# EC-273

# Digital Circuit And Systems Lab

# Experiment 1-2

Pranav Mittal(19095076)

Guguloth Yakesh(19095038)

Rahul Ponnala(19095073)

# Experiment 1

Aim:- Verification and interpretation of truth table for AND, OR, NOT, NAND, NOR, Ex-OR, Ex-NOR gates. using RTL (Resistor Transistor Logic), DTL (Diode Transistor Logic) and TTL (Transistor Transistor Logic) logics in simulator 1 and verify the truth table for AND, OR, NOT, NAND, NOR, Ex-OR, Ex-NOR gates in simulator 2.

Theory :-

Logic gates are the basic building blocks of any digital system. Logic gates are electronic circuits having one or more than one input and only one output. The relationship between the input and the output is based on a certain logic. Based on this, logic gates are named as

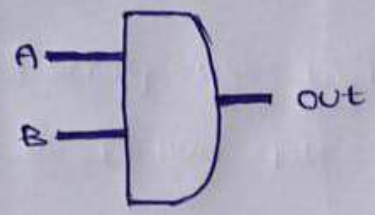
1) AND gate
2) OR gate
3) NOT gate
4) NAND gate
5) NOR gate
6) Ex-OR gate
7) Ex-NOR gate.

## 1) AND gate

The AND gate is an electronic circuit that gives a high output (1) only if all its inputs are high.

i.e, A.B or can be written as AB

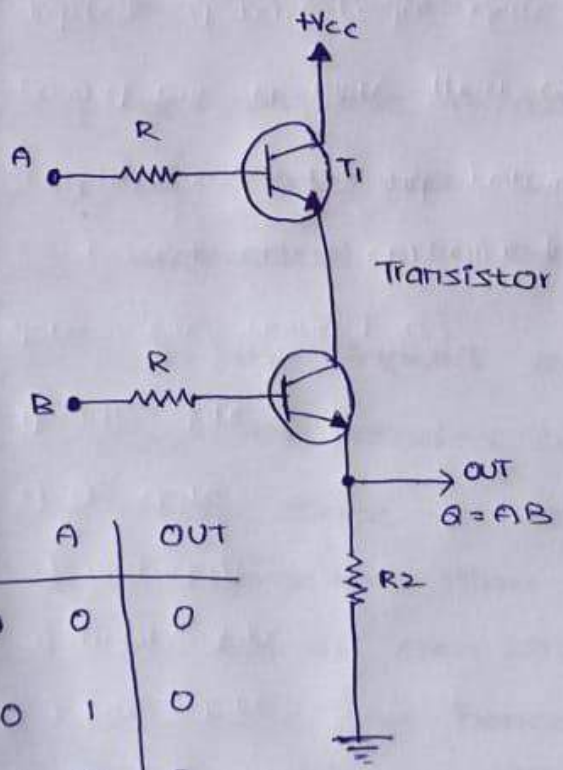
$$Y = A.B$$




AND Gate.

| input | | output |
|---|---|---|
| A | B | Y = A.B |
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

Truth table.



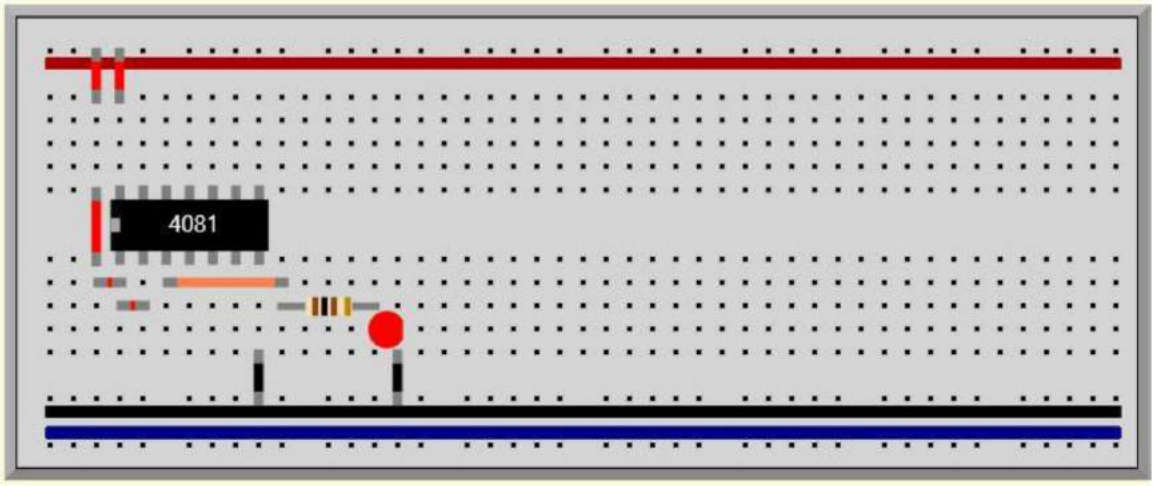

Transistor switches.

| B | A | OUT |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

⇒ AND Gate through RTL logic

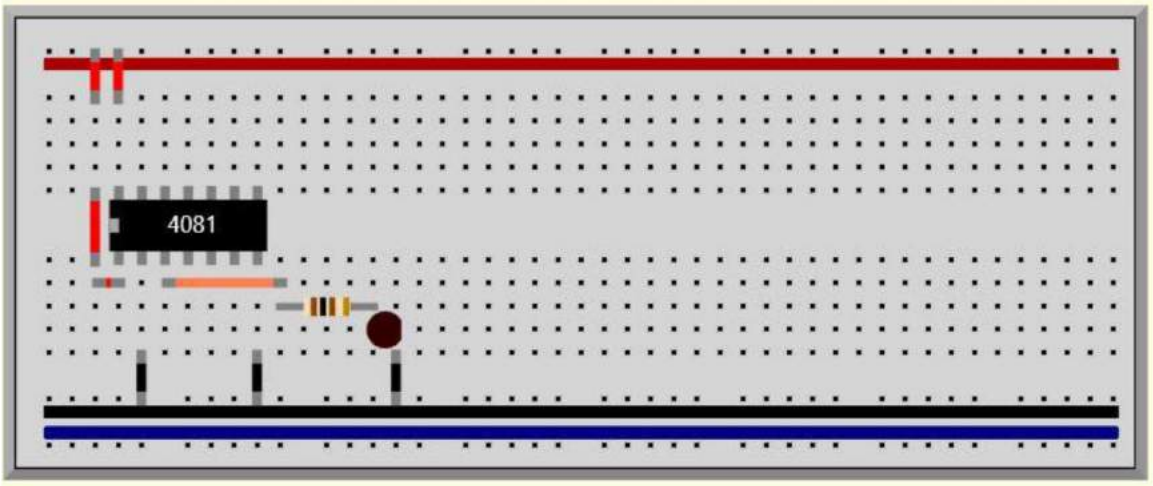
AND:

Input: 1 1, Ouptut:1




AND:

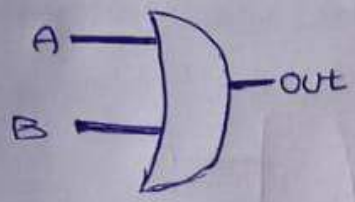
Input: 1 0,Output:0



## 2) OR gate

The OR gate is an electronic circuit that gives a high output (1) if one or more of its inputs are high.

$$Y = A + B$$




OR Gate

| Input | | Output |
|---|---|---|
| A | B | Y = A + B |
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 1 |

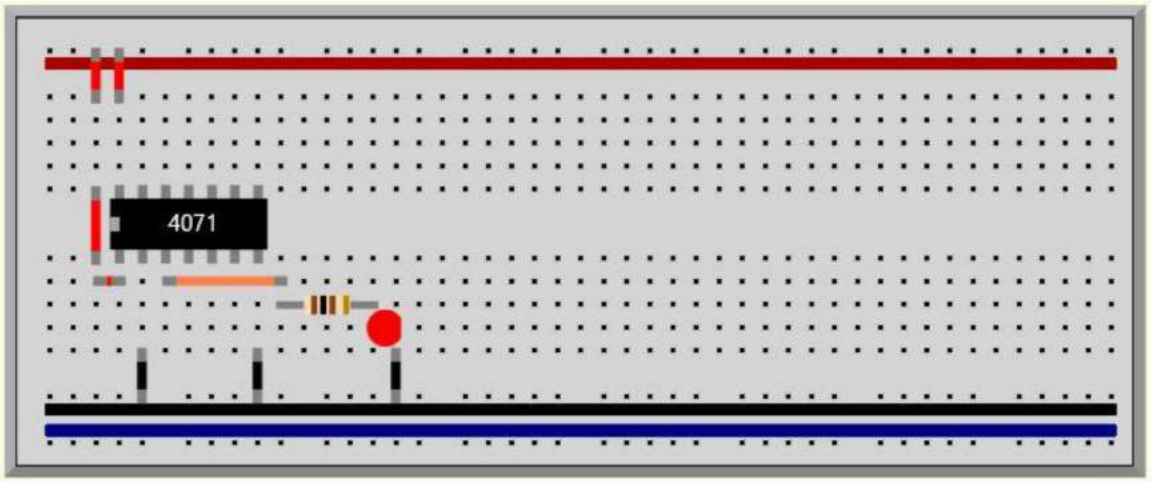
→ OR gate can be realized by DRL (Diode-Resistance Logic) or by TTL (Transistor Transistor logic),

Presently, we will use a diode at every input of the OR gate, the anode part is joined together and a resistor, connected with the cathode is grounde.

