United International University

Department of CSE

Course Code: CSE 1326

Course Name: Digital Logic Design Laboratory

Experiment no: 03

Experiment Name: Implementing Full Adder

Submitted by:

Name: Talha Jubayer

Student ID: 0112410062

Section: A

Date of Performance: 20-11-2024

Date of Submission: 24-11-2024

Objective:

The objective of this lab is to sum 2 numbers using Digital logic and circuits and learn how a circuit compute summation using just binary digits and binary logic gates and full bit adder. Implementation of full bit adder and learning how this addition works is the main target of this experiment. Analyzing given problem and finding its simplified equation contains a huge importance to implement in trainer board and understand the process.

Apparatus/Instruments used in the Lab:

The components used in this experiment are

1. ICs Used:

- 74LS08 (AND Gate)
- 74LS32 (OR Gate)
- 74LS86 (XOR Gate)
- 7483 (4-bit binary full adder)

2. Logisim
3. Trainer board
4. Multimeter
5. Jumper wire

Theory:

Digital computers perform a variety of information-processing tasks. Among the basic functions encountered are the various arithmetic operations. The most basic and simplest addition is adding two single-digit binary numbers. There are four possible outcomes:

- 1+ 0 = 0 0 + 1 = 1

- 1 + 0 = 1 1 + 1 = 10

The first three operations produce a sum whose length is one digit, but when both augend and addend bits are equal to 1, the binary sum consists of two digits. Here, the "10" means 1 carry and 0 in the ones place. The higher significant bit of this result is called a carry. When the augend and addend numbers contain more significant digits, this carry-over is added to the next pair of digits. A circuit that performs the addition of two single-digit binary is called a half-adder. One that performs the addition of three bits (two single-digit binary and a previous carry) is a full-adder. The name of the former stems from the fact that two half-adders can be employed to implement a full-adder. The two adder circuits are the easiest and simplest to implement and work on.

Half Adder:

From the theory of a half-adder, we got to know that this circuit needs two binary inputs and will have two binary outputs. The input variables are called the augend and addend bits; the output variables produce the sum and carry. It is necessary to specify two output variables. We will assign symbols x and y to the two inputs and S and C for sum and carry to the outputs. Now we are ready to formulate a truth table to find out the function of the half-adder. This truth table is-

A	**B**	**C**	**S**
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

Now that we have the truth table, we can find the function from this truth table by applying SOP or POS. In this report we will apply POS or Product of Sum method to find the function from this truth table. The simplified Boolean expression of SOP are-

Minterm/SOP for C:

$M_3 = A.B = 1$

Therefore, C = A.B

Minterm/SOP for S:

$M_1 = A'B$

$M_2 = AB'$

$S = A'B + AB'$

$= A \oplus B$ [We know that, $x \oplus y = x'y + xy'$]

Therefore, $S = A \oplus B$

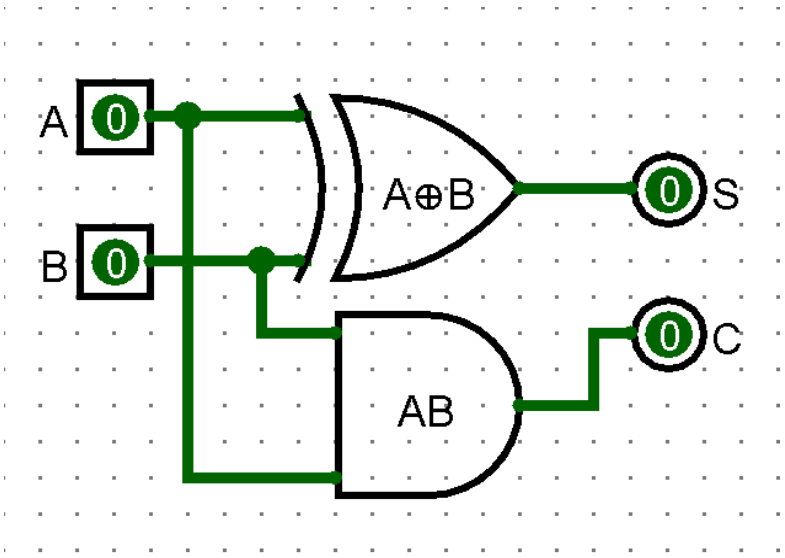

Figure 1-Implementation of Half Adder

Full Adder:

