Date :
Page No.

Rules for Boolean Algebra

	AND laws	OR laws
$A+0=A$	$A.0=0$	$A+0=A$
$A+1=1$	$A.1=A$	$A+1=1$
$A.0=A$	$A.A=A$	$A+A=A$
$A+A=A$	$A.\bar{A}=0$	$A+\bar{A}=1$
$A+\bar{A}=1$		

$A.A=A$

$\bar{\bar{A}}=A$

$A+AB=A$

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

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

* **De Morgan's Theorem**

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

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

SOP → Sum of Products (Σm) Minterm

POS → Product of Sum (ΠM) Maxterm

SOP → NAND - NAND logic has less gates

POS → NOR - NOR logic has less gates.

BCD and XS3 Addition

BCD → Binary coded Decimal
↳ weighted code

8 4 2 1 → Standard BCD
2^3 2^2 2^1 2^0

eg: 8 - 1000
7 - 0111

BCD code - 27 11 → Start from unit place if there are two same digits
1 0 0 0 1
✓2 1 0 0 0
× 2 0 0 1 1 → in this, max. no. of bits are used that's why it is incorrect

use should use min. no. of bits

74	0111 0100
23	0010 0011
97	1001 0111

74	0111	0100
33	0011	0011
107	10000	0111
	0110	0110
	10110	1001

8	4	2	1
0	0	0	0
1	0	0	0
2	0	0	1
3	0	0	1
4	0	1	0
5	0	1	0
6	0	1	1
7	1	1	1

(if last value is greater than 9 then add 6 (0110) to that sum).

if carry generate or sum is greater than 9 add 6.

Date :
Page No.

66	0110	0110
66	0110	0110
132	1100	1100
(Add 6)	0110	0110
	10011	0010
	1 3	2

77	0111	0111
94	1001	0100
171	10000	1011
(Add 6)	0110	0110
	10111	0001
	1 7	1

XS3 Addition

$0101 \xrightarrow{XS3} 1000$

9's complement ↓ $0100 \longleftarrow 0111$

9's Comp. → Subtracting a no. from 9

eg 9 − 5 ; 99 − 54 , 999 − 547

XS3 code is adding 3 to BCD code

74 + 49 = 123

10's Comp. → Adding 1 to 9's comp.

9's comp. can be obtained by taking 1's comp. of its XS3 code and then reverting back to original code.

Add 3 in BCD Code

$7+3=10$
$4+3=7$
$4+3=7$
$9+3=12$ (10)

Addition

Date :
Page No.

eg. 74 + 49 = 123

	1010	0111	
	0111	1100	(12)
	0001	0010	0011
Add 3	0011	0011	0011
	0100	0101	0110
	4	5	6

123
333
456

if Carry is generated add 3
" " " not " Subtract 3

21	0101	0100	
+ 33	0110	0110	
54	1011	1010	
(+3 87)	− 0011	0011	
	1000	0111	
	8	7	

Subtraction :

4	0100	
− 7	+1000	→ Take 1's comp. of that no. (7), Add Then add that no. instead of (−)
−3	~~0001~~	
	1100	

(again, Take this 1's comp.) 0011 Ans (if Carry is generated then ans is +ve)

(if Carry is not gen. then ans is −ve)

8	1000	
− 4	+ 1011	
4	(1)0011	(if Carry is generated then add that Carry)
	+1	
	0100	

Date :
Page No.

Minterm → (SOP) 1
Maxterm → (POS) 0

* COMBINATION & SEQUENTIAL

Combinational circuit → are the circuit in which output is dependent on the present input.

Sequential circuit :- are the circuit in which o/p is dependent on the present and previous inputs. The previous inputs are stored in flip-flop. (one bit storage)

HALF ADDER (Combinational circuit: Adds two bits)

it has 2 inputs

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

Sum →

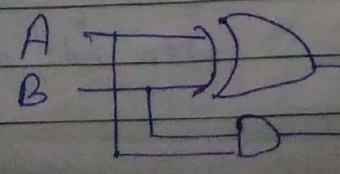
	B'	B
A'	0	1
A	1	0

$= A'B + AB'$ → (XOR)

Carry →

	B'	B
A'	0	0
A	0	1

$= AB$



(Circuit dig)

Date :
Page No.

FULL ADDER (has 3 inputs)

A	B	C	Sum	Carry
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Sum

A \ BC	B'C'	B'C	BC	BC'
A'	0_0	1_1	0_3	1_2
A	1_4	0_5	1_7	0_6

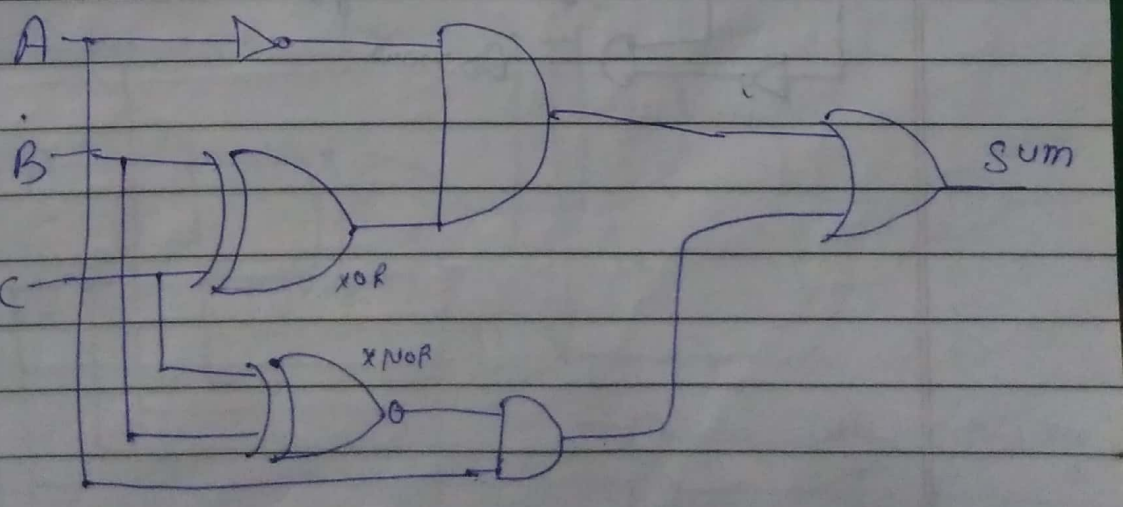
$= A'B'C + A'BC' + AB'C' + ABC$

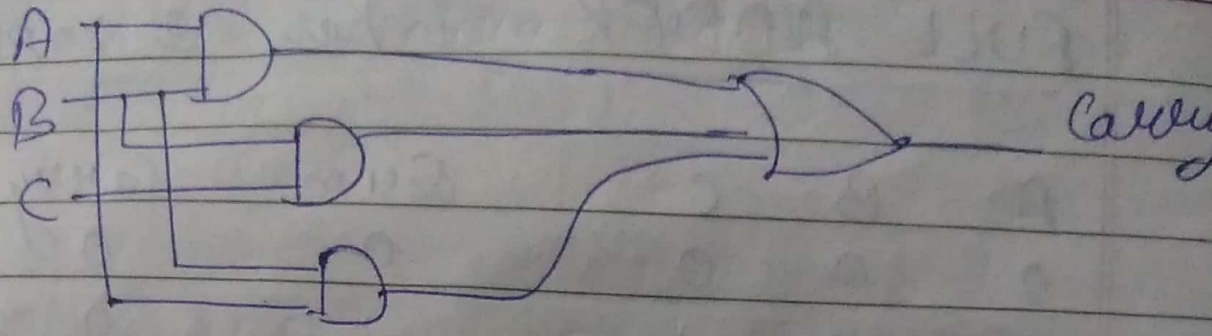
$= A'(B'C + BC') + A(B'C' + BC)$

Carry

	B'C'	B'C	BC	BC'
A'	0_0	0_1	1_3	0_2
A	0_4	1_5	1_7	1_6

$= AC + BC + AB$





* HALF SUBTRACTOR

A	B	Diff.	Borrow
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

Diff.

	B'	B
A'	0	1
A	1	0

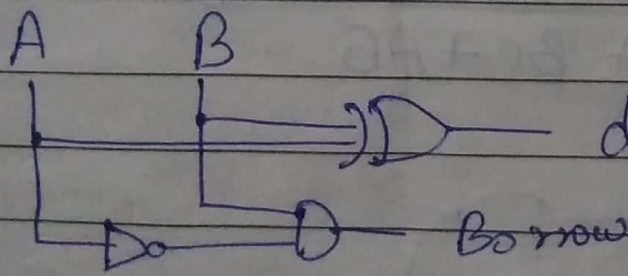
$= A'B + AB'$

(XOR)

Carry

	B'	B
A'	0	1
A	0	0

$= A'B$



Page No.

FULL SUBTRACTOR

A	B	C	Diff.	Borrow
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	1
1	0	0	1	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

Diff.

	B'C'	B'C	BC	BC'
A'	0_0	1_1	0_3	1_2
A	1_4	0_5	1_7	0_6

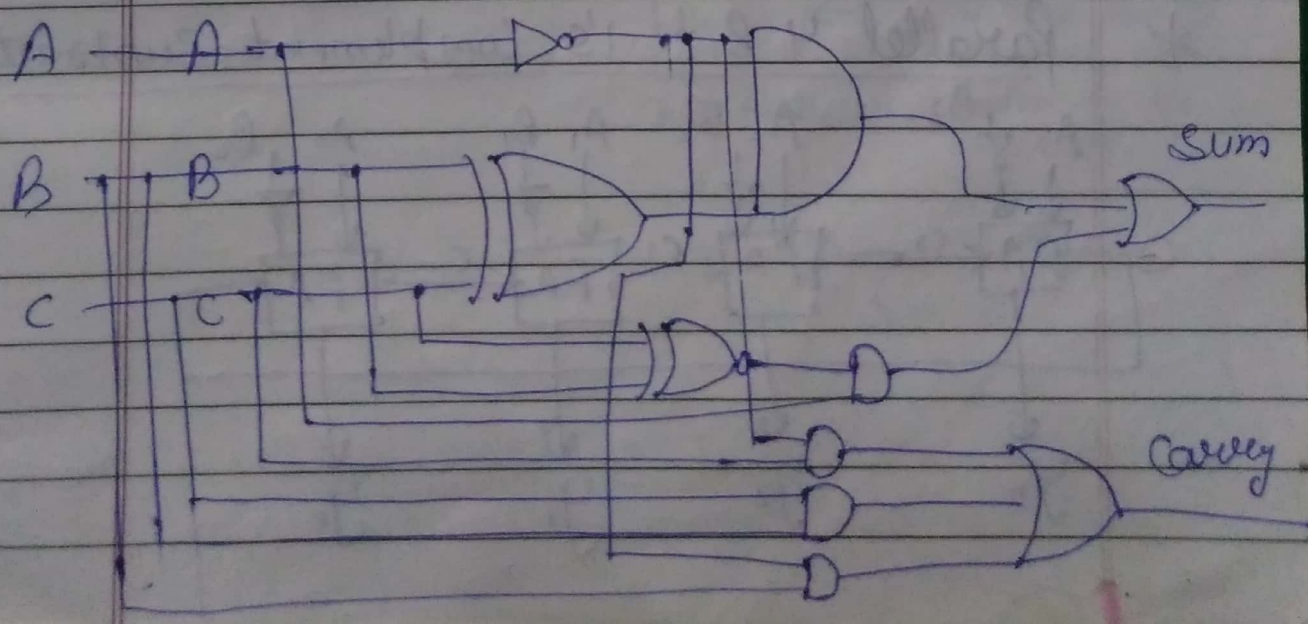
$= A'B'C + A'BC' + AB'C' + ABC$

$= A'(B'C + BC') + A(B'C' + BC)$

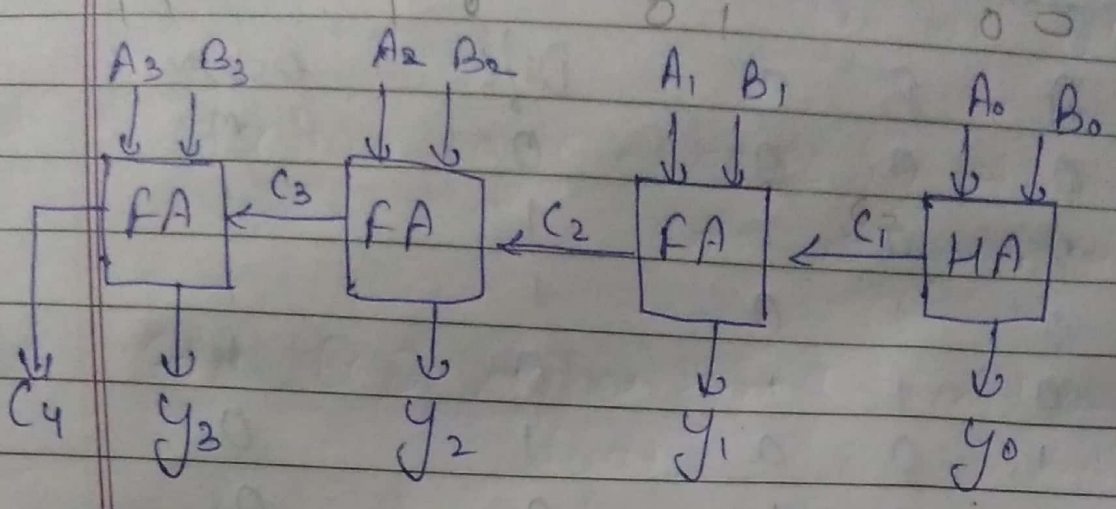
Carry.

	B'C'	B'C	BC	BC'
A'	0_0	1_1	1_3	1_2
A	0_4	0_5	1_7	0_6

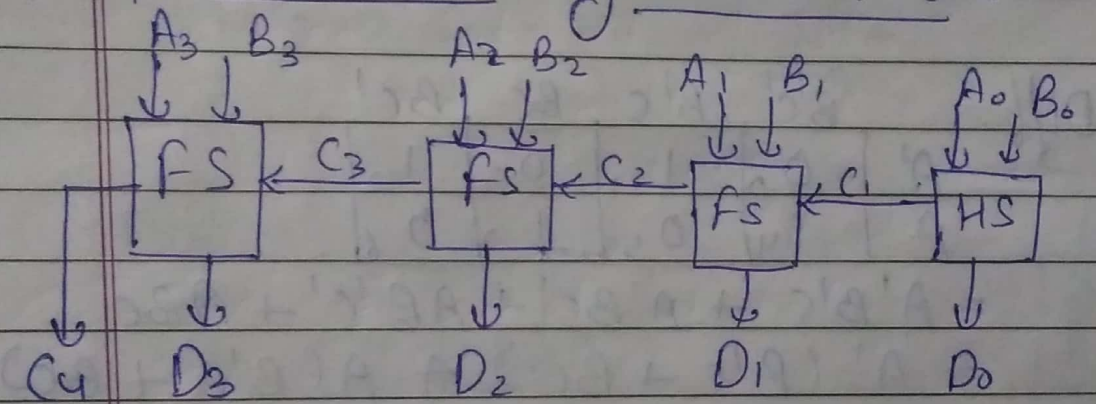
$= A'C + BC + A'B$



* Parallel Binary Adder

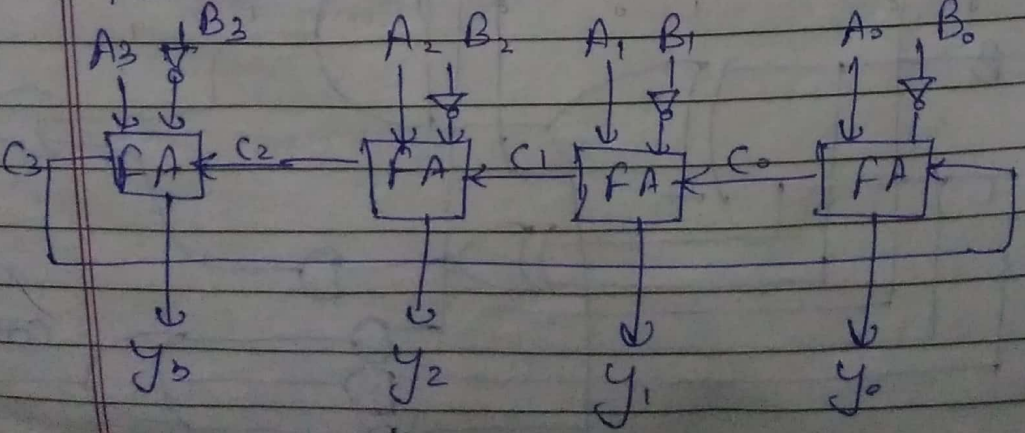


* Parallel Binary Subtractor



$$\begin{array}{r} A_3\ A_2 \quad A_1\ A_0 \\ +\ B_3\ B_2 \quad B_1\ B_0 \\ \hline C_4\ Y_3\ Y_2\ Y_1\ Y_0 \end{array}$$

* Parallel 4 Bit 1's Complement Subtractor



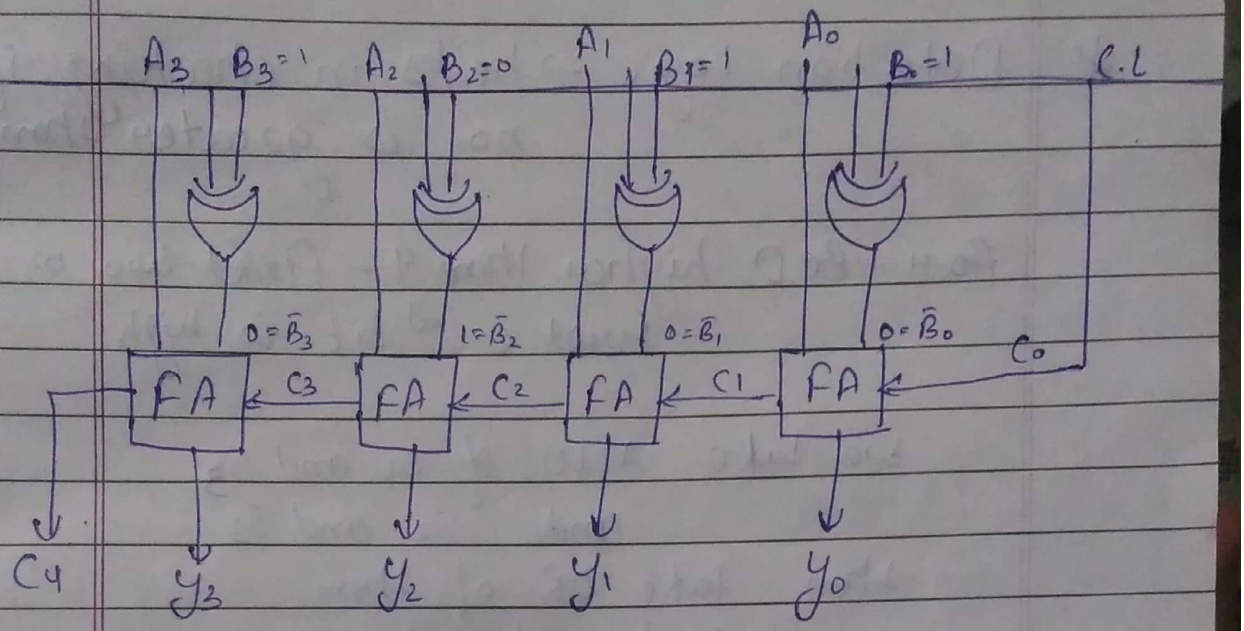
CL → Control line

Date :
Page No.

