

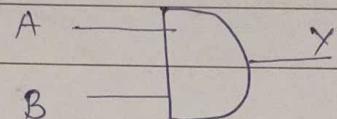
Experiment - 1

Aim: To study and verify the truth tables of logic gates using ICs and to realize basic and universal gates.

IC - 7408 And gate

Truth Table.

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

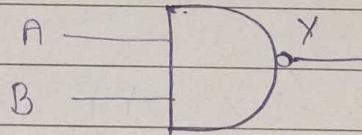


Symbol.

IC 7400 NAND gate.

Truth Table

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

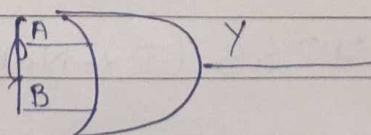


Symbol.

IC 7432 OR gate.

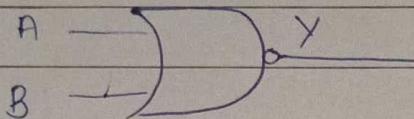
Truth Table.

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



Symbol.

IC 7402 (NOR gate)

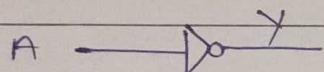


Symbol

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

Truth Table.

IC 7404 (Not gate)

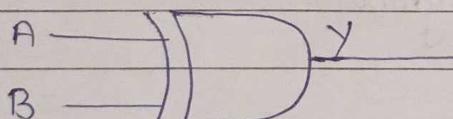


Symbol

A	y
0	1
1	0

Truth Table.

IC 7486 (Ex-or).

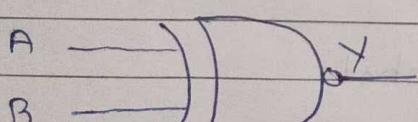


Symbol

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

Truth Table.

IC 747266 (Ex-Nor)

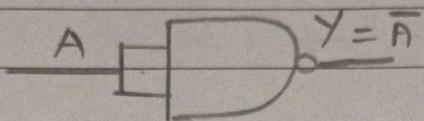


Symbol

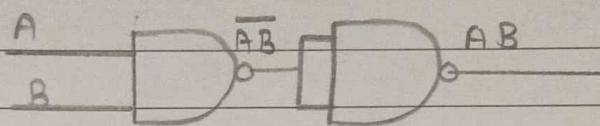
A	B	Y
0	0	1
0	1	0
1	0	0
1	1	1

Truth Table

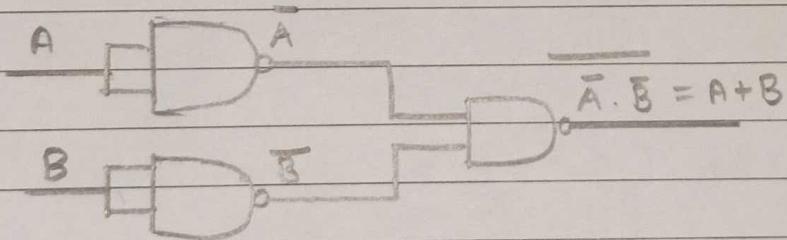
NOT using NAND



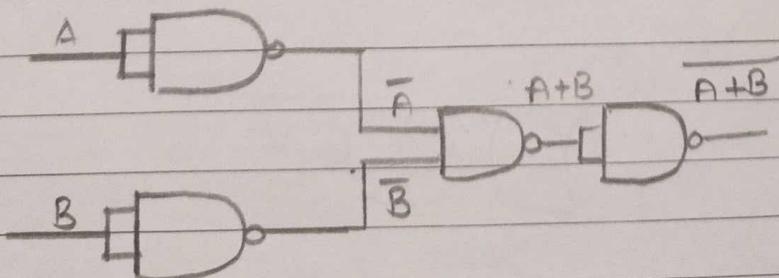
AND using NAND



OR using NAND

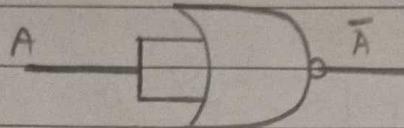


NOR using NAND

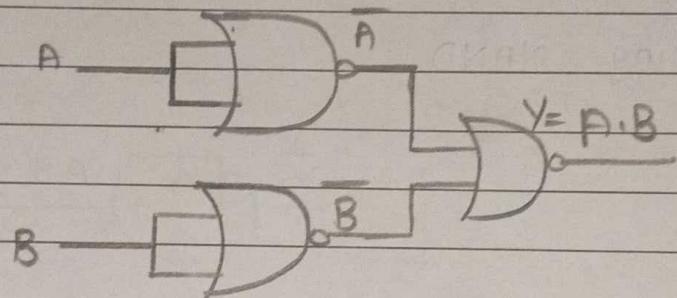


(2)

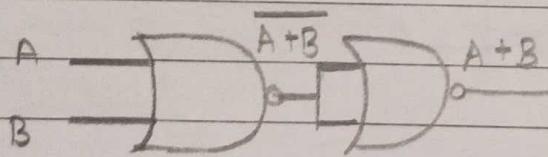
NOT gate using NOR gate.



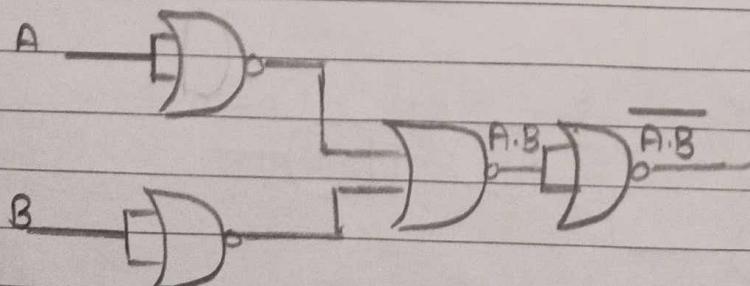
AND gate using NOR gate



OR using NOR gate



NAND using NOR gate.