A full-adder is a circuit that do the arithmetic sum of three input bits. It consists of three inputs and two outputs. Two of the input variables, denoted by A and B, represent the two significant bits to be added. The third input, C_{in}, represents the carry from the previous lower significant position. Two outputs are necessary because the arithmetic sum of three binary digits ranges in value from 0 to 3, and binary 2 or 3 needs two digits. The two outputs are designated by the symbols S for sum and C_{out} for carry. The binary variable S gives the value of the least significant bit of the sum. The binary variable C gives the output carry. If all possible combinations outcome and input of full-adder placed in truth table and find out the simplified expression of C_{out} and S(Sum), it will be,

$$\boldsymbol{C_{out} = AB + C_{in}(A \oplus B) \text{ and } S = C \oplus (A \oplus B)}$$

Now that we got the expression for C_{out} and S, we can implement this and get our desired circuit that calculate sum of 2 single Digit Binary number with 3 input including Carry. Logic diagram for implemented simplified expression of C and S given in Figure 2.

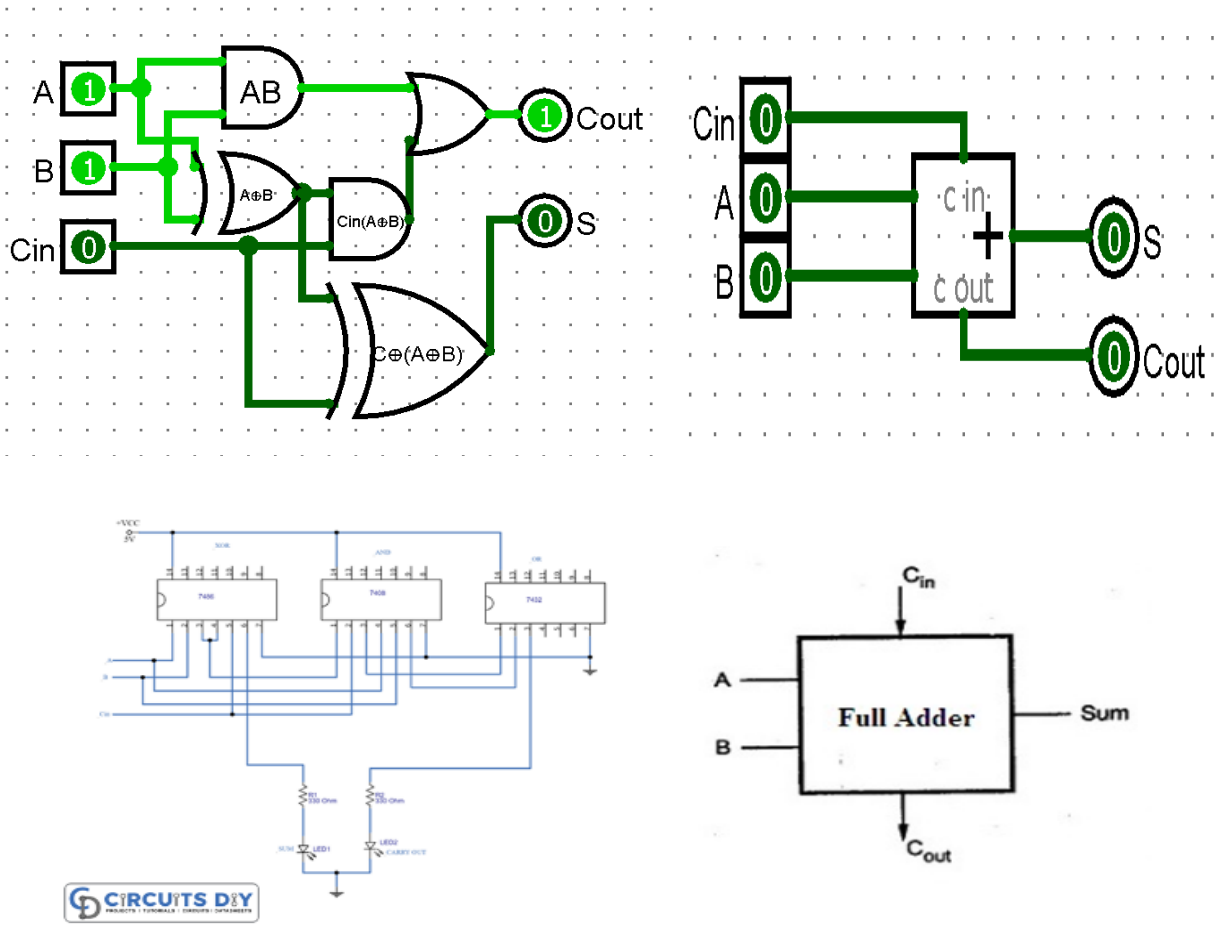

Figure 2- Full Adder logic and circuit diagram

Ripple Carry Adder:

A ripple carry adder is a type of digital adder circuit used to perform binary addition of multiple bits. It is called a "ripple carry" adder because the carry bit ripples through each stage of the adder, propagating from the least significant bit (LSB) to the most significant bit (MSB). The addition process in a ripple carry adder is sequential, meaning that each stage performs its operation based on the carry bit from the previous stage.

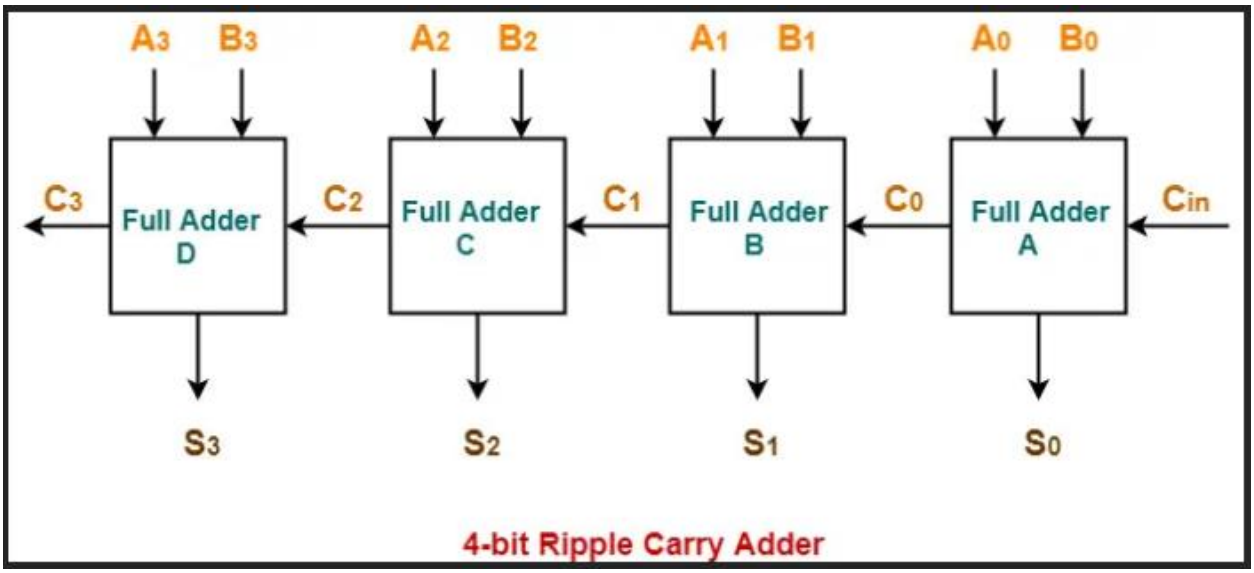

Figure 3- source: https://www.gatevidyalay.com/ripple-carry-adder

The basic building block of a ripple carry adder is a full adder. A full adder takes three inputs: two binary bits to be added (A and B) and a carry-in bit (Cin) and produces two outputs: the sum bit (S) and the carry-out bit (Cout). The carry-out bit from each full adder becomes the carry-in bit for the next stage.

The main drawback of a ripple carry adder is its propagation delay. Since the carry bit must ripple through each stage, the final sum output is only available after the longest propagation delay, which occurs when the carry must ripple through all the stages. This can limit the speed and efficiency of the adder, especially for large numbers of bits.

Even there's some drawback ripple carry adders are still used in applications where speed is not the primary concern or when the number of bits is relatively small.