CL = 0

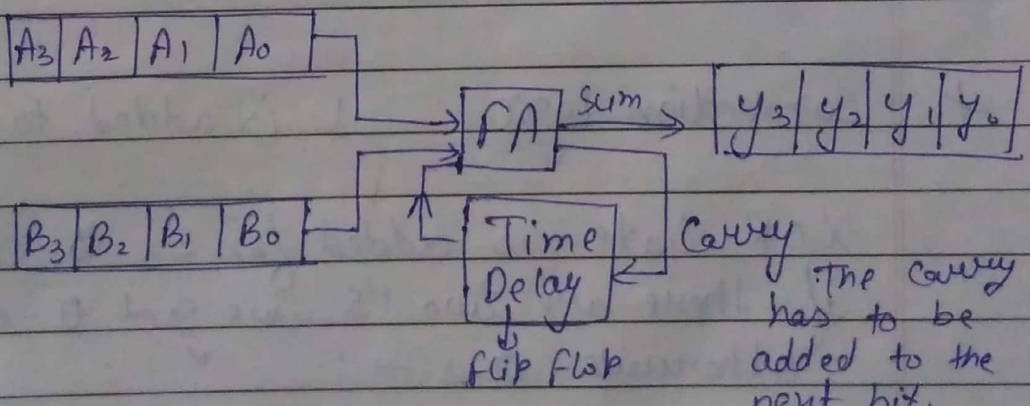
* 9's complement adder / Subtractor

CL = 1



Adder → C.L = 0

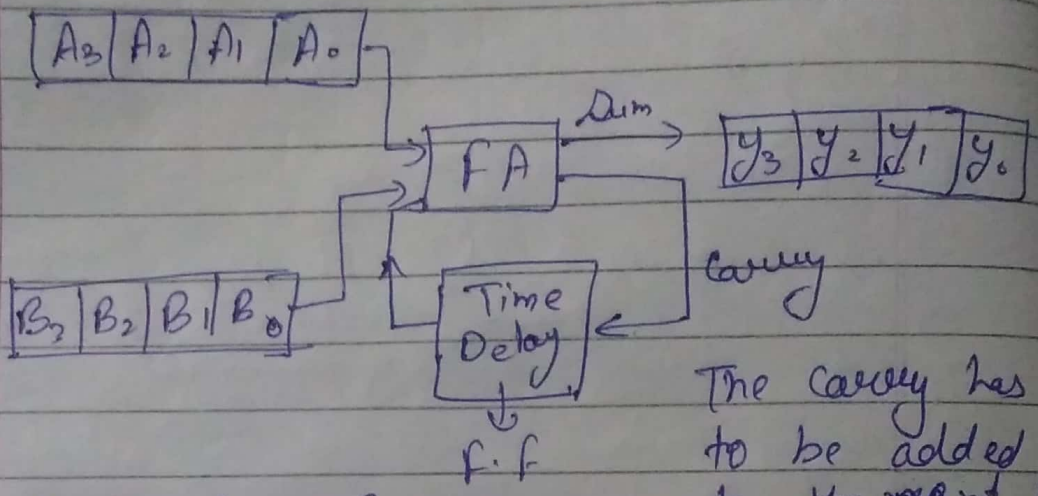
* SERIAL ADDER



Date :

DE

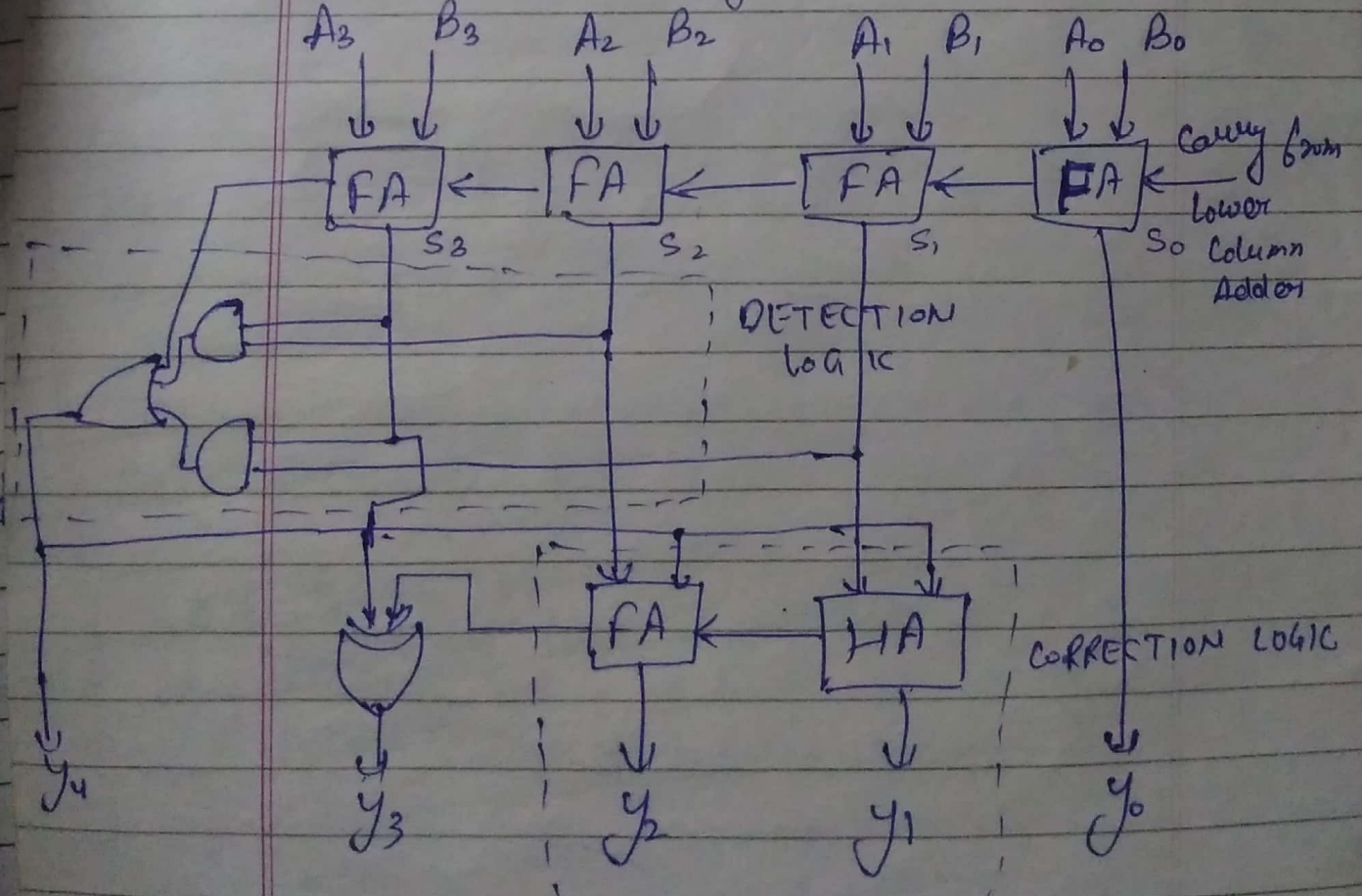
SERIAL ADDER



The carry has to be added to the next bit

BCD ADDER

If no. is gr. than 9, we add 6



y4 . . carry

Gray → Reflected code where only 1 bit changes at a time

Page No.
Date :

BINARY TO GRAY

	B_2	B_1	B_0	G_2	G_1	G_0
0	0	0	0	0	0	0
1	0	0	1	0	0	1
2	0	1	0	0	1	1
3	0	1	1	0	1	0
4	1	0	0	1	1	0
5	1	0	1	1	1	1
6	1	1	0	1	0	1
7	1	1	1	1	0	0

Binary code to Gray code

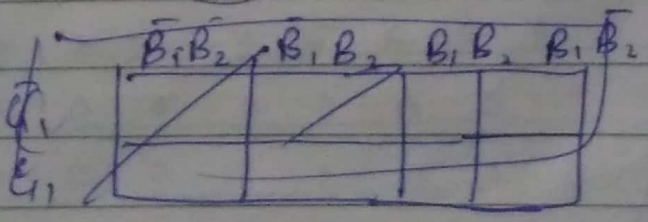
B → 0 1 0

G → 0 1 1

Gray to Binary

G 0 1 1

B 0 1 0



$G_2 = B_2$

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

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

G_1

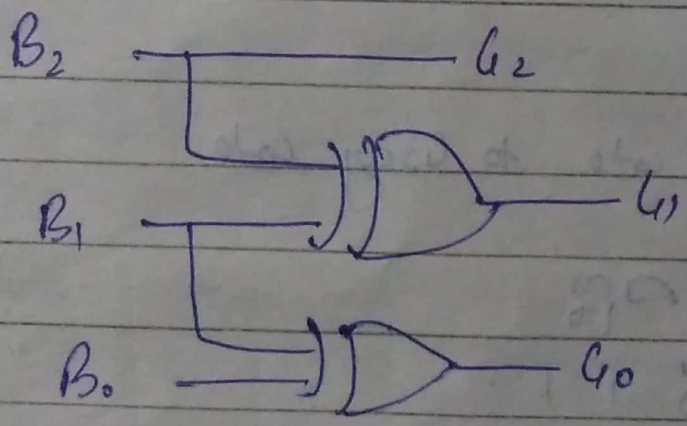
	$\bar{B}_1\bar{B}_0$	$\bar{B}_1 B_0$	$B_1 B_0$	$B_1\bar{B}_0$
$\bar{B}_2$	0	0	1	1
B_2	1	1	0	0

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

G_2

	$\bar{B}_1\bar{B}_0$	$\bar{B}_1 B_0$	$B_1 B_0$	$B_1\bar{B}_0$
$\bar{B}_2$	0 (0)	0 (1)	0 (3)	0 (2)
B_2	1 (4)	1 (5)	1 (7)	1 (6)

$$G_1 = B_1\bar{B}_2 + \bar{B}_1 B_2$$
$$G_2 = B_2$$
$$G_0 = \bar{B}_1 B_0 + B_1\bar{B}_0$$



Date :

Gray To Binary

	G_2	G_1	G_0	B_2	B_1	B_0
0	0	0	0	0	0	0
1	0	0	1	0	0	1
2	0	1	1	0	1	0
3	0	1	0	0	1	1
4	1	1	0	1	0	0
5	1	1	1	1	0	1
6	1	0	1	1	1	0
7	1	0	0	1	1	1

B_2

	$\bar{G}_1\bar{G}_0$	$\bar{G}_1 G_0$	$G_1 G_0$	$G_1\bar{G}_0$
$\bar{G}_2$	0	0	0	0
G_2	1	1	1	1

$B_2 = G_2$

NOT CARE ~~Condi~~ Condition

Q Design a min. Mood Logic Circuit to detect the decimal 3, 4, 5 or 6 in 5211 Code.

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
B_3	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
B_2	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
B_1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
B_0	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
	0	0	X	X	0	1	X	1	1	1	X	X	0	0	X	X0

Date :

	5	2	1	1	
3	0	1	0	1	- 5
4	0	1	1	1	- 7
5	1	0	0	0	- 8
6	1	0	0	1	- 9

Put 1 on this Place

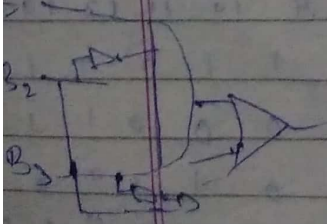
0	0 0 0 0	- 0
1	0 0 0 1	- 1
2	0 1 0 0	- 4
7	1 1 0 0	- 12
8	1 1 0 1	- 13
9	1 1 1 1	- 15

Put 0 on this Place

And Put x on the remaining Place.

	$B_1'B_0'$	$B_1'B_0$	B_1B_0	B_1B_0'
$B_3'B_2'$	0_0	0_1	X_3	X_2
$B_3'B_2$	0_4	1_5	1_7	X_6
B_3B_2	0_{12}	0_{13}	0_{15}	X_{14}
B_3B_2'	1_8	1_9	X_{11}	X_{10}

$$B_3'B_2 + B_1B_2'B_3$$



	8	4	2	1	
5	0	1	1	1	- 7
6	1	0	1		- 6
7	1	0	0		- 8
8	1	1	0	0	- 12
9	1	1	0	1	- 13
10	1	1	1	1	- 15
11	1	1	1	0	- 14
12	1	0	1	0	- 10

$\cup y_4$

Page No.
Date :

Q Design a logic circuit where if the decimal no. greater than 5 no's occurs it gives 1.

	0	1	2	3	4	5	6	7	8	9	10	11	12
B_3	0	0	0	0	0	0	0	0	1	1	1	1	1
B_2	0	0	0	0	1	1	1	1	0	0	0	0	1
B_1	0	0	1	1	0	0	1	1	0	0	1	1	0
B_0	0	1	0	1	0	1	0	1	0	1	0	1	1
	0	0	0	0	0	0	1	1	1	1	x	x	x

	$B_1'B_0'$	B_1B_0'	B_1B_0	$B_1'B_0$
$B_3'B_1'$	0_0	0_1	0_3	0_2
B_3B_2'	0_4	0_5	1_7	1_6
B_3B_2	X_{12}	X_{13}	X_{15}	X_{14}
$B_3'B_2$	1_8	1_9	X_{11}	X_{10}

Design a 4 input SOP circuit that give output 1 any time the gray codes before 5 to 12 appear in input & 0 for all of the codes.

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
B_3	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1
B_2	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1
B_1	0	0	1	1	0	0	1	1	0	1	1	1	0	0	1
B_0	0	1	0	1	0	1	0	1	0	0	0	1	1	1	0

diff b/w decoder & demux and encoder & decoder

In MUX & DEMUX output is depends upon the select lines but

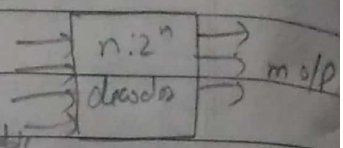
In encoder & decoders output is depends on Inputs

Date :

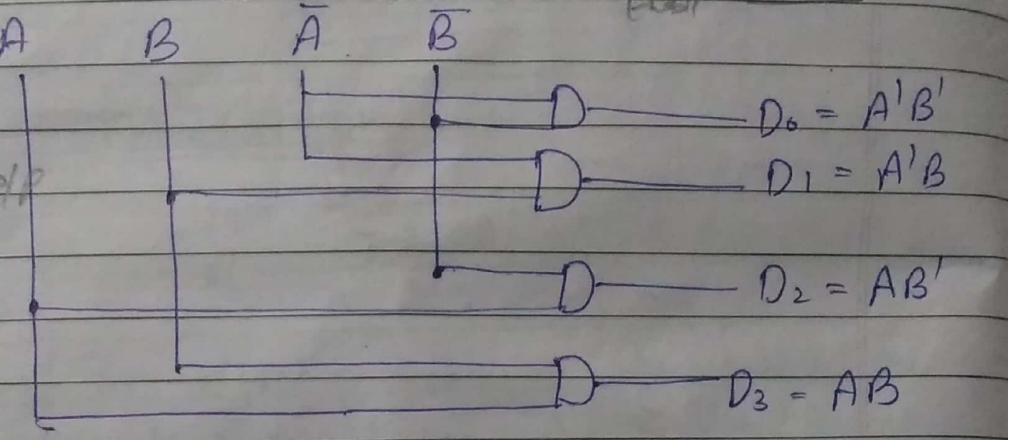
* **Decoders**: is a combinational circuit that receive binary information from n inputs lines to 2^n output lines But only one o/p is work at a time

n i/p → 2^n o/p

eg 3 → 8 o/p



i/p & multiple o/p Decoder converts coded i/p into coded o/p

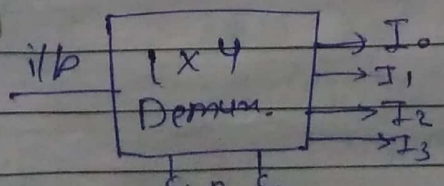


Decoder is provided with an enable i/p to activate decoded o/p based on

i/p A	B	D_0	D_1	D_2	D_3
0	0	1	0	0	0
0	1	0	1	0	0
1	0	0	0	1	0
1	1	0	0	0	1

* **DEMULTIPLEXER** (Similar to decoder) convert serial data to Parallel data

(DEMUX)



$n - 2^n$, n → Select lines

1 inputs → n outputs

select lines control the routing of the discrete data input to the output.

Date :
Page No. DIVYA

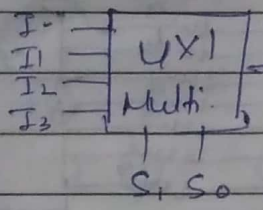
I/P	S_1	S_0	O/P
A	0	0	$I_0 = A \cdot \bar{S}\bar{S} I_0$
0	0	1	$I_1 = \bar{S} S I_1$
1	1	0	$I_2 = S \bar{S} I_2$
$\bar{A}$	1	1	$I_3 = S S I_3$

The output appears based on the status of select lines.

MULTIPLEXER (MUX)

(Convert Parallel info. data to Serial data)

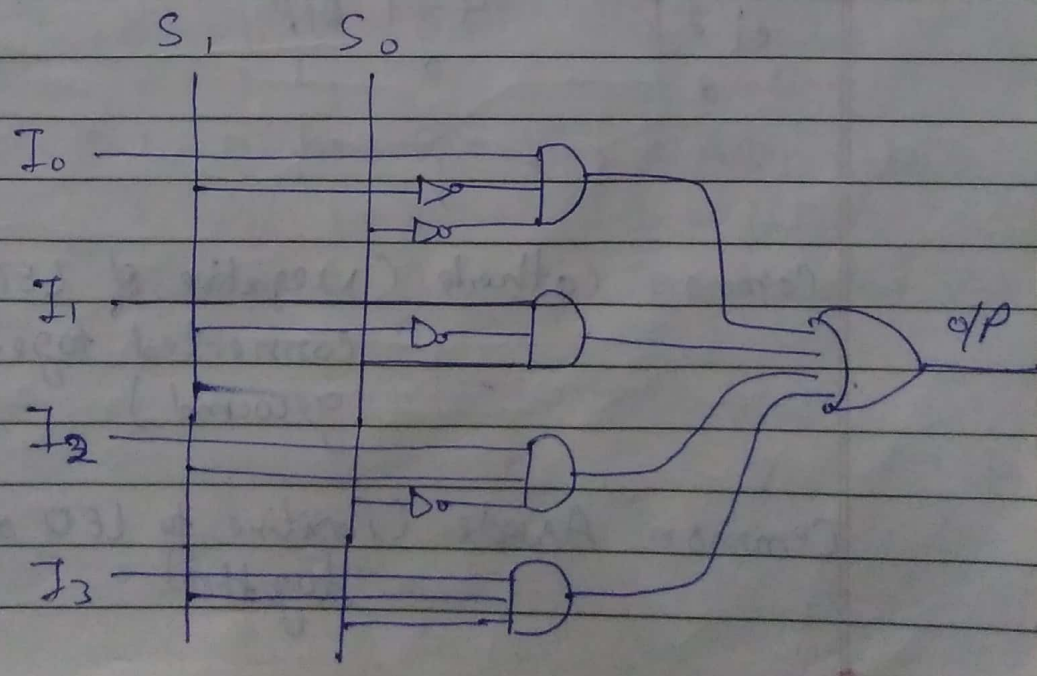
2^n inputs → 1 output



n – Select lines

2^n i/p → 1 o/p , n → Select lines

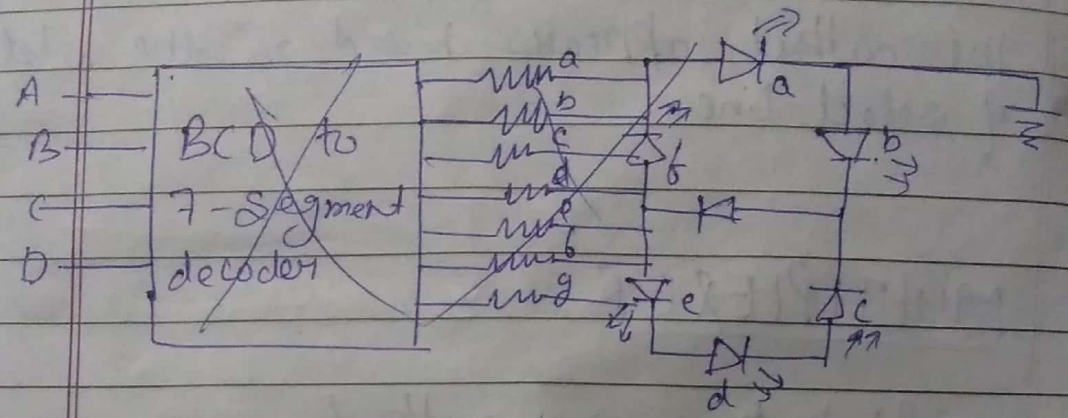
2^3

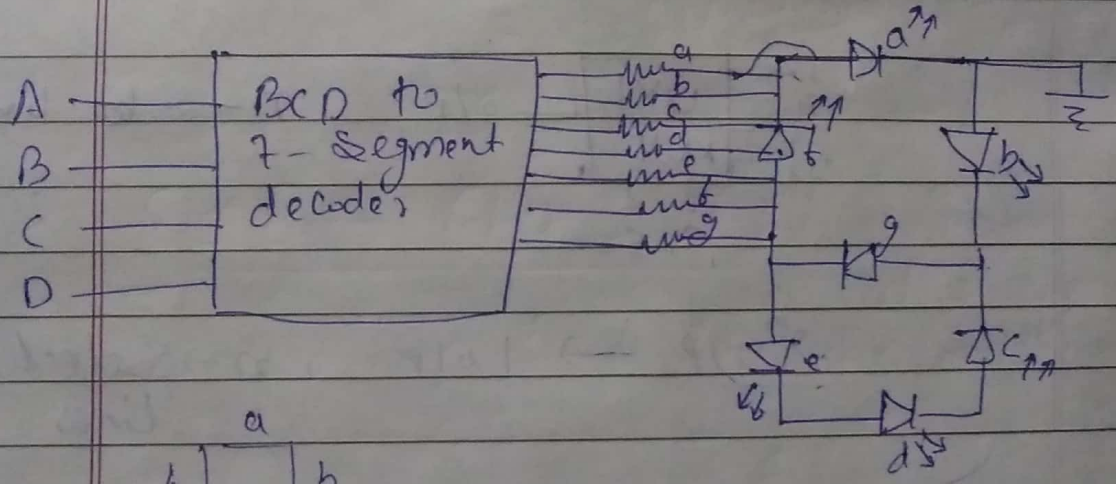


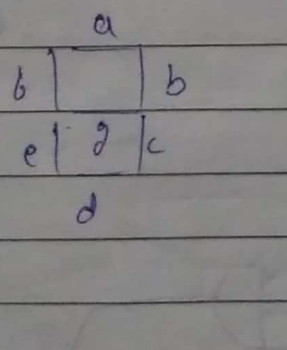
Date :
Page No.

Encoder Many i/p e many o/p but only $2^n \rightarrow n$ i/p wale at a

BCD to 7-Segment Display







4-7 O/P

Common cathode (Negative of LED are connected together to ground)

Common Anode (Positive to LED are tied together)

Date :
Page No.