Experiment-2, 3

Aim:

To do logic simplification, implementation using basic gates.

$$Y = (A+C)(CD+AC)$$

$$\text{Equation: } (A+C)(CD+AC)$$

logic simplification (kmaps).
 $(A+C)(CD+AC)$.

$$= ACD + AC + CD + AC$$

$$= ACD + ACD + A\bar{C}D + ACD + \bar{A}CD =$$

$$= ACD + A\bar{C}D + \bar{A}CD$$

$$111000110011$$

kmap:

		AC	00	01	11	10
		D	0	0	1	0
		C	0	1	1	0
1	1	0	0	1	1	0
1	1	0	0	1	1	0

$$\therefore \text{logic simplification} = C(A+D)$$

Using boolean laws:

$$(A+C)(CD+AC)$$

$$= (A+C)(A+D)C$$

$$= (A+C)C \quad [\because (A+B)(A+C) = A+BC]$$

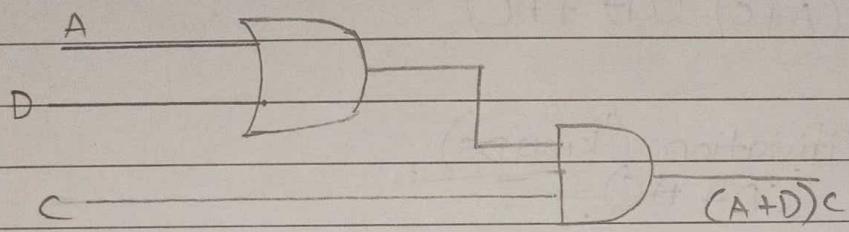
$$= AC + CDC$$

$$= AC + CD$$

$$= C(A+D).$$

* Implementation using Basic gates $(A+D)C$

Here 2 gates are required, one AND gate & one OR gate.

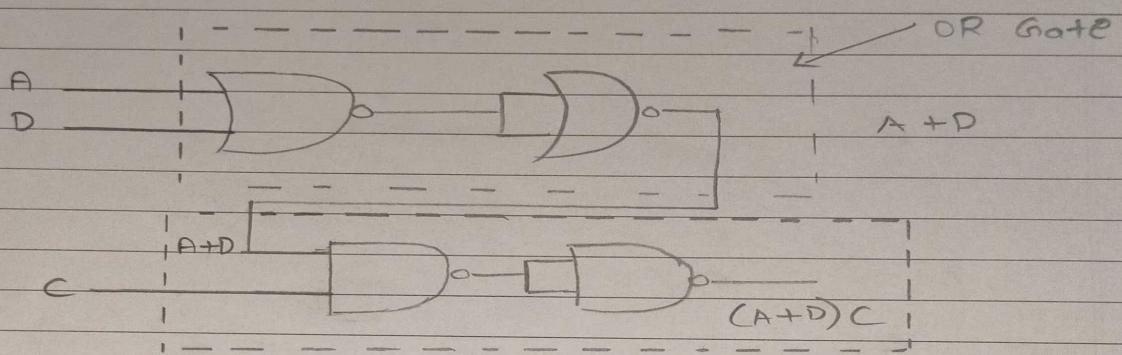


* Observation Table

A	D	C	$A+D$	$(A+D)C$
0	0	0	0	0
0	0	1	0	0
0	1	0	1	0
0	1	1	1	1
1	0	0	1	0
1	0	1	1	1
1	1	0	1	0
1	1	1	1	1

*

Implementation using universal gates
Here implementation is done by using universal gates [NOR & NAND gates].



*

Observation Table.

A	D	C	$\overline{A+D} = A+D$	$\overline{(A+D)} \cdot C = (A+D) \cdot C$
0	0	0	0	0
0	0	1	0	0
0	1	0	1	0
0	1	1	1	1
1	0	0	1	0
1	0	1	1	1
1	1	0	1	0
1	1	1	1	1

Experiment - 4

Aim: To realize binary to gray code converter
and gray code to binary code converter.

Truth Table :

Binary to gray.

Binary				Gray			
b_3	b_2	b_1	b_0	g_3	g_2	g_1	g_0
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1
0	0	1	0	0	0	1	1
0	0	1	1	0	0	1	0
0	1	0	0	0	1	1	0
0	1	0	1	0	1	1	1
0	1	1	0	0	1	0	1
0	1	1	1	0	1	0	0
1	0	0	0	1	1	0	0
1	0	0	1	1	1	0	1
1	0	1	0	1	1	1	1
1	0	1	1	1	1	1	0
1	1	0	0	1	0	1	0
1	1	0	1	1	0	1	1
1	1	1	0	1	0	0	1
1	1	1	1	1	0	0	0

60004220115

* K-map for binary to gray.

① K-map for G_0 .

$b_3 b_2 \backslash b_1 b_0$	00	01	11	10
00	0	1	0	1
01	0	1	0	1
11	0	1	0	1
10	0	1	0	1

$$G_0 = \overline{b_1} \overline{b_0} + \overline{b_1} b_0 = b_1 \oplus b_0.$$

② K-map for G_1 .

$b_3 b_2 \backslash b_1 b_0$	00	01	11	10
00	0	0	1	1
01	1	1	0	0
11	1	1	0	0
10	0	0	1	1

$$G_1 = \overline{b_1} b_2 + b_1 \overline{b_2}$$

$$G_1 = b_1 \oplus b_2$$

③ K-map for G_3 .

$b_3 b_2 \backslash b_1 b_0$	00	01	11	10
00	0	0	0	0
01	0	0	0	0
11	1	1	1	1
10	1	1	1	1

~~$$G_3 = b_3 b_2 + b_3 \overline{b_2}$$

$$= b_3 \oplus b_2$$~~

$$G_3 = b_3$$

④ K-map for G_{B2} .

$b_3 b_2 \backslash b_1 b_0$	00	01	11	10
00	0	0	0	0
01	1	1	1	1
11	0	0	0	0
10	1	1	1	1

~~$$G_2 = \overline{b_3} b_2 + b_3 \overline{b_2}$$

$$= b_3 \oplus b_2$$~~

* k-maps for Gray to Binary.

