B.TECH. (2020-24)
Artificial Intelligence

Lab File

(Experiment 1 & 2)

on

DIGITAL ELECTRONICS AND COMPUTER ORGANIZATION

[CSE207]

Submitted To

Ms Neha Arora

Submitted By

HITESH

A023119820027

4CSE11 (AI)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
AMITY SCHOOL OF ENGINEERING AND TECHNOLOGY
AMITY UNIVERSITY UTTAR PRADESH
NOIDA (U.P)

INDEX

S. No.	Experiment Number	Date of Allotment of Experiment	Signature of Faculty
1	Exp 1 To verify and interpret the logic and truth table for AND, OR, NOT, NAND, NOR, Ex-OR, Ex-NOR gates.	07/01/2022	
2	Exp 2 To study and design the single bit half adder and full adder combinational circuits by using (1) XOR and (2) NAND gates respectively to verify and interpret their logic and truth tables and analyse the working of half adder and full adder circuit.	14/01/2022	
3			
4			
5			
6			
7			
8			
9			
10			

EXPERIMENT – 1

LOGIC GATES

EXPERIMENT – 1

AIM

To verify and interpret the logic and truth table for AND, OR, NOT, NAND, NOR, Ex-OR, Ex-NOR gates.

PLATFORM/TOOL USED

- Simulator: https://circuitverse.org/simulator
- Virtual LAB: https://de-iitr.vlabs.ac.in/exp/realization-of-logic-functions/

THEORY

Logic gates are electronic circuits which perform logical functions on one or more inputs to produce one output.

They 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, there are seven logic gates –

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

When all the input combinations of a logic gate are written in a series and their corresponding outputs written along them, then this input/ output combination is called **Truth Table**.

The **Karnaugh Map** (**K-Map**) method is a systematic way of simplifying Boolean expressions. With the help of the K-map method, we can find the simplest POS and SOP expression, which is known as the minimum expression. The K-map provides a cookbook for simplification.

Just like the truth table, a K-map contains all the possible values of input variables and their corresponding output values. However, in K-map, the values are stored in cells of the array. In each cell, a binary value of each input variable is stored.

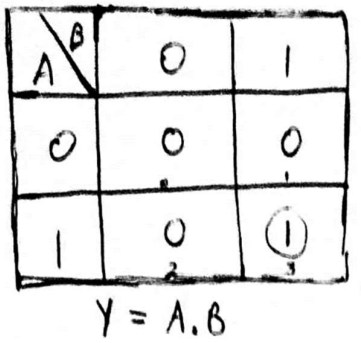
1) AND Gate

- The AND gate is an electronic circuit that gives a high output (1) only if all its inputs are high. A dot (.) is used to show the AND operation i.e. A.B or can be written as AB.
- The IC number of AND Gate is 7408.

Truth Table of AND Gate

INPUT		OUTPUT
A	**B**	**Y = A.B**
0	0	0
0	1	0
1	0	0
1	1	1

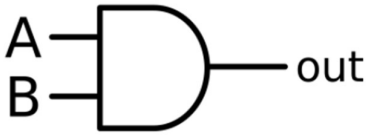
K- Map of AND Gate



Boolean Expression of AND Gate

Y = A.B

Logic Symbol of AND Gate



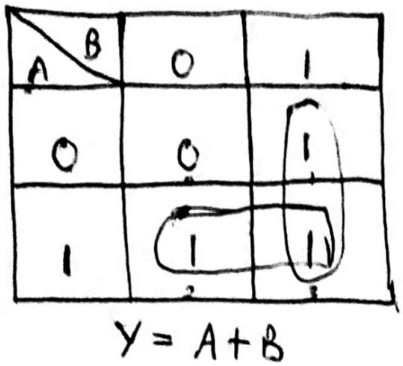
2) OR Gate

- The OR gate is an electronic circuit that gives a high output (1) if one or more of its inputs are high. A plus (+) is used to show the OR operation.
- The IC number of OR Gate is 7432.