Implementation using Mux.

$$f = \Sigma(3,4,6,7,9,11,13)$$

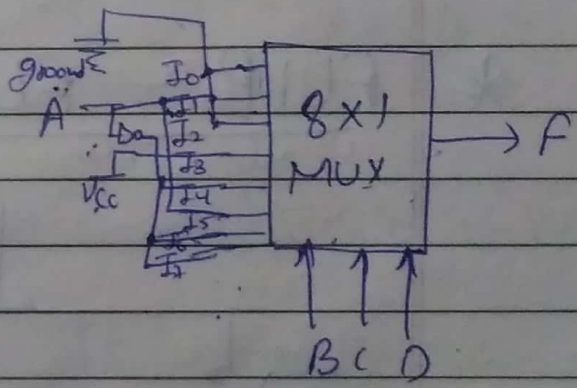
	I_0	I_1	I_2	I_3	I_4	I_5	I_6	I_7
A'	0	1	2	(3)	(4)	5	(6)	(7)
A	8	(9)	10	(11)	12	(13)	14	15
	0	A	0	1	A'	A	A'	A'

	A	B	C	D	f
0	0	0	0	0	0
1	0	0	0	1	0
2	0	0	1	0	0
3	0	0	1	1	1
4	0	1	0	0	1
5	0	1	0	1	0
6	0	1	1	0	1
7	0	1	1	1	1
8	1	0	0	0	0
9	1	0	0	1	1
10	1	0	1	0	0
11	1	0	1	1	1
12	1	1	0	0	0
13	1	1	0	1	1
14	1	1	1	0	0
15	1	1	1	1	0

4x1

I_0 — I_3

8x1



For 4 ABCD ~~inputs~~

1 → inputs

3 → Select lines

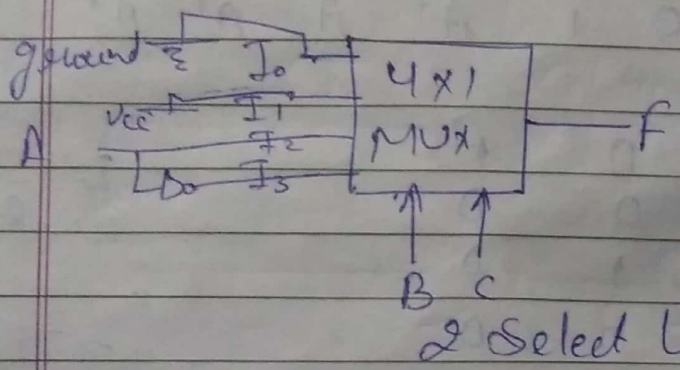
2^n → I/P

n - Select lines

Date :
Page No.

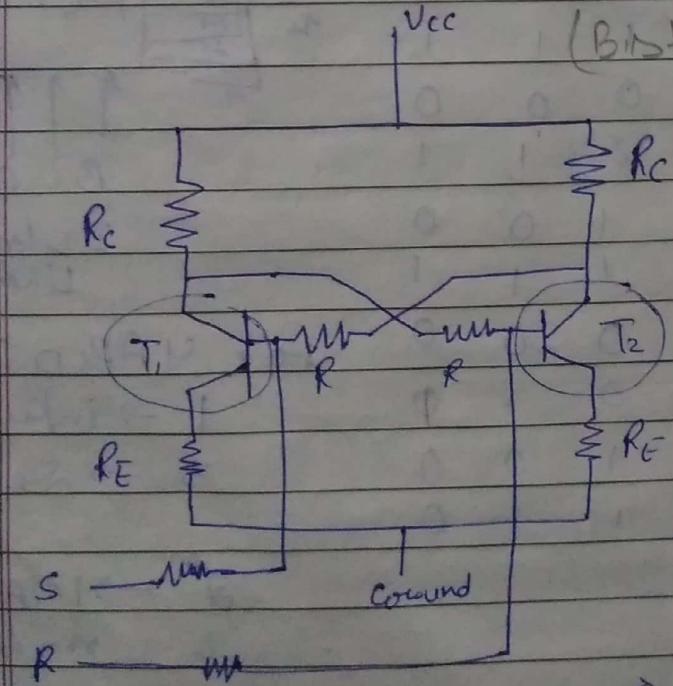
Q implement the boolean function. $\Sigma(1,3,5,6)$ using MUX.

	I_0	I_1	I_2	I_3
A'	0	(1)	2	(3)
A	4	(5)	(6)	7
	0	1	A	A'



Q

* FLIP – FLOPS (F.F) : Store one info. at a time (Bistable multivibrator)



(internal architecture of flip-flop)

Q_n → Existing state

S	R	Q_{n+1}
0	0	Q_n
0	1	0
1	0	1
1	1	Race *

Truth Table of SR

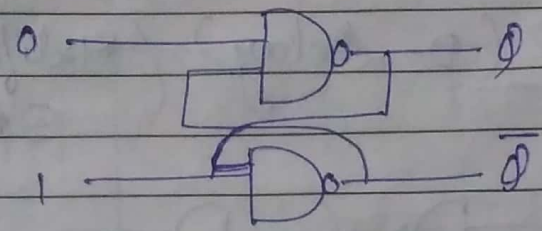
Race → depends on which transistor state first, it is indeterminant

Date :
Page No.

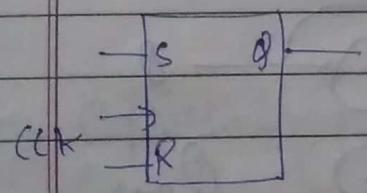
Types of F.F

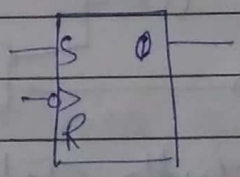
SR → SET RESET
D → Delay
JK → Jack - Kilby
T → Toggle

LATCH



CLK → Clock Signal



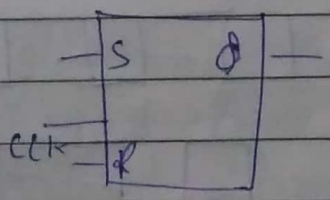


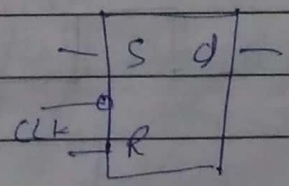
CP → Clock Pulse

→ +ve edge trigger
→ -ve edge trigger

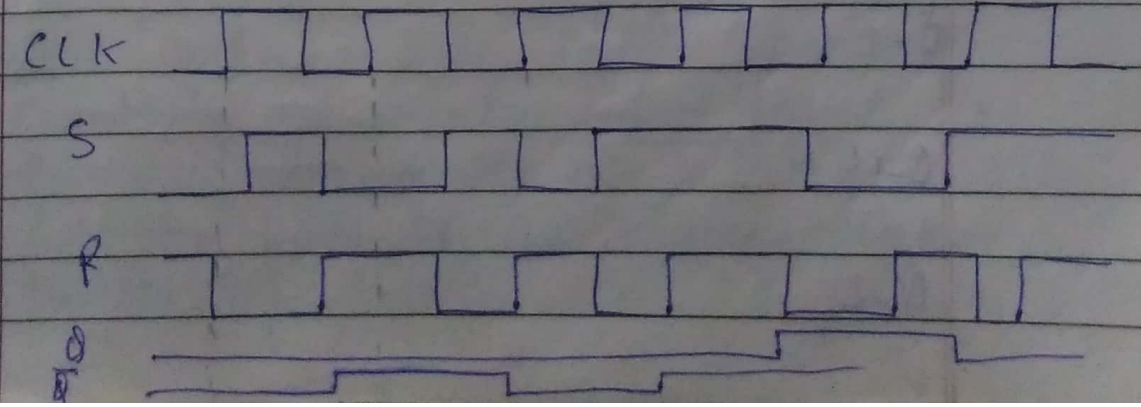
-ve edge

CLK or CP

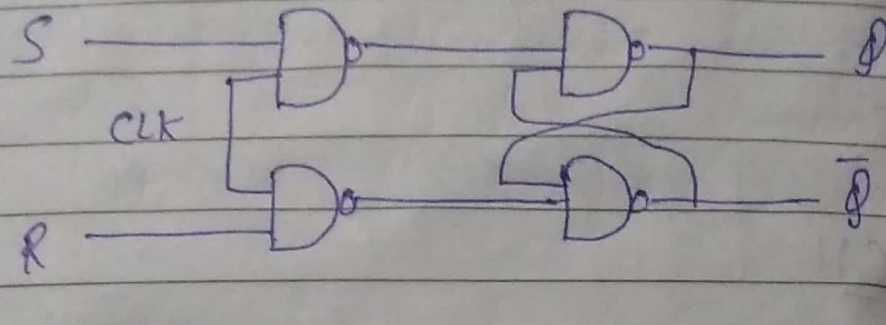




(—) +ve level trigger
(—○)-ve level trigger

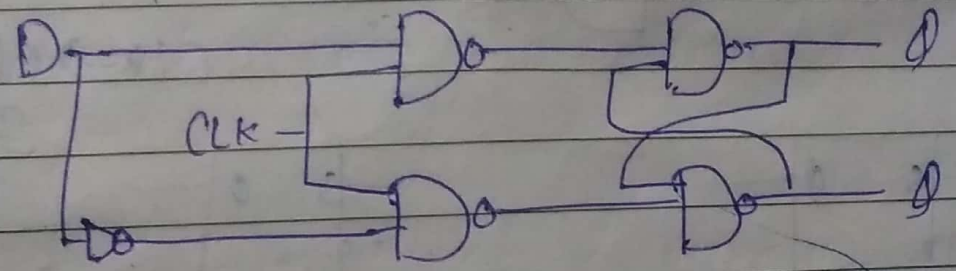


Circuit dig. for S.R F.F.



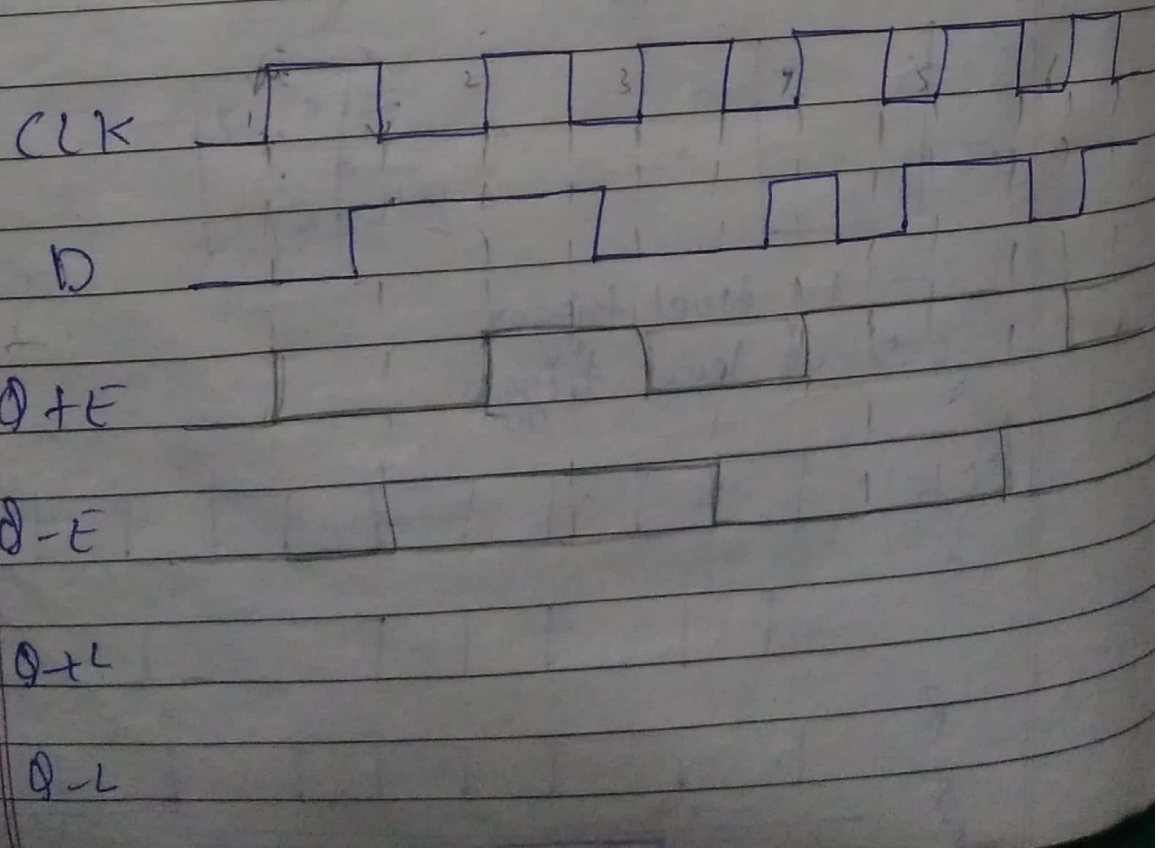
S	R	Q_{n+1}
0	0	Q_n
0	1	0
1	0	1
1	1	Race

D - FF (D - Delay) (Modification of SR F.F.)



Truth Table,

D	Q_{n+1}
0	0
1	1



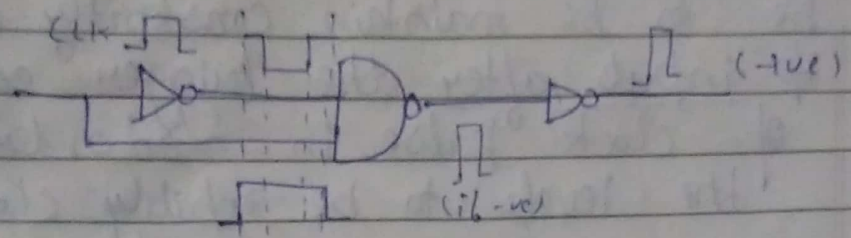
Q 0 1

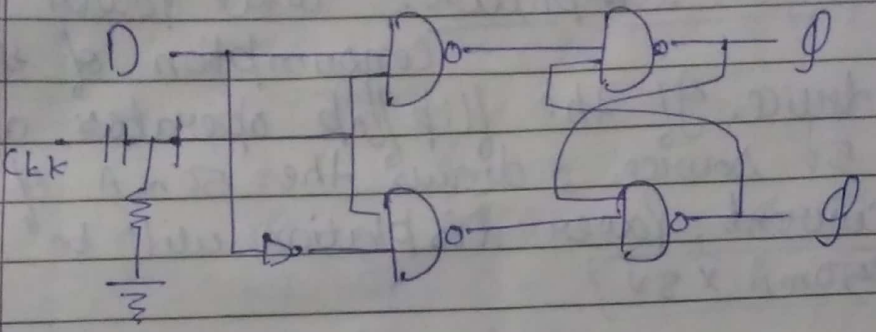
$\bar{Q}$ 0 1

Date :

Page No.

How to generate edge





* Operating characteristics of Flip Flop

① **Propagation delay time:** A Propagation delay is a interval of time require after the input signal has been apply for the resultant output change.

② **Whole Time (Setup time):** It is the min. interval require for the control inputs to be maintain constantly ed to the input prior the triggers edge of the clock pulse in order for the levels to be rebibly clocked in the flip flop

Date :
Page No.

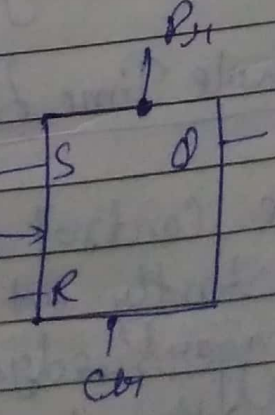
Whole Time: The min. time require for the control levels should be, to be maintain constantly to the ip inputs after the trigger edge of clock pulse in order for the levels to be rebibly clocked.

(3) Power Dissipation: Total Power consumption of the device. If the flip flop operates on a 5V source & draws the 50mA of current, Power dispiation will be 50mA x 5V

(4) Maximum clock frequency: is the highest frequency at which FF can be reliably rebibly triggers

* ASYNCHRONOUS INPUTS

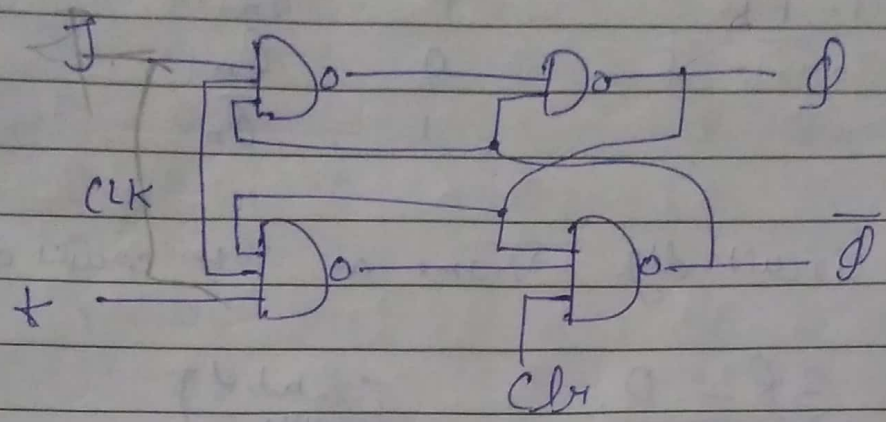
PRESET
CLEAR



Pr	Clr	Q
0	0	invalid
0	1	1
1	0	0
1	1	Q

Date :
Page No.

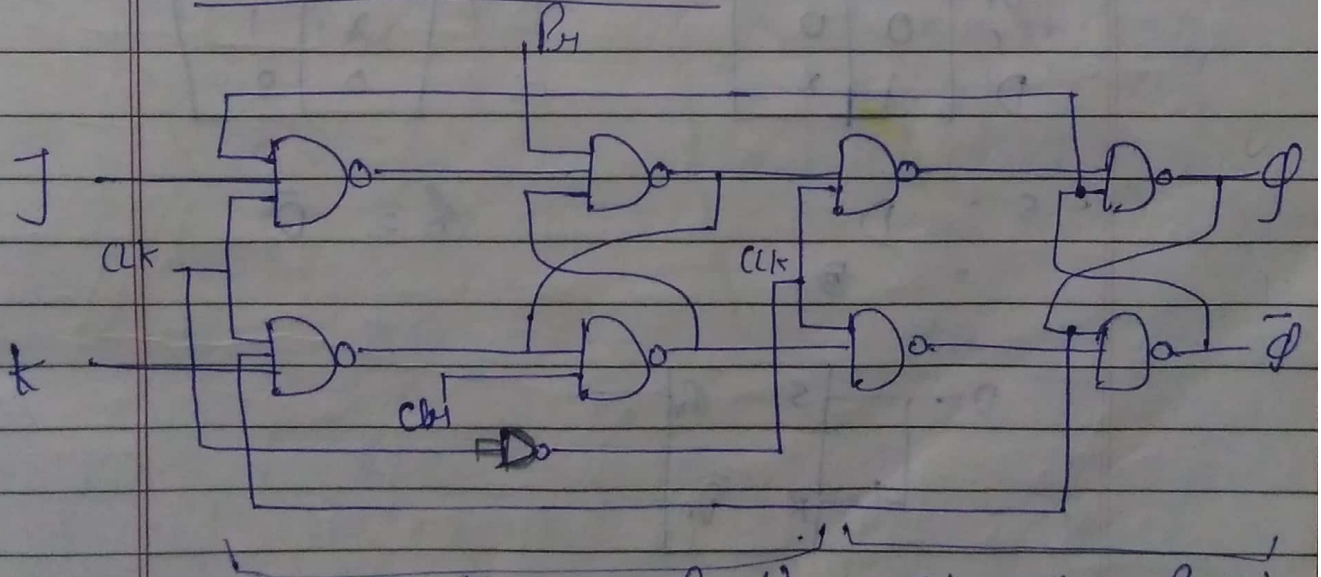
* JK-ff (Jack-Kilby)



J	K	Q_{n+1}
0	0	Q_n
0	1	0
1	0	1
1	1	$\bar{Q}_n$

J K Q $\bar{Q}$

* Master-Slave ff



Master (+ve Part) Slave (-ve Part)
(+ve level output) (-ve level output)

Master ff output → -ve level (Slave)
change in input does not effect the outputs.
(in -ve level trigger)

DIVYA

FLIP-FLOP CONVERSATION

→ T-FF (Toggle F-F)

T	Q_{n+1}
0	Q_n
1	$\bar{Q}_n$

Circuit dig. → same as JK circuit dig.

→ SR-D

(inputs of final FF) [inputs of starting FF]

D	Q_n	Q_{n+1}	S	R
0	0	0	0	X
0	1	0	0	1
1	0	1	1	0
1	1	1	X	0

(X = 0 or 1)

S

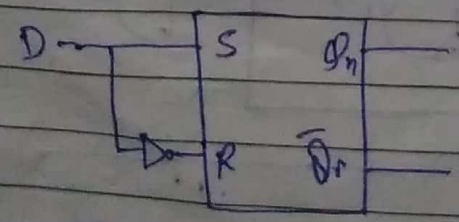
Q_n \ Q_{n+1}	0	1
0	0	0
1	1	X

R

Q_n \ Q_{n+1}	0	1
0	X	1
1	0	0

S = D

R = $\bar{D}$



Circuit dig.

T - SR

S	R	Q_n	Q_{n+1}	T

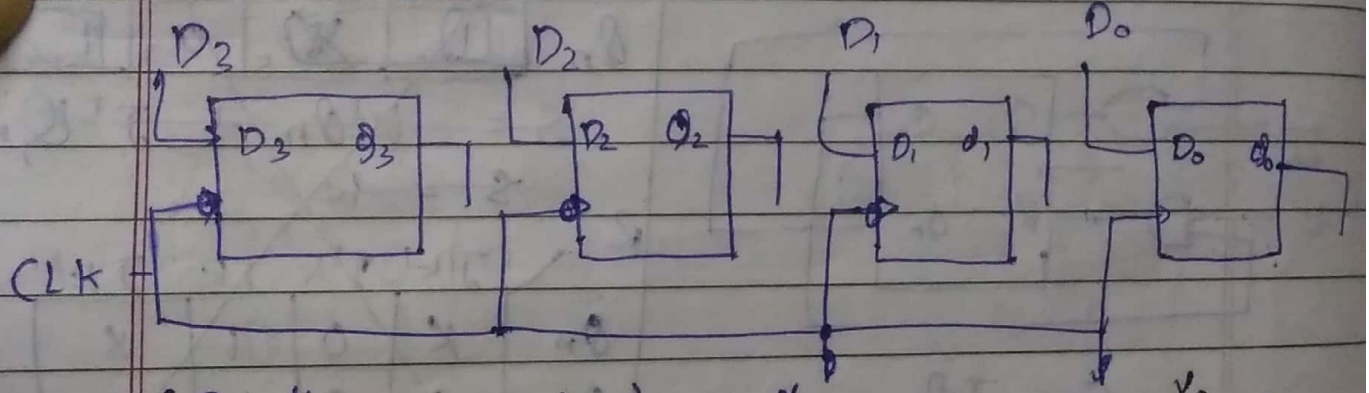
REGISTERS

(Combination of FF) (Grp. of memory elements that work together to form.)

