Roll No

INDEX

Sr. No	Experiment Description	Experiment Date	Submission Date	Remarks/Signature
02.	IMPLEMENT AND REALIZE BOOLEAN EXPRESSION WITH LOGIC GATE.			
01.	TO VERIFY THE TRUTH TABLES FOR LOGIC GATES - NOT, OR, AND, NAND, NOR, XOR, XNOR USING CMOS LOGIC GATES AND TTL LOGIC GATES.			
03.	IMPLEMENT HALF ADDER, FULL ADDER, HALF SUBTRACTOR AND FULL SUBRACTOR USING ICS.			
04.	DESIGN AND DEVLOP A MULTIPLEXER AND DE-MULTIPLEXER USING MULTIPIER ICS.			

INDEX

Experiment no-01

★ **Aim :-**

To verify the truth tables for all logic gates - NOT, OR, AND, NAND, NOR, XOR, XNOR using CMOS logic gates and TTL logic gates.

★ **Apparatus :-**

All the basic gates mentioned above.

★ **Theorey :-**

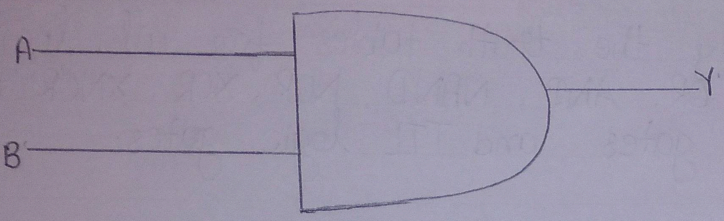
AND, OR, and NOT are basic gates. ~~XOR~~ NAND and NOR are universal gates. basically logic gates are electronic circuits because they are made up of number of electronic devices and component. Input and output of logic gates can occurs only in two levels. These two level are HIGH and LOW or ON and OFF or simply 1 AND 0. A Table which lists all possible combination of input variables and the corresponding output is called 'truth table'.

- AND gate :-

An AND gate has two or more input but has only one output. The output assumes the logic 1 state only when each input is at logic 1 state and 0 state only when each input is at logic 0 state.

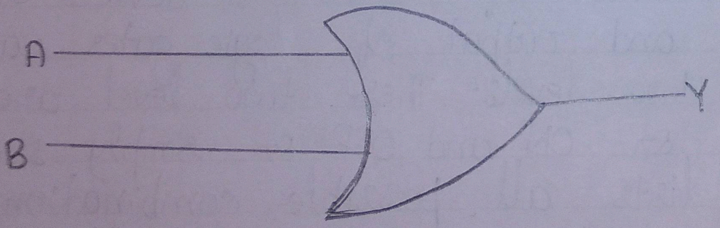
→ $\boxed{Y = A \cdot B}$

AND gate (Ic no→ 7408)



$\boxed{Y = AB}$

OR gate (Ic no.→ 7432)



$\boxed{Y = A+B}$

• OR gate

input b
to be
logic
each

→

• NOT g

has
device
The
its
input

→ Y

• NAN

mea
outp
put assu
of

→ Y

- OR gate :—

An OR gate may have two or more input but only one output. Its output assumes to be 1 state, even if one of its input is in logic 1 state and assumes to be 0 state, when each of inputs is in logic 0.

→ $\boxed{Y = A+B}$

- NOT gate :-

A NOT gate is also known as inverter, has only one input and one output. It is a device whose output is always compliment of input. The output of NOT gate assumes to be 1 when its input is in logic 0 state and 0 when its input is in logic 1 state.

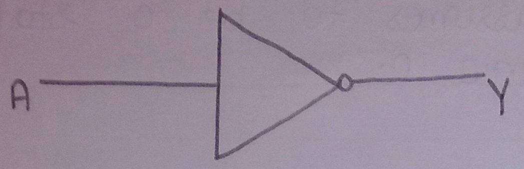
→ $\boxed{Y = \bar{A}}$

- NAND gate :-

NAND gate is universal gate. NAND means NOTAND i.e. AND output is NOTed. The output is at 0 level, only when each of its nput assume a logic 1 level. For any other combination of inputs, the output is logic 1 level.

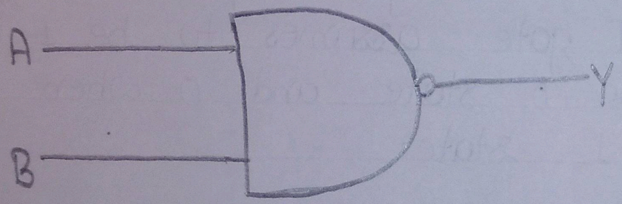
→ $\boxed{Y = \overline{A \cdot B}}$

NOT Gate (Ic no-7404)



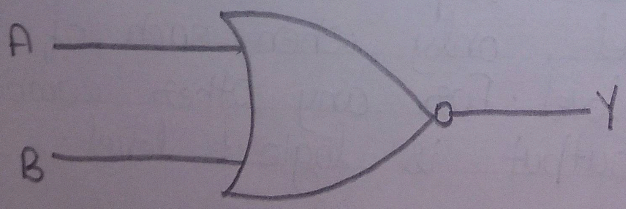
$\boxed{Y = \bar{A}}$

NAND gate (Ic no-7400)



$\boxed{Y = \overline{A \cdot B}}$

NOR gate (Ic no-7402)



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

- NOR gate :-

NOR gate is a universal gate. NOR means NOT OR i.e. OR output is NOTed. The output is logic 1 level, only when each of its input assumes a logic 0 level. For any other combination of inputs, the output is 0 level.