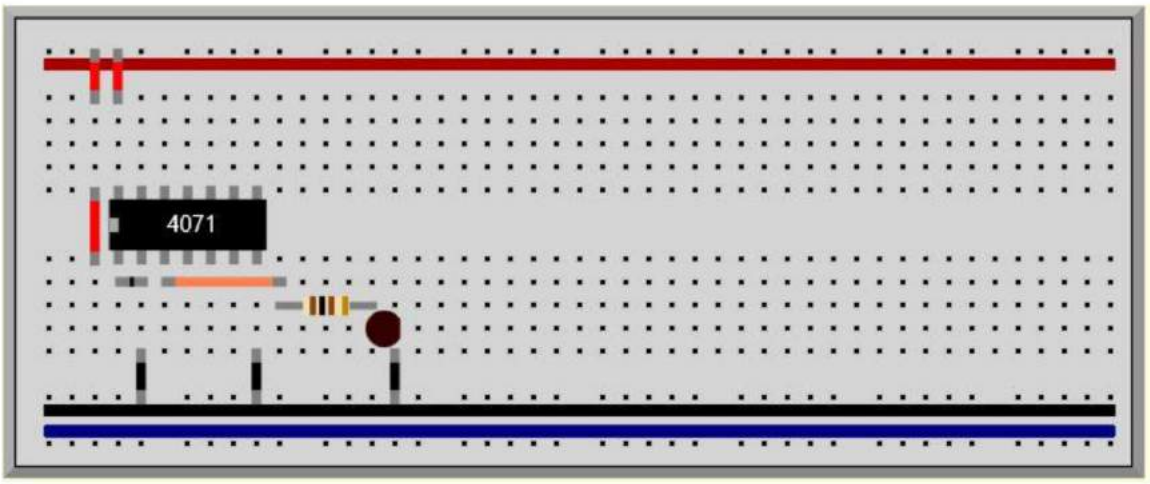
→ when both the inputs are at logic 0 or low state then the diodes D1 and D2 become reverse biased. since the anode terminal of diode is at lower voltage level than the cathode terminal, so diode will act as open circuit so there is no voltage across resistor and hence output voltage is same as ground. when either of the diodes is at logic 1 or high state then the diode corresponding to that input is forward bias.

OR

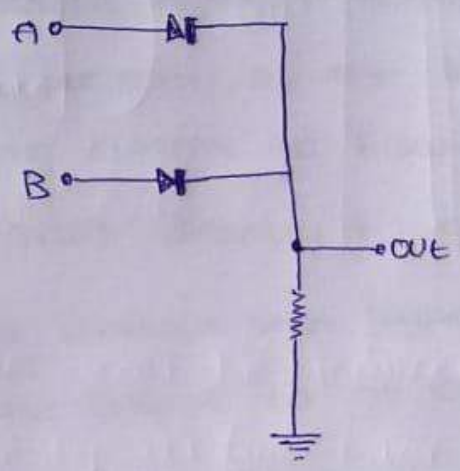
Input:1 0 , Output:1




Input:0 0,Output:0



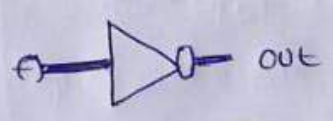
Since this time anode is at high voltage than cathode, therefore current will flow through forward biased diode and this current then appears on resistor causing high voltage at output terminal also. Hence at output we get high or logic 1 or +5V. So, if any or both inputs are high, the output will be high or "1"




OR Gate through DRL logic

NOT gate :-

→ The NOT gate is an electronic circuit that produces an inverted version of the input at its output. If the input variable is A, the inverted output is known as NOT A. This is also shown as A' or A with a bar over the top.




NOT Gate

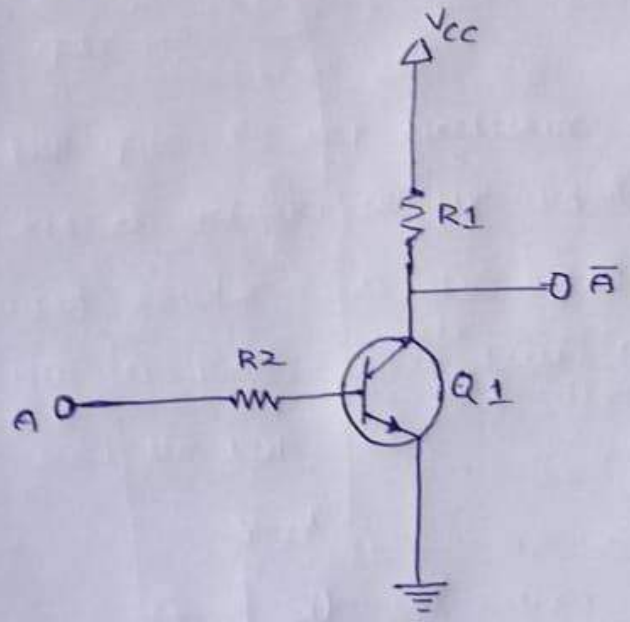
| Input | Output |
|---|---|
| A | Y |
| 0 | 1 |
| 1 | 0 |

Truth table

→ NOT gate can be realised through transistor. The input is connected through resistor R2 to the transistor's base. When no voltage is present on the input, the transistor turns off. When the transistor is off, no current flows through the collector-emitter path.

Thus, current from the supply voltage ($V_{cc}$) flows through resistor $R_1$ to the output. In this way, the circuit's output is high when its input is low. When voltage is present at the input, the transistor turns on, allowing current to flow through the collector-emitter circuit directly to ground. This ground path creates a shortcut that bypasses the output, which causes the output to go low.

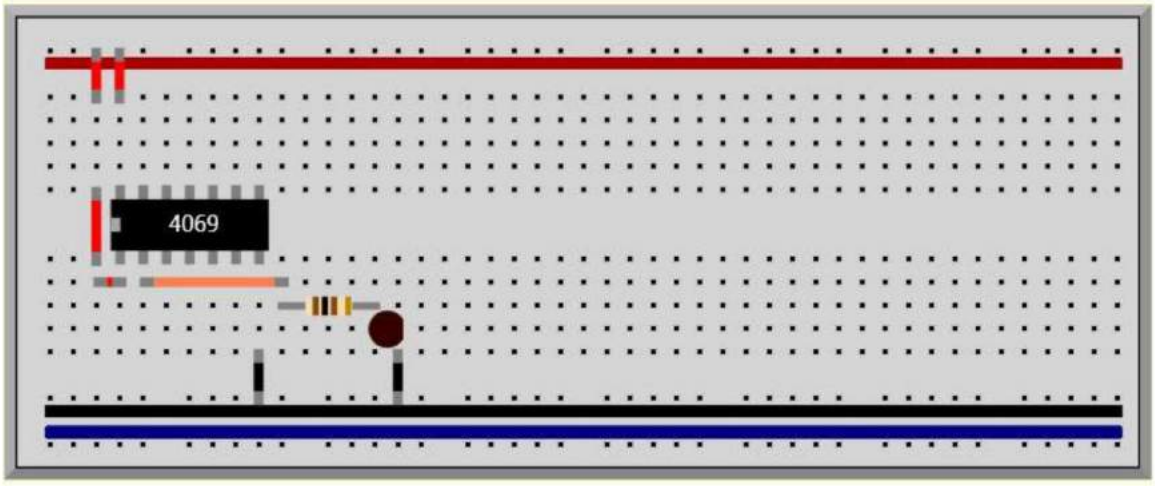
In this way, the output is high when the input is low and low when the input is high.



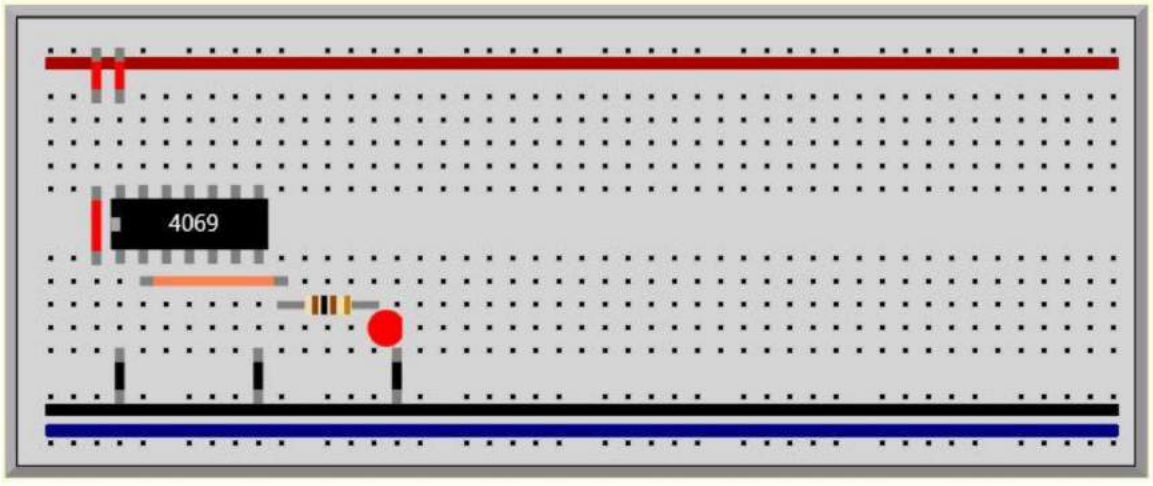

NOT Gate through Transistor.

NOT

Input: 1, Output: 0



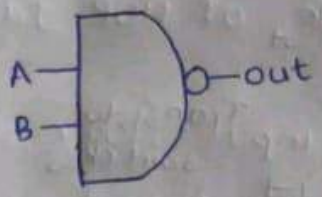

Input: 0, Output: 1



4) NAND gate:

This is NOT-AND gate which is equal to an AND gate followed by a NOT gate. The outputs of all NAND gates are high if any of the inputs are low. The symbol is an AND gate with a small circle on the output. The small circle on the onput. The small circle represents inversion.