①

$G_1, G_0 \backslash G_3, G_2$	00	01	11	10
00	0	1	0	1
01	1	0	1	0
11	0	1	0	1
10	1	0	1	0

$$\begin{aligned}
 B_0 &= \overline{G_3} \overline{G_2} \overline{G_1} \overline{G_0} + \overline{G_3} \overline{G_2} \overline{G_1} G_0 \\
 &\quad + \overline{G_3} \overline{G_2} G_1 \overline{G_0} + \overline{G_3} G_2 \overline{G_1} \overline{G_0} \\
 &\quad + \overline{G_3} G_2 G_1 \overline{G_0} + \overline{G_3} \overline{G_2} G_1 G_0 \\
 &\quad + \overline{G_3} \overline{G_2} G_1 \overline{G_0} + \overline{G_3} G_2 G_1 \overline{G_0} \\
 &= G_0 \oplus G_1 \oplus G_2 \oplus G_3.
 \end{aligned}$$

②

K-map for b_1

$G_1, G_0 \backslash G_3, G_2$	00	01	11	10
00	0	0	1	1
01	1	1	0	0
11	0	0	1	1
10	1	1	0	0

$$\begin{aligned}
 B_1 &= \overline{G_1} \overline{G_0} \oplus G_3 + \overline{G_1} G_0 \overline{G_3} + G_1 \overline{G_0} G_3 + G_1 \overline{G_0} \overline{G_3} \\
 \therefore B_1 &= G_1 \oplus G_0 \oplus G_3
 \end{aligned}$$

③

K-map for b_2

$G_1, G_0 \backslash G_3, G_2$	00	01	11	10
00	0	0	0	0
01	1	1	1	1
11	0	0	0	0
10	1	1	1	1

$$B_2 = \overline{G_1} \overline{G_0} + G_1 \overline{G_0}$$

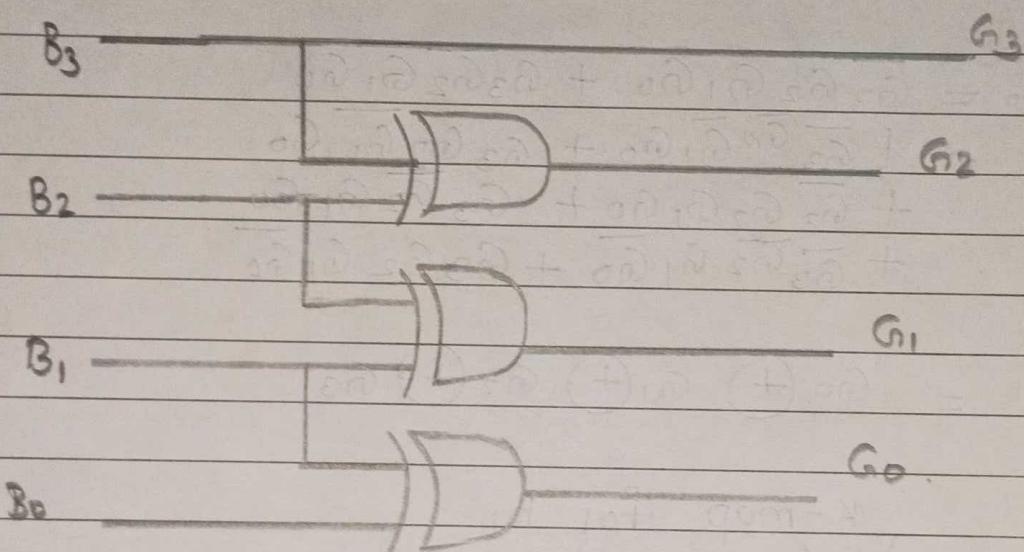
$$B_2 = G_1 \oplus G_0.$$

FOR EDUCATIONAL USE

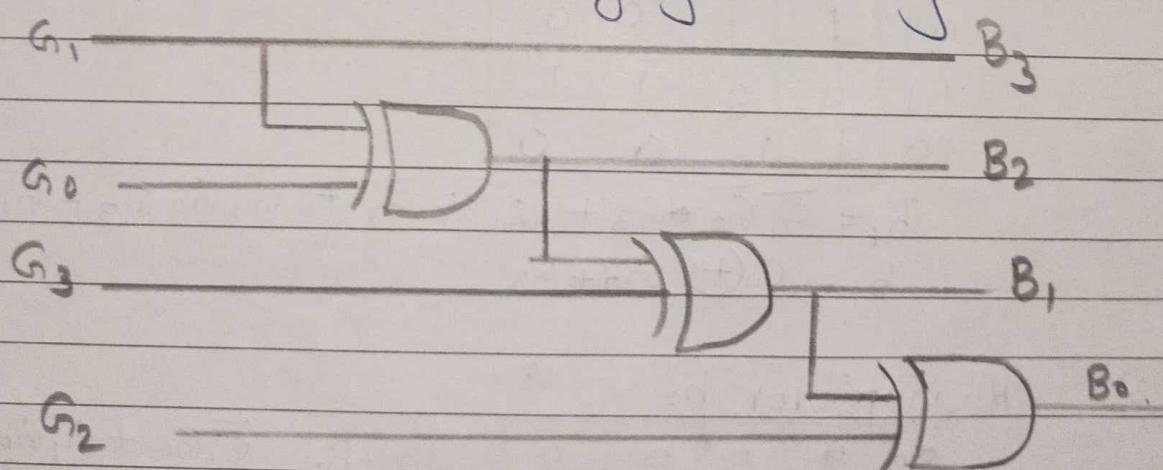
G_1, G_0	G_3, G_2	00	01	11	10
00	00	0	0	0	0
01	00	0	0	0	0
11	11	1	1	1	1
10	11	1	1	1	1

$$B_3 = G_1$$

* Implementation for binary to gray.



* Implementation for gray to binary.



Experiment No: 5

Aim: To realize arithmetic circuits
 i) Half adder
 ii) Full adder iii) Half subtractor iv) Full subtractor.

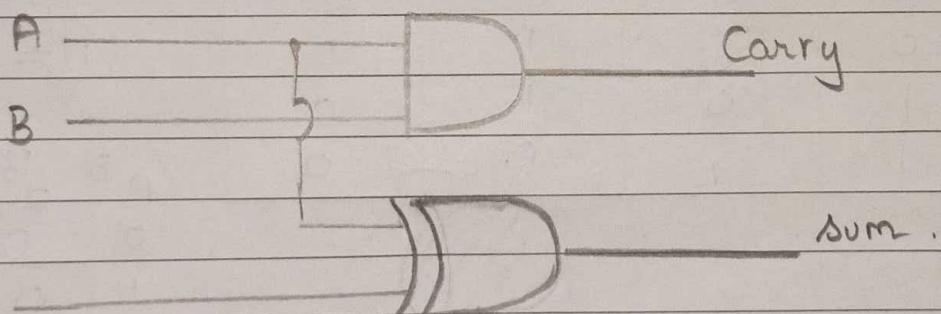
* Truth table of half adder.

