline{B} \rightarrow D \rightarrow Y$$

AB	Y = $\overline{A+B}$
00	1
01	0
10	0
11	0

XOR

$$A \oplus B \rightarrow D \rightarrow Y$$

AB	Y = AOB
00	0
01	0
10	1
11	0

XNOR

$$A \otimes B \rightarrow D \rightarrow Y$$

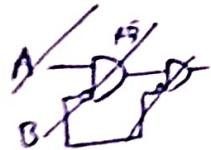
AB	Y = AOB
00	0
01	0
10	0
11	1

→ Equality detector

Lab I Design a 2 ip AND using min no of following logic gates.

$$A \rightarrow D \rightarrow B \rightarrow Y = AB$$

AB	Y
00	0
01	0
10	1
11	0



- go from o/p
- formulate i/p

$$B \rightarrow \overline{D} \rightarrow A \rightarrow D \rightarrow Y = AB$$

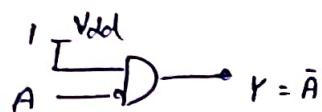


→ Session -2

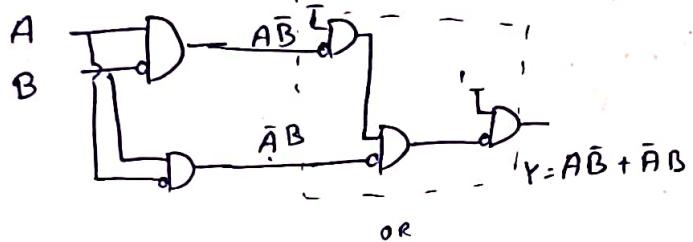
Lab II - Implement the logic for all possible gates using
min no of following gates



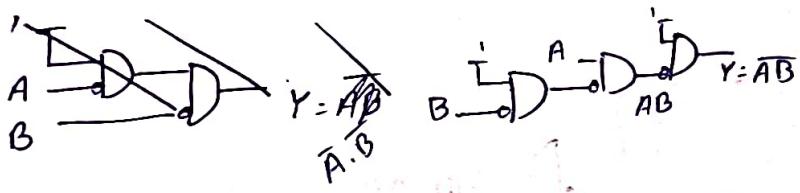
1) NOT



2) XOR



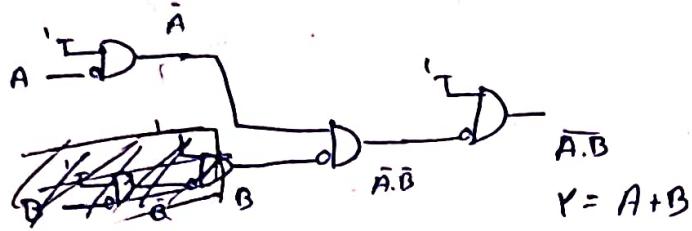
5) NAND



4) AND

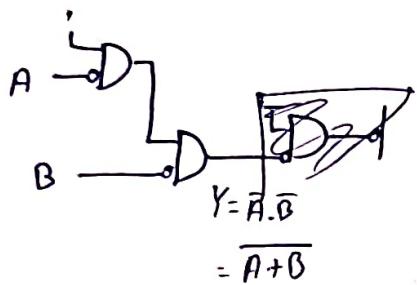


2) OR

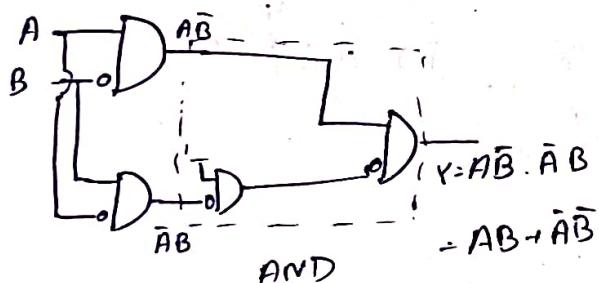


$$\overline{\bar{A} \cdot \bar{B}} = A + B$$

3) NOR

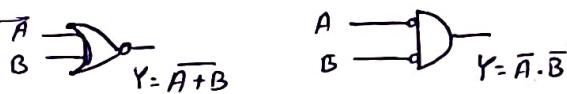


7) XNOR



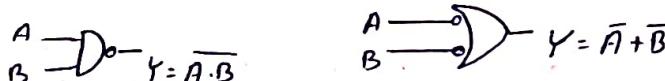
- Gelfond Library - Oei
- George Boole -
- Boolean Algebra
- Shannon expansion -
- K-map /
- Bell lab - Brattain, Bardeen, Whistler Shockley
- Jack Kilby → first IC JK flipflop

De-Morgan's Theorem NOR = bubbled AND



A	B	$\overline{A+B}$	$\bar{A} \cdot \bar{B}$	$\bar{A} \cdot \bar{B}$
0	0	1	1	1
0	1	0	10	0
1	0	0	01	0
1	1	0	00	0

NAND = bubbled OR



AB	$\overline{A \cdot B}$	$\bar{A} + \bar{B}$	$\bar{A} + \bar{B}$
00	1	11	1
01	1	10	1
10	1	01	1
11	0	00	0

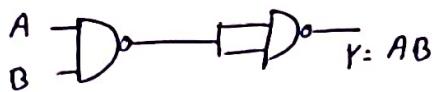
Lab 31

Using min no of 2 i/p NAND design or implement all possible logic gates

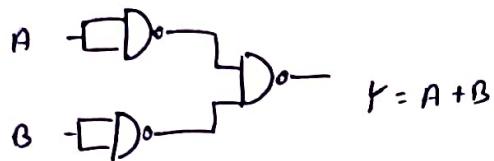
1) NOT



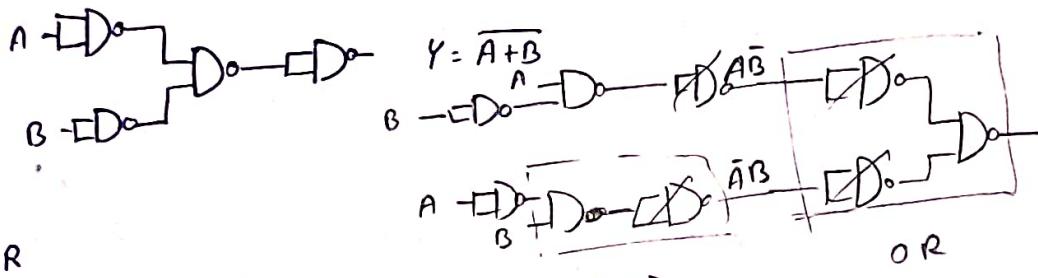
2) AND



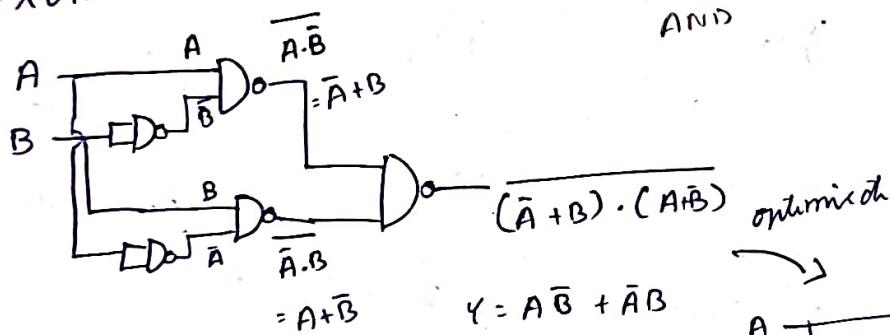
3) OR



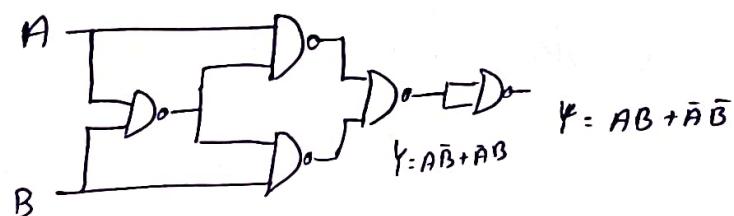
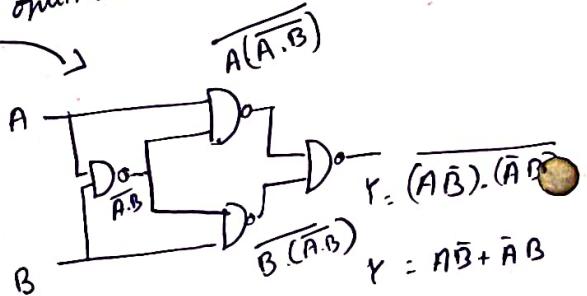
4) NOR



5) XOR



6) XNOR



XOR

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

diff plane

OR

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

AND

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

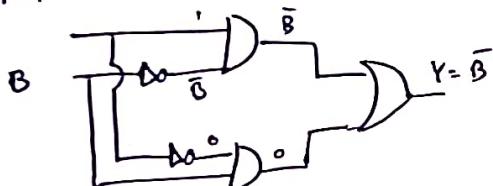
XNOR

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

!!! Min. no. of gate

Logic optimization for XOR

$$A=1$$



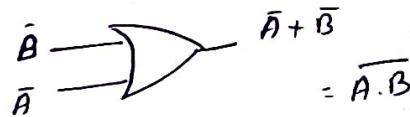
$$A$$

$$B=1$$

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

	NAND	NOR
NOT	1	1
AND	2	3
OR	3	2
NAND	1	4
NOR	4	1
XOR	4	5
XNOR	5	4

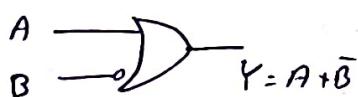
$$A=1, B=1$$



Assignment I

Implement the logic for all possible logic gates using min no of following gate

no of following gate

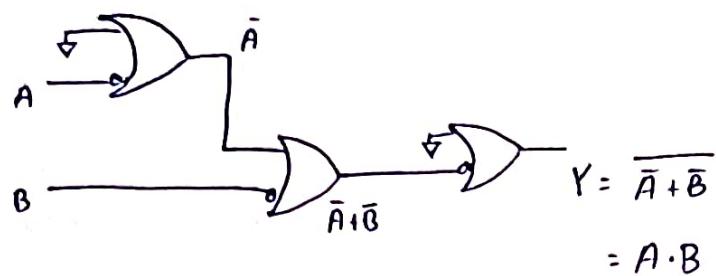


A	B	Y
0	0	1
0	1	0
1	0	1
1	1	1

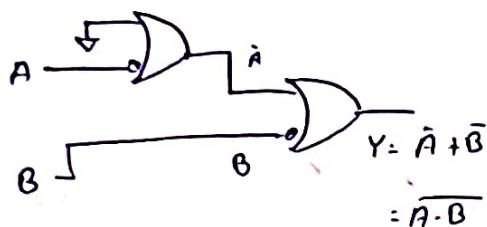
i) NOT



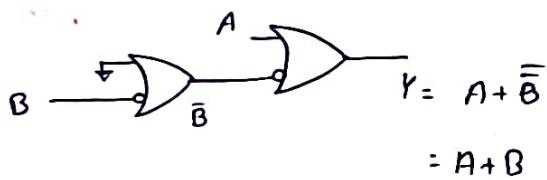
2) AND



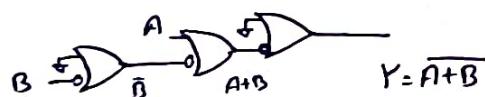
3) NAND



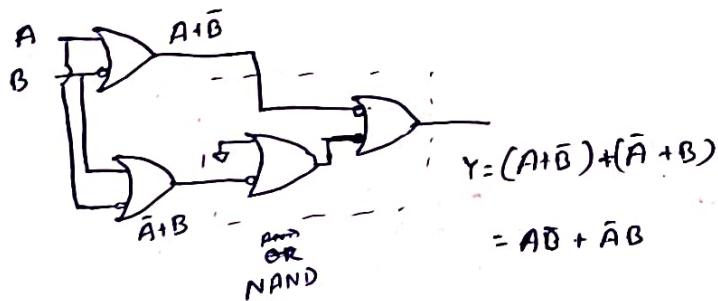
4) OR



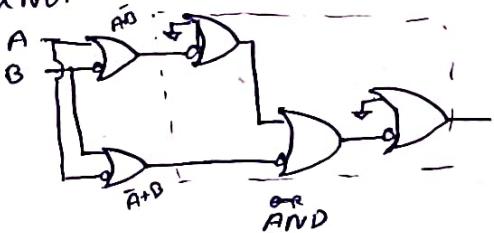
5) NOR



6) XOR

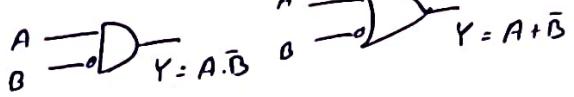
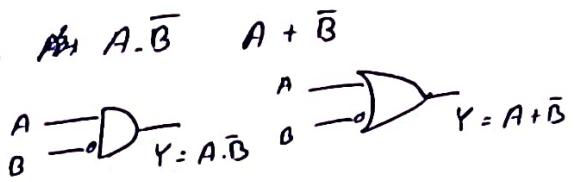


7) XNOR



NAND, NOR are universal logic gates because by using them

we can design any logic.



these are also universal logic gates

03/09/2024

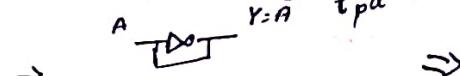
Combinational Circuit

$o/p = f(\text{present } i/p)$

↓
No feedback

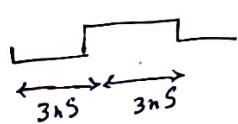
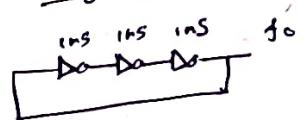
No memory

propagation delay:



!!! $\rightarrow 6 \text{ ns} = \text{Time period}$
NOT 3 ns

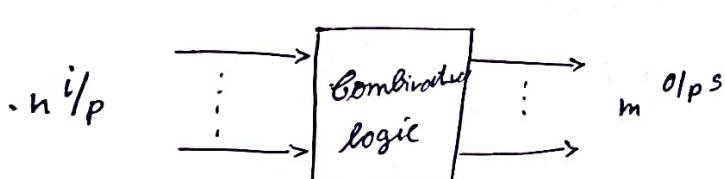
\Rightarrow Ring Oscillator



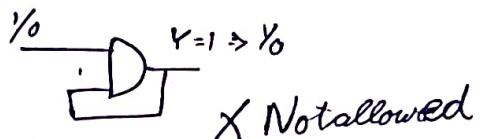
$$f = \frac{1}{6 \text{ ns}}$$

$$\Rightarrow f = \frac{1}{2n t_{pd}}$$

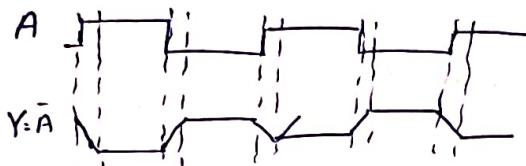
n - no of NOT gate



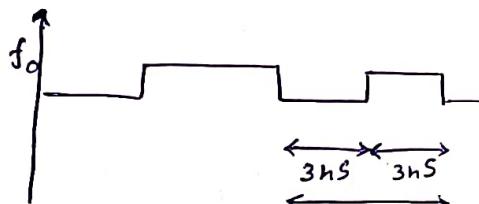
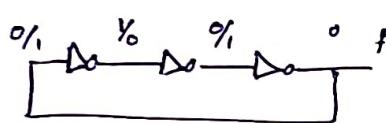
Combinational circuit elements are MUX - deMUX, decoder, encoder
 Practical scenario, during the design we should avoid combinational loops / combo loops. Because combo loops are having oscillatory behaviour. For example



Consider NOT gate having t_{pd} propagation delay



Here, o/p is 180° out of phase with i/p we can use this NOT gate to have Ring oscillator as shown in following



$6ns \Rightarrow$ Time period

$$f_o = \frac{1}{6ns}$$

$$= \frac{1000}{6} \times 10^6$$

$$= 166.67 \text{ MHz}$$

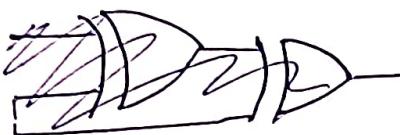
$$\Rightarrow f_o = \frac{1}{2n t_{pd}}$$

n - no of NOT gate

t_{pd} - propagation delay

1. Design NOT gate with min no of XOR gate

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



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



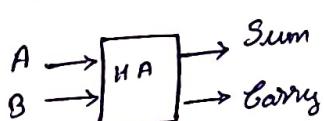
2. Design NOT gate with min no of XNOR gate



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

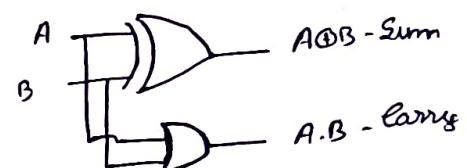
Arithmetic circuits.

Half Adder

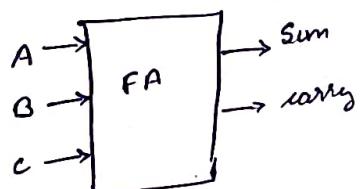


AB	Sum	Carry
00	0	0
01	1	0
10	1	0
11	0	1

XOR AND



Full Adder



ABC	Sum	Carry
000	0	0
001	1	0
010	1	0
011	0	1
100	1	0
101	0	1
110	0	1
111	1	1

Gray code \leftarrow

00	01	11	10
----	----	----	----

K-map

Sum

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

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

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

Carry

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

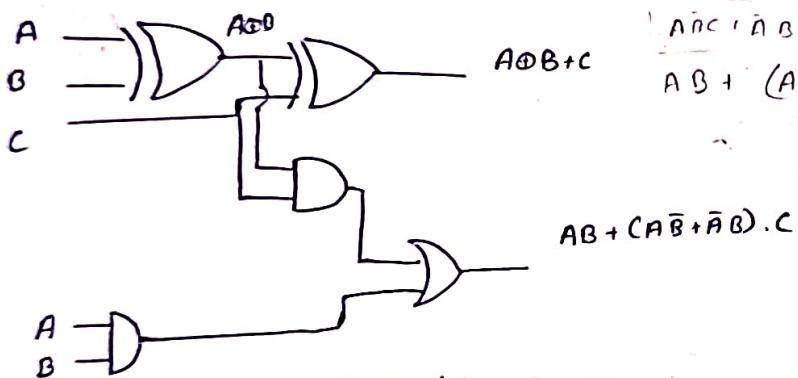
$$\Rightarrow AB + BC + CA$$

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

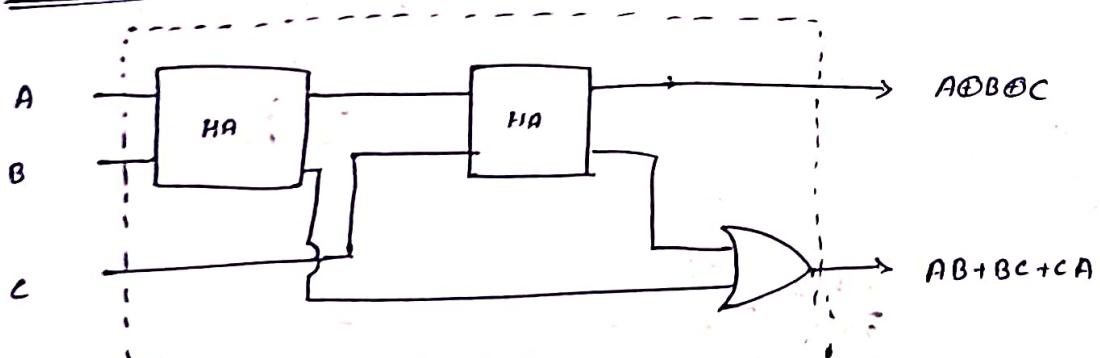
$$AB + A\bar{B}C + \bar{A}BC$$

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

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



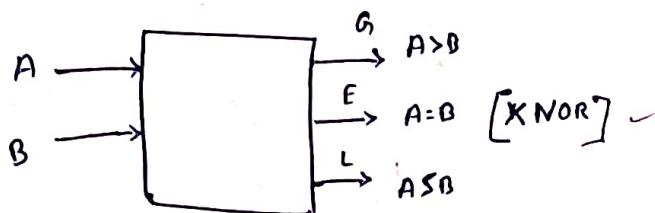
Full adder design using Half Adder & OR gate



Full adder.

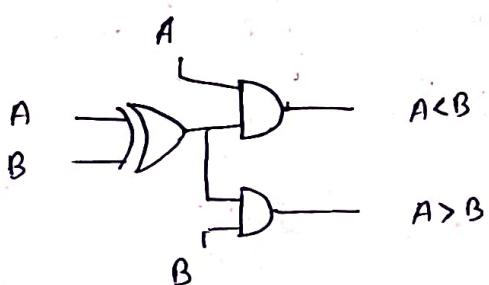
Exercise.5

Design 1 bit comparator

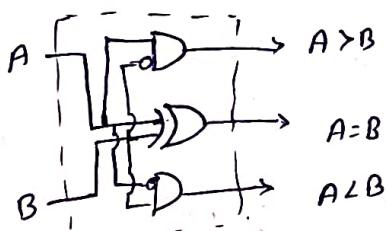


A	B	E
0	0	1
0	1	0
1	0	0
1	1	1

A	B	E
0	0	0
0	1	1
1	0	1
1	1	0



min gates



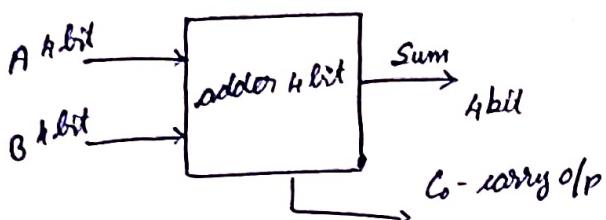
→ Number system, Bin, Hex, gray, BCD, 2's comp, → addition,

EDA of AMD Simon \rightarrow Different type of adder
carry

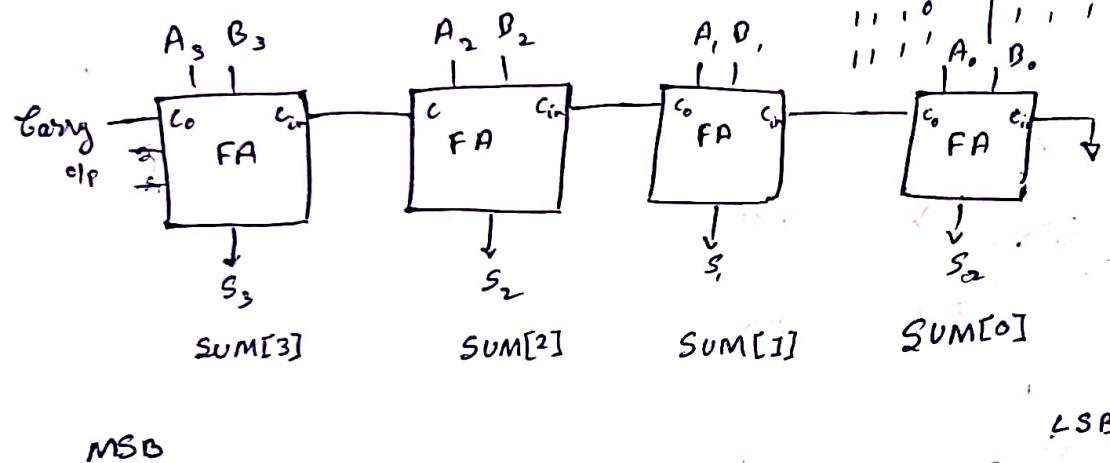
Evening session

Exercise - 6

Design 4 bit adder



A	B	Sum	Carry
0000	0000	0000	0
0000	0001	0001	0
0001	0010	0010	0
0010	0011	0011	0
0011	0100	0100	0
0100	0101	0101	0
0101	0110	0110	0
0110	0111	0111	0
0111	1000	1000	0
1000	1001	1001	0
1001	1010	1010	0
1010	1011	1011	0
1011	1100	1100	0
1100	1101	1101	0
1101	1110	1110	0
1110	1111	1111	0



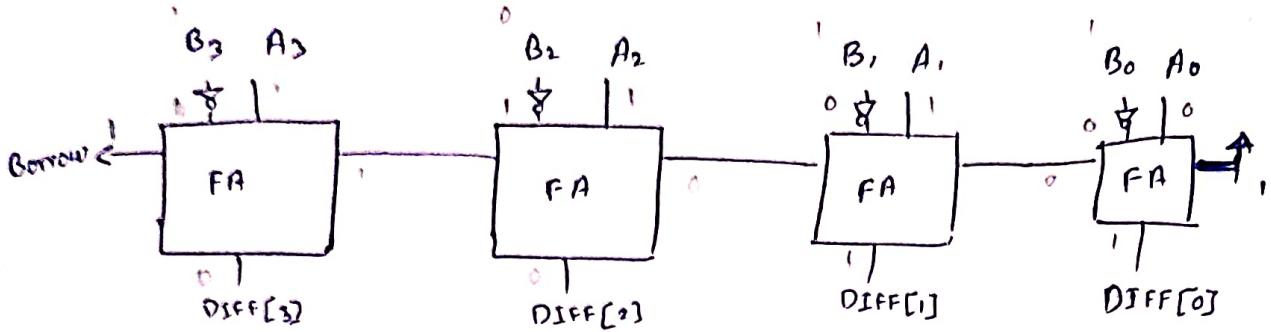
Ripple carry adder.

\rightarrow carry setting register

Exercise - 7

I have full adder & logic gate. Design 4-bit binary subtractor using min no. of full adders = min no. of logic gates.

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

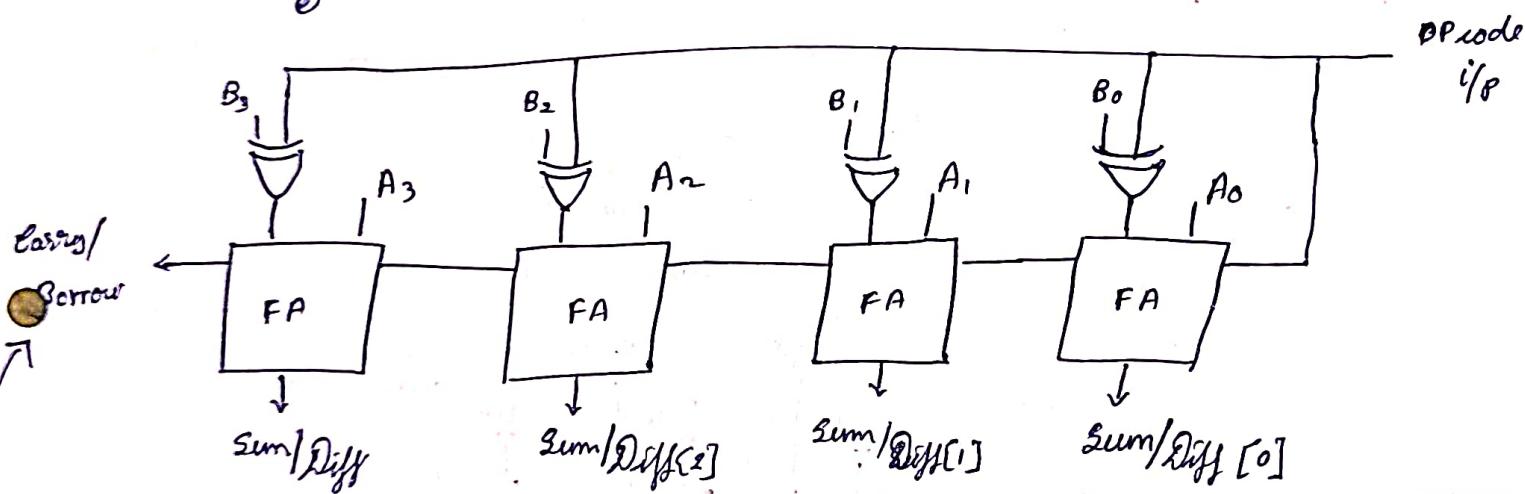


$$\begin{array}{r}
 0100 \\
 0101 \\
 \hline
 1111
 \end{array}
 \quad
 \begin{array}{r}
 1110 \\
 1011 \\
 \hline
 0010
 \end{array}$$

Exercise - 8

use logic optimization to design 4 bit adder subtractor

Opecode	operation
0	$A + B = 0$
1	$A - B$
0	$A + \bar{B} + 1$



Resource sharing is powerful technique to optimize area. In this common resources are shared. For example. If I will to design ADDER/SUBTRACTOR then I will use common resource in adder & additional control gates so that I can get either addition or subtraction.

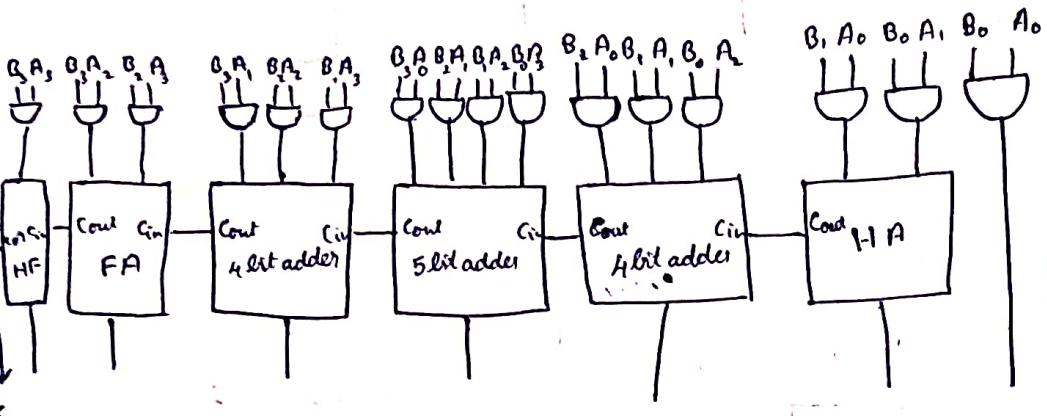
4 bit adder/subtractor using resource sharing

- Booth Multiplier
- Sequential Multiplier

Exercise-8⁹

Design 4-bit binary multiplier using min logic gates & address

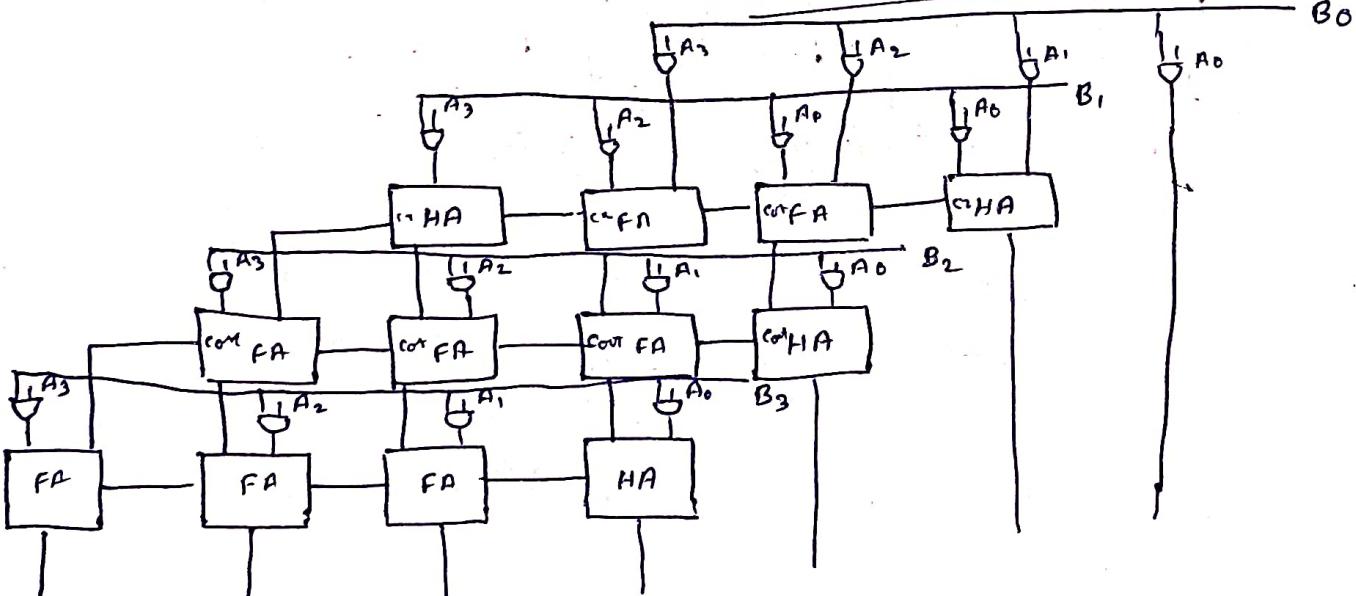
$$\begin{array}{r}
 1010 \\
 \times 1010 \\
 \hline
 10000 \\
 +1010 \\
 \hline
 110100 \\
 +1010 \\
 \hline
 1110100 \\
 \end{array}$$



multiplier o/p

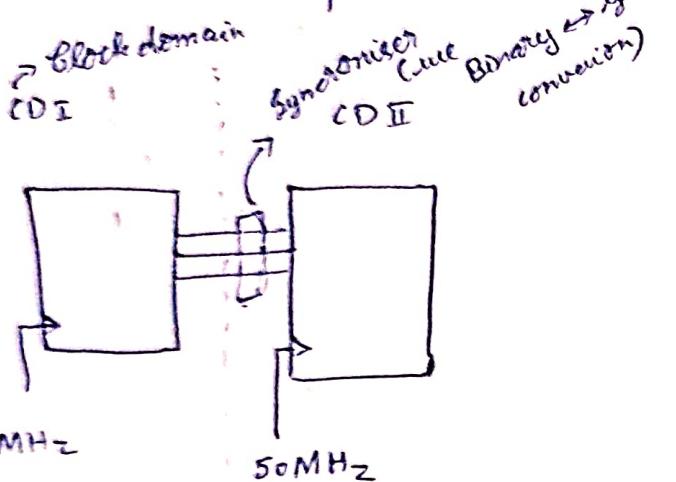
$$\begin{array}{r}
 A_3 \quad A_2 \quad A_1 \quad A_0 \\
 B_3 \quad B_2 \quad B_1 \quad B_0 \\
 \hline
 \end{array}$$

$$\begin{array}{c}
 A_3 B_0 \quad A_2 B_0 \quad A_1 B_0 \quad A_0 B_0 \\
 A_3 B_1 \quad A_2 B_1 \quad A_1 B_1 \quad A_0 B_1 \\
 A_3 B_2 \quad A_2 B_2 \quad A_1 B_2 \quad A_0 B_2 \\
 A_3 B_3 \quad A_2 B_3 \quad A_1 B_3 \quad A_0 B_3 \\
 \hline
 \end{array}$$



04/09/2024

Binary	Gray code
0 0	0 0
0 1	0 1
1 0	1 1
1 1	1 0



Binary

→ more power loss in switching.

gray

→ reduced & switching loss

as, only one bit change in
two consecutive no

→ used in multiple clock domain

design
→ why?

FIFO to get address gray pointer / counter are used.

Gray Code

- are unique cyclic codes and are used to minimize power as only one bit changes in two consecutive gray numbers. Application used in multiple clock domain design. e.g. in Sync. FIFO who have gray pointer to consider following design who to have 2-bit gray code

B ₁ , B ₀	C ₁ , C ₀
0 0	0 0
0 1	0 1
1 0	1 1
1 1	1 0

B ₁	\bar{B}	B
0	0	0
1	1	1

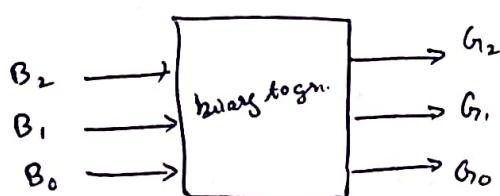
$$G_{i_1} = B$$

\bar{B}	B
0	1
1	0

$$G_{i_0} = B, \bar{B}_0 + \bar{B}, B_0$$

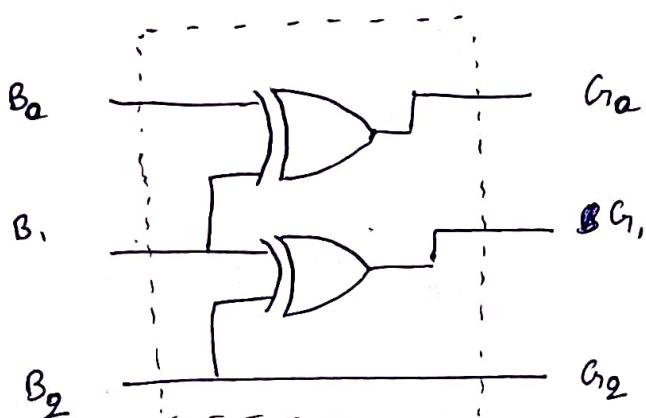
Exercise 10

Design 3 bit binary to gray code converter using min no of logic gates



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

LSB



msb

3 bit binary to gray convert

$G_2 \ \bar{B}_2 \bar{B}_0 \ \bar{B}_1 \bar{B}_0 \ B_2 \bar{B}_0 \ B_1 \bar{B}_0$
\bar{B}_2 0 0 0 0
B_2 1 1 1 1

$$G_2 = B_2$$

$G_1 \ \bar{B}_2 \bar{B}_0 \ \bar{B}_1 \bar{B}_0 \ B_2 \bar{B}_0 \ B_1 \bar{B}_0$
\bar{B}_2 0 0 1 1
B_2 1 1 0 0

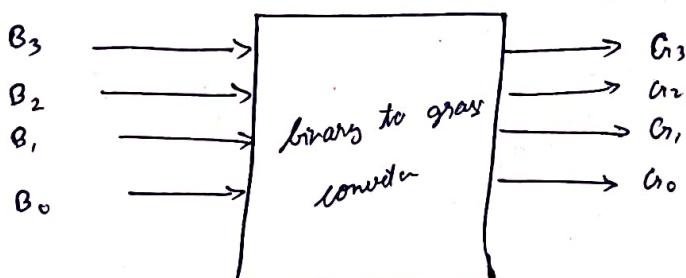
$$G_1 = B_1 \bar{B}_2 + \bar{B}_1 B_2$$

$G_0 \ \bar{B}_2 \bar{B}_0 \ \bar{B}_1 \bar{B}_0 \ B_2 \bar{B}_0 \ B_1 \bar{B}_0$
\bar{B}_2 0 1 0 1
B_2 0 1 0 1

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

Exercise 11

Design 4 bit binary to gray code converter



	Binary $B_3 B_2 B_1 B_0$	Gray G_3, G_2, G_1, G_0	Decimal
0	0 0 0 0	0 0 0 0	0
1	0 0 0 1	0 0 0 1	1
2	0 0 1 0	0 0 1 1	3
3	0 0 1 1	0 0 1 0	2
4	0 1 0 0	0 1 1 0	6
5	0 1 0 1	0 1 1 1	7
6	0 1 1 0	0 1 0 1	5
7	0 1 1 1	0 1 0 0	4
8	1 0 0 0	1 1 0 0	12
9	1 0 0 1	1 1 0 1	13
A	1 0 1 0	1 1 1 1	15
B	1 0 1 1	1 1 1 0	14
C	1 1 0 0	1 0 1 0	10
D	1 1 0 1	1 0 1 1	11
E	1 1 1 0	1 0 0 1	9
F	1 1 1 1	1 0 0 0	8

G_3	$\bar{B}_3 \bar{B}_2$	$\bar{B}_3 B_2$	$B_3 \bar{B}_2$	$B_3 B_2$
$\bar{B}_3 \bar{B}_2$	0	0	0	0
$\bar{B}_3 B_2$	0	0	0	0
$B_3 \bar{B}_2$	1	1	1	1
$B_3 B_2$	1	1	1	1

$$G_3 = B_3$$

G_2	$\bar{B}_3 \bar{B}_2$	$\bar{B}_3 B_2$	$B_3 \bar{B}_2$	$B_3 B_2$
$\bar{B}_3 \bar{B}_2$	0	0	0	0
$\bar{B}_3 B_2$	1	1	1	1
$B_3 \bar{B}_2$	0	0	0	0
$B_3 B_2$	1	1	1	1

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

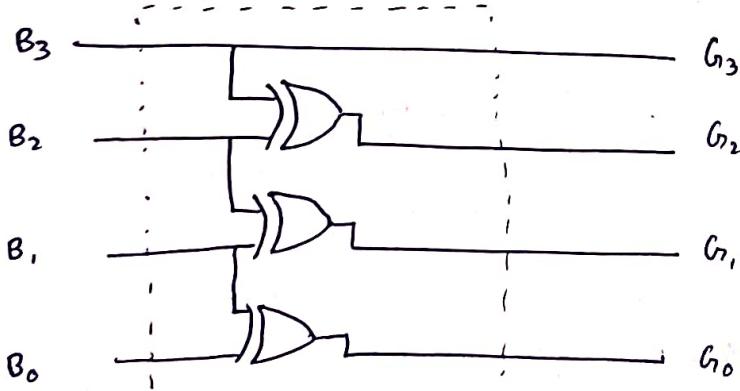
G_1	$\bar{B}_3 \bar{B}_2$	$\bar{B}_3 B_2$	$B_3 \bar{B}_2$	$B_3 B_2$
$\bar{B}_3 \bar{B}_2$	0	0	1	1
$\bar{B}_3 B_2$	1	1	0	0
$B_3 \bar{B}_2$	1	1	0	0
$B_3 B_2$	0	0	1	1

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

G_0	$\bar{B}_3 \bar{B}_2$	$\bar{B}_3 B_2$	$B_3 \bar{B}_2$	$B_3 B_2$
$\bar{B}_3 \bar{B}_2$	0	1	0	1
$\bar{B}_3 B_2$	0	1	0	1
$B_3 \bar{B}_2$	0	1	0	1
$B_3 B_2$	0	1	0	1

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

MSB



LSB

4 bit binary to gray code.