→ $\boxed{X = \overline{A+B}}$

- XOR gate :-

An X-OR gate is a two input, one output logic circuit, whose output assumes a logic 1 state when one and only one of its two inputs assume a logic 1 level. under the condition when both inputs are same either 0 or 1, the output assumes logic 0 state.

→ $\boxed{X = A \oplus B}$

- Ex-NOR gate :-

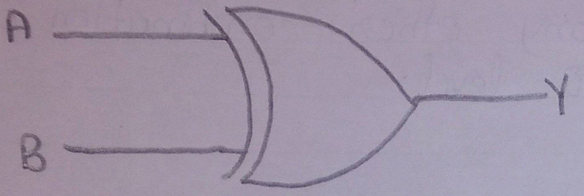
The output of XNOR gate is 1 ~~either~~ if both inputs are same either 0 or 1.

★ Procedure :-

1.) connect the supply (+5v) to the circuit.

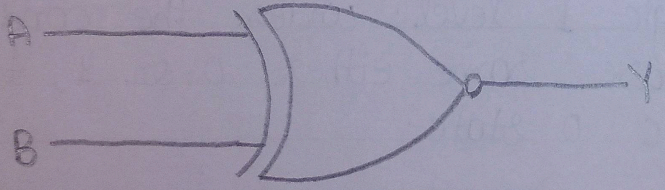
2.) Press switches for inputs "A" and "B".

XOR gate (Ic no- 74136)



$$\boxed{Y = A \oplus B = A'B + AB'}$$

XNOR gate (Ic no-   )



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

3.)

- For AND gate → The bulb does not glow if any one or both the switches are OFF and glows only if both the switches are ON.

- For OR gate → glows only if any one or both the switches are ON else it won't glow.

- For NOT gate → The bulb glows if switch for input B is OFF.

- For NAND gate → bulb glows if any one or both the switches are OFF else it won't glow.

- For NOR gate → bulb glows if both the switches are OFF else it won't glow.

- For Ex-OR gate → bulb glows if one of the switches ~~or~~ is ON and one of the switches is OFF.

- For Ex-NOR gate → bulb glows if both switches are ON or if both the switches are OFF.

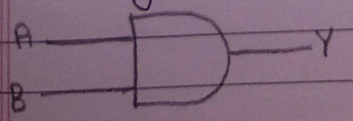
4.) Repeat step-2 and step-3 for all state inputs.

★ Observation:-

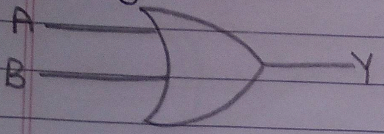
Truth table

A	B	Y=A·B
0	0	0
0	1	0
1	0	0
1	1	1

- AND gate



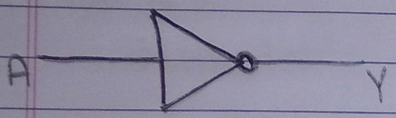
- OR gate



Truth table

A	B	$Y = A+B$
0	0	0
0	1	1
1	0	1
1	1	1

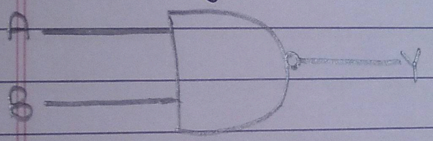
- NOT gate



Truth table

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

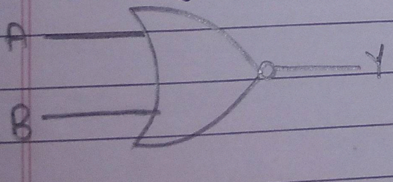
- NAND gate



Truth table

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

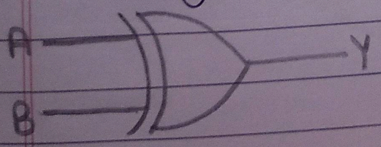
- NOR gate



Truth table

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

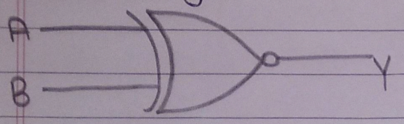
- Ex-OR gate



Truth table

A	B	$Y = A \oplus B$
0	0	0
0	1	1
1	0	1
1	1	0

- Ex-NOR gate



Truth table

A	B	$Y = A \odot B$
0	0	1
0	1	0
1	0	0
1	1	1

★ Result :-

Truth table has been verified.

* ——— * ——— *

Experiment no-02

★ **Aim :-**

Implement and realise boolean expression with logic gate.