Solution of the mentioned problems:

(1)

Implement a 1-bit full-adder using XOR gates.

Truth Table:

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

Equations (Using SOP) from the truth table:

Minterm for C_{out} :

$C_{minterm} = m_3 + m_5 + m_6 + m_7$

$= A'B\,C_{in} + AB'\,C_{in} + AB\,C_{in}' + AB\,C_{in}$

$= AB\,(C_{in}' + C_{in}) + C_{in}\,(A'B + AB')$

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

Minterm for S :

$S_{minterm} = m_1 + m_2 + m_4 + m_7$

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

$= C\,(A'B' + AB) + C'\,(A'B + AB')$

$= C\,(A \oplus B)' + C'\,(A \oplus B)$

$= C \oplus (A \oplus B)$

Diagram:

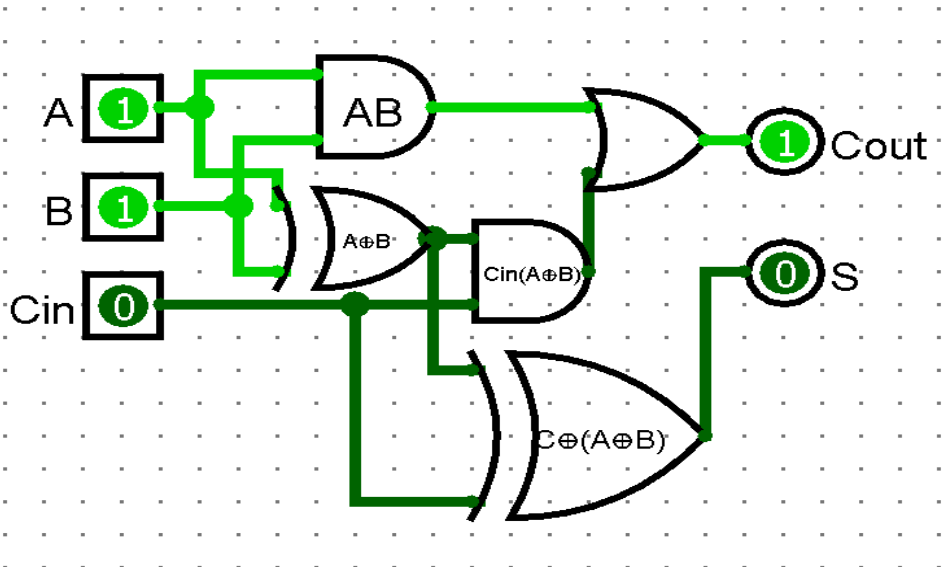

Figure 4-Implemented Full adder using XOR gate

(2)