Exercise 12

Design 4-bit gray to binary code converter.

B_3	\bar{G}_1, \bar{G}_0	\bar{G}_1, G_0	G_1, G_0	G_1, \bar{G}_0
\bar{G}_3, \bar{G}_2	0 0 0 0			
\bar{G}_3, G_2	0 0 0 0			
G_3, G_2	1 1 1 1			
G_3, \bar{G}_2	1 1 1 1			

$$B_3 = G_3$$

B_3	\bar{G}_1, \bar{G}_0	\bar{G}_1, G_0	G_1, G_0	G_1, \bar{G}_0
\bar{G}_3, \bar{G}_2	0 0 0 1 1			
\bar{G}_3, G_2	1 1 0 0			
G_3, G_2	0 0 1 1			
G_3, \bar{G}_2	1 1 0 0			

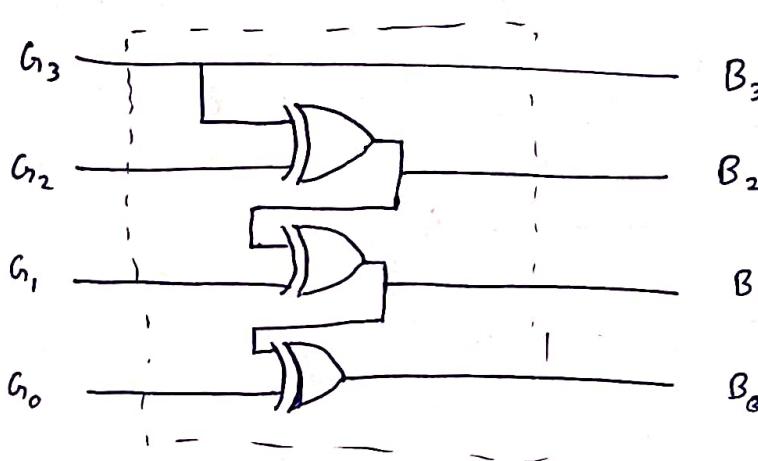
$$\begin{aligned}
 B_3 &= G_1 \bar{G}_3 \bar{G}_2 + \bar{G}_1 \bar{G}_2 \bar{G}_3 + G_1 G_2 \bar{G}_3 + \bar{G}_1 \bar{G}_2 G_3 \\
 &= G_1 (\bar{G}_3 \bar{G}_2 + G_3 G_2) + \bar{G}_1 (G_3 \bar{G}_2 + \bar{G}_3 G_2) \\
 &= G_1 (\bar{G}_3 \bar{G}_2 + \bar{G}_3 G_2) + \bar{G}_1 (G_3 \bar{G}_2 + \bar{G}_3 G_2) \\
 &= G_1 \oplus (G_3 \bar{G}_2 + \bar{G}_3 G_2) \\
 &= G_1 \oplus G_2 \oplus G_3
 \end{aligned}$$

B_2	\bar{G}_1, \bar{G}_0	\bar{G}_1, G_0	G_1, G_0	G_1, \bar{G}_0
\bar{G}_3, \bar{G}_2	0 0 0 0			
\bar{G}_3, G_2	1 1 1 1			
G_3, G_2	0 0 0 0			
G_3, \bar{G}_2	1 1 1 1			

$$B_2 = G_2 \bar{G}_3 + \bar{G}_2 G_3 \quad B_2 =$$

B_0	\bar{G}_1, \bar{G}_0	\bar{G}_1, G_0	G_1, G_0	G_1, \bar{G}_0
\bar{G}_3, \bar{G}_2	0 1 1 0			
\bar{G}_3, G_2	1 0 1 0			
G_3, G_2	0 1 0 1			
G_3, \bar{G}_2	1 0 1 0			

$$B_0 =$$

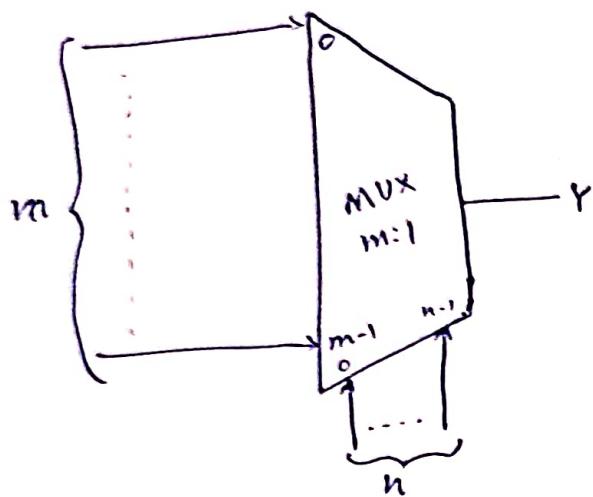


G_3, G_2, G_1, G_0
 $\downarrow \oplus \downarrow \oplus \downarrow \oplus \downarrow$
 B_3, B_2, B_1, B_0

4-bit Gray to binary.

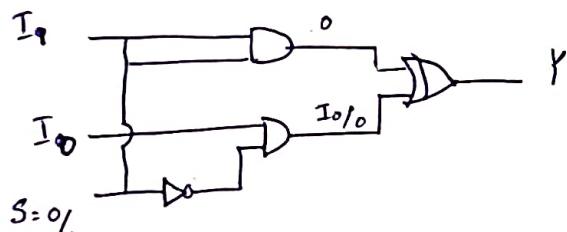
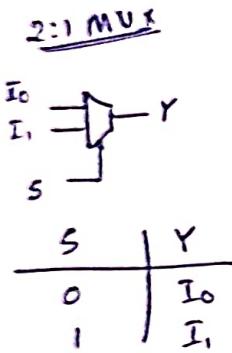
Note:
 Inference it is
 a building block of
 a 2-bit XDR
 So, will be n-bit XOR
 Group!!! carryless similar
 with XNOR but start
 with '1' instead of '0'

Multiplexers (a) MUX



$\Rightarrow m$ - i/p number
 n - select line number

$$2^n = m$$



$$\Rightarrow Y = \bar{S} I_0 + S I_1$$

- it is called as many to one switch consider MUX having m i/p's & single o/p then to select one of i/p at time, we need to have n select lines. Relationship between i/p & select line is given by $m = 2^n$ Referdig

Understanding of basic MUX \rightarrow Consider 2:1 MUX which has 2 i/p (I_0, I_1) & single o/p (Y) & single select (S) Referdig truth table

Consider i) 4:1 MUX

i/p = 4
selectline = 2

iii) 16:1 MUX

i/p = 16
selectline = 4

ii) 8:1 MUX

i/p = 8
selectline = 3

iv) 12:1 MUX

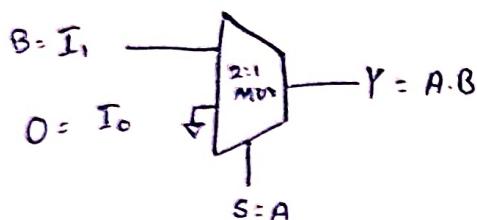
i/p = 12
selectline = 4

Application MUX is universal logic.

1) Design 2*i/p* AND using min no of 2:1 MUX

- Gray to one-hot
- Code conv.

$$\begin{array}{l} B \rightarrow G \\ A \rightarrow D \end{array}$$



A	B	Y = A.B
0	0	0
0	1	0
1	0	0
1	1	1

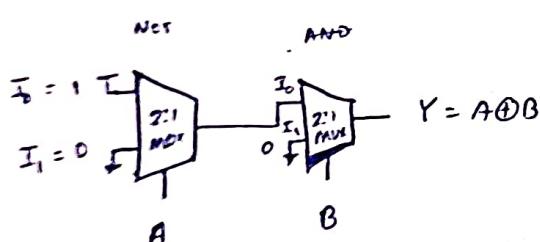
- MUX

$$\begin{array}{l} S = A \\ I_1 = B \\ I_0 = 0 \end{array}$$

$$Y = A \cdot B$$

2) Design 2*i/p* XOR using min no of 2:1 MUX

M



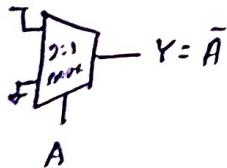
A	B	Y = A ⊕ B
0	0	0
0	1	1
1	0	1
1	1	0

half subtractor

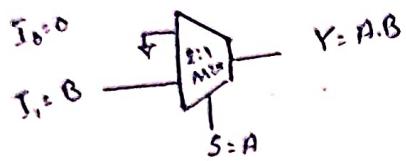
1-bit comparator

Assignment Design all possible logic gate using 2:1 MUX

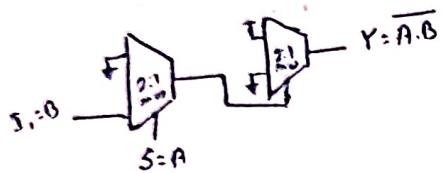
i) NOT



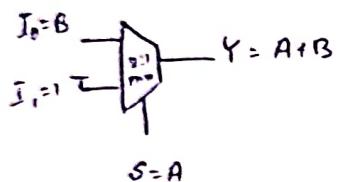
2) AND



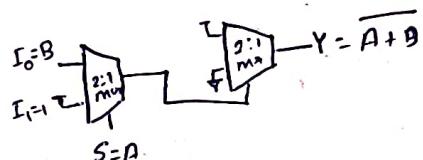
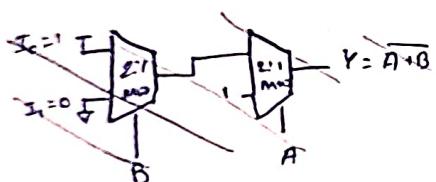
3) NAND



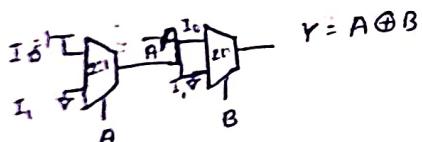
4) OR



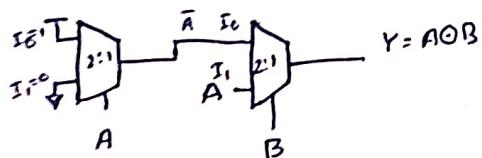
5) NOR



6) XOR

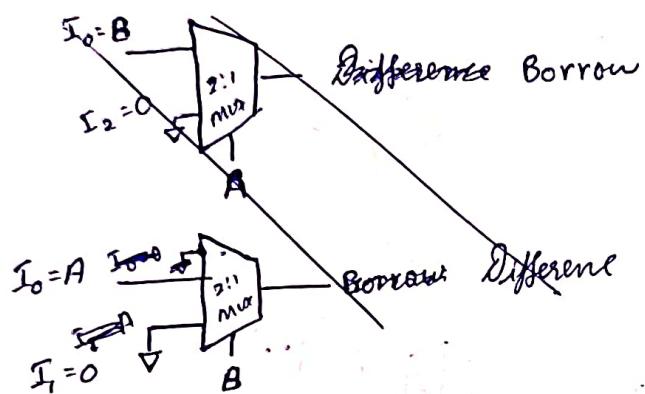
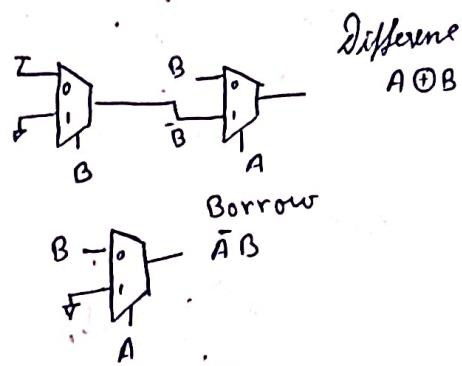


7) XNOR



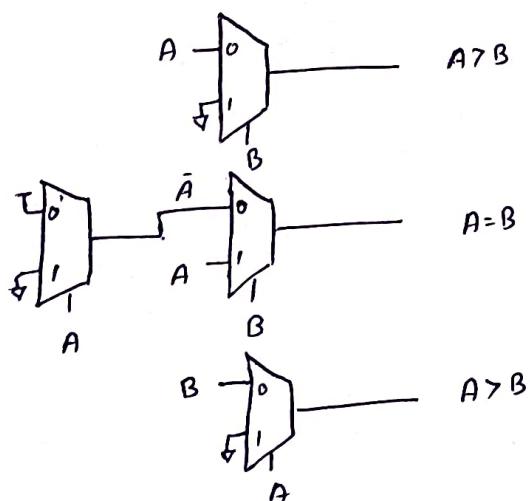
Design half subtractor using 2:1 MUX

A	B	Difference	Borrow
0	0	0	0
0	1	01	1
1	0	1	0
1	1	0	0



Design 1-bit comparator using 2:1 MUX

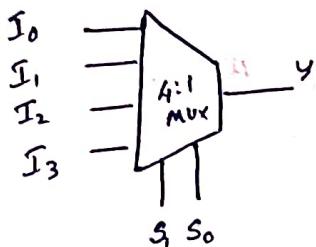
A	B	$A > B$	$A = B$	$A \leq B$
0	0	0	1	0
0	1	0	0	1
1	0	1	0	0
1	1	0	1	0



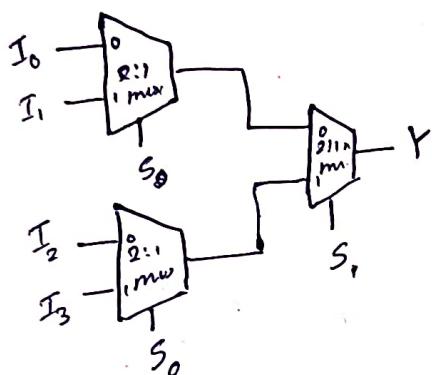
Evening session

Exercise 13

Implement 4:1 MUX with min no of 2:1 MUX

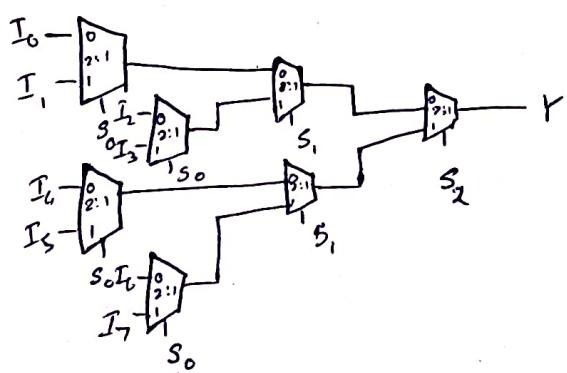
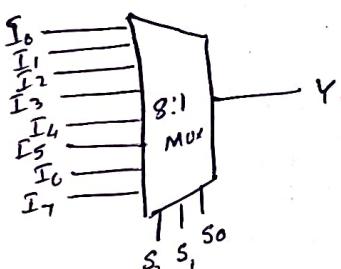


S ₁	S ₀	Y
0	0	I ₀
0	1	I ₁
1	0	I ₂
1	1	I ₃



Exercise 14

Implement 8:1 MUX using min no of 2:1 MUX



S ₂	S ₁	S ₀	Y
0	0	0	I ₀
0	0	1	I ₁
0	1	0	I ₂
0	1	1	I ₃
1	0	0	I ₄
1	0	1	I ₅
1	1	0	I ₆
1	1	1	I ₇

INFERENCE:
Individual building
block of MUX

Exercise 15

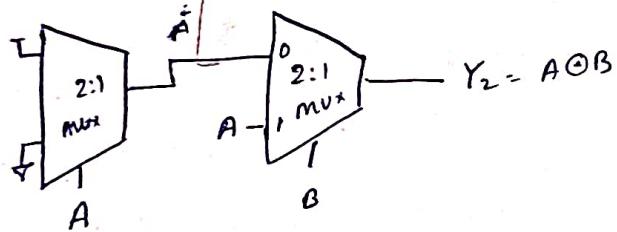
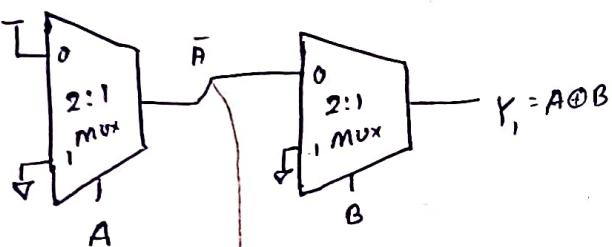
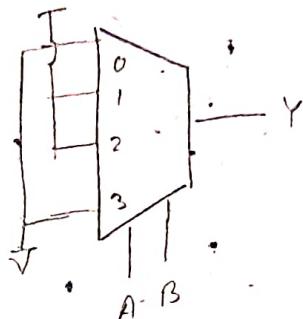
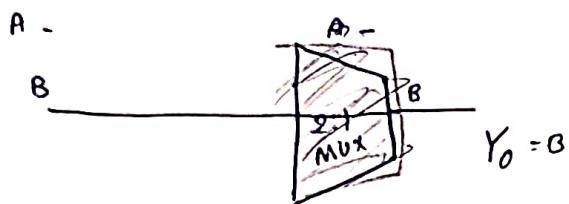
Consider $Y_0 = \Sigma(1, 3)$ Implement following function suitable MUX

$$Y_1 = \Sigma(1, 2)$$

$$Y_2 = \Sigma(0, 3)$$

AB	Y_0	Y_1	Y_2
00	0	0	1
01	1	1	0
10	0	1	0
11	1	0	1

Y_0	\bar{B}	B
0	0	1
1	0	1

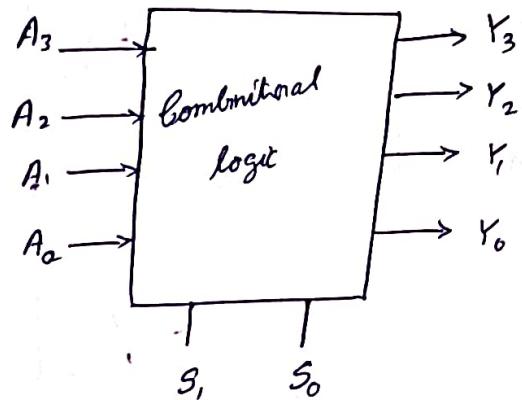


Exclusive complement could be used in previous

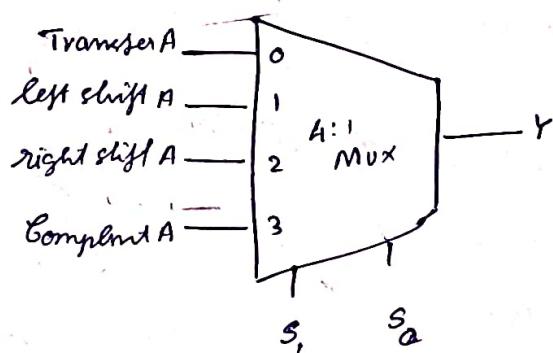
Exercise 16

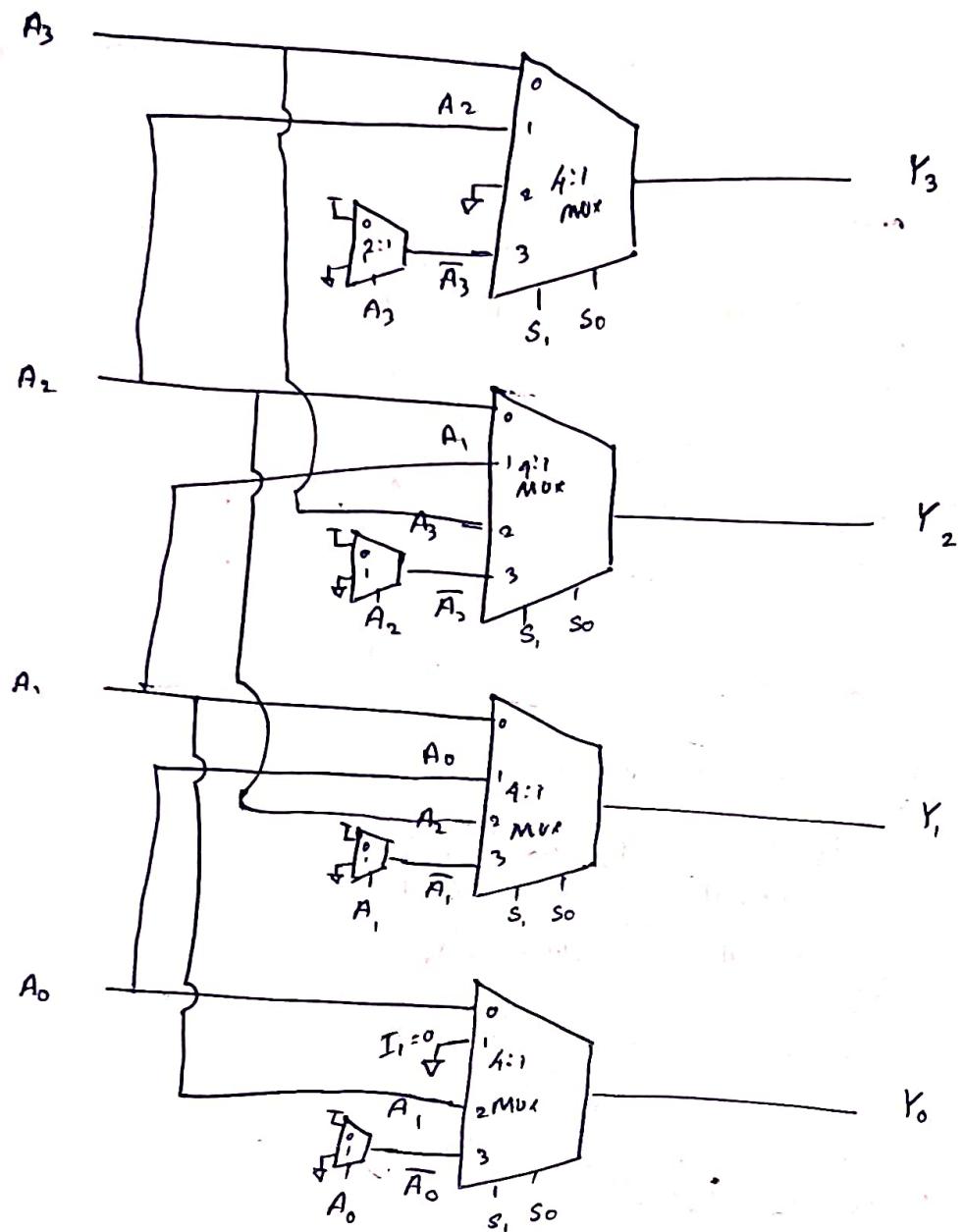
Design combinational circuits using min no. of suitable MUX to perform following operation

S_1	S_0	operation
0	0	Transfer A
0	1	left right shift A
1	0	right shift A
1	1	Complement A



S_1	S_0	operation
0	0	$Y_3 = A_3, Y_2 = A_2, Y_1 = A_1, Y_0 = A_0$
0	1	$Y_3 = A_2, Y_2 = A_1, Y_1 = A_0, Y_0 = 0$
1	0	$Y_3 = 0, Y_2 = A_3, Y_1 = A_2, Y_0 = A_1$
1	1	$Y_3 = \bar{A}_3, Y_2 = \bar{A}_2, Y_1 = \bar{A}_1, Y_0 = \bar{A}_0$

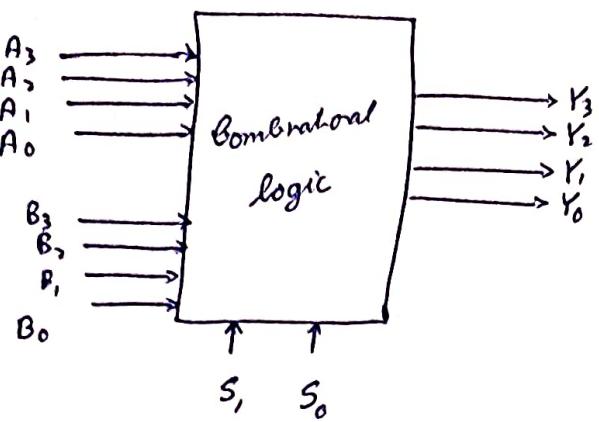




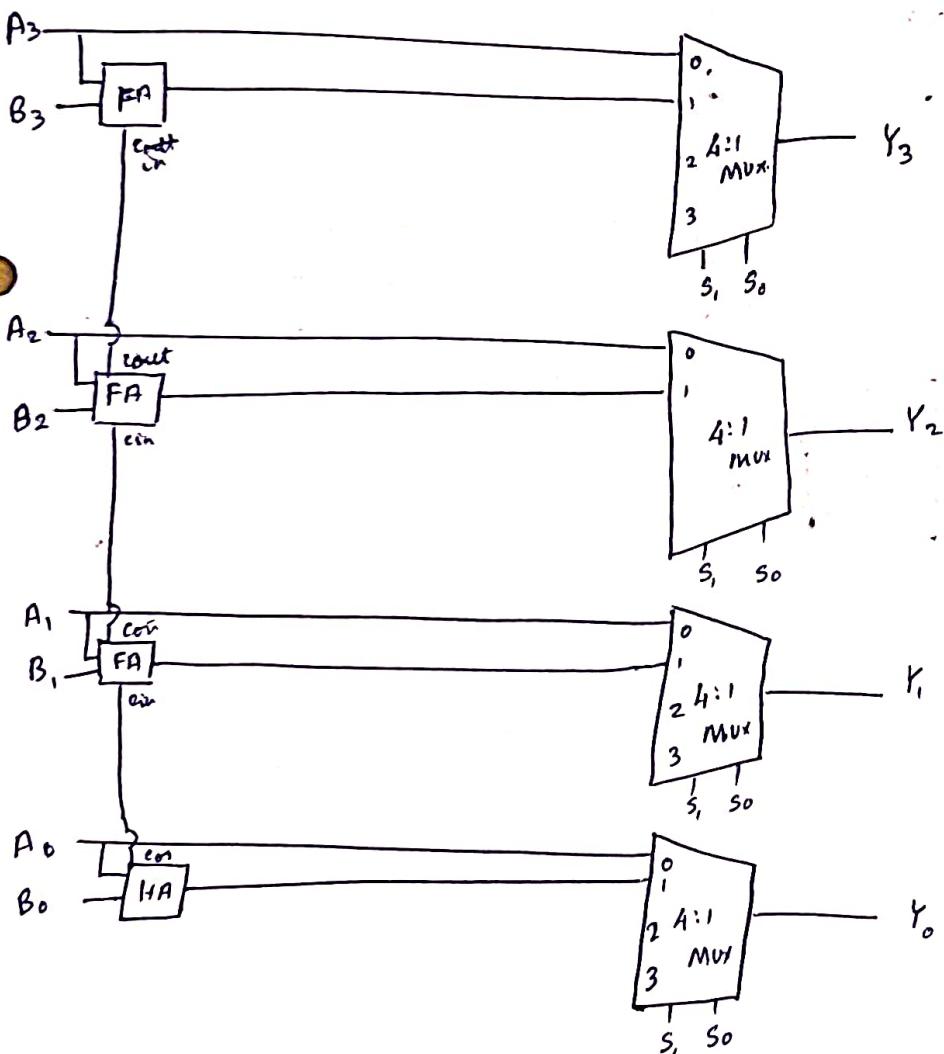
Exercise 11
Design combinational circuit to perform following operation

S_1	S_0	operation
0	0	Transfer A
0	1	$A+B$
1	0	$A-B-1$
1	1	$A-1$

Consider A, B as 4-bit binary number.



S_1	S_0	Y_3	Y_2	Y_1	Y_0
0	0	A_3	A_2	A_1	A_0
0	1		$A + B$		(Adder)
1	0	$A - B - 1 \Rightarrow A + \bar{B} + 1 - 1 \Rightarrow A + \bar{B}$			(Subtractor)
1	1		$A - 1$		



X → Not acceptable as
wastage of power & resources
as all operations are performed

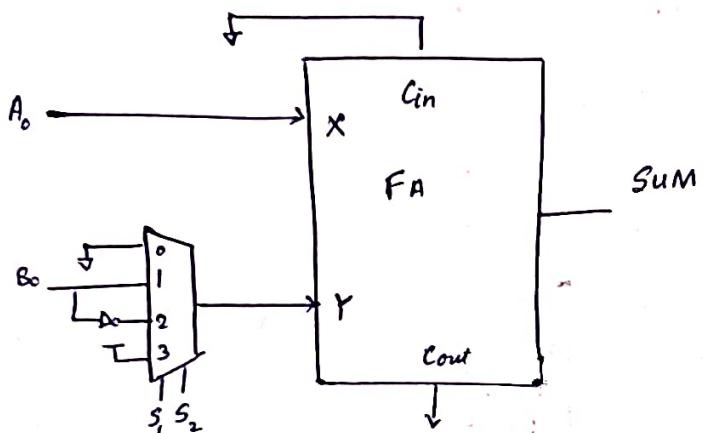
- 1) All operations are happening at a time
- 2) More wastage of power due to high power dissipation
- 3) Area is very huge.

06-09-2024

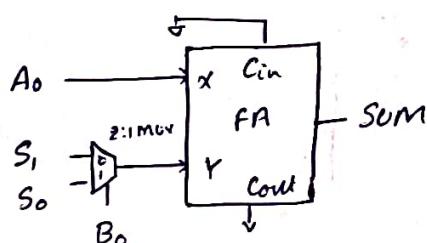
System design optimisation.

S_1	S_0	X	Y	C_{in}
0	0	A	0	0
0	1	A	B	0
1	0	A	\bar{B}	0
1	1	A	1	0

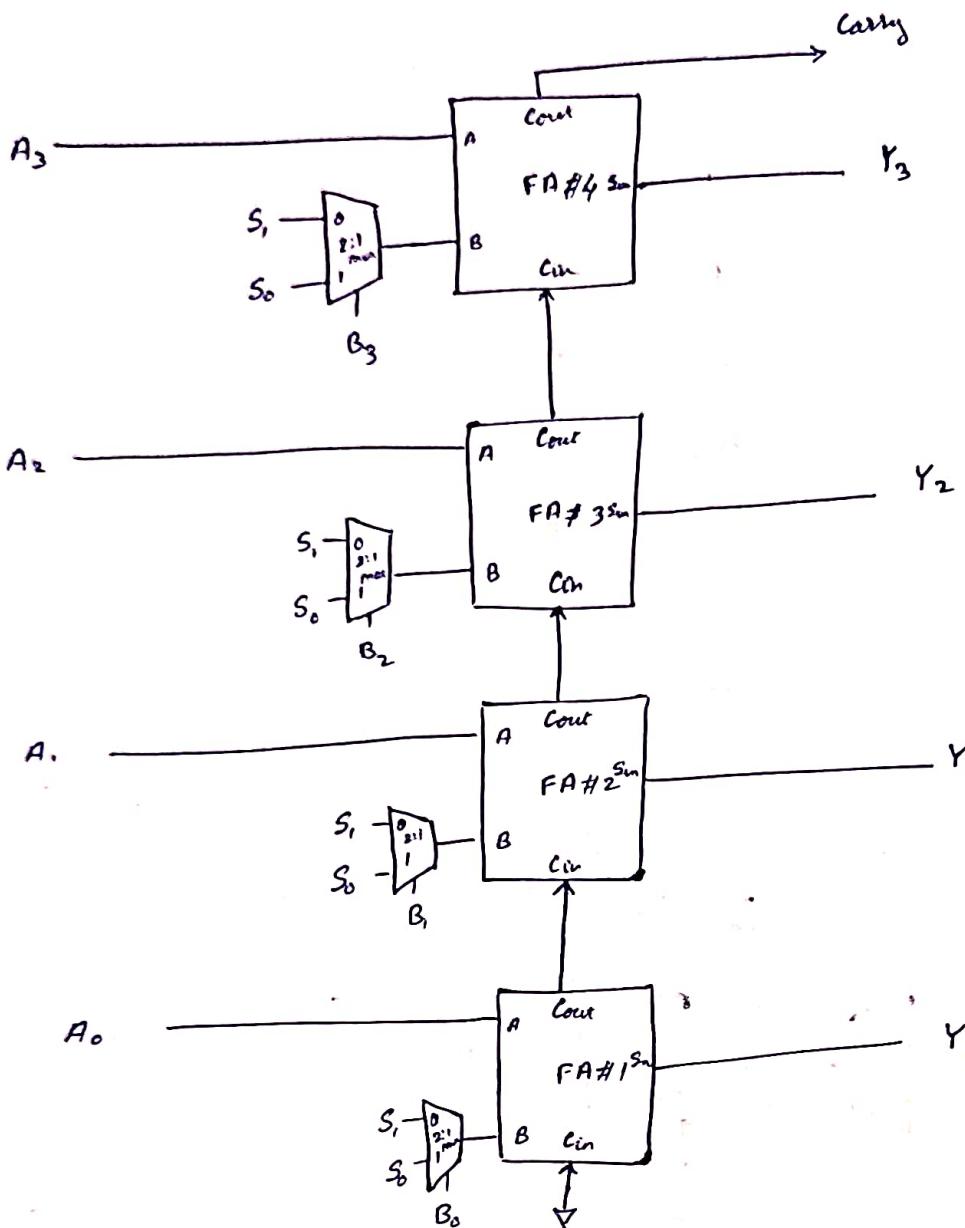
S_1	S_0	operation
0	0	Transfer A
0	1	$A + B$
1	0	$A - B - 1$
1	1	$A - 1$



Further optimising for 4:1 mux with 2:1 mux



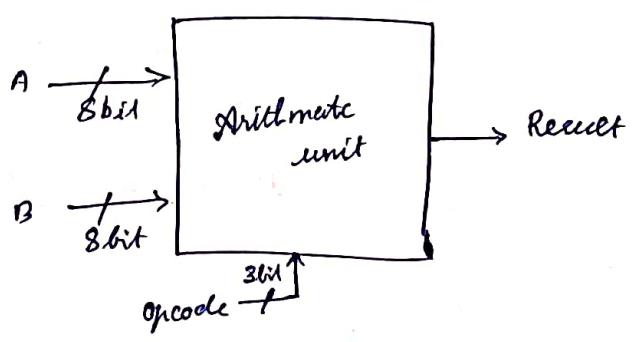
S_1, S_0	Y
00	0
01	B
10	\bar{B}
11	1



Exercise 18

Design the arithmetic unit to perform the following operation using combinational circuit (ckt) element

min no of opcode	combinational operation
0 0 0	Transfer A
0 0 1	$A + B$
0 1 0	$A + B + 1$
0 1 1	$A - B$
1 0 0	$A - B - 1$
1 0 1	$A + 1$
1 1 0	$A - 1$



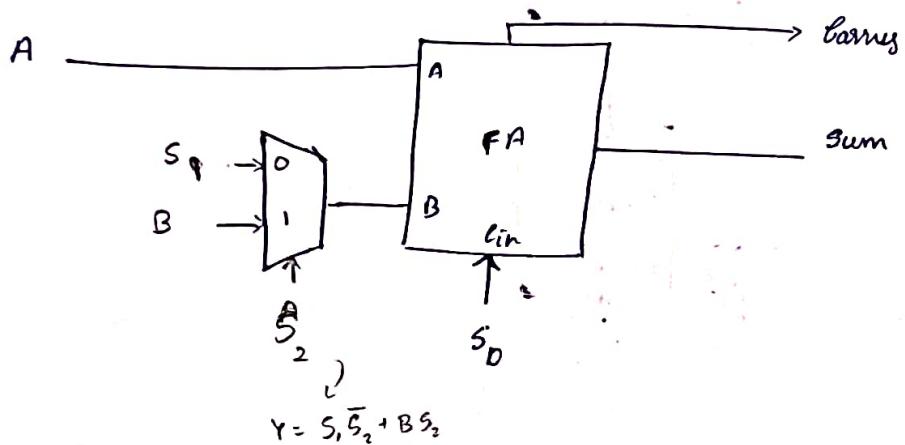
S_2	S_1	S_0	X	Y	Cin		S_2	S_1	S_0	
0	0	0	A	0	0		0	0	0	0
0	0	1	A	B	0		1	1	0	6
0	1	0	A	B	1		1	1	1	7
0	1	1	A	\bar{B}	1		1	0	1	5
1	0	0	A	\bar{B}	0		1	0	0	4
1	0	1	A	0	1		0	0	1	1
1	1	0	A	1	0		0	1	0	2

Repeat of opcode

$$\leftarrow \begin{array}{r} 0 \\ 1 \\ 1 \\ -3 \end{array}$$

Transfer A

Same as 0

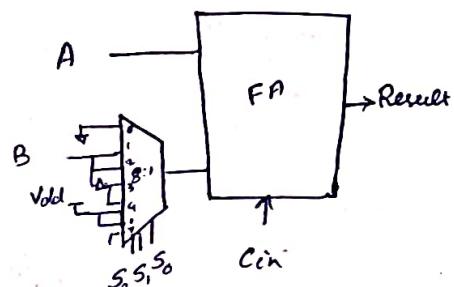


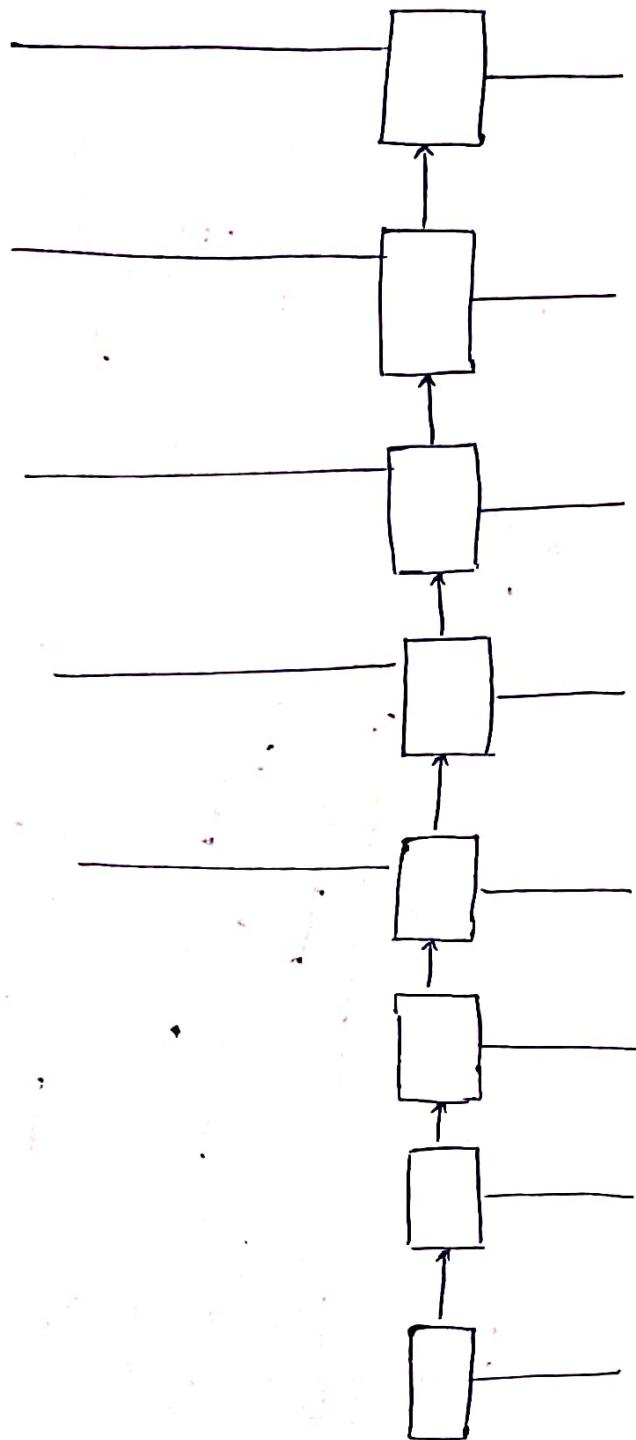
Issues in logic design without optimisation

1) 8 → 8:1 MUX × 2) with out optimisation

we need additional combo logic to control the carry i/p

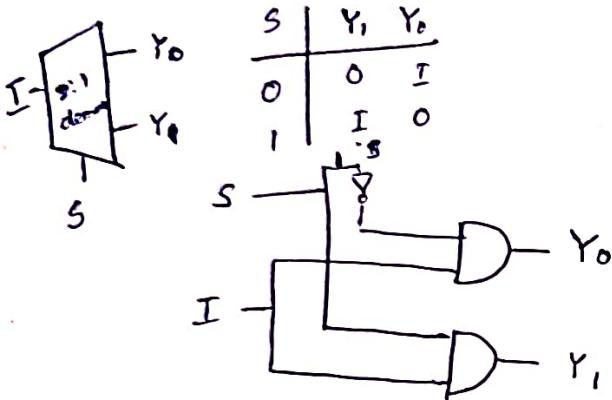
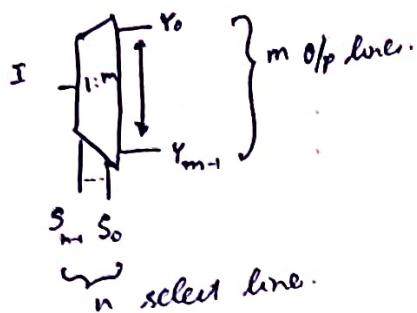
⇒ By re-arranging opcode we can optimize the carry-in



