LAB REPORT-6

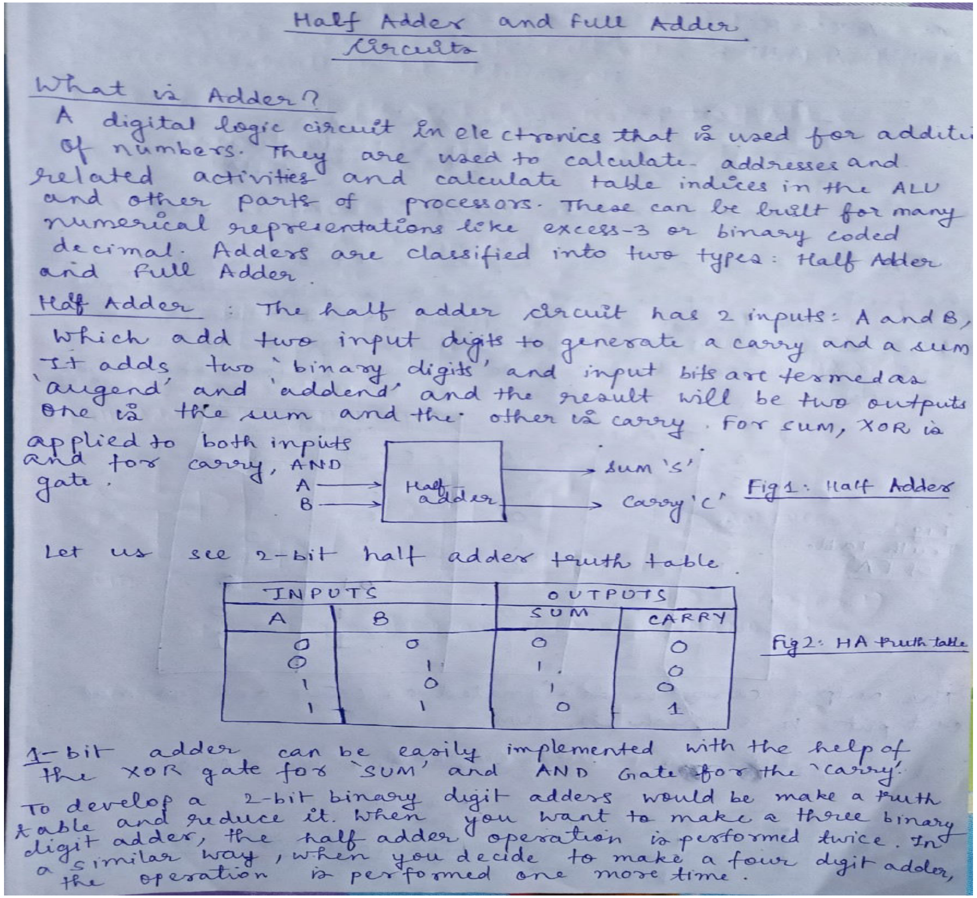
NAME:SWARUP PADHI
ROLL:20CS30062
BATCH:5

Part-1

AIM:

- Verify the truth table of half adder and full adder by using XOR and NAND gates respectively.
- Analyse the working of half adder and full adder circuit with the help of LEDs.

Theory and equations(half adder and full adder circuit)



Half Adder and full Adder Circuits

What is Adder?

A digital logic circuit in electronics that is used for additi of numbers. They are used to calculate addresses and related activities and calculate table indices in the ALU and other parts of processors. These can be built for many numerical representations like excess-3 or binary coded decimal. Adders are classified into two types: Half Adder and full Adder.

Half Adder : The half adder circuit has 2 inputs: A and B, which add two input digits to generate a carry and a sum It adds two 'binary digits' and input bits are termed as 'augend' and 'addend' and the result will be two outputs one is the sum and the other is carry. For sum, XOR is applied to both inputs and for carry, AND gate.

Fig 1: Half Adder

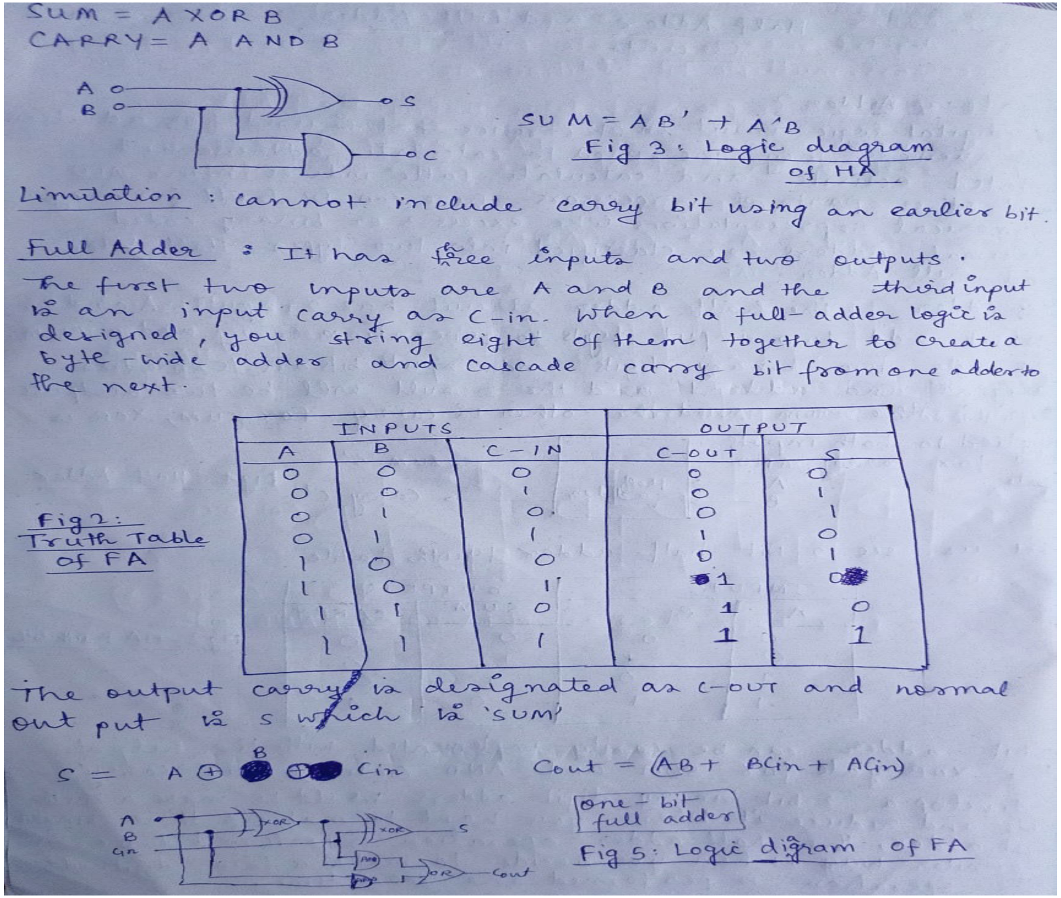
Let us see 2-bit half adder truth table.

INPUTS		OUTPUTS	
A	B	SUM	CARRY
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Fig 2: HA truth table

1-bit adder can be easily implemented with the help of the XOR gate for 'SUM' and AND Gate for the 'carry'.

To develop a 2-bit binary digit adders would be make a truth table and reduce it. When you want to make a three binary digit adder, the half adder operation is performed twice. In a similar way, when you decide to make a four digit adder, the operation is performed one more time.



SUM = A XOR B
CARRY = A AND B

SUM = AB' + A'B

Fig 3: Logic diagram of HA

Limitation : cannot include carry bit using an earlier bit.

Full Adder : It has three inputs and two outputs. The first two inputs are A and B and the third input is an input carry as C-in. When a full-adder logic is designed, you string eight of them together to create a byte-wide adder and cascade carry bit from one adder to the next.

INPUTS			OUTPUT	
A	B	C-IN	C-OUT	S
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

Fig 2: Truth Table of FA

The output carry is designated as C-OUT and normal out put is S which is 'SUM'

$S = A \oplus B \oplus Cin$ $Cout = (AB + BCin + ACin)$

one-bit full adder

Fig 5: Logic digram of FA

Half Adder is used in computers, calculators, devices used for digital measuring etc. Full Adder is used in digital processors, multiple bit addition, etc.

Full Adder Implementation using Half Adders

The implementation of a FA can be done through two half-adders which are connected logically.