★ **Learning objective :-**

- To simply the boolean expression and to build the logic circuit.
- Given a truth table to derive the boolean expression and build the logic circuit to realize it.

★ **Component required :-**

Ic 7400, IC 7408, IC 7432, IC 7406, IC 7402, connecting wires, Bread board

★ **Theory :-**

Canonical forms, Any boolean function can be written in disfunctivenormal form (Sum or Min- term) or confunctive normal form (product of Min- term).

A boolean function can be represented by Karnough map in which each cell corresponds to a midterm. The cells are arranged in such a way that any two immediately adjacent cells corresponds to two min terms of distance. There is more than one way to construct a map with this property.

Karnough maps

For a function of two variable say f(x,y)

	$\bar{x}$	x
y	f(0,0)	f(1,0)
$\bar{y}$	f(0,1)	f(1,1)

For a function of three variable say f(x,y,z)

	$\bar{x}\bar{y}$	$\bar{x}y$	xy	$x\bar{y}$
$\bar{z}$	f(0,0,0)	f(0,1,0)	f(1,1,0)	f(1,0,0)
z	f(0,0,1)	f(0,1,1)	f(1,1,1)	f(1,0,1)

For a function of four variable say f(w,x,y,z):-

	$\bar{w}\bar{x}$	$\bar{w}\cdot x$	wx	$w\bar{x}$
$\bar{y}\bar{z}$	f(0,0,0,0)	f(0,1,0,0)	f(1,1,0,0)	f(1,0,0,0)
$\bar{y}z$	f(0,0,0,1)	f(0,1,0,1)	f(1,1,0,1)	f(1,0,0,1)
yz	f(0,0,1,1)	f(0,1,1,1)	f(1,1,1,1)	f(1,0,1,1)
$y\bar{z}$	f(0,0,1,0)	f(0,1,1,0)	f(1,1,1,0)	f(1,0,10)

Realisation of Boolean expression

CD \ AB	$\bar{A}\bar{B}$	$\bar{A}B$	AB	$A\bar{B}$
$\bar{C}\bar{D}$				1
$\bar{C}D$				1
CD				1
$C\bar{D}$	1	1	1	1

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

After simplify
$Y=A\bar{B}$

- ~~Realisation~~

- For the gi
using basic

Inputs

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

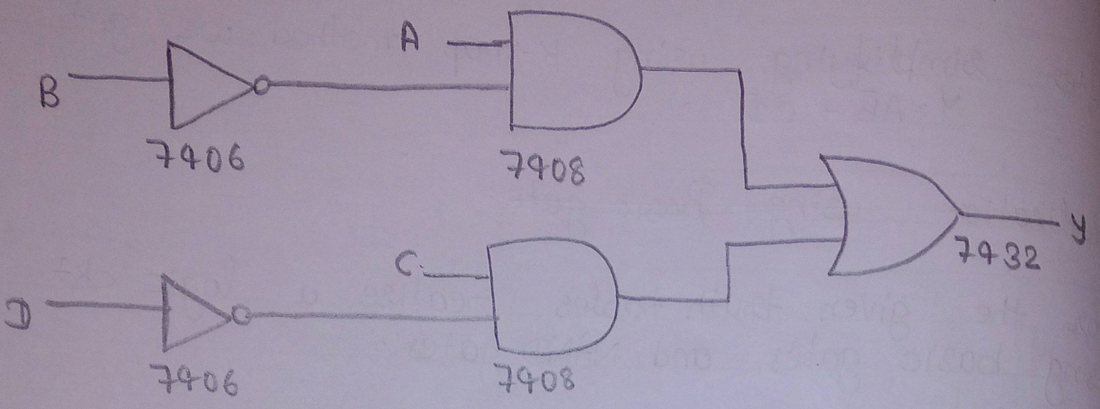
After simplifying using K-map method we get
$Y = A\bar{B} + C\bar{D}$

- ~~Realisation using Basic gate~~:

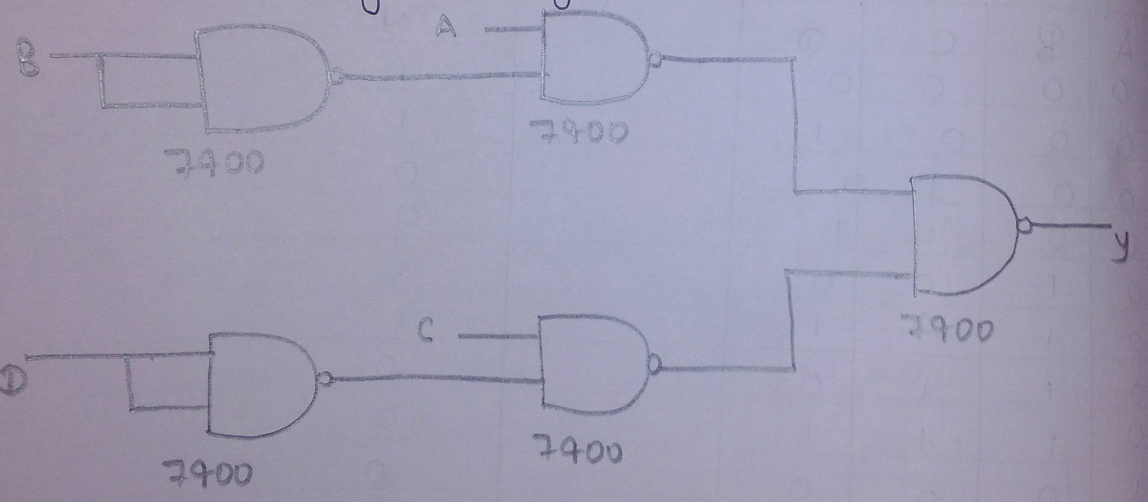
- For the given truth tables realise a logic ckt using basic gates and NAND gates:-