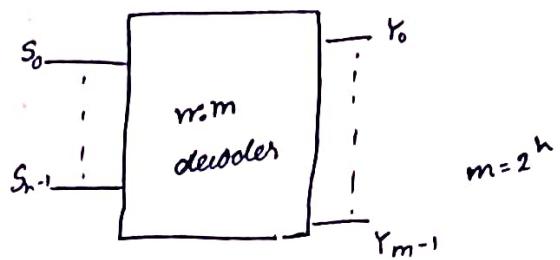


Evening session

De-MUX



Decoder

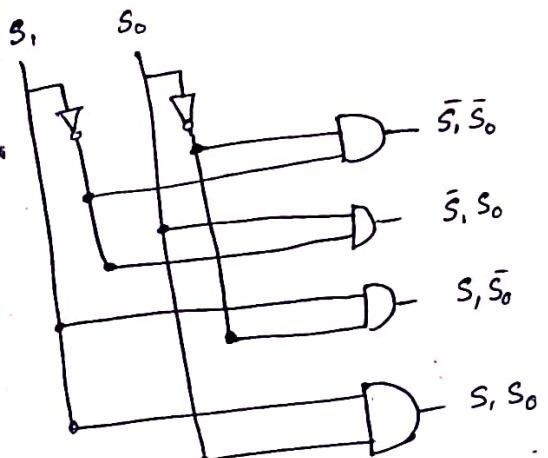


S_1, S_0	Y_3	Y_2	Y_1	Y_0
0 0	0	0	0	1
0 1	0	0	1	0
1 0	0	1	0	0
1 1	1	0	0	0

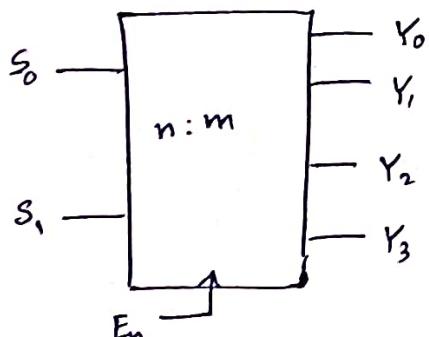
\bar{S}_1, \bar{S}_0

S_1, S_0

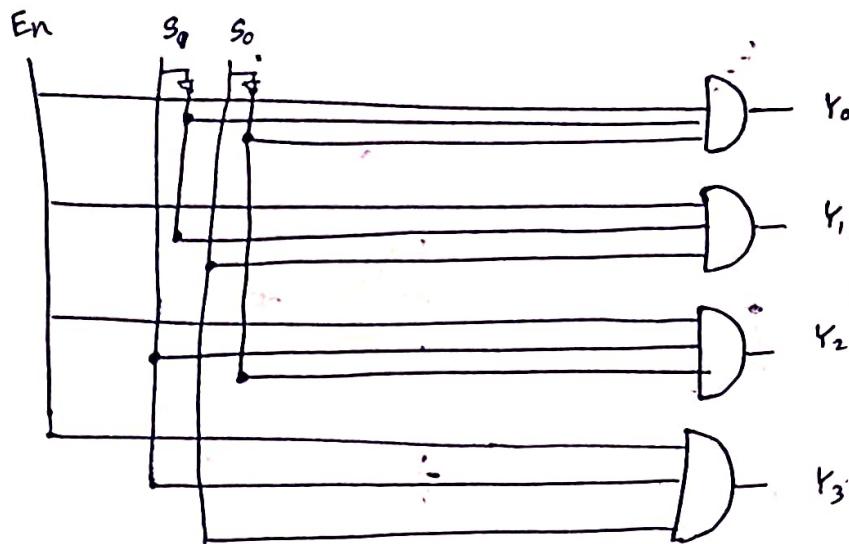
S, S_0



2:4 decoder with active high enable

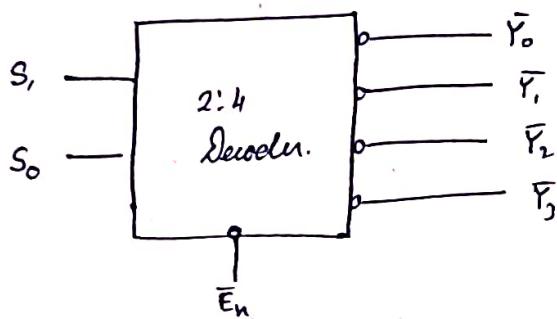


E_n, S_1, S_0	Y_3	Y_2	Y_1	Y_0	$E_n \bar{S}_1, \bar{S}_0$
1 0 0	0	0	0	1	$E_n \bar{S}_1, \bar{S}_0$
1 0 1	0	0	1	0	$E_n \bar{S}_1, S_0$
1 1 0	0	1	0	0	$E_n S_1, \bar{S}_0$
1 1 1	1	0	0	0	$E_n S_1, S_0$
0 x x	0	0	0	0	$E_n 0$

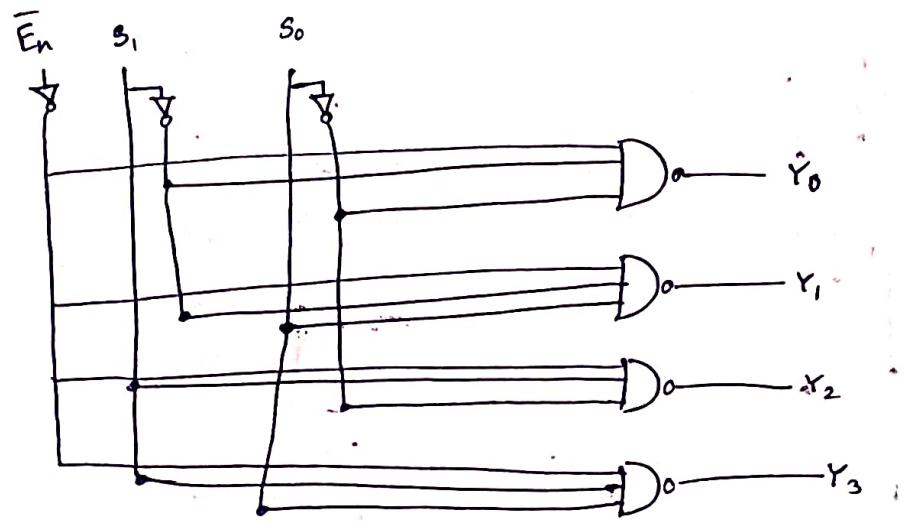


Exercise 19

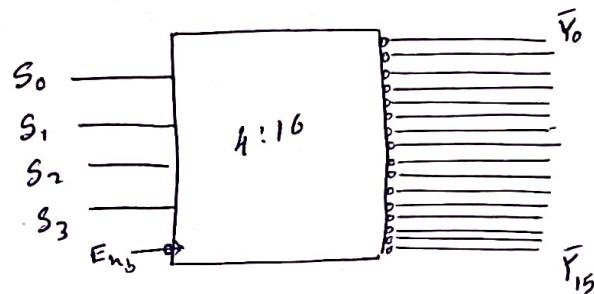
Design 2:4 decoder having active low enable & active low o/p

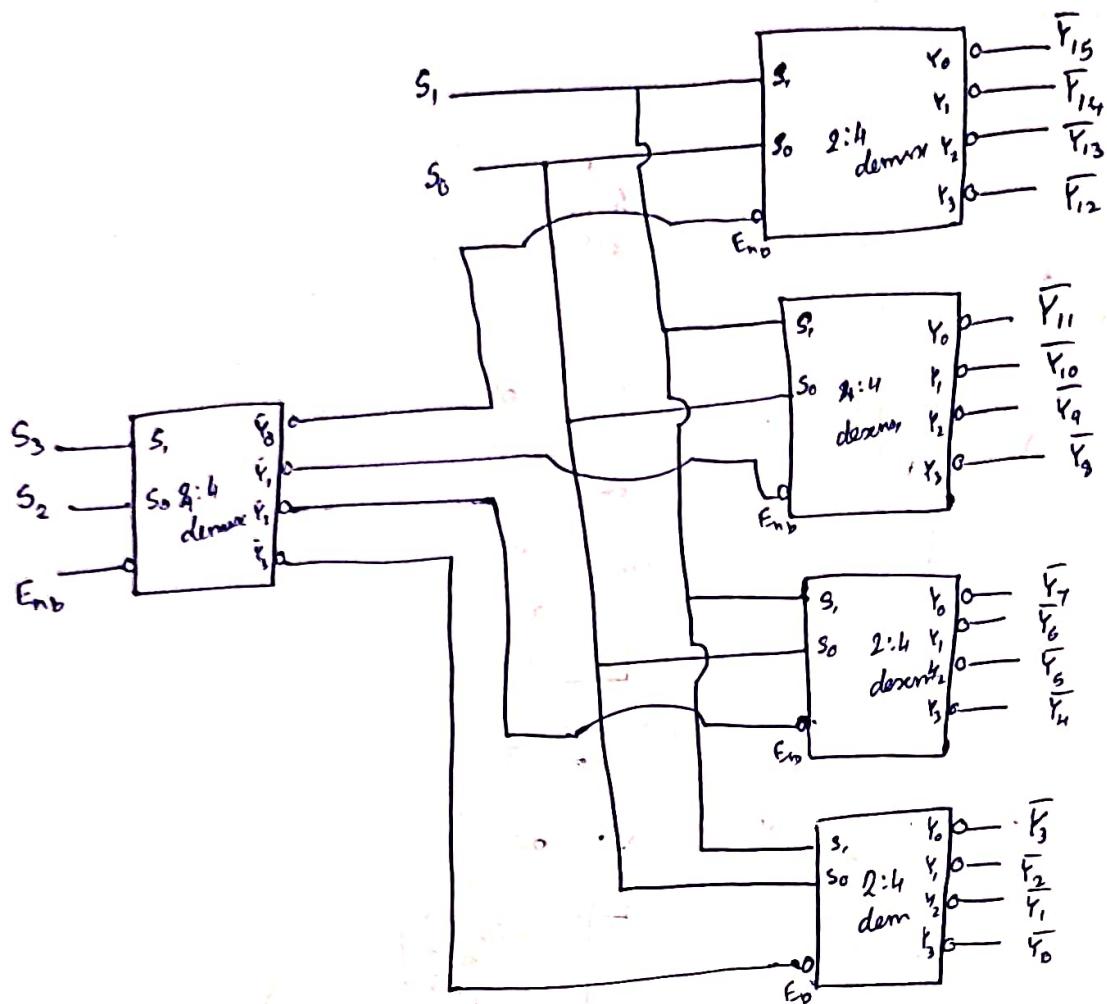


\bar{E}_n	S_1	S_0	\bar{Y}_3	\bar{Y}_2	\bar{Y}_1	\bar{Y}_0	$\bar{E}_n \bar{S}_1 \bar{S}_0$
0	0	0	1	1	1	0	$\bar{E}_n \bar{S}_1 \bar{S}_0$
0	0	1	1	1	0	1	$\bar{E}_n \bar{S}_1 S_0$
0	1	0	1	0	1	1	$\bar{E}_n S_1 \bar{S}_0$
0	1	1	0	1	1	1	$\bar{E}_n S_1 S_0$
1	x	x	1	1	1	1	



Exercise 20
Design / Implement 4:16 decoder using min no of 2:4 decoders.





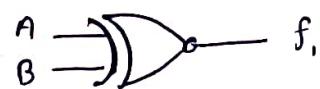
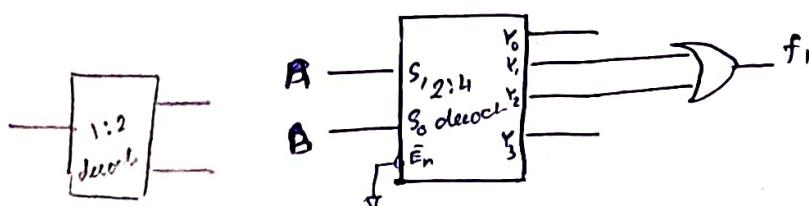
Exercise 21

Implement following function using suitable decoder & additional min no of logic gates

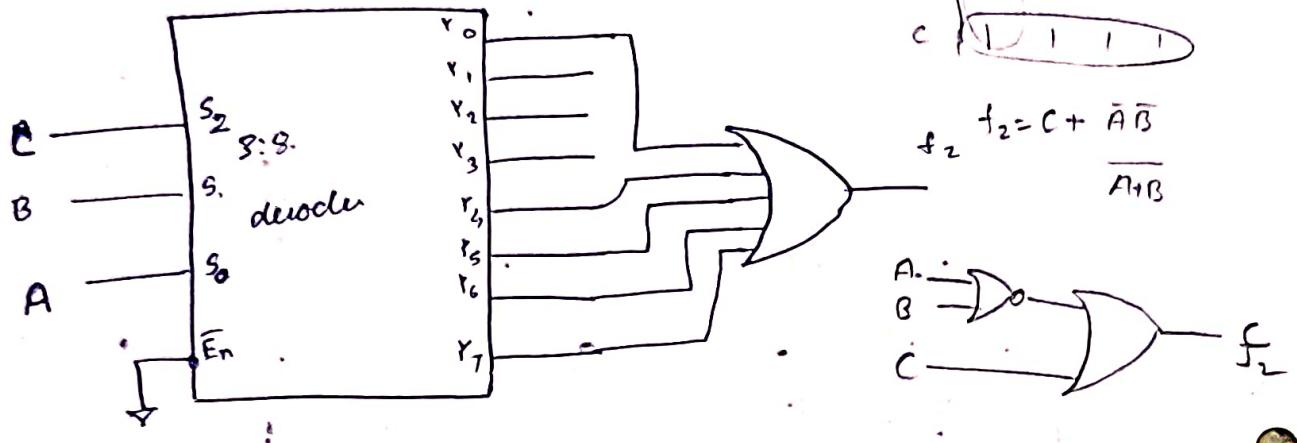
$$a) f_1(A, B) = \Sigma(1, 2)$$

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

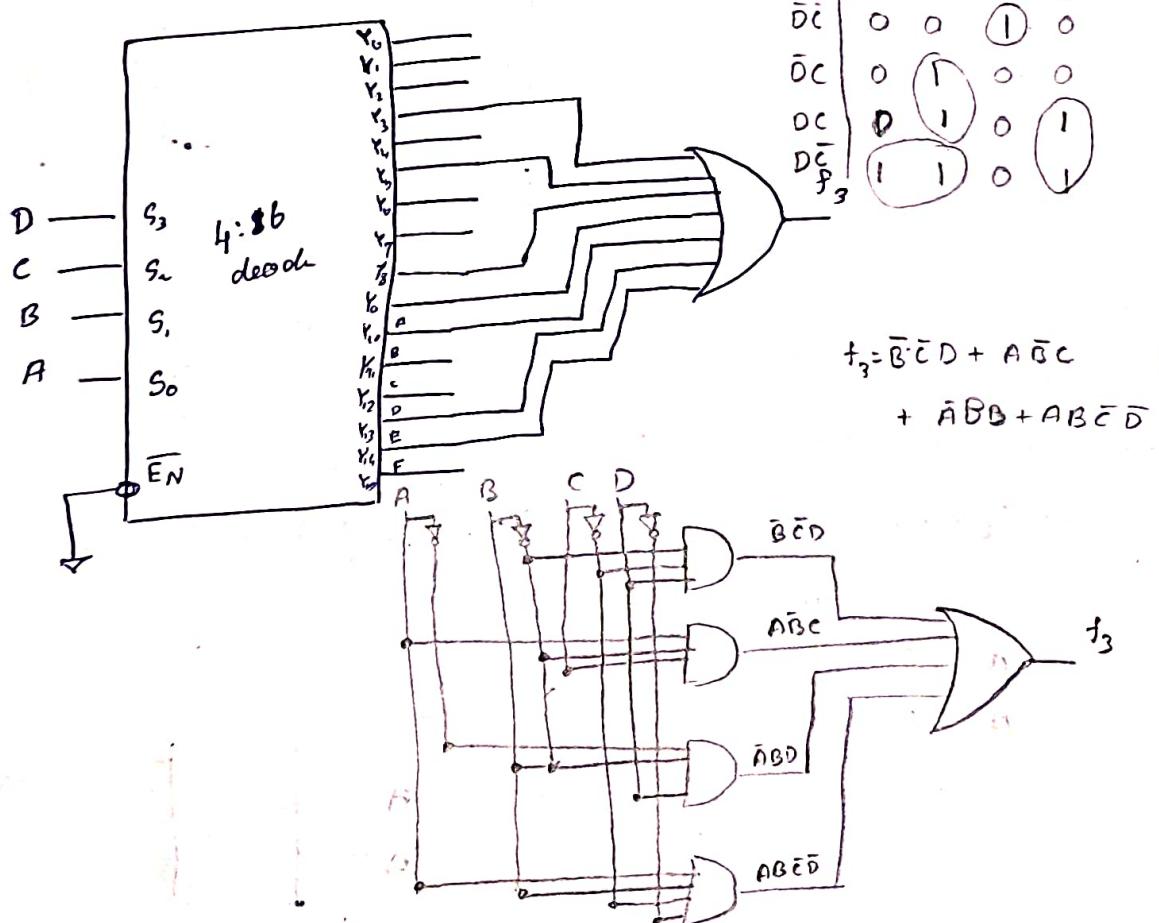
A	B	f_1
0	0	0
0	1	1
1	0	1
1	1	0



b) $f_1(C, B, A) = \Sigma(0, 4, 5, 6, 7)$, $f_3(D, C, B, A) = \Sigma(3, 5, 6, 9, 10, D, E)$



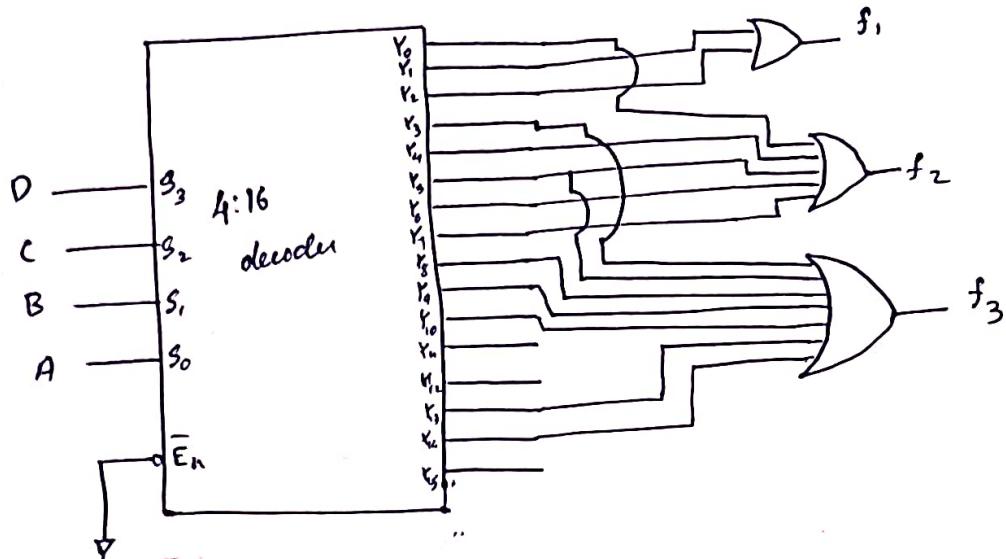
c) $f_3(D, C, B, A) = \Sigma(3, 5, 6, 9, A, D, E)$



06-09-2024

With optimization

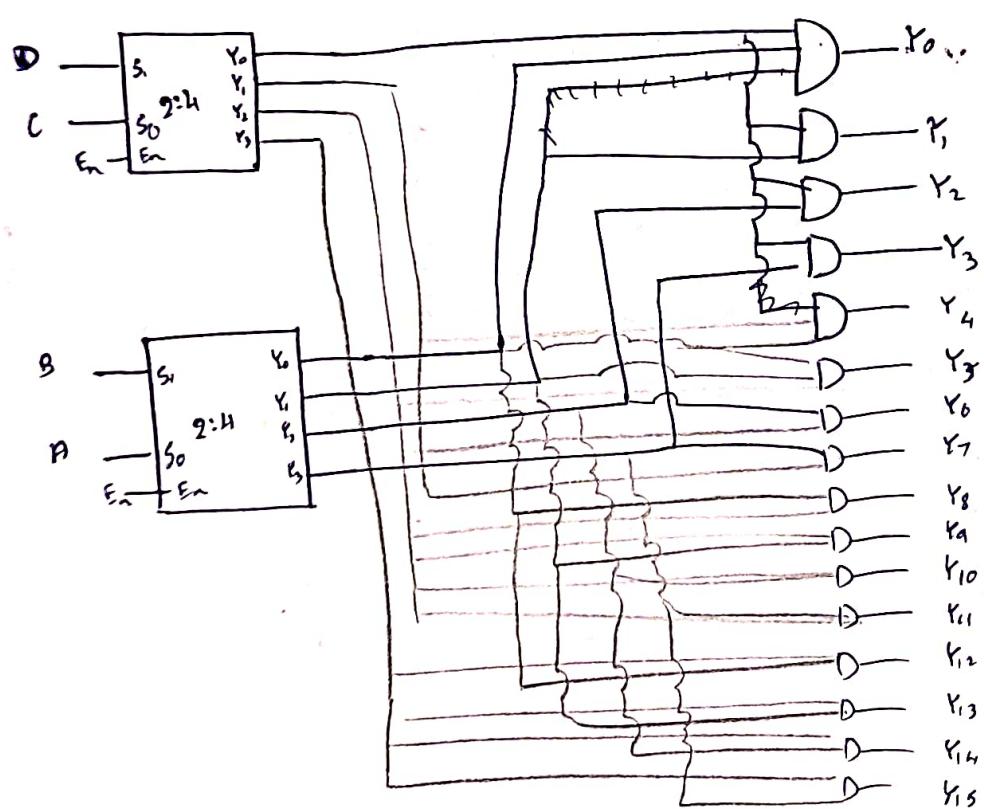
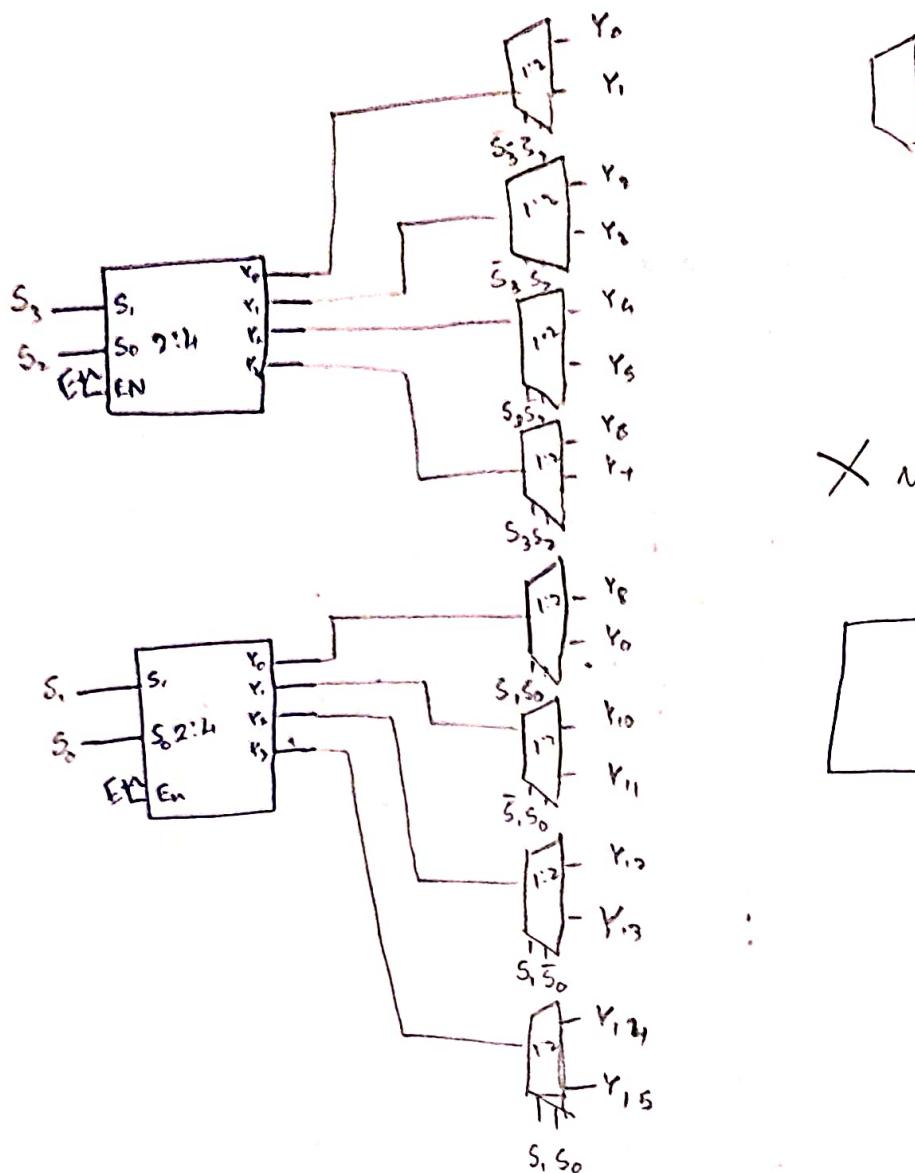
- using single 4:16 decoder for all f_1, f_2, f_3



Exercise 22

Using ~~one~~ only 2:4 decoder & additional min logic gate implement 4:16 decoder. Consider decoder has active high enable & active high logic lines.

E_n	S_3	S_2	S_1	S_0	Y_{15}	Y_{14}	Y_{13}	Y_{12}	Y_{11}	Y_{10}	Y_9	Y_8	Y_7	Y_6	Y_5	Y_4	Y_3	Y_2	Y_1	Y_0
0	x	x	x	x	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
1	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	1	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0



Exercise 23

Design the combinational logic to multiply multiply no by 2

& divide no by 2

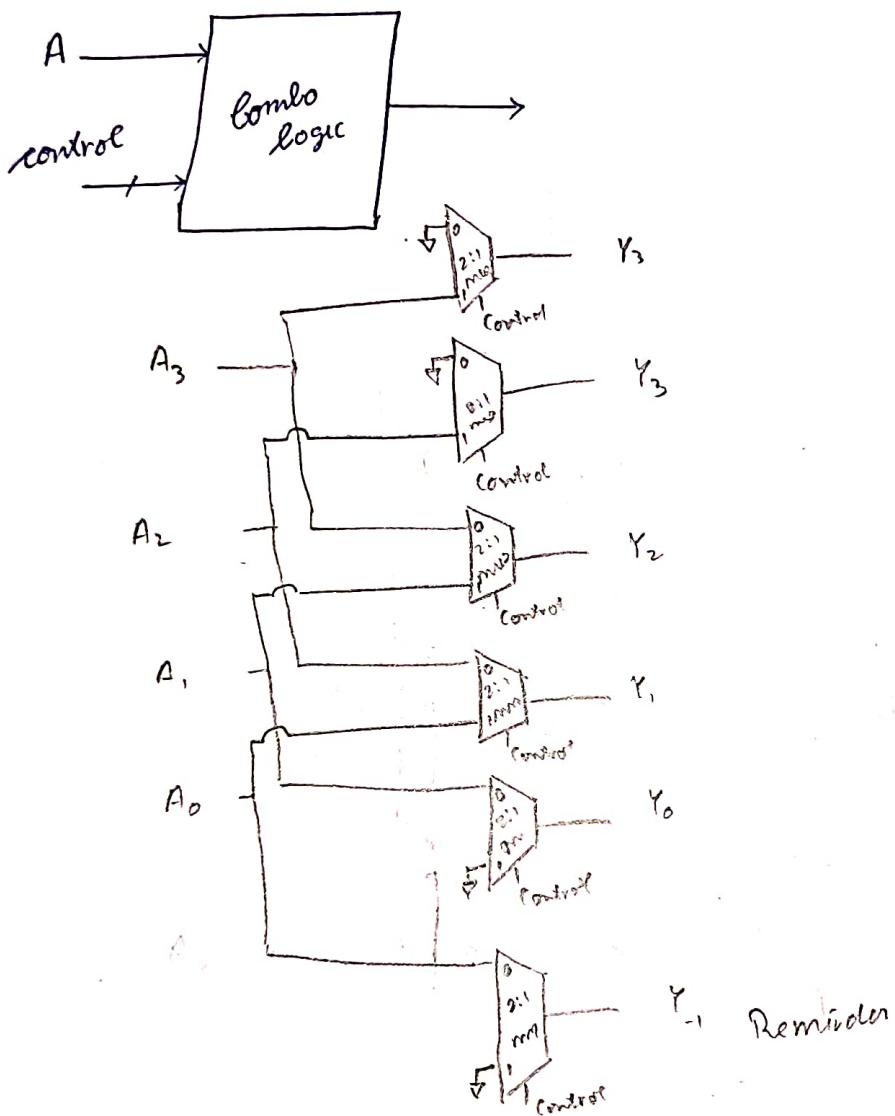
$$1) \begin{array}{r} \text{4 bit binary no} \\ \times 2 \end{array}$$

$$2) \begin{array}{r} \text{4 bit binary no} \\ \div 2 \end{array}$$

left shift 1-bit

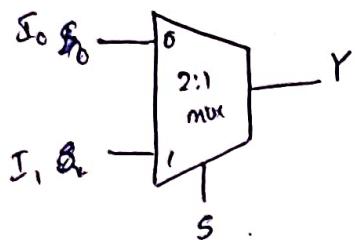
right shift 1-bit

Control	Operation
0	multiply $A * 2$
1	divide $A/2$



Exercise 24

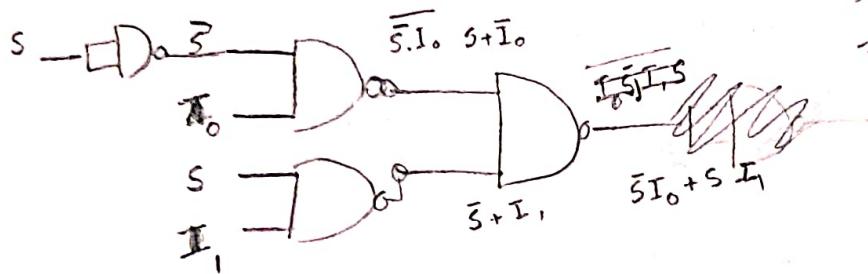
Implement 2:1 mux using min no of NAND gate



$$Y = I_0 \bar{S} + I_1 S$$

$$= \overline{S + \bar{I}_0} + \overline{\bar{S} + I_1}$$

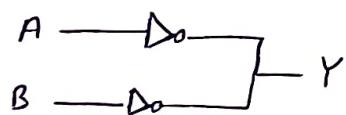
$$= \bar{S} \bar{I}_0 + S I_1$$



Exercise 25

In standard TTL logic if o/p of two NOT gates are shorted

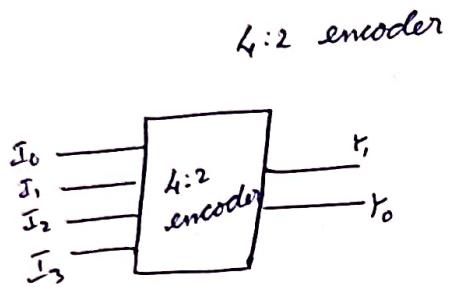
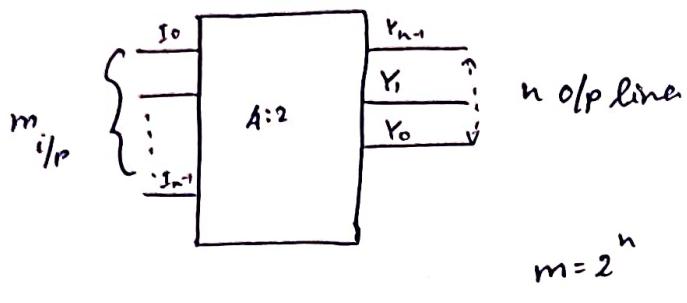
Find \$Y\$



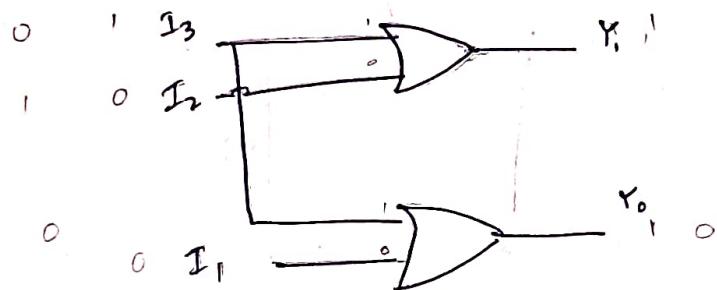
A	B	Y
0	0	1
0	1	0
1	0	0
1	1	0

\$\Rightarrow\$ equivalent NOR

Evening session
Encoder



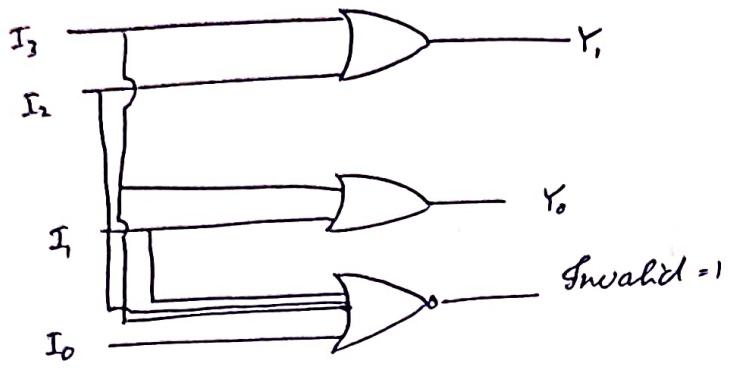
I_3	I_2	I_1, I_0	Y_1	Y_0	I_3, I_2	I_1, I_0	Y_1	Y_0	I_3, I_2	I_1, I_0	Y_1	Y_0
1	0	0 0	0	1 0 1	\bar{I}_3, \bar{I}_2	x 0 x 0			\bar{I}_3, \bar{I}_2	\bar{I}_1, \bar{I}_0	x 0 x 0	
0	1	0 0	0	1 0	\bar{I}_3, I_2	1 x x x			\bar{I}_3, I_2	\bar{I}_1, I_0	x x x x	
0	0	1 0	0	0 1	I_3, \bar{I}_2	x x x x			I_3, \bar{I}_2	I_1, \bar{I}_0	1 x x x	
0	0	0 1	1	0 0	I_3, I_2	1 x x x			I_3, I_2	I_1, I_0	x x x x	



When $I_3, I_2, I_1, I_0 = 0001$ (or) 0000 then

$Y_1, Y_0 = 00$. So, If the encoder o/p connected to i/p of another digital set then it become difficult to understand $Y_1, Y_0 = 00$ due to encoder i/p 0001 (or) 0000

↳ Inefficien 4:2 encoder design because I_0 not used in design



← This is better design

So, to modify our design include assumed that only one invalid o/p in the design to indicate i/p is high at a time all i/p's are design due to that $Y_1, Y_2 = 00$

Exercise 27

Design 4:2 priority encoder using min no of logic gates

I_3 has highest priority

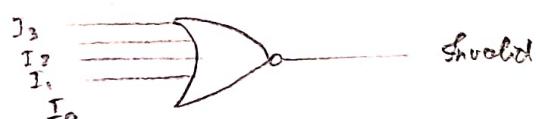
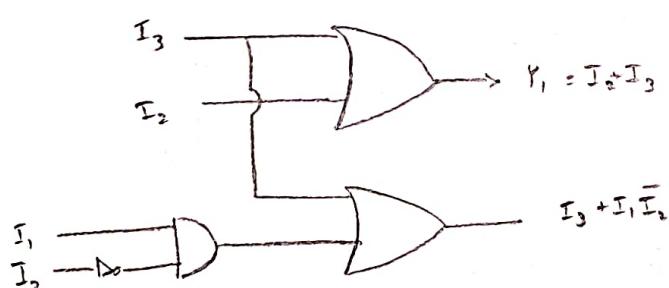
I_3	I_2	I_1	I_0	Y_1	Y_0	Invalid
1	x	x	x	11	0	0
0	1	x	x	10	0	0
0	0	1	x	01	0	0
0	0	0	1	00	0	0
0	0	0	0	00	0	1

$I_1 \bar{I}_0$	$\bar{I}_1 I_0$	$\bar{I}_1 \bar{I}_0$	$I_1 I_0$
0	0	0	0
1	1	1	1
1	1	1	1
1	1	1	1

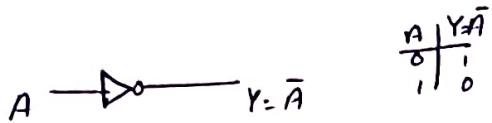
$$Y_1 = I_2 + I_3$$

$\bar{I}_1 \bar{I}_0$	$\bar{I}_1 I_0$	$I_1 \bar{I}_0$	$I_1 I_0$
0	0	1	1
0	0	0	0
1	1	1	1
1	1	1	1

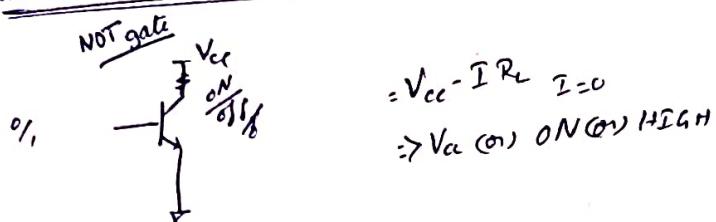
$$Y_0 = I_3 + I_1 \bar{I}_2$$



Introduction to logic family



Transistor Transistor logic

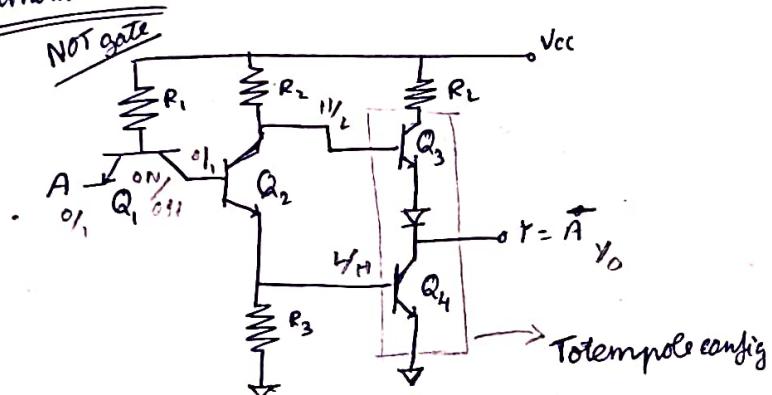


Fan-out, Max drive current

↳ Sourcing capacity

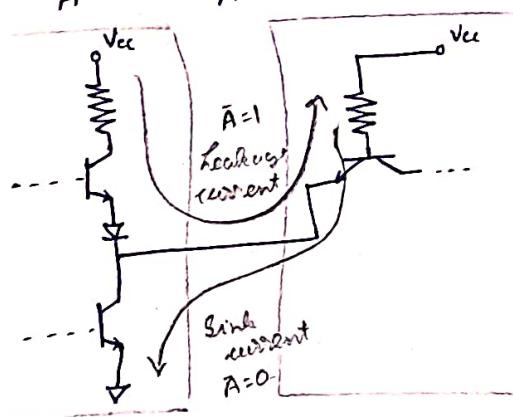
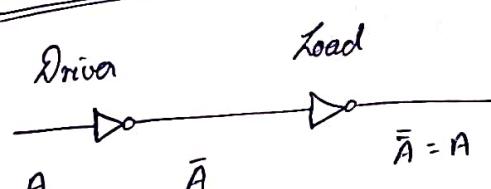
Sinking capacity

standard TTL



- Voltage profile
- Current profile
- Fan-in, Fan-out
- Propagation delay
- Power dissipation
- Compactability

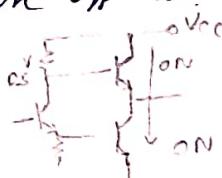
Basic Fundamentals



↳ Why protection diode ??