Truth Table of OR Gate

INPUT		OUTPUT
A	**B**	**Y = A+B**
0	0	0
0	1	1
1	0	1
1	1	1

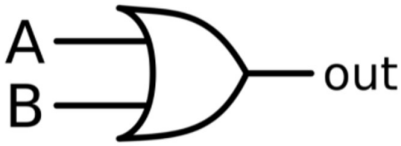
K- Map of OR Gate



Boolean Expression of OR Gate

Y = A+B

Logic Symbol of OR Gate



3) NOT Gate

- The NOT gate is an electronic circuit that produces an inverted version of the input at its output. It is also known as an inverter. If the input variable is A, the inverted output is known as NOT A.
- The IC number of NOT Gate is 7404.

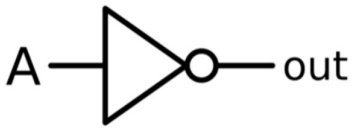
Truth Table of NOT Gate

INPUT	OUTPUT
A	**Y = $\overline{A}$**
0	1
1	0

Boolean Expression of NOT Gate

Y = $\overline{A}$

Logic Symbol of NOT Gate



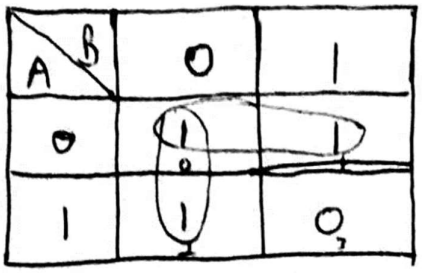
4) NAND Gate

- This is a 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 represents inversion.
- The IC number of NAND Gate is 7400.

Truth Table of NAND Gate

INPUT		OUTPUT
A	**B**	**Y = $\overline{AB}$**
0	0	1
0	1	1
1	0	1
1	1	0

K- Map of NAND Gate



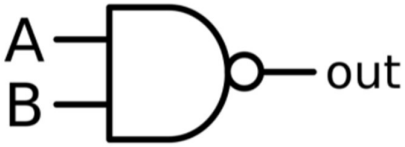
$$Y = \bar{A} + \bar{B}$$

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

Boolean Expression of NAND Gate

Y = $\overline{AB}$

Logic Symbol of NAND Gate



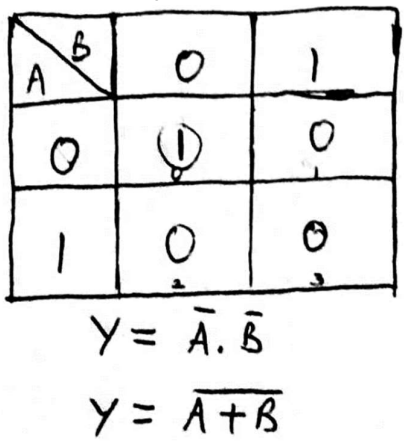
5) NOR Gate

- This is a 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 a small circle on the output. The small circle represents inversion.
- The IC number of NOR Gate is 7402.

Truth Table of NOR Gate

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

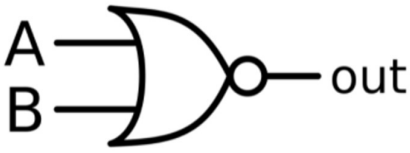
K- Map of NOR Gate



Boolean Expression of NOR Gate

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

Logic Symbol of NOR Gate



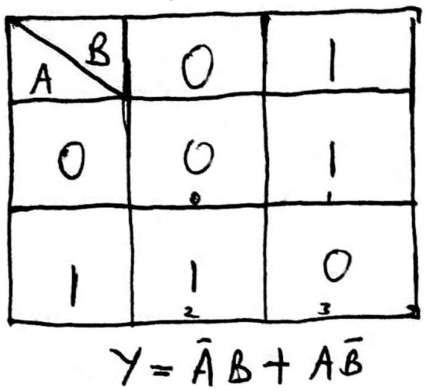
6) Ex-OR Gate

- The 'Exclusive-OR' gate is a circuit which will give a high output if either, but not both of its two inputs are high. An encircled plus sign ($\oplus$) is used to show the Ex-OR operation.
- The IC number of X-OR Gate is 7486.