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

$$S = A \oplus B$$

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

Circuit diagram of half adder. $C = A \cdot B$



* Truth Table of full adder.

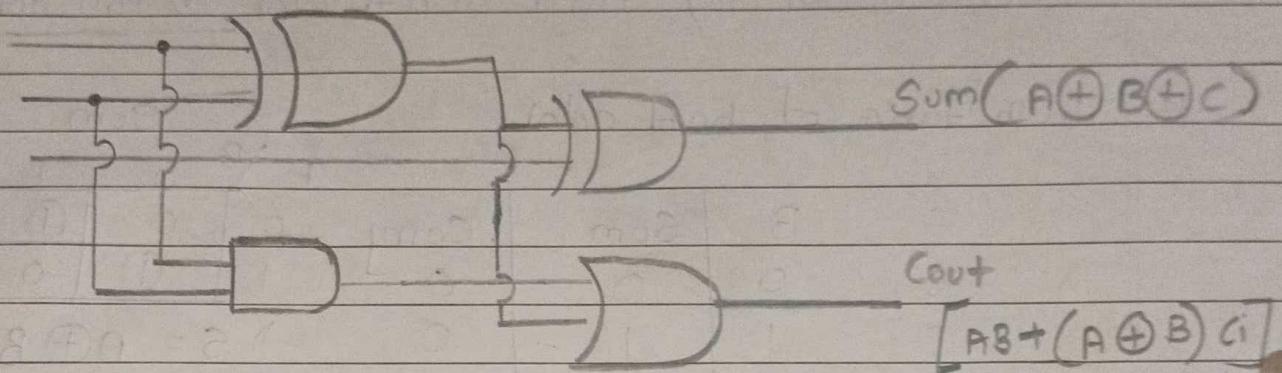
			Sum	
A	B	Cin	Sum	Carryout
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1

$$\text{Sum} = A \oplus B \oplus \text{Cin}$$

			Sum	
A	B	Cin	Sum	Carryout
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

$$\therefore \text{Carryout} = \text{BCin} + \text{ACin} + AB$$

* Circuit diagram of full adder.



* Truth Table of half subtractor.

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

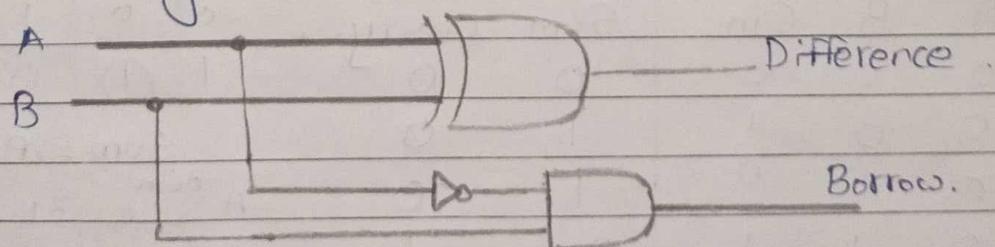
DIFF	
A	B
0	0
1	0

$D = A \oplus B$

Borrow	
A	B
0	0
1	0

$$\text{Borrow} = \bar{A}B$$

* Circuit diagram of half subtractor.



* Truth Table of full subtractor.

A	B	B_i	D	B_o
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

* Logic simplification.

Bout:

A	B_i			
	00	01	11	10
0	0	1	1	1
1	0	0	1	0

$$\therefore B_o = \bar{A}B + \bar{A}B_{int} + B\bar{B}_{int}$$

Diff:

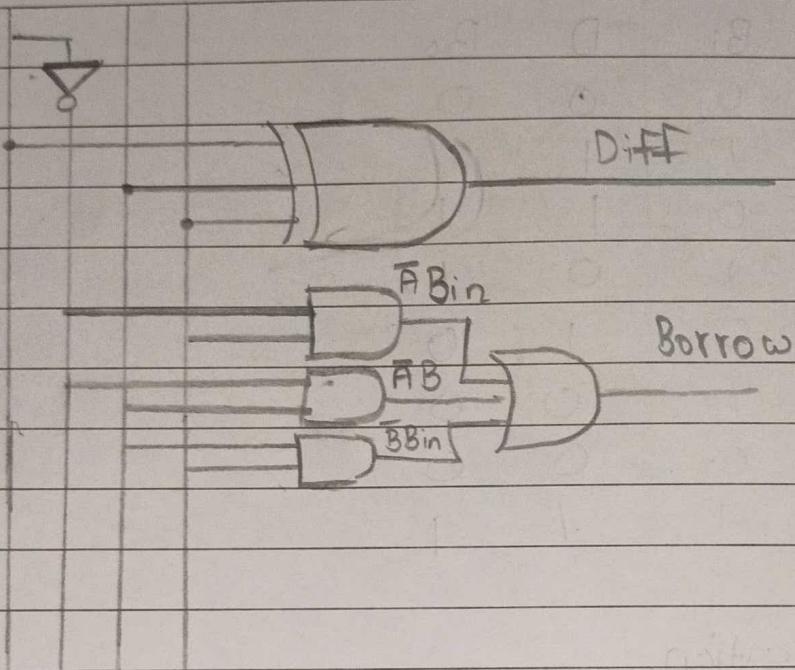
A	B_i			
	00	01	11	10
0	0	1	0	1
1	1	0	1	0

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

*

Circuit diagram of full subtractor.