$$S = (A \oplus B) \oplus Cin$$

$$Cout = AB + Cin(A \oplus B).$$

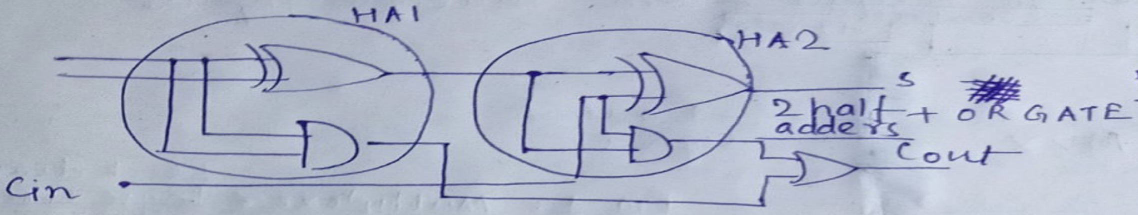


Fig6: Logic digram of FA using HA and OR gate

Full Adder using NAND gates

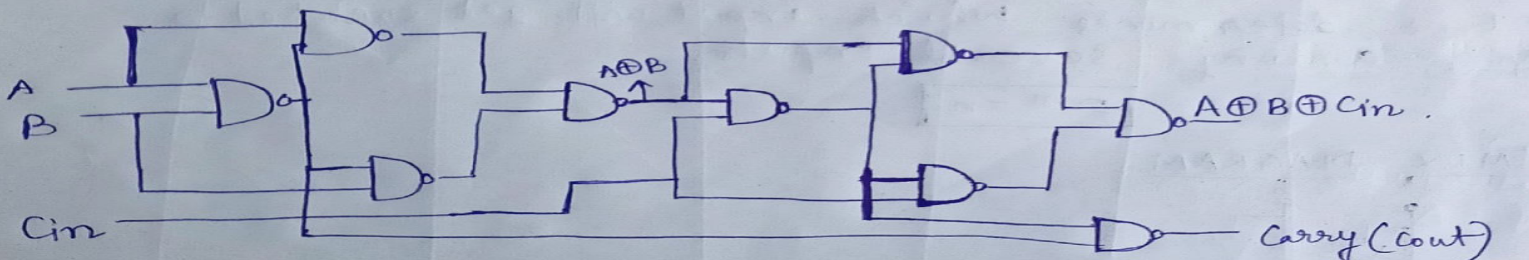
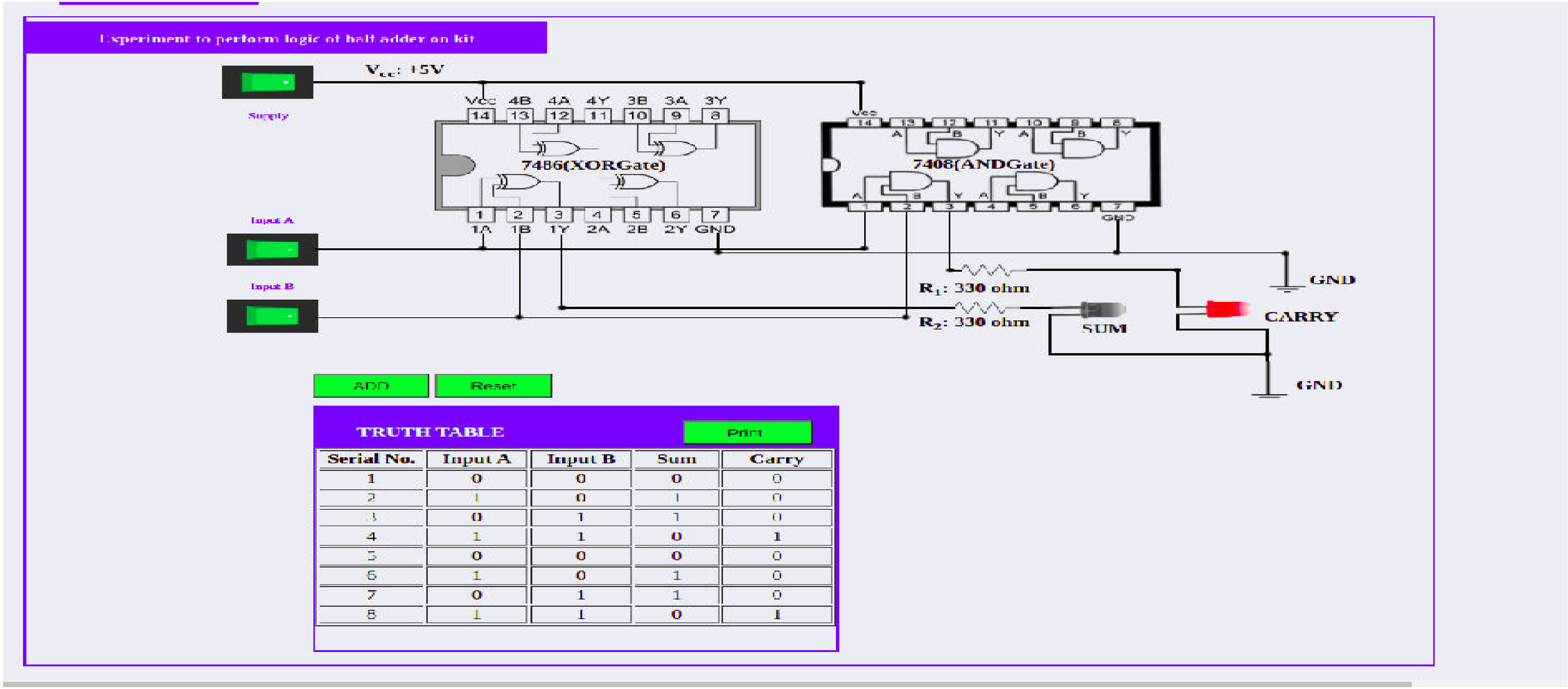


Fig 7: Logic diagram of FA using NAND gates

Simulations(Half Adder)

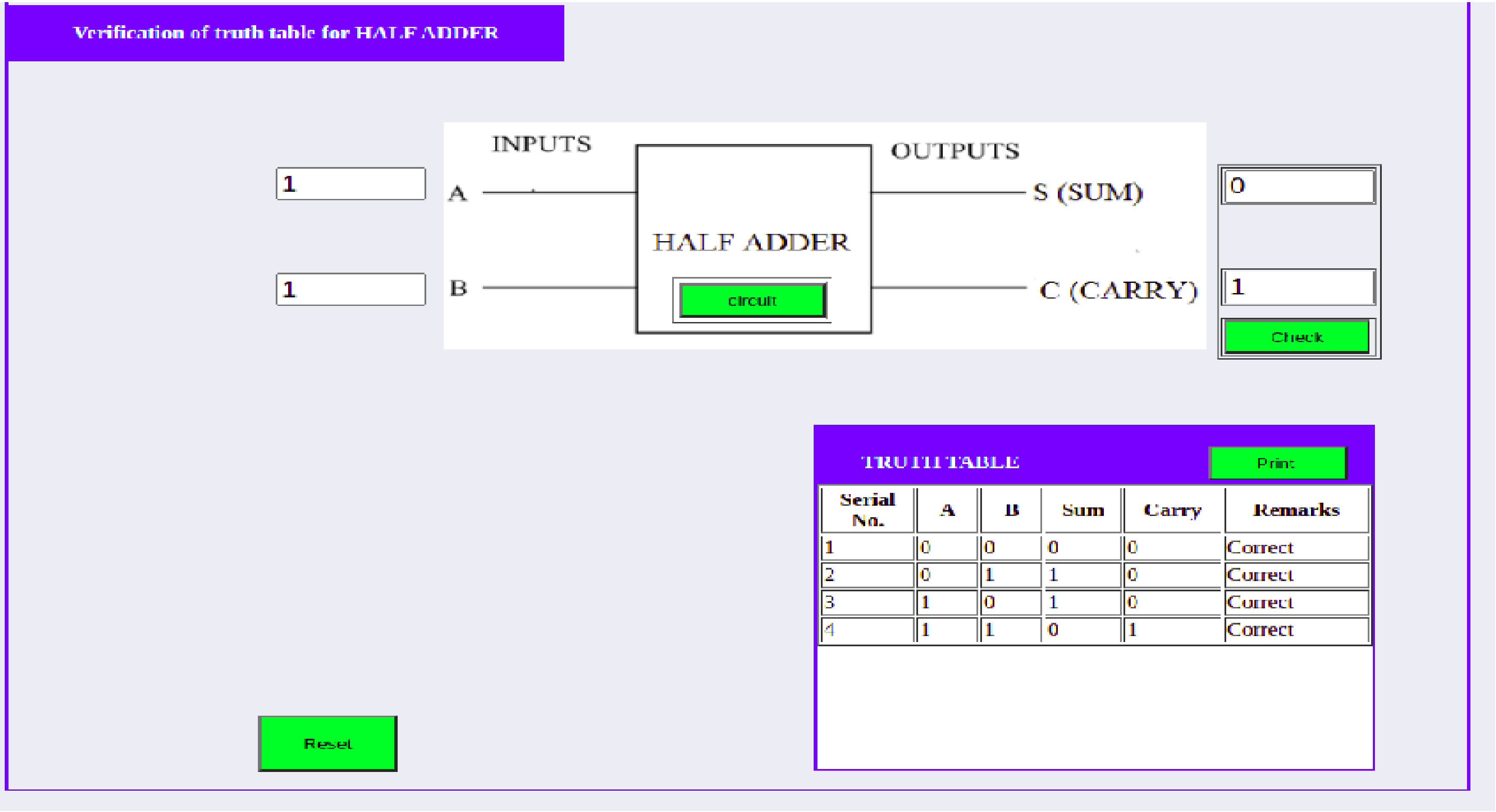


Experiment to perform logic of half adder on kit

TRUTH TABLE

Serial No.	Input A	Input B	Sum	Carry
1	0	0	0	0
2	1	0	1	0
3	0	1	1	0
4	1	1	0	1
5	0	0	0	0
6	1	0	1	0
7	0	1	1	0
8	1	1	0	1

Truth table verification



Verification of truth table for HALF ADDER

A 1

B 1

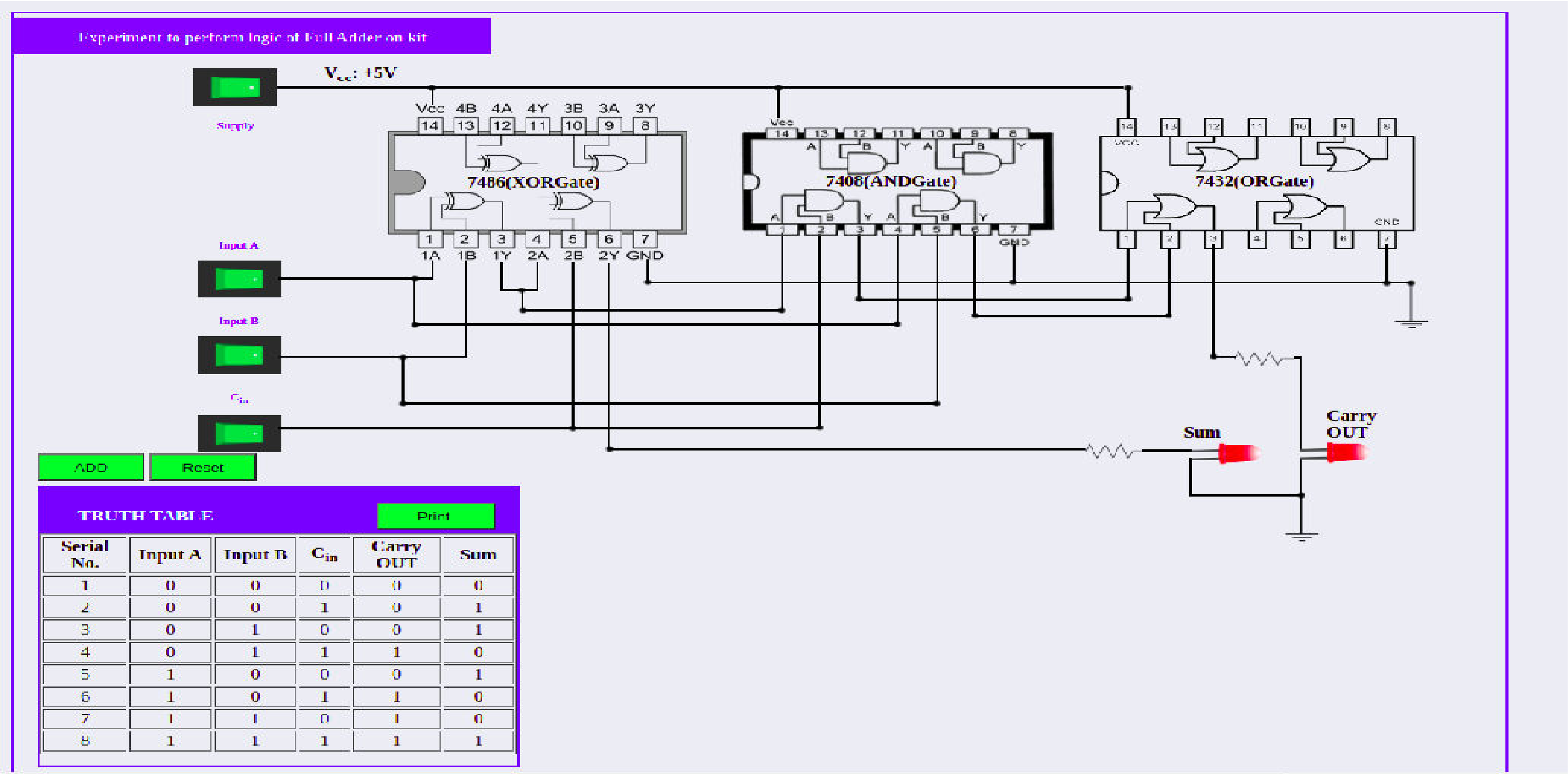
S (SUM) 0

C (CARRY) 1

TRUTH TABLE

Serial No.	A	B	Sum	Carry	Remarks
1	0	0	0	0	Correct
2	0	1	1	0	Correct
3	1	0	1	0	Correct
4	1	1	0	1	Correct

Simulations(Full Adder)