Truth Table of XOR Gate

INPUT		OUTPUT
A	**B**	**Y= A$\oplus$B**
0	0	0
0	1	1
1	0	1
1	1	0

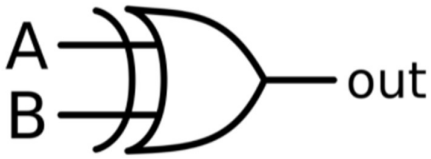
K- Map of XOR Gate



Boolean Expression of XOR Gate

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

Logic Symbol of XOR Gate



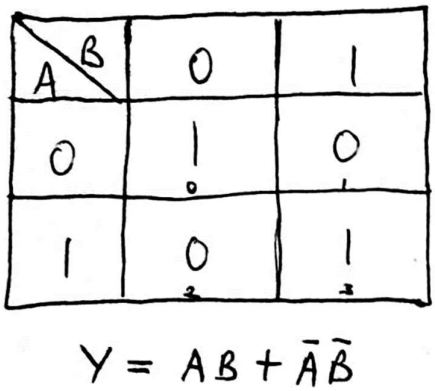
7) Ex-NOR Gate

- The 'Exclusive-NOR' gate circuit does the opposite to the EX-OR gate. It will give a low output if either, but not both of its two inputs are high. The symbol is an EX-OR gate with a small circle on the output. The small circle represents inversion.
- The IC number of Ex-NOR Gate as CMOS IC is the 4077, and the TTL IC is the 74266.