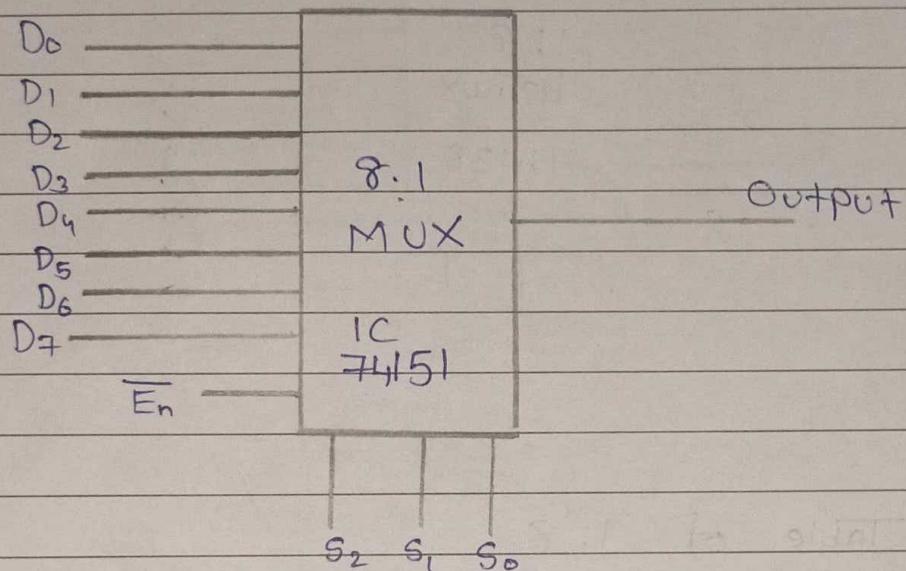
A B Bin



Experiment - 6.

Aim: To study multiplexer and De-multiplexer IC.
Realization of 16:1 mux using 8:1 mux.

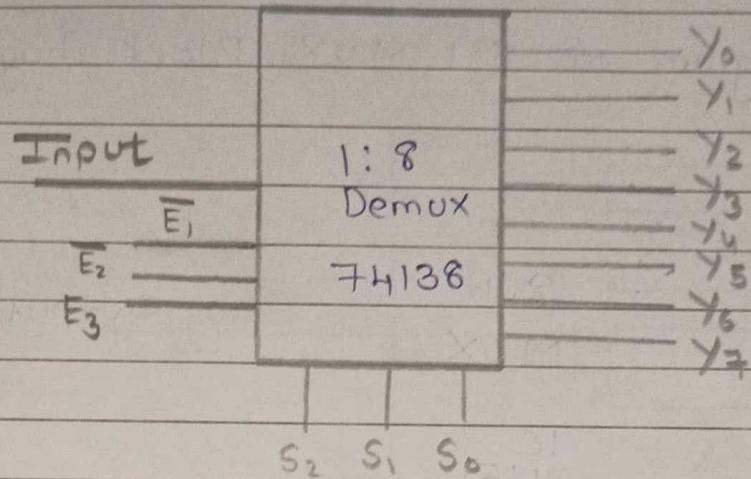
* Circuit Diagram of 8:1 MUX Block diagram.



* Truth Table of 8:1 MUX

\overline{EN}	S_0	S_1	S_2	Output
1	X	X	X	X
0	0	0	0	D_0
0	0	0	1	D_1
0	0	1	0	D_2
0	0	1	1	D_3
0	1	0	0	D_4
0	1	0	1	D_5
0	1	1	0	D_6
0	1	1	1	D_7

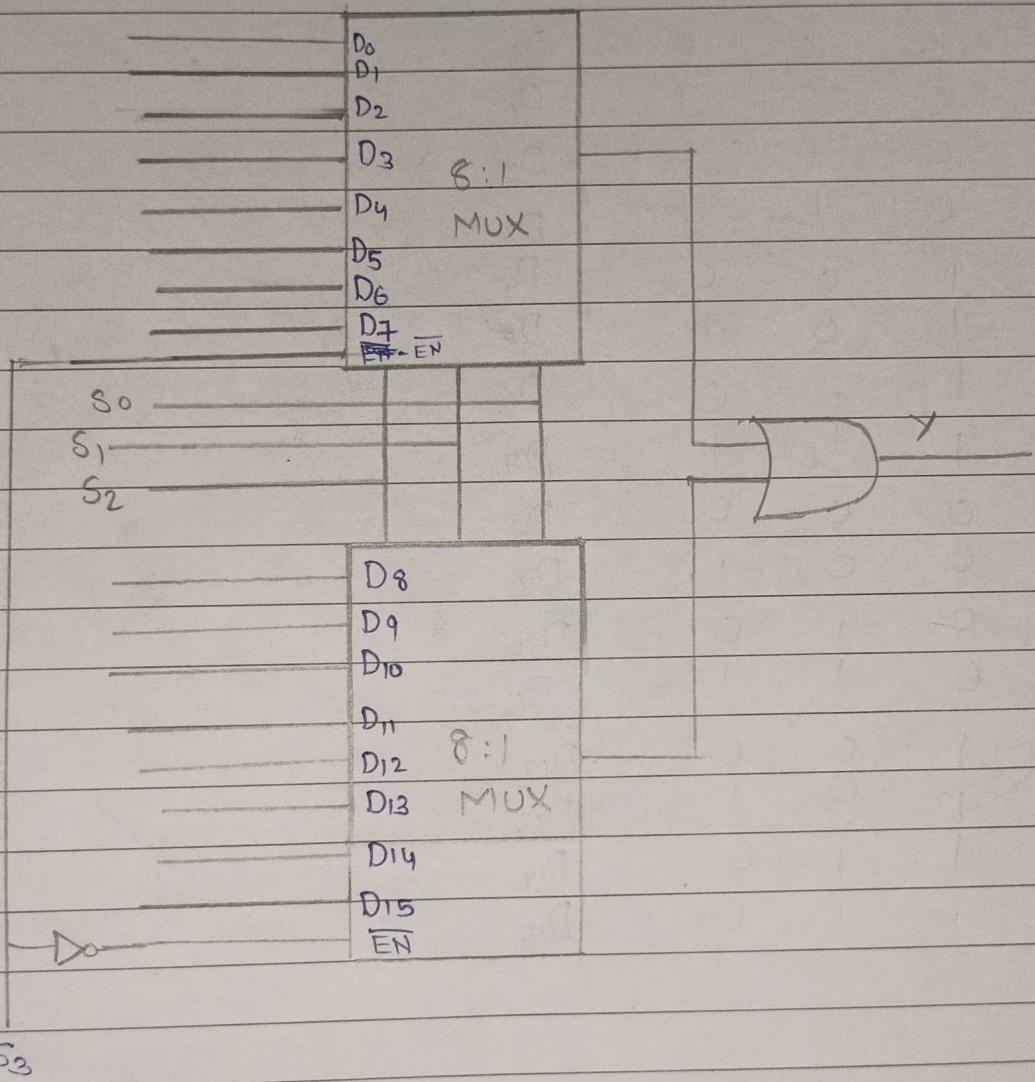
* Circuit diagram of 1:8 demux.