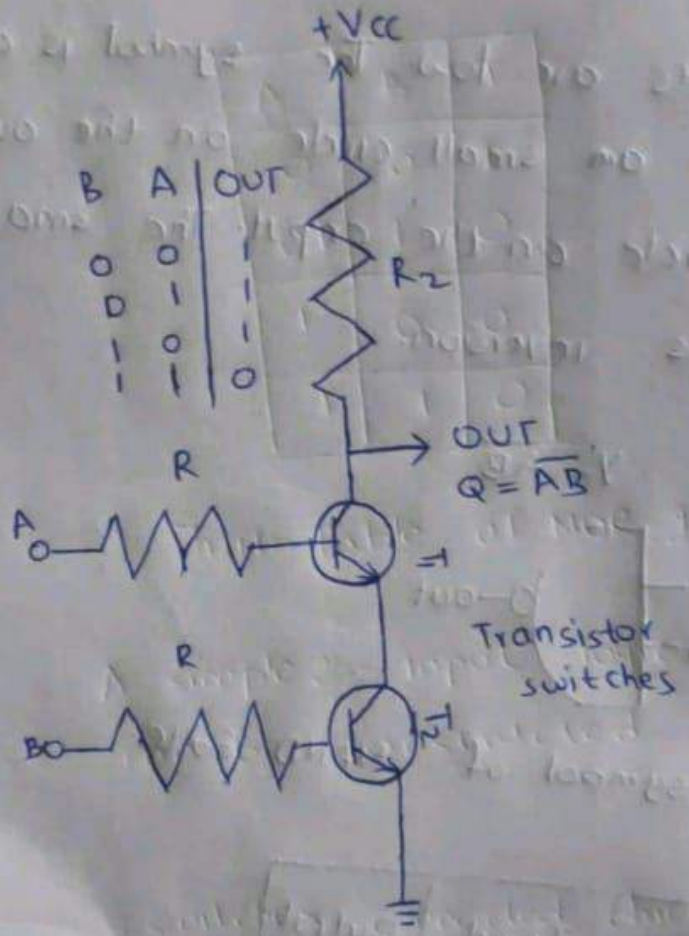
$Y = \overline{AB}$




symbol of NAND Gate.

| Input A | Input B | Output Y |
|---|---|---|
| 0 | 0 | 1 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

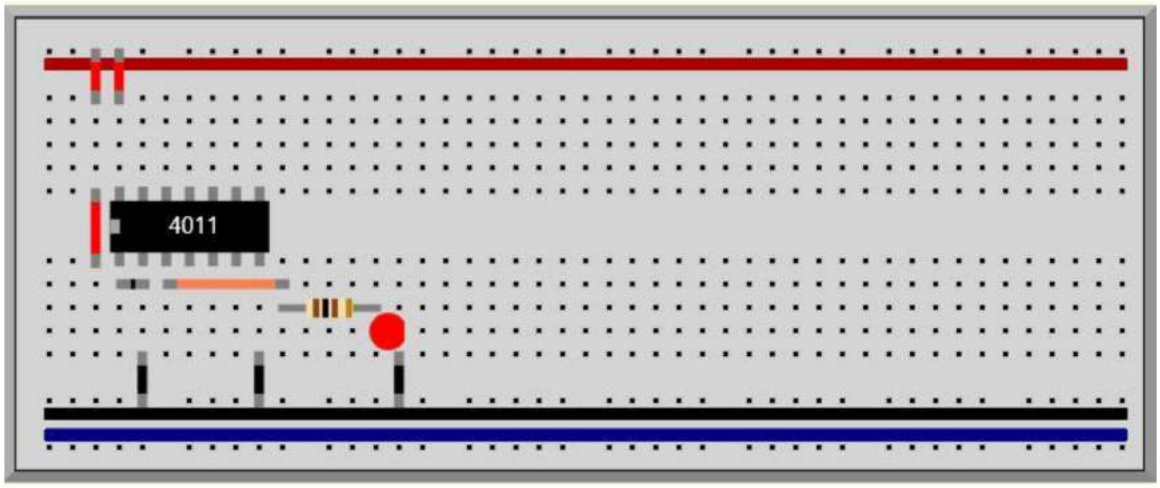
A Simple 2-input logic NAND gate can be constructed using RTL (Resistor-transistor-logic) switches connected together as shown below with the inputs connected directly to the transistor bases. Either transistor must be cut-off or "off" for an output at Q



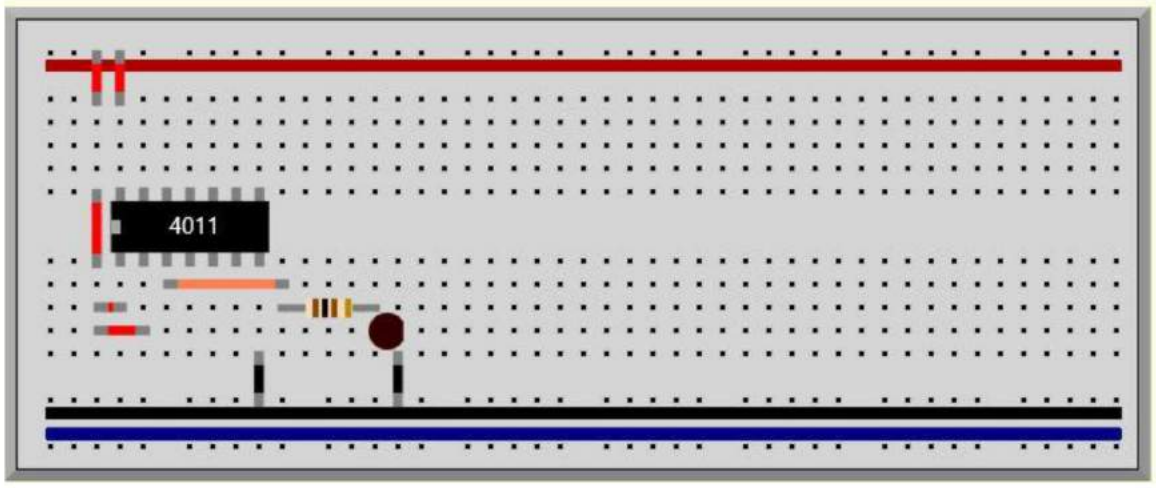

NAND gate through RTL Logic

NAND:

Input:0 0 , Output:1




Input: 1 1 , Output:0



5. NOR gate

This is NOT-OR gate which is equal to an OR gate followed by a NOT gate. The outputs of all NOR gates are low if any of the inputs are high. The symbol is an OR gate with an small circle on the output. The small circle represents inversion.

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

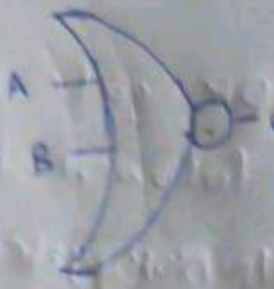



Figure - 18

Logic symbol of NOR gate

| A | B | F |
|---|---|---|
| 0 | 0 | 1 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 0 |

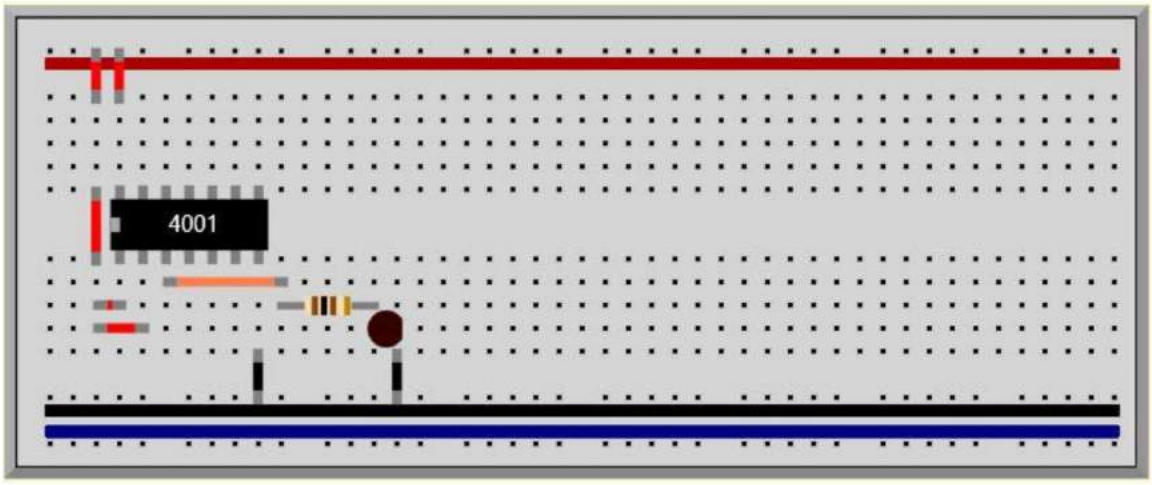
Truth Table of NOR gate.

A simple 2 - input logic NOR gate can be constructed using RTL (Resistor - transistor - logic)

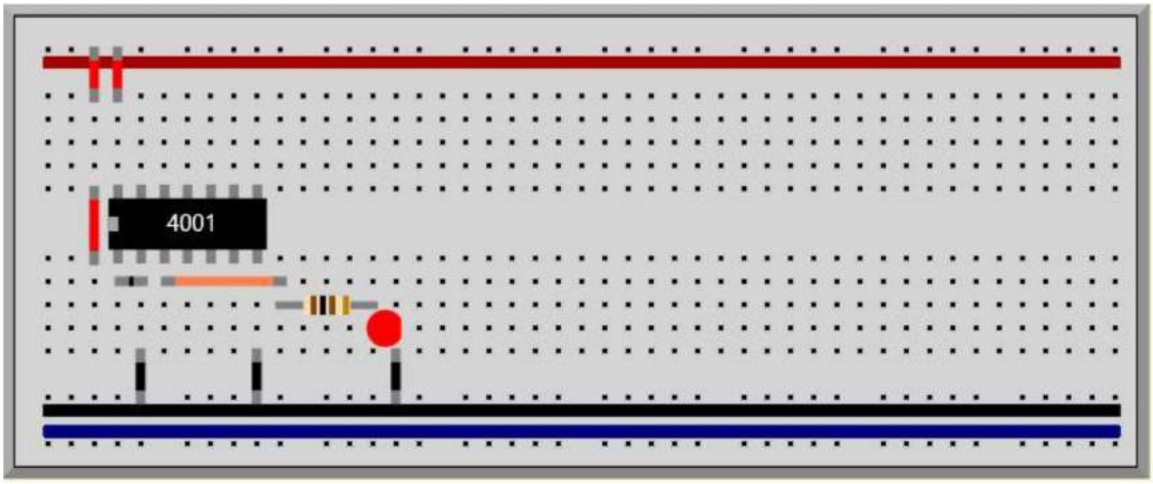
switches connected directly to the transistor bases. Both transistors must be cutoff or "OFF" for an output at Q.