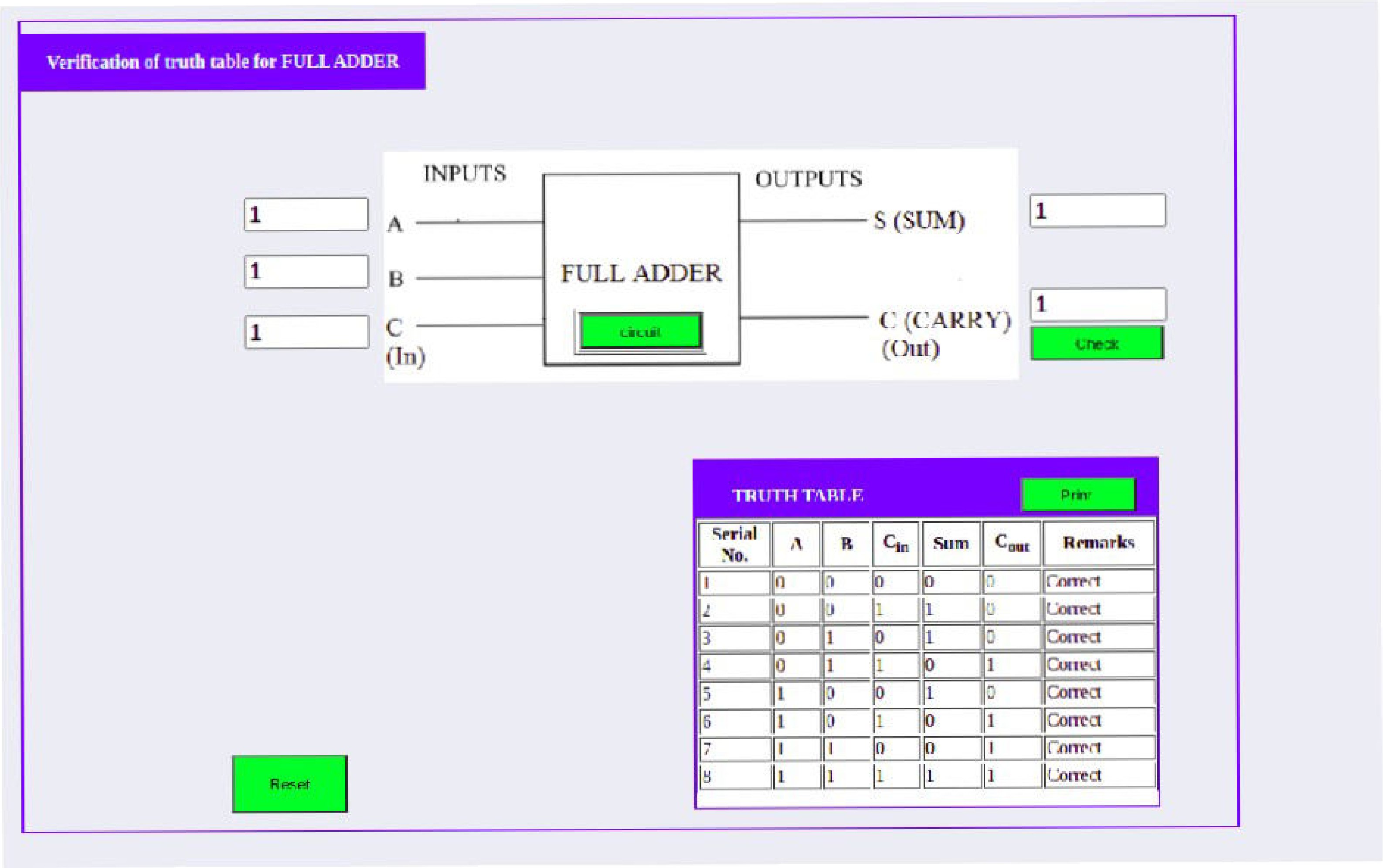


Experiment to perform logic of Full Adder on kit

TRUTH TABLE

Serial No.	Input A	Input B	C_{in}	Carry OUT	Sum
1	0	0	0	0	0
2	0	0	1	0	1
3	0	1	0	0	1
4	0	1	1	1	0
5	1	0	0	0	1
6	1	0	1	1	0
7	1	1	0	1	0
8	1	1	1	1	1

Truth table verification



A: 1

B: 1

C (In): 1

S (SUM): 1

C (CARRY) (Out): 1

Serial No.	A	B	C_{in}	Sum	C_{out}	Remarks
1	0	0	0	0	0	Correct
2	0	0	1	1	0	Correct
3	0	1	0	1	0	Correct
4	0	1	1	0	1	Correct
5	1	0	0	1	0	Correct
6	1	0	1	0	1	Correct
7	1	1	0	0	1	Correct
8	1	1	1	1	1	Correct

Discussions and Conclusion

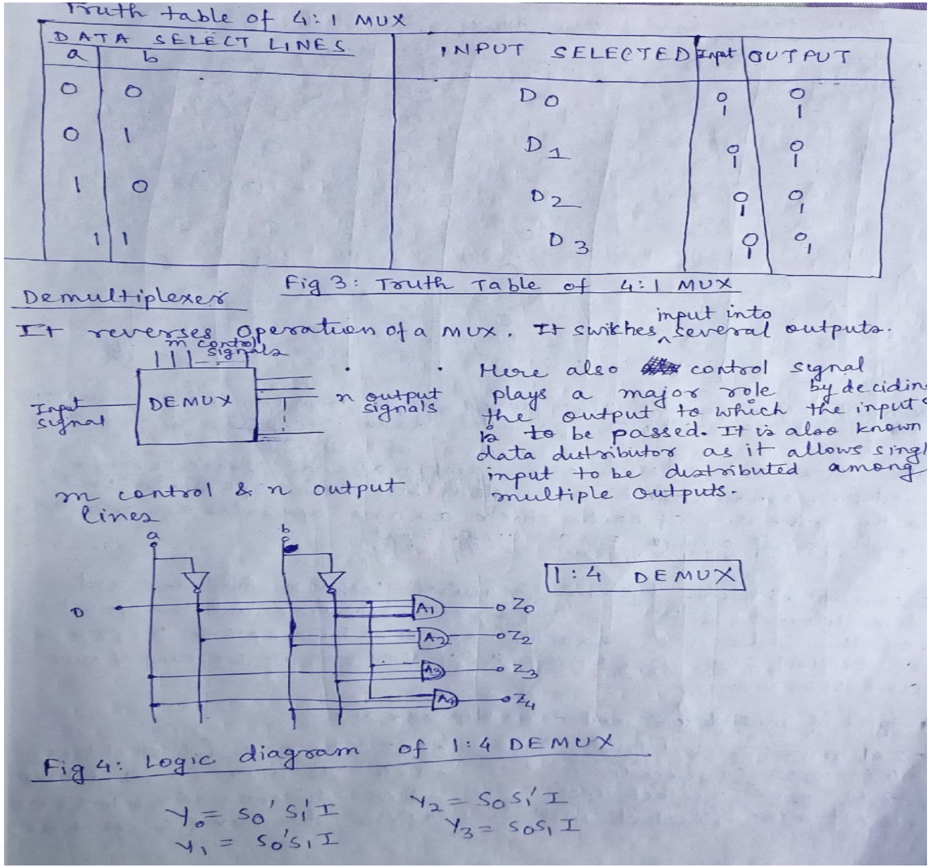
- Half adders and Full adders can also be implemented using NAND and NOR gates. The Karnaugh maps help us in reducing the expressions further and we can build a half adder using just 5 NOR gates, and full adder using 9 NOR gates. Half adder can also be constructed using 5 NAND gates and full adder can be constructed using 9 NAND gates.
- Same gates are preferred as transistors,MOSFET’s are not ideal and it ensures similar delays and same value of other attributes which help us in analysing circuit better.
- We can add n-bit binary numbers by cascading the full adder circuit one after the other,i.e,using half adder for the least significant bit and full adder for subsequent bits.
- A full adder can be expressed in terms of a half adder circuit using two half adders and a or gate.They form basis of ALU of our computers that we use everyday.

Part-2

AIM:

- To analyse the truth table and working of 1x4 De-Multiplexer by using 3-input NAND and 1-input NOT logic gate ICs
- To analyse the truth table and working of 4x1 Multiplexer by using 3-input AND, 3-input OR, and 1-input NOT logic gate ICs.

Theory and Equations(Multiplexer and Demultiplexer)



Truth table of 4:1 MUX

DATA SELECT LINES		INPUT SELECTED	Input	OUTPUT
a	b			
0	0	D0	0 1	0 1
0	1	D1	0 1	0 1
1	0	D2	0 1	0 1
1	1	D3	0 1	0 1

Fig 3: Truth Table of 4:1 MUX

Demultiplexer

It reverses Operation of a MUX. It swiktches input into several outputs.

m control signals
Input signal
DEMUX
n output signals

Here also control signal plays a major role by decidin the output to which the input is to be passed. It is also known data distributor as it allows singl input to be distributed among multiple outputs.

m control & n output lines

1:4 DEMUX

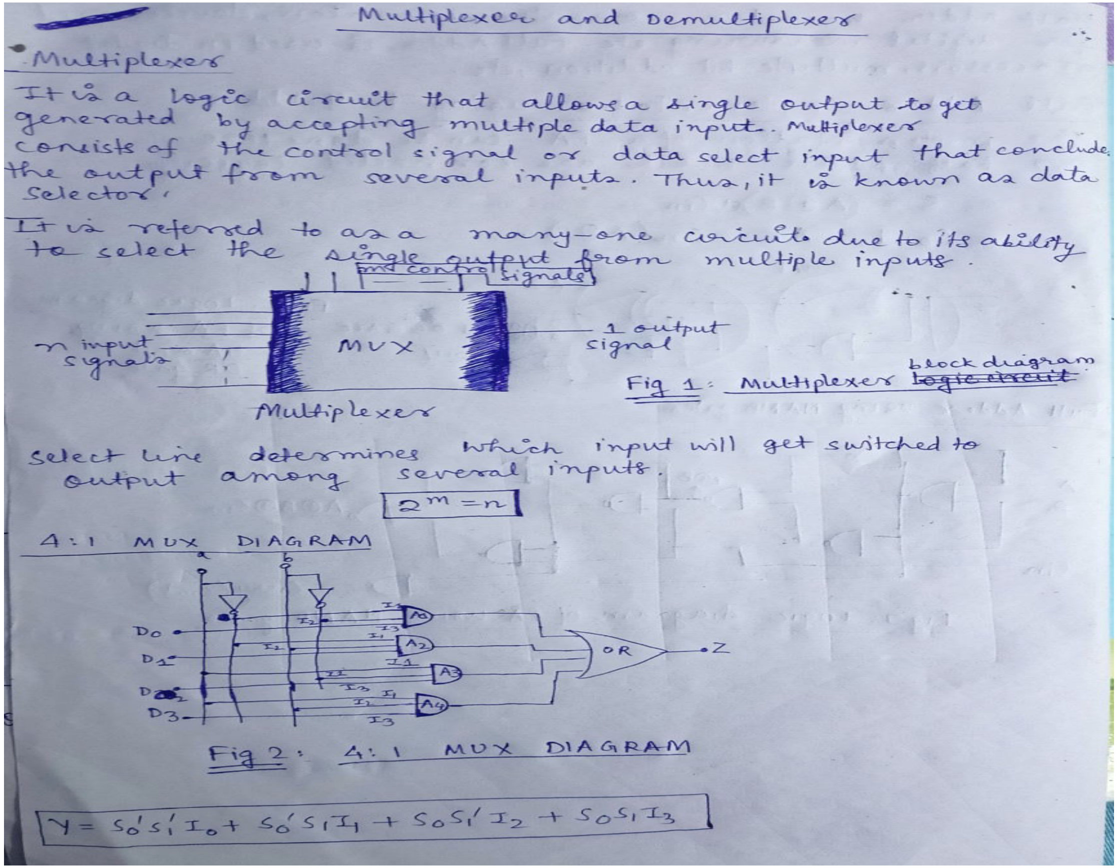
Fig 4: Logic diagram of 1:4 DEMUX

$Y_0 = S_0' S_1' I$

$Y_1 = S_0' S_1 I$

$Y_2 = S_0 S_1' I$

$Y_3 = S_0 S_1 I$



Multiplexer and Demultiplexer

Multiplexer

It is a logic circuit that allows a single output to get generated by accepting multiple data input. Multiplexer consists of the control signal or data select input that conclude the output from several inputs. Thus, it is known as data Selector.