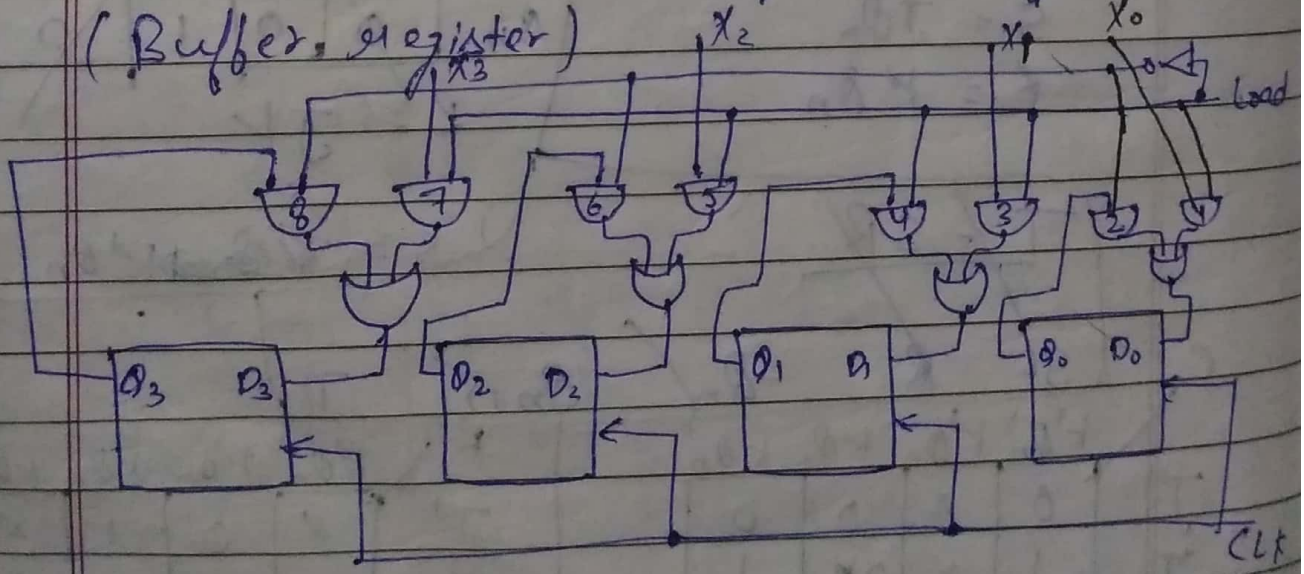
Types of Registers

→ SHIFT (Shift the bits, data from 1 FF to another)

→ BUFFER (Store bits, data)



(Buffer register)



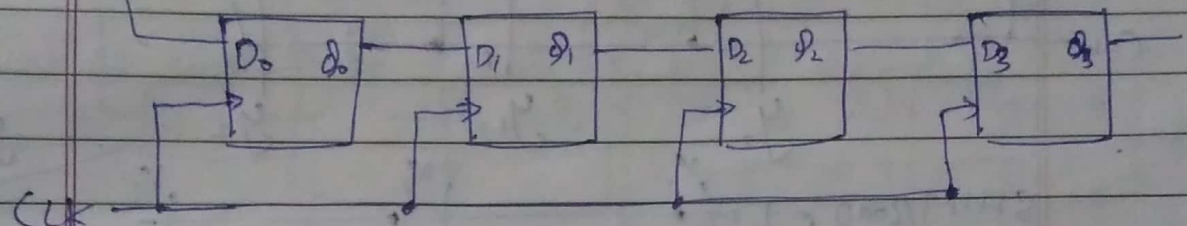
Controlled Buffer Register.

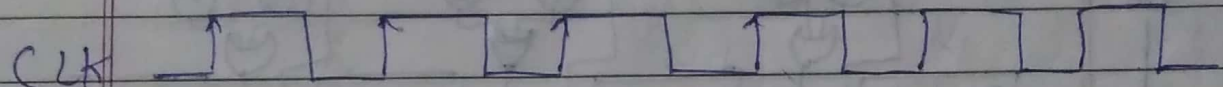
Date :
Page No.

SHIFT
- SISO (Serial in Serial out)
- SIPO (Serial in Parallel out)
- PISO (Parallel in Serial out)
- PIPO (Parallel in Parallel out)

x_3
x_2
x_1
x_0 let $x_0 = 1$

SISO





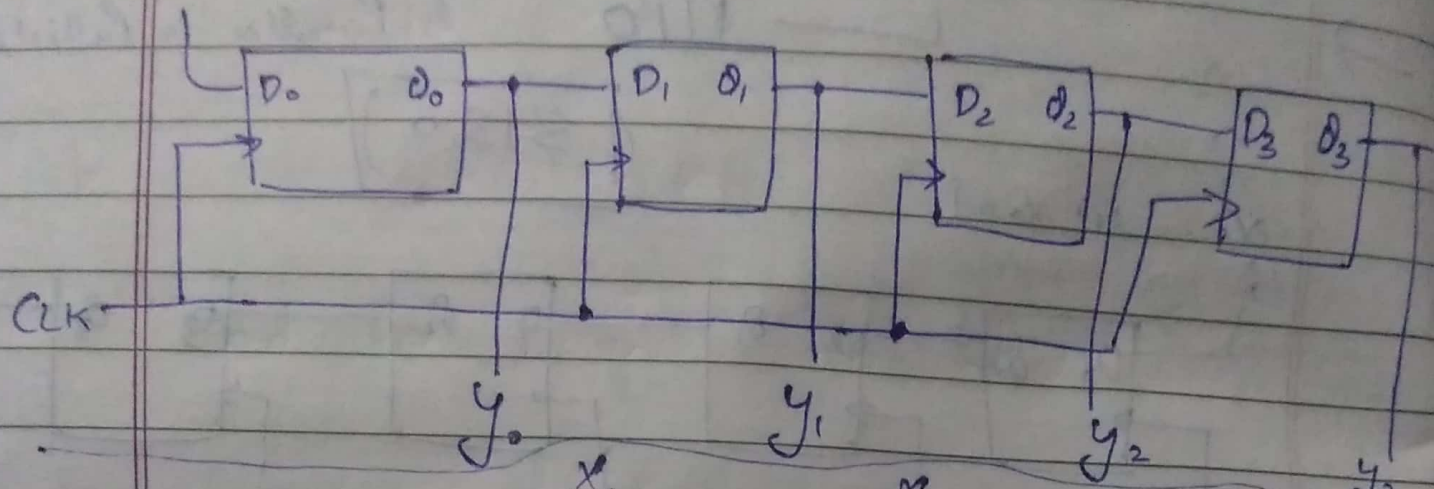
Q_0	1	1		
Q_1	0	1		
Q_2	0	0	1	
Q_3	0	0	0	1

1011

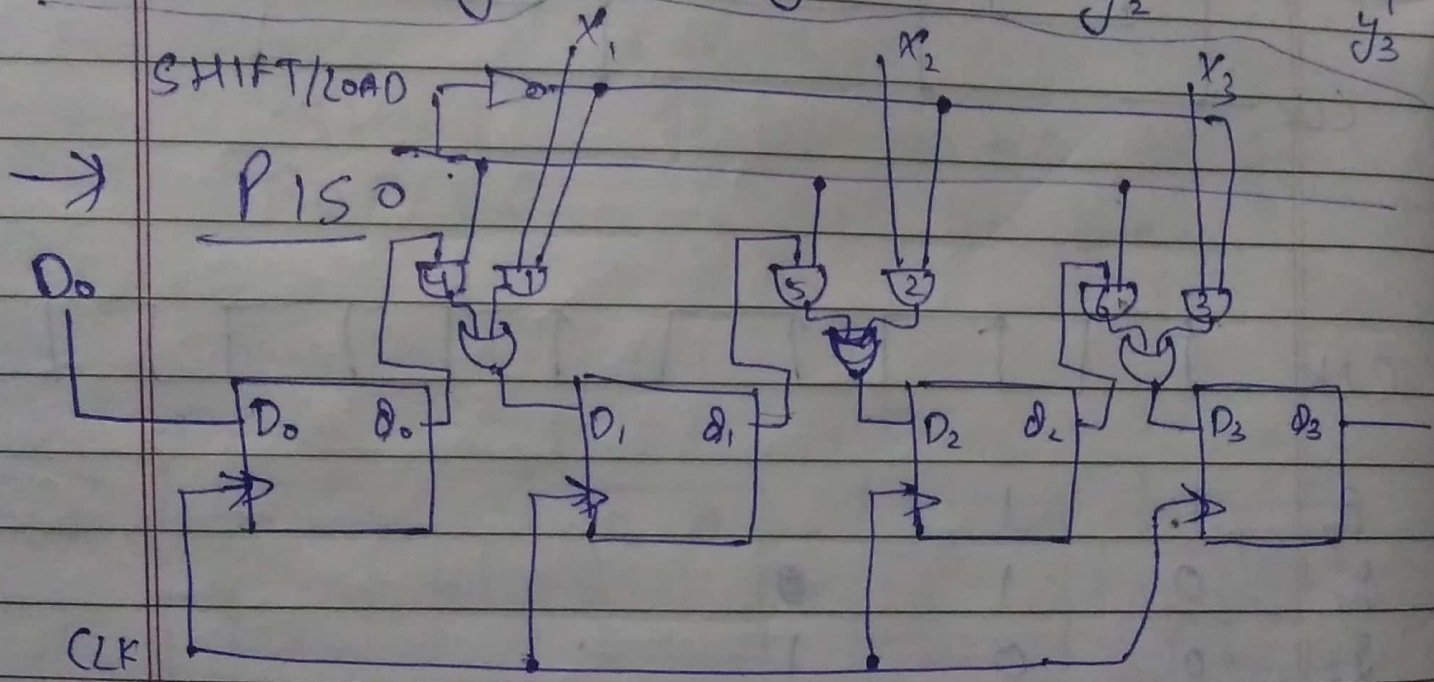
CLK	Q_0	Q_1	Q_2	Q_3
1	1	0	0	0
2	1	1	0	0
3	0	1	1	0
4	1	0	1	1

Date :
Page No.

→ SIPO



→ PISO

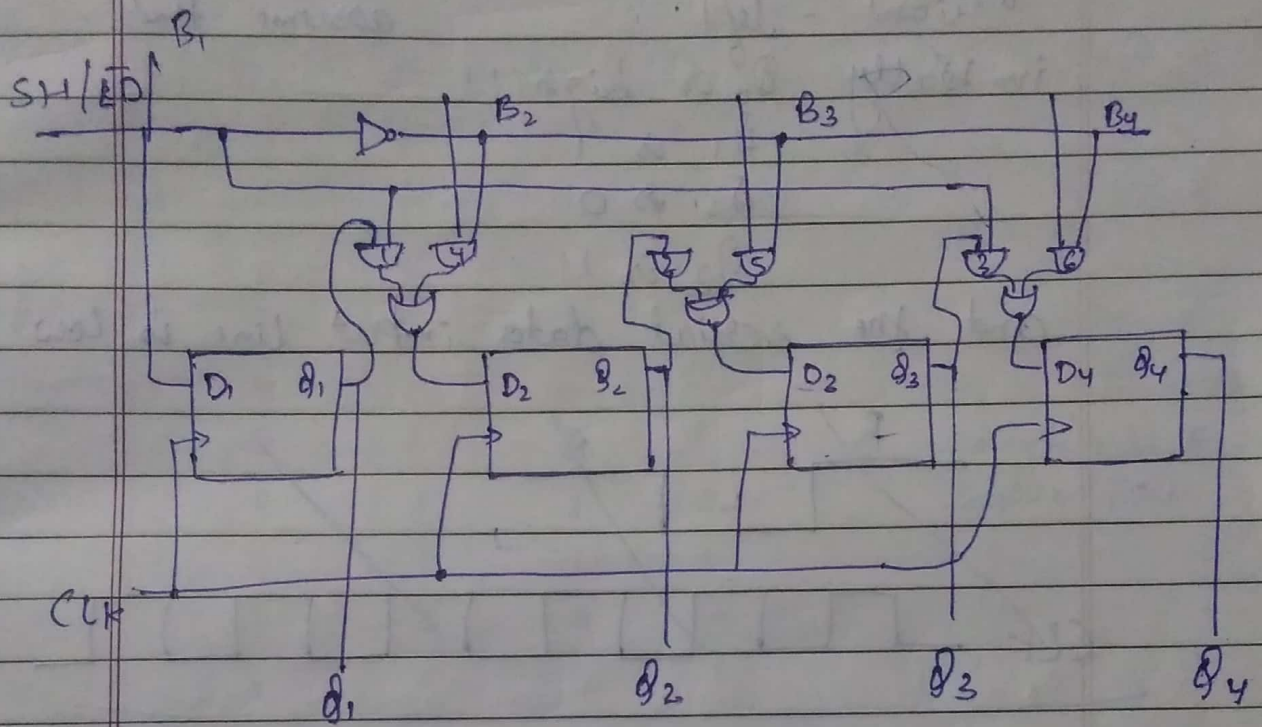


Signal 1 → Shift the data
0 → Load " "

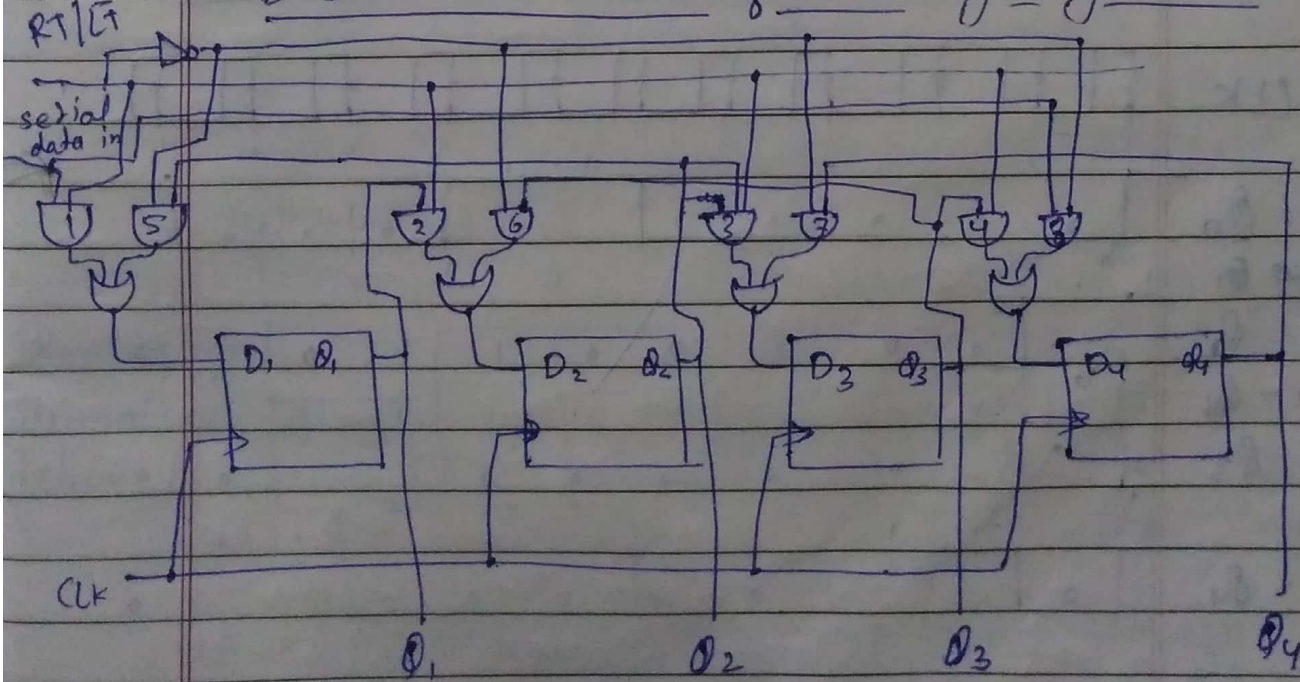
Mid term Syllabus → till (15).

Date :
Page No.

→ Parallel - In - Parallel out Shift Register



* Bidirectional shift Resig Register



up counter
down counter

output of 1 FF effects the CLK of other FF

Date :

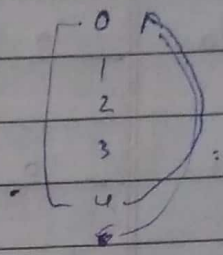
Page No.

COUNTERS

- Asynchronous (Ripple)
- synchronous (all clock are tight together)

Modulus

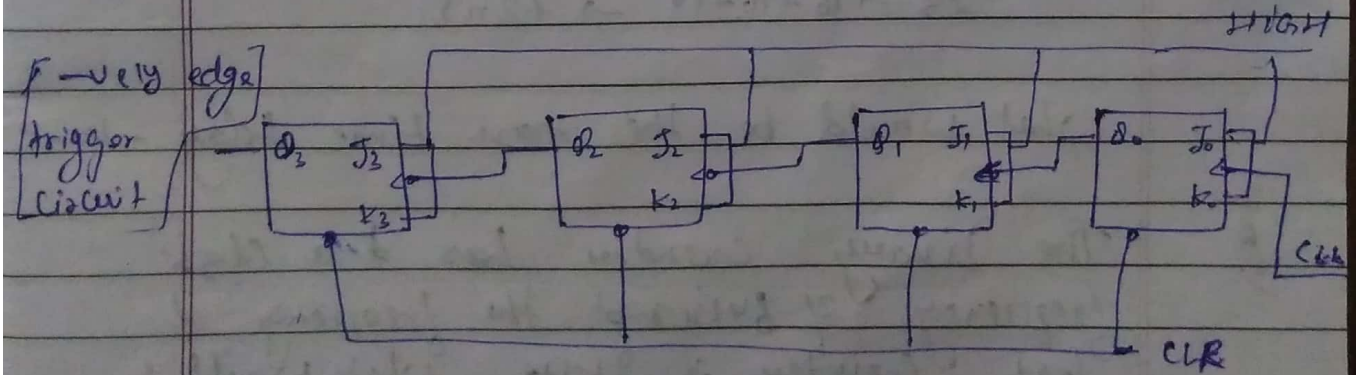
Q_3	Q_2	Q_1	Q_0
0	0	0	0
0	0	0	1
0	0	1	0
0	0	1	1
0	1	0	0
0	1	0	1
0	1	1	0
0	1	1	1
1	0	0	0
1	0	0	1
1	0	1	0
1	0		
1		1	1
1	1	0	0
1	1	0	1
1	1	1	0
1	1	1	1