* Truth Table of 1:8

$\overline{E_1}$	$\overline{E_2}$	E_3	S_0	S_1	S_2	y_0	y_1	y_2	y_3	y_4	y_5	y_6	y_7
1	X	X	X	X	0X	1	1	1	1	1	1	1	1
X	1	X	X	X	X	1	1	1	1	1	1	1	1
X	X	0	X	X	X	1	1	1	1	1	1	1	1
0	0	1	0	0	0	0	1	1	1	1	1	1	1
0	0	1	1	0	0	1	0	1	1	1	1	1	1
0	0	1	0	1	0	1	1	0	1	1	1	1	1
0	0	1	1	1	0	1	1	1	0	1	1	1	1
0	0	1	0	0	1	1	1	1	1	0	1	1	1
0	0	1	1	0	1	1	1	1	1	1	0	1	1
0	0	1	0	1	1	1	1	1	1	1	1	0	1
0	0	1	1	1	1	1	1	1	1	1	1	1	0

* Circuit diagram of 16:1 from 8:1



Observation Table

S_3	S_2	S_1	S_0	y
0	0	0	0	D_0
0	0	0	1	D_1
0	0	1	0	D_2
0	0	1	1	D_3
0	1	0	0	D_4
0	1	0	1	D_5
0	1	1	0	D_6
0	1	1	1	D_7
1	0	0	0	D_8
1	0	0	1	D_9
1	0	1	0	D_{10}
1	0	1	1	D_{11}
1	1	0	0	D_{12}
1	1	0	1	D_{13}
1	1	1	0	D_{14}
1	1	1	1	D_{15}

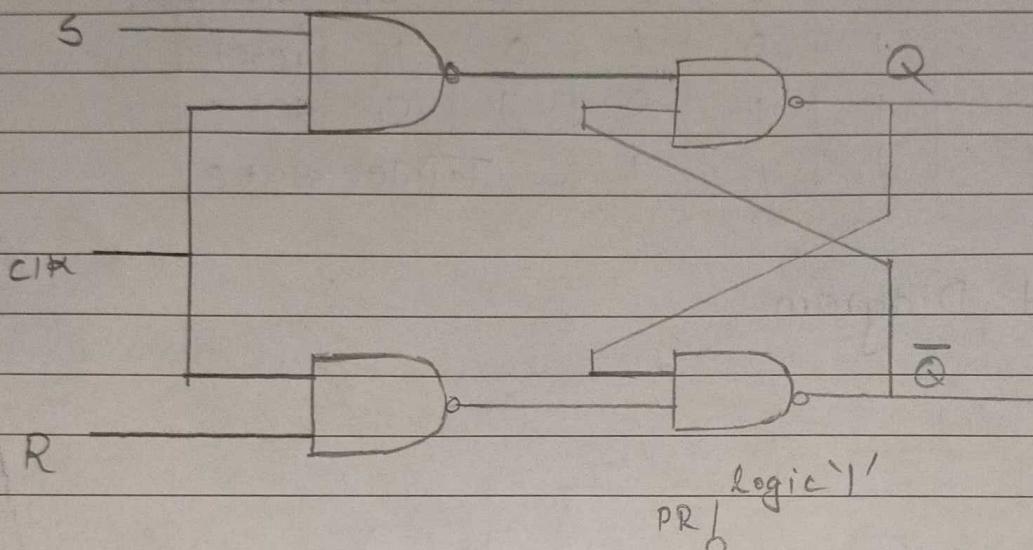
Experiment 7

Aim: Study of Flip Flops using ICs.

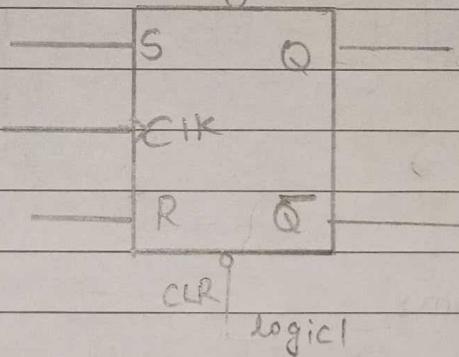
Mihir Vora

6000H220115

* Circuit Diagram of SR flip flop.



* Block Diagram.



* Truth Table.

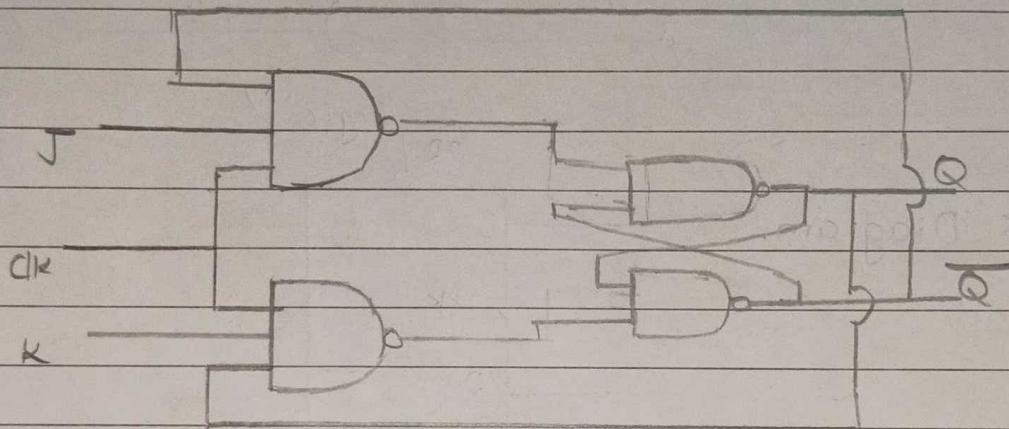
Clock	S	R	Q_{n+1}	State
0	X	X	Q_n	X
1	0	0	Q_n	No change
1	0	1	0	Reset
1	1	0	1	Set
1	1	1	X	Invalid state.

So, when clk is high and if 'R' is on then state will reset & if if 'S' is on then Q will be 1. but if set & reset are high then it will cause an invalid state.