Truth Table of XOR Gate

INPUT		OUTPUT
A	B	$Y = \overline{A \oplus B}$
0	0	1
0	1	0
1	0	0
1	1	1

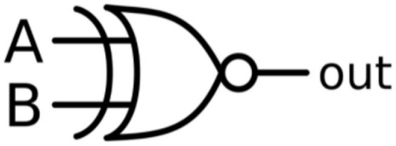
K- Map of XOR Gate



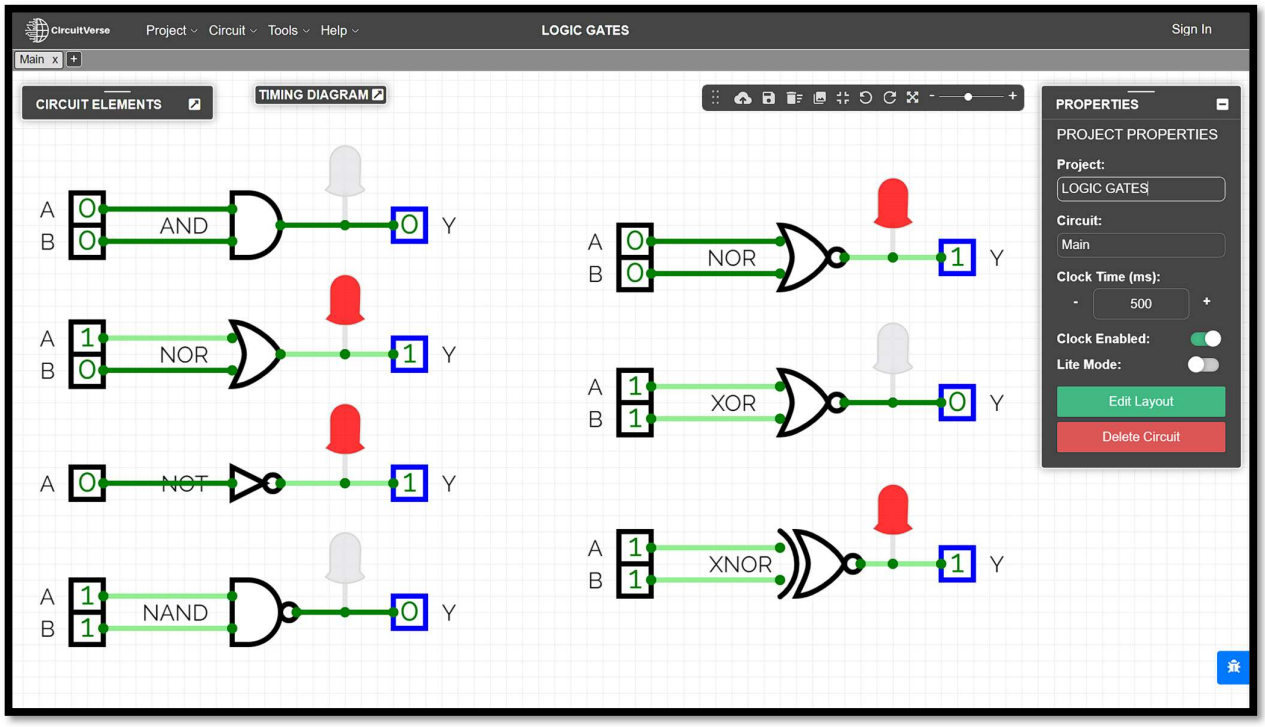
Boolean Expression of XOR Gate

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

Logic Symbol of XOR Gate



SIMULATED OUTPUT



RESULT

The Boolean Expressions and Truth Tables of Logic and Universal gates have been studied and verified using the online simulator.

CRITERIA	TOTAL MARKS	MARKS OBTAINED	COMMENTS
Concept (A)	2		
Implementation (B)	2		
Performance (C)	2		
Total	6		

EXPERIMENT – 2

COMBINATIONAL CIRCUITS:

HALF ADDER & FULL ADDER

EXPERIMENT – 2

AIM

To study and design the single bit half adder and full adder combinational circuits by using (1) XOR and (2) NAND gates respectively to verify and interpret their logic and truth tables and analyse the working of half adder and full adder circuit.

PLATFORM/TOOL USED

- Simulator: https://circuitverse.org/simulator
- Virtual LAB: https://de-iitr.vlabs.ac.in/exp/half-full-adder/

THEORY

An **Adder** is a digital logic circuit in electronics that is extensively used for the addition of numbers. In many computers and other types of processors, adders are even used to calculate addresses and related activities and calculate table indices in the ALU and even utilized in other parts of the processors.

Adders are a key component of arithmetic logic unit. Adders can be constructed for most of the numerical representations like Binary Coded Decimal (BCD), Excess – 3, Gray code, Binary etc. out of these, binary addition is the most frequently performed task by most common adders. Apart from addition, adders are also used in certain digital applications like table index calculation, address decoding etc.

Binary addition is like that of decimal addition. Some basic binary additions are shown below.

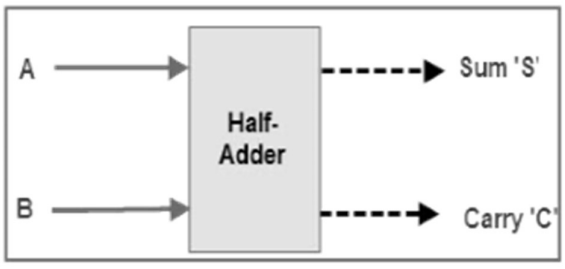
0	0	1	1
+0	+1	+0	+1
0	1	1	(carry) 1 0

Adders are basically classified into two types: Half Adder and Full Adder.

1) Half Adder

- Half adder is a combinational circuit that performs simple addition of two binary bits.
- It has 2 inputs that represent the 2 bits to be added and 2 outputs, with one producing the SUM output and the other producing the CARRY.
- 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:

Block Diagram for HALF ADDER



Truth Table for HALF ADDER

INPUT		OUTPUT	
A	**B**	**Carry (C)**	**Sum (S)**
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