It is referred to as a many-one circuit due to its ability to select the single output from multiple inputs.

Fig 1: Multiplexer block diagram ~~logic circuit~~

Select line determines which input will get switched to output among several inputs.

$\boxed{2^m = n}$

4:1 MUX DIAGRAM

Fig 2: 4:1 MUX DIAGRAM

$\boxed{Y = S_0' S_1' I_0 + S_0' S_1 I_1 + S_0 S_1' I_2 + S_0 S_1 I_3}$

Fig 5: Truth table for 1:4 DEMUX

A	B	D	Z_0	Z_1	Z_2	Z_3
0	0	0	0	0	0	0
0	0	1	1	0	0	0
0	1	0	0	0	0	0
0	1	1	0	1	0	0
1	0	0	0	0	0	0
1	0	1	0	0	1	0
1	1	0	0	0	0	0
1	1	1	0	0	0	1

Demultiplexer circuit also plays a major role in the communicatio system as sometimes parallel data reception is required

Both MUX and DEMUX are combinational logic circuits used in communication system but their operation is exactly reverse of each other as one operates on multiple inputs and other on a single input.

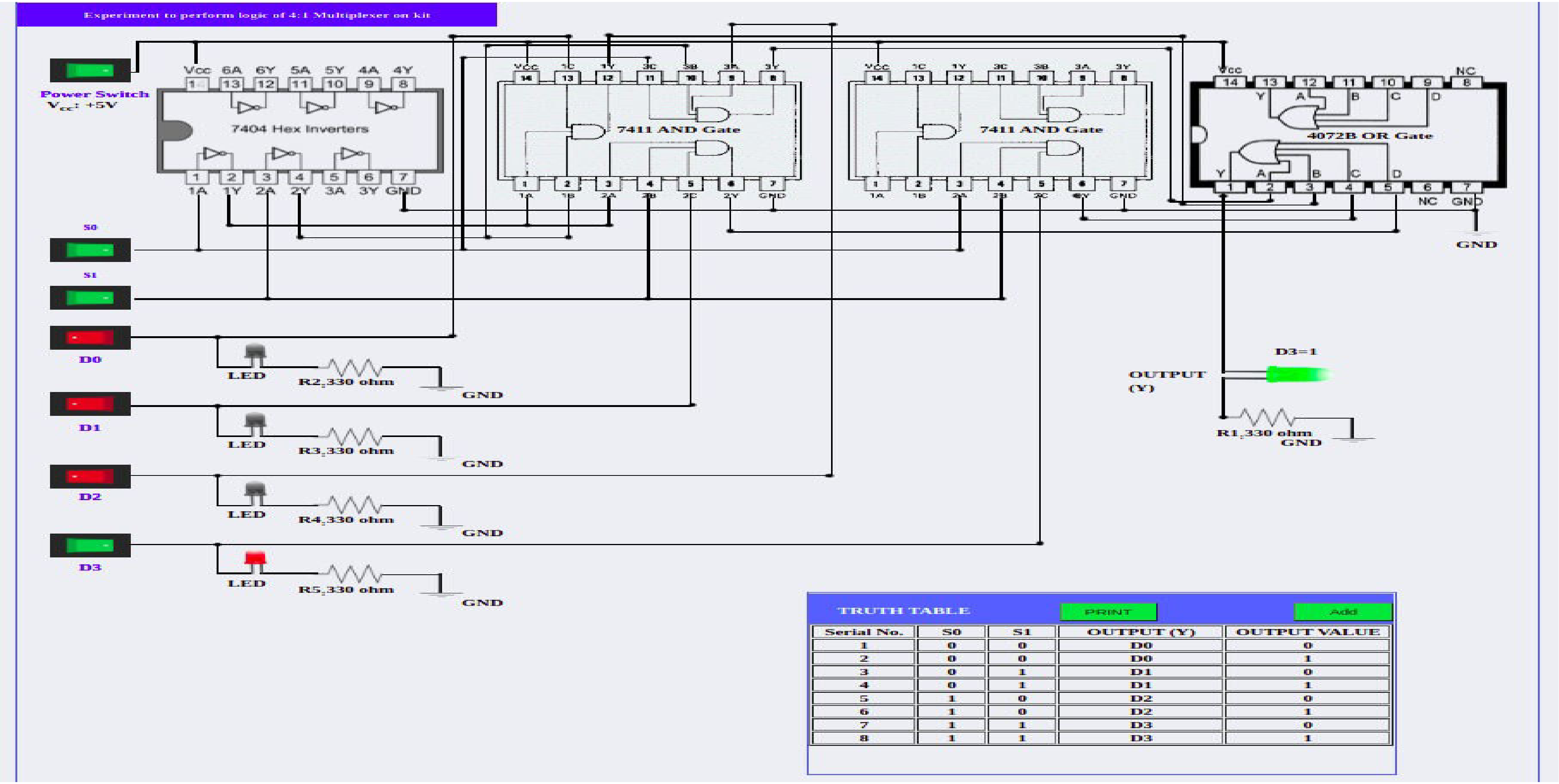
A communication system requires both multiplexer and demultiplexer due to its bi-directional nature but the operation of the two are exactly opposite to each other. The presence of control signals plays a crucial role in working of MUX and DEMUX.

4-types of multiplexer and demultiplexer are

- 2×1 mux (1 select line)
- 4×1 mux (2 select lines)
- 8×1 mux (3 select lines)
- 16×1 mux (4 select lines)

- 1×2 de-mux (1 select line)
- 1×4 de-mux (2 select lines)
- 1×8 de-mux (3 select lin
- 1×16 de-mux (4 select line

Simulation(4:1 Multiplexer)

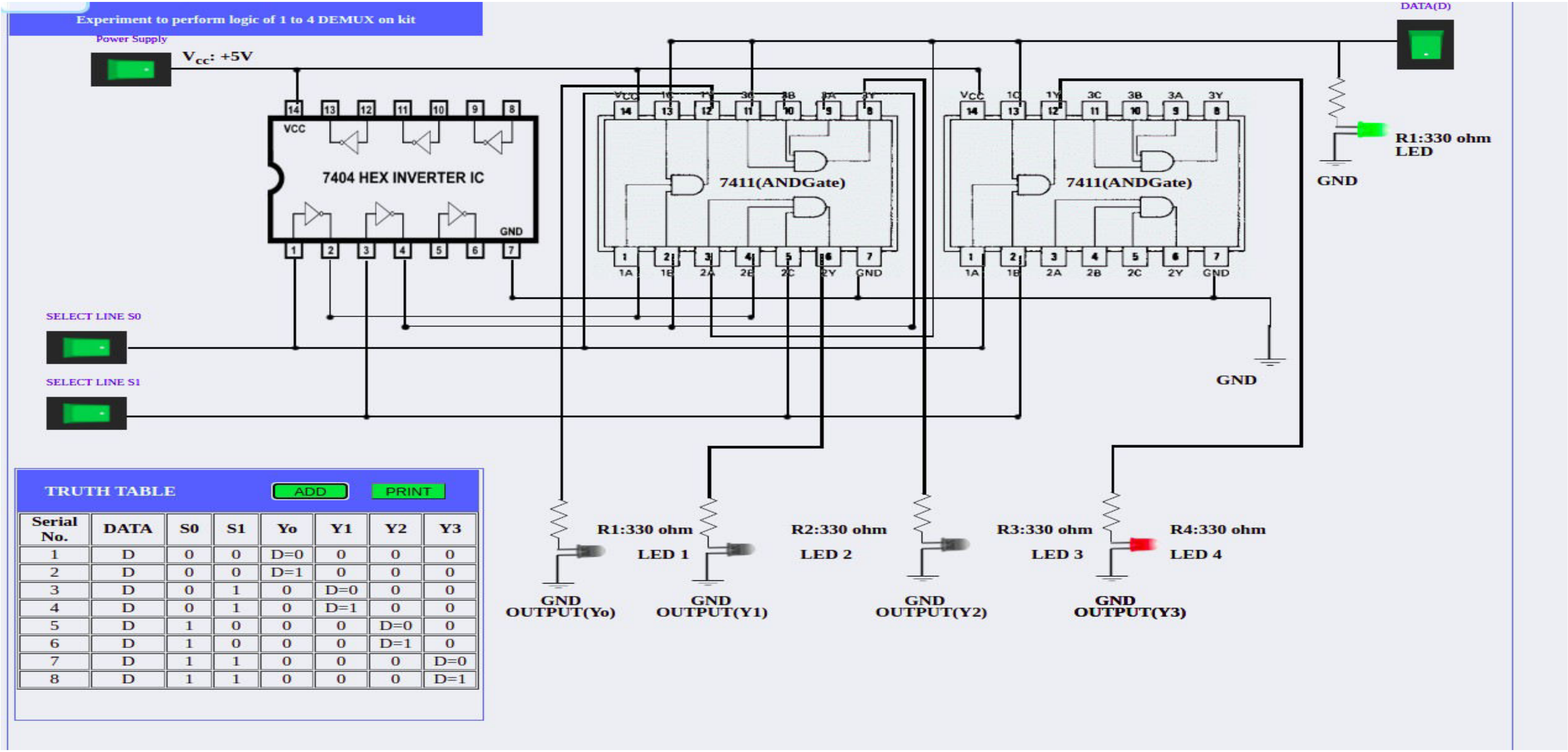


Experiment to perform logic of 4:1 Multiplexer on kit

TRUTH TABLE

Serial No.	S0	S1	OUTPUT (Y)	OUTPUT VALUE
1	0	0	D0	0
2	0	0	D0	1
3	0	1	D1	0
4	0	1	D1	1
5	1	0	D2	0
6	1	0	D2	1
7	1	1	D3	0
8	1	1	D3	1

Simulation(1:4 Demultiplexer)



TRUTH TABLE

Serial No.	DATA	S0	S1	Yo	Y1	Y2	Y3
1	D	0	0	D=0	0	0	0
2	D	0	0	D=1	0	0	0
3	D	0	1	0	D=0	0	0
4	D	0	1	0	D=1	0	0
5	D	1	0	0	0	D=0	0
6	D	1	0	0	0	D=1	0
7	D	1	1	0	0	0	D=0
8	D	1	1	0	0	0	D=1