Inputs				outputs
A	B	C	D	Y
0	0	0	0	1
0	0	0	1	1
0	0	1	0	0
0	0	1	1	0
0	1	0	0	1
0	1	0	1	1
0	1	1	0	0
0	1	1	1	0
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	0
1	1	0	0	0
1	1	0	1	1
1	1	1	0	0
1	1	1	1	1

• Realisation using basic gates :-



• Realisation using NAND gates :-



★ Procedur

(i) check

(ii) Insert

(iii) Make

★ Result

Specifi

★ Procedure :-

(i) check the component for their working.
(ii) Insert the approopirate IC into the IC base.
(iii) Make connection as shown in fig.

★ Result :-

Specified, simplified and veried the boolean function.

* ——— * ——— *

Experiment no-03

★ Aim :- · Implement Half adder, full adder
· Half subtractor, Full subtractor using ICs.

★ Learing objective :-

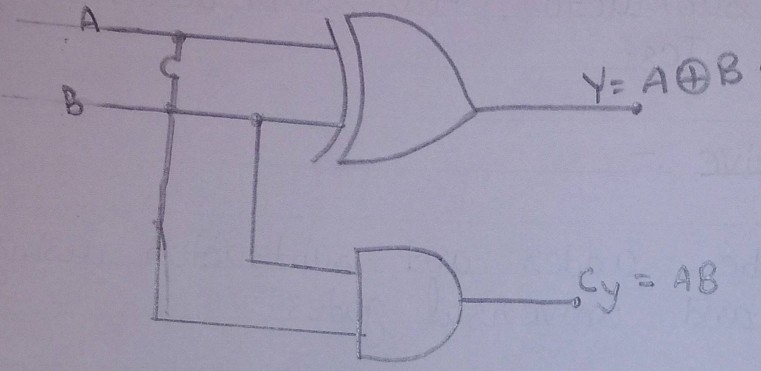
- To realise the adder and subtractor using basic gates and universal gates.
- To realise full adder using two half adder.
- To realise full subtractor using two half subtractor.

★ Component required :-
IC 7400, IC 7408, IC 7486, IC 7432, Patch cards, Ic trainer kit.

★ Theory kit :-

- Half adder :- A combination of logic ckts that performs the addition or two data bits A and B is called half adder. Addiction will result in two output bits one of which is the sum bits and other in the ~~borrow~~ carry bit c. The boolean function describing the half adder are

$$S = A \oplus B$$
$$C = AB$$



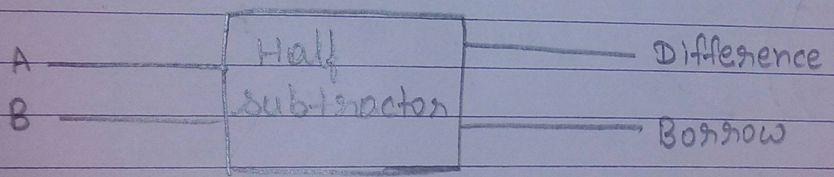
HALF ADDER

- **Adders :-**

Adders are digital circuits that carry out addition of numbers. Adders are a key component of arthimetic logic unit. Aparts from ~~this~~ addition, adders are also used in certain digital application like table index calculation, address decoding, etc.

- **Half Subtractor :-**

The half-subtractor is a combinational circuit which is used to perform subtraction of two bits. It has two inputs. A (minuend) and B (Subtrahend) and two outputs difference and Borrow.