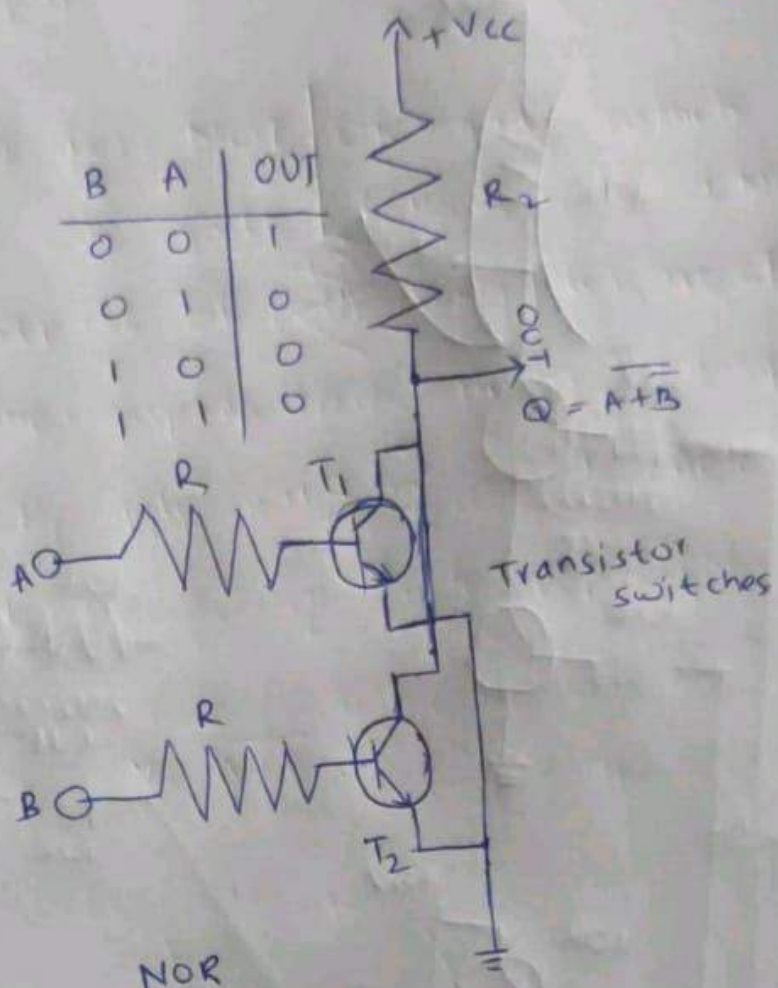
NOR

Input:1 1,Output:0




Input:0 0 ,Output:1

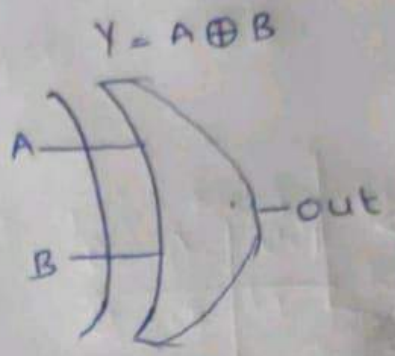





NOR gate through RTL logic.

6) Ex-OR gate

The 'Exclusive-OR' gate is a circuit which will give a high output if either but not both of its two outputs are high. An encircled plus sign (⊕) is used to show the Ex-OR operation.




Logic symbol of Ex-OR gate

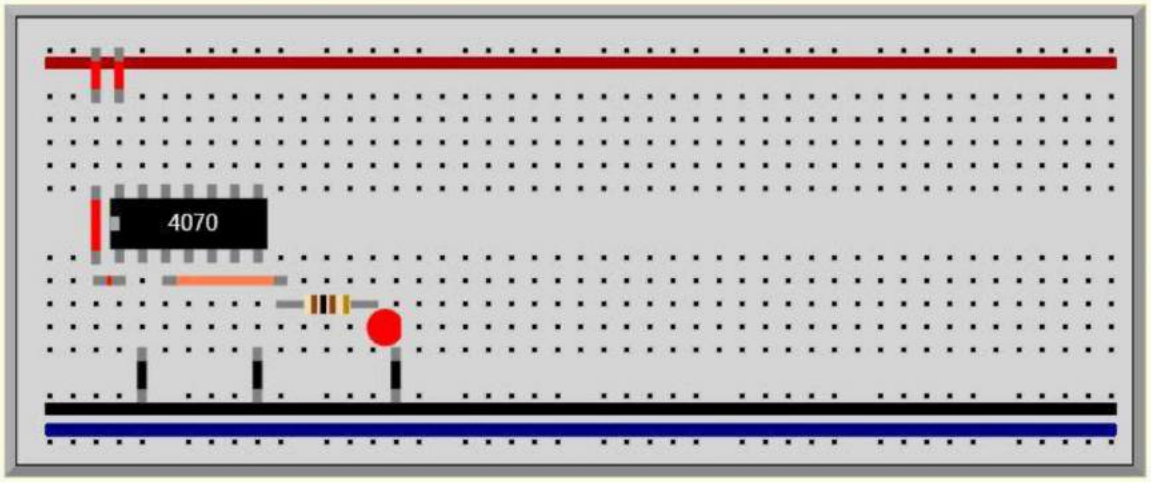
| A | B | AXORB |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

Truth Table of Ex-OR gate

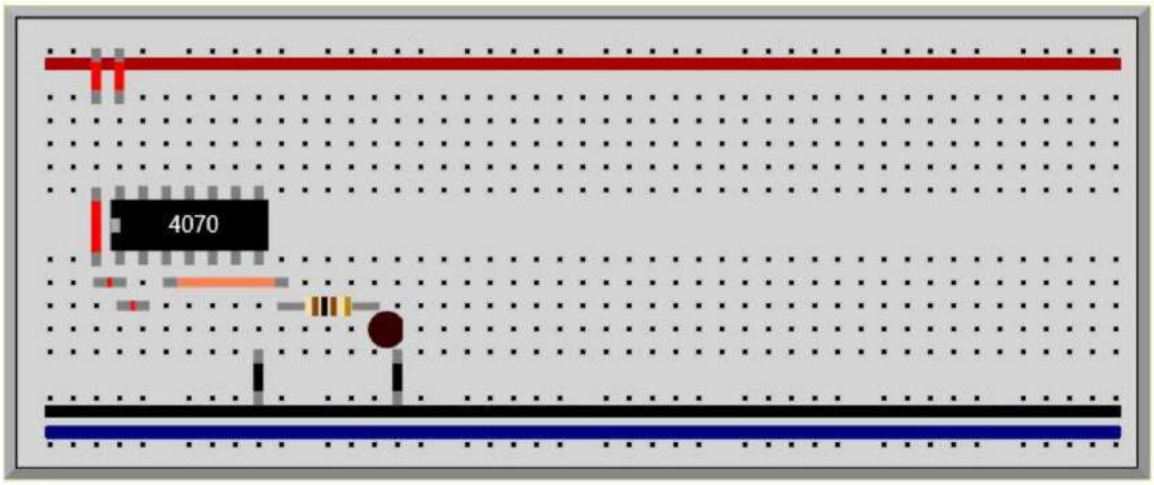
Ex-OR gate is created from AND, NAND and OR gates. The output is high only when both the inputs are different.

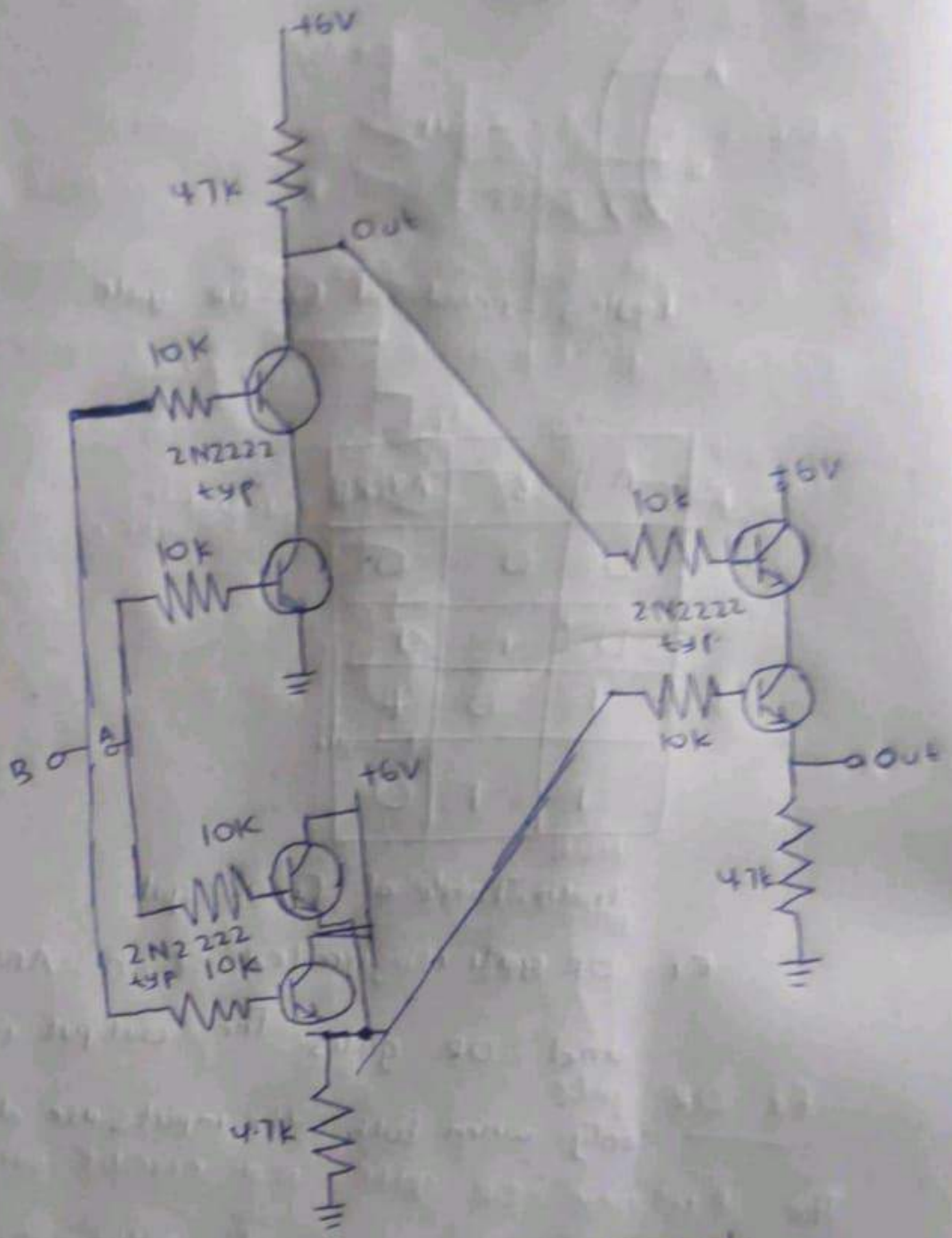
XOR

Input:1 0 ,Output:1




Input:1 1,Output:0






Ex-OR gate through RTL Logic.