Discussions and Conclusion

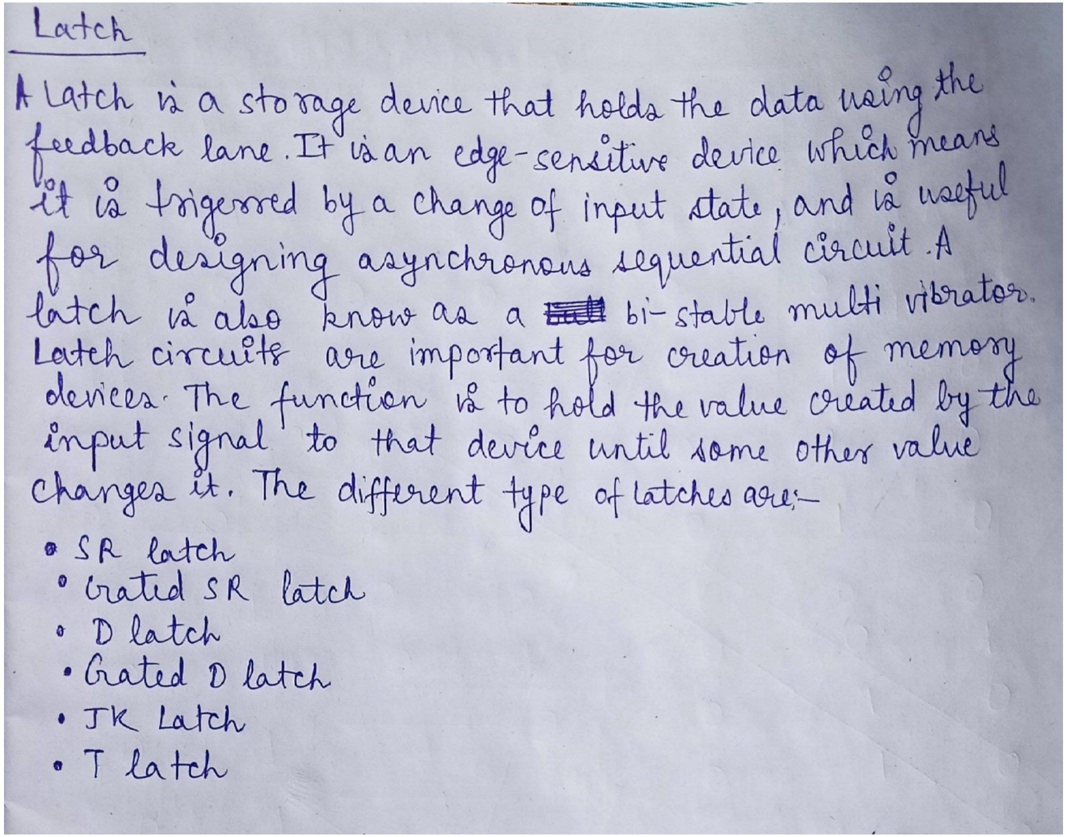
- Multiplexers are used to encode a set of input signals. It performs parallel to serial conversion and are thus known as data selectors.
- Demultiplexers perform the reverse operation, by decoding an input signal. They behave as a data distributor and perform serial to parallel conversion
- Mux helps in increasing transfer of data at a particular time over a network thereby reducing wire usage and complexity fo circuit as we can select which line to choose.
- De-Mux helps in effectively distributing the original signals from a single input.

PART-3

AIM:

- Latches and Construction of SR NOR gate latch.
- Verify the truth table of one bit and two bit comparator using logic gates.

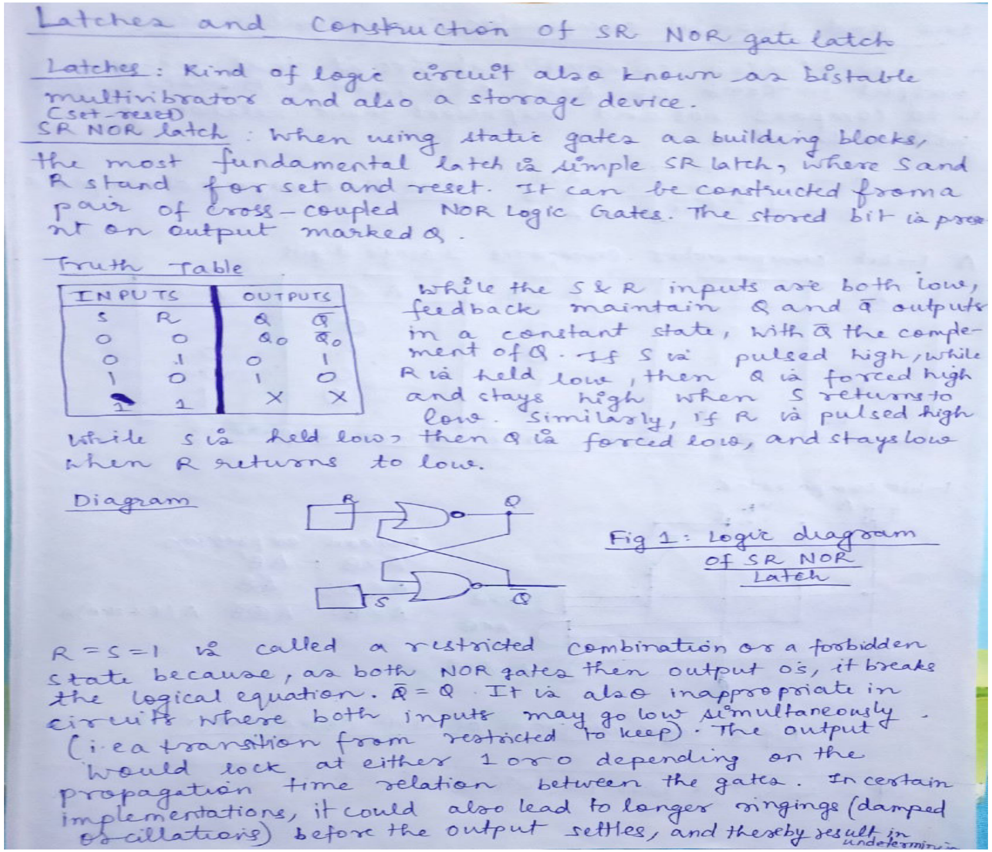
Theory and equations(SR NOR GATE LATCH)



Latch

A Latch is a storage device that holds the data using the feedback lane. It is an edge-sensitive device which means it is trigerred by a change of input state, and is useful for designing asynchronous sequential circuit. A latch is also know as a bi-stable multi vibrator. Latch circuits are important for creation of memory devices. The function is to hold the value created by the input signal to that device until some other value changes it. The different type of latches are:-

- SR latch
- Grated SR latch
- D latch
- Grated D latch
- JK Latch
- T latch



Latches and Construction of SR NOR gate latch

Latches : Kind of logic circuit also known as bistable multivibrator and also a storage device.

(set-reset)
SR NOR latch : When using static gates as building blocks, the most fundamental latch is simple SR latch, where S and R stand for set and reset. It can be constructed from a pair of cross-coupled NOR Logic Gates. The stored bit is present on output marked Q.

Truth Table

INPUTS		OUTPUTS	
S	R	Q	$\bar{Q}$
0	0	Q_0	$\bar{Q}_0$
0	1	0	1
1	0	1	0
1	1	X	X

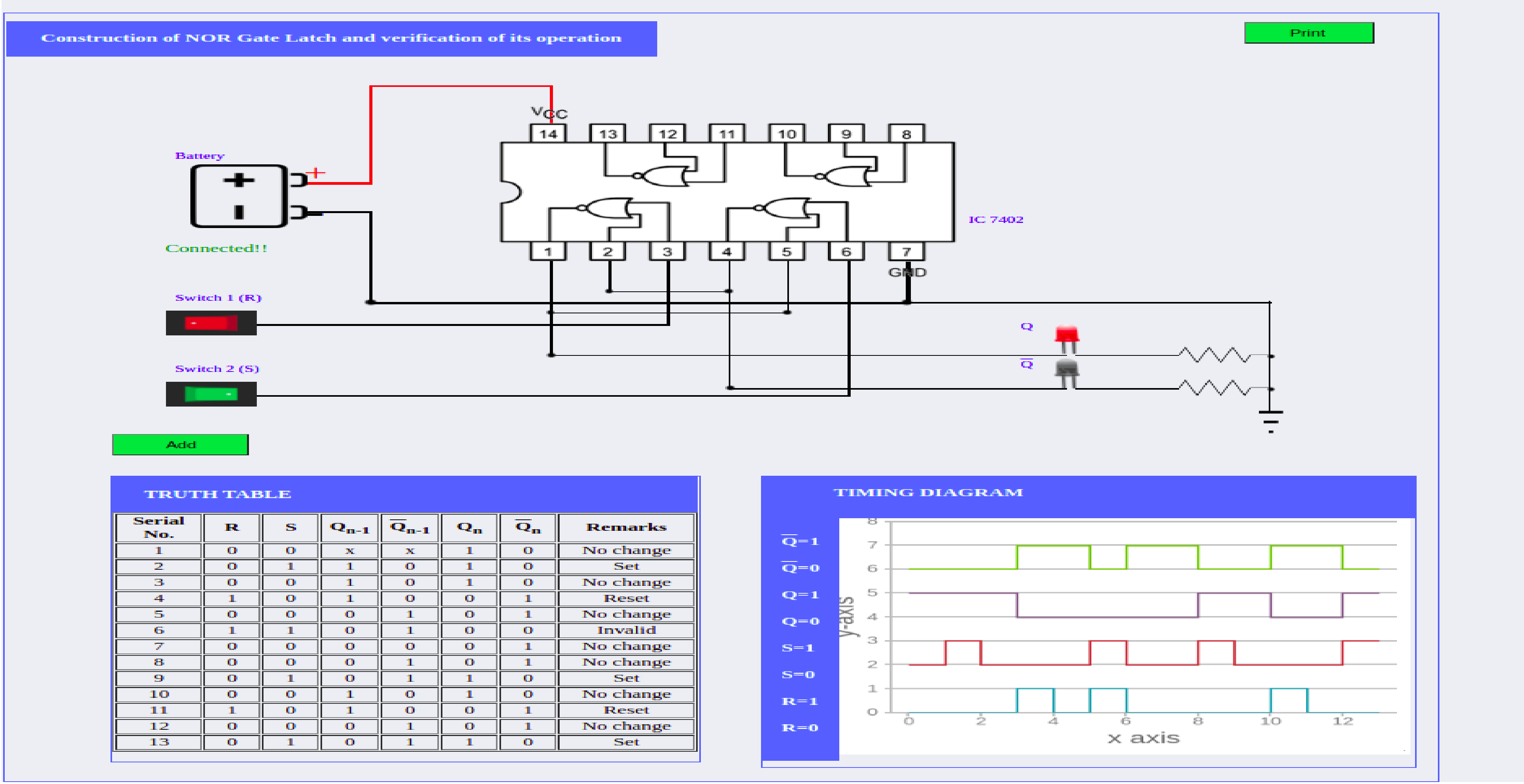
While the S & R inputs are both low, feedback maintain Q and $\bar{Q}$ outputs in a constant state, with $\bar{Q}$ the complement of Q. If S is pulsed high, while R is held low, then Q is forced high and stays high when S returns to low. Similarly, if R is pulsed high while S is held low, then Q is forced low, and stays low when R returns to low.