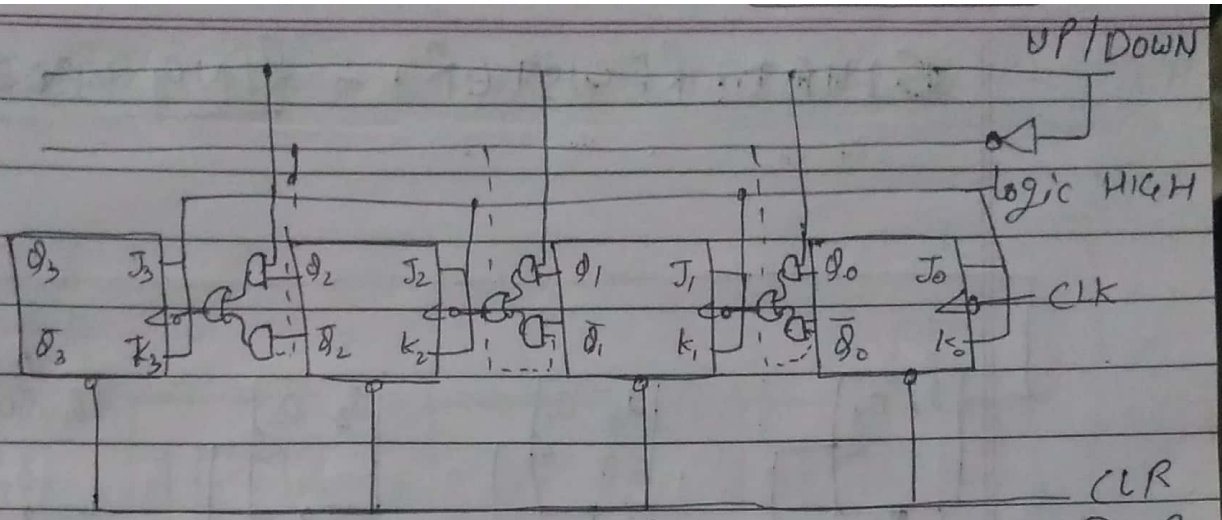


∴ Modulus → 5

No. of diff states through which counter passes

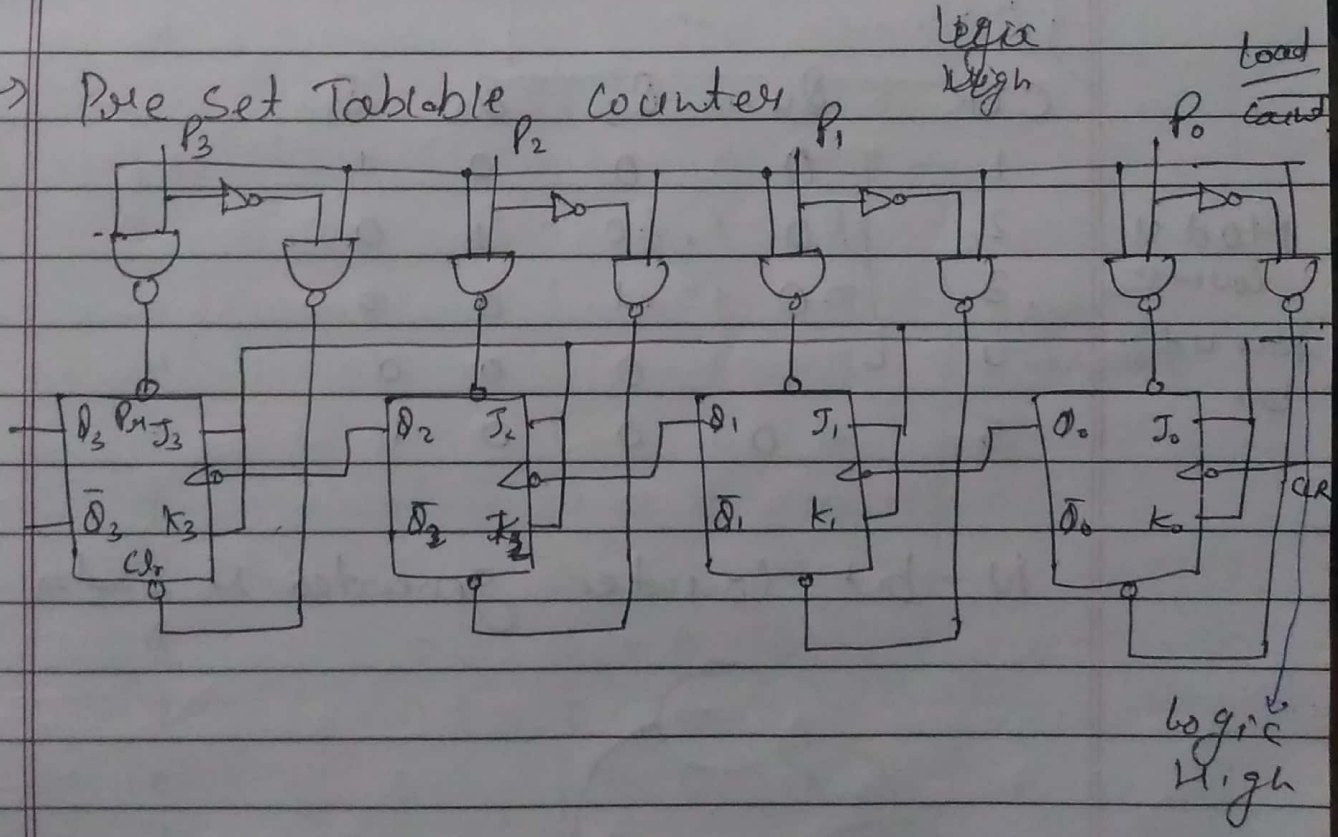
(up counter)





→ ASYNCHRONOUS UP/DOWN COUNTER

→ Pre Set Tablable counter

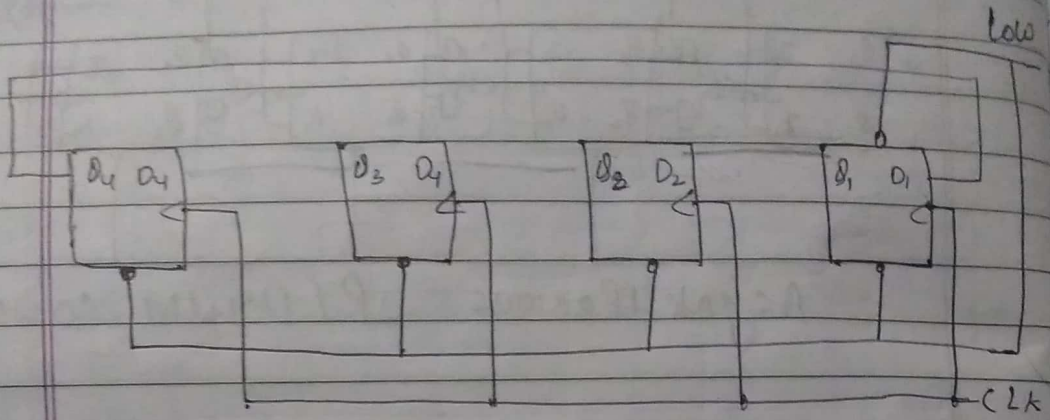


$P_R \rightarrow 1 \rightarrow$ set

$C_R \rightarrow 1 \rightarrow$ Reset

Date :
Page No.

SHIFT REGISTER – RING COUNTER

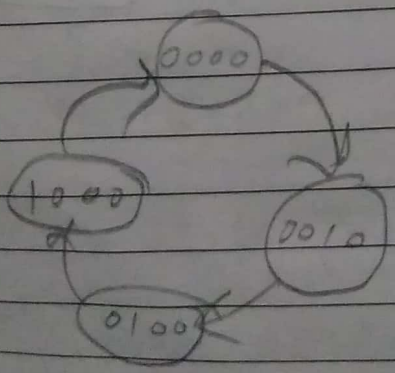


CLK	Q_4	Q_3	Q_2	Q_1
1	0	0	0	1
2	0	0	1	0
3	0	1	0	0
4	1	0	0	0
5	0	0	0	1

Mod 4 counter

Have 4 diff bit

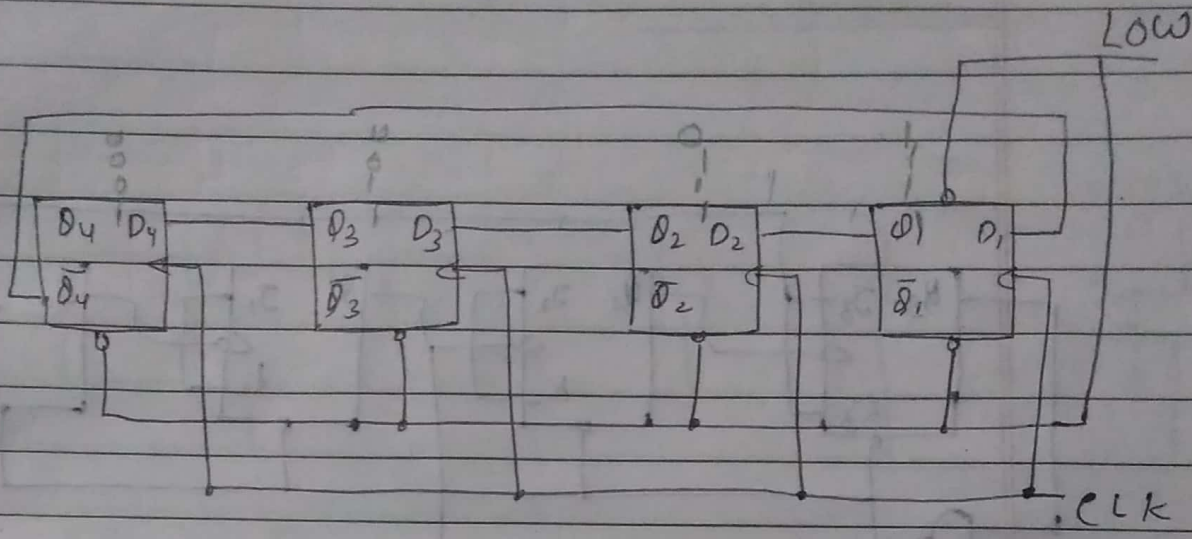
N-bit counter generates N states



Present	Next
0000	0010
0010	0100

Date :
Page No.

SHIFT REGISTER - TWISTED RING COUNTER



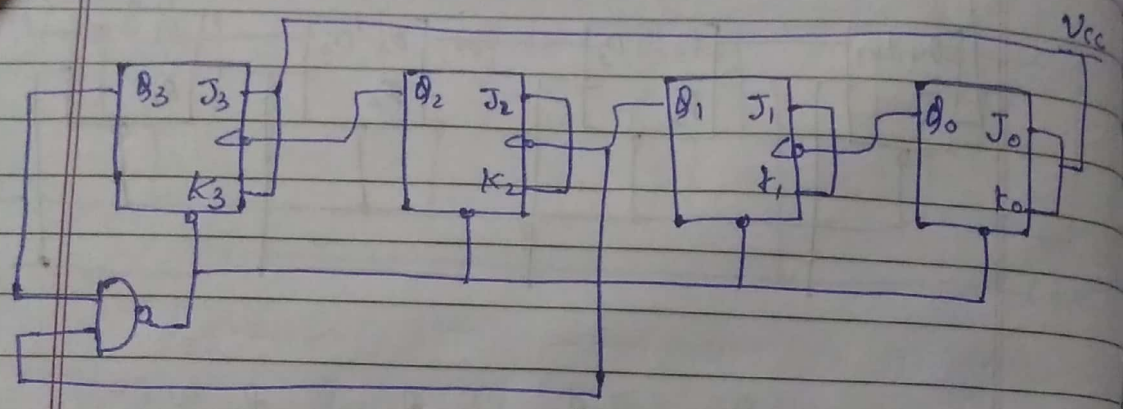
CLK	Q_4	Q_3	Q_2	Q_1
1	0	0	0	1
2	0	0	1	1
3	0	1	1	1
4	1	1	1	1
5	1	1	1	0
6	1	1	0	0
7	1	0	0	0
8	0	0	0	0
9	0	0	0	1
10	0	0	1	1

Mod 8

N-bit generates 2N-States

Date :
Page No.

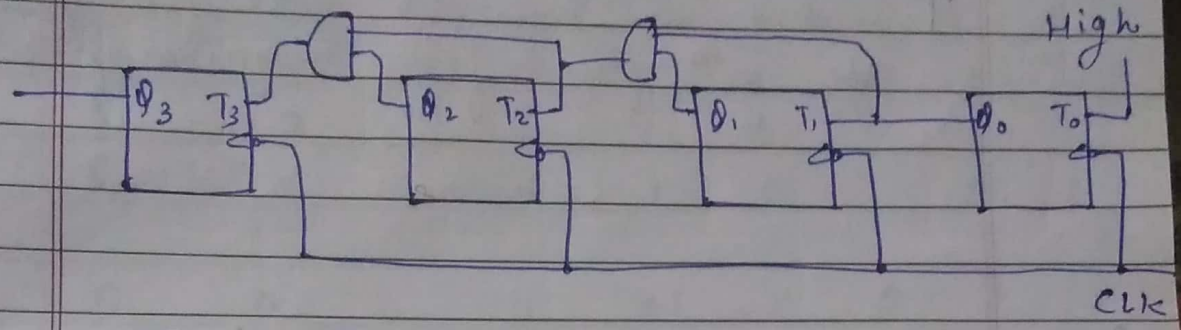
DECADE COUNTER - ASYNCHRONOUS



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

Date :
Page No.

SYNCHRONOUS COUNTER - SERIES CARRY



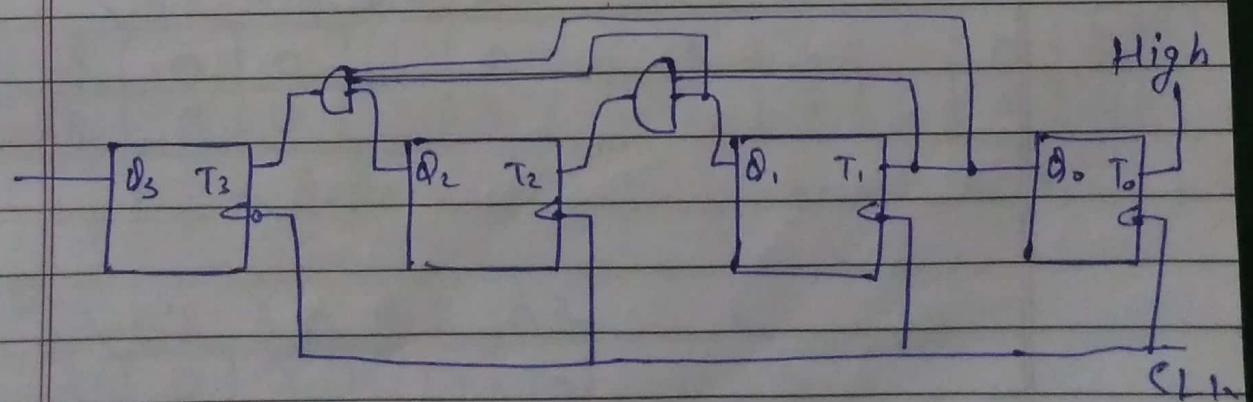
$T_{min} = T_f + (n-2) T_G$

$n \rightarrow$ no. of F.F.
$2 \rightarrow$ no. of AND Gates
if no. of AND Gates are 3
$\therefore T_f + (n-3) T_G$

$T_0 = 1$
$T_1 = Q_0 \bar{Q}_3$
$T_2 = Q_0 Q_1$
$T_3 = Q_2 Q_1 Q_0$

Truth Table same as Asynchronous

SYNCHRONOUS COUNTER - PARALLEL CARRY



Date :
Page No.

Non-SEQUENTIAL COUNTING – using D-design, JK-design, T-design

0, 2, 1, 3, 5

	0	1	2	3	4	5	6	7
Q_2	0	0	0	0	1	1	1	1
Q_1	0	0	1	1	0	0	1	1
Q_0	0	1	0	1	0	1	0	1
D_2	0	0	0	1	x	0	x	x
D_1	1	1	0	0	x	0	x	x
D_0	0	1	1	1	x	0	x	x

D – Design

Excitation Table

o/p Transition	D
0 → 0	0
0 → 1	1
1 → 0	0
1 → 1	1

D_2

	$Q_1'Q_0'$	$Q_1'Q_0$	Q_1Q_0	Q_1Q_0'
Q_2'	0	0	1	0
Q_2	x	0	x	x

$D_2 = Q_1Q_0$

D_1

	$Q_1'Q_0'$	$Q_1'Q_0$	Q_1Q_0	Q_1Q_0'
Q_2'	1	1	0	0
Q_2	x	0	x	x

$D_1 = Q_1'Q_2'$

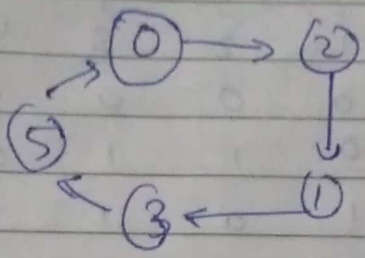
D_0

	$Q_1'Q_0'$	$Q_1'Q_0$	Q_1Q_0	Q_1Q_0'
Q_2'	0	1	0	1
Q_2	x	0	x	x

$D_0 = Q_1 + Q_0Q_2'$

Date :
Page No.

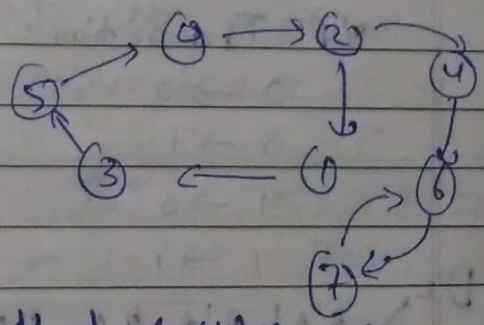
D_2	0	0	0	1	0	0	0	1
D_1	1	1	0	0	0	0	0	0
D_0	0	1	1	1	0	0	1	1



BUSH : Counter rotate blw the used state

BUSHless: Counter rotate blw the unused state.

eg:-



How to check that counter is in Bush state or in bushless.

unused States are 4 6 7

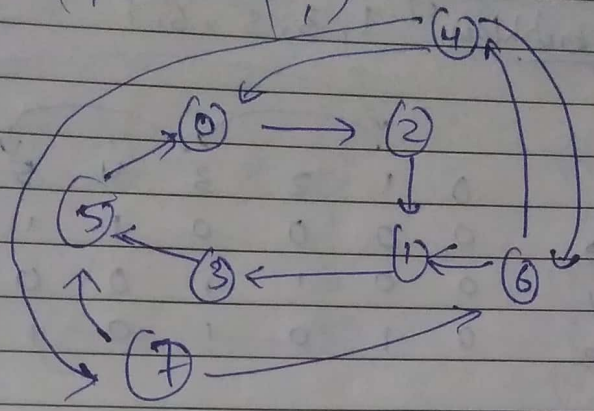
(output) (new state) (inputs)

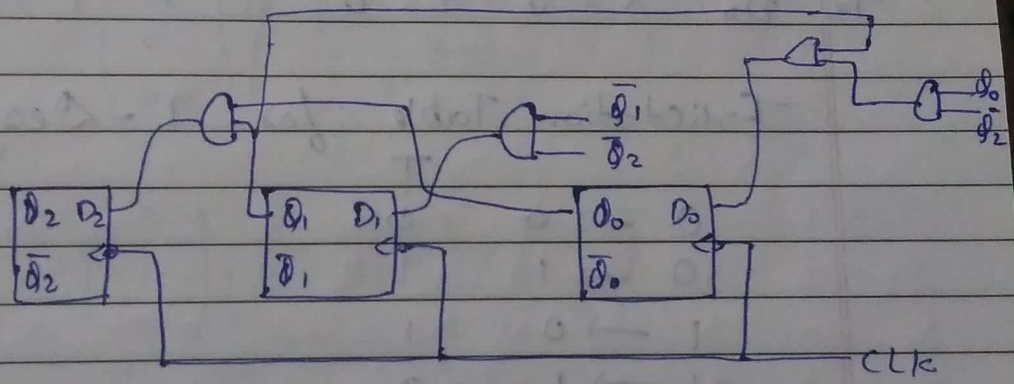
(4) $\begin{pmatrix}1\\0\\0\end{pmatrix}\begin{pmatrix}0\\0\\0\end{pmatrix} \longleftrightarrow \begin{pmatrix}0\\0\\0\end{pmatrix}$ (0)

(6) $\begin{pmatrix}1\\1\\0\end{pmatrix}\begin{pmatrix}0\\0\\1\end{pmatrix} \longleftrightarrow \begin{pmatrix}0\\0\\1\end{pmatrix}$ (1)

Date :
Page No.

(7) $\begin{pmatrix}1\\1\\1\end{pmatrix}\begin{pmatrix}1\\0\\1\end{pmatrix} \leftrightarrow \begin{pmatrix}1\\0\\1\end{pmatrix}$





ues 0, 4, 2, 1, 6, 0 using JK Design

	0	1	2	3	4	5	6	7
Q_2	0	0	0	0	1	1	1	1
Q_1	0	0	1	1	0	0	1	1
Q_0	0	1	0	1	0	1	0	1

J_2	1	1	0	x	x	x	x	x
k_2	x	x	x	x	1	x	1	x
J_1	0	1	x	x	1	x	x	x
k_1	x	x	1	x	x	x	1	x
J_0	0	x	1	x	0	x	0	x
k_0	x	1	x	x	x	x	x	x

Excitation Table of JK

	J	K
0 → 0	0	x
0 → 1	1	x
1 → 0	x	1
1 → 1	x	0

J_2

	$Q_1'Q_0'$	$Q_1'Q_0$	Q_1Q_0	Q_1Q_0'
Q_2'	1	1	x	0
Q_2	x	x	x	x

$J_2 = Q_1'$

J_1

	$Q_1'Q_0'$	$Q_1'Q_0$	Q_1Q_0	Q_1Q_0'
Q_2'	0	1	x	x
Q_2	1	x	x	x

$J_1 = Q_0 + Q_2$

Date :
Page No.