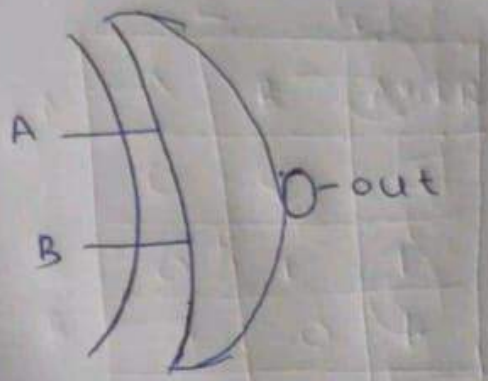
7.) Ex-NOR gate

The 'Exclusive-NOR' gate circuit does the opposite to the Ex-OR gate. It will give output if either, but

not both of its two inputs are high.

The symbol is an Ex-OR gate with a small circle represents inversion.

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



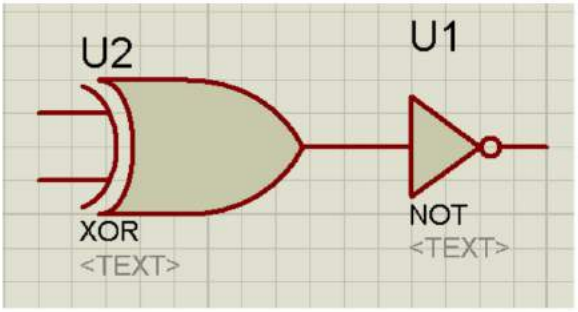

logic symbol of Ex-NOR gate

XNOR Truth Table

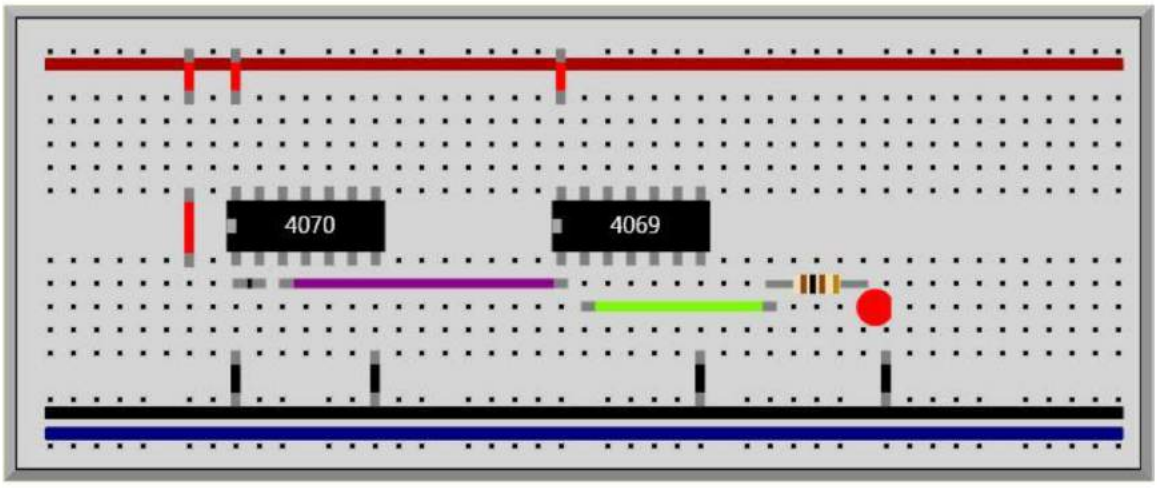
| A | B | Q |
|---|---|---|
| 0 | 0 | 1 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

Ex-NOR gate is created from AND, NOT and OR gates. The output is high only when both the inputs are same.

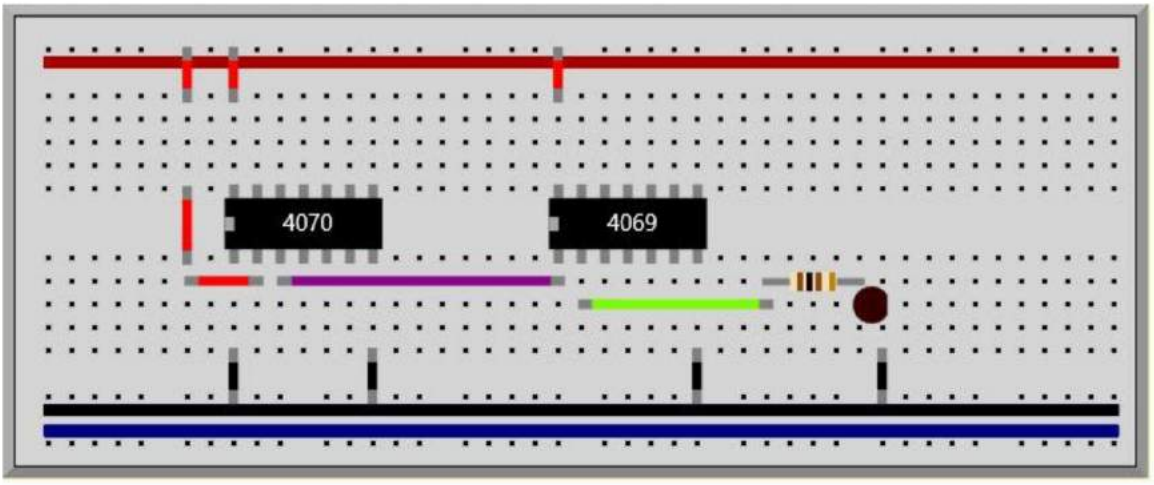
XNOR




Input:0 0 , Output:1




Input:0 1 , Output:0



# Construction of half/full adder using XOR and NAND gates and verification of its operation

## Experiment 2

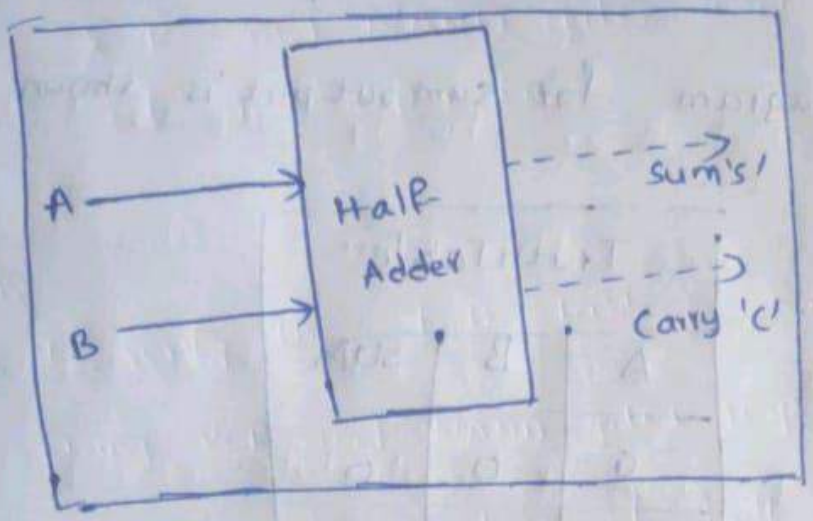
**Aim:**

To verify the truth table of half adder by using XOR and NAND gates respectively and analyse the working of half adder and full adder circuit with the help of LEDs in stimulator 1 and verify the truthtable only of half adder and full adder in simulator 2.

**Theorey:**

Introduction

Adders are digital circuits that carry out addition of numbers. Adders are a key component of arithmetic logic unit. Adders can be constructed for the most of the numerical representations like Binary Coded Decimal (BCD) Excess-3, Gray code, Binary etc. out of these binary addition is the mostly frequently performed task by most common adders. A part from addition, adders are also used




Truth Table

| Input | | Output | |
|---|---|---|---|
| A | B | Sum | Carry |
| 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 1 |

Block diagram and truth table of half adder

The sum output of the binary addition carried out above is similar to that of Ex-OR operation while the carry output is similar to that of and AND operation. The same can be verified with help of Karnaugh Map.

digital
in certain applications like table index calculation, address decoding etc

Binary addition is similar to that of decimal addition, some basic binary additions are shown below

| | | | |
|---|---|---|---|
| 0 | 0 | 1 | 1 |
| +0 | +1 | +0 | +1 |
| 0 | 1 | 1 | carry 1 0 |

Schematic representation of half adder

1) Half Adder :

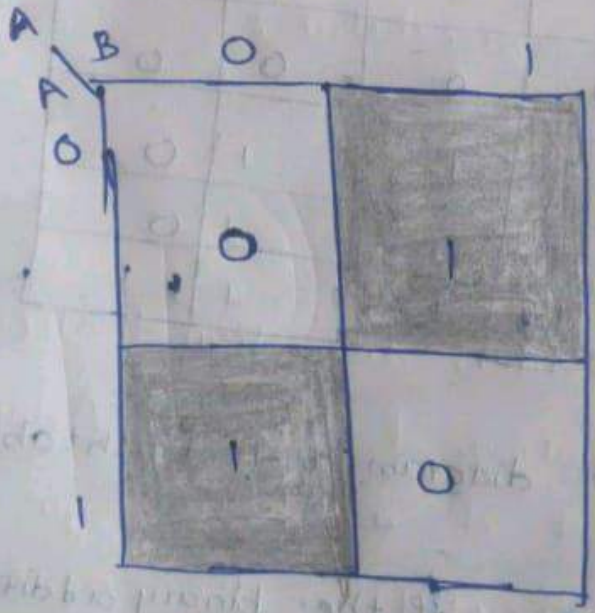
Half adder is a combinational circuit that performs simple addition of two binary numbers