Diagram

Fig 1: Logic diagram of SR NOR Latch

R = S = 1 is called a restricted combination or a forbidden state because, as both NOR gates then output 0's, it breaks the logical equation. $\bar{Q} = Q$. It is also inappropriate in circuits where both inputs may go low simultaneously. (i.e a transition from restricted to keep). The output would lock at either 1 or 0 depending on the propagation time relation between the gates. In certain implementations, it could also lead to longer ringings (damped oscillations) before the output settles, and thereby result in undetermin

Simulations(SR NOR LATCH) and Truth table verification with timing diagram



Construction of NOR Gate Latch and verification of its operation

TRUTH TABLE

Serial No.	R	S	Q_{n-1}	$\overline{Q}_{n-1}$	Q_n	$\overline{Q}_n$	Remarks
1	0	0	x	x	1	0	No change
2	0	1	1	0	1	0	Set
3	0	0	1	0	1	0	No change
4	1	0	1	0	0	1	Reset
5	0	0	0	1	0	1	No change
6	1	1	0	1	0	0	Invalid
7	0	0	0	0	0	1	No change
8	0	0	0	1	0	1	No change
9	0	1	0	1	1	0	Set
10	0	0	1	0	1	0	No change
11	1	0	1	0	0	1	Reset
12	0	0	0	1	0	1	No change
13	0	1	0	1	1	0	Set

TIMING DIAGRAM

Discussions and Conclusion

- When R and S are both zero, the two outputs are complementary to one another and are maintaine at a constant state. By using one output as the input of another NOR gate, we achieve the condition that the two outputs are complementary to one another. Both R =1and S=1 is inavlid input as then both outputs will be 0 and therefore the relation Q=Q bar cannot be held true.
- They have memory retaining capacity due to which they can withstand fluctuations after being set until a signal is passed and it used to store memory of bits

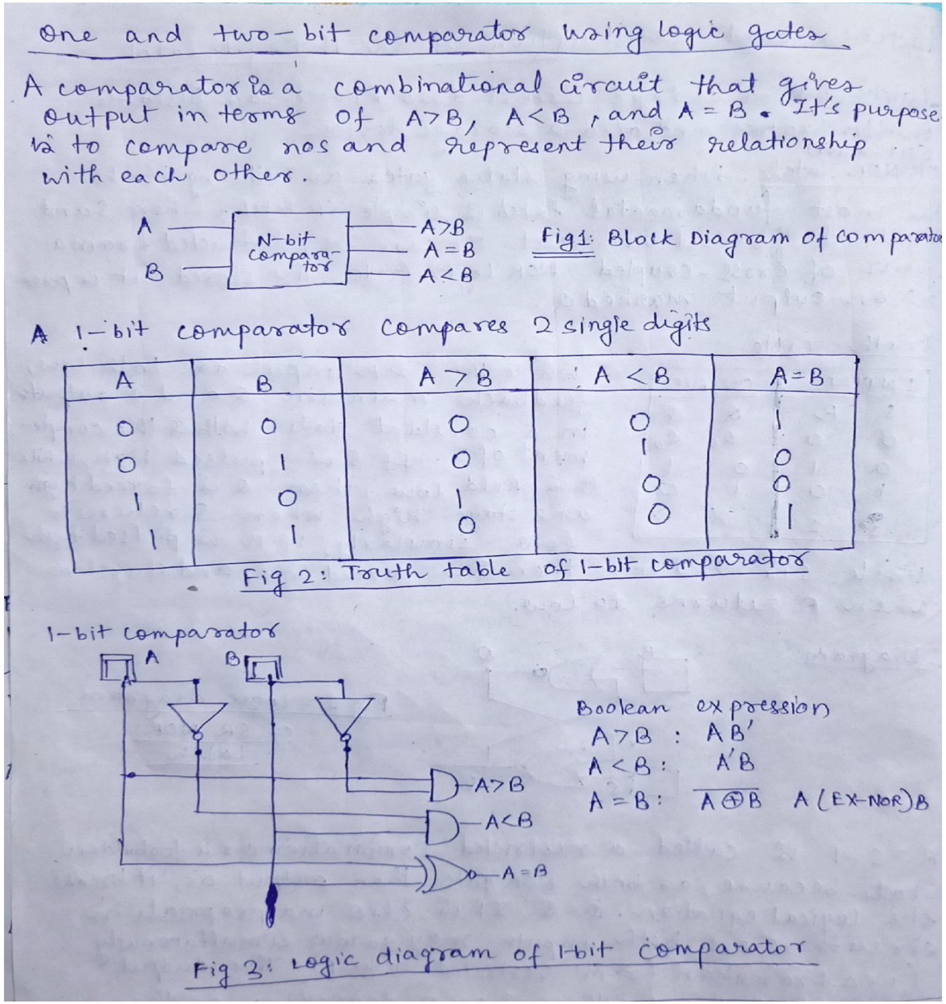
PART-4

AIM:

- To analyse the truth table of 1-bit comparator by using NOT, AND and NOR logic gate ICs
- To analyse 2-bit comparator by using 1-input NOT, 3-input AND, 2-input

  AND, 3-input OR and 2-input Ex-NOR logic gate ICs
- To understand the working of 1-bit comparator and 2- bit comparator with the help of LEDs display.

Theory and equations(1-bit and 2-bit comparator)



One and two-bit comparator using logic gates

A comparator is a combinational circuit that gives output in terms of A>B, A<B, and A = B. It's purpose is to compare nos and represent their relationship with each other.

A
B
N-bit Compara-tor
A>B
A=B
A<B

Fig1: Block Diagram of comparator

A 1-bit comparator compares 2 single digits

A	B	A > B	A < B	A = B
0	0	0	0	1
0	1	0	1	0
1	0	1	0	0
1	1	0	0	1

Fig 2: Truth table of 1-bit comparator

1-bit comparator

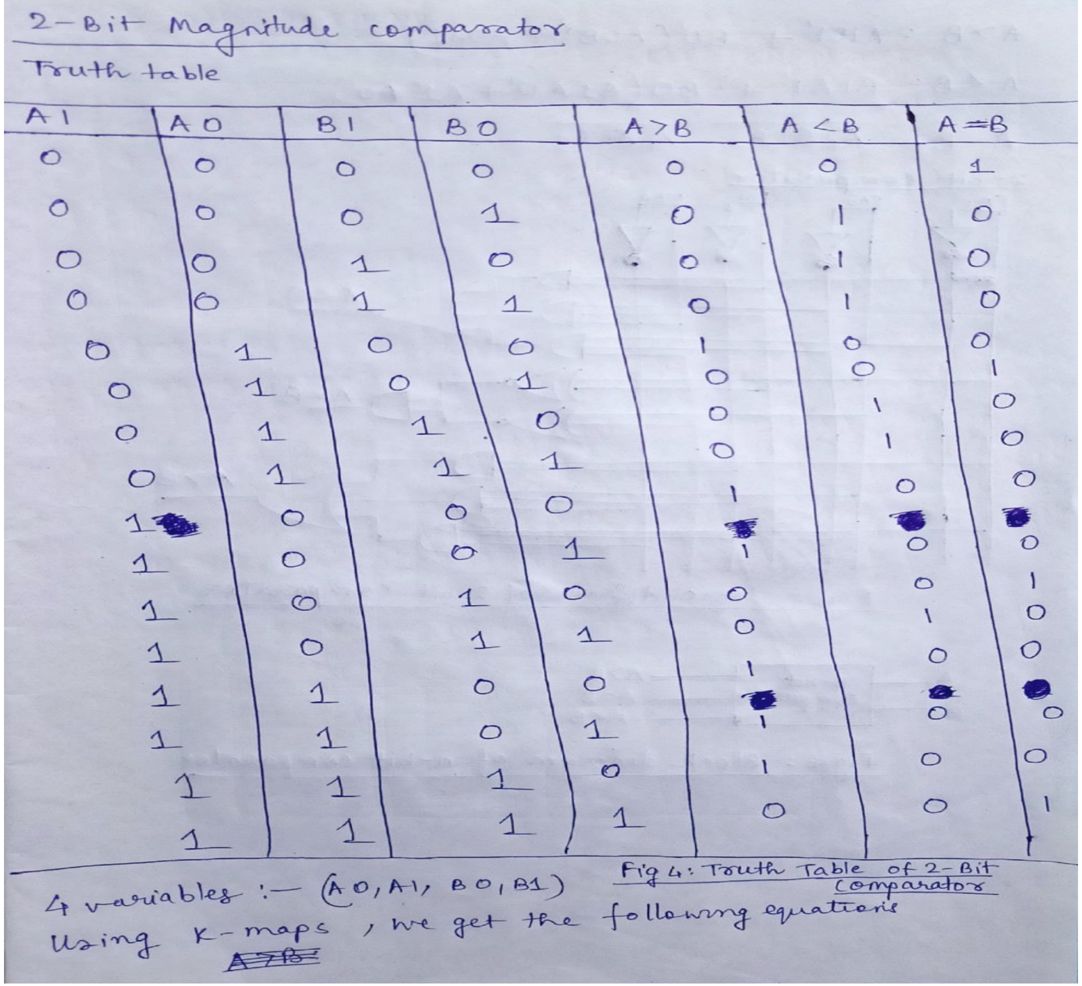
Boolean expression

A>B : AB'

A<B : A'B

A = B : $\overline{A \oplus B}$ A (EX-NOR) B

Fig 3: Logic diagram of 1-bit comparator



2-Bit Magnitude comparator

Truth table

A1	A0	B1	B0	A>B	A<B	A=B
0	0	0	0	0	0	1
0	0	0	1	0	1	0
0	0	1	0	0	1	0
0	0	1	1	0	1	0
0	1	0	0	1	0	0
0	1	0	1	0	0	1
0	1	1	0	0	1	0
0	1	1	1	0	1	0
1	0	0	0	1	0	0
1	0	0	1	1	0	0
1	0	1	0	0	0	1
1	0	1	1	0	1	0
1	1	0	0	1	0	0
1	1	0	1	1	0	0
1	1	1	0	1	0	0
1	1	1	1	0	0	1

Fig 4: Truth Table of 2-Bit comparator

4 variables :- (A0, A1, B0, B1)