K-Map for HALF ADDER

For SUM

A \ B	0	1
0	0	(1)
1	(1)	0

$Sum = AB' + A'B$

$Sum = \mathrm{A} \oplus \mathrm{B}$

For CARRY

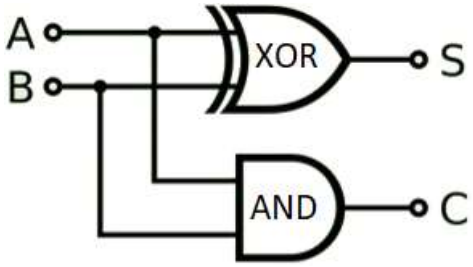
A \ B	0	1
0	0	0
1	0	(1)

$Carry = AB$

Boolean Expression for HALF ADDER

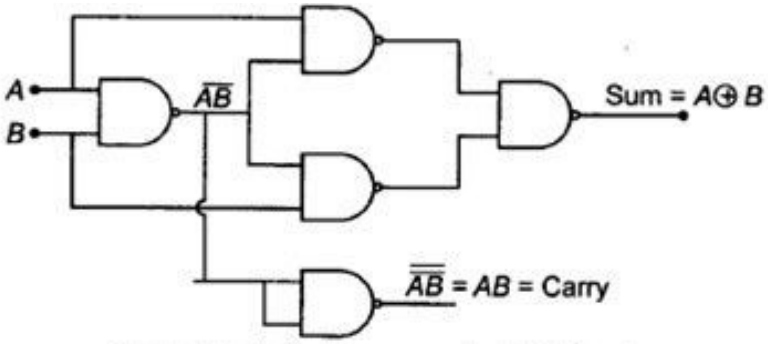
$\boxed{Sum = AB' + A'B = \mathrm{A} \oplus \mathrm{B}}$ $\boxed{Carry = AB}$

Logic Diagram for HALF ADDER



- Five NAND gates are required to design a half adder.

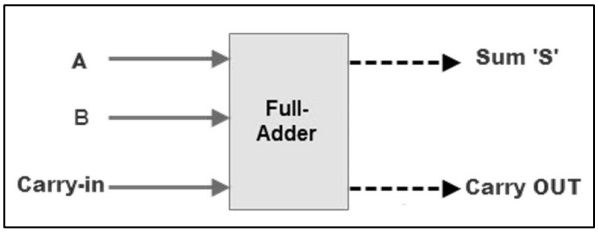
Realization of Half Adder using 5 NAND gates



2) Full Adder

- Full adder is a digital combinational circuit used to calculate the sum of three binary bits. Full adders are complex and difficult to implement when compared to half adders.
- It consists of 3 inputs and 2 outputs.
- The full adder overcomes the limitation of the half adder, which can be used to add two bits only.
- 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 C_{IN}. It calculates the sum of three bits along with the carry. The output carry is called 'Carry-out' and is represented by C_{OUT}.
- As there 3 input variables, 8 different output combinations are possible.
- The block diagram and truth table of a full adder with A, B and C_{IN} as inputs and S, Carry OUT as outputs can be tabulated as follows:

Block Diagram for FULL ADDER



Truth Table for FULL ADDER

INPUT			OUTPUT	
A	**B**	**C_{IN}**	**SUM**	**C_{OUT}**
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