- For NOT o/p $A=1$ then it is expected

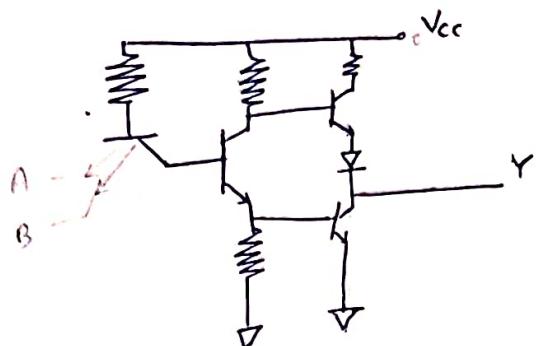
Q_1 is ON \Rightarrow its collector potential low
 $\&$ emitter potential should be high
 \Rightarrow practically low potential can change in the range of 0.8V to 1.1V $\&$ if diode not used it will turn on both transistor $Q_3 \& Q_4$ due to max current it will burn off as shown



07-09-2024

Exercise 28

Draw the internal structure of 2^{1/p} NAND gate for TTL family

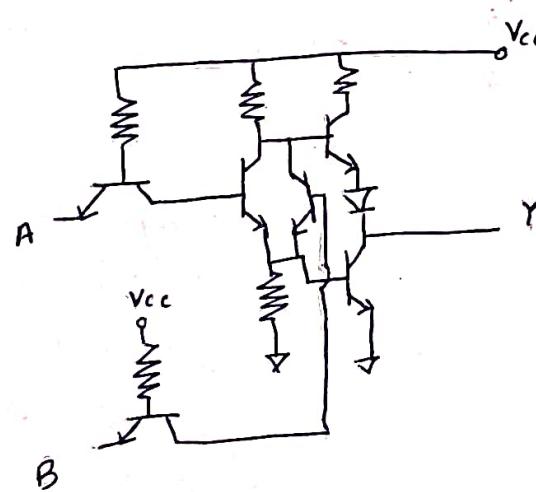


A	B	$Y = \overline{A \cdot B}$
0	0	1
0	1	1
1	0	1
1	1	0

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

Exercise 29

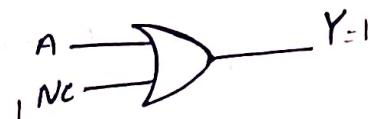
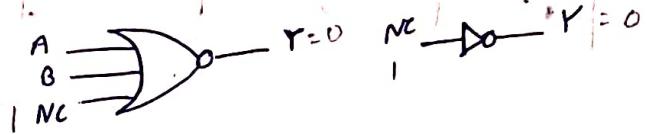
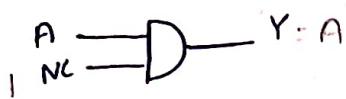
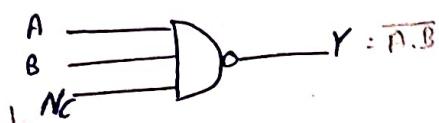
Draw the internal structure of 2^{1/p} NOR gate for TTL family



A	B	$Y = \overline{A + B}$
0	0	1
0	1	0
1	0	0
1	1	0

Standard TTL

one of i/p of gates are not connected as shown in fig
what is the o/p of standard TTL



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

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

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

Don't keep any un-used i/p unconnected

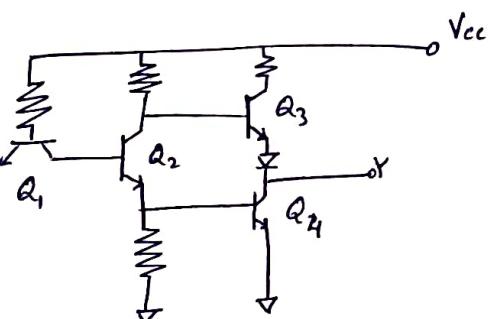
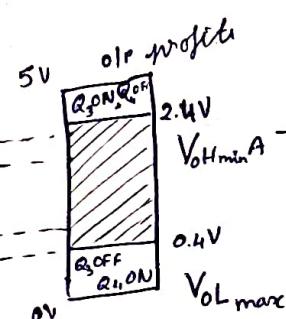
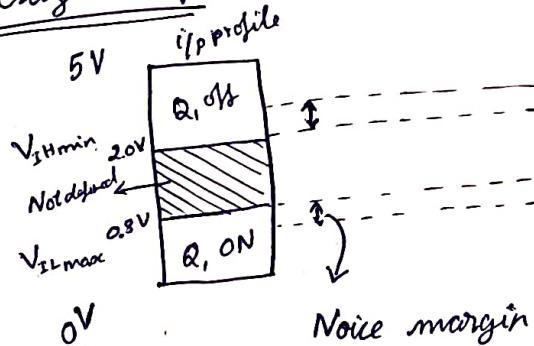
- un-used i/p's should be connected to the required logic

levels, either to logic LOW or HIGH.

- in case of NAND, AND un-used i/p's (or ~~not~~) should be connected to logic HIGH

- un-used i/p's of NOR, OR should be connected to logic LOW

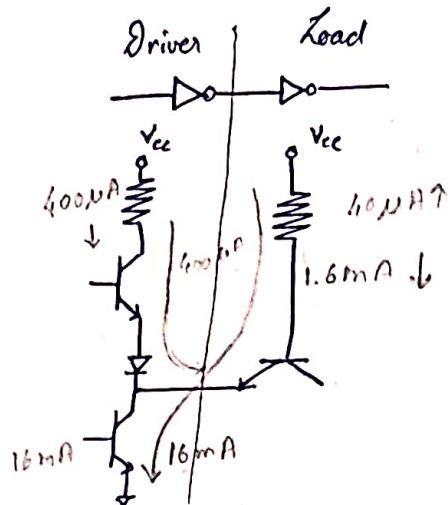
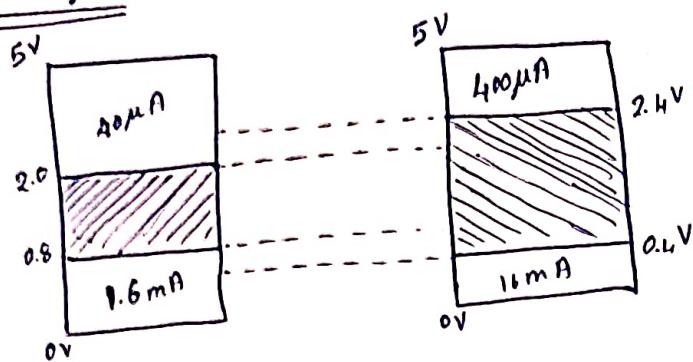
Voltage Profile



Noise margin

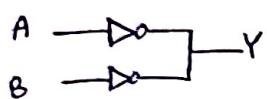
NOTE: for +3.3V, +1.8V the concept remains same but there will be change in $V_{I\max}$, $V_{I\min}$, $V_{O\max}$, $V_{O\min}$

Current Profile

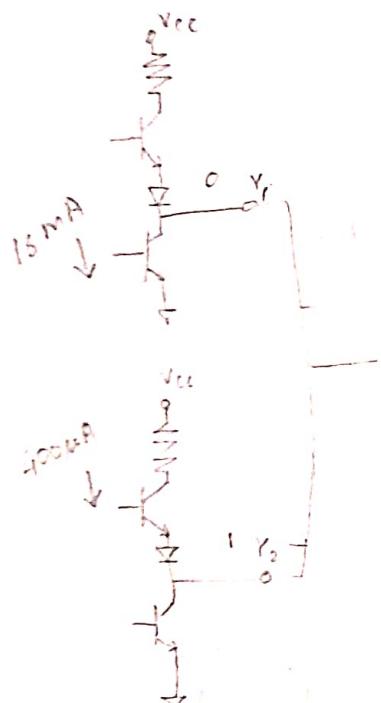


Fan-out is max no. of load driver by driver satisfactorily

Standard TTL two NOT gates are shorted



		$Y = \overline{A+B} \rightarrow \text{NOR}$
A	B	Y
0	0	1
0	1	0
1	0	0
1	1	0



Don't short o/p of two standard TTL logic gates

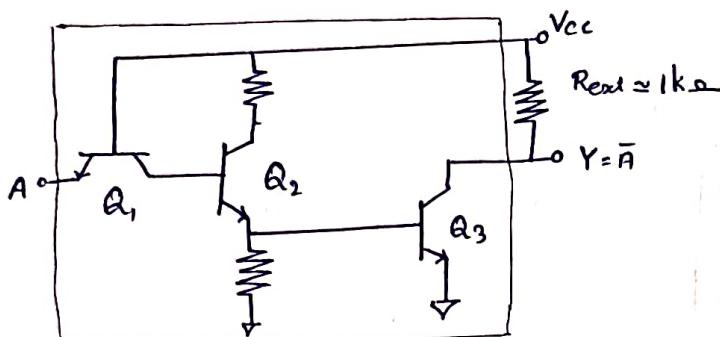
Power dissipation

$$P = I_o^2 R_L$$

MAX power dissipated by logic gated

Open collector TTL

• Collector is kept open therefore external resistor is required to pull collector to +5V which can pull collector Voltage to $+V_{cc}$.
Here we can short o/p of two NOT gates using external resistor.
• following are 9's

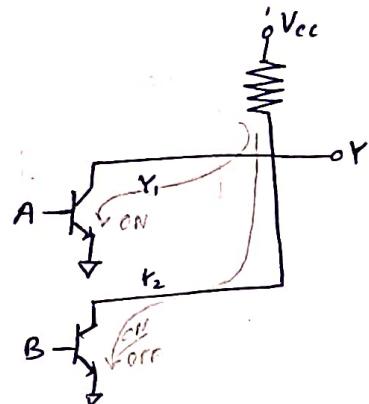
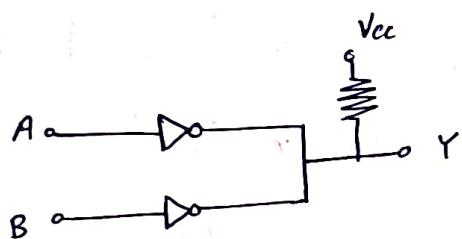


Advantage

- power decapsulation high
-

Disadvantage

- propagation delay ↑

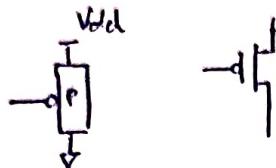


A	B	$Y = \bar{A} + B$ (NOR)
0	0	H
0	1	L
1	0	L
1	1	L

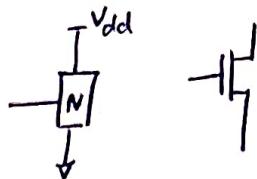
CMOS

Complementary metal oxide semiconductor

PMOS



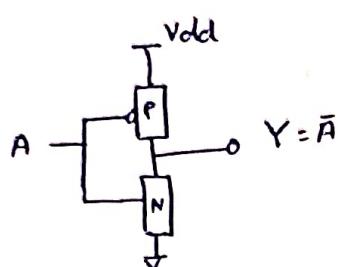
NMOS



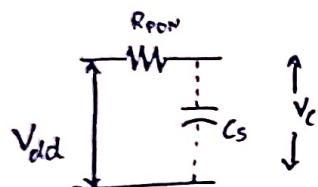
pass strong 1
gate = 0

pass strong 0
gate = 1

CMOS



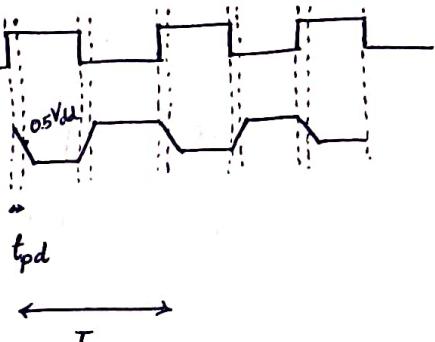
A	Y
0	H
1	L



$$V_c = V(1 - e^{-t/\tau})$$

$$\tau = R_{PON} C_s$$

A Y

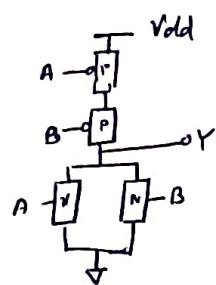
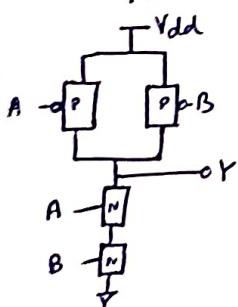


Exercise 30

Sketch 2 input CMOS NAND gate, NOR gate

A	B	Y _i = $(\overline{A} \cdot \overline{B})$
0	0	1
0	1	1
1	0	1
1	1	0

	Y = $(\overline{A} + \overline{B})$
0	1
0	0
1	0
1	0



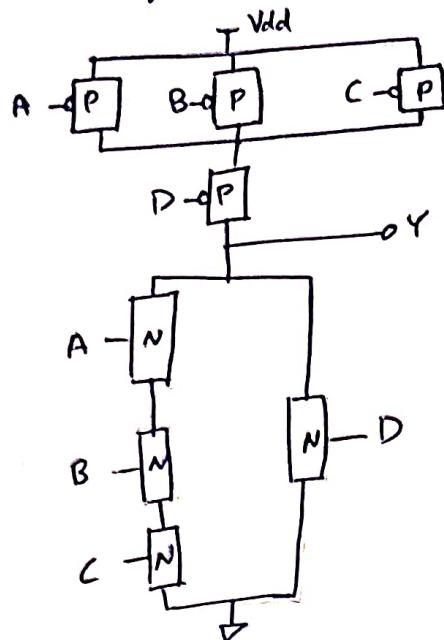
Exercise 31

Implement following using PMOS & NMOS

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

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

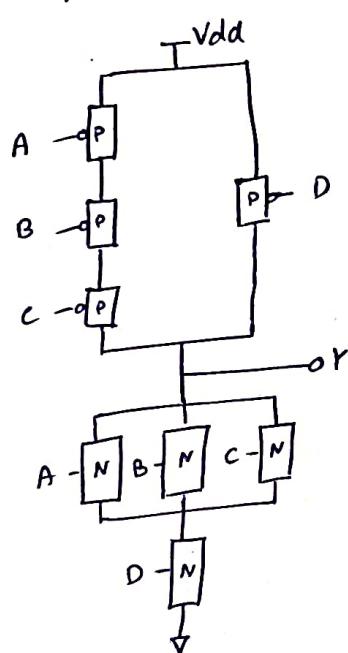
$$= (\overline{A} + \overline{B} + \overline{C}) \cdot \overline{D}$$



Exercise 32

Implement following using PMOS & NMOS

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

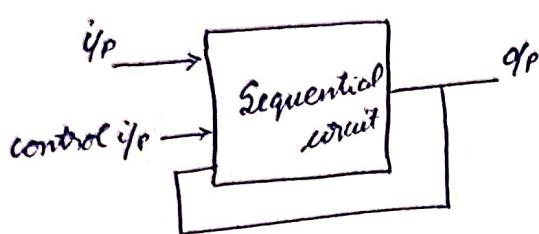


- Hazards, glitch
- Hold time
- ECL & Complementary ^{Further} CMOS
- CMOS D-latch, D-flipflop

09/09/2024

Basis of sequential circuits

$o/p = f_n(\text{present state } i/p \leftarrow \text{past } o/p)$

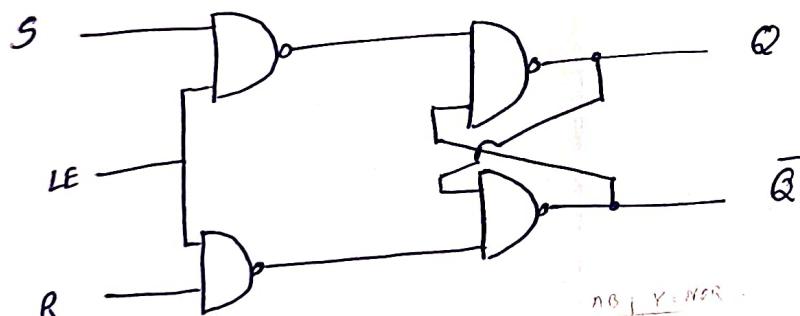
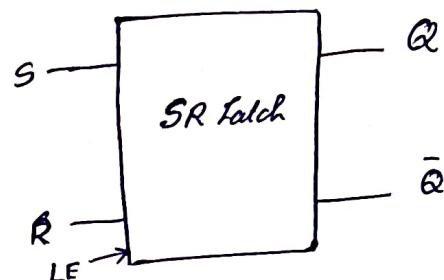


Latches	flip-flop
	Edge sensitive
level sensitive	active HIGH, LOW
	raising edge triggered, falling edge triggered

$\frac{H}{\frac{L}{2}}$
Verilog
nosedge

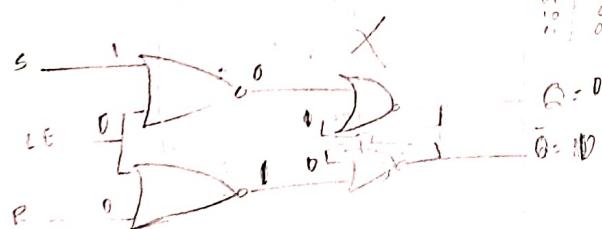
SR Latch

LE	S R		Q	\bar{Q}
	0	0		
1	0	1	0	1
1	1	0	1	0
1	1	1	Indeterminate $\rightarrow Q = \bar{Q}$	
0	x	x	No change	



NB	Y: NOR
00	1
01	0
10	0
11	0

SR using NOR



Level triggered \rightarrow edge triggered

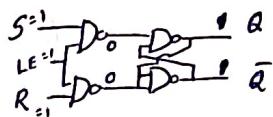
SR using NOR

$D \rightarrow S$

$S, R \rightarrow S, R$

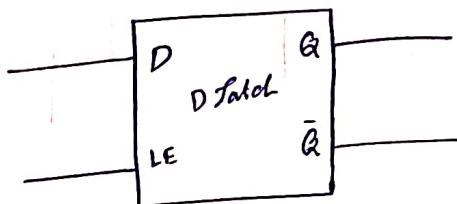
$S, R \rightarrow S, R$

Don't use SR Latch as $S=1, R=1 \wedge LE=1$, both ops Q, \bar{Q} will try to become '1' which is practically not possible



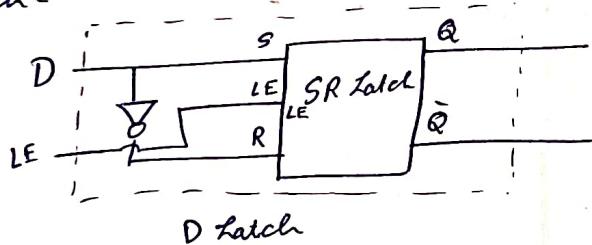
D-Latch - positive level sensitive

LE	D	Q	\bar{Q}
1	0	0	1
1	1	1	0
0	x	No change	



If we have SR Latch which is positive level sensitive & if we use a NOT gate between $S \wedge R$ i/p & data i/p D in given to S i/p as shown \downarrow then we get a positive level sensitive

D latch -



Latch vs Flip flop

• Level sensitive

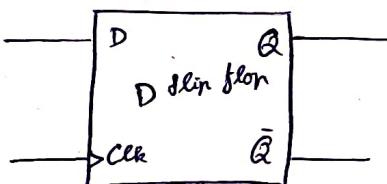
(ie) during active level $Q=D$

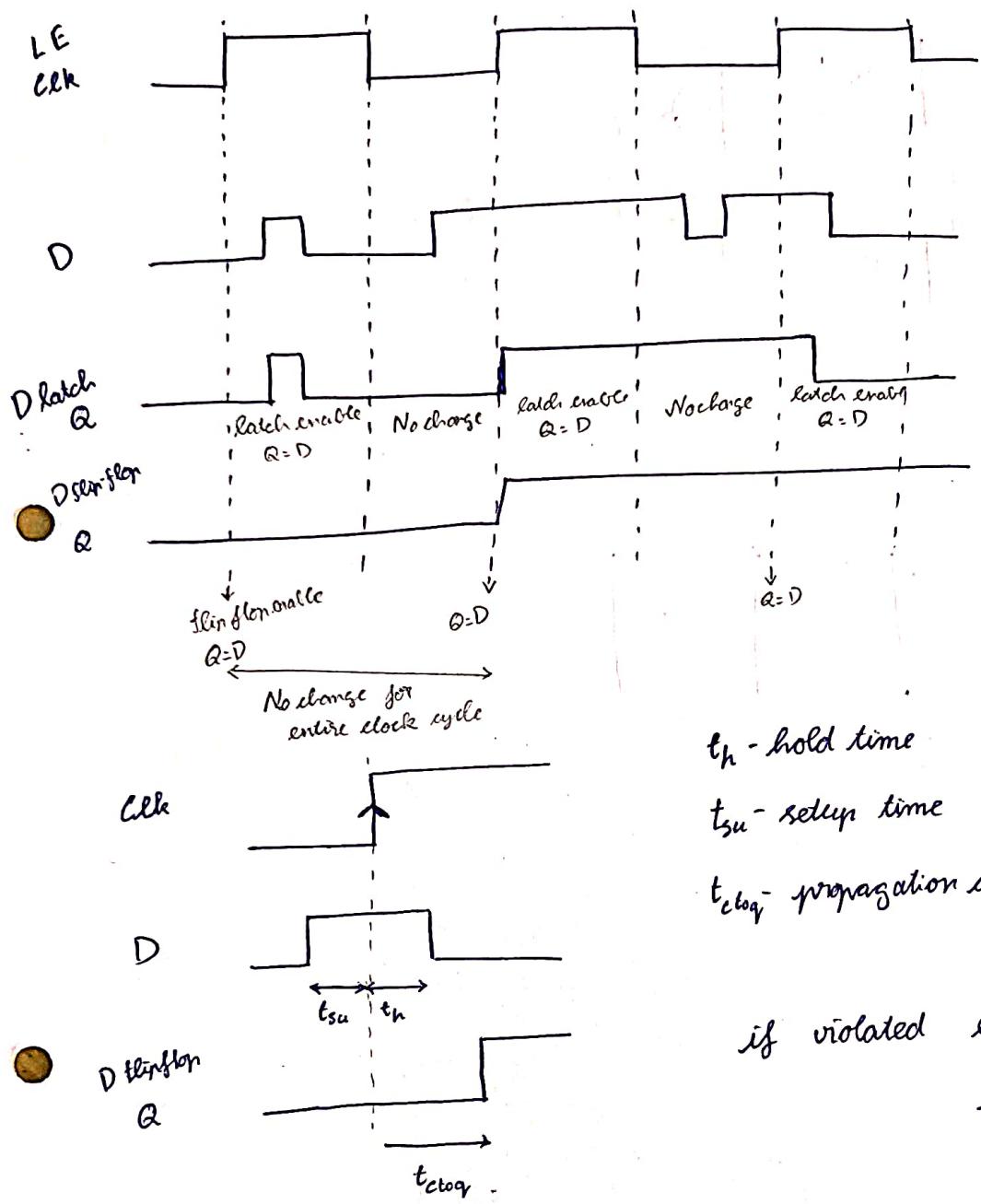
• Edge sensitive

(ie) during active edge $Q=D$

D-Flip-flop

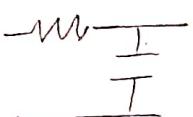
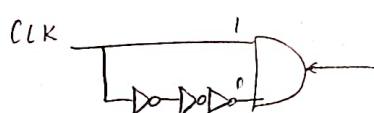
Clk	D	Q	\bar{Q}
↓	0	0	1
↑	1	1	0
→	x	No change	



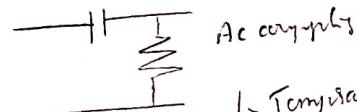


if violated either t_h , t_{su} or t_p will become metastable

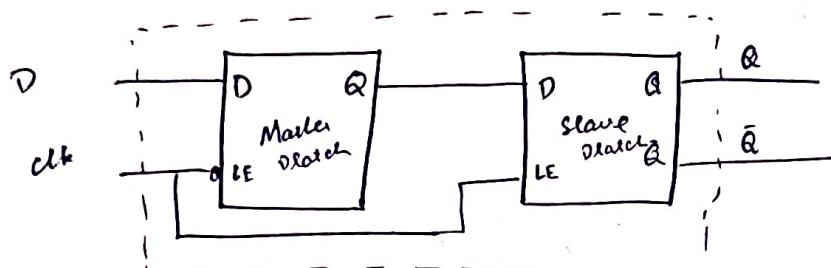
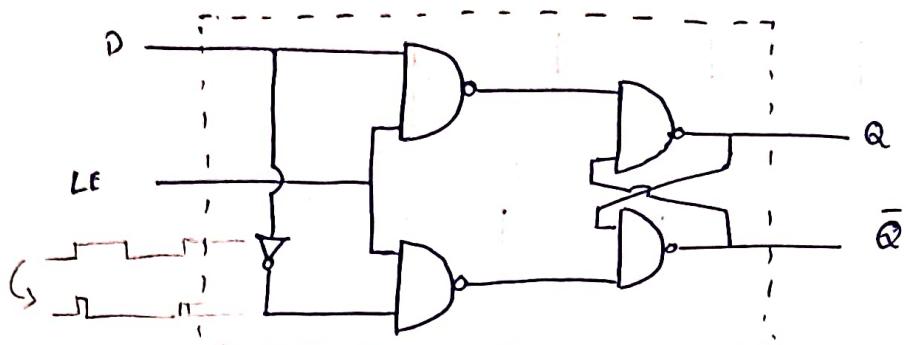
Exercise 33
Sketch the logic circuit of rising edge sensitive flip-flop.



DC-coupling
AC-coupling



Temperature variation effect +
cannot be used for different

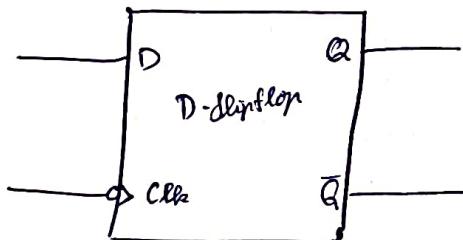


D flip flop

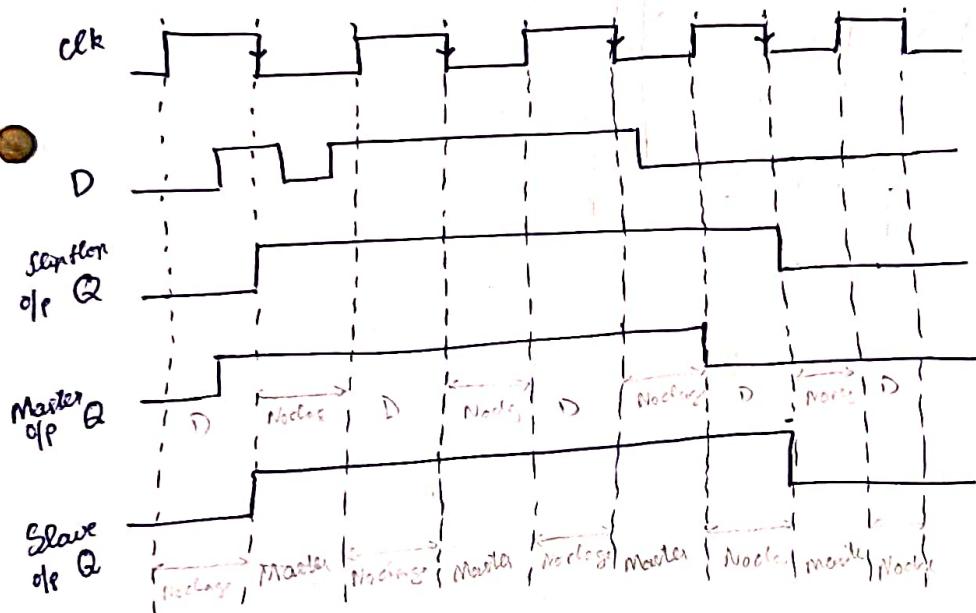
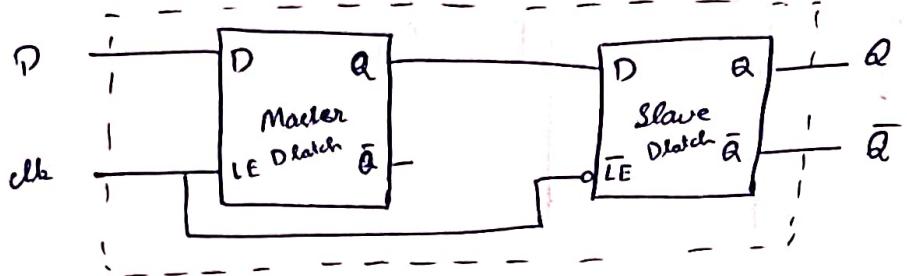
Exercise 3.1

Sketch the logic circuit of negative edge sensitive D flip flop

& complete the timing diagram

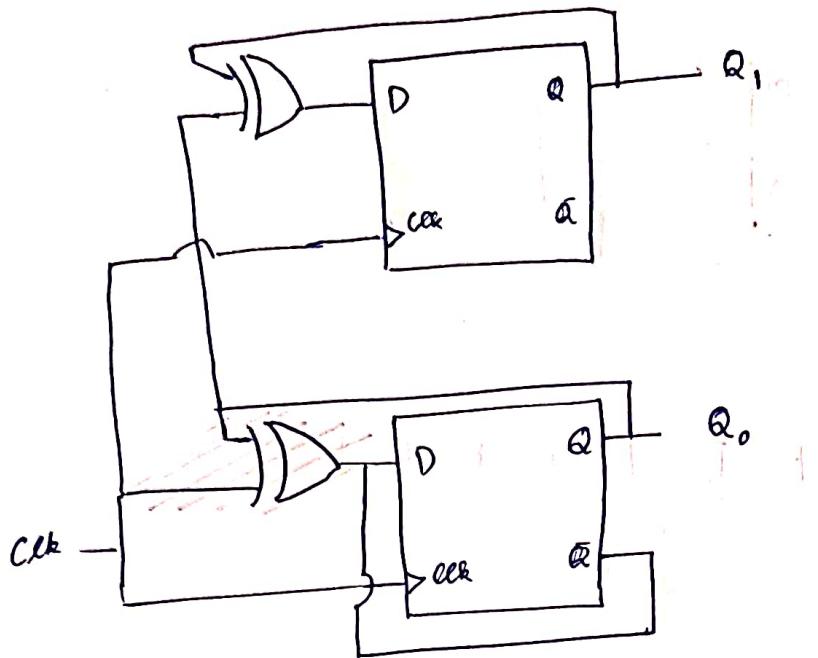


clk	D	Q	Q̄
↑	0	0	1
↓	1	1	0
↑	x		No change



Exercise 35
 Applications of D flip flop. \rightarrow Using min no. of raising edge sensitive
 D flip flop & min-no of additional logic gate design sequential
 circuit to get o/p

	Q, Q_0
0	0 0
0	0 1
1	1 0
1	1 1



D 4 states $= 2^2 \Rightarrow$ number of stages.

No of shift-flip

2 - D flip flop.

2) State Table

Present state		Next state	
Q_1	Q_0	Q_1'	Q_0'
0	0	0	1
0	1	1	0
1	0	1	1
1	1	0	0

3) Excitation table

Present state		Next state		Excitation I/P	
Q_1	Q_0	Q_1'	Q_0'	D_1	D_0
0	0	0	1	0	1
0	1	1	0	1	0
1	0	1	1	1	1
1	1	0	0	0	0

4) Deduce the next state logic

$$O/P = f(PS, i/P)$$

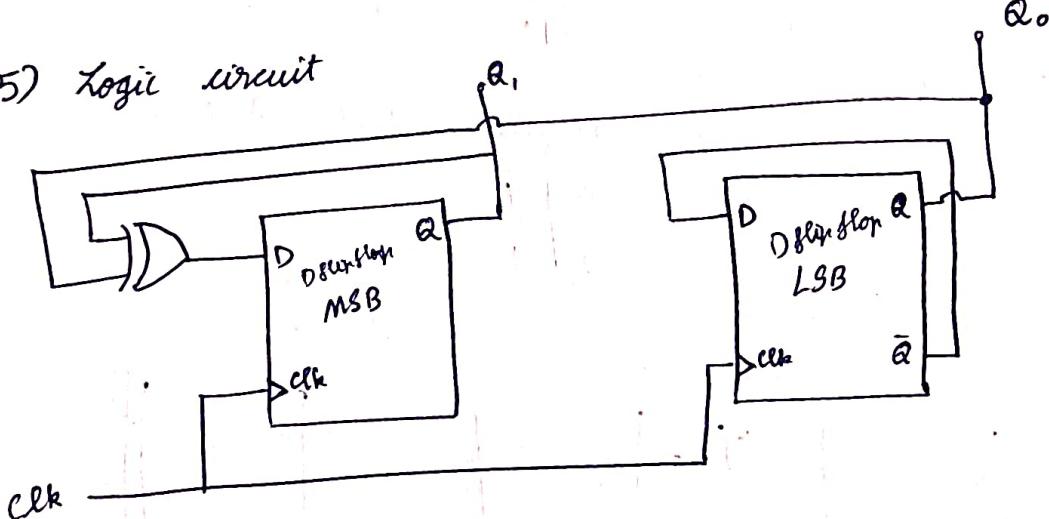
$$\begin{array}{c} D_1 \quad \bar{Q}_0 \quad Q_0 \\ \hline \bar{Q}_1 & 0 & 1 \\ Q_1 & 1 & 0 \end{array}$$

$$\begin{array}{c} D_0 \quad \bar{Q}_0 \quad Q_0 \\ \hline \bar{Q}_1 & | & 0 \\ Q_1 & | & 0 \end{array}$$

$$D_0 = \bar{Q}_1$$

$$D_1 = Q_1 \oplus Q_0$$

5) Logic circuit



Synchronous sequential circuit

↳ all circuits will receive clock from same clock source or common clock source.

Exercise 36

Design 3bit synchronous binary up counter using rising edge sensitive D flip flop & min no of additional logic gates

$$1) \text{ 8 states} = 2^3$$

3 Flip flop

2) State table

Present state			Next state		
Q_2	Q_1	Q_0	Q_2'	Q_1'	Q_0'
0	0	0	0	0	1
0	0	1	0	1	0
0	1	0	0	1	1
0	1	1	1	0	0
1	0	0	01	0	1
1	0	1	1	1	0
1	1	0	1	1	1
1	1	1	0	0	0

Q_2	$\bar{Q}_2\bar{Q}_1$	\bar{Q}_2Q_1	$Q_2\bar{Q}_1$	$Q_2\bar{Q}_1$
\bar{Q}_2	0	0	1	0
Q_2	1	1	0	1

$$D_2 = \bar{Q}_2 Q_2 + Q_2 \bar{Q}_1 + Q_2 Q_1 Q_0 \\ = Q_2 \oplus Q_1 Q_0$$

3) Excitation Table

Present state			Next state			Excitation i/p		
Q_2'	Q_1'	Q_0'	Q_2'	Q_1'	Q_0'	D_2	D_1	D_0
0	0	0	0	0	1	0	0	1
0	0	1	0	1	0	0	1	0
0	1	0	0	1	1	0	1	1
0	1	1	1	0	0	1	0	0
1	0	0	01	0	1	0	0	1
1	0	1	1	1	0	1	1	0
1	1	0	1	1	1	1	1	1
1	1	1	0	0	0	0	0	0

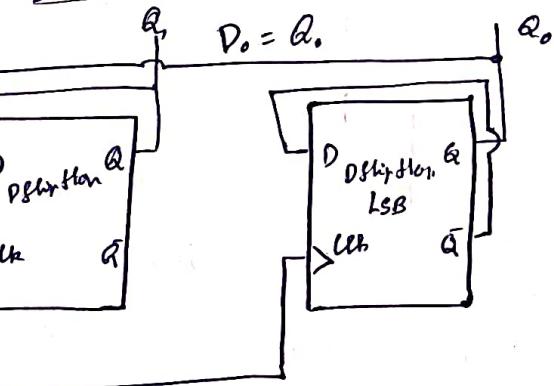
4) Deduce the next state logic

$$Q'_p = f(P_S, i/p)$$

D_1	$\bar{Q}_0\bar{Q}_1$	\bar{Q}_0Q_1	$Q_0\bar{Q}_1$	$Q_0\bar{Q}_1$
\bar{Q}_2	0	1	0	1
Q_2	1	0	1	0

$$D_1 = Q_0 \oplus Q_1$$

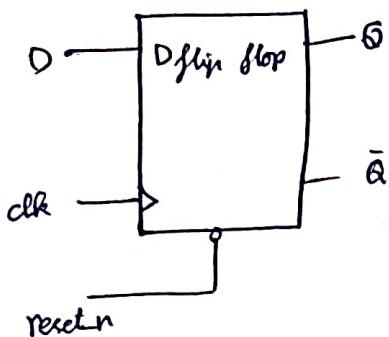
D_0	$\bar{Q}_0\bar{Q}_1$	\bar{Q}_0Q_1	$Q_0\bar{Q}_1$	$Q_0\bar{Q}_1$
\bar{Q}_2	1	0	0	1
Q_2	1	0	0	1



CLK

Asynchronous reset indicates it is independent of clock
when ever resetn is 0 flip flop will be reseted

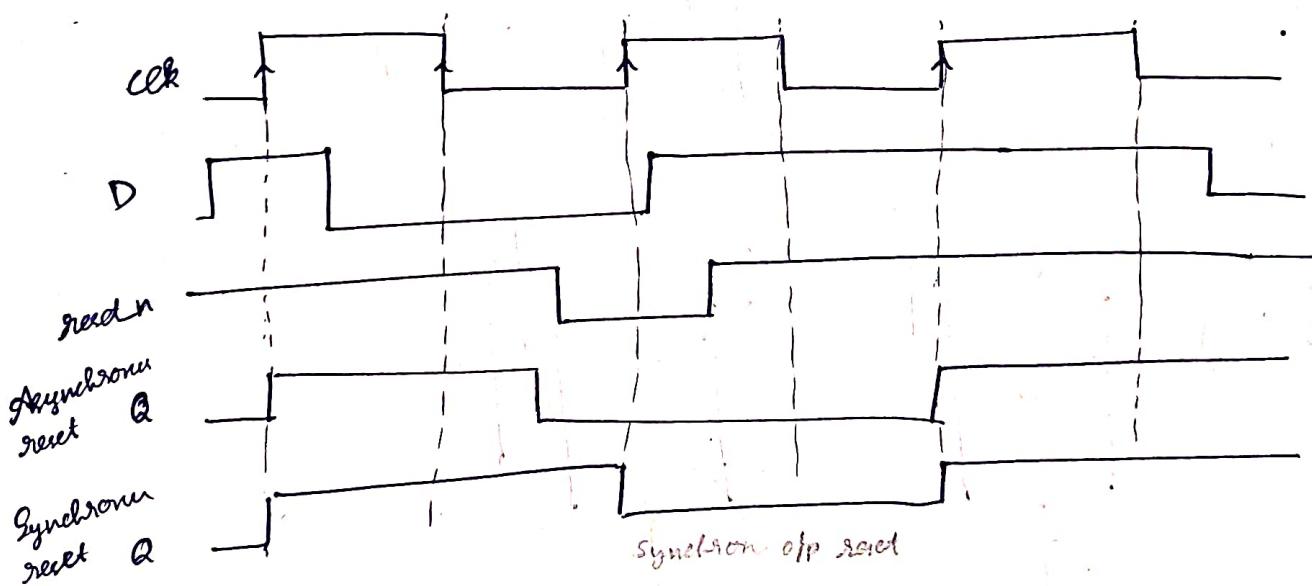
$$Q=0$$



resetn	clk	D	Q	\bar{Q}
0	x	0	0	1
1	↓	x	0	1
1	↓	1	1	0
1	↓	x	No change	

Synchronous reset

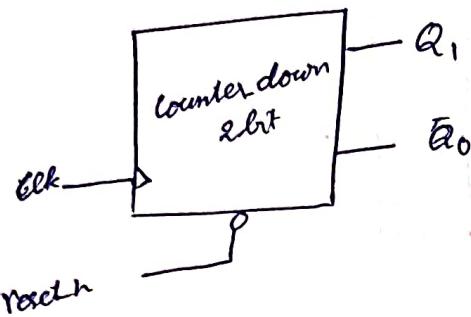
resetn	clk	D	Q	\bar{Q}
0	↑	x	No change	
0	↓	x	0	1
1	↓	0	0	1
1	↓	1	1	0
1	↓	x	No change	



10/09/2024

Exercise 31

Design 2-bit synchronous binary down counter using positive edge triggered D flipflop & min no of logic gate. Consider active low asynchronous reset.