If we assume A and B as the two bits whose addition is to be performed the block diagram and a truth table for half adder with A, B as inputs and Sum, Carry as outputs can be tabulated as follows

The truth table and K Map simplification and logic diagram for sum output is shown below

Truth Table

| A | B | SUM |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |



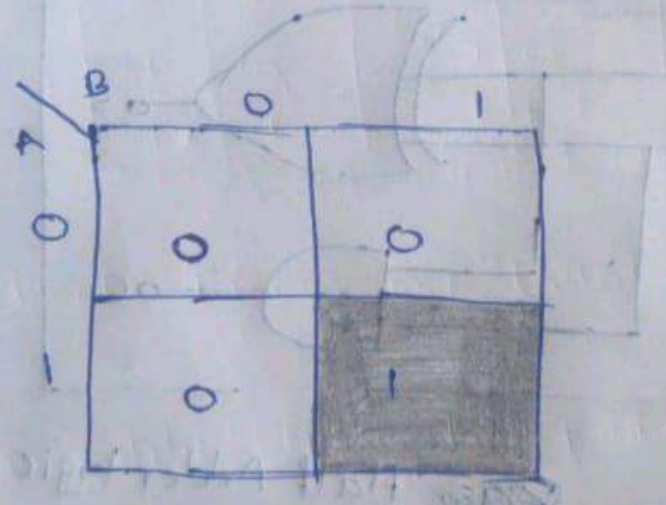


Truth Table, k Map simplification and Logic diagram for sum output of half adder

$$Sum = AB' + A'B$$

The truth table and k Map simplification and logic diagram for carry is shown below.

| Truth Table | | |
|---|---|---|
| A | B | Carry |
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |





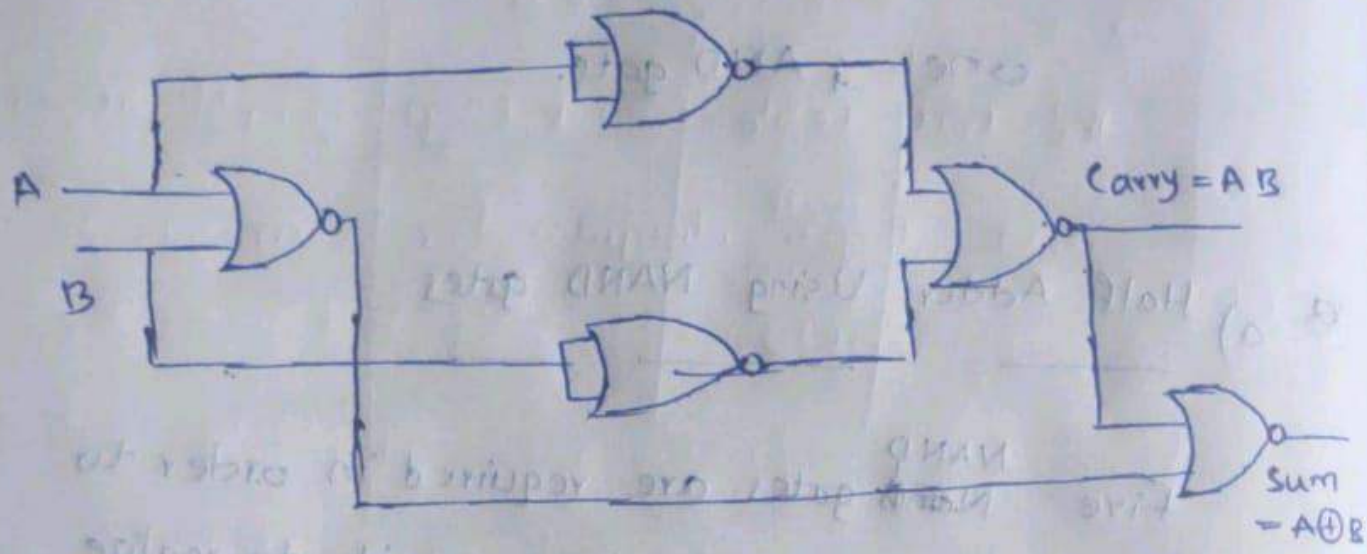
Truth Table, k Map simplification & logic diagram for sum output of half adder.

Carry = AB

If A and B are binary outputs, to the half adder, then the logic function to calculate ~~carry~~ sum S is Ex-OR of A and B and logic function to calculate carry C is AND of A and B. Combining these two, the logical circuit to implement the combinational circuit of half adder is shown below.


Half Adder logic diagram.

As we know that NAND and NOR are called universal gates as any logic system can be implemented using these two, the half adder circuit can also be implemented using them. We know that a half

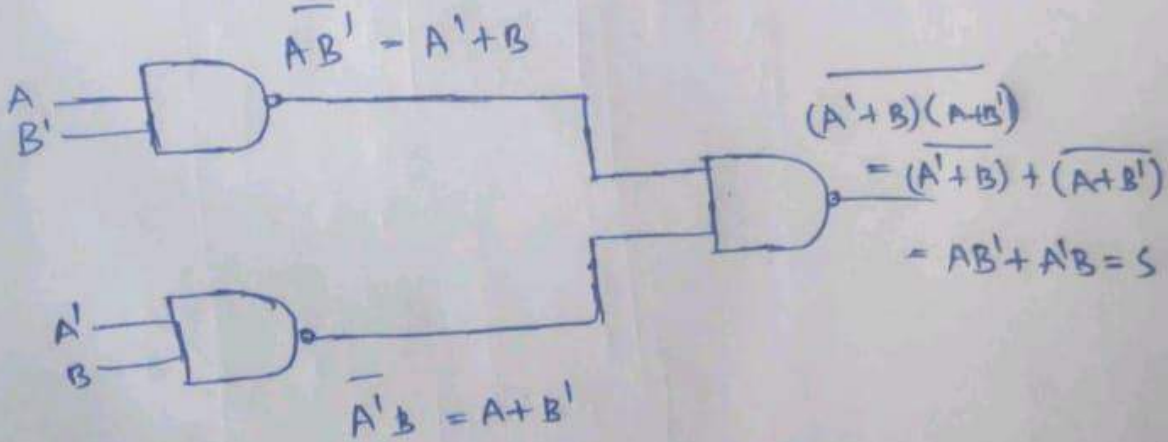


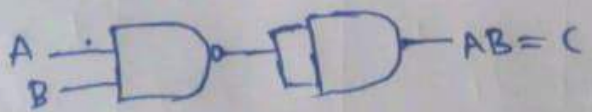

Realization of half adder
using NAND gates

adder Circuit has one Ex-OR gate and one AND gate.

a) Half Adder Using NAND gates

Five NAND gates are required in order to design a half adder. The circuit to realize half adder using NAND gates is shown below.



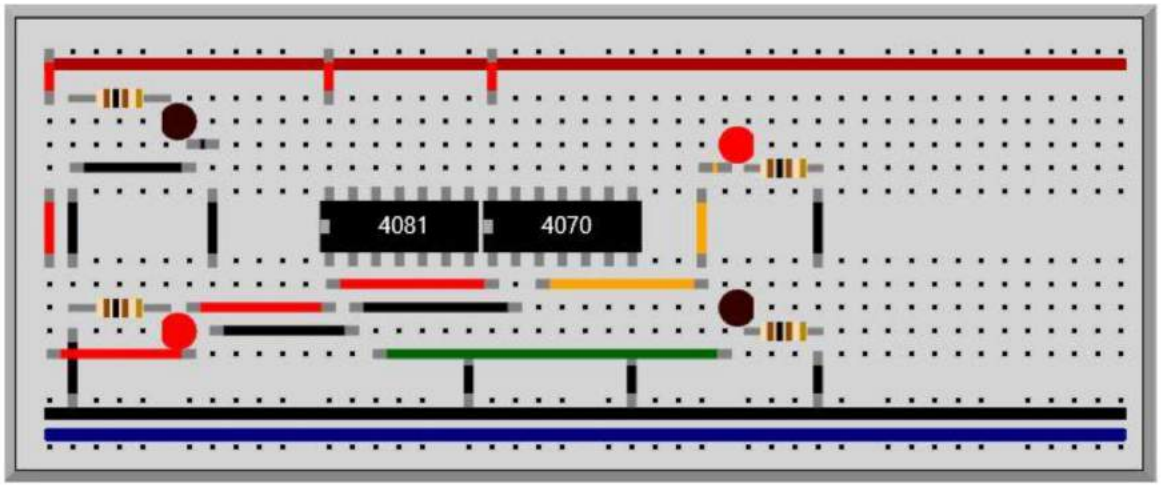

Realization of half adder using NAND gates.

b) Half Adder Using NOR gates

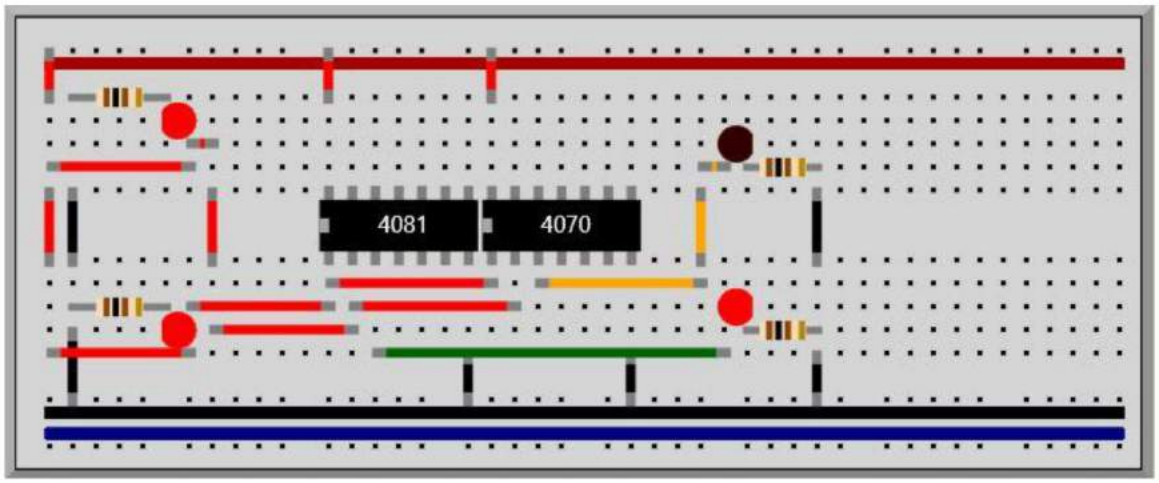
Five NOR gates are required in order to design a half adder. The circuit to realize half adder using NOR gates is shown below.