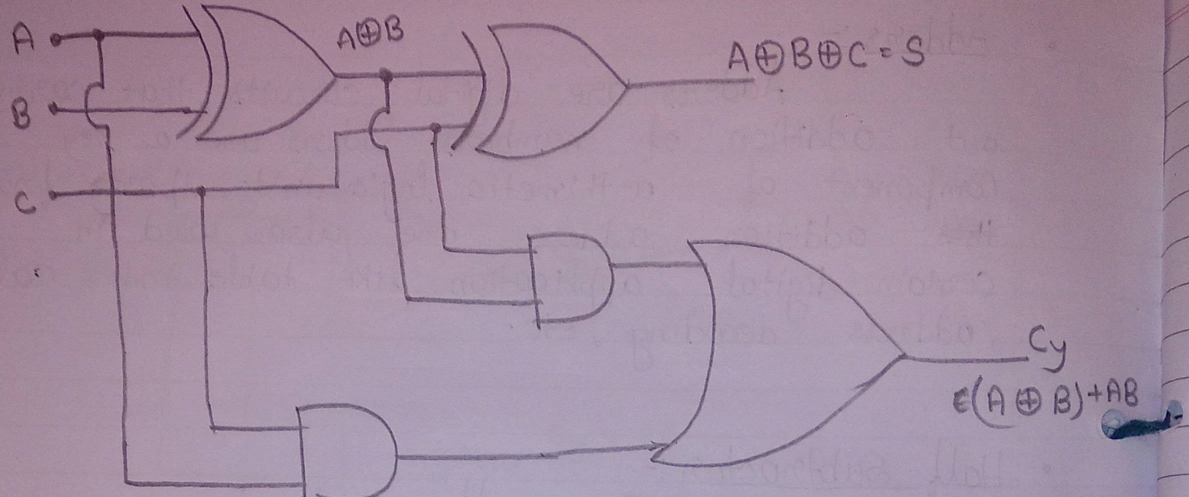
- **Truth table :-**

Inputs		outputs	
A	B	Difference	borrow
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

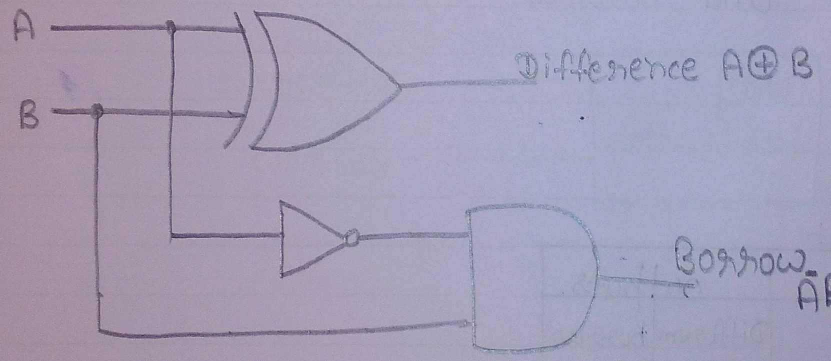
From the above truth table we can find the ~~difference between expr~~ boolean expression :-

$$\text{Difference} = A \oplus B$$

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



FULL ADDER



HALF SUBTRACTOR

- Logic circuit for Half subtractor :-

For realization of a half subtractor we require one XOR, one NOT and one AND gate.

- Full Subtractor :-

A full subtractor is a combinational circuit that performs ~~com~~ subtraction involing three bits, namely A , B and Bin and it produce two output Difference (D) and Bout (Borrow).

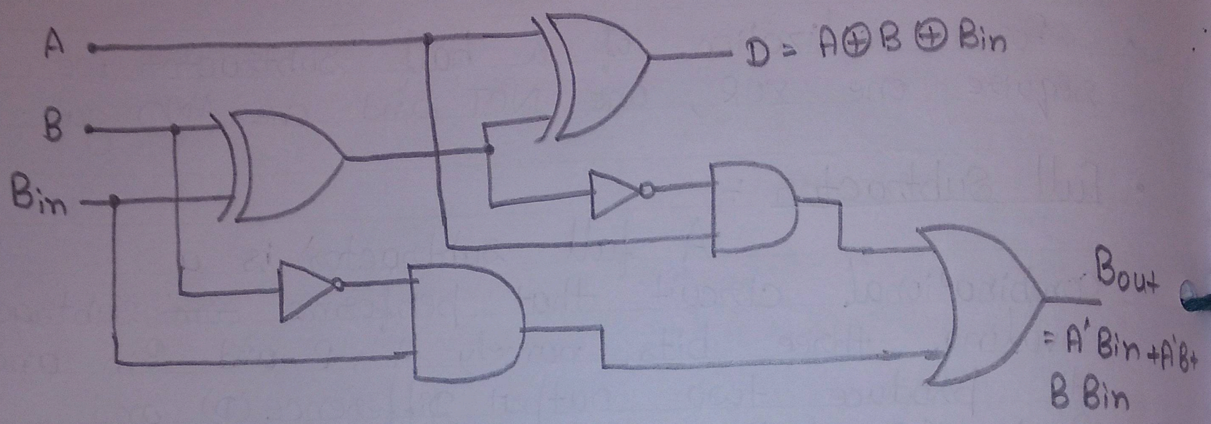
Truth table :-

A	B	B_{in}	D	B_{out}
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

From the truth table we can find boolean expression :-

$$D = A \oplus B \oplus Bin$$

$$B = A' Bin + A' B + B Bin$$



FULL SUBTRACTOR

Logic circuit for full subtractor

For realisation of a full subtractor, we required two ~~ex~~ XOR , two NOT , two AND and one OR gate·

★ Procedure :-

(i) check the component for working·
(ii) Insert Ic into the Ic base·
(iii) verify truth table and observe output·

* ——— * ——— *

Experiment no-04

★ Aim :- Design and devlopment of Multiplexer and De- multiplexer using multiplexer Ics.