Using K-maps , we get the following equations

~~A>B~~

$A > B$: $A1B1' + B0'(A0B1' + A0A1)$

$A < B$: $B1A1' + B0(B1A0' + A1'A0')$

$A = B$: $\overline{A_0 \oplus B_0} \cdot \overline{A_1 \oplus B_1}$

2-bit comparator

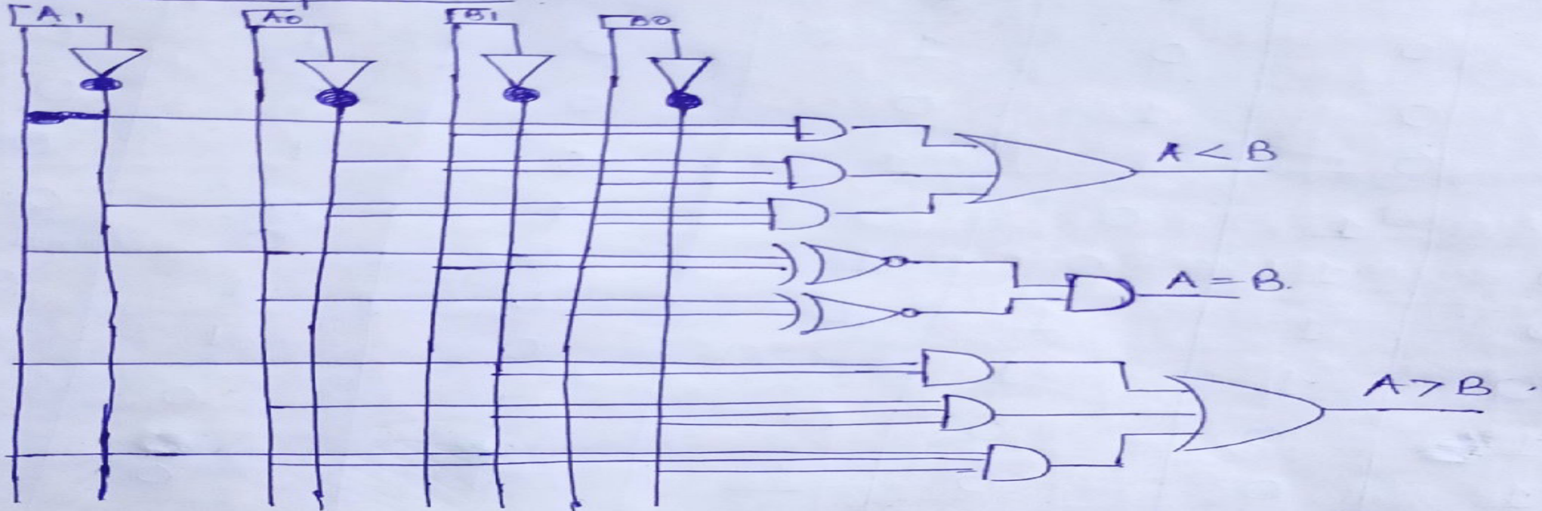


Fig 5: Logic digram of 2-bit comparator

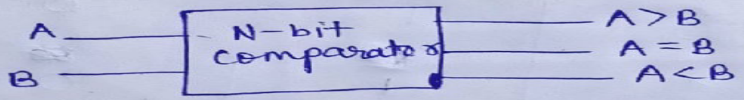
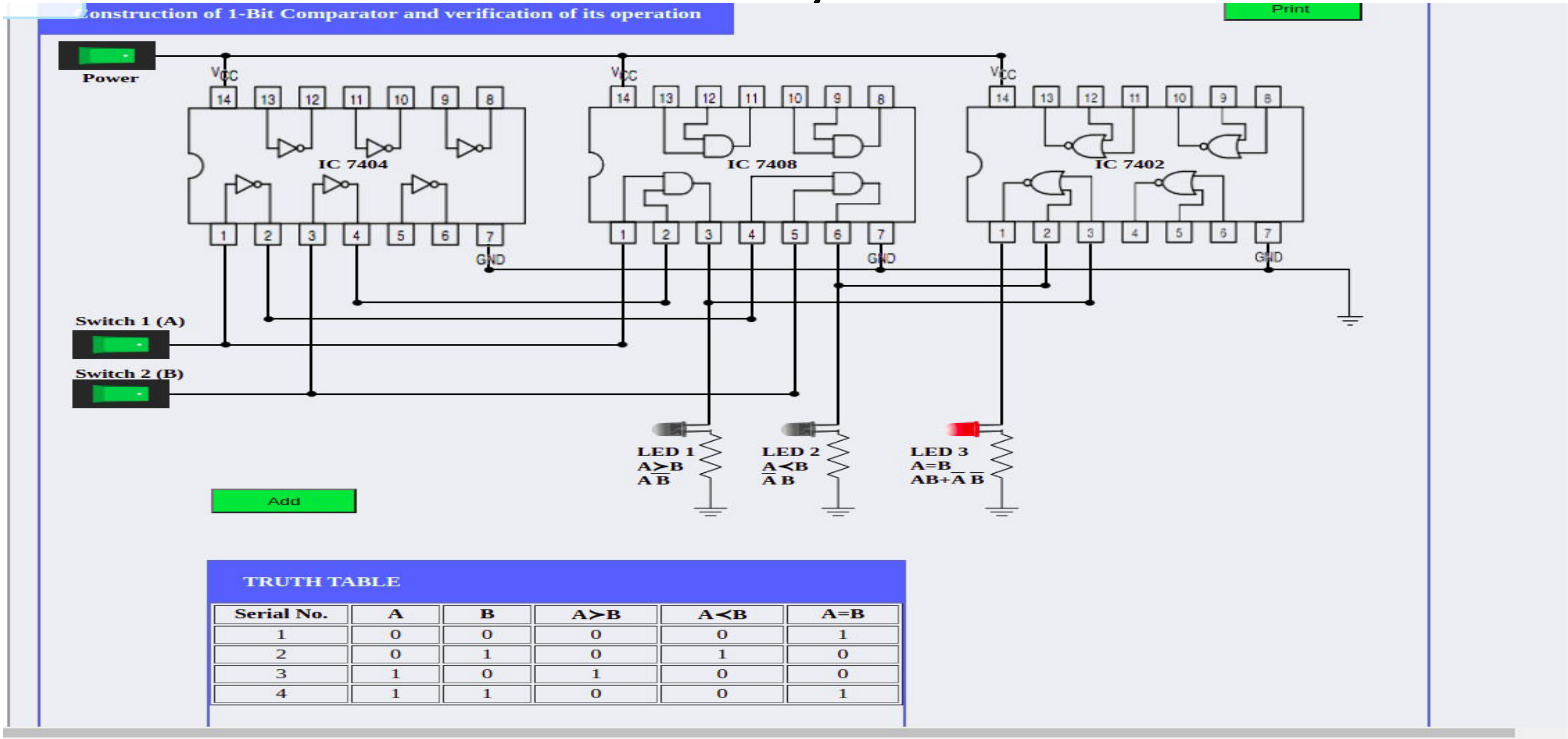


Fig 6: Block diagram of n-bit comparator

Simulation(1-bit comparator circuit with ic’s and led)