1) 4 states = 2^2
2 flip flops

2) State Table

Present state		Next state	
Q_1	Q_0	Q_1^+	Q_0^+
0	0	1	1
0	1	1	0
1	0	0	1
1	1	0	0

3) Excitation table

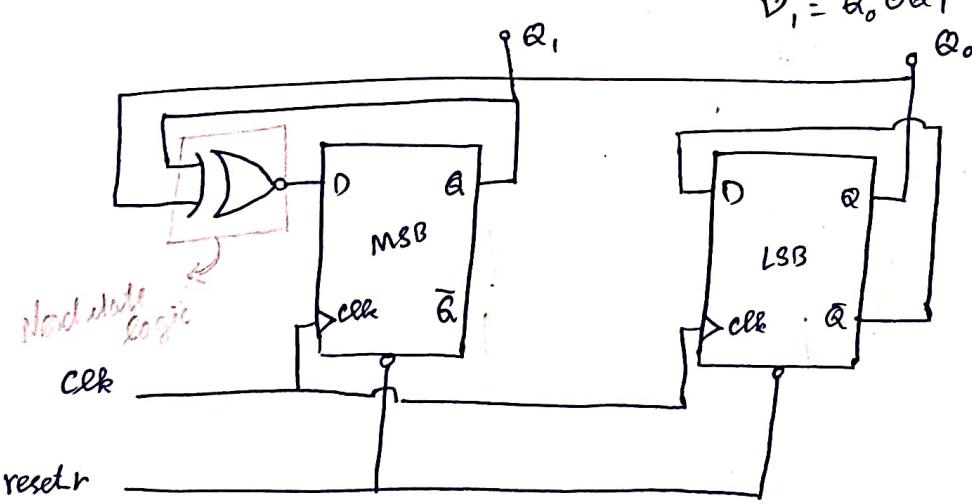
Present state		Next state		Excitation i/p	
Q_1	Q_0	Q_1^+	Q_0^+	D_1	D_0
0	0	1	1	1	1
0	1	1	0	1	0
1	0	0	1	0	1
1	1	0	0	0	0

$$\begin{array}{c} D_1 \\ \bar{Q}_1 \\ Q_1 \end{array} \left| \begin{array}{cc} \bar{Q}_0 & Q_0 \\ \textcircled{1} & 0 \\ 0 & \textcircled{1} \end{array} \right.$$

$$\begin{array}{c} D_0 \\ \bar{Q}_0 \\ Q_0 \end{array} \left| \begin{array}{cc} \bar{Q}_0 & Q_0 \\ 1 & 0 \\ 1 & 0 \end{array} \right.$$

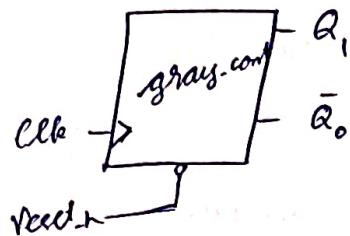
$$D_1 = Q_0 \oplus Q_1$$

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



Exercise 38

Design 2-bit synchronous gray counter



1) 4 states = 2^2
2 Flip Flop.

2) State Table

Present state		Next state	
Q_1	Q_0	Q_1^+	Q_0^+
0	0	0	1
0	1	1	1
1	1	1	0
1	0	0	0

3) Excitation Table

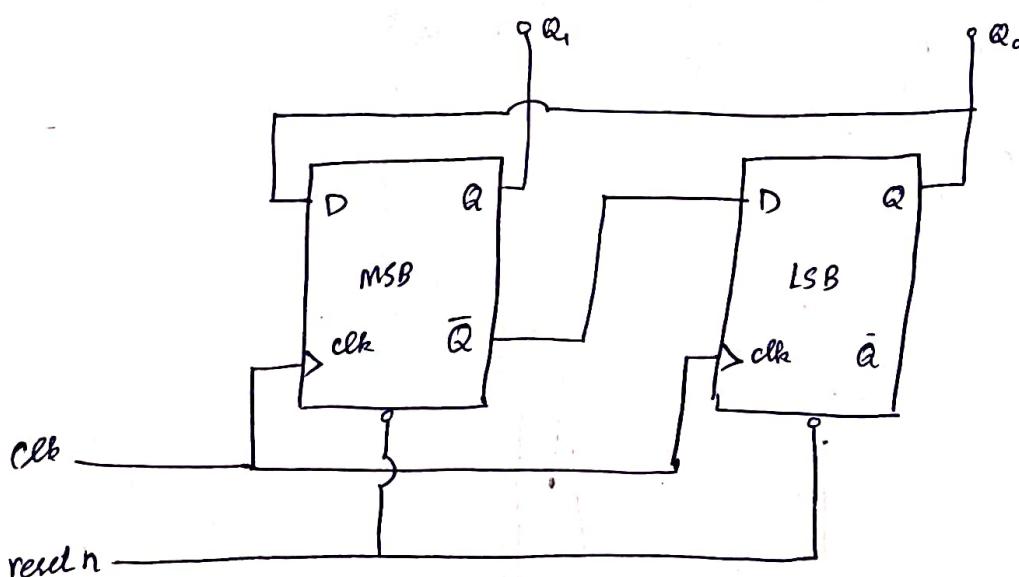
Present state	Q_1	Q_0	Next state		Excitation	
			Q_1^+	Q_0^+	D_1	D_0
0	0	0	0	1	0	1
0	0	1	1	1	1	1
0	1	1	1	0	1	0
0	1	0	0	0	0	0

D_1	\bar{Q}_0	Q_0
\bar{Q}_1	0	1
Q_1	0	1

D_0	\bar{Q}_1	Q_1
\bar{Q}_0	1	1
Q_0	0	0

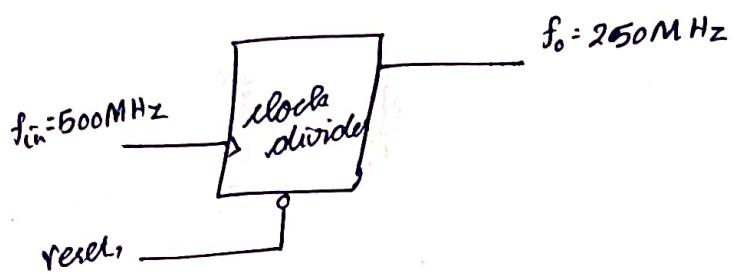
$$D_1 = Q_0$$

$$D_0 = \bar{Q}_1$$



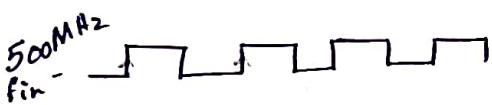
Exercise 39

Design the synchronous sequential circuit to generate 250 MHz o/p clock if i/p clock frequency is 500 MHz use positive edge sensitive D flip-flop having asynchronous low reset.



1) 2 states = 2
1 Flip Flop

2) State table

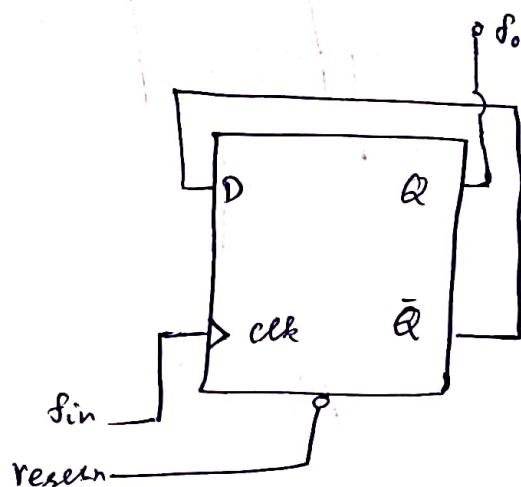


3) Excitation Table

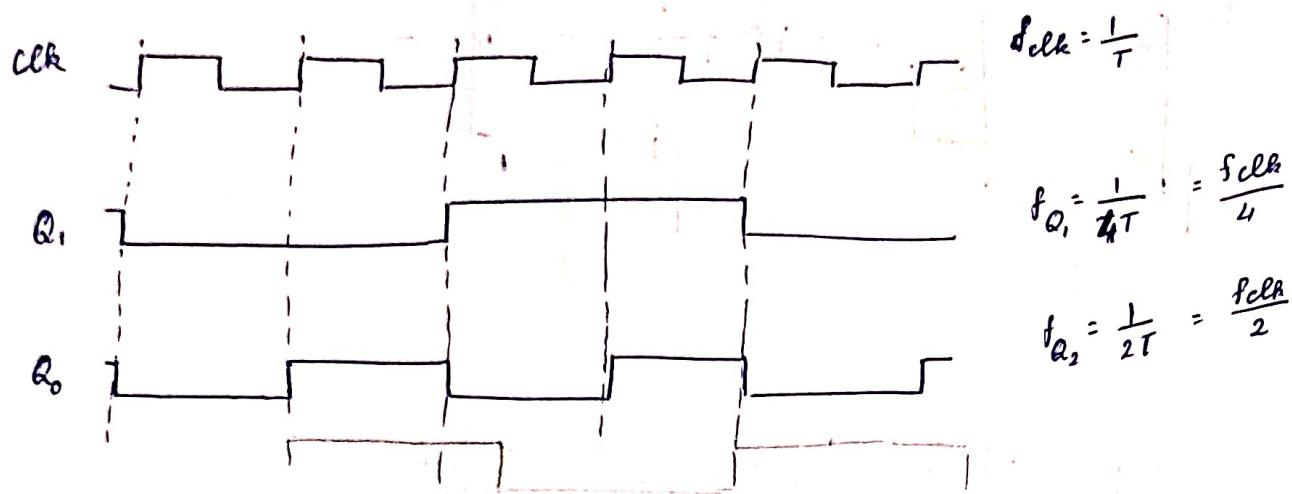
Present state Q	Next state Q^+	Excitation i/p D
0	1	1
1	0	0

Present state Q	Next state Q^+
0	1
1	0

$$D = \bar{Q}$$

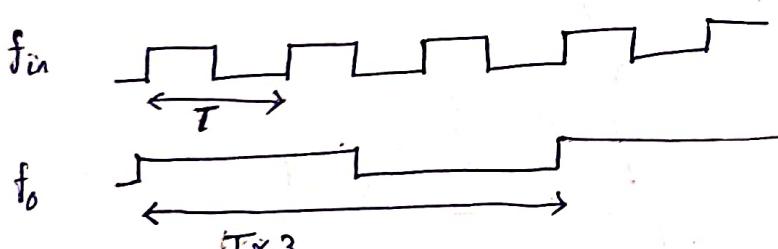
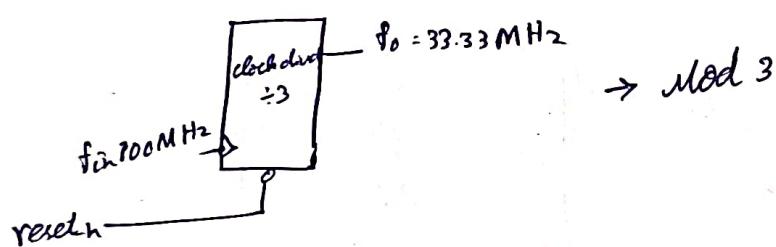


Timing diagram of 2-bit binary up-counter



Exercise 40

Design the synchronous sequential circuit to generate 33.33MHz o/p clock if i/p clock frequency is 100MHz use positive edge sensitive D flip flop having asynchronous active low reset use additional min no of logic gates.



State table		Next state	
Present state		Q_1^+	Q_0^+
Q_1	Q_0	Q_1^+	Q_0^+
0	0	0	1
0	1	1	0
1	0	0	0

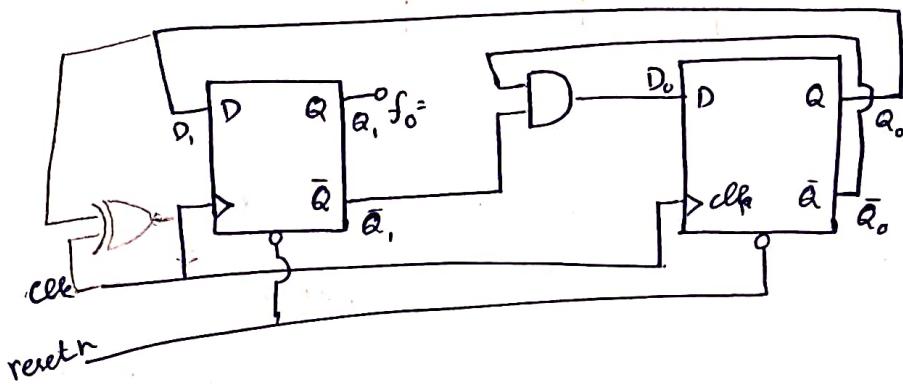
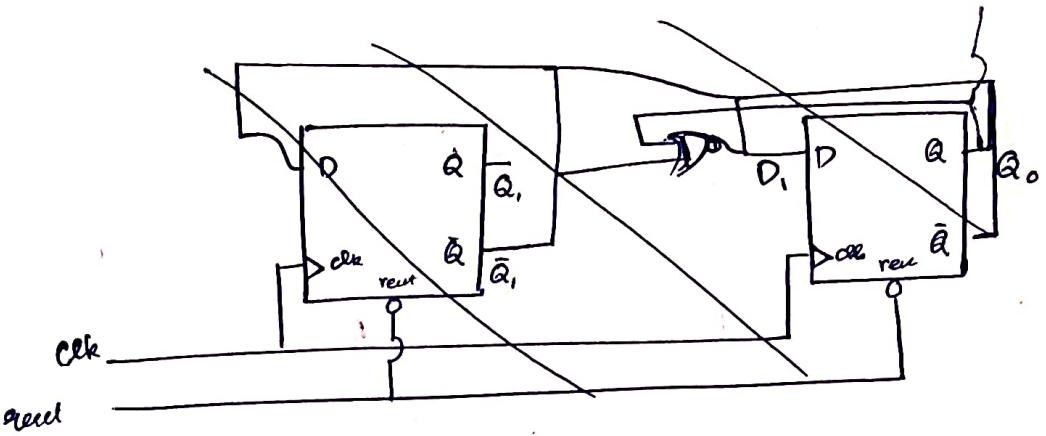
Excitation Table		Excitation	
Present state		Next state	ip
Q_1	Q_0	Q_1^+	D_1
0	0	0	1
0	1	1	0
1	0	0	0

D_1	\bar{Q}_0	Q_0
Q_1	0	1
Q_1	1	0
Q_1	x	x

$$D_1 = Q_0$$

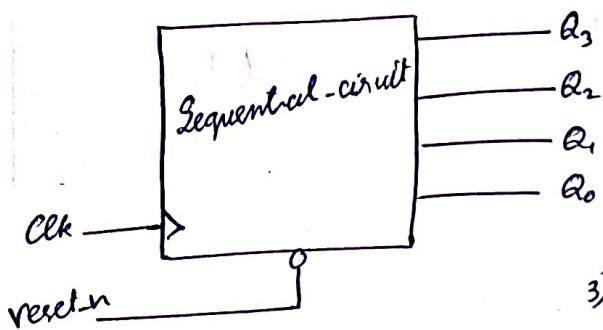
$$D_0 = \overline{Q_0} + \overline{Q_1}$$

(69)



$\Rightarrow \text{Mod}^3$

Exercise 4!
Design the synchronous circuit to get output as $\overline{Q_1 Q_2 Q_3}$



III Ring counter

1) 4 flip flop.

2) State table

				Present state	Next state
				$Q_3 Q_2 Q_1 Q_0$	$Q_3 Q_2^+ Q_1^+ Q_0^+$
				$Q_3 Q_2 Q_1 Q_0$	$Q_3 Q_2^+ Q_1^+ Q_0^+$
3	1	0	0	0	0 1 0 0
4	0	1	0	0 0 1 0	0 0 1 0
2	0	0	1	0 0 0 1	0 0 0 1
1	0	0	0	1 0 0 0	1 0 0 0

3) Excitation table

		Present state	Next state	Excitation i/p
		$Q_3 Q_2 Q_1 Q_0$	$Q_3 Q_2^+ Q_1^+ Q_0^+$	$D_3 D_2 D_1 D_0$
		0 1 0 0	0 1 0 0	0 1 0 0
		0 0 1 0	0 0 1 0	0 0 1 0
		0 0 0 1	0 0 0 1	0 0 0 1
		1 0 0 0	1 0 0 0	1 0 0 0

D_3	$\bar{Q}_1 \bar{Q}_0$	$\bar{Q}_1 Q_0$	$Q_1 Q_0$	$Q_1 \bar{Q}_0$
$\bar{Q}_3 \bar{Q}_2$	X	1 X	0	
$\bar{Q}_3 Q_2$	0	X X	X	
$Q_3 Q_2$	X	X X	X	
$Q_3 \bar{Q}_2$	0	X X	X	

$$D_3 = Q_0$$

D_1	$\bar{Q} \bar{Q}_0$	$\bar{Q} Q_0$	$Q \bar{Q}_0$	$Q Q_0$
$\bar{Q}_3 \bar{Q}_2$	X 0	X 0		
$\bar{Q}_3 Q_2$	1 X	X X	X	
$Q_3 Q_2$	X X	X X	X X	
$Q_3 \bar{Q}_2$	0	X X	X X	X

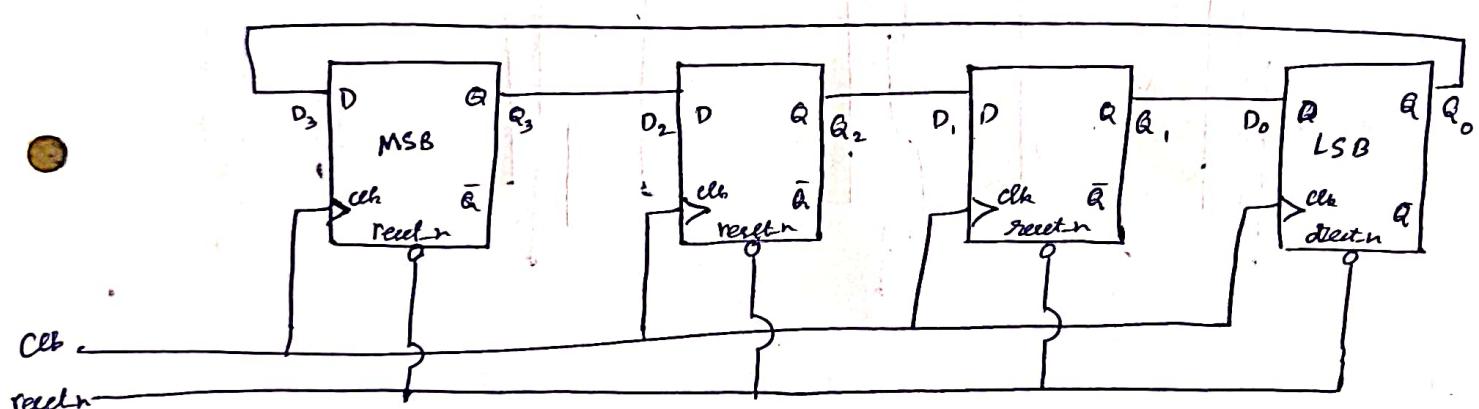
$$D_1 = Q_2$$

D_2	$\bar{Q}_1 \bar{Q}_0$	$\bar{Q}_1 Q_0$	$Q_1 Q_0$	$Q_1 \bar{Q}_0$
$\bar{Q}_3 \bar{Q}_2$	X 0	X 0		
$\bar{Q}_3 Q_2$	0 X	X X	X X	
$Q_3 Q_2$	X X	X X	X X	X X
$Q_3 \bar{Q}_2$	1	X X	X X	X X

$$D_2 = Q_3$$

D_0	$\bar{Q}_1 \bar{Q}_0$	$\bar{Q}_1 Q_0$	$Q_1 Q_0$	$Q_1 \bar{Q}_0$
$\bar{Q}_3 \bar{Q}_2$	X 0	X 1		
$\bar{Q}_3 Q_2$	0 X	X X	X X	
$Q_3 Q_2$	X X	X X	X X	X X
$Q_3 \bar{Q}_2$	0	X X	X X	X X

$$D_0 = Q_1$$

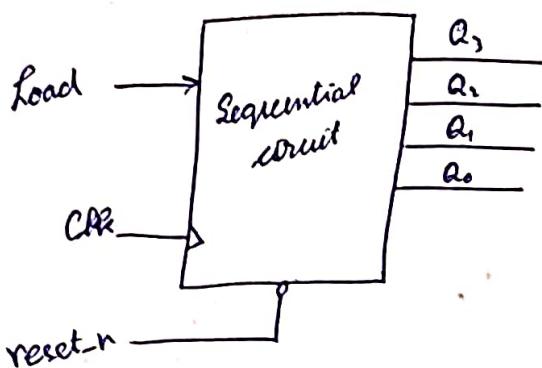


Limitation

→ Assumption $Q_3 Q_2 Q_1 Q_0 = 1000$ at initial

Exercise 4.2

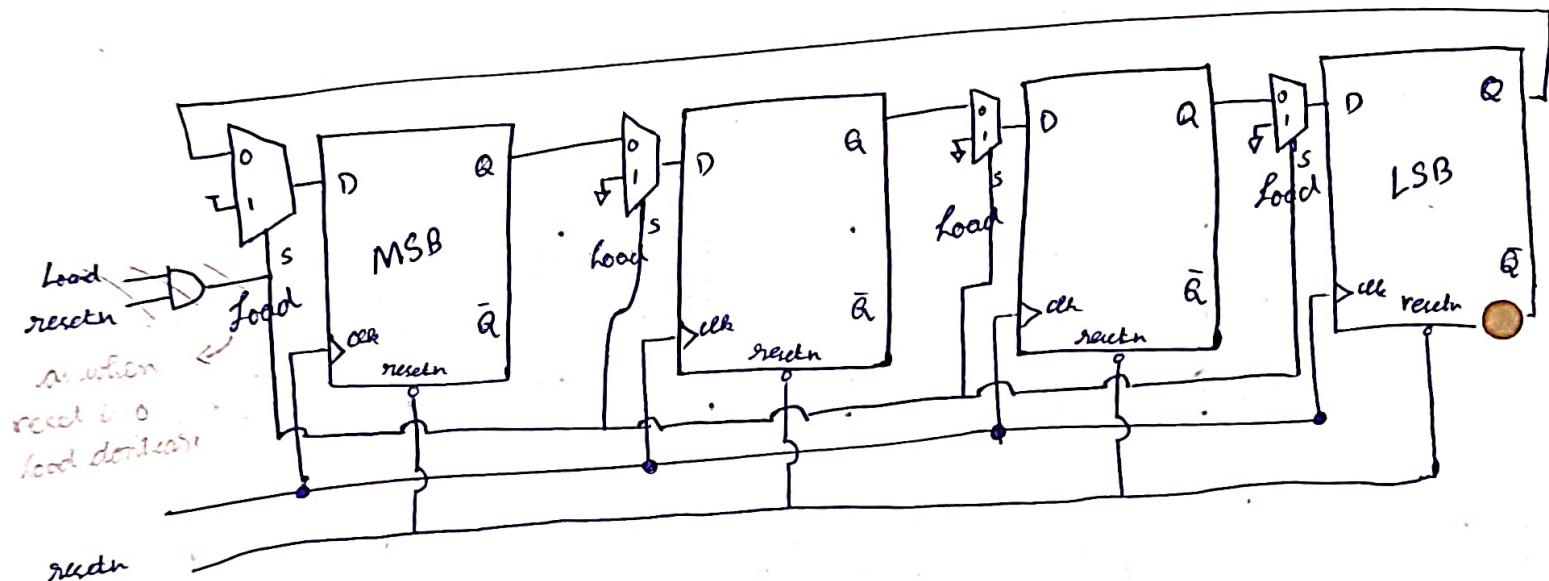
Design the synchronous sequential circuit to set 0₄ at 8421)



resetn	load	clk	Q ₃	Q ₂	Q ₁	Q ₀
0	X	X	0	0	0	0
1	1	—	1	0	0	0
1	0	—	normal operation 4, 2, 1, 8 ...			

NAND MUX	
0 0	0
0 1	0
1 0	0
1 1	1

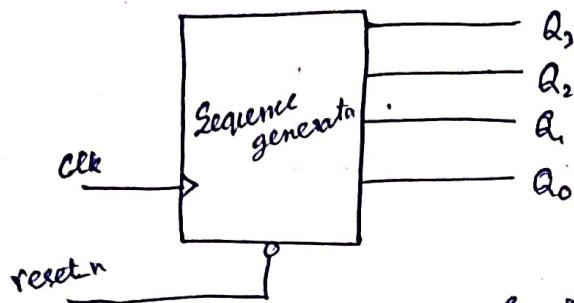
↓ select



→ Synchronous load ↑

Evening session
Exercise 4.3

Design sequential circuit to generate o/p sequence $0, 8, 12, 14, 15, 7, 3, 1$
using positive edge triggered flip flop consider sync-active low reset.



1) 4 digits \rightarrow 4 flip flop

2) State Table

Present state				Next state			
Q_3	Q_2	Q_1	Q_0	Q_3^+	Q_2^+	Q_1^+	Q_0^+
0 0 0 0	1 0 0 0			0 0 0 0			
1 0 0 0	1 1 0 0			1 0 0 0			
1 1 0 0	1 1 1 0			1 1 0 0			
1 1 1 0	1 1 1 1			1 1 1 0			
1 1 1 1	0 1 1 1			1 1 1 1			
0 1 1 1	0 0 1 1			0 1 1 1			
0 0 1 1	0 0 0 1			0 0 1 1			
0 0 0 1	0 0 0 0			0 0 0 1			

D_3	$\bar{Q}_2\bar{Q}_3$	\bar{Q}_2Q_3	Q_2Q_3	$Q_2\bar{Q}_3$
$\bar{Q}_2\bar{Q}_3$	1	0	0	x
\bar{Q}_2Q_3	x	x	0	x
$Q_2\bar{Q}_3$	1	x	0	1
Q_2Q_3	1	x	x	x

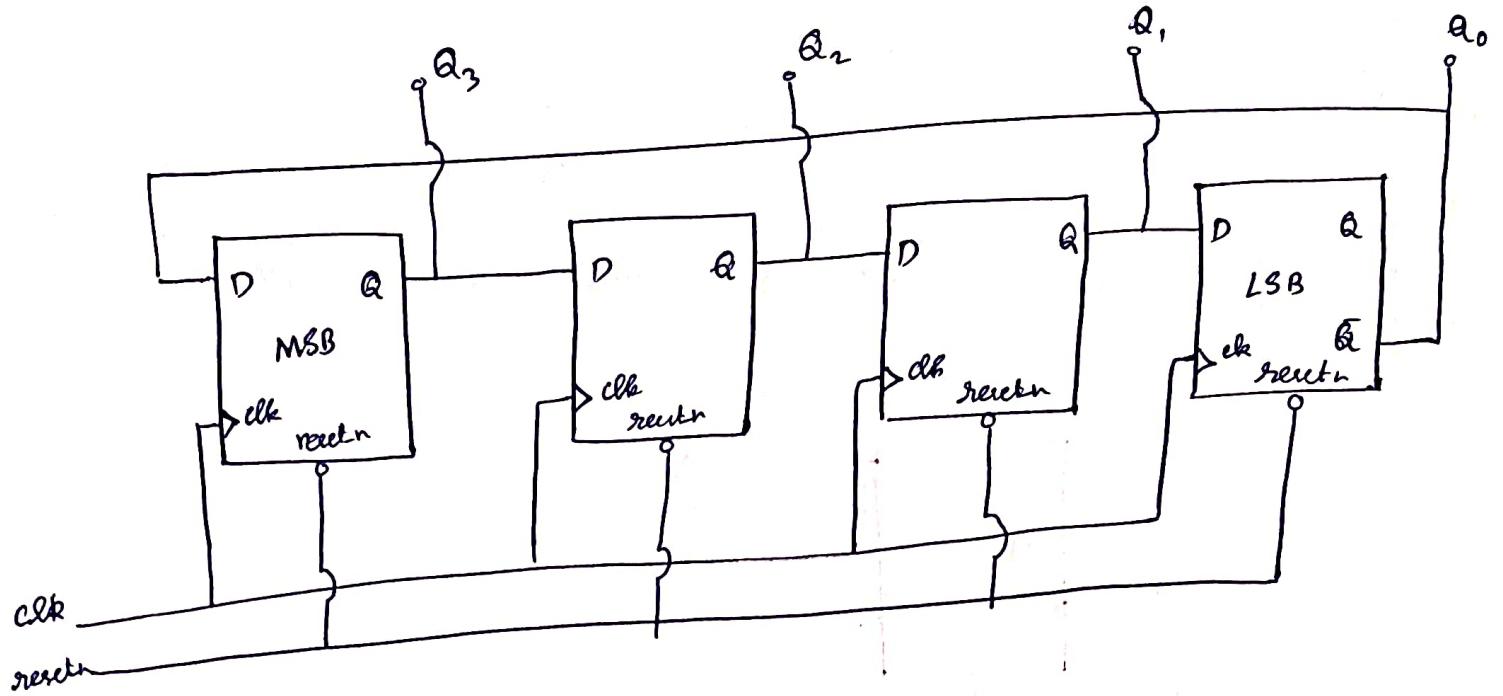
$$D_3 = \bar{Q}_2$$

$$D_2 = Q_3$$

$$D_1 = \bar{Q}_2$$

$$D_0 = Q_1$$

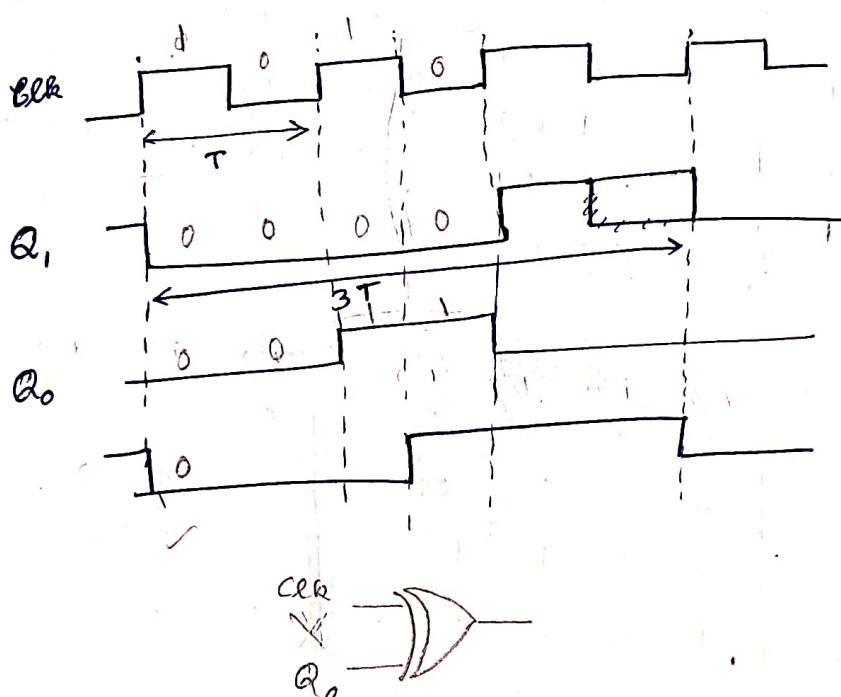
D_2	$\bar{Q}_2\bar{Q}_3$	\bar{Q}_2Q_3	Q_2Q_3	$Q_2\bar{Q}_3$	\bar{Q}_2
$\bar{Q}_2\bar{Q}_3$	0	0	0	x	$\bar{Q}_2\bar{Q}_3$
\bar{Q}_2Q_3	x	x	0	x	\bar{Q}_2Q_3
$Q_2\bar{Q}_3$	1	x	1	1	$Q_2\bar{Q}_3$
Q_2Q_3	1	x	x	x	Q_2Q_3



Johnson counter \uparrow - (Twisted ring counter)

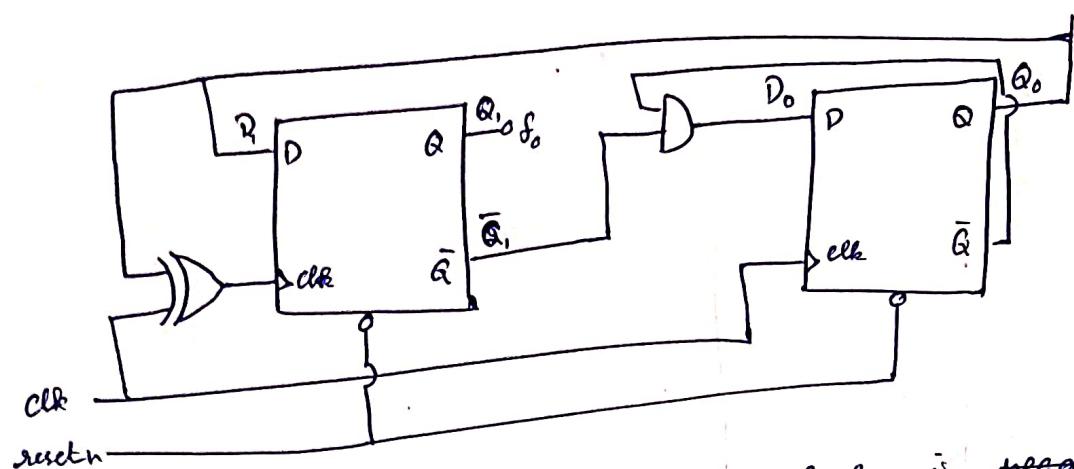
Exercise 6b

Design $\div 3$ mod 3 synchronous binary upcounter to have 50% duty cycle



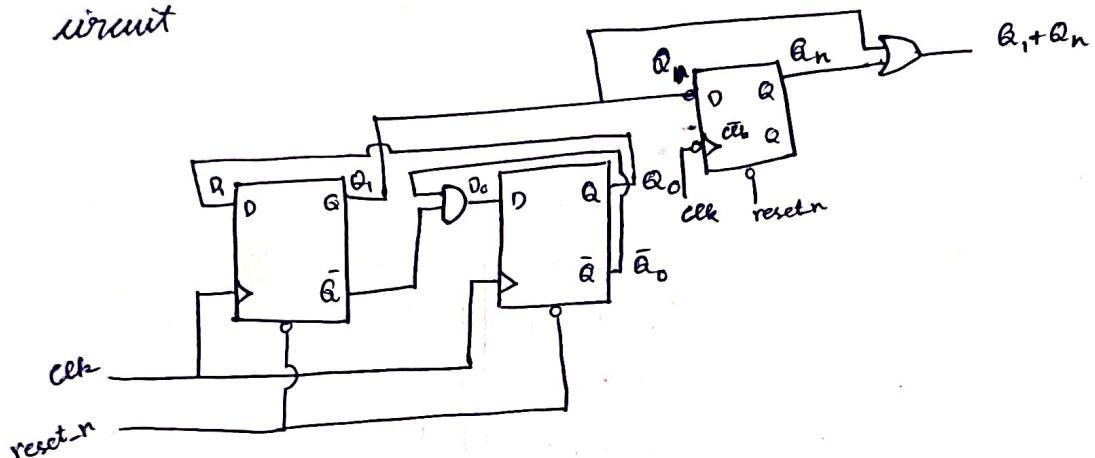
	Q_1	Q_0
0	0	0
1	0	1
2	1	0

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

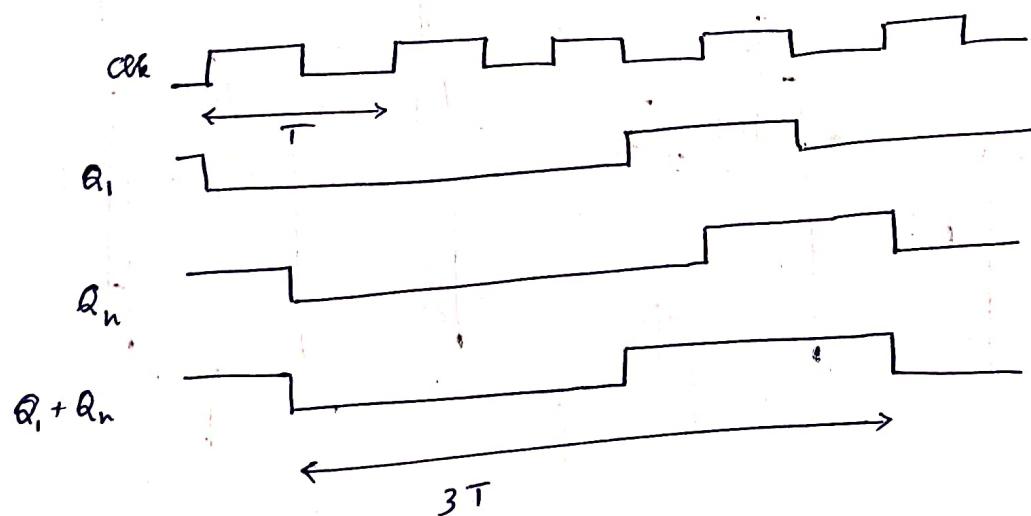
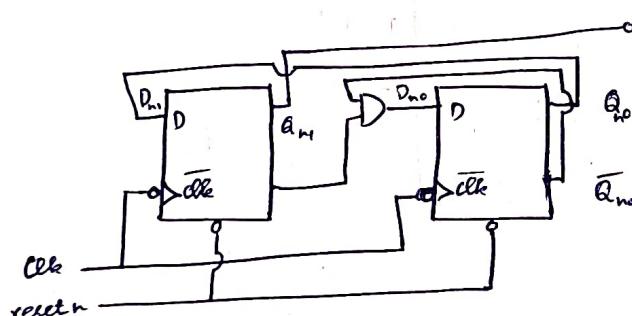


X Not allowed to change clock in negative synchronous circuit

•

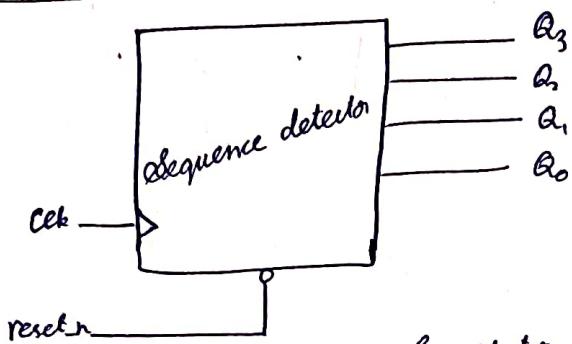


•



Exercise 15

Design a synchronous circuit to generate a sequence sync-active low reset
 $\{0, 2, 4, 6, 8, 10, 12, 14\}$

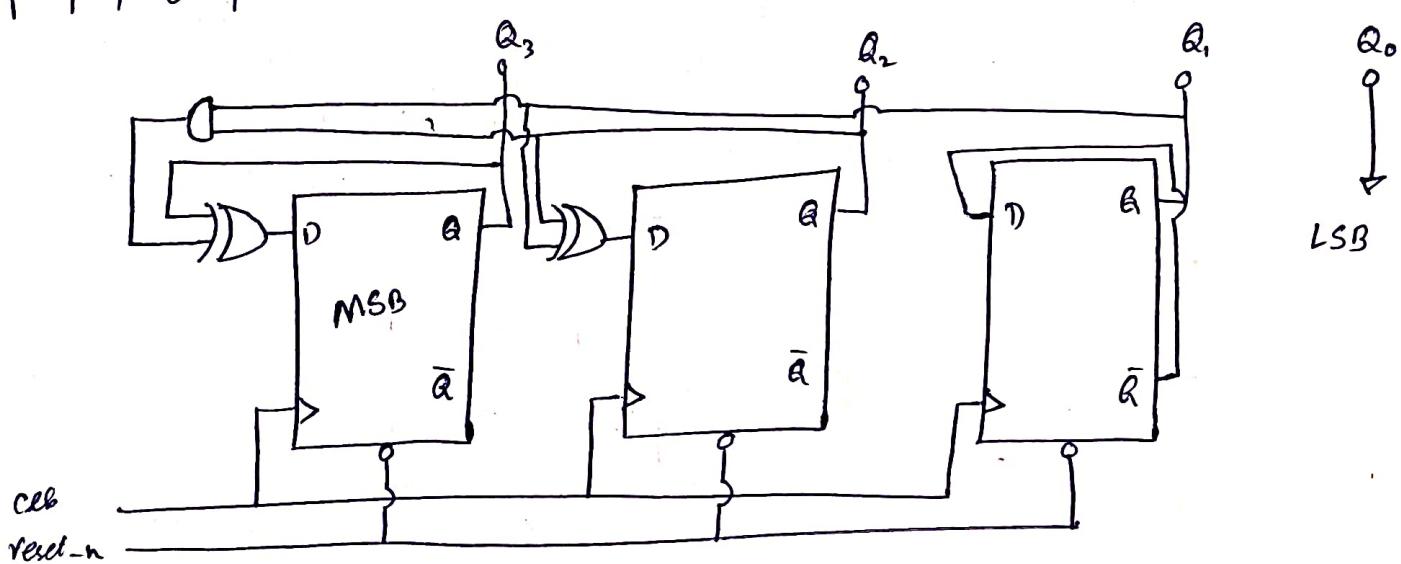


A - Tim Flon

State table				Next state			
Present state				Q_3^+	Q_2^+	Q_1^+	Q_0^+
Q_3	Q_2	Q_1	Q_0				
0	0	0	0	0	0	1	0
0	0	1	0	0	1	0	0
0	1	0	0	0	1	1	0
0	1	1	0	0	1	1	0
0	1	1	0	1	0	0	0
1	0	0	0	1	0	1	0
1	0	1	0	1	0	1	0
1	1	0	0	1	1	1	0
1	1	1	0	1	1	1	0
1	1	1	0	0	0	0	0

Excitation table

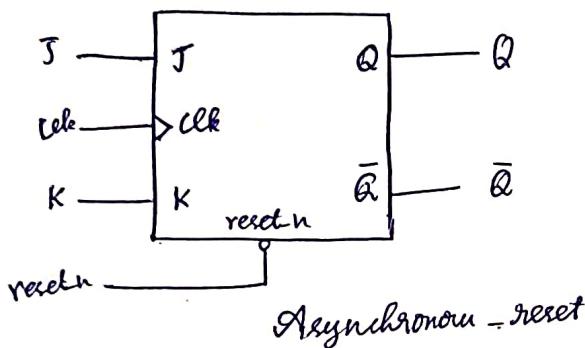
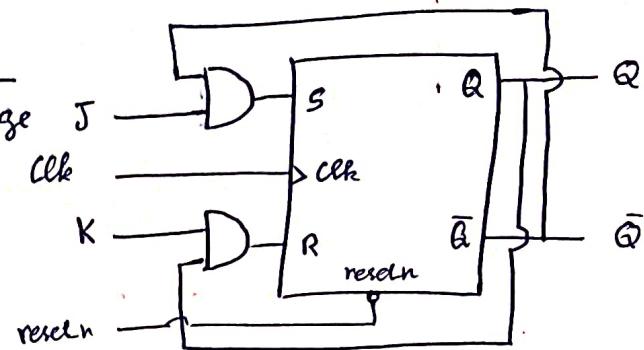
Present state Q_3, Q_2, Q_1, Q_0	Next state $Q_3^+, Q_2^+, Q_1^+, Q_0^+$				Excitation i/p D_3, D_2, D_1, D_0	
	Q_3^+	Q_2^+	Q_1^+	Q_0^+		
0 0 0 0	0	0	0	1	0	0 0 1 0
0 0 1 0	0	0	1	0	0	0 1 0 0
0 1 0 0	0	1	0	0	0	0 1 1 0
0 1 1 0	0	1	1	0	0	1 0 0 0
1 0 0 0	1	0	0	0	0	1 1 0 0
1 0 1 0	1	0	1	0	0	1 1 1 0
1 1 0 0	1	1	0	0	0	1 1 1 0
1 1 1 0	1	1	1	0	0	0 0 0 0



11.09.2024

JK Flip Flop.
→ Jack Kilby

J	K	Q	\bar{Q}
0	0	No change	
0	1	0	1
1	0	1	0
1	1	Toggle	reset



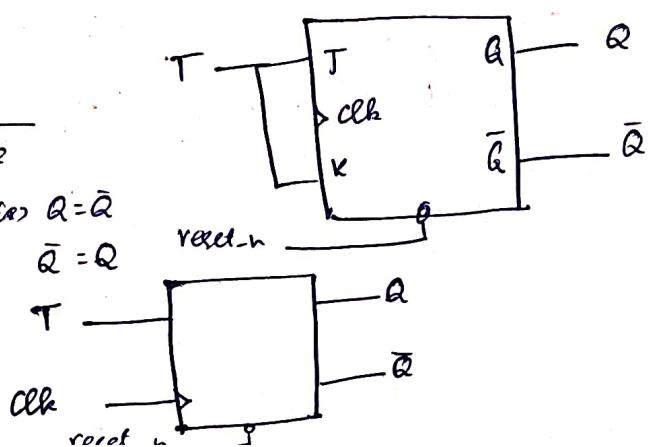
JK Flip Flop not used in digital design as both inputs need to be controlled only D Flip Flop are used.