★ Theorey :-

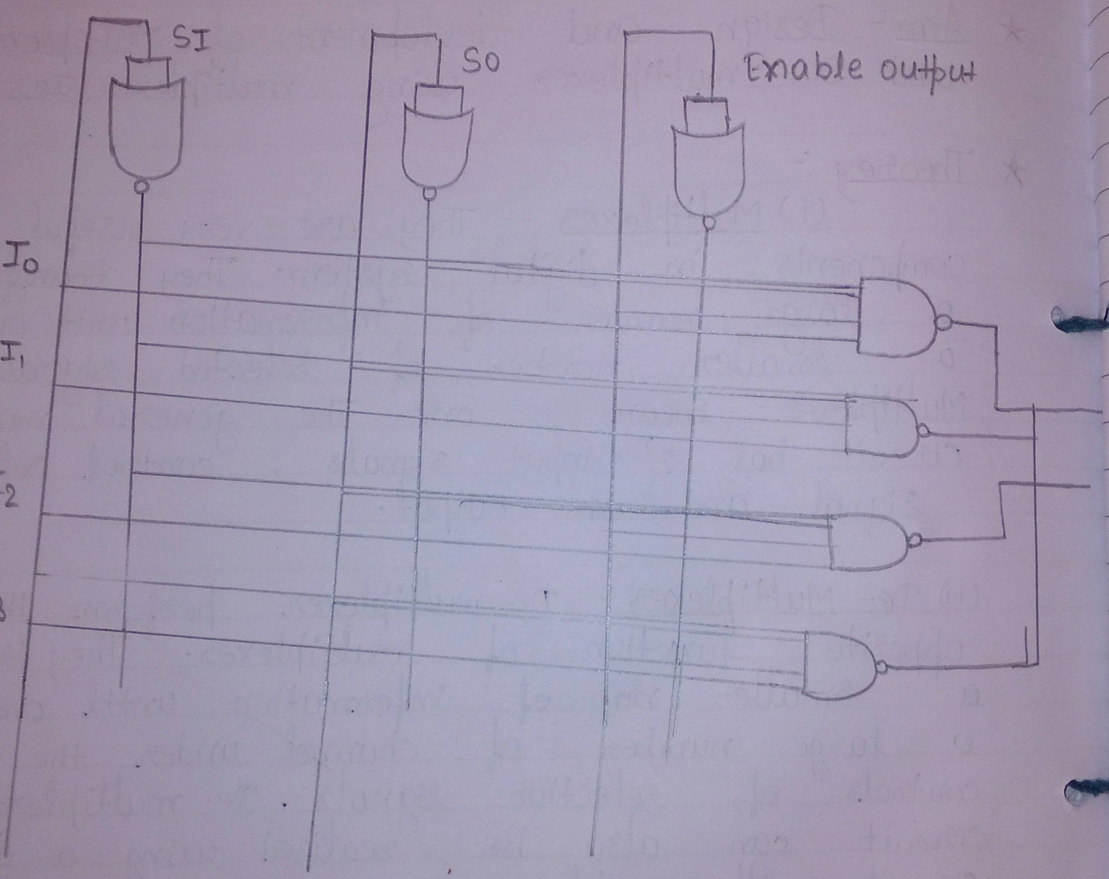
(i) Multiplexer :- They are very useful components in digital system. They transfer a large number of information unit over a smaller number of selected signals. Multiplexer means to one. The general multiplexer circuit has 2^n input signals, control select signal and one output.

(ii) De- Multiplexer :- De-multiplexer perform the opposite function of multiplexer. They transfer a smaller no. of information until over a large number of channel under the controls of selection signals. De- multiplexer circuit can also be realised using a decoder circuit will enable.

★ Equipment :-
IC 7400, IC 7410, IC 7420, IC 7404, IC 74153, IC 74139, patch cards, Ic trainer kit.

~~i). 4:1 M.~~

REALIZATION USING NAND GATE OF MULTIPLEXER



i) 4:1

Inputs

· outp

y

ii) De-

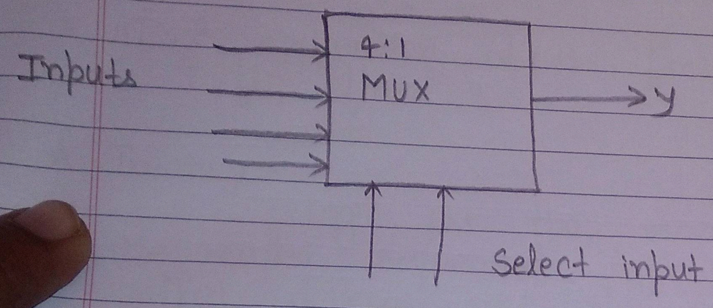
ENABLE INPU

1

0

★ P

i) 4:1 Multiplexer :-



· output of Multiplexer

$$y = E'S/S'0' I_0 + E'SI' SOI + E'SISOI2 + E'SI' SOI3$$

ii) De- Multiplexer :-

Realisation using NAND gate

ENABLE INPUT	DATA INPUT	SELECT INPUTS		OUTPUTS			
		S_1	S_0	y_3	y_2	y_1	0
1	0	×	×	×	×	×	×
0	1	0	0	0	0	0	1
0	1	0	1	0	0	1	0
0	1	1	0	0	1	0	0
0	1	1	1	1	0	0	0

★ Procedure :-

(i) Check all the component for their working.