Half Adder

Input:0 1, Output:Sum:1, Carry:0



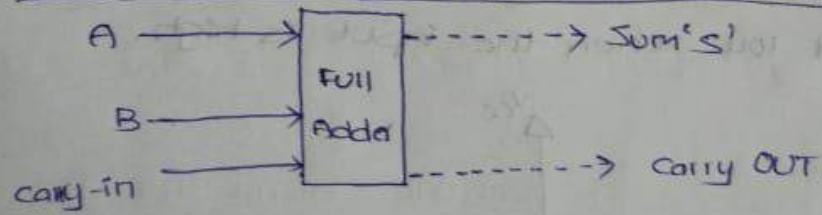

Input:1 1, Output:Sum:0, Carry:1



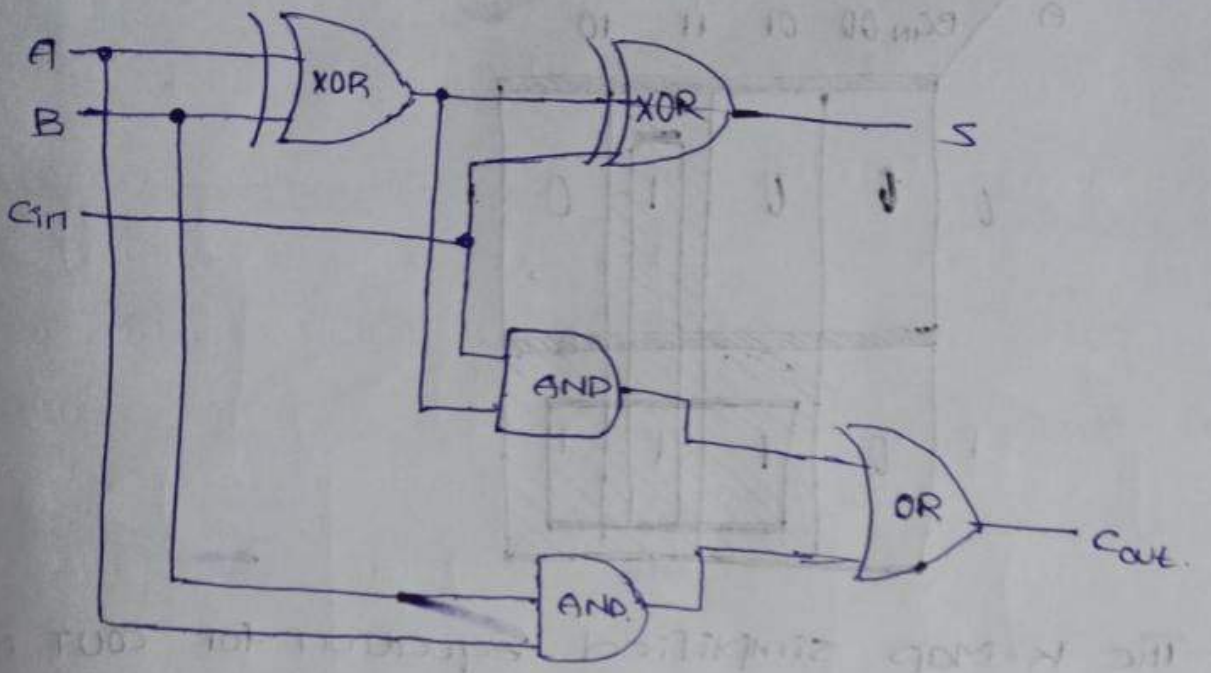
2) full Adder

→ It is a digital circuit used to calculate the sum of three Binary bits. full adders are complex and difficult to implement when compared to half adders. Two of the three bits are same as before which are A. The augend bit and B, the addend bit. The additional third bit is carry bit from the previous stage and is called 'carry' - in generally represented by & CIN. It calculates the sum of three bits along with the carry. The output carry is called carry-out and is respresented by carry OUT.

⟺ FULL Adder BLOCK Diagram and Truth Table :-

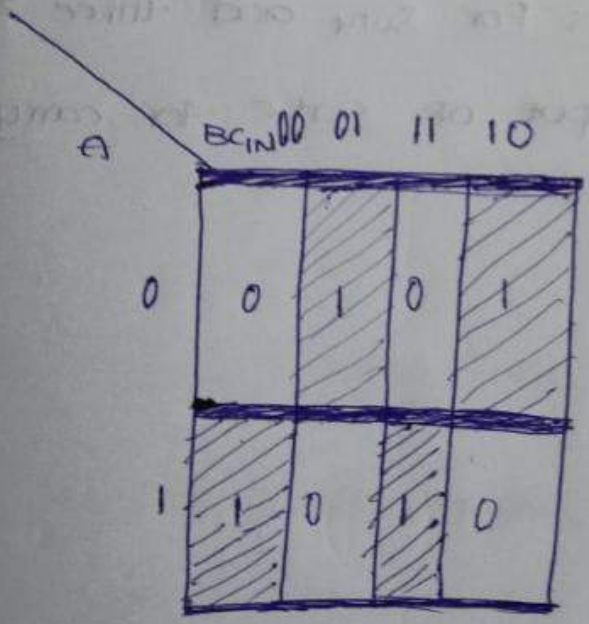



| Input | | | Output | |
|---|---|---|---|---|
| A | B | Cin | Sum | Carry |
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 | 0 |
| 0 | 1 | 0 | 1 | 0 |
| 0 | 1 | 1 | 0 | 1 |
| 1 | 0 | 0 | 0 | 1 |
| 1 | 0 | 1 | 0 | 1 |
| 1 | 1 | 0 | 0 | 1 |
| 1 | 1 | 1 | 1 | 1 |



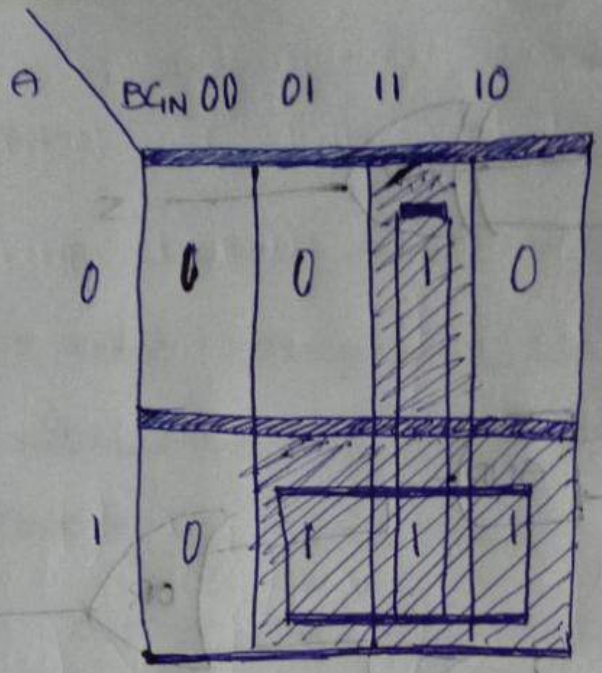

: Full Adder Logic Diagram

Based on the truth table, the Boolean function for sum (S) and carry-out (COUT) derived using K-Map




→ The K-Map simplified equation for sum is

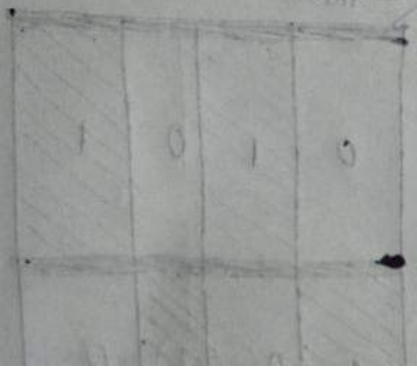
$S = A'B'C_{in} + A'BC_{in}' + ABC_{in}$




:-The K-Map simplified equation for COUT is

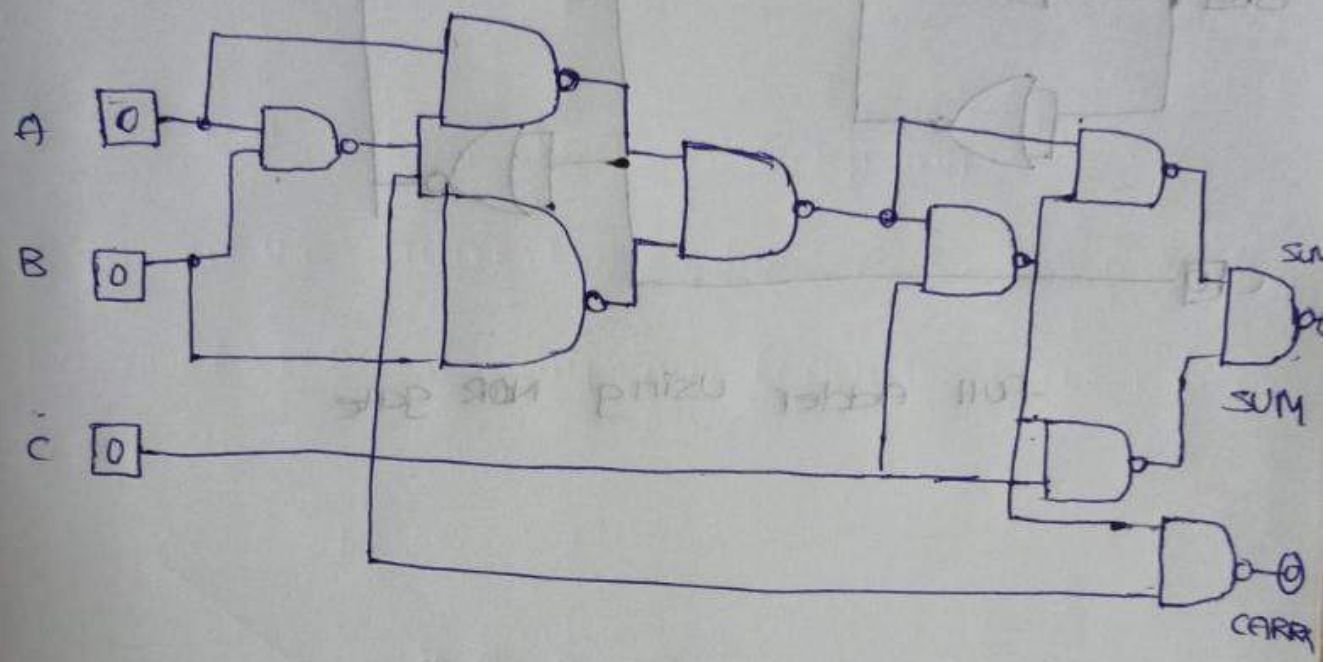
$$COUT = AB + ACIN + BCIN$$

In order to implement a combinational circuit for full adder, it is clear from the equations derived above, that we need four 3-input AND gates and one 4-input OR gates for Sum and three 2-input AND gates and one 3-input OR gate for carry-out.