J_0

	$Q_1'Q_0'$	$Q_1'Q_0$	Q_1Q_0	Q_1Q_0'
Q_2'	0	X	X	1
Q_2	0	X	X	0

$J_0 = Q_1 Q_2'$

K_2

	$Q_1'Q_0'$	$Q_1'Q_0$	Q_1Q_0	Q_1Q_0'
Q_2'	X	X	X	X
Q_2	1	X	X	1

$K_2 = 1$

K_1

	$Q_1'Q_0'$	$Q_1'Q_0$	Q_1Q_0	Q_1Q_0'
Q_2'	X	X	X	1
Q_2	X	X	X	1

$K_1 = 1$

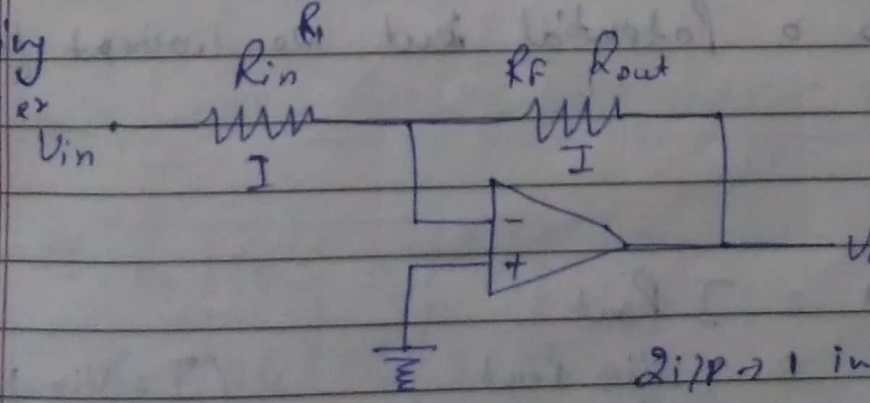
K_0

	$Q_1'Q_0'$	$Q_1'Q_0$	Q_1Q_0	Q_1Q_0'
Q_2'	X	1	X	X
Q_2	X	X	X	X

$K_0 = 1$

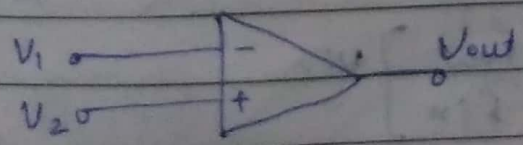
* Digital To Analog Converter

Inverting Amplifier

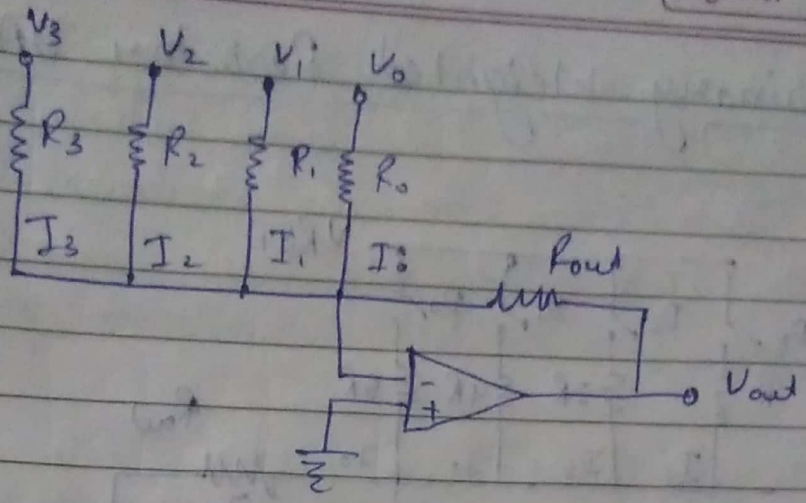


IC (-741)
14 bi

2 i/p → 1 inverting input
1 non-inverting input



Date :
Page No.



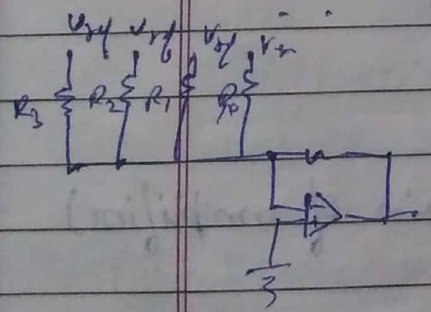
$$I = I_0 + I_1 + I_2 + I_3$$

$$I_0 = \frac{V_0}{R_0}$$

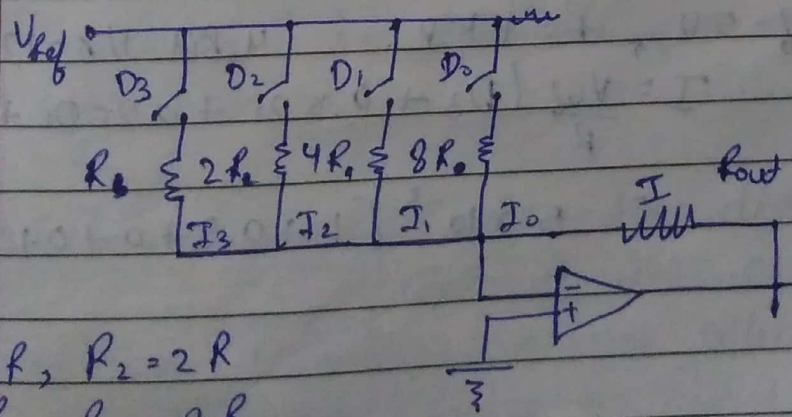
$$V_{out} = \left[\frac{V_3}{R_3} + \frac{V_2}{R_2} + \frac{V_1}{R_1} + \frac{V_0}{R_0}\right] R_{out}$$

If ~~three~~ a, Common V_{ref}.

$$\therefore V_{out} = \left[\frac{1}{R_3} + \frac{1}{R_2} + \frac{1}{R_1} + \frac{1}{R_0}\right] V_{ref.} R_{out}$$



Binary weighted Registors

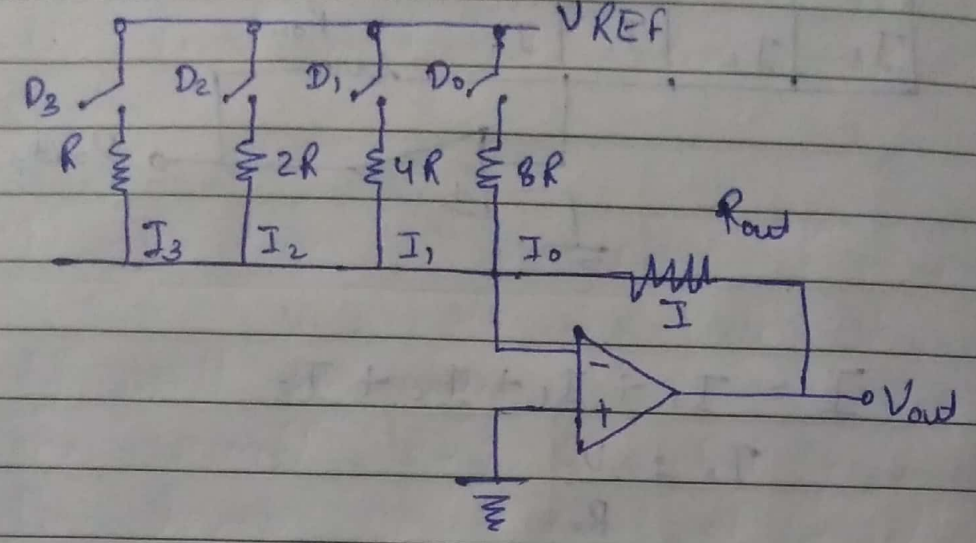


$R_3 = R$, $R_2 = 2R$

$R_1 = 4R$, $R_0 = 8R$

Date :
Page No.

Binary Weighted Resistor DAC



$$\therefore I = \frac{V_{ref}}{R}\left[1 + \frac{1}{2} + \frac{1}{4} + \frac{1}{8}\right] = 1.875\,\frac{V_{ref}}{R}$$

$$I = \frac{V_{ref}}{R}\left[D_3 + 0.5D_2 + 0.25D_1 + 0.125D_0\right]$$

$$V_{out} = -\frac{R_f}{R_1} V_{out}$$

$$V_{out} = \left(1 + \frac{R_f}{R_1}\right) V_{out} \quad \text{(gain of amplifier)}$$

eg: $V_{ref} = 5V$, $f = 1\,K\Omega$, 4 Bit I/P = 1101

$$I = \frac{V_{ref}}{R}\left(D_3 + 0.5D_2 + 0.25D_1 + 0.125D_0\right)$$

$$= 5 \times 10^{-3}\left[1 + 0.5 + 0 + 0.125\right)$$

DAC → Digital to analog Converter

Date :
Page No.

Specification of DAC

(1) Resolution :- is the ratio of the LSB implement to the maximum output.

$$\frac{1}{2^n - 1}$$

eg. 1101, $\frac{1}{2^n - 1} = \frac{1}{2^4 - 1} = \frac{1}{15}$

it is also define as the smallest implement in the voltage that can be observed by the circuit and depends Primarily on the no. of i/p bits in the I/P words. Greater the no. of bits

(2) Accuracy :- It refers to how close each output current is to its ideal value. It is a measure of difference blw the actual outputs voltage & the expect o/p voltage.

(3) Monotonicity :- A Monotonic DAC is 1 that produces an o/p for each successive increase in the digital input.

(4) Settling Time :- when the digital input convert the changes approx. voltage changes occurs as the switches are ~~fotto~~ on closed. There is a broken or

Date :
Page No.

finite time to require to reach the new output level. This time interval that elapse from the input change through the time what the o/p can come close enough

(5) Temperature Sensitivity : For any fix digital i/p the analog o/p will be vary
The overall temp. sensitivity is due to temp. sensitivity of a reference voltage and the offset voltage of an open. (operational amplifier)

Drawback : exact value of resister is not given.

Ques: An 8 bit DAC as a full scale o/p of 2mA and a full scale error of ±0.5%. what is the possible range of output for an i/p of 10000000 & what is the step size.

Sol Resolution $\frac{1}{2^n-1} = \frac{1}{2^8-1} = \frac{1}{255}$

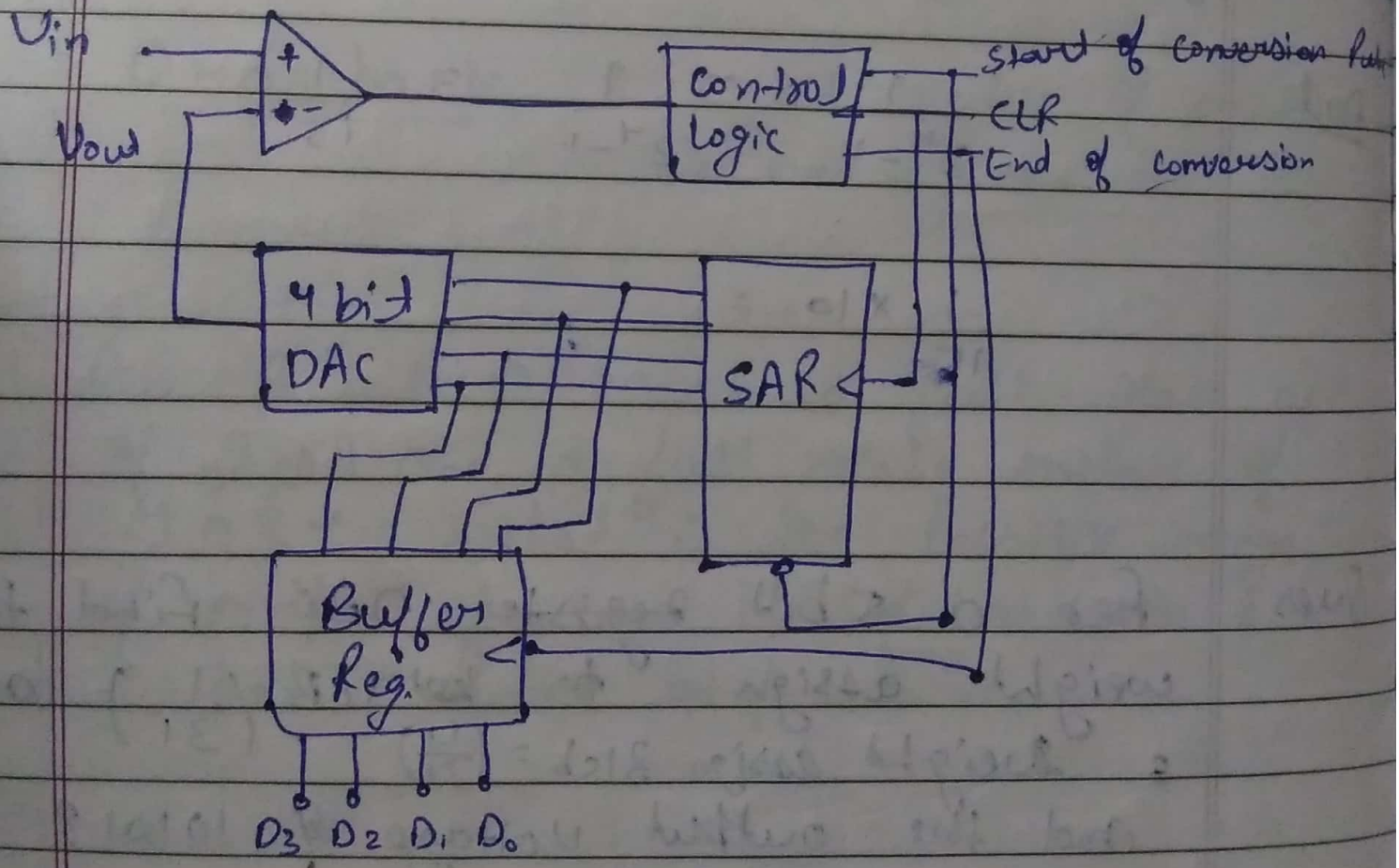
step size $= \frac{1}{255} \times I = \frac{1}{255} \times 2mA$

$$\frac{1}{2^n - 1} \leq 5\,mV \Rightarrow n = 11$$

$$\frac{1}{2^{10} - 1}$$

ANALOG TO DIGITAL CONVERTER (ADC) IC-0801

Successive Approximation Method



SAR – Successive Approximation Register

108
- 64
44
- 32
12

64 32 16 8 4 2 1
1 1 0 1 1 0 0 0

Date :
Page No. 8 4 2 1

0 0 0 0 - 0V
0 0 0 1 - 1V
1 0 0 0 - 8V

let $V_{in} = 10.8V$
$V_{in} > V_{out}$

When $V_{in} - V_{out} = 0$ then, the end of conversion takes Place.

Ques An 8 bit ~~DAC~~ ADC at a resolution of 20mV, what will be its digital outputs for an analog input of 2.17 volts

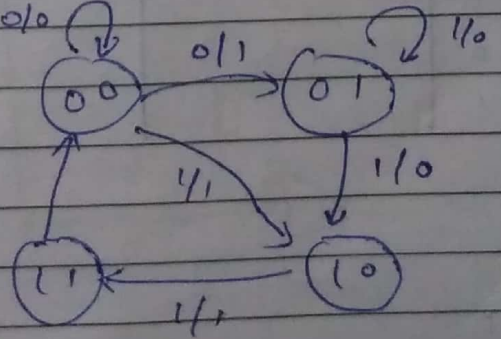
State Equation
State Table
State Diagram

Ans 11011000

2.17 = 108
20

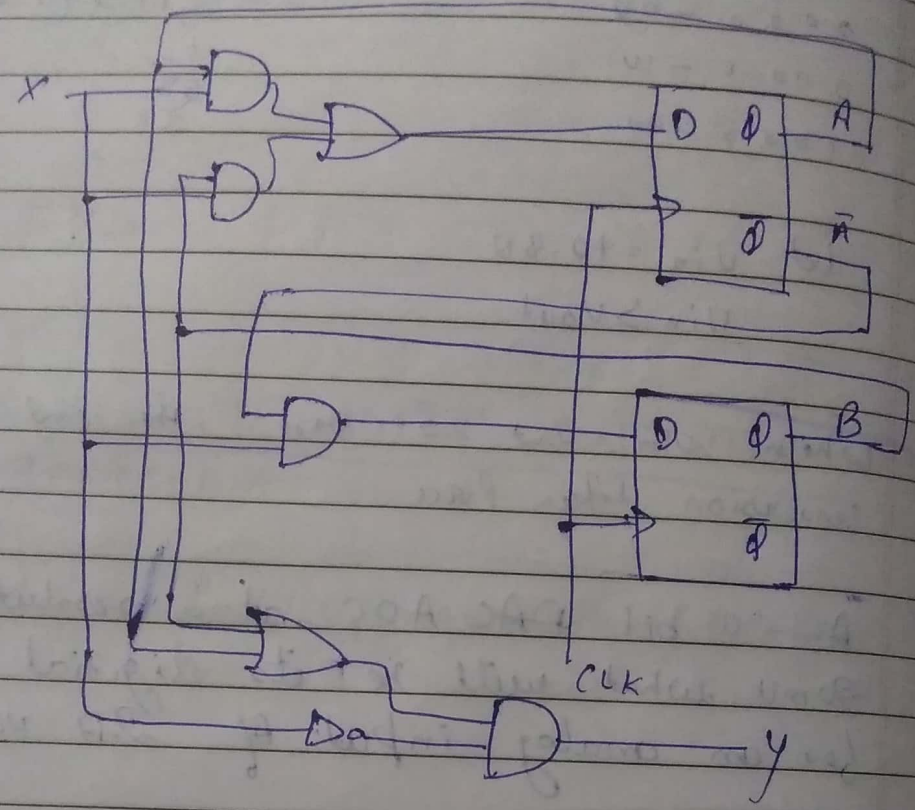
$(108)_{10}$ =

State ~~Equation~~ Diagram



Date :
Page No.

used to find the analytic behaviour of FF



PS		I/P	NS O/P		O/P
A(t)	B(t)	x	A(t+1)	B(t+1)	y
0	0	0			
0	0	1			
0	1	0			
0	1	1			
1	0	0			
1	0	1			
1	1	0			
1	1	1			

Next State of Sequential Clock Circuit.

Date :
Page No.

$A(t+1) = A(t).x(t) + B(t).x(t)$

$B(t+1) = \bar{A}(t).x(t)$

$y(t) = [A(t) + B(t)].x'(t)$

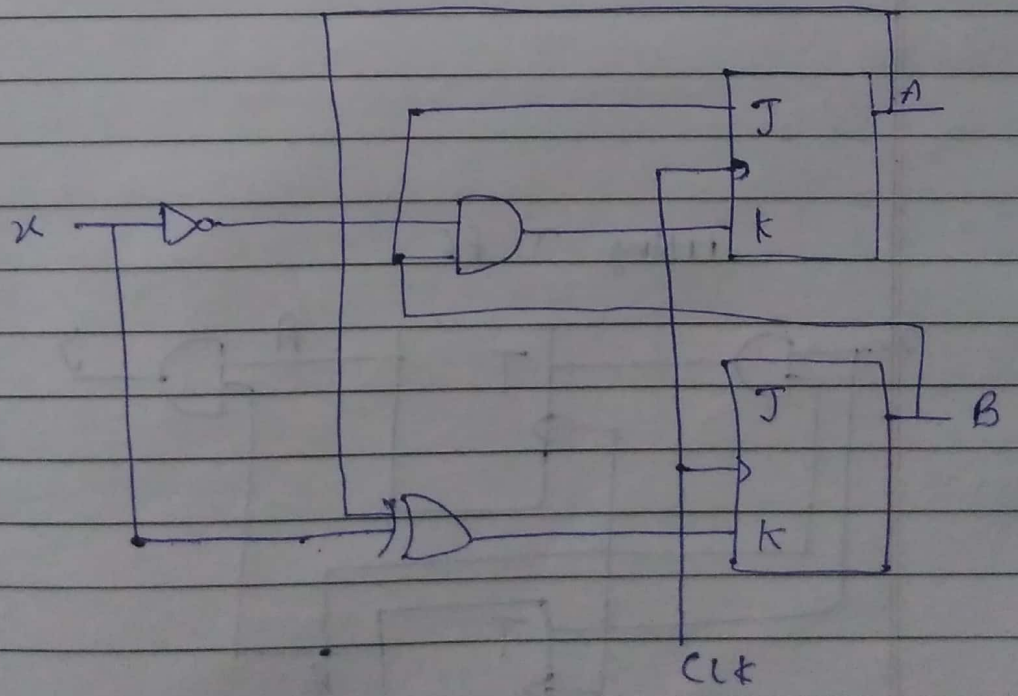
Output is the only fun. of ~~input~~ Present State

MOORE MODEL

if output is the fun. of present state as well as input

∴ MEALY MODEL

* STATE MACHINES



Date :
Page No.

State Table

P/S N/S flip flop inputs

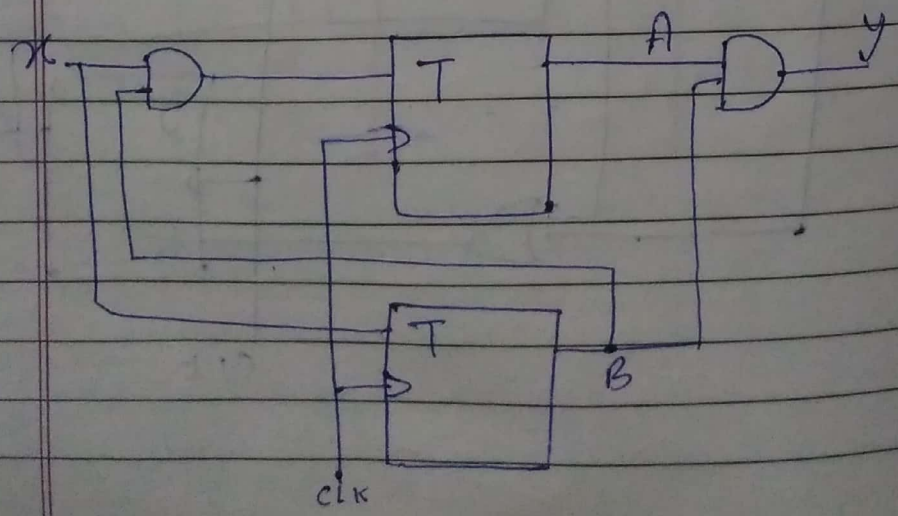
$A(t+1) = A(t) \oplus B(t) + x \cdot A(t)$

$B(t+1) = B(t)' \cdot x' + B(t)[A(t) \oplus x]'$

State Table

P/S			N/S		flip flop I/P			
A(t)	B(t)	x	A(t+1)	B(t+1)	J_A	K_A	J_B	K_B

using T FF.



$A'C + AB' \cdot x$

Date :

Page No.