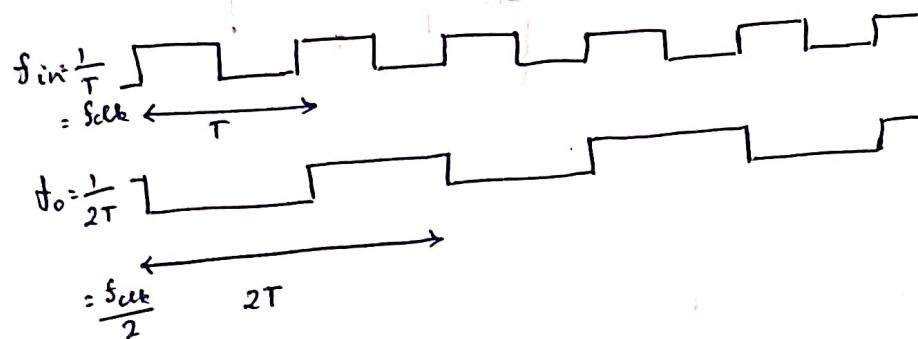
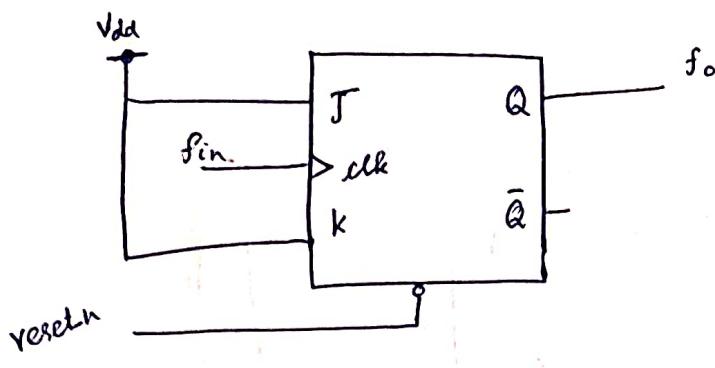
To prevent race-around condition

- Propagation delay should be greater than $t_{on} \rightarrow$ time of high pulse.
- $t_{pd} > t_{on}$
- master-slave configuration

T Flip Flop

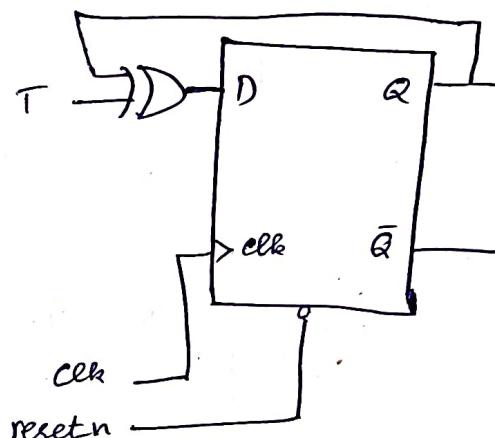
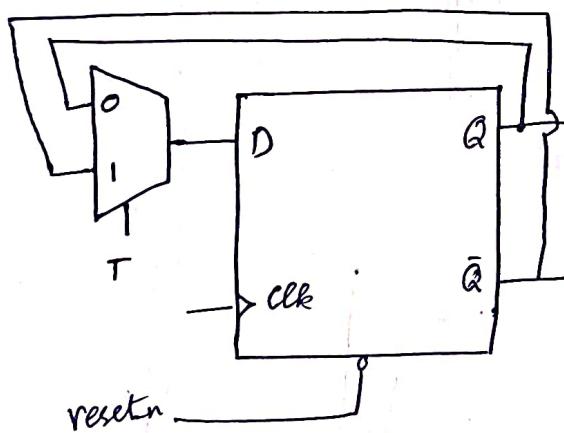
T	Q	\bar{Q}
0	No change	
1	Toggle (i.e. $Q = \bar{Q}$)	$\bar{Q} = Q$





Exercise 46
 Design T flip flop using D flip flop using rising edge of clock
 & active low resetn

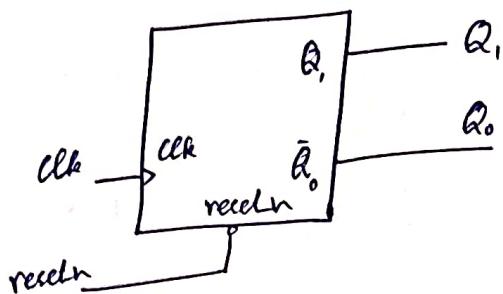
T	Q	\bar{Q}
0		No change
1		toggle



Exercise 17

Design 2 bit synchronous binary upcounter & min.no of logic gates
 JK flip flop

async - active Low reset



1) 4 stage = 2^2
 2 Flip Flops.

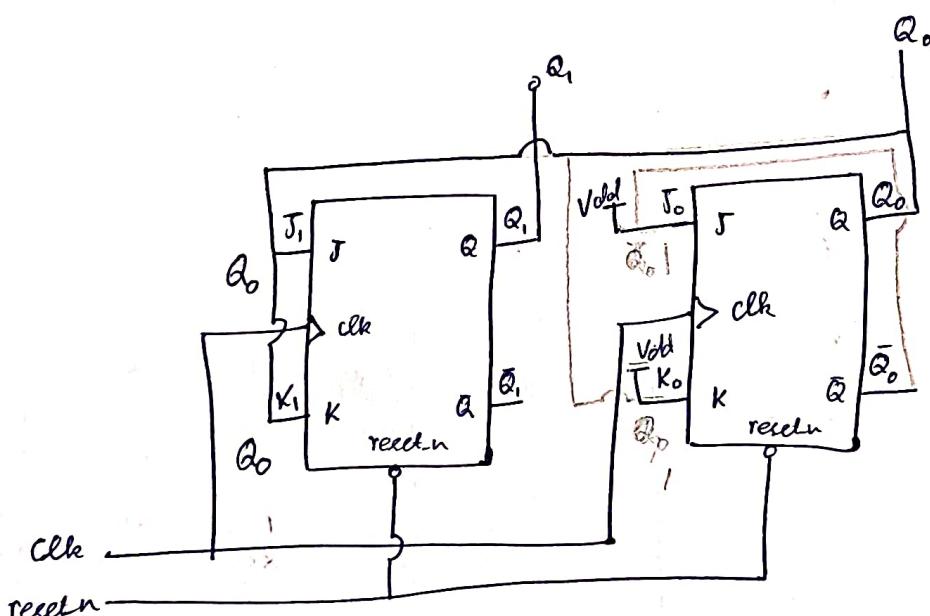
2) state table

Present state		Next state	
Q_1	Q_0	Q_1'	Q_0'
0	0	0	1
0	1	1	0
1	0	1	1
1	1	0	0

Excitation table

Present state		Next state		Excitation i/p	
Q_1	Q_0	Q_1'	Q_0'	$J, K,$	$J_0 K_0$
0	0	0	1	0 x	1 x
0	1	1	0	1 x	x 1
1	0	1	1	x 0	1 x
1	1	0	0	x 1	x 1

→ 1
↓
 Q_0



Excitation table for JK flip flop

Present state Q	Q^+	Next state		Excitation i/p	
		Q'	Q	J	K
0	0	0	0	0	0
0	1	1	0	0	1
1	0	1	1	1	0
1	1	0	1	0	1

$\Rightarrow 0 \times$

$\Rightarrow 1 \times$

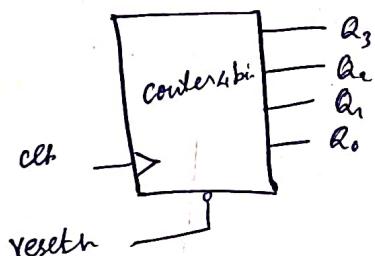
$\Rightarrow \times 1$

$\Rightarrow \times 0$

Evening session

Exercise 48

Design a synchronous 4-bit binary up-counter using min no of JK flip flop
consider active low asynchronous resetn



A - Flip Flop

State Table

Present state $Q_3\ Q_2\ Q_1\ Q_0$	Next state $Q_3^+\ Q_2^+\ Q_1^+\ Q_0^+$	Present state				Next state			
		Q_3	Q_2	Q_1	Q_0	Q_3^+	Q_2^+	Q_1^+	Q_0^+
0 0 0 0	0 0 0 1	0	0	0	1	0	0	0	1
0 0 0 1	0 0 1 0	0	0	1	0	0	0	1	0
0 0 1 0	0 0 1 1	0	0	1	1	0	0	1	1
0 0 1 1	0 1 0 0	0	1	0	0	0	1	0	0
0 1 0 0	0 1 0 1	0	1	0	1	0	1	0	1
0 1 0 1	0 1 1 0	0	1	1	0	0	1	1	0
0 1 1 0	0 1 1 1	0	1	1	1	0	1	1	1
0 1 1 1	1 0 0 0	1	0	0	0	1	0	0	0

Excitation Table

Present state $Q_3\ Q_2\ Q_1\ Q_0$	$Q_3^+\ Q_2^+\ Q_1^+\ Q_0^+$	Next state				Excitation i/p			
		Q_3^+	Q_2^+	Q_1^+	Q_0^+	$J_3\ K_3$	$J_2\ K_2$	$J\ K$	$J_0\ K_0$
0 0 0 0	0 0 0 1	0	0	0	1	0 X	0 X	0 X	1 X
0 0 0 1	0 0 1 0	0	0	1	0	0 X	0 X	1 X	X 1
0 0 1 0	0 0 1 1	0	0	1	1	0 X	0 X	X 0	1 X
0 0 1 1	0 1 0 0	0	1	0	0	0 X	0 X	X 1	X 1
0 1 0 0	0 1 0 1	0	1	0	1	0 X	X 0	0 X	1 X
0 1 0 1	0 1 1 0	0	1	1	0	X 0	X 0	X 0	X 1
0 1 1 0	0 1 1 1	0	1	1	1	X 0	X 1	X 1	X 1
0 1 1 1	1 0 0 0	1	0	0	0	1 X			

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
1	1	1	1
1	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
1	1	1	1
1	1	1	1

J_3	Q, Q_0
$Q_3 Q_2$	0 0 0 0
-	0 0 1 0
-	x x x x
-	x x x x

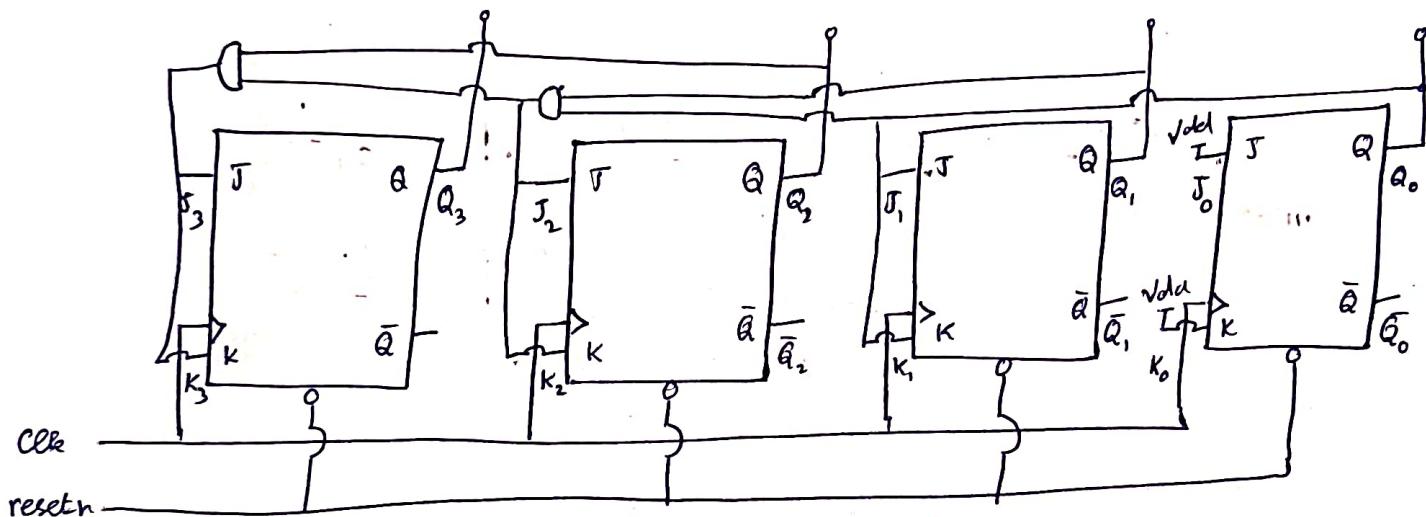
$$J_3 = \overline{Q_3} + Q_2(Q_0 Q_0)$$

J_2	Q, Q_0
$Q_3 Q_2$	0 0 1 0
-	x x x x
-	x x x x
-	0 0 1 0

$$J_2 = Q_1 Q_0$$

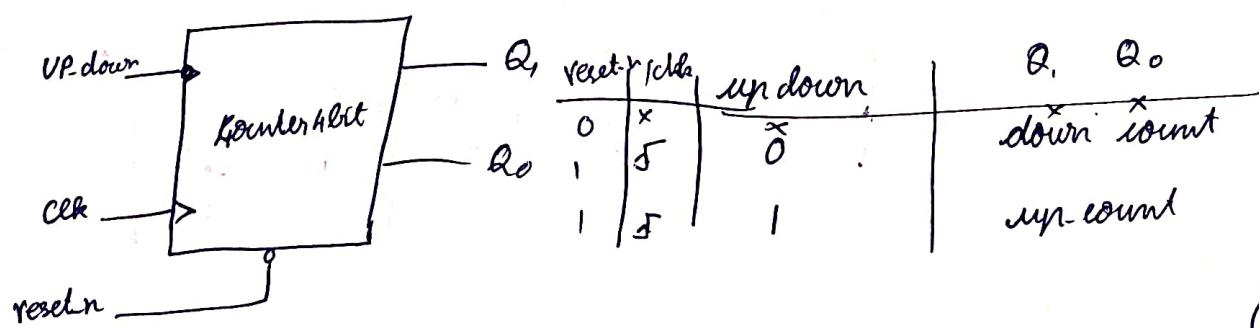
J_1	Q, Q_0
$Q_3 Q_2$	0 1 x x
-	0 1 x x
-	0 1 x x
-	0 1 x x

$$J_1 = Q_0$$



Exercise 49

Design a synchronous 2 bit binary up/down counter using min no. of positive edge sensitive JK flip-flops. Consider active low asynchronous resetn



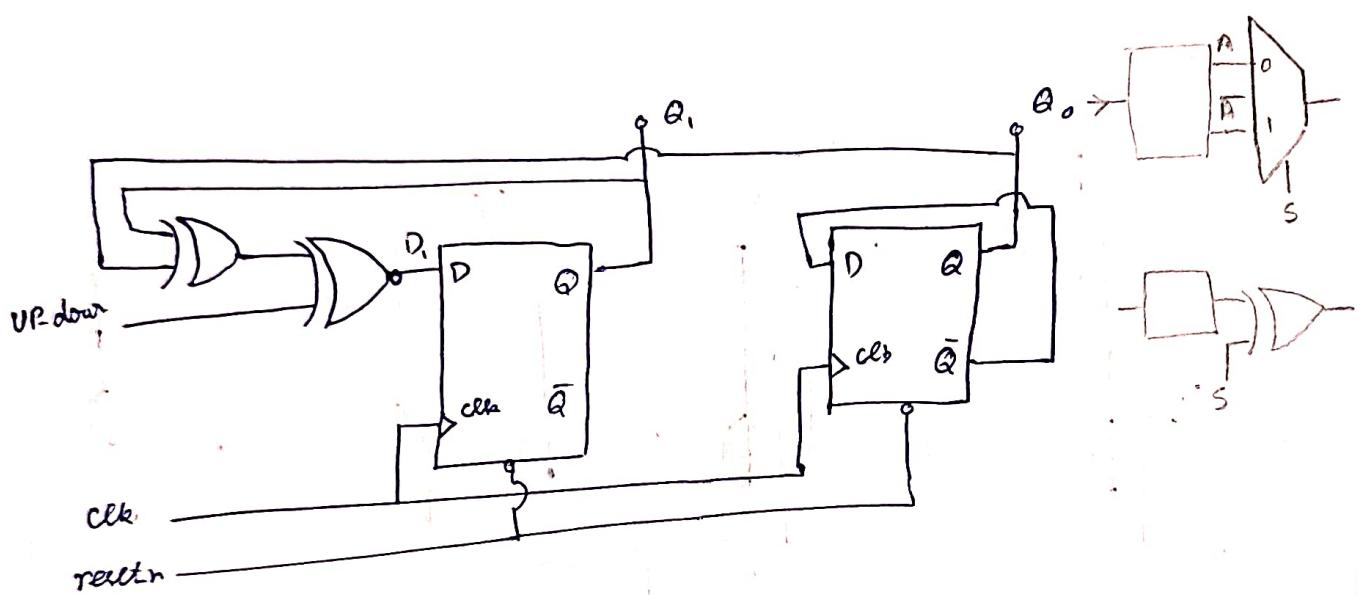
2-bit down counter

state table

Present state		Next state	
Q_1	Q_0	Q_1^+	Q_0^+
1	1	1	0
1	0	0	1
0	1	0	0
0	0	1	1

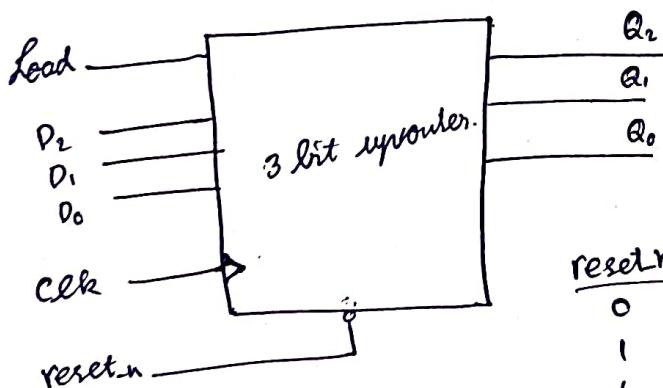
Excitation table

Present state		Next state		Excitation & p	
Q_1	Q_0	Q_1^+	Q_0^+	D_1	D_0
1	1	1	0	1	0
1	0	0	1	0	1
0	1	0	0	0	0
0	0	1	1	1	1

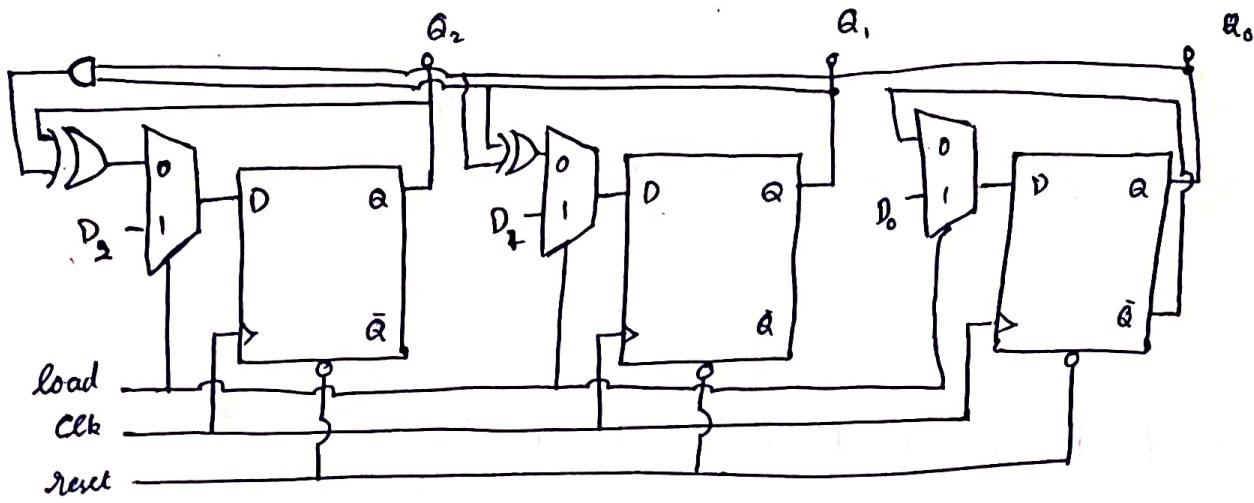


Exercise 50

Design 3 bit synchronous presetable up counter having rising edge of clock & active low asyn resetn



resetn	load	clk	$Q_2 Q_1 Q_0$
0	x	x	0 0 0
1	1	-	D2 D1 D0
1	0	-	asynchronous



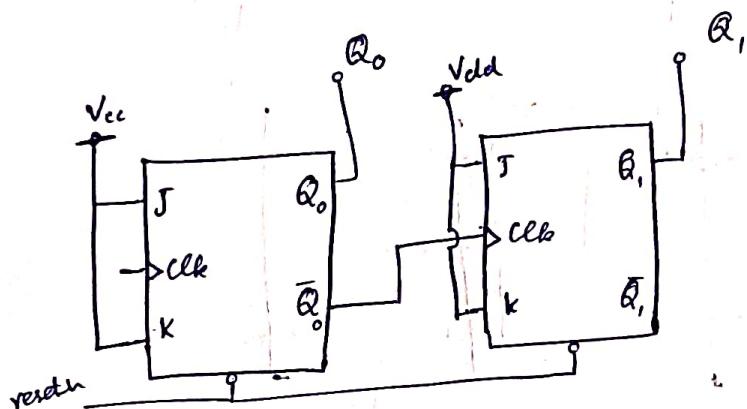
$$Q_2 + (Q_1 \oplus Q_0) \bar{L} + D_2 L$$

$$\bar{Q}_0 \bar{L} + D_0 L$$

$$Q_1 \oplus Q_0 \bar{L} + D_1 L$$

Exercise 5:
What is the circuit & what is o/p of following logic

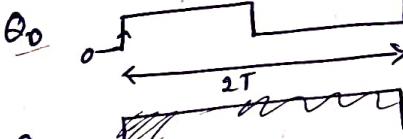
circuit



clk



$$f = \frac{f_{clk}}{2}$$

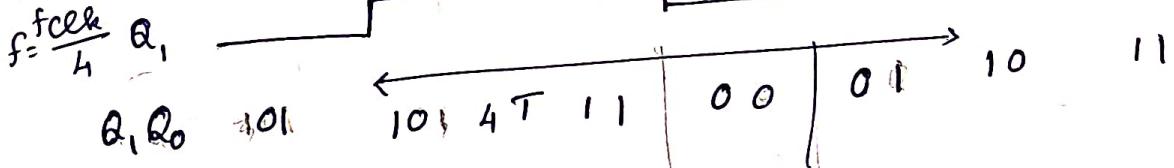


Q0

Q0-bar

Inference 2

2-bit async
up counter



Q1, Q0 101

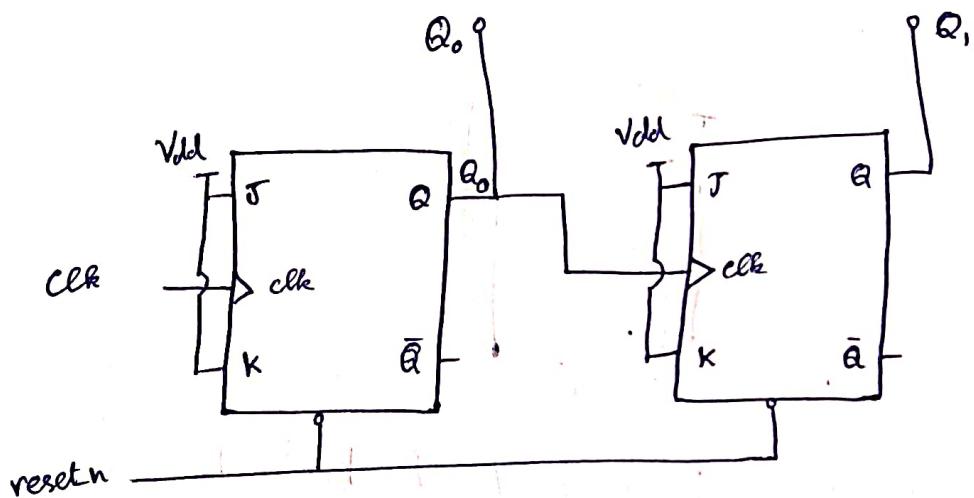
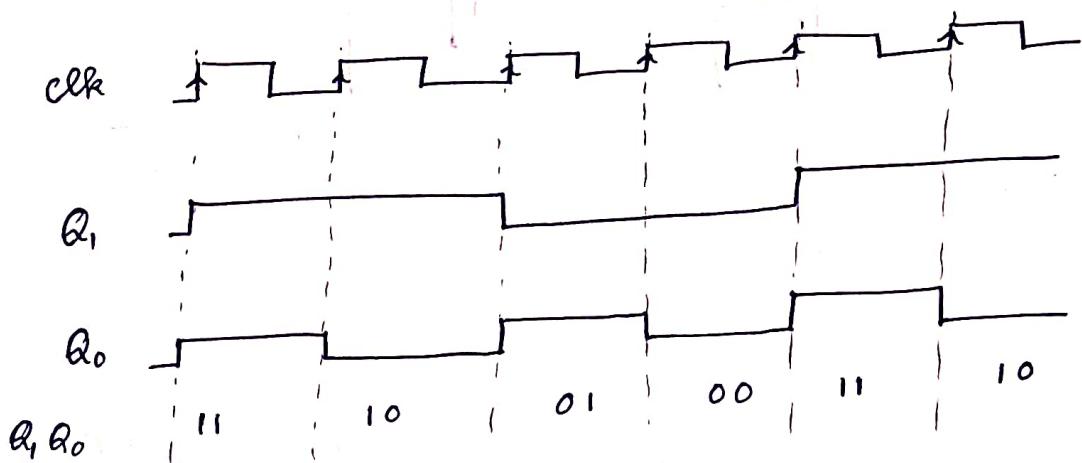
101 4T 11

00

01 10 11

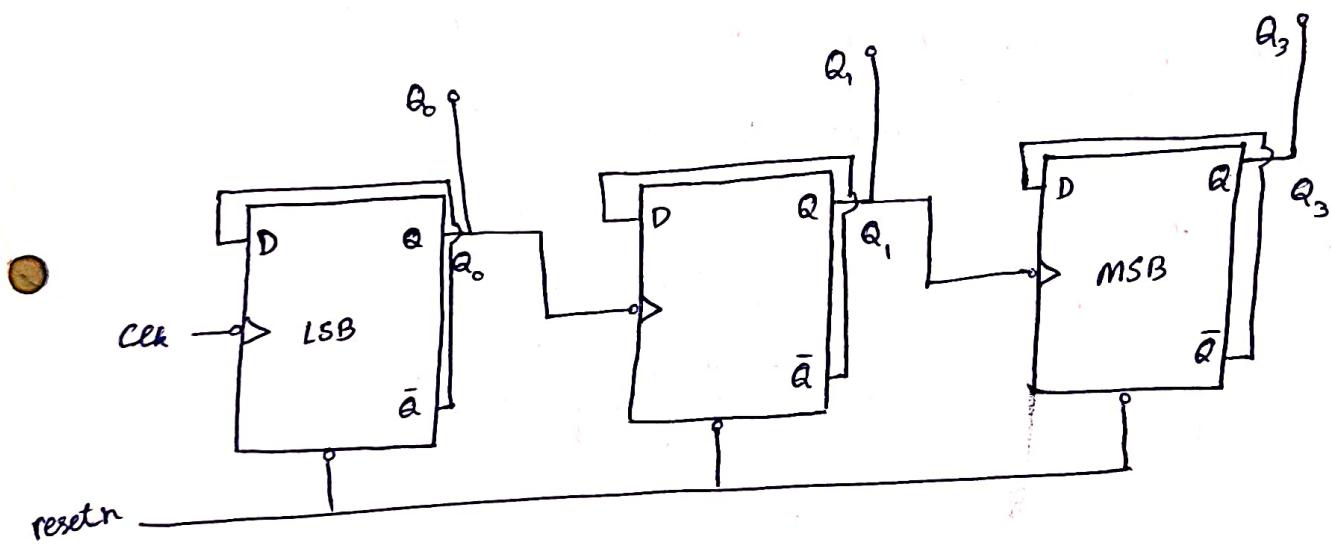
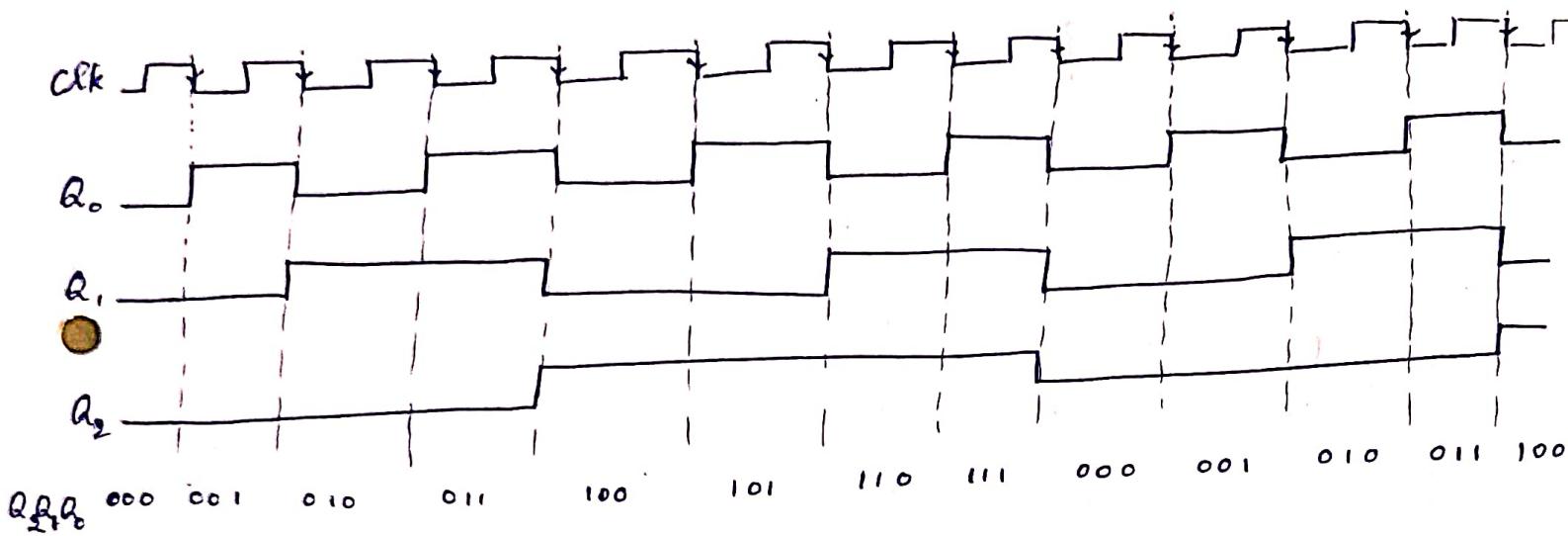
(83)

12-09-2024
Exercise 5.2
 Design 2 bit asynchronous binary down counter uses positive
 edge sensitive JK flip flop use active low asynchronous
 reset-n.

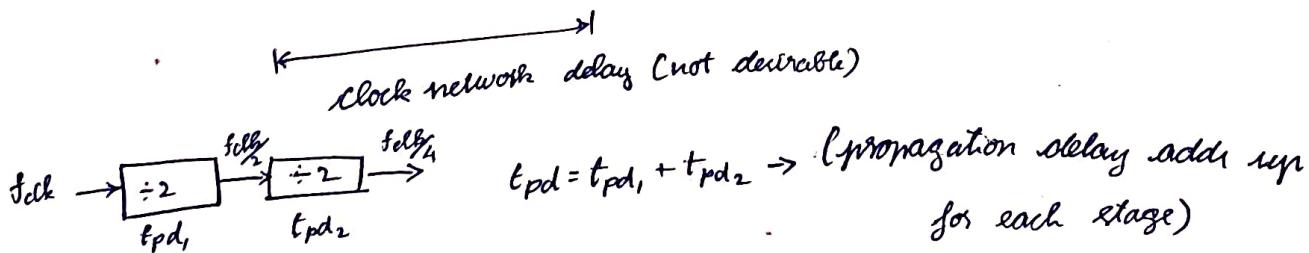
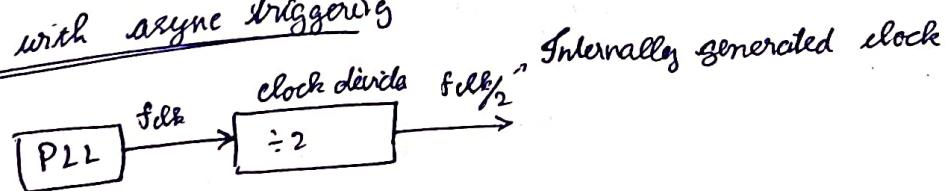


Exercise 53

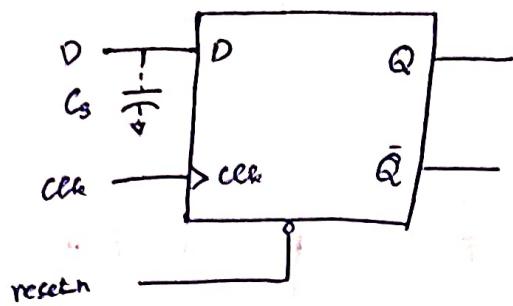
Design 3 bit asy whole binary up counter using D flip flops having negative edge sensitive clock & active low asy whole reset.