K-Map for FULL ADDER

For SUM

A \ BC_{IN}	00	01	11	10
0	0	(1)	0	(1)
1	(1)	0	(1)	0

$Sum = A'B'C_{in} + A'BC'_{in} + AB'C'_{in} + ABC_{in}$

$\text{Sum} = \text{A} \oplus \text{B} \oplus \text{C}_{IN}$

For $Carry_{out}$

A \ BC_{IN}	00	01	11	10
0	0	0	1	0
1	0	1	1	1

$C_{out} = AB + AC_{in} + AC_{in}$

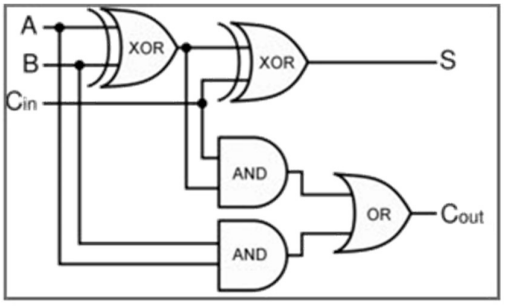
$C_{out} = AB + C_{in}(A \oplus B)$

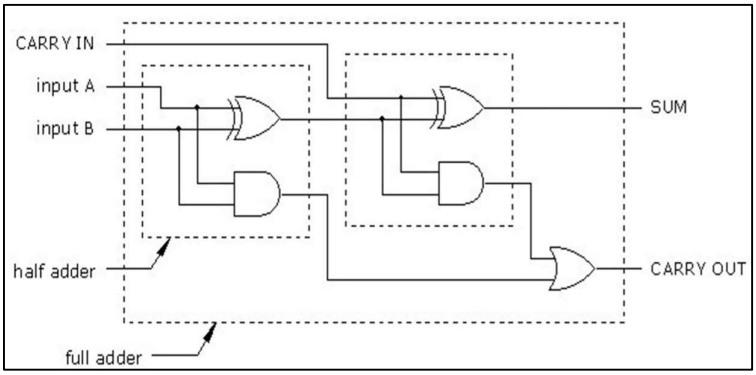
Boolean Expression for HALF ADDER

$$\boxed{Sum = A'B'C_{in} + A'BC'_{in} + AB'C'_{in} + ABC_{in} = \text{A} \oplus \text{B} \oplus \text{C}_{IN}}$$