$A(t+1) = A(t) \oplus B(t) \cdot x$

$B(t+1) = x \oplus B(t)$

Eq. of F.F

D → $Q(t+1) = D$

T → $Q(t+1) = Q(t) \oplus T$

J, K → $Q(t+1) = J\bar{Q} + \bar{K}Q$

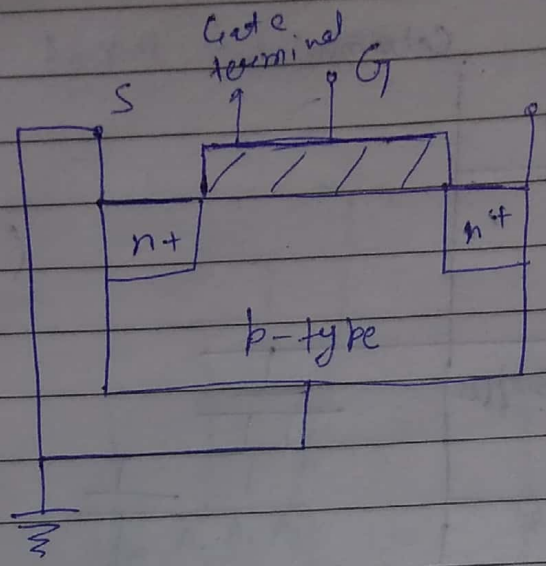
State Table

PS			NS		flip flop i/p
A(t)	B(t)	x	A(t+1)	B(t+1)	J_A K_A J_B K_B
0	0	0			
0	0	1			
0	1	0			
0	1	1			
1	0	0			
1	0	1			
1	1	0			
1	1	1			

Date :
Page No.

p-Type semiconductor (— metal layer)

MOS



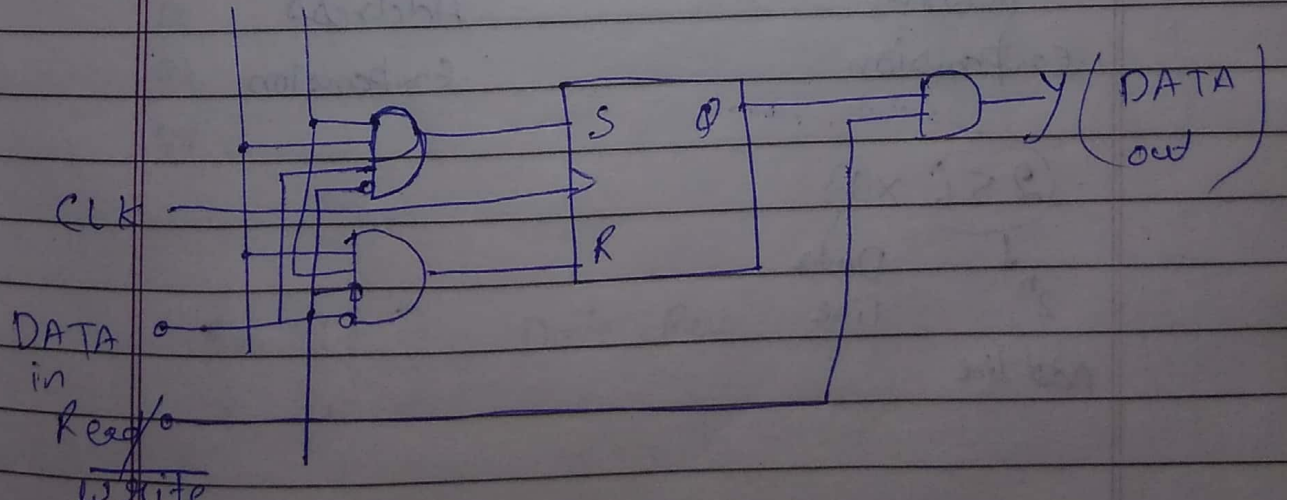
n+ → heavily load

BJT → Current Controlled Device

MOS - 2 Type → Voltage / field Controlled device
- → Enhancement Mode
- → Depletion Mode

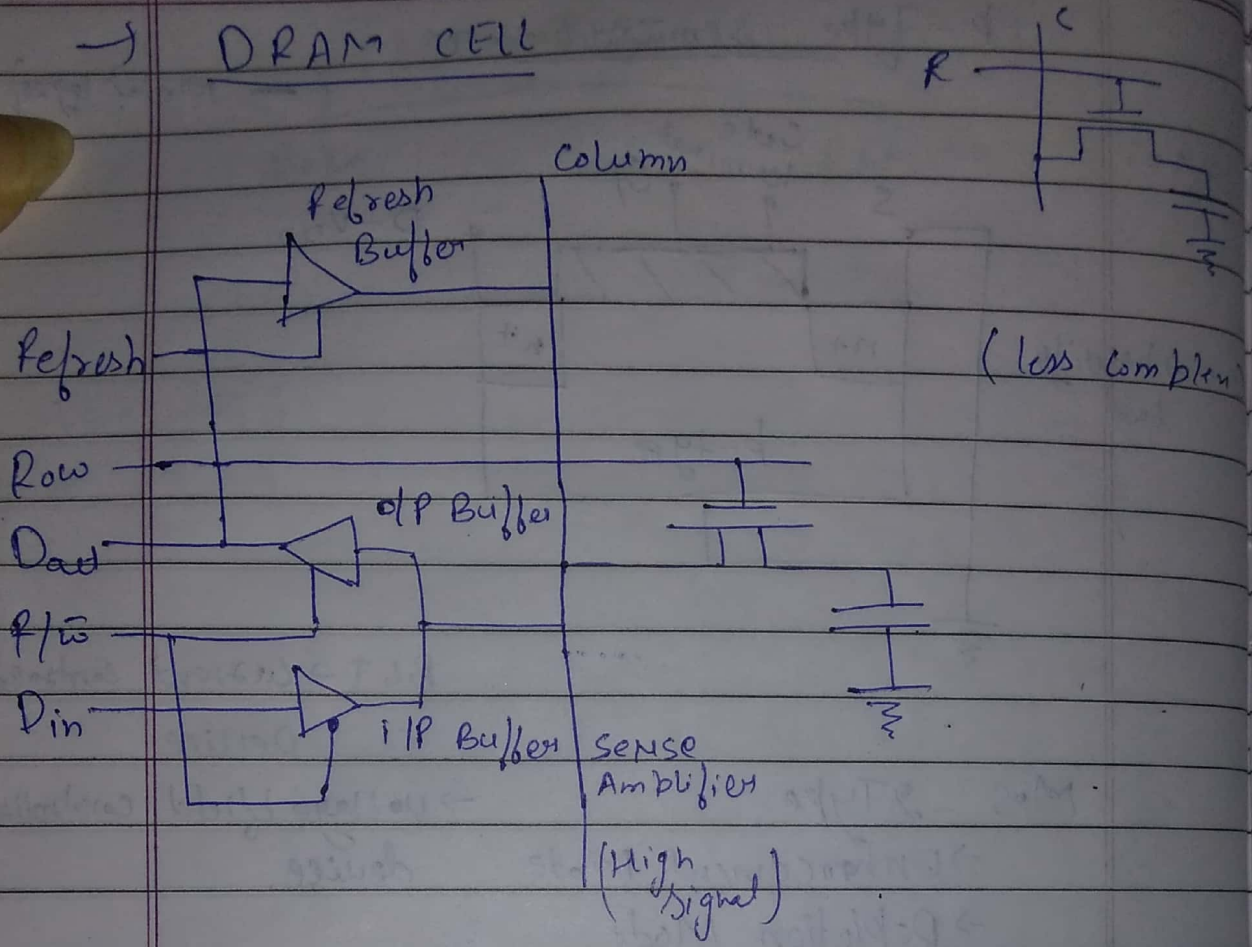
RAMS

→ SRAM CELL (More Complex)



Date :
Page No.

→ DRAM CELL



(Address)

(less complex)

(Data in) Din — 1 → Read the data

(Data out) Dout — 1 → Write the data

(Data)

MEMORY EXPANSION

Word expansion

Address Expansion

256 × 1

2^n — Add line

Data Line

in case of ROM → R/$\overline{W}$ are not required.

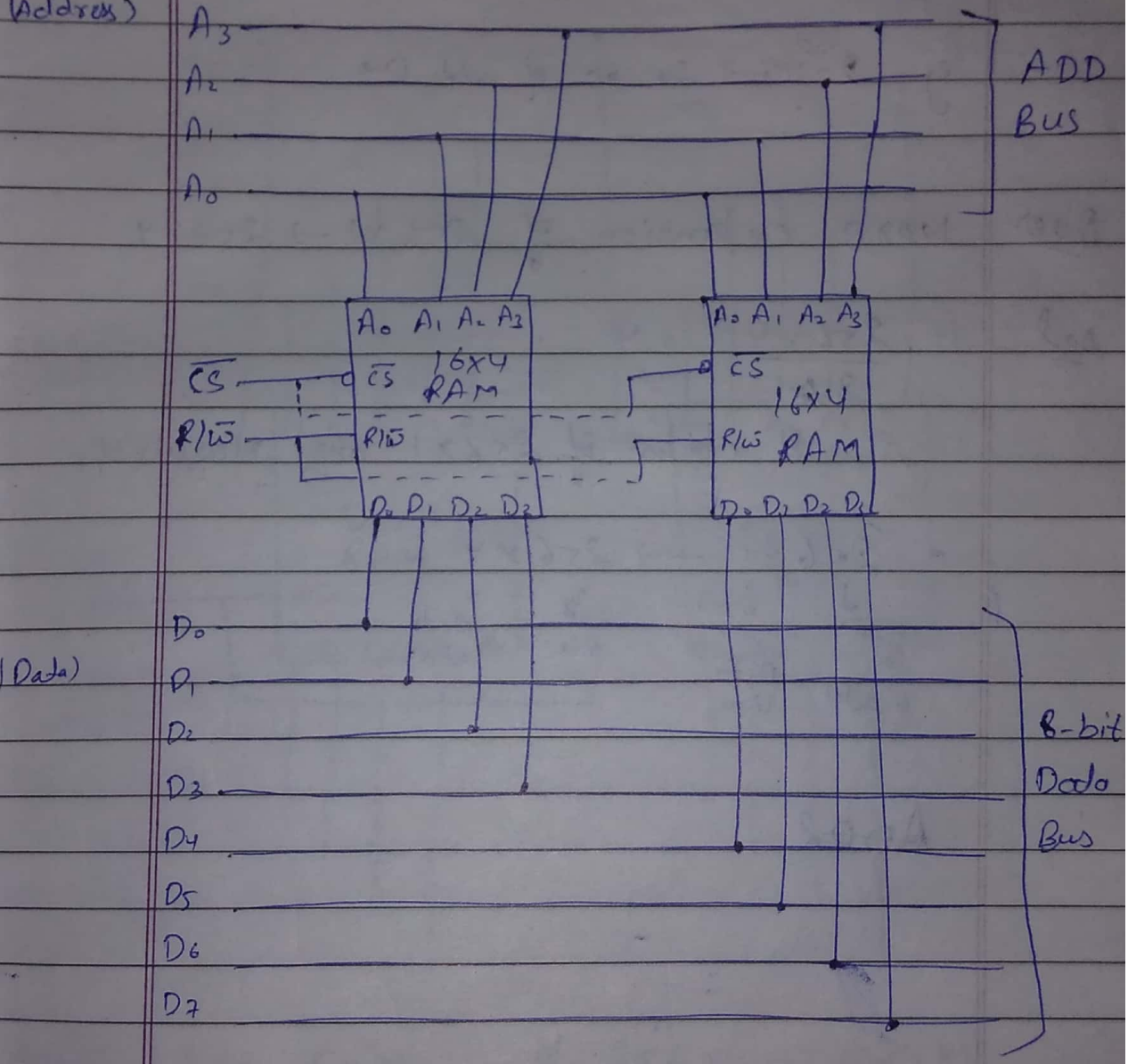
Date :
Page No.

Word Expansion

$16 \times 8 \rightarrow 16 \times 8$

$\frac{16 \times 8}{16 \times 4} = 2$

∴ 2 No. of chips are require
2 chips of 16×4



32 Bit
16 - Bit
] — Data Bus

Date :
Page No.

Address → how many location have to store info.
Word → how many bits have to store info.

$2^{(n)}$ → No. of add line

eg $2^{(20)}$ - 1MB → 20 No. of add line

Ques Word expansion of 256 x 1 → 256 x 4

Sol

$$\frac{256 \times 4}{256 \times 1} = 4$$

∴ 4 chips of 256 x 1 are required

256 x 1 → 256 x 4
8 Add line, 1 Data line → 8 Add line, 4 Data line

2^8 → 8 Add line, 1 Data line

ADDR

CS → chip select

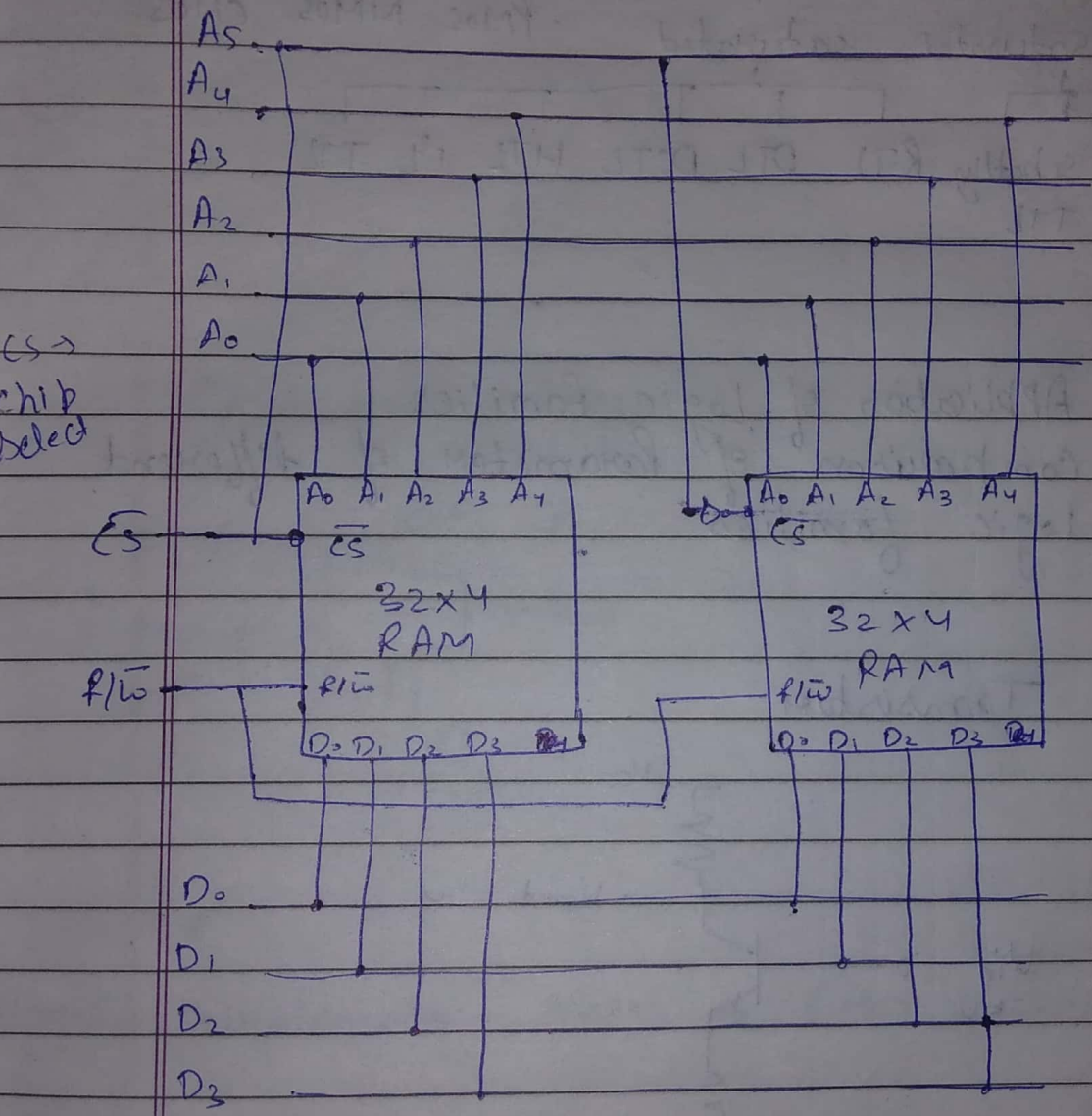
Ques 1

Ques 2

Date :
Page No.

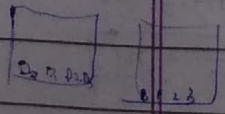
→ Address expansion

32 → x4 → 64 x 4



Ques 1 Add. Expansion of 256 x 4 → 1024 x 4

Ques 2 16 x 4 → 32 x 8 (min cnt.)



Date :
Page No.

VHDL

LIBRARY IEEE
USE IEEE.STD_LOGIC_1164.ALL

ARCHITECTURE BEHAVIOR of ABC IS BEGIN
( )
entity

END arch ABC;

Name of arch.

ENTITY ABC IS
PORT (
A, B : IN BIT;
C : OUT BIT);
END ABC

LIBRARY
ENTITY
ARCHITECTURE

(entity should not be a key word)

- Data flow Modeling
- Behavioral Modeling
→ Structural Modeling

```
LIBRARY IEEE;
USE IEEE.STD_LOGIC_1164.ALL;

ENTITY NAND_GATE IS
PORT (A, B : IN BIT;
      X : OUT BIT);
END NAND_GATE;

ARCHITECTURE arch NAND OF NAND_GATE IS
BEGIN
```

Xilinx Kit → SPARTAN-3E / VERITEX-6

XILINX - Software

Date :
Page No.

x <= A NAND B
END archNAND

Symbols

<= Assignment of signals
:= Assignment of variables
=> Syntax used for case statements
= value assignment

PRE DEFINED DATA TYPES

① - SIGNAL x : BIT ; or STD_LOGIC ;
② - SIGNAL y : BIT_VECTOR (3 DOWNTO 0);
③ - SIGNAL w : BIT_VECTOR (0 TO 7);

User Defined Data Type

TYPE my_integer IS RANGE -64 To 64;

ENUMERATED DATA TYPE

TYPE COLOR IS (RED, GREEN, BLUE, YELLOW);
↳ (2 bit used)

① x is declared as a 1 digit signal of type BIT or std logic

② 4 Bit vector, with left side bit are most MSB
③ Right most bit represents MSB

Date :
Page No.

3 Types of Modelling

Concurrent / Sequentially

DATA Flow (Simple modelling)

Behavioral (in wic Statements are used such as Process, Procedure, function)

Structural (in wic logic circuit are used)

DATA Flow

CODE for 4 to 1 Multiplexer

```
LIBRARY IEEE;
USE IEEE_STD_LOGIC_1164.ALL;

ENTITY ABC IS
PORT (
a,b,c,d,e,f : IN STD_LOGIC;
        y : OUT STD_LOGIC);
END ABC;

ARCHITECTURE arch ABC OF ABC IS
Begin
   y <= (a AND NOTe AND NOT f) OR
        (b AND NOTe AND f) OR
        (C AND NOTe AND NOT f) OR
        (d AND e AND f);
END arch ABC;
```

Date :
Page No.

Another code for Multiplexer

```
LIBRARY IEEE;
USE IEEE.STD_LOGIC_1164.ALL;

ENTITY ABC IS
PORT (
a,b,c,d      : IN STD_LOGIC;
             y : OUT STD_LOGIC);
         SEL : IN STD_LOGIC VECTOR(1 DOWN
END ABC;                              0))

ARCHITECTURE arch ABC OF ABC IS
Begin
        y <= a WHEN SEL = "00" ELSE
             b WHEN SEL = "01" ELSE
             c WHEN SEL = "10" ELSE
             d;
END arch ABC;
```

Page No.

Code for multiplexer

```
LIBRARY IEEE;
USE IEE.STD - LOGIC - 1164. ALL;

ENTITY ABC IS
PORT (
a. b. c, d    : IN STD_LOGIC;
          y : OUT STD_LOGIC;
        SEL : IN INTEGER RANGE 0 TO 3);
END ABC;

ARCHITECTURE arch ABC OF ABC IS
Begin
     y <= a WHEN SEL = 0 ELSE
          b WHEN SEL = 1 ELSE
          c WHEN SEL = 2 ELSE
          d;
END arch ABC;
```

Code for encoder

```
LIBRARY IEEE;
USE IEEE.STD_LOGIC_1164.ALL;
ENTITY ABC IS
PORT
   (x : IN STD_LOGIC_VECTOR (7 DOWNTO 0);
    y : OUT STD_LOGIC_VECTOR (2 DOWN TO 0));
END ABC;
ARCHITECTURE arch ABC OF ABC IS
Begin
     y <= "000" WHEN x = "00000010" ELSE
          ;
```

Date :
Page No.

```
    :
    "111" WHEN x = "10000000" ELSE
    "ZZZ"
END arch_ABC;
```

Behavioural

Code for DFF

```
LIBRARY IEEE;
USE IEEE.STD_LOGIC_1164.ALL;

ENTITY ABC IS
PORT ( d, clk, rst : IN STD_LOGIC;
                q : OUT STD_LOGIC);
END ABC;

ARCHITECTURE arch ABC OF ABC IS
Begin
    PROCESS (clk, rst)
  Begin
      IF (rst = '1') THEN
          q <= '0'
  ELSE IF (clk'EVENT AND clk='1') THEN
          q <= d
  END IF;
  END PROCESS
  END arch ABC;
```

Code for Decade Counter

Date :
Page No.