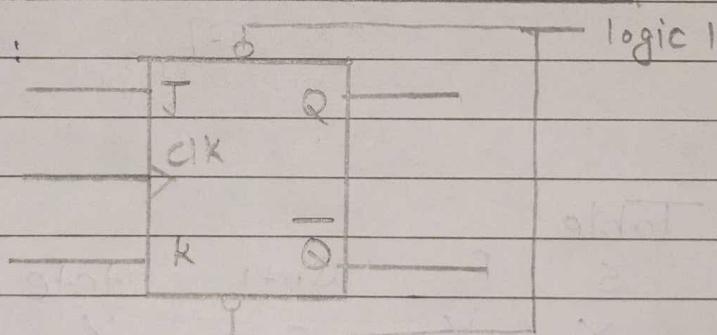
* Truth Table of JK flip-flop.

clk	J	K	Q_{n+1}	\bar{Q}_{n+1}	State
1	0	0	Q_n	\bar{Q}_n	No change
1	0	1	0	1	reset
1	1	0	1	0	set
1	1	1			Toggle state

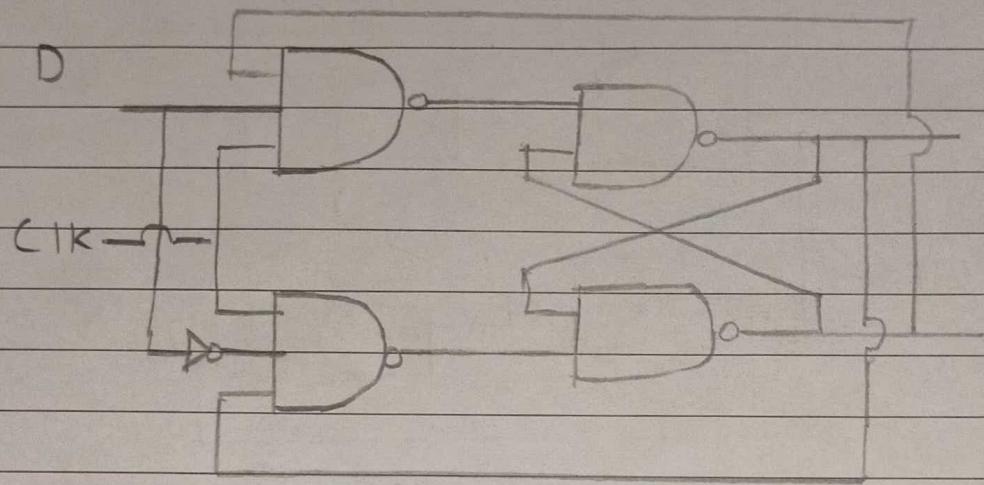
* Circuit Diagram



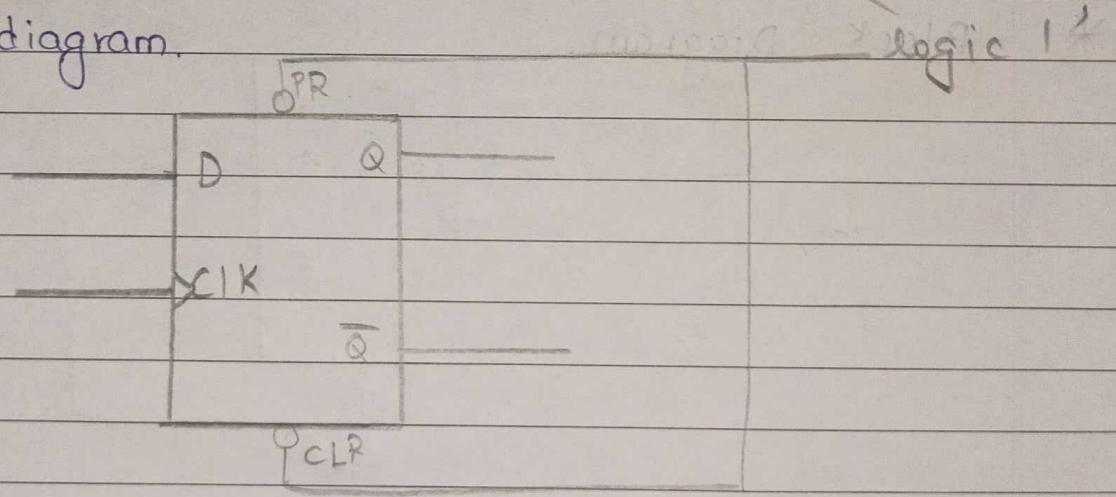
* Block Diagram:



* Circuit Diagram of D flip flop.



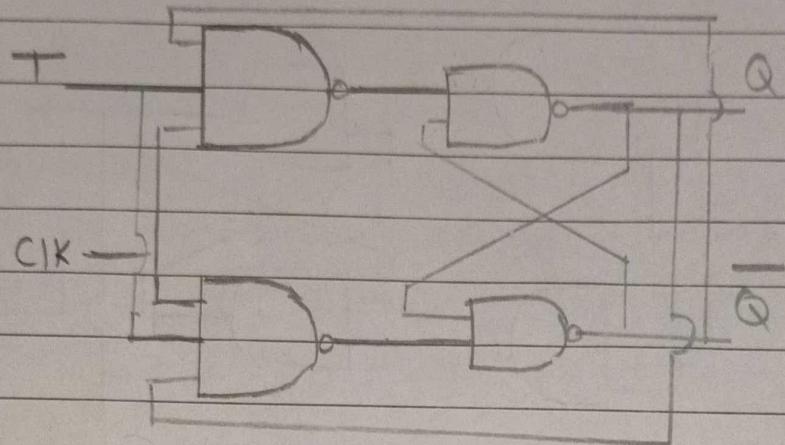
* Block diagram.



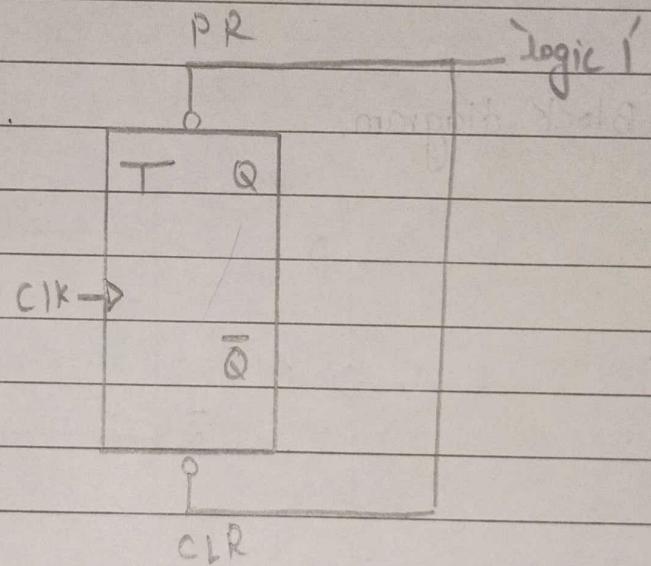
* Truth Table of D Flip Flop.