(ii) Insert the appropirate Ic into the Ic base.

(iii) Make connection as shown in the circuit.

(iv) verify the truth table and observe the output.

* ———— * ———— *

Experiment no-05

★ AIM :- verification of the function S,R,D, JK and T flip flops

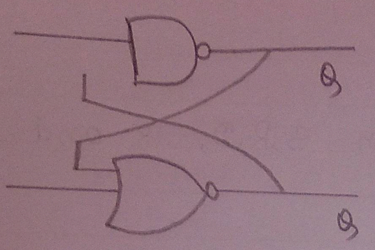
★ Theorey :-

Logic ckt that incorporate memory cells are called sequential logic circuit, their output depends not only upon the present ~~i/p~~ value of the input but also upon the present value ~~of~~ for their operation. The latch flip flop is a basic bi-slable memory element widely used squential logic ckts.

S.R Latch :-

A S.R latch consist of two cross coubled NOR gates. An S.R flip-flop can also be design using cross. coupled NAND gates as shown in figure. The truth table of the ckt are listed below. A clocked S-R flip flop has an additional clock input so the S and R input are active only when the clock is high.

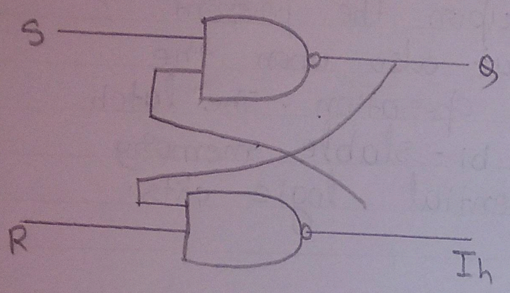
A Patch combine the S and R input of 2nd SR latch into one input by adding an input an investor when the clock is high the state of flip-flop is latched and cannot change until the clock is ~~high latched and c~~ high again.



TRUTH TABLE

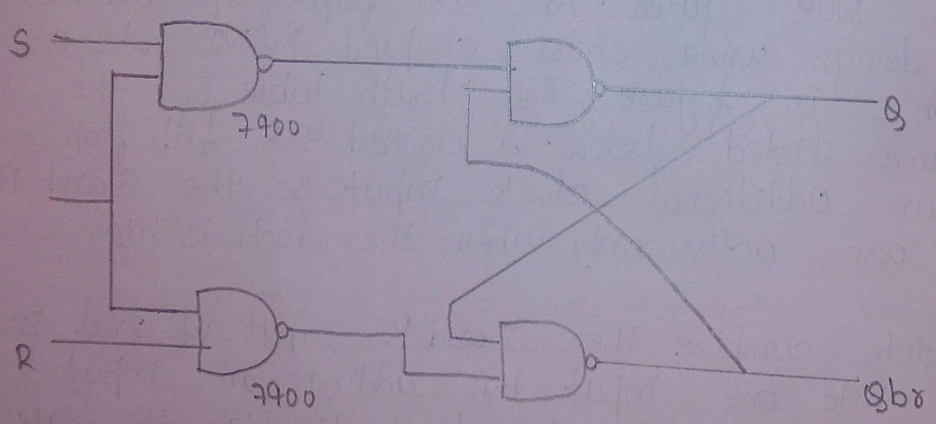
S	R	Q^2	Qh^2
0	0	Q	Qb
0	1	0	1
1	0	1	0
1	1	0	0

1) S.R LATCH



TRUTH TABLE

S	R	Q^2	Qb^2
0	0	1	1
0	1	1	0
1	0	0	1
1	1	Q	Qb



A SR flip flop can be converted T-flip flop by connecting S input to Qb to R to Q.

Truth table SR flip flop :-

S	R	Q+	Qb+
0	0	Q	Qb
0	1	0	1
1	0	1	0
1	1	0	0

Truth Table :-

clock	J	K	Q+	Q+
1	0	0	Q	Q
1	0	1	0	1
1	1	0	1	0
1	1	1	Q	Q

★ Procedure :-

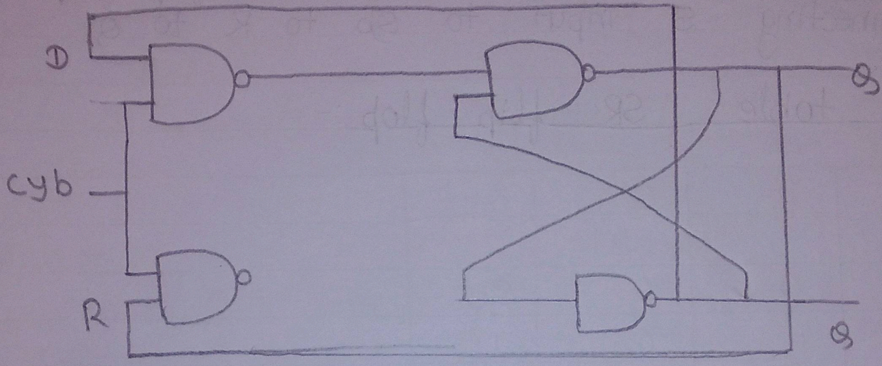
(i) check all the components for their working.