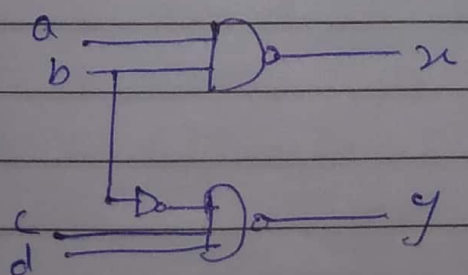
```
LIBRARY IEEE;
USE IEEE.STD_LOGIC_1164.ALL;

ENTITY ABC IS
PORT (   clk: IN STD_LOGIC;
         digit: OUT INTEGER RANGE 0 TO 9);
END ABC;

ARCHITECTURE arch ABC OF ABC IS
BEGIN
     Count: PROCESS (CLK)
     VARIABLE temp: INTEGER RANGE 0 TO 1
BEGIN
  IF (CLK'EVENT AND CLK='1') THEN
  temp := temp + 1;
  IF (temp = 10) THEN temp := 0;
  END IF;
  END IF;
digit <= temp;
```

Structural

```
LIBRARY IEEE;
USE IEEE.STD_LOGIC_1164.ALL;
ENTITY PROJ IS
PORT (a, b, c, d: IN STD_LOGIC;
          x, y: OUT STD_LOGIC);
END PROJ;
```

Date :
Page No.

```
ARCHITECTURE ABC OF PROJ IS
  COMPONENT inverter IS
  PORT (a : IN STD_LOGIC;
        b : OUT STD_LOGIC);
  END COMPONENT;
COMPONENT NAND_2 IS
  PORT (a,b : IN STD_LOGIC;
        c : OUT STD_LOGIC);
  END COMPONENT;
  COMPONENT NAND_3 IS
  PORT (a,b,c : IN STD_LOGIC;
        d : OUT STD_LOGIC);
  END COMPONENT;
  SIGNAL w : STD_LOGIC;
  BEGIN
    U01 : inverter PORTMAP (b, w);
    U02 : nand 2 PORTMAP (a, b, x);
    U03 : nand 3 PORTMAP (w, c, d, y);
END ABC;
```



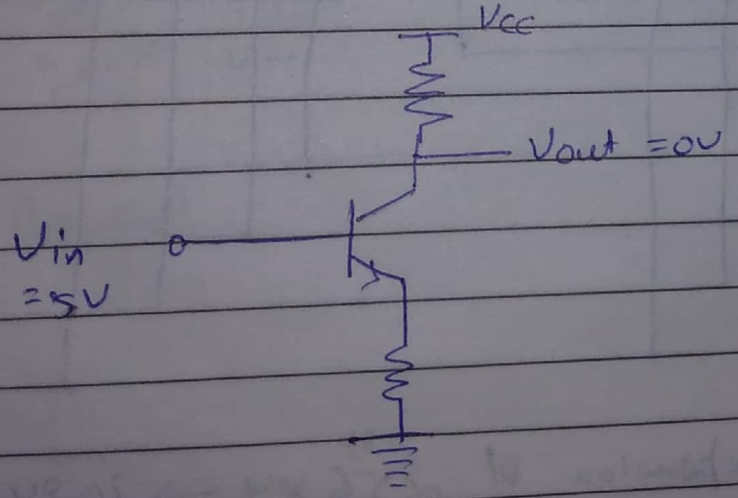
Date :
Page No.

LOGIC FAMILIES

- BIPOLAR
  - Non-saturated
    - ECL
  - saturated
    - Schottky TTL
    - RTL
    - DTL
    - DCTL
    - HTL
    - I^2L
    - TTL
- UNIPOLAR
  - PMOS
  - NMOS
  - CMOS

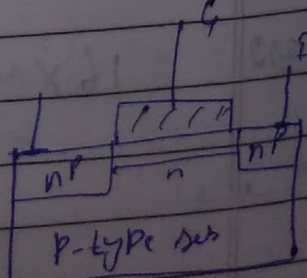
→ Application of logic families
→ Comparison of Parameter of different logic families.

Transistor



CCMOS

- depletion mode or MOSfet



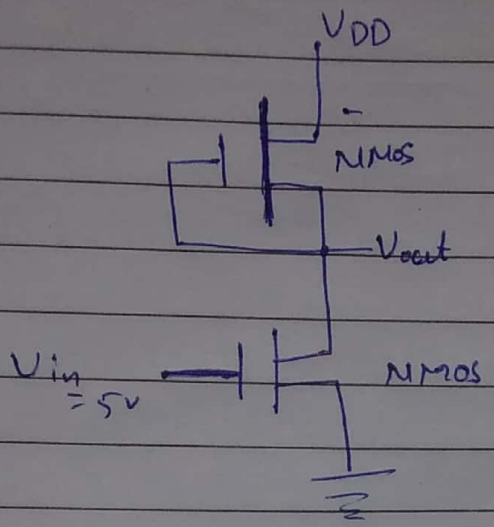
mosfet

BJT as inverter

Date : MOS as capacitance

Page No.

Circuit of inverter

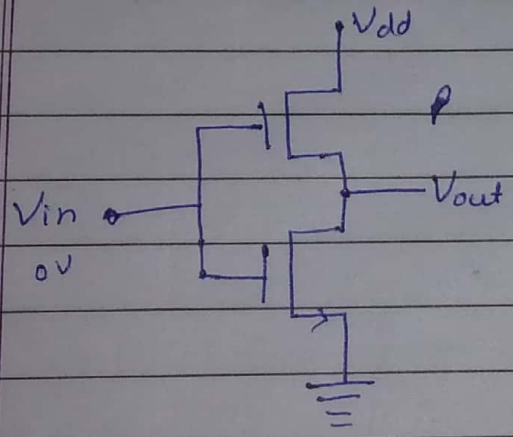


(MOS as Register)

Source terminal is grounded.

(NMOS bas inverter)

CMOS base inverter

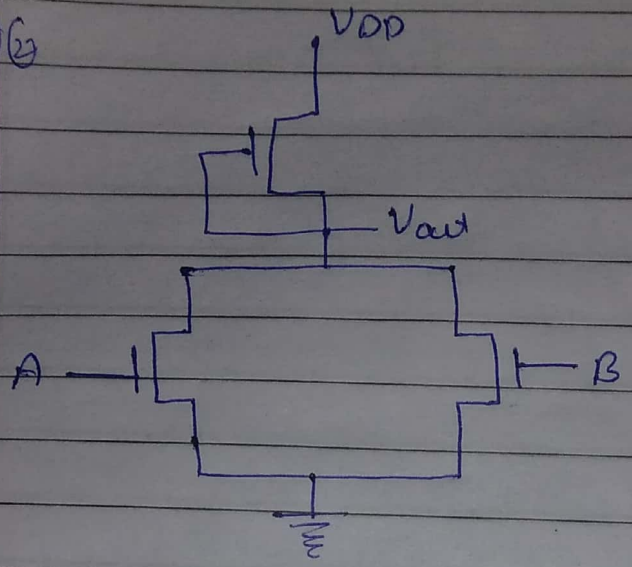


(CMOS) Complementary MOSE → PMOS & NMOS

Date :
Page No.

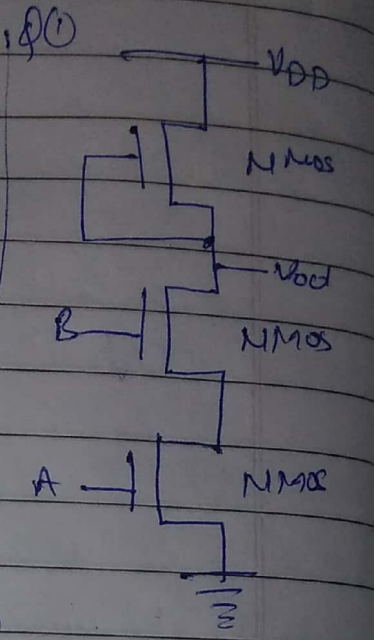
Simple depletion NMOS inverter

Q(2)



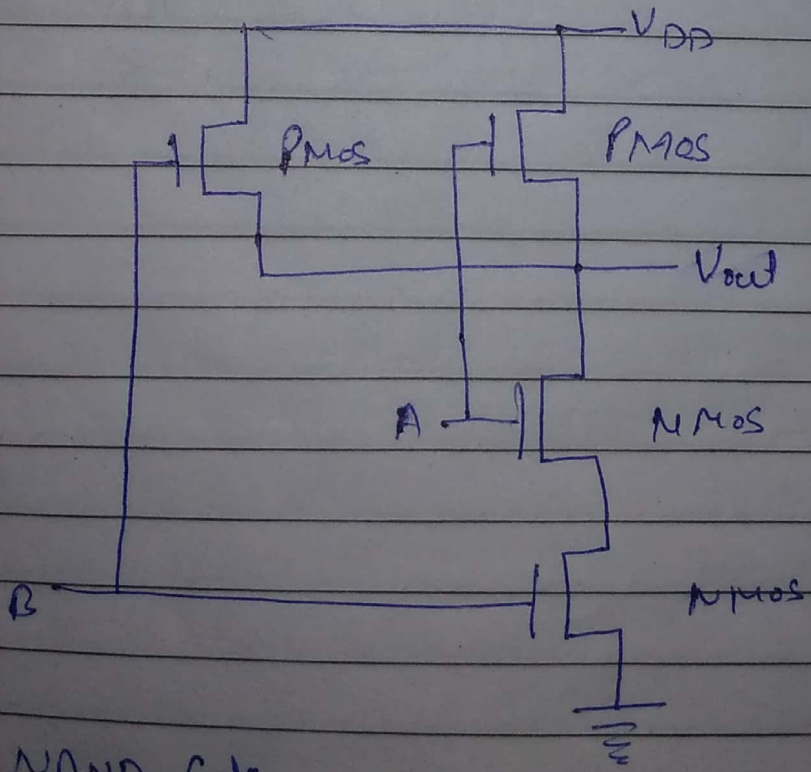
NOR Gate

Q(1)



AND Gate

Q(3)



NAND Gate

Bipolar use BJT Transistor.

RTL → Resistor Transister Logic
DTL → Doide Transister Logic
DCTL → Doide couple Transister Logic
HTL → High threosld Logic
I^2L → injection injection Logic
TTL → Transister Transister Logic
ECL → Emitor couple Logic

- RTL is the oldest logic family
- DTL is low cost family that give raise to TTL.
- HTL as the max. noise imunity
- ECL as the least Propagation delay
- I^2L has highest Packing density
- CMOS has extremly low ~~Propagation~~ power distributed
- TTL is very design logic family
  It can be used for many complex fun.

Characteristics of digital logic family.

(1) Propagation delay: ~~Average from gat~~ of

Propagation delay of logic gate is the result of finite of active devices. The Gate Prop. delay from logic 1 to logic 0 is usually different from the prop. delay from logic 0 to logic 1. Hence, Average of two delay times

Date :
Page No.

is used as a measure of Prop. delay

(2) FAN IN : No. of input that gate can handle without degrading output

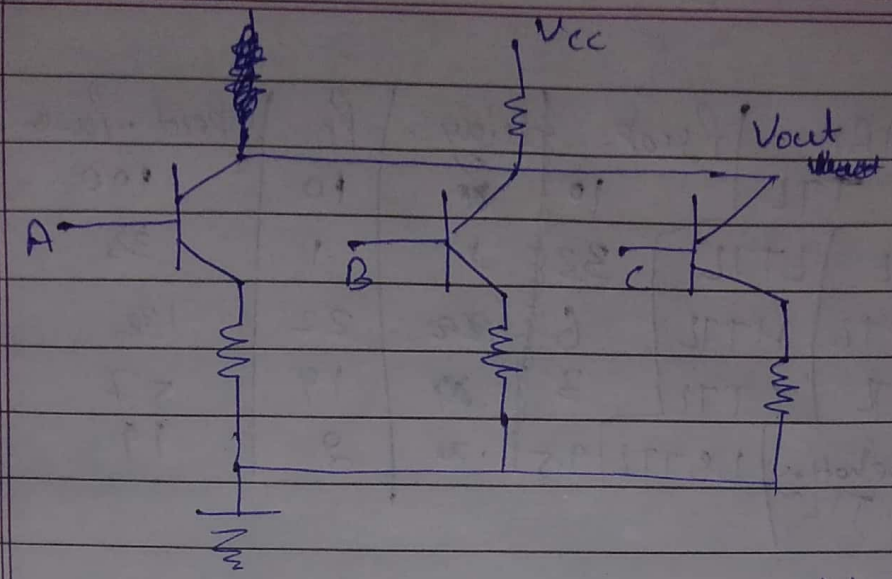
FAN out : No. of output that logic gate can handle

(3) Noise Margin:

The circuit avaibility to tolerate noise signals is noise imunity and a comtitative measure of this is called Noise Margin. So a high state noise margin is a different the lowest possible o/p & min. voltage required for high
The difference blw largest possible o/p & the max. i/p voltage for a low

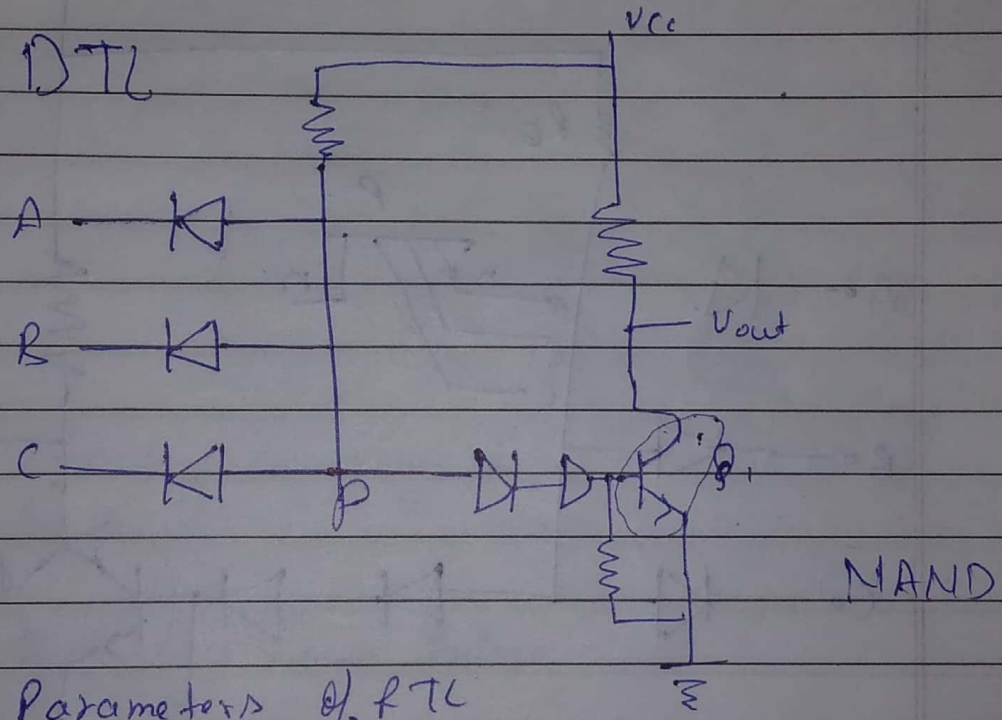
operting
Normal temp. for the digital IC's is 0 to 70°c for industrial app.
-55 to 125°c for military application

→ RTL



NOR

→ DTL



NAND

Parameters of RTL

Exhibit Poor noise margin
Poor fan out
low speed operation
High Propagation delay
High Power dissipation

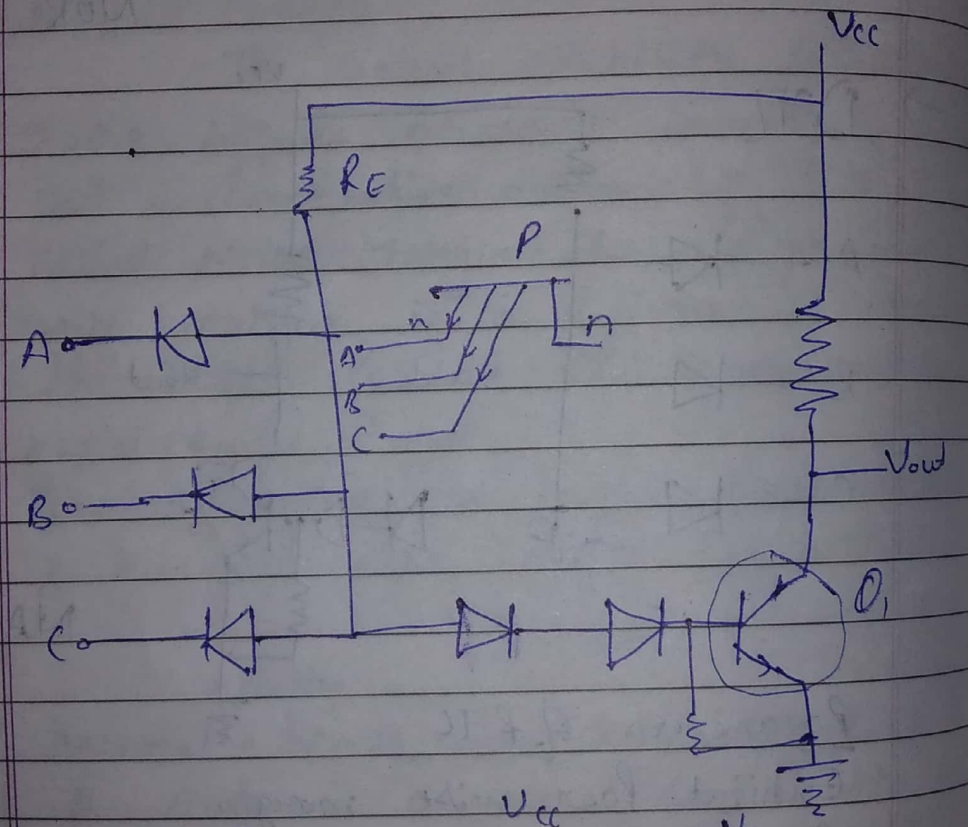
Parameters of DTL

(1) It has Propagation delay 30
(2) Power dis. 12 mv
(3) fan out is 8
(4) Noise Margin of 1V

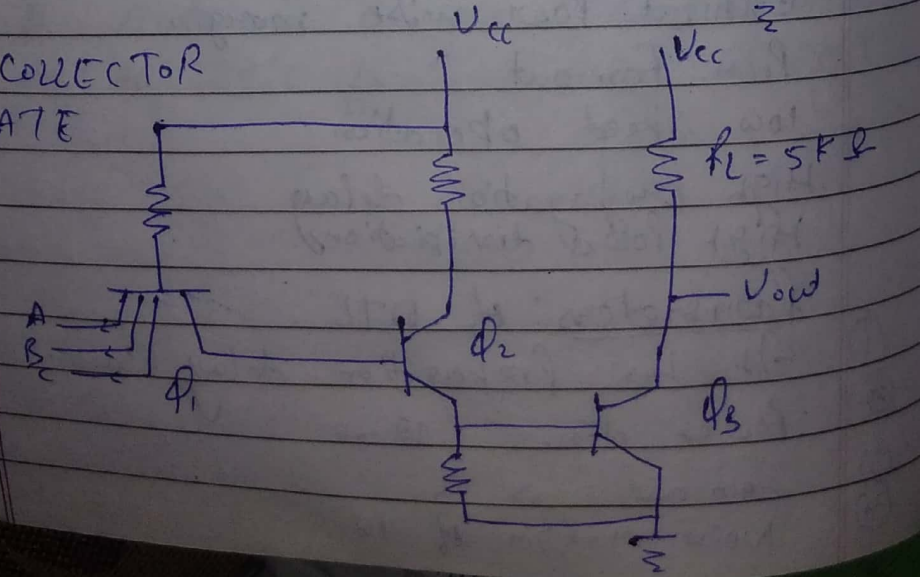
Date :
Page No.

TTL

Name	ABD	Prop. delay	Po	Speed - Power Prod
Std. TTL	TTL	10	10	100
Low Power TTL	LTTL	33	1	33
High speed TTL	HTTL	6	22	132
Schottky TTL	STTL	3	19	57
Low Power Schottky TTL	LSTTL	9.5	2	19



OPEN COLLECTOR O/P GATE

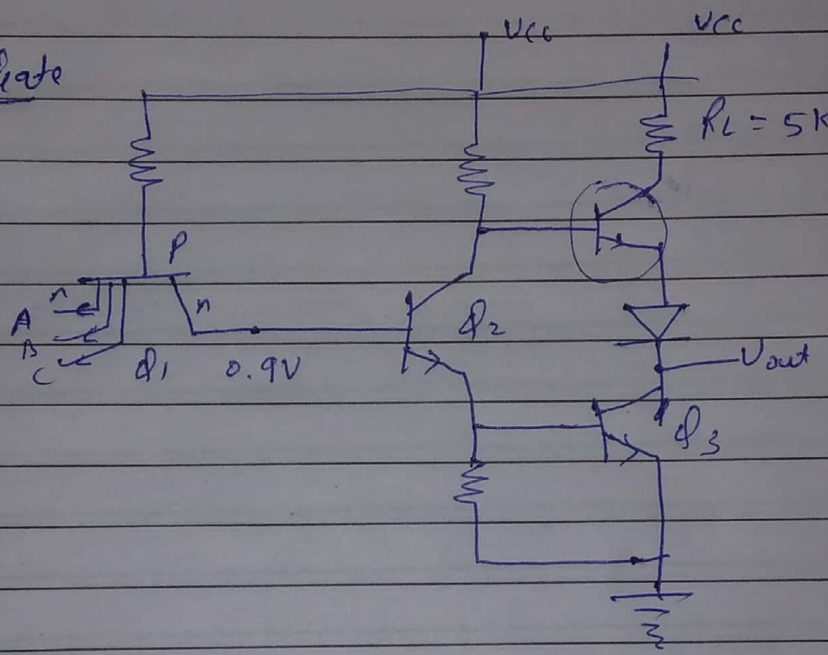


Date :
Page No.

TTL
OPEN COLLECTOR
TOTEM POLE

Q. Specific application of open Collector o/p Gate

To Tem Gate



Pulled → connecting to up
Pull down → " " down

Active & Passive Pull on