Issues with asy whole triggering



Timing parameters, metadability, STA (Static timing analysis)

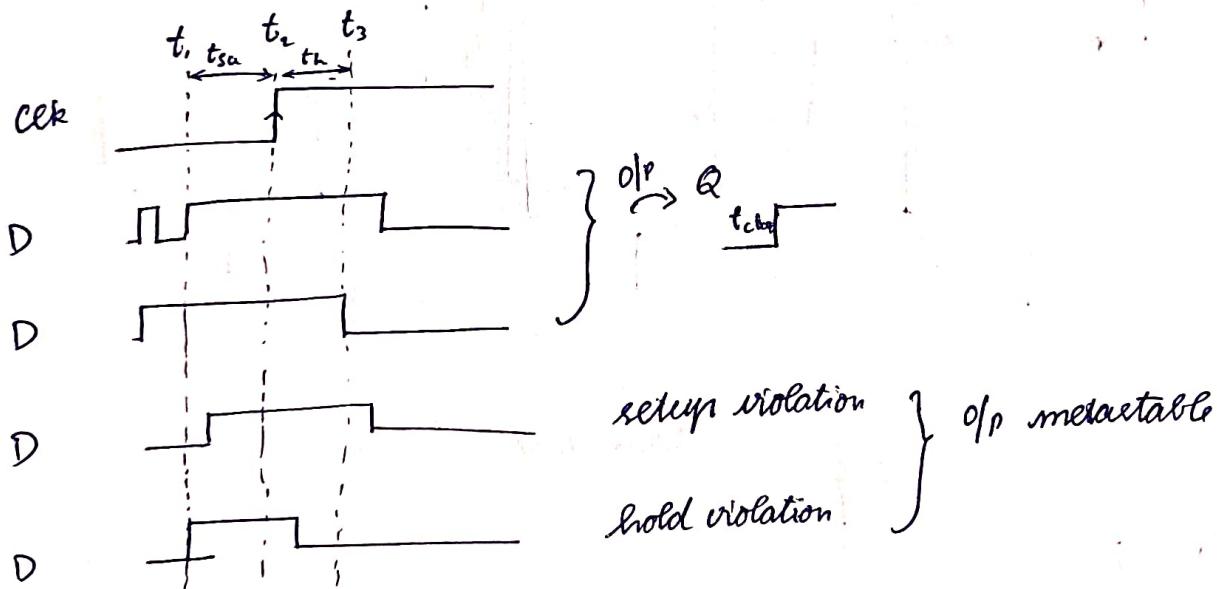


Timing parameters

$t_{ctog} = t_{pff}$ = flip flop propagation delay.
= clock to Q delay

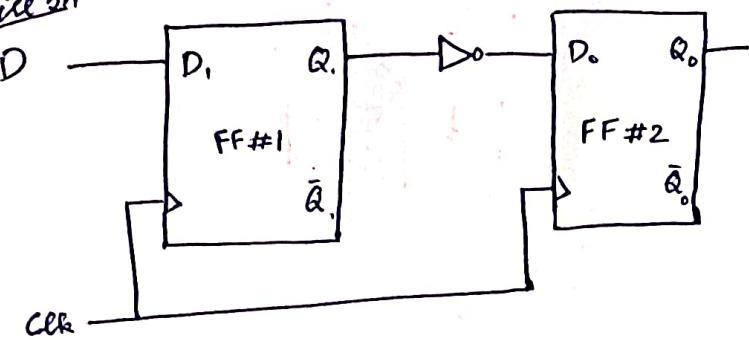
t_{su} = The min amount of time for which D should be stable before arrival of active edge of clock

t_h = The min amount of time for which D should be stable after arrival of active edge of clock



Application

Exercise 5.4



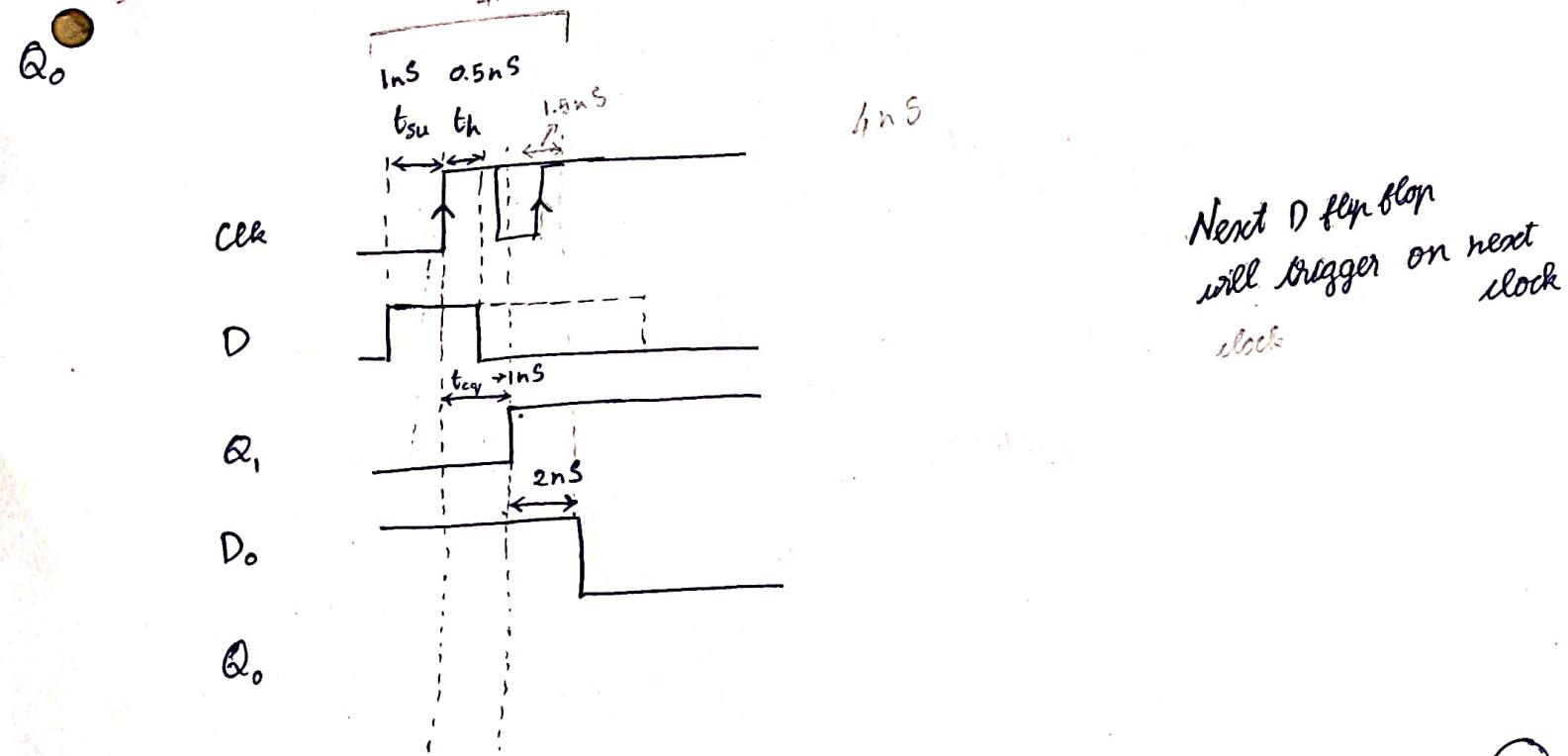
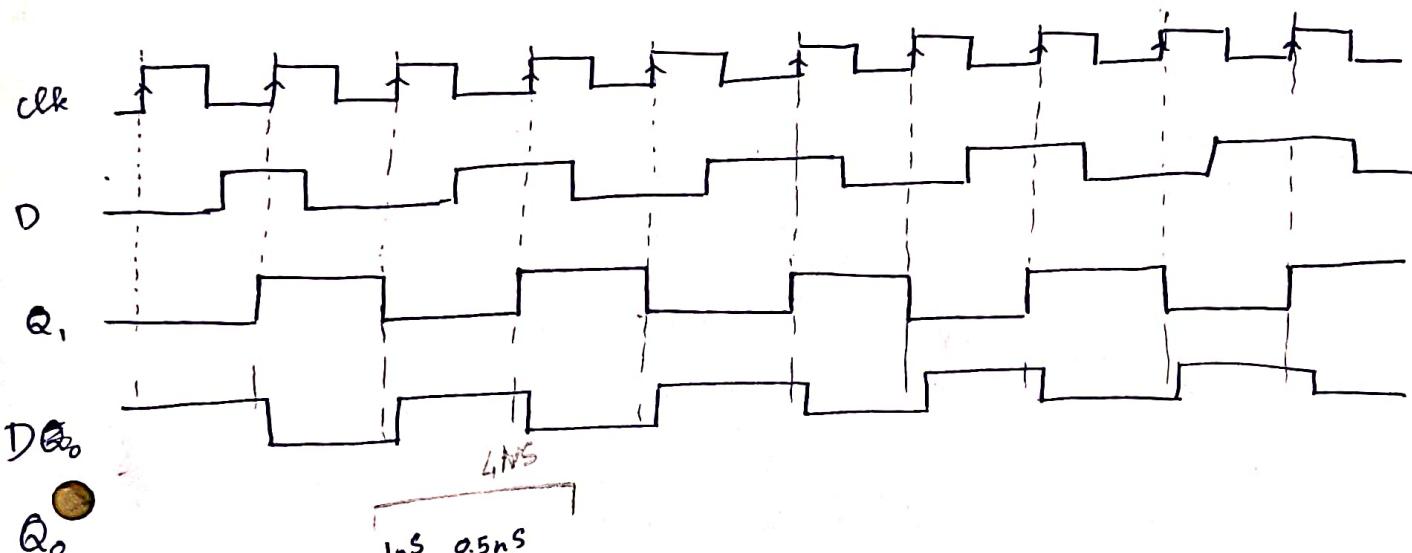
$$t_{clog_1} = t_{clog_2} \\ = t_{etog} = 1 \text{ nS}$$

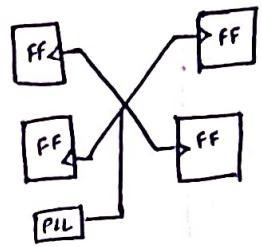
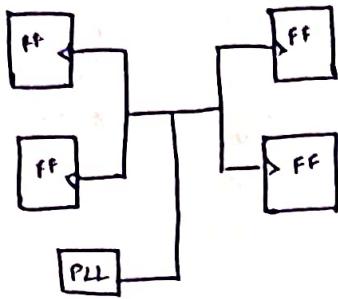
$$t_{combo} = t_{inv} = 2 \text{ nS}$$

$$t_{su_1} = t_{su_2} = t_{su} = 1 \text{ nS}$$

$$t_{th_1} = t_{th_2} = t_h = 0.5 \text{ nS}$$

Find f_{max} - max clock frequency for which the above logic circuit will work





$$T = t_{clock} + t_{combo} + t_{su} \Rightarrow 1 + 2 + 1 = 4 \text{ ns}$$

Required time

$$RT = T - t_{su_2}$$

Arrival time

$$AT = t_{clock} - t_{combo}$$

$$\text{Setup slg} = RT - AT$$

$$\text{Setup slg} \geq 0 \Leftrightarrow \min^0$$

Taking setup slg as 0

$$\Rightarrow RT - AT = 0$$

$$= (T - t_{su_2}) - (t_{clock} + t_{combo}) = 0$$

$$\Rightarrow T = t_{clock} + t_{combo} + t_{su}$$

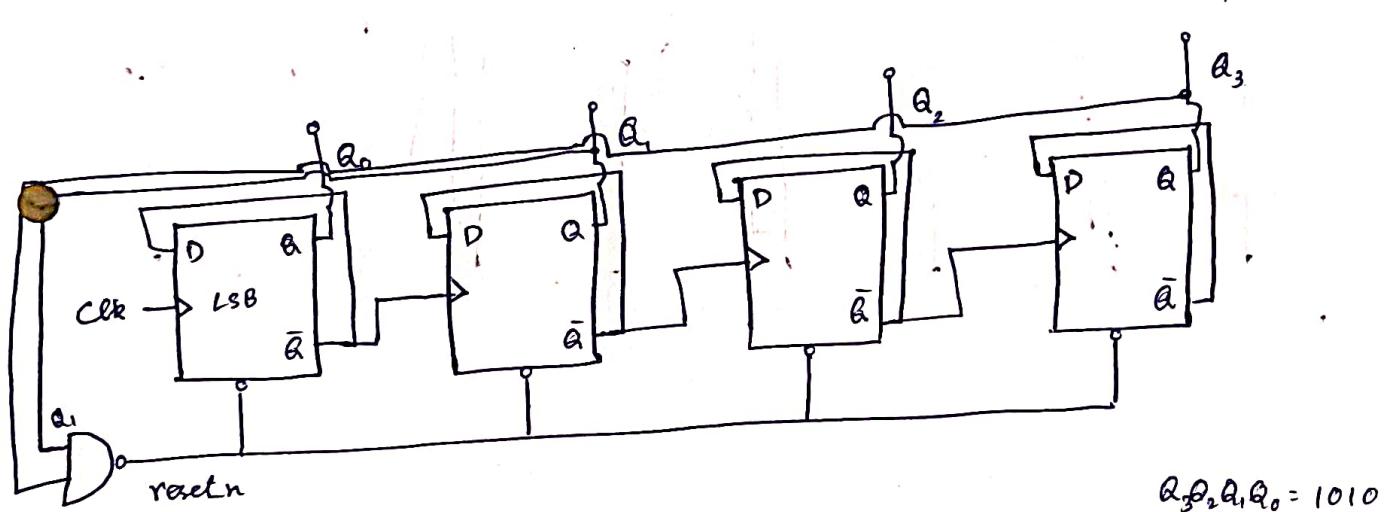
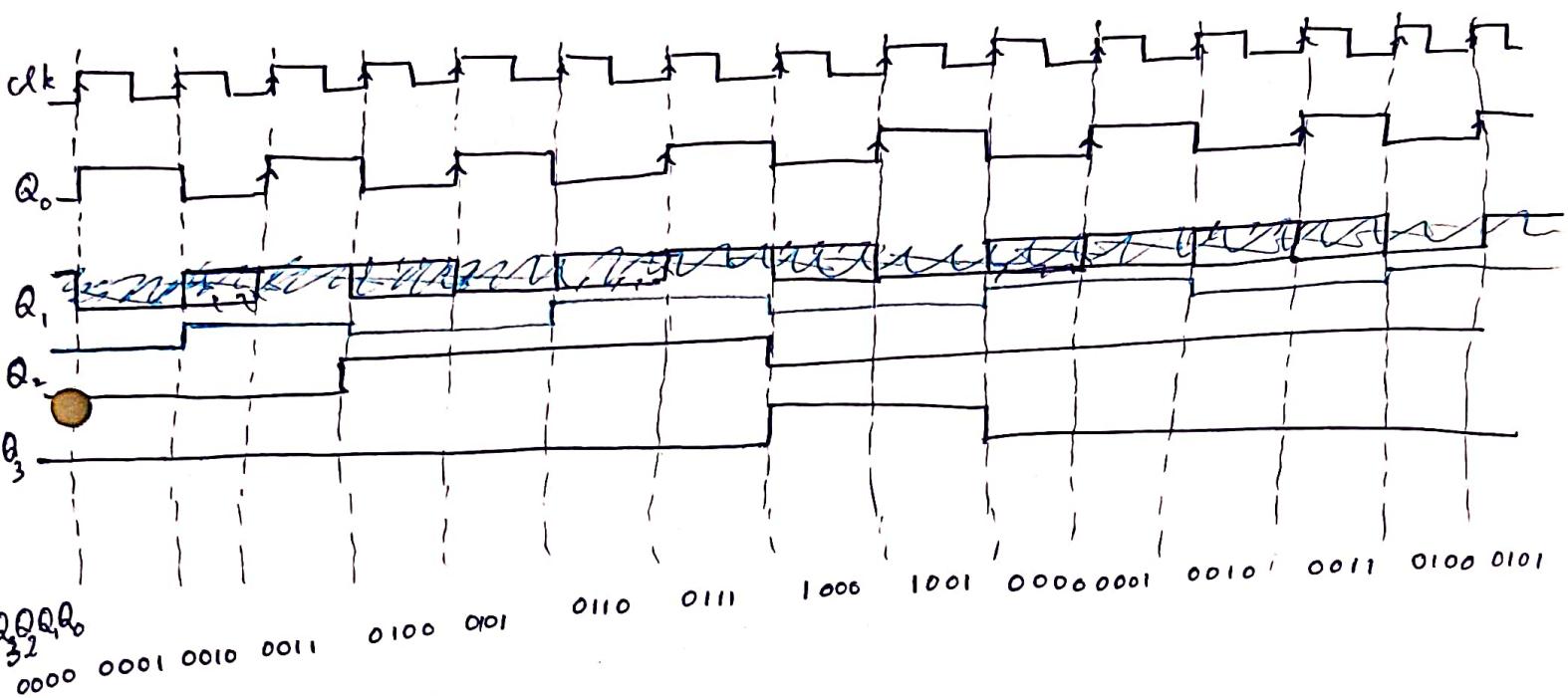
$$= 4 \text{ ns}$$

$$\Rightarrow f_{max} = \frac{1}{T}$$

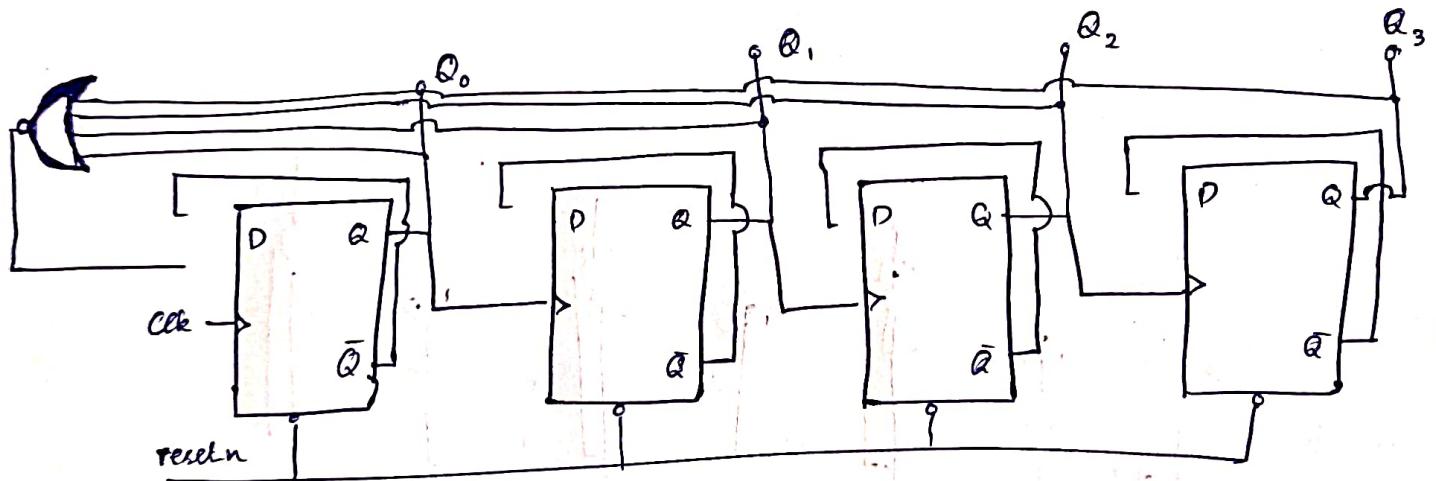
$$= \frac{1}{4 \text{ ns}}$$

$$= 250 \text{ MHz}$$

Design BCD up-counter using posedge sensitive D flipflop & minimum number of logic gates use sync active low resetn



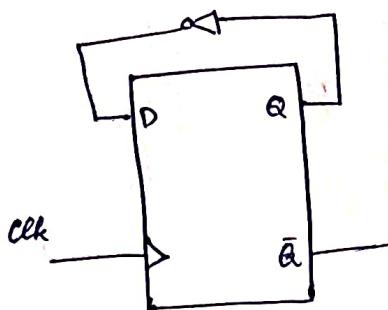
$$Q_3Q_2Q_1Q_0 = 1010$$



14.09.2024

Exercise 55

Find $f_{max} = ?$



$$t_{ctog} = 1 \text{ ns}$$

$$t_{su} = 1 \text{ ns}$$

$$t_n = 0.5 \text{ ns}$$

$$t_{combo} = 1 \text{ ns}$$

$$AT = T_{ctog} + t_{combo}$$

$$RT = T_{clk} - t_{su2}$$

Setup time $\Rightarrow RT - AT$

$$0 = T_{clk} - t_{su2} - t_{ctog} + t_{combo}$$

$$T_{clk} = t_{su} + t_{ctog} + t_{combo}$$

$$= 1 + 1 + 1$$

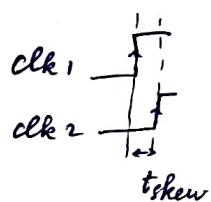
$$T_{clk} = 3 \text{ ns}$$

$$f_{clk} = \frac{1}{T_{clk}}$$

$$= 333.3 \text{ MHz}$$

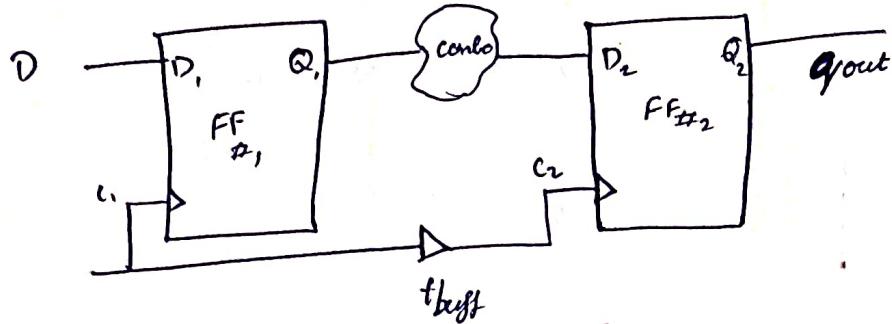
Clock Skew

The difference between arrival
of clock edges



Exercise 56

Find $t_{max} = ?$



$$\begin{aligned} T_{clock} &= \downarrow \\ T_{su} &= \downarrow \\ T_h &= \downarrow \\ T_{combo} &= \downarrow \end{aligned}$$

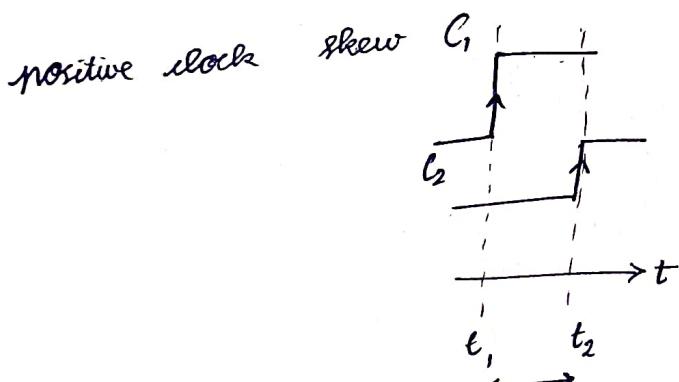
$$\begin{aligned} t_{clock} &= 1 \text{ ns} & t_{combo} &= 2 \text{ ns} \\ t_{su} &= 1 \text{ ns} & t_h &= 0.5 \text{ ns} \\ t_h &= 0.5 \text{ ns} & t_{buff} &= 1 \text{ ns} \end{aligned}$$

$$\begin{aligned} AT &= t_{clock} + t_{combo} \quad \checkmark \quad \xrightarrow{\text{with skew}} t_{clock} + t_{su} + t_{buff} \\ RT &= T_{clock} - t_{su} \cancel{+ t_{buff}} \end{aligned}$$

$$T_{clock} = t_{clock} + t_{combo} + t_{su} + t_{buff}$$

$$= \cancel{5 \text{ ns}} \quad 1 \text{ ns} \quad \cancel{5.5 \text{ ns}} \quad 181.8 \text{ MHz}$$

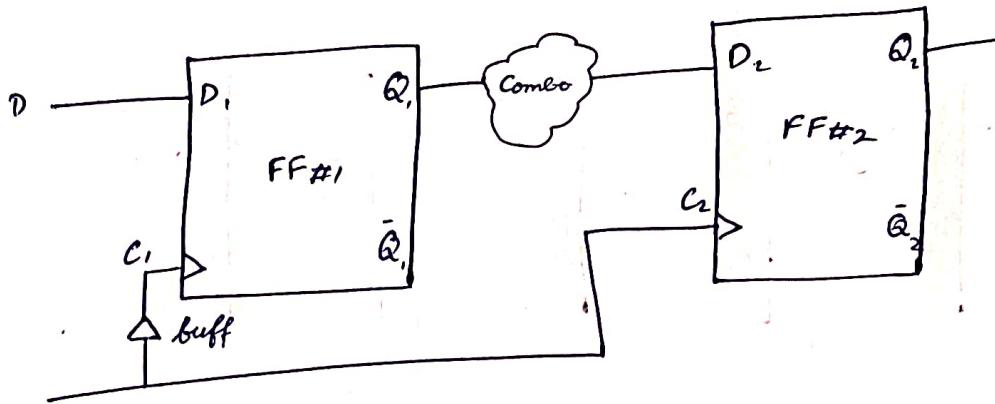
$$\begin{aligned} T_{clock} &= \cancel{200 \text{ MHz}} \quad \checkmark \quad \cancel{t_{clock}} = 1 + 2 + 1 - 1 \\ &= \cancel{200 \text{ MHz}} \quad \cancel{t_{su}} = 3 \text{ ns} \Rightarrow 333.3 \text{ MHz} \quad \xrightarrow{\text{with skew}} 250 \text{ MHz} \end{aligned}$$



clock arrive after data \rightarrow clock & data travel in same direction.

Exercise 5.1

Find $f_{max} = ?$



$$t_{clock} = 1 \text{ ns}$$

$$t_{tsu} = 1 \text{ ns}$$

$$t_{combo} = 2 \text{ ns}$$

$$t_n = 0.5 \text{ ns}$$

$$t_{buff} = 1 \text{ ns}$$

$$RT = T_{clk} - t_{tsu}$$

$$AT = t_{clock} + t_{buff} + t_{combo}$$

$$T_{clk} = t_{clock} + t_{buff} + t_{combo} + t_{tsu}$$

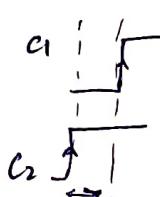
$$= 1 + 1 + 1 + 2$$

$$T_{clk} = 5 \text{ ns}$$

$$f = \frac{1}{T} = \frac{1}{5} \times 10^9$$

$$f = 200 \text{ MHz}$$

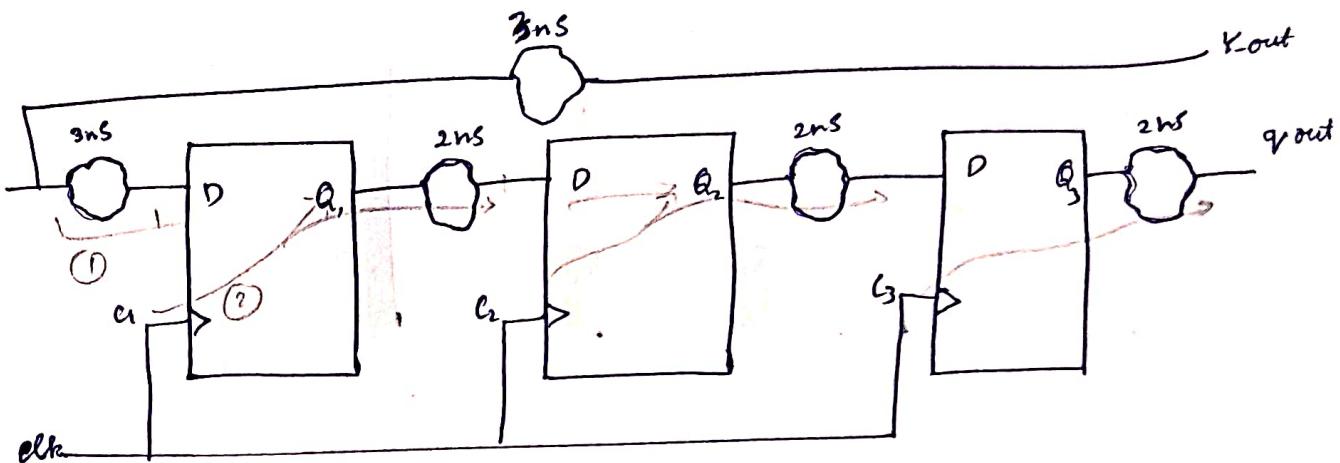
Negative clock skew



clock of first flop flop is delayed derivation triggered first

Exercise 58

Find t_{max} ?



$$t_{cbog} = 1 \text{ ns}$$

$$t_{su} = 1 \text{ ns}$$

$$t_h = 0.5 \text{ ns}$$

$$RT = T_{clk} - t_{setup} \Rightarrow \text{remains unchanged}$$

$$AT = t_{cbog} + t_{combo}$$

$$= 1 + 2$$

$$AT = 3 \text{ ns (max)}$$

$$\text{Setup slg} = RT - AT \geq 0$$

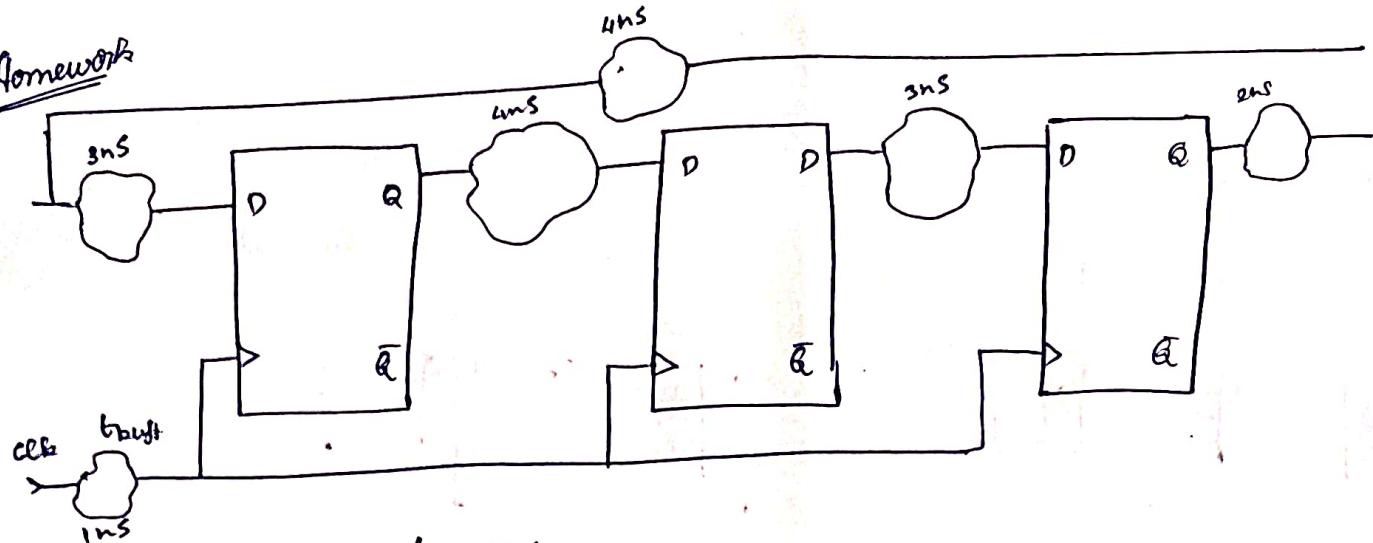
$$T - t_{setup} - 3 \text{ ns} \geq 0$$

$$T = 4 \text{ ns}$$

$$f = \frac{1}{T} = 250 \text{ MHz}$$

Timing paths in the design

Homework



$$t_{su} = 1\text{ ns}$$

$$t_h = 0.1\text{ ns}$$

$$t_{clock} = 1\text{ ns}$$

$$t_{buff} = 1\text{ ns}$$

$$RT = T_{clock} - t_{su}$$

$$AT = t_{buff} + t_{clock} + t_{combo}$$

$$= 1 + 1 + 4$$

$$= 6\text{ ns}$$

$$RT - AT \geq 0$$

$$T - t_{setup} - 6\text{ ns} \geq 0$$

$$T = 7\text{ ns}$$

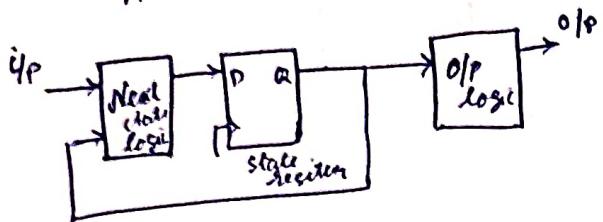
$$f = \frac{1}{T}$$

$$= 142.9\text{ MHz}$$

FSM - Finite State Machine

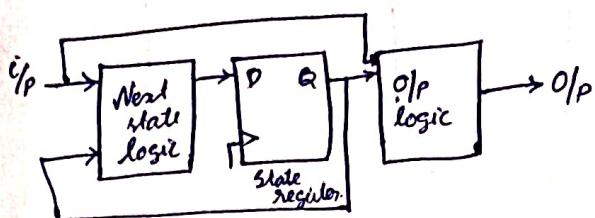
Moore machine.

$$O/P = f(\text{Current state})$$



Mealy machine

$$O/P = f(\text{Current state}, i/p)$$



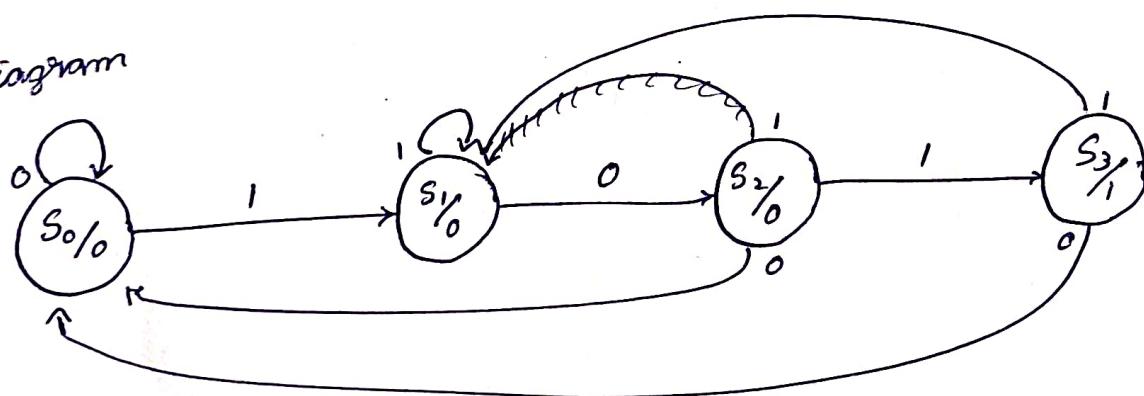
Applications

- sequence detector, control unit design. depending on requirements

In state machines next state logic is combinational logic
& output logic is combinational logic

Design 101 non-overlapping sequence detector using Moore machine FSM

State diagram



State table & excitation table

i/p Din	Present state		Next state		Excitation i/p		o/p q
	Q_1	Q_0	Q_1^+	Q_0^+	D_1	D_0	
0	0	0	0	0	0	0	0
1	0	0	0	1	0	1	0
0	0	1	1	0	1	0	0
1	0	1	0	1	0	1	0
0	1	0	0	0	0	0	0
1	1	0	1	1	1	1	0
0	1	1	0	0	0	0	1
1	1	1	0	1	0	1	1

Din	Q_1 , Q_0
0	1 0 0 0
0	0 0 0 1

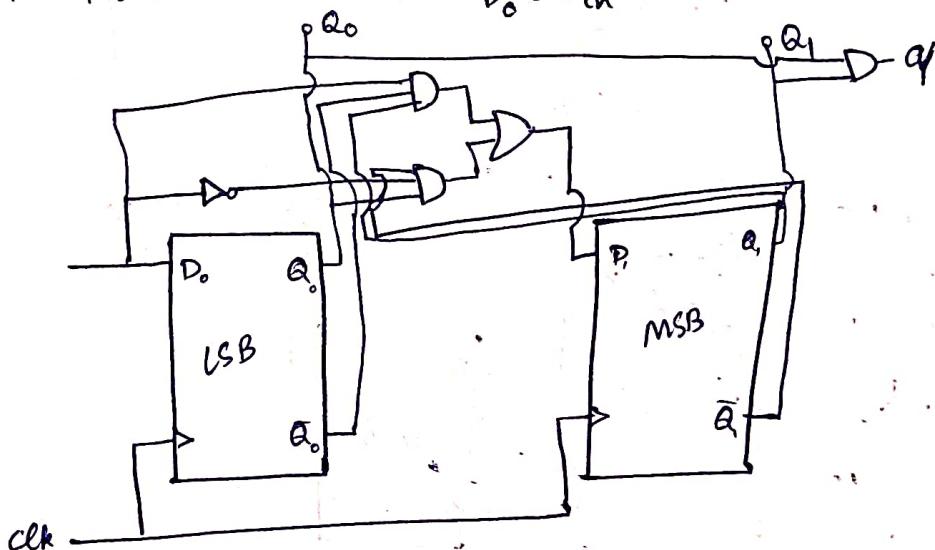
Din	Q_1 , Q_0
0	0 0 0 0
1	1 1 1 1

Q_1 , Q_0	o/p
0 0	0
0 1	1

$$q = Q_1, Q_0$$

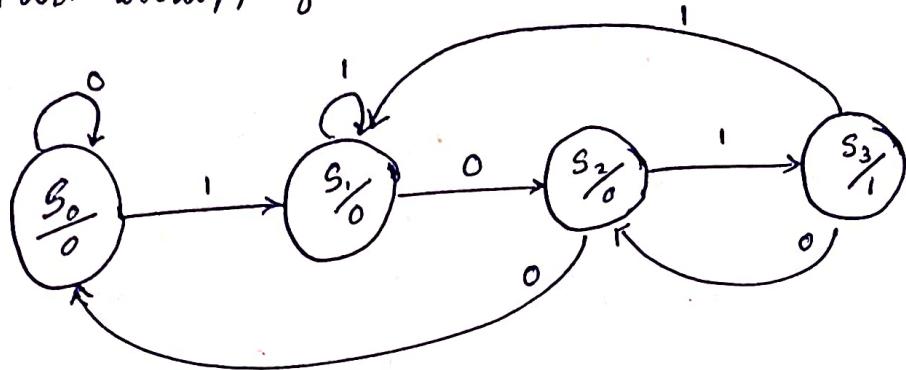
$$D_1 = \bar{D} \bar{Q}_1 Q_0 + D Q_1 \bar{Q}_0$$

$$D_0 = \bar{D}_{in}$$



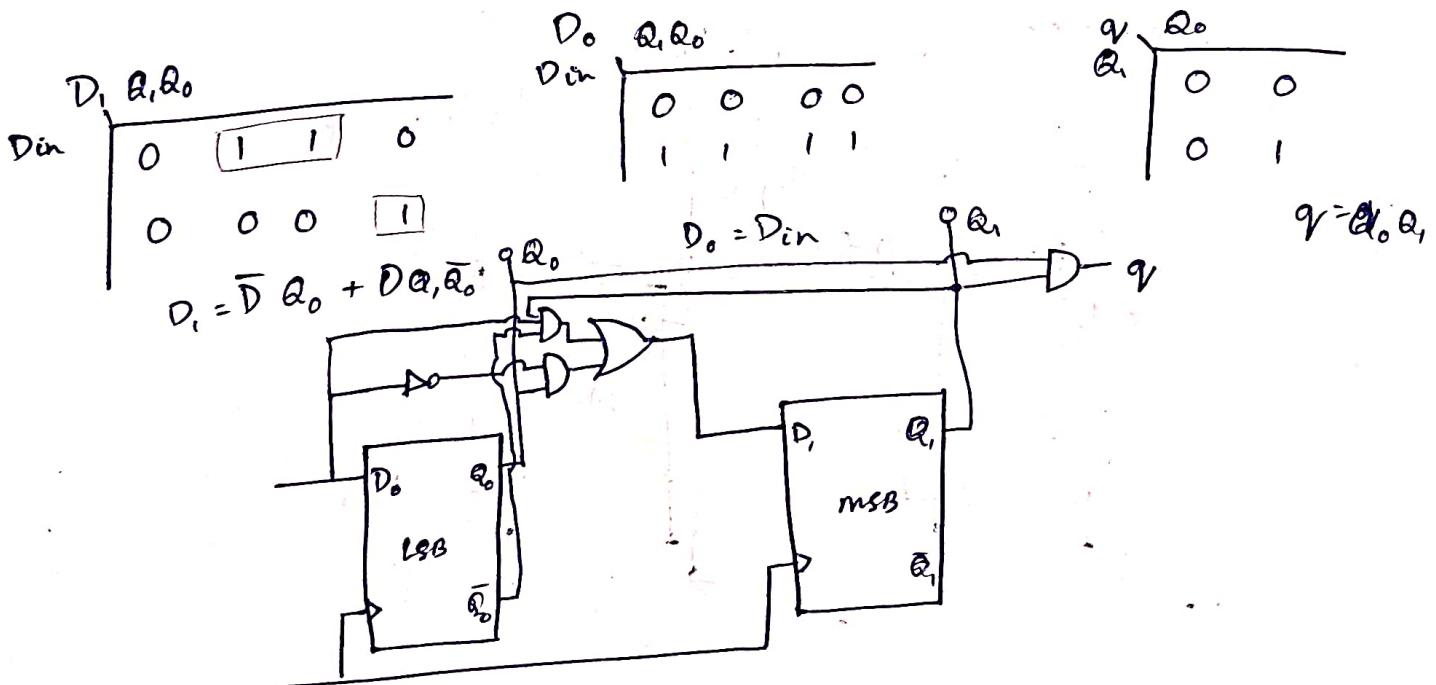
Exercise -

Design 1 bit overlapping Moore machine to detect sequence 101



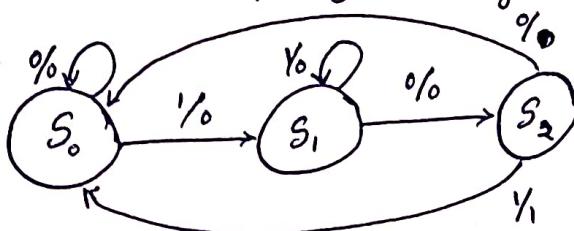
state table & Excitation table

D _{in}	Present state		Next state		Excitation i/p		q
	Q ₁	Q ₀	Q ₁ ⁺	Q ₀ ⁺	D ₁	D ₀	
0	0	0	0	0	0	0	0
1	0	0	0	1	0	1	0
0	0	1	1	0	1	0	0
1	0	1	0	1	0	1	0
0	1	0	0	0	0	0	0
1	1	0	1	1	1	1	0
0	1	1	1	0	1	0	1
1	1	1	0	1	0	1	1



Exercise 60

Draw 101 non-overlapping mealy sequence detector state diagram & design



state table & excitation table

din	Present state		Next state		Excitation IP		q1
	Q1	Q0	Q1 ⁺	Q0 ⁺	D1	D0	
0	0	0	0	0	0	0	0
0	0	0	0	1	0	1	0
0	0	1	1	0	1	0	0
1	0	1	0	1	0	1	0
0	1	0	0	0	0	0	0
1	1	0	0	0	0	0	1

$$\begin{array}{|c|c|c|} \hline D_1 & Q_1 & Q_0 \\ \hline 0 & 1 & \times 0 \\ 0 & 0 & \times 0 \\ \hline \end{array}$$

$$D_1 = \overline{D_{in}} \overline{Q}_1 S_0$$

$$= \overline{D_{in}} Q_0$$

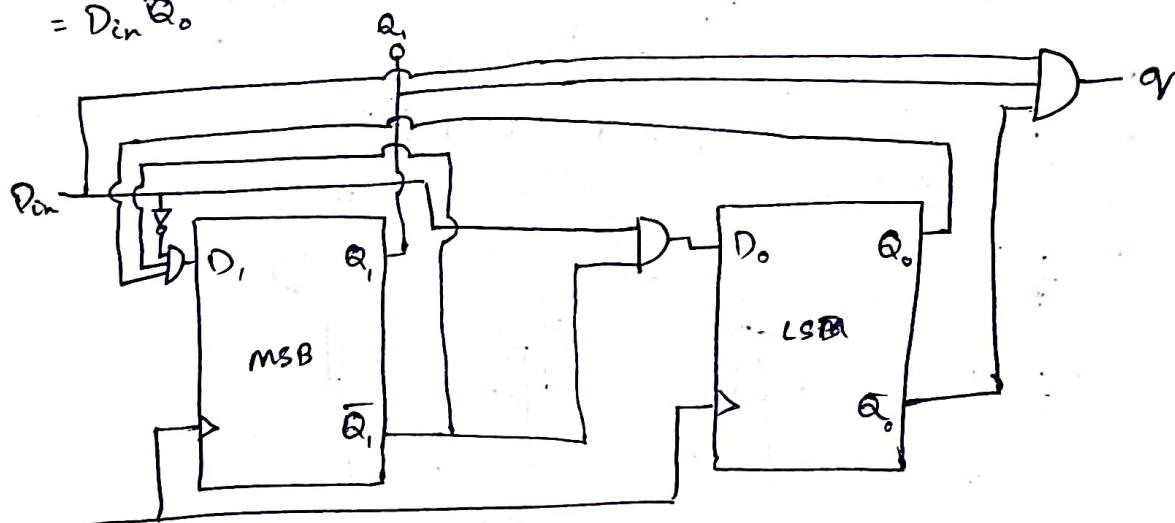
$$\begin{array}{|c|c|c|} \hline D_0 & Q_1 & Q_0 \\ \hline D_{in} & 0 & 0 & \times 0 \\ 1 & 1 & \times 0 \\ \hline \end{array}$$

$$D_0 = D_{in} \overline{Q}_1$$

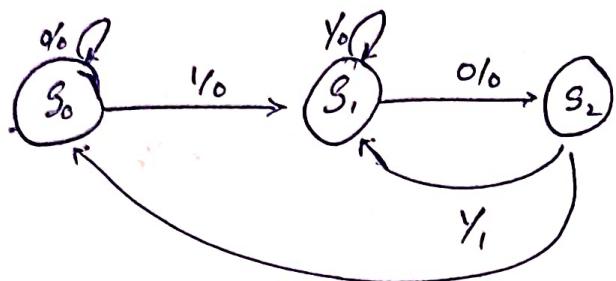
$$\begin{array}{|c|c|c|} \hline D_{in} & Q_1 & Q_0 \\ \hline 0 & 0 & \times 0 \\ 0 & 0 & \boxed{\times 1} \\ \hline \end{array}$$

$$q_1 = D_{in} Q_1 \overline{Q}_0$$

$$= D_{in} Q_1$$



To detect 101 overlapping sequence sketch a ^{mealy} state diagram & design mealy machine for 1-bit overlapping sequence.