$$\boxed{C_{out} = AB + AC_{in} + AC_{in} = AB + C_{in}(A \oplus B)}$$

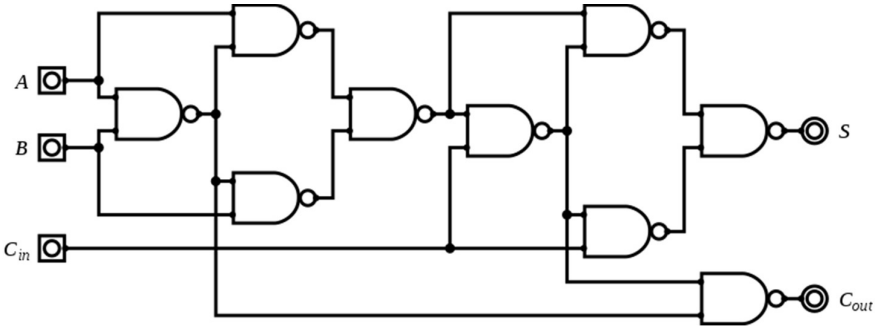
Logic Diagram for FULL ADDER



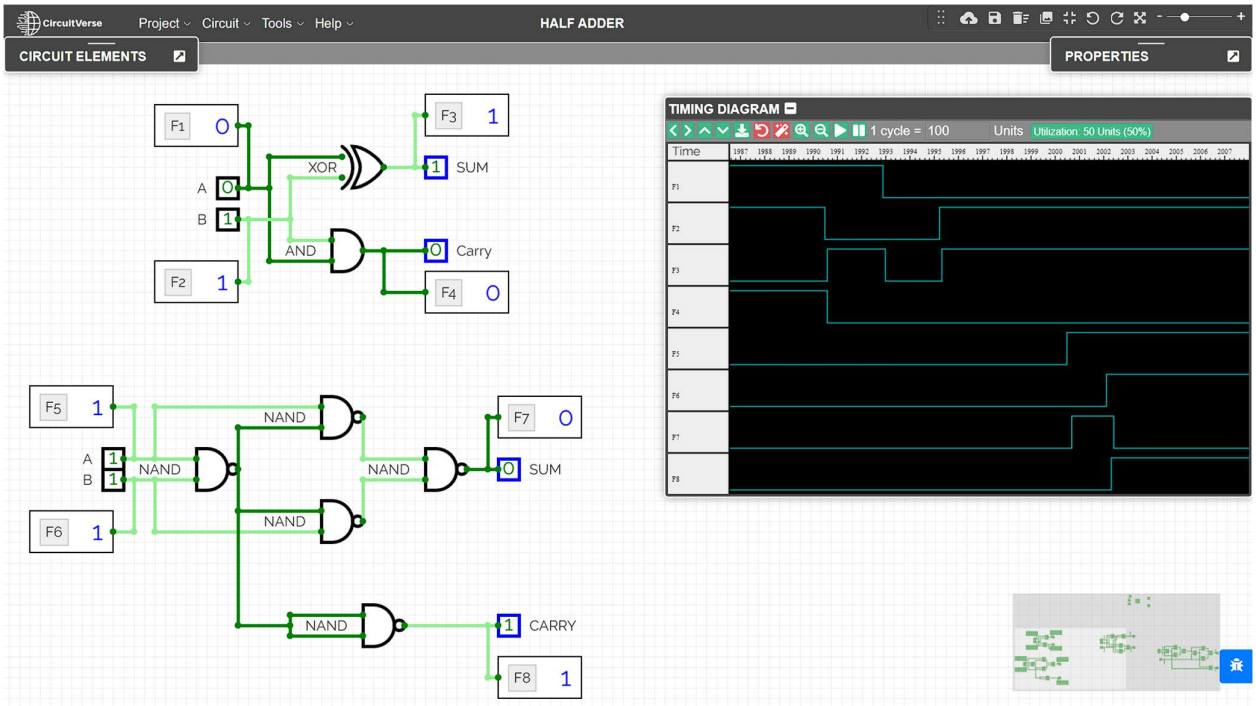


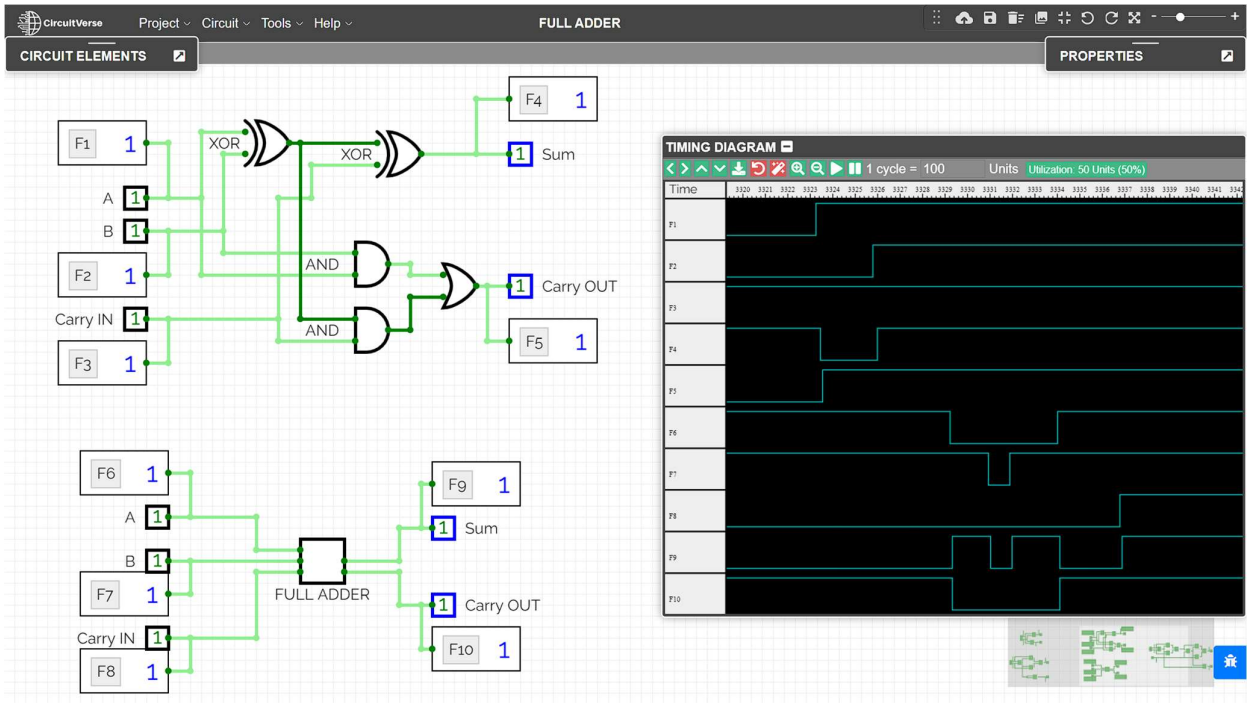
- Nine NAND gates are required to design a full adder.