(ii) Insert all appropriate Ic into Ic base.

(iii) Make connection as shown in the ckt diagram.

(iv) verify the truth table and observe the output.

CONVERSION OF SR-FLIP to FLIP-FLOP



INPUT TABLE

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

AIM :- Implement parallel and serial full adder using Ic.

Apparatus Required :-

- Digital Trainer kit
- Ic - 7483 and connecting wires.

Theory:

Adder:-

An adder is a logic circuit which add two or three bits at a time and give sum and carry as the result.

parallel adder:-

A. n-bit parallel bit can be constructed using number of full adder circuit connected in parallel the carry.

output of. each is connected to the carry input of the next higher-order adder, since all of the bits of the aligends and added are feed into the adder circuit symmetric and the additions in each position are known as parallel adder

$A_3 A_2 A_1 A_0$ → Augends bit
$B_3 B_2 B_1 B_0$ → Added bit
$S_3 S_2 S_1 S_0$ → sum bit

serial adder:-

A serial odder is used to add binary numbes in serial form. the two binary number to be added serially are store in two Shift Resistor.

The circuit add one pair at a time with the help of one full adder. The carry output form the full adder is applied to a D-flip-flop, so it called serial adder

procedure:

1 Connect ground and Vcc to 7483 Ic from trainer kit through Patch cords

2. connected ~~output~~ input A_0, A_1, A_2 and A_3 and B_0, B_1, B_2 and B_3 to logic gates switches.

3. connect. carry in form pin no 13 to ground so. that carry input (Cy_i) will be logic 0' state position

4 connect S_0, S_1, S_2 and S_3 carry out (Cy_o) from pin no. 9, 6, 2, 15, 14 to the output

display 5. verify the truth table for diff combination of input

Result :-

for various combinatio of selected input lines observed the LED output and varified the truth table.

Truth table

The truth table operation of 4-bit parallel Adder is shown in fig.

input									output				
A_0	A_1	A_2	A_3	B_0	B_1	B_2	B_3	CY_1	S_0	S_1	S_2	S_3	CY_0
0	0	0	1	0	1	0	1	1	1	1	0	0	1
1	0	0	0	1	1	1	0	6	0	0	0	0	1

Precaution :

1. All IC's should be cheaked before starting the experiment
2. All the connection should be tight
3. Always connect ground first and the connect Vcc.
4. suitable type wire shoulld be used for diff. type of circuit
5. The kit should be off before change of connection
6. After completed the experiment switch off the supply of the AIM.

AIM: Design controlled shift resistor.

LEARNING objective:

- To illustrate the operation of shift resistor
- To study diff shift resister configuration

Component Required:

Ic-7495, Ic-7474, patch cords and Ic-trainer kit.

Theory:

Shift resistor are a type of sequential logic circuit, mainly for storage of digital data.

They are group of flip-flop connected in a chain, so that the output from one flip-flop becomes the input of next flip-flop are derived by a common block, and all the set or reset similary

The serial in/out shift resistor accept data serially - that is one bit at a time

LTIPLEXER
PIER

ons a single line. It produced the stored information on its output also in serial form.

Types of shift resistor:

i> SISO:- In this of resistor, we output of flip-flop is connected to the input of the next flip flop.

output of the resistor is obtained form the last flip-flop

Truth table :

CLK	serial in	serial out
1	1	0
2	0	0
3	0	0
4	1	1
5	x	0
6	x	0
7	x	1

ii> SIPO :-

This is similar to SISO Expect that the output is taken from each flip-flop. Thereby the shifted value as shown at once.

Truth table

CLK	Data	Output Q_A	Q_B	Q_C	Q_D
1	1	1	0	0	0
2	0	0	1	0	0
3	0	0	0	0	1
4	1	1	0	1	1

iii> PIPO :-

upon giving clock pulse data is loaded in parallel in all flip-flop output is taken from each of flip-flop

Truth table

CLK	Data Input				output			
	D_A	D_B	D_C	D_D	Q_A	Q_B	Q_C	Q_D
1	1	0	0	1	1	0	0	1
2	1	0	1	0	1	0	1	0

iv> PISO :-

IC - 7474 consist of two D flip-flop with PRESET and CLEAR The pin diagram.

Truth table

CLK	Q_3	Q_2	Q_1	Q_0	O/P
0	1	0	0	1	1
1	0	0	0	0	0
2	0	0	0	0	0
3	0	0	0	0	1