## (2.1) Full Adder using NAND gates

→ As mentioned earlier, a NAND gate is one of the universal gates and can be used to implement any logic design. The circuit of full adder using only NAND gates is:

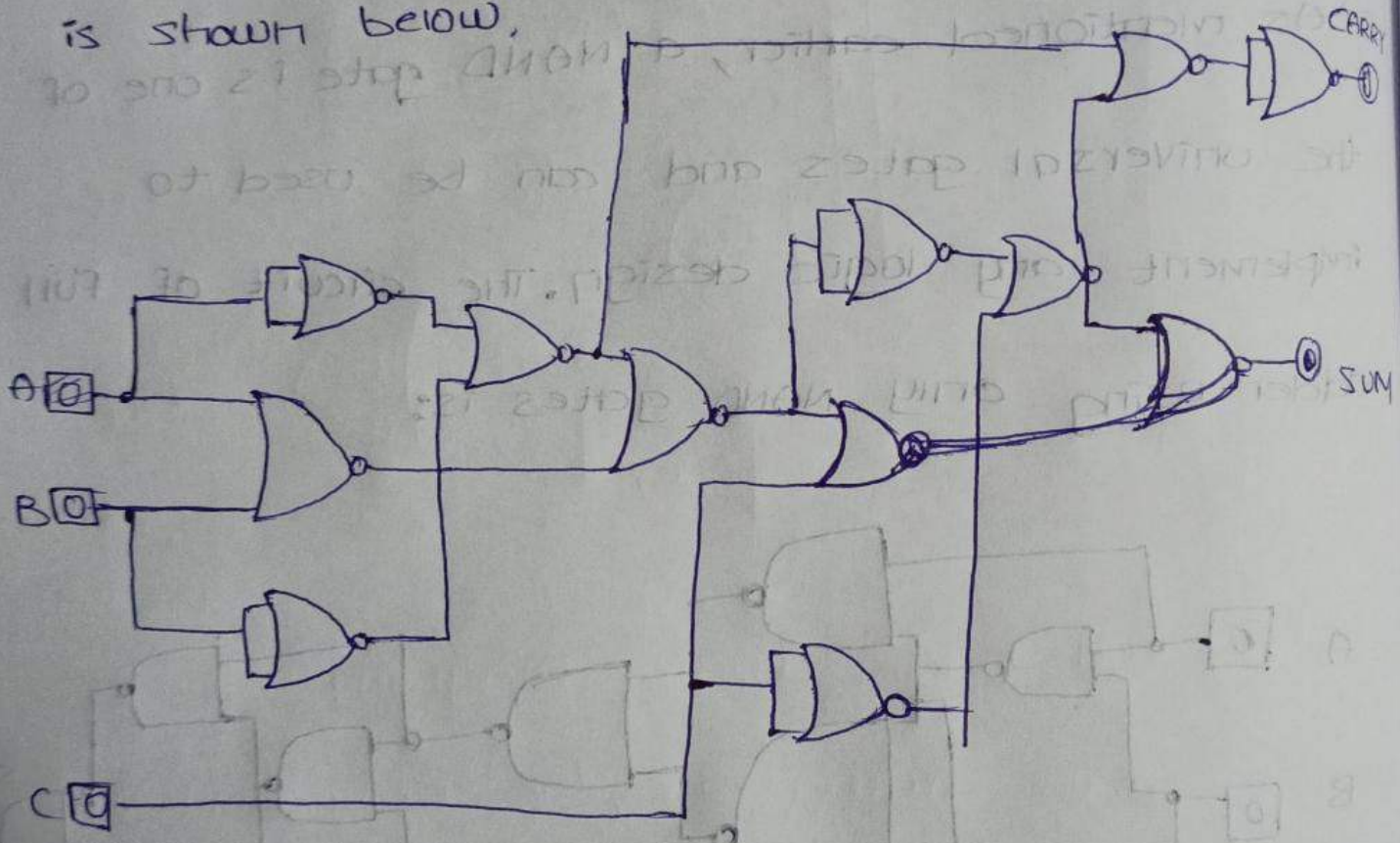



% Full Adder using NAND Gates

## (2.2) Full Adder using NOR gates

→ As mentioned earlier, a NOR gate is one of the universal gates and can be used to implement any logic design.

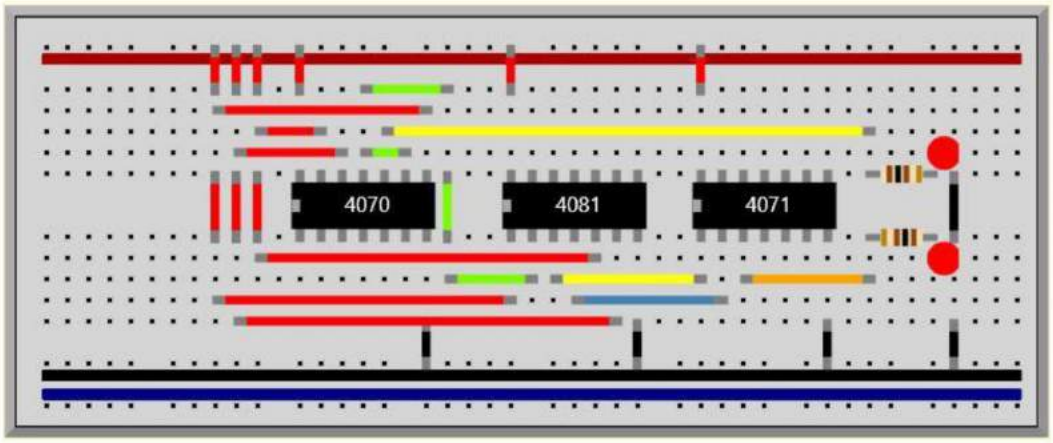
The circuit of full adder using only NOR gate is shown below,



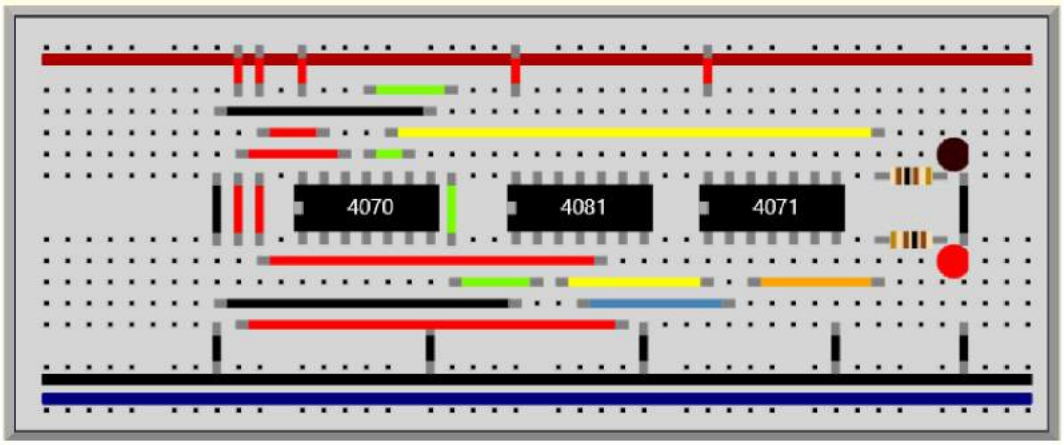

Full Adder using NOR gate

Full Adder

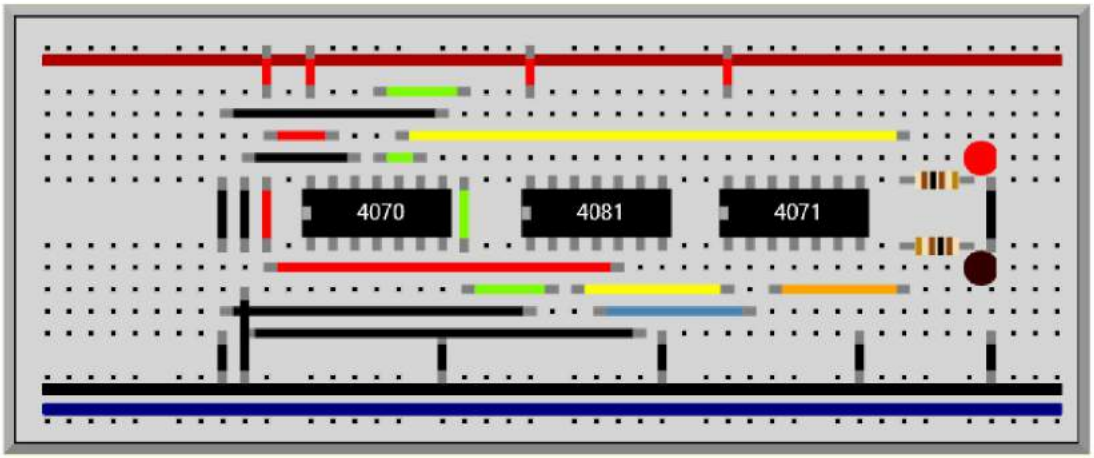
Input:1 1 1,Output:Sum:1,Carry:1




Input:1 1 0,Output:Sum:0,Carry:1




Input:1 0 0,Output:Sum:1,Carry:0



Input:0 0 0,Output:Sum:0,Carry:0