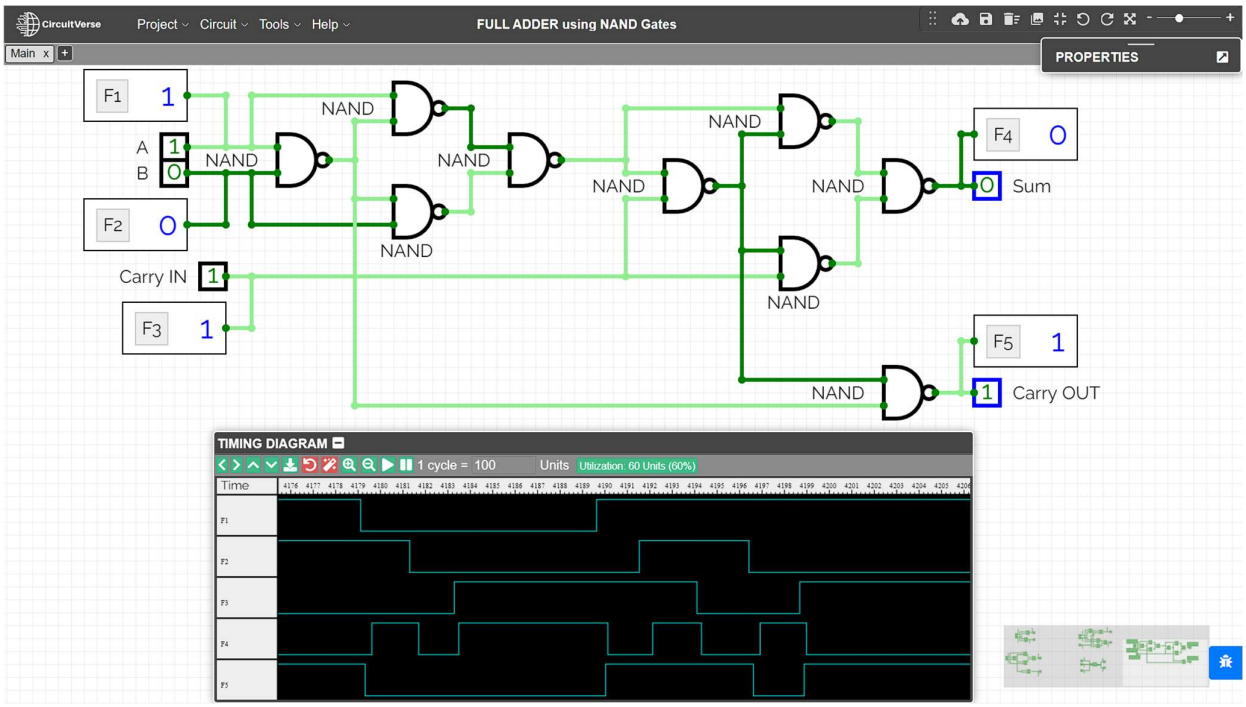
Realization of FULL Adder using 9 NAND gates



SIMULATED OUTPUT







RESULT

The Working of Half adder and Full adder combinational circuit have been analysed and the Truth Tables and Boolean Expressions of half adder and full adder using XOR and NAND gates have been studied and verified using the online simulator.

CRITERIA	TOTAL MARKS	MARKS OBTAINED	COMMENTS
Concept (A)	2		
Implementation (B)	2		
Performance (C)	2		
Total	6		