1. Four 1-bit Full Adders

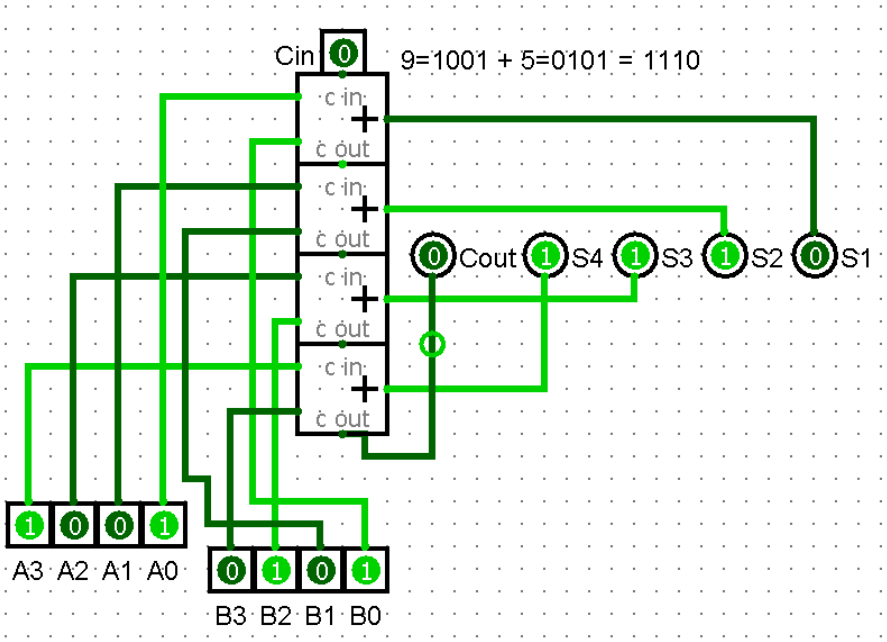

Figure 5-Implementing four 1-bit full adders

2. Two 2-bit Full Adders

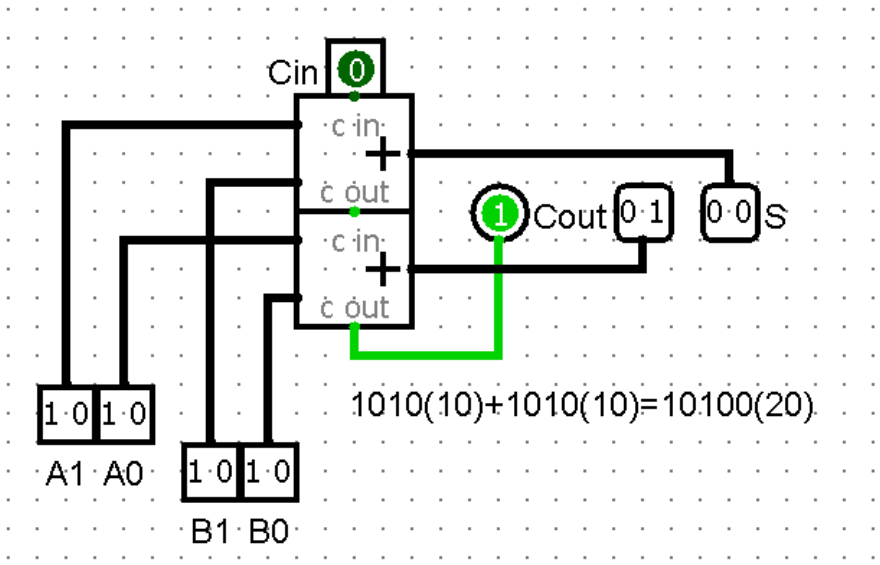

Figure 6-Implementing two 2-bit Full adders

3. One 4-bit Full Adder

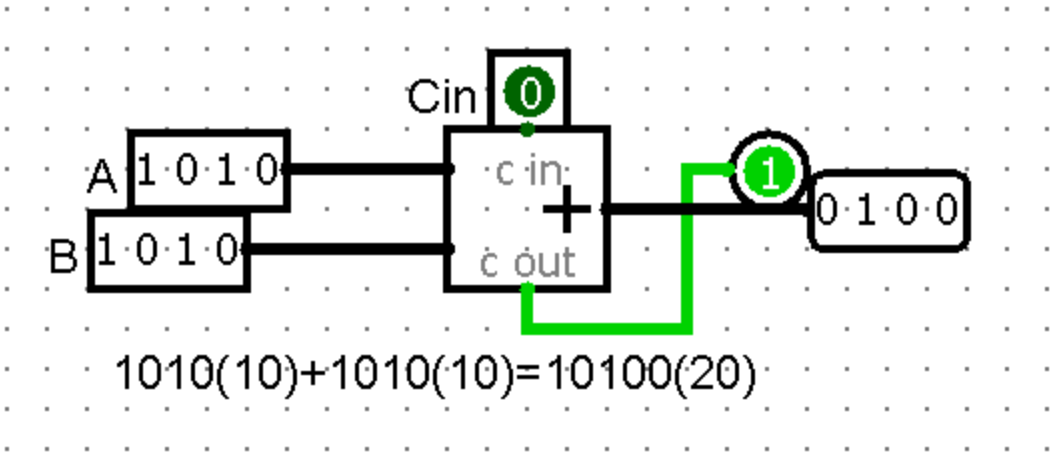

Figure 7-Implementing one 4-bit full adder

Discussion:

From this lab class I have learned how a digital circuit calculate sum of 2 digit. It helped me understanding how a Full Adder works and how to implement the circuit in the trainer board. I have also learned to check every pin of IC before implementing in the trainer board. Make sure to place all the output value serially, else it will be hard to understand the output.