C1K	D	Q_{n+1}
0	X	X
1	0	0
1	1	1

* Circuit Diagram of T-flip flop.



* Block Diagram.



* Truth Table of T-flip flop.

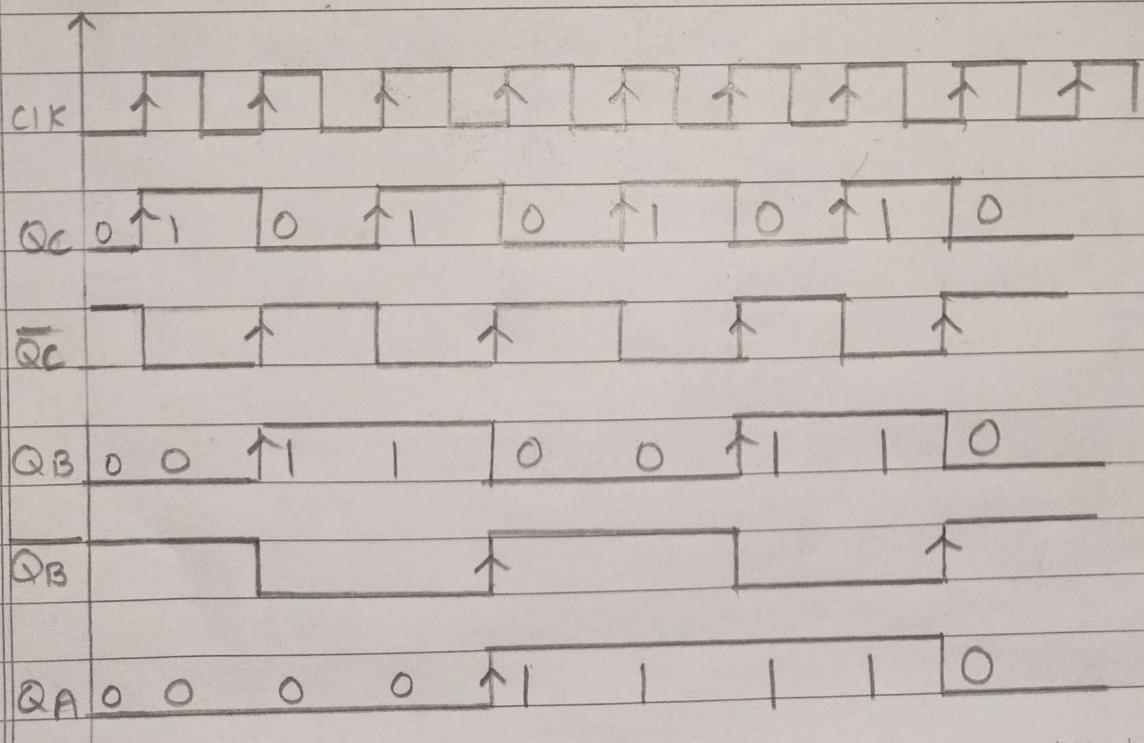
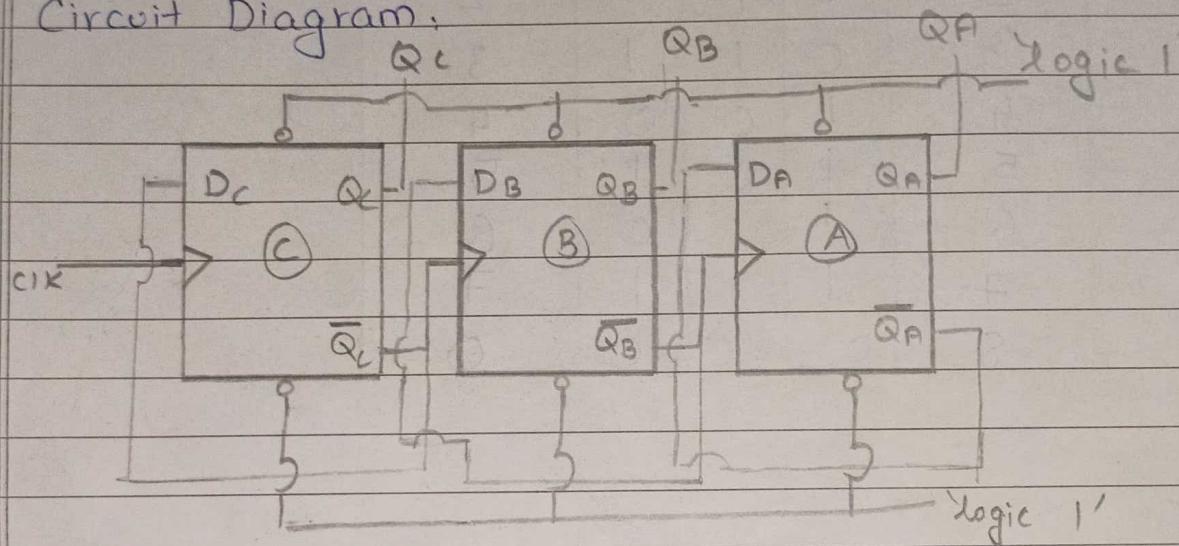
C1K	T	Q_{n+1}	\bar{Q}_{n+1}
0	X	X	X
1	0	Q_n	\bar{Q}_n
1	1	\bar{Q}_n	Q_n

Experiment 8

Aim: To realize asynchronous 3 bit up counter.

3^{bit} up counter (asynchronous) using D-Flip Flop.
rising edge

Circuit Diagram:



* Observation Table.

Count	QA	QB	QC	Decimal O/P
0	0	0	0	0
1	0	0	1	1
2	0	1	0	2
3	0	1	1	3
4	1	0	0	4
5	1	0	1	5
6	1	1	0	6
7	1	1	1	7
8	0	0	0	0