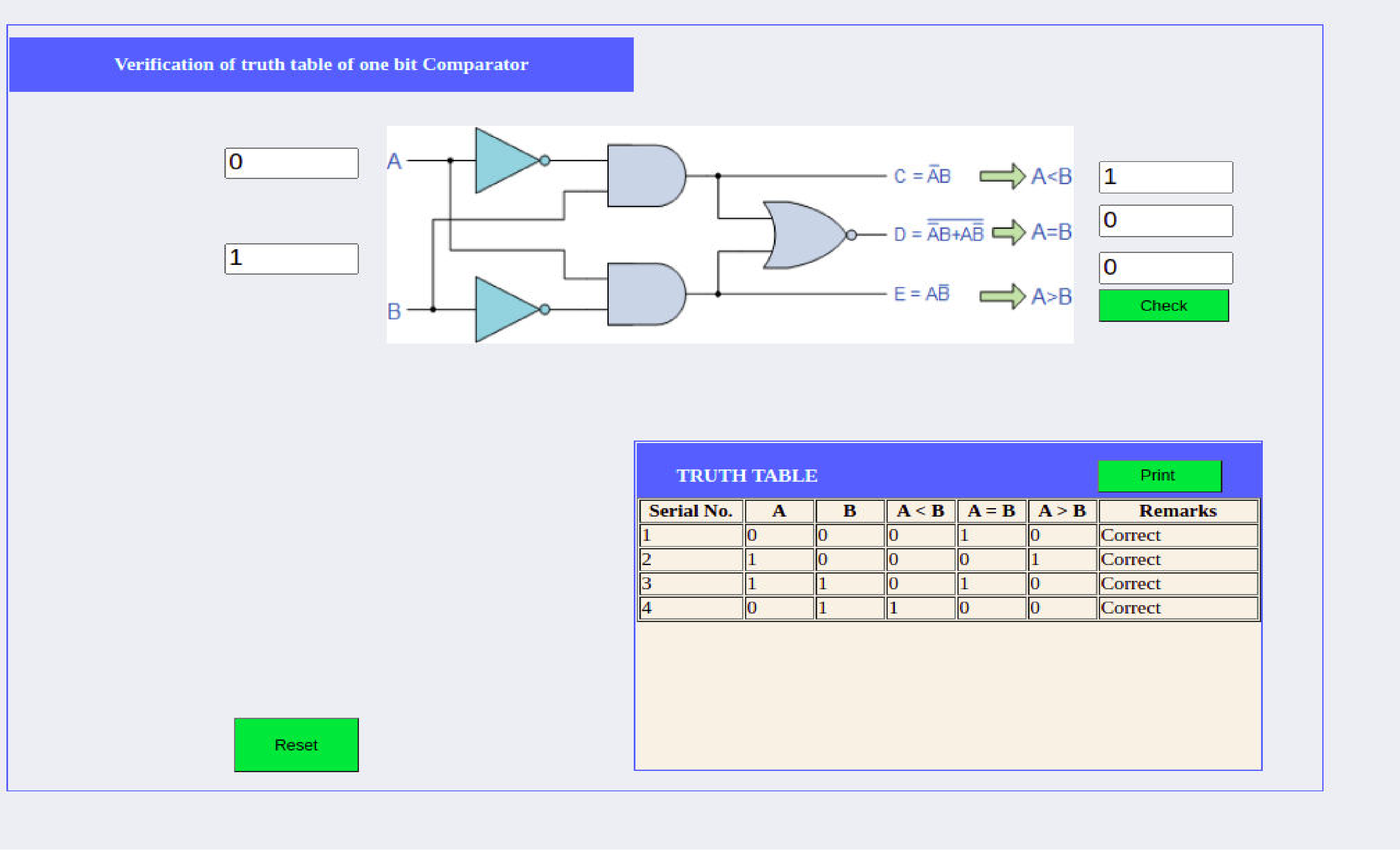


onstruction of 1-Bit Comparator and verification of its operation

TRUTH TABLE

Serial No.	A	B	A>B	A<B	A=B
1	0	0	0	0	1
2	0	1	0	1	0
3	1	0	1	0	0
4	1	1	0	0	1

Verification of truth table of 1-bit comparator



Verification of truth table of one bit Comparator

0

1

1

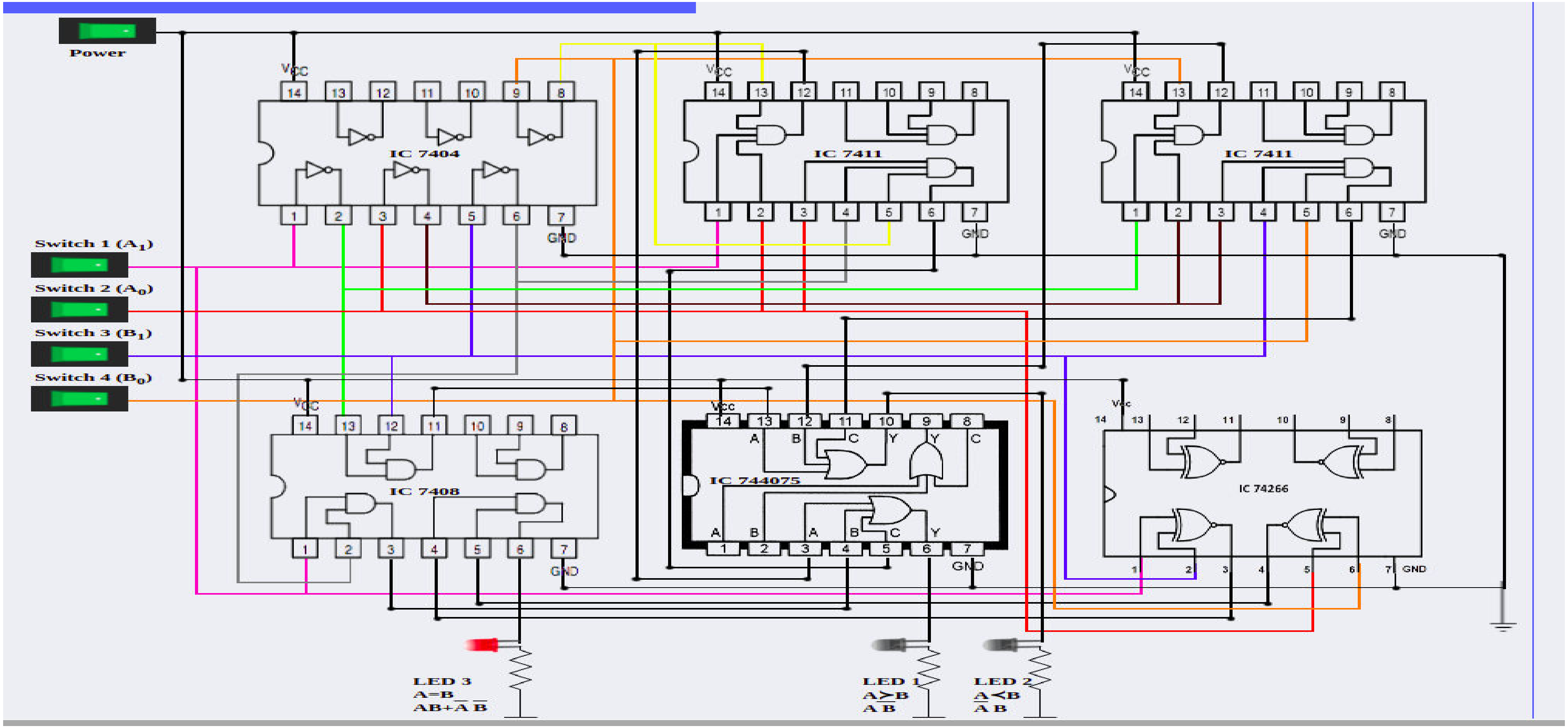
0

0

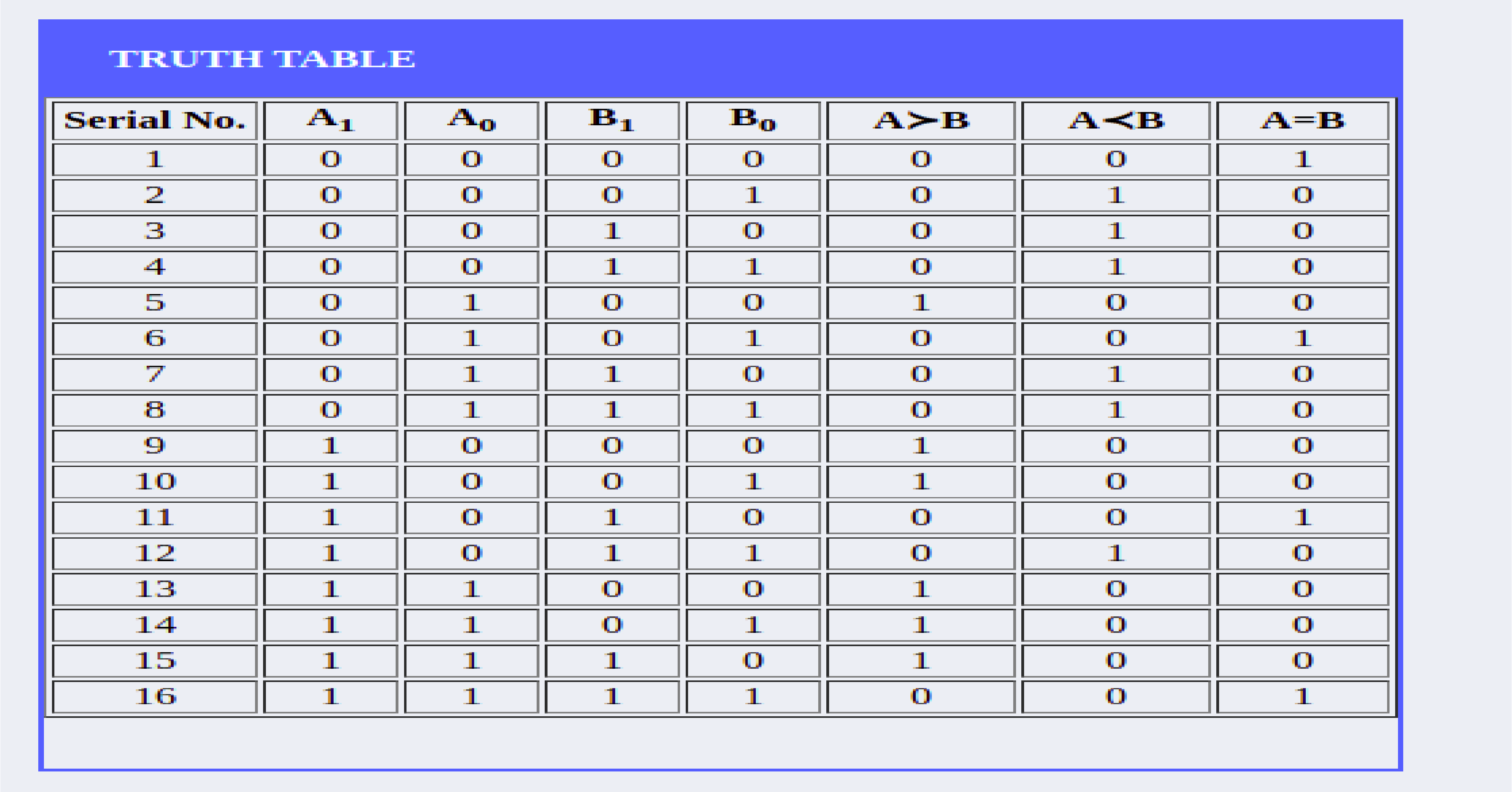
TRUTH TABLE

Serial No.	A	B	A < B	A = B	A > B	Remarks
1	0	0	0	1	0	Correct
2	1	0	0	0	1	Correct
3	1	1	0	1	0	Correct
4	0	1	1	0	0	Correct

Simulation(2-bit comparator circuit with ic’s and led)



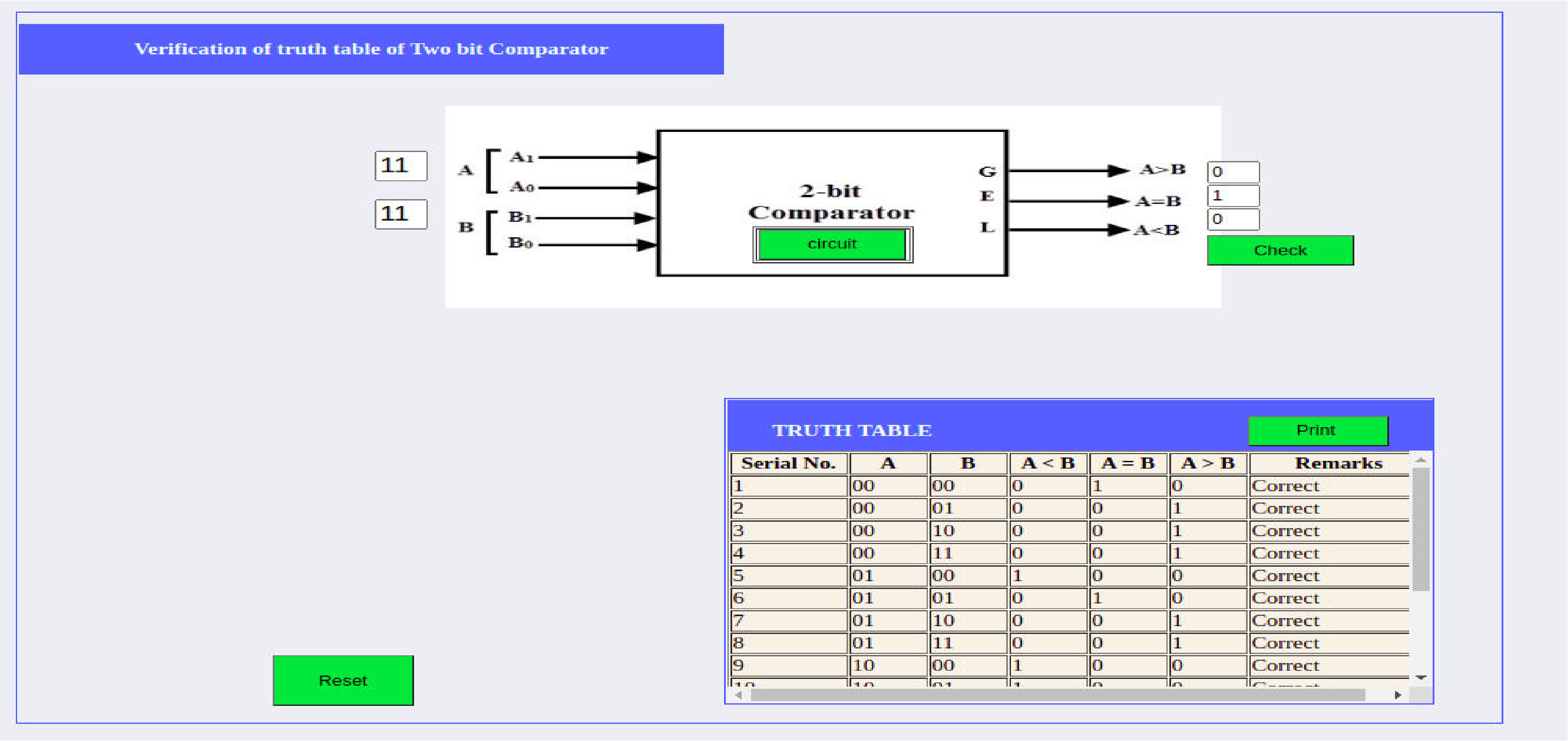
Truth table of 2-bit comparator



TRUTH TABLE

Serial No.	A_1	A_0	B_1	B_0	$A>B$	$A<B$	$A=B$
1	0	0	0	0	0	0	1
2	0	0	0	1	0	1	0
3	0	0	1	0	0	1	0
4	0	0	1	1	0	1	0
5	0	1	0	0	1	0	0
6	0	1	0	1	0	0	1
7	0	1	1	0	0	1	0
8	0	1	1	1	0	1	0
9	1	0	0	0	1	0	0
10	1	0	0	1	1	0	0
11	1	0	1	0	0	0	1
12	1	0	1	1	0	1	0
13	1	1	0	0	1	0	0
14	1	1	0	1	1	0	0
15	1	1	1	0	1	0	0
16	1	1	1	1	0	0	1

Verification of truth table of 2-bit comparator



Verification of truth table of Two bit Comparator

11

11

A>B: 0

A=B: 1

A<B: 0

TRUTH TABLE

Serial No.	A	B	A < B	A = B	A > B	Remarks
1	00	00	0	1	0	Correct
2	00	01	0	0	1	Correct
3	00	10	0	0	1	Correct
4	00	11	0	0	1	Correct
5	01	00	1	0	0	Correct
6	01	01	0	1	0	Correct
7	01	10	0	0	1	Correct
8	01	11	0	0	1	Correct
9	10	00	1	0	0	Correct
10	10	01	1	0	0	Correct

Discussions and Conclusion

- We can perform multiple bit comparison in this way by comparing bit by bit from most significant bit (msb) to least significant bit (lsb).
- We can use K-map to derive boolean expression for n-bit comparator from truth table.
- Comparators are widely used for the following :-

  In biometric applications or verifying passwords

  In control applications, where physical variables like temperature, potential, position, etc are compared with respect to a reference value and represented by binary numbers.