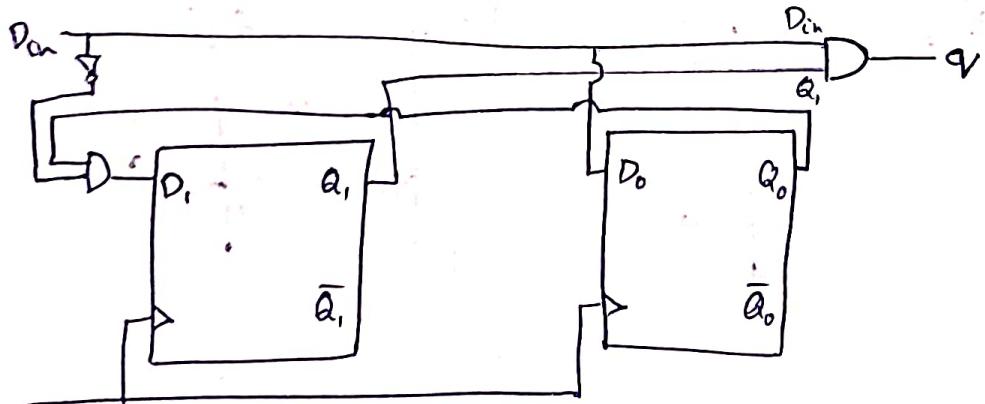
state table & excitation table

y _p P _{in}	Present state		Next state		Excitation d/p		O/p Y _V
	Q ₁	Q ₀	Q ₁ ⁺	Q ₀ ⁺	D ₁	D ₀	
0	0	0	0	0	0	0	0
1	0	0	0	1	0	1	0
0	0	1	1	0	1	0	0
1	0	1	0	1	0	1	0
0	1	0	0	0	0	0	0
1	1	0	0	1	0	1	1

$$\begin{array}{cccc} D_1 & & & \\ \hline 0 & 1 & x & 0 \\ 0 & 0 & x & 0 \end{array} \Rightarrow \overline{D_{in}} Q_0$$

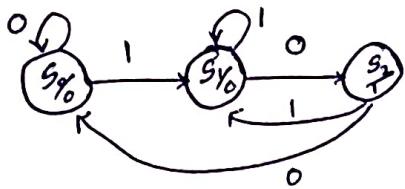
$$\begin{array}{cccc} D_0 & & & \\ \hline 0 & 0 & x & 0 \\ 1 & 1 & x & 1 \end{array} \Rightarrow D_{in}$$

$$\begin{array}{cccc} q & & & \\ \hline 0 & 0 & x & 0 \\ 0 & 0 & x & 1 \end{array} = D_{in} Q_1$$



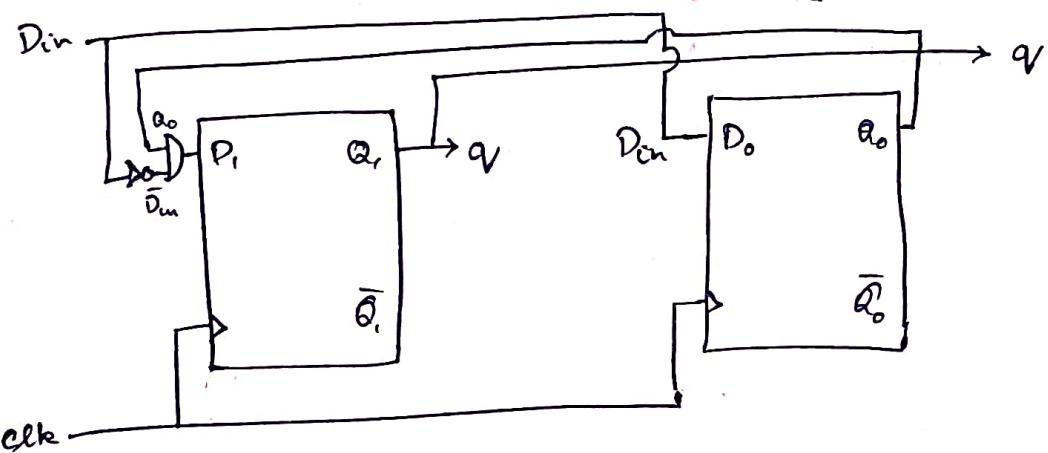
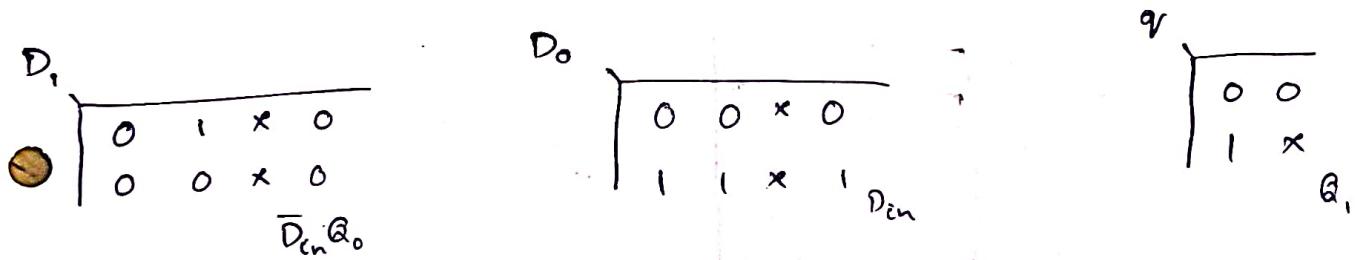
Exercise.62

Design moore machine to detect sequence 10



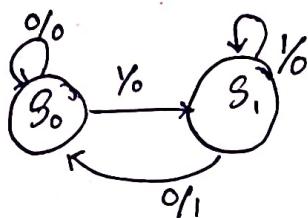
state table & excitation table

D _{in}	Present state		Next state		Excitation i/p		Q/P
	Q ₁	Q ₀	Q ₁ ⁺	Q ₀ ⁺	D _i	D _o	
0	0	0	0	0	0	0	0
1	0	0	0	1	0	1	0
0	0	1	1	0	1	0	0
1	0	1	0	1	0	1	1
0	1	0	0	0	0	0	1
1	1	0	0	1	0	1	1



Exercise 6.3

Design the mealy machine to detect 10 sequence

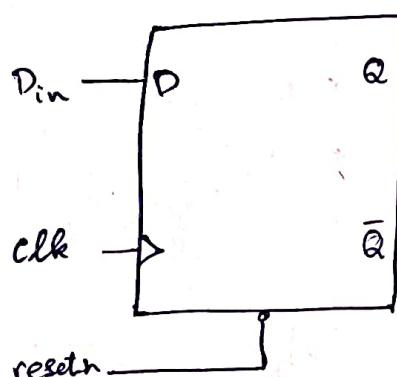


state diagram & Excitation table

i/p Din	Present state Q	Next state Q^+	Excitation i/p D	o/p q
0	0	0	0	0
1	0	1	1	0
0	1	0	0	1
1	1	1	1	0

$$D \begin{bmatrix} 0 & x & x & 0 \\ x & 1 & 1 & x \end{bmatrix} \Rightarrow D_{in}$$

$$q \begin{bmatrix} 0 & x & x & 1 \\ x & 0 & 0 & x \end{bmatrix} \Rightarrow -$$



15-09-2024

Hazards in digital circuits.

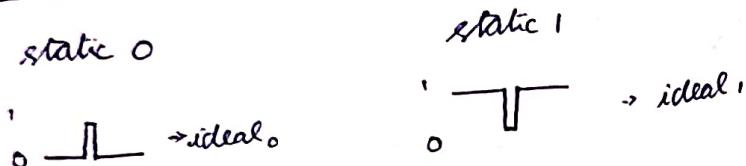
- un-wanted transitions at o/p
- may be due to glitch.

Types of hazards

- static
- dynamic

- in combinational circuit \rightarrow finite propagation delay

Static

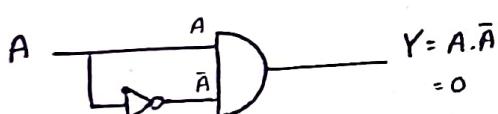


Dynamic

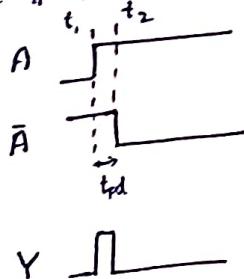
L-L-L \rightarrow not constant

\rightarrow usually in large density combinational circuit

Static 0 example

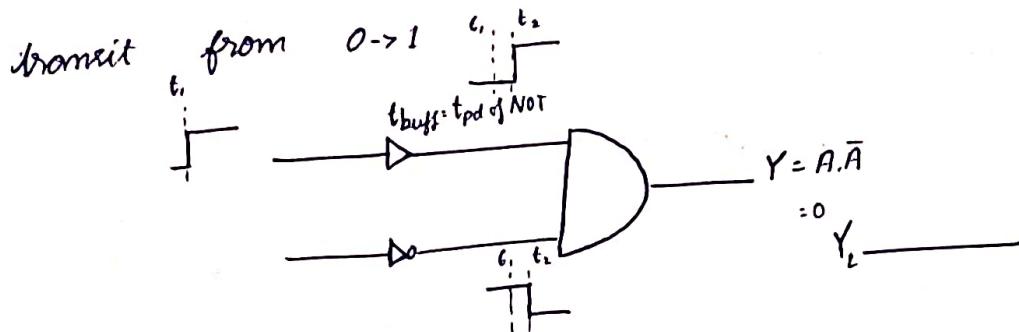


@ t_p AND gate

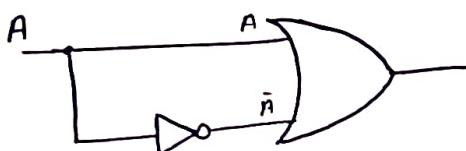


\rightarrow O/p should be 0 ideally
 \rightarrow but, for t_{pd} duration O/p transition observed
 (i.e.) static 0 hazard.

If we can insert a buffer in data path & we balance the delay we can get a valid o/p $A \cdot \bar{A} = 0$ whenever A

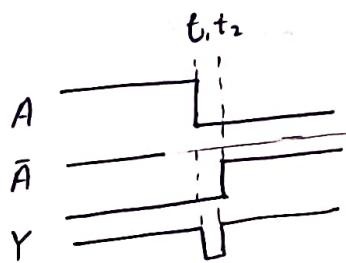


Static!



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

= 1

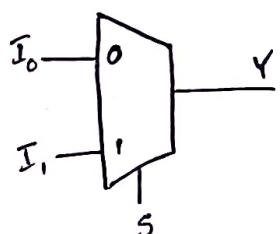


→ similarly adding t_{buff} at i/p of OR eliminate static t_{pd}

Exercise 64

Check for hazards & suggest how to eliminate function hazard

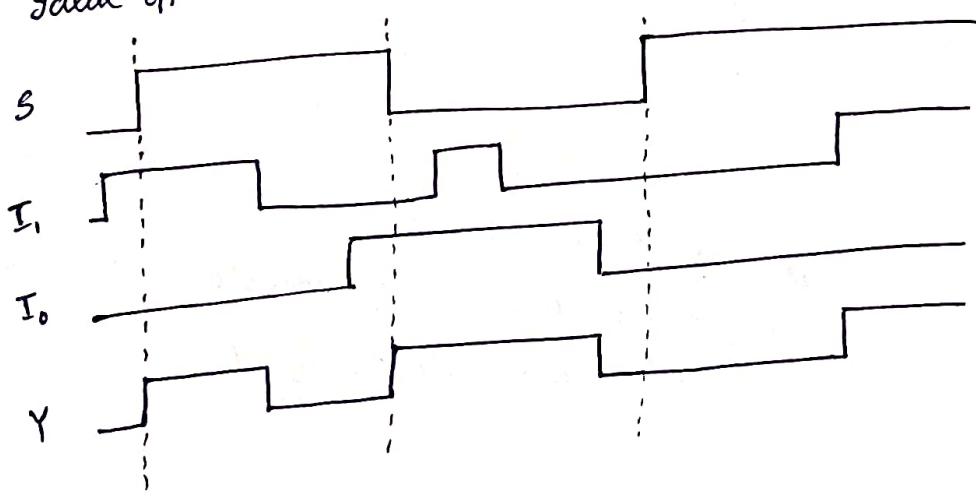
hazard



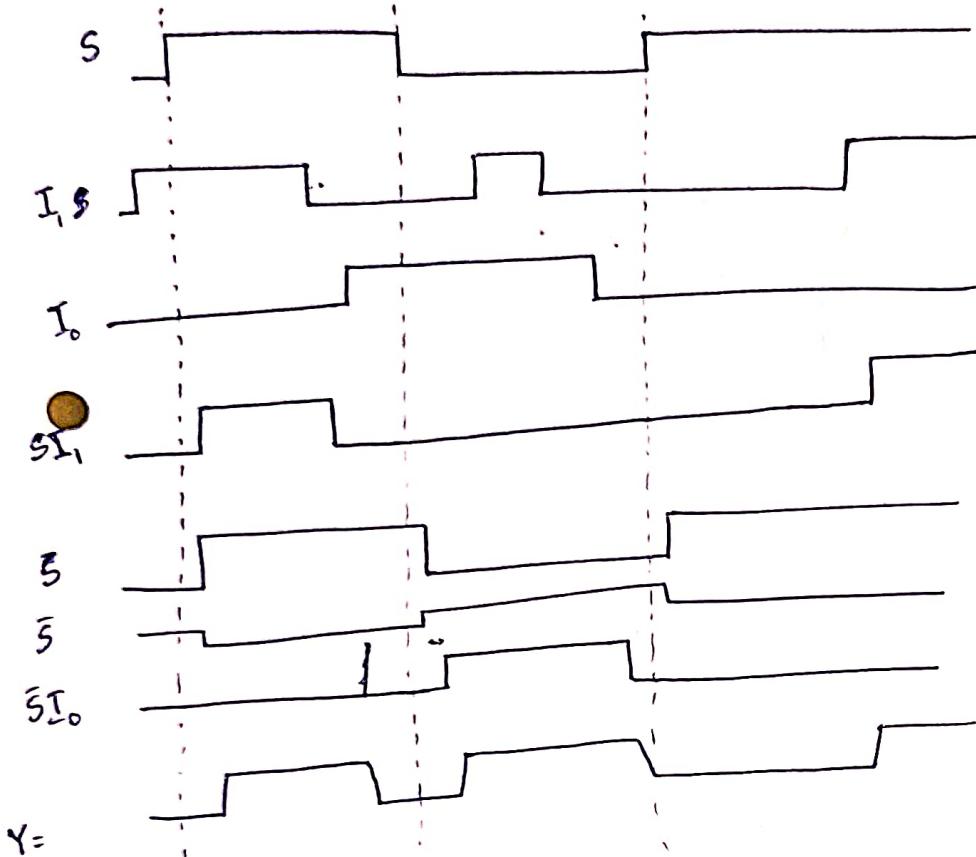
S	Y
I0	I0
I1	I1

$$Y = \bar{S} I_0 + S I_1$$

Ideal o/p



→ By eliminating static 0 or static 1 hazards dynamic hazards will be eliminated. by modifying circuit



→ O/p delayed by propagation delay of AND gate $t_{pd} = 1\text{ns}$

→ O/p delayed by t_{pd} of NOT $= 1\text{ns}$

→ delayed o/p

S	I_1	I_o	Y
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

S	I_1, I_o	Y
0	0 1 1 0	0
0	0 0 1 1	1

To eliminate hazard

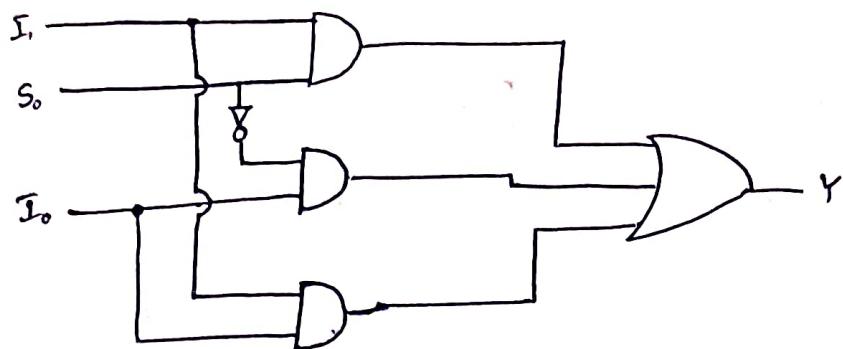
S	I_1, I_o	Y
0	0 1 1 0	0
0	0 0 1 1	1

$\Rightarrow \bar{S}I_o + SI_1$

To compensate for transmission static 1 hazard

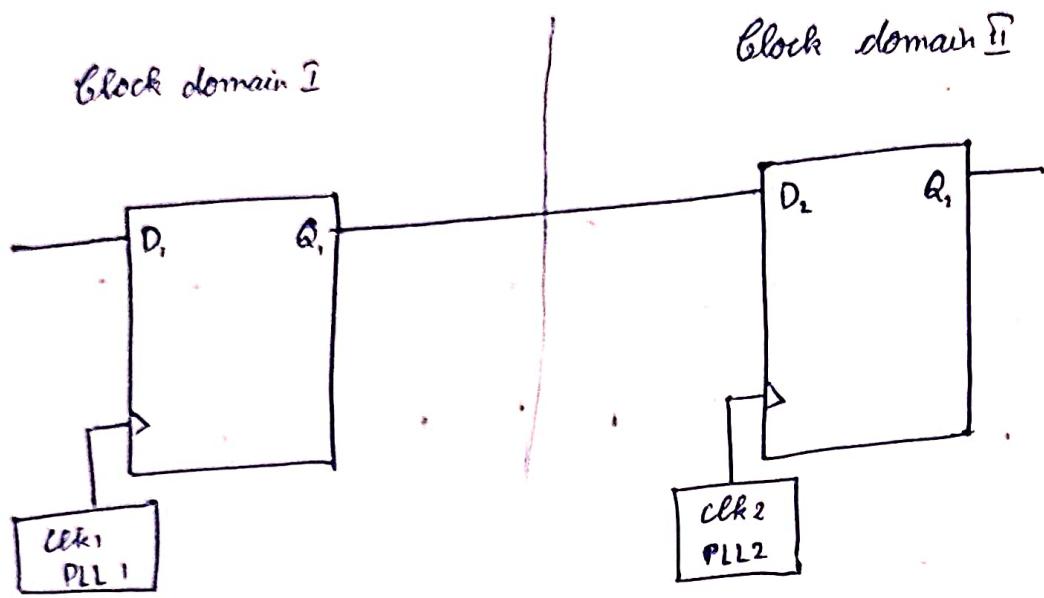
Homework

Analyze the hazards in 2:1 MUX & to eliminate hazard include extra logic gate as shown & check for all & condition of truth table & prove that designed ckt is hazard free

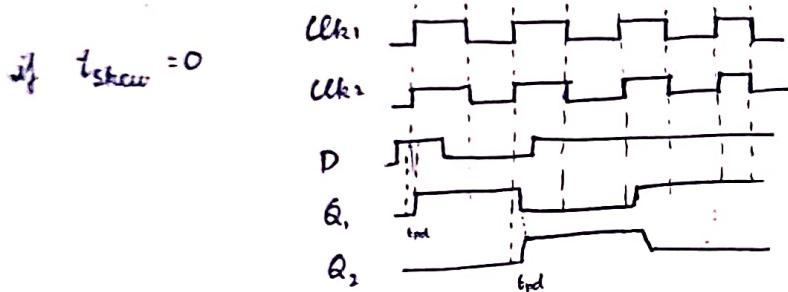


Application of hazard elimination

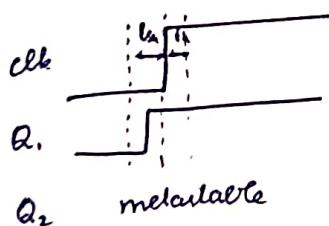
→ Multiple clock domain design.



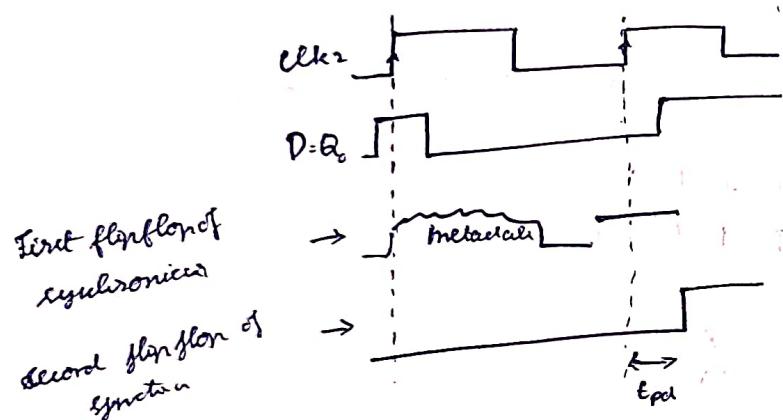
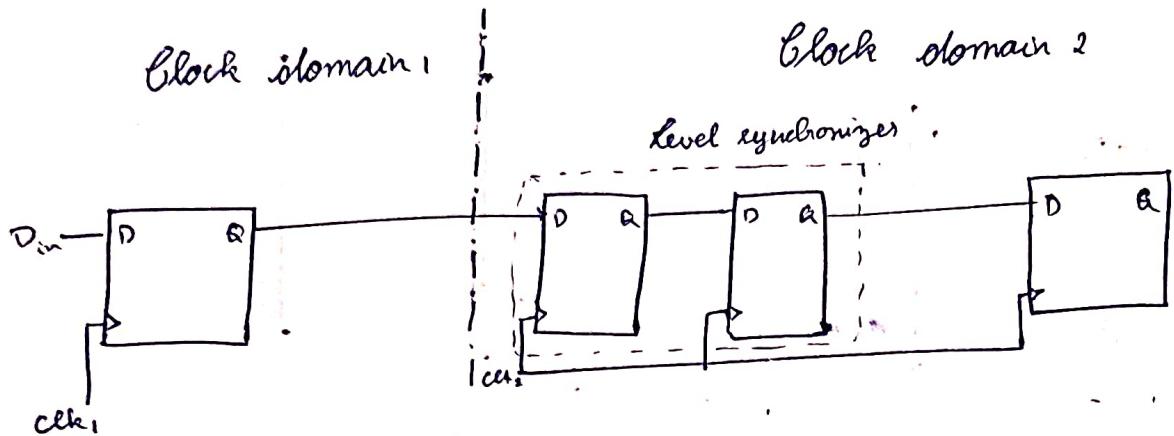
→ clock domain crossing.



practically $t_{skew} = 0$ not possible



To remove metastability while passing one clock domain to another clock domain use a level synchronizer.



In clock domain crossing - for control path we can use level synchronizer & in data path we can use FIFO synchronizer

18-09-2024

Other synchronizing methods

- MVX
- handshaking mechanism.
- tracing multiple signals from CD I \rightarrow II

Control path

- 2 FF (or) 3 FF level synchronizer

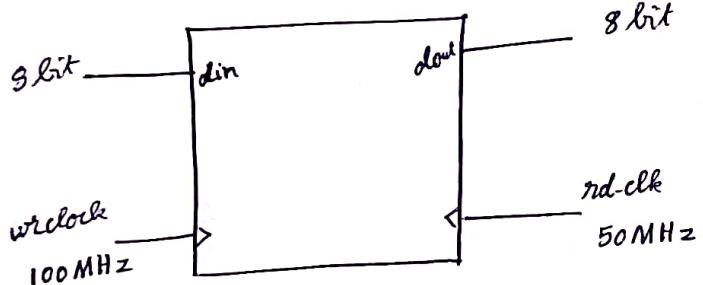
Data path

- FIFO \rightarrow first in first out

Development of $\mu P \rightarrow$ to see later

Exercise 65

What should be depth of FIFO if write-clock = 100 MHz, rd-clock = 50 MHz
 & no. of data bytes 100 to be transferred from clock domain I to II



A A	
A A	
B B	
C C	
D D	
C C	
C C	
D D	
D D	

1 byte = 8 bits

Time period of 100 MHz $\Rightarrow 10 \text{ ns}$; Burst length = No of bytes $T_{\text{req}} = 100 \times 10 \text{ ns} = 1000 \text{ ns}$

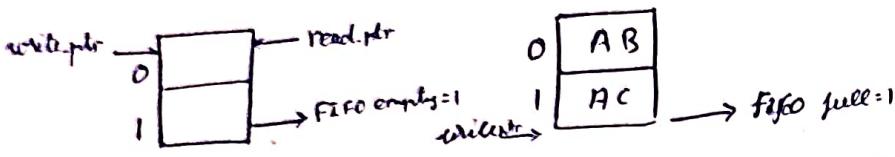
T_{read} - - - $50 \text{ MHz} \Rightarrow 20 \text{ ns}$; Freq to read (f_r) = 2000 ns

No. of reads in 1000ns $\Rightarrow n = \frac{T_{\text{burstwrite}}}{T_r}$

$$n = \frac{1000 \text{ ns}}{20 \text{ ns}} = 50$$

$$\begin{aligned}
 \text{To avoid data loss, depth of FIFO} &= 100-n \\
 &= 100-50 \\
 &= 50
 \end{aligned}$$

FIFO empty, FIFO full logic



$\text{if } ([\lceil \text{wr.ptr} \rceil], \text{wr.ptr}[0]) == \text{rd.ptr}$
 then
 FIFO.full = 1
 else
 FIFO.full = 0

Exercise 6b

What should be depth of FIFO if wrclk = 80 MHz, rdclk = 50 MHz & no of data bytes 120 to be transferred from clock domain I to clock domain II

T

$$T_{wr} = 12.5 \text{ ns}$$

$$T_{burst} = 1500 \text{ ns}$$

$$T_{rd} = 20 \text{ ns}$$

$$\begin{aligned}
 \text{No of reads in } 1500 \text{ ns} &= \frac{1500}{20} \\
 &= 75
 \end{aligned}$$

$$\begin{aligned}
 \text{Depth of FIFO} &= 120 - 75 \\
 &= 45
 \end{aligned}$$

Exercise 6c

What should be depth of FIFO if wrclk = 50 MHz, rdclk = 80 MHz & no of data bytes 120 to be transferred from clock domain I \rightarrow II

\rightarrow No data loss or read speed higher
one data buffer is sufficient.

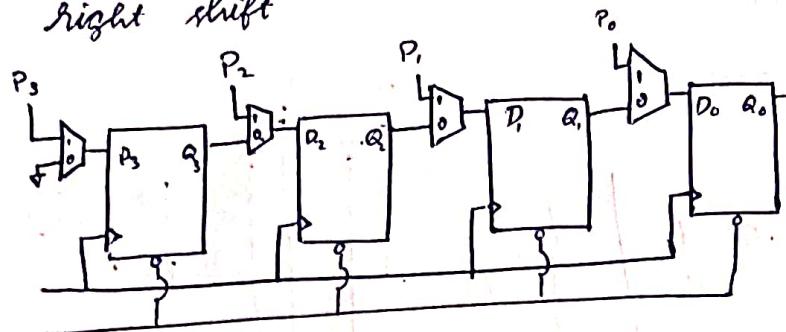
Register \rightarrow storage element

• PIPD, SISO

• Shift register

- load data & shift data

right shift

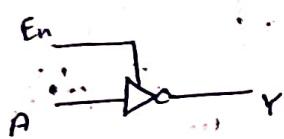


Homework

left shift, LIFO

20-09-2024

Tristate

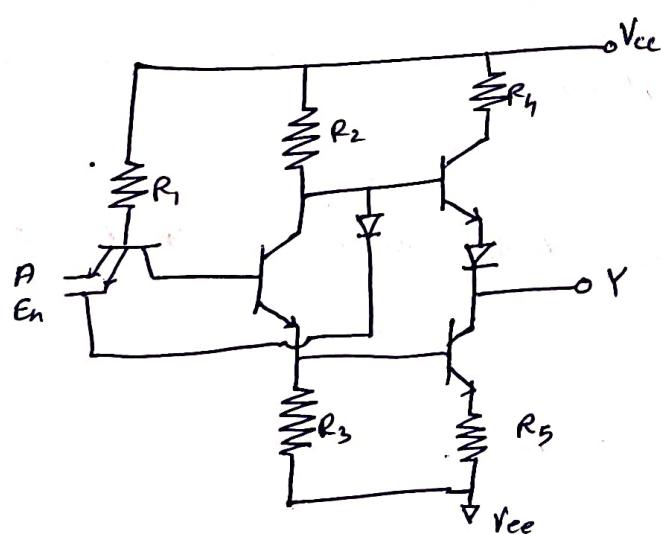


En	A	Y
0	x	Z'
1	0	1
1	1	0

Tristate logic & application of tristate bus.

\rightarrow $I_{O,Bus}$, rw lines

Tristate from standard TTL



The extra diode offsets biasing when $E_n = 0$ thus making output a high impedance state.

Logic array

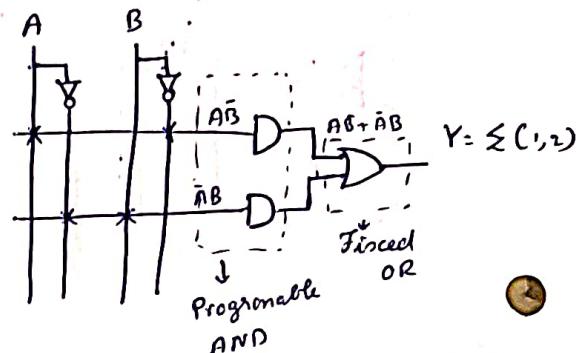
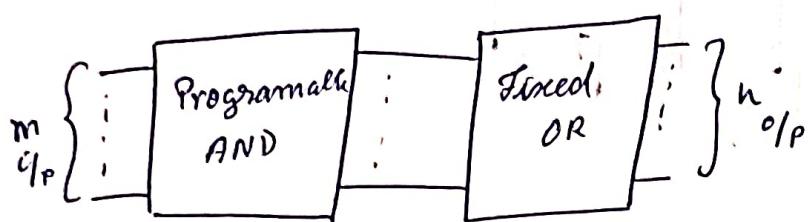
PLD - Programmable logic devices.

↳ PAL, PLA, CPLD, FPGA

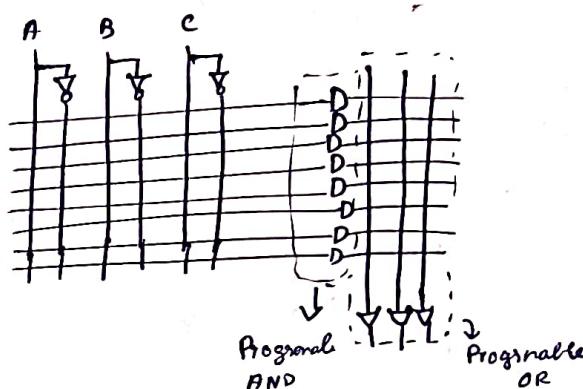
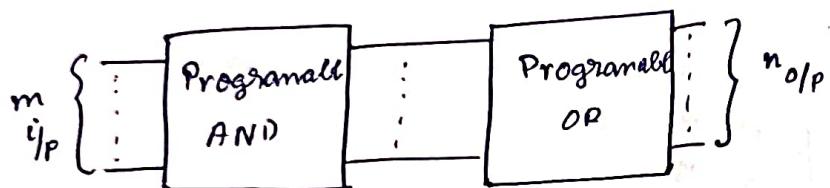
10k + gates
→ Multiple PLA
Registers

CLB
BRAM
DSP
Registers
LUT

PAL - Programmable array logic



PLA - Programmable logic array



Tabulation method

Quine-McCluskey method

→ it is a tabulation method if more than 4 variables are there then this method is suitable for boolean expression simplification.

Step 1: arrange the following min term according to no. of 1's in ascending order.

Step 2: Take min. terms in continuous group & make pairs

Step 3: use '·' symbol where one bit change & keep remaining as same

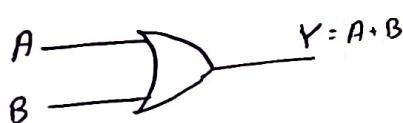
Step 4: Repeat above 2 steps until we get prime implicants
(i) all bits present are different

Step 5: Make a prime implicant table which consists of obtaining min terms as rows & given minterm as columns

Step 6: + '·' in min term which are covered by prime implicant

Step 7: Add the essential prime implicant to final formula

For example consider OR gate



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

Table 2

group	Decimal No	A	B
0	0	0	0
	1	0	1
1	2	1	0
2	3	1	1

group	A	B
(0, 1)	0	-
(0, 2)	-	0
(2, 3)	1	-
(1, 3)	-	1

$$\begin{aligned} \rightarrow & \text{ min terms} \\ \rightarrow & \text{ prime implicant} \\ \downarrow & A\bar{B} + AB = A \\ \downarrow & \bar{A}\bar{B} + A\bar{B} = B \end{aligned}$$

$$\begin{array}{c|ccc} & 1 & 2 & 3 \\ \hline 1 & & & \\ 1 & & & \\ & & & \\ \hline & A+B & & \end{array}$$

$$f(x, y, z, w) = \sum(0, 1, 2, 4, 6, 8, 9, 11, 13, 15)$$

group	Minterms	x	y	z	w
0	0	0	0	0	0
1	1	0	0	0	1
	2	0	0	1	0
	4	0	1	0	0
	8	1	0	0	0
2	6	0	1	1	0
	9	1	0	0	1
3	11	1	0	1	1
	13	1	1	0	1
4	15	1	1	1	1

group	Pair	x	y	z	w
0	(0, 1)	0	0	0	-
	(0, 2)	0	0	-	0
	(0, 4)	0	-	0	0
	(0, 8)	0	0	0	1
	(1, 9)	-	0	0	-
1	(2, 6)	0	-	1	0
	(4, 6)	0	1	-	0
	(8, 9)	1	0	0	-
2	(9, 11)	1	0	-	1
	(9, 13)	1	-	0	1
3	(11, 15)	1	-	1	-
	(13, 15)	1	1	-	1

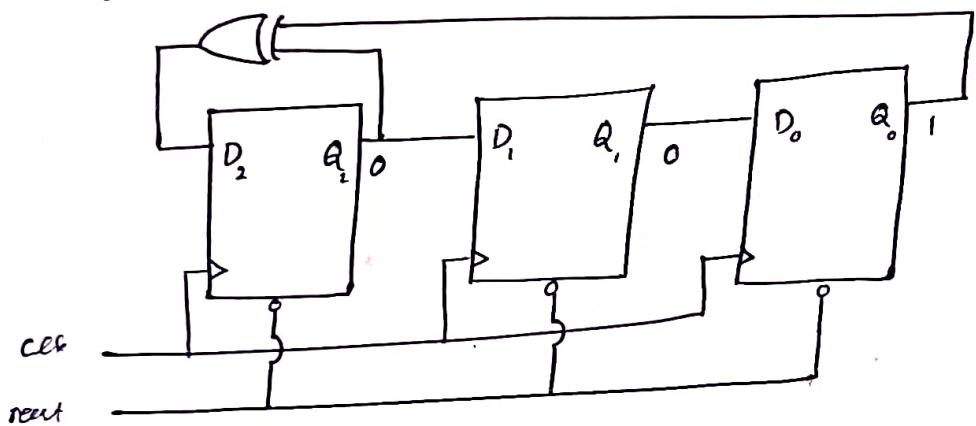
Table 3

group	quad	x	y	z	w
0	(0,1,8,9)	-	0	0	-
	(0,2,4,6)	0	-	-	0
1	(9,11,15,16)	1	-	-	1

$f = \bar{Y}\bar{Z} + \bar{X}\bar{W} + XW$

LFSRLinear Feedback Shift Register

- we can use XOR gate & flip flop as shown in following fig to get LFSR.



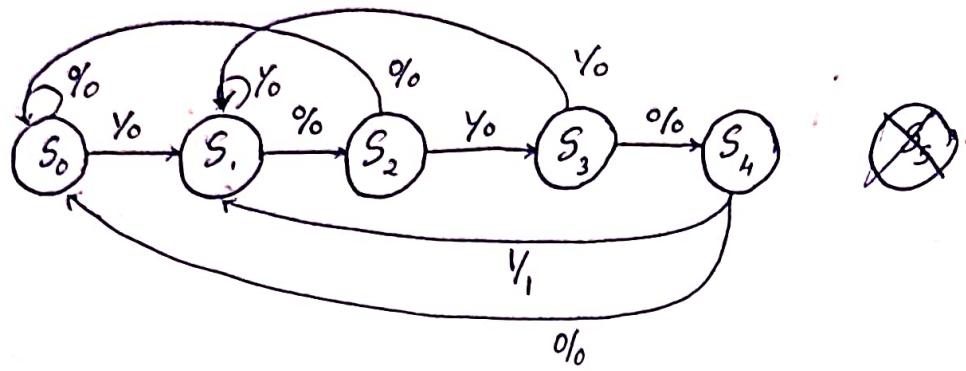
clk	Q ₂	Q ₁	Q ₀
↓	0	0	1
↓	1	0	0
↓	1	1	0
↓	1	1	1
↓	0	1	1
↓	0	0	1
↓	0	1	0
↓	0	0	1

Homework

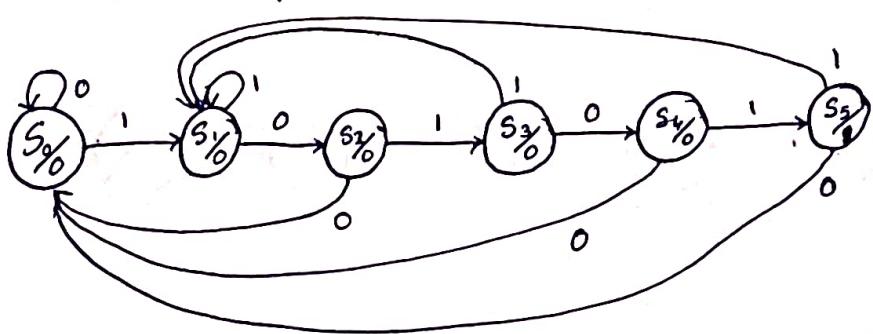
1. use shared doc. & design 8bit RCA, CLA, Carry Select adder
2. Draw ECL NOT gate & compare with TTL NOT gate

END

Design a 1-bit overlapping mealy machine 10101



Design a 1-bit overlap moore machine 10101



1. ADD gate using $A\bar{B}$
2. All possible logic gates using $A\bar{B}$
3. with use of NABND
4. $A + \bar{B}$
5. NOT with exor
6. NOT with exnor
7. half adder
8. full adder
9. 1-bit comparator
10. 4 bit adder
11. 4 bit subtractor
12. 4 bit adder, subtractor
13. 4 bit binary multiplier.
14. Binary to Gray
15. Gray to Binary
16. 2 i/p AND with 2:1 MUX
17. all logic gates using MUX 2:1
18. half subtractor using MUX 2:1
19. 1-bit comparator using MUX 2:1
20. 4:1 MUX with 2:1 MUX
21. 8:1 MUX with 2:1 MUX

22. SoP implementation in MUX

23. Transfer A
shift A using suitable MUX
Complement A

Sequential design.

1. logic circuit of rising edge sensitive D-flip flop.
2. logic circuit of falling edge sensitive D-flip flop.
3. Sequence generator 00, 01, 10, 11 (2-bit up-counter)
4. 3-bit synchronous up-counter.
5. 2-bit down counter.
6. 2-bit gray counter.
7. Clock divider $\div 2$
8. Clock divider $\div 3 \text{ (mod } 3\text{)}$
9. 8, 4, 2, ... Ring counter
10. 0, 8, 12, 14, 15, 7, 3, 1, ... Johnson counter (Twisted ring counter)
11. Clock divider mod 3 with 50% duty cycle.
12. Sequence generator 0, 2, 4, 6, 8, 10, 14, ...
13. D T flip flop from D flip flop.
14. 2-bit up-counter JK flip flop.
15. 4-bit up-counter JK flip flop.
16. 2-bit up/down counter JK flip flop
17. 3-bit up counter with pre-settable value

18. 2-bit down counter. JK flip flop
 19. 3-bit sync up counter D flip flop
 20. 101 non overlapping sequence detector moore machine
 21. 101 1-bit overlapping sequence detector moore machine
 22. 101 non-overlapping sequence detector Mealy machine.
 23. 101 1-bit overlapping sequence detector Mealy machine.
 24. 10 sequence detector Moore machine
 - 25 10 sequence detector Mealy machine
- 10101 sequence detector Moore machine & Mealy machine.

Static Timing analysis

1. Find f_{max} maximum frequency of operation
2. Find for setup violation, hold violation.
3. Analysis the impact of buffer in clock of each flip flop
+ in data path of each flip flop

Hazards

1. Check for hazard, identify type & suggest a fix in MUX 2:1

Block domain crossing

1. Use of level synchroniser & FIFO synchroniser.
other synchronizing methods
MUX, hand shaking, passing multiple signals
2. FIFO depth calculation