North South University

Department of Electrical & Computer Engineering

LAB REPORT-5

Course Code: CSE231L

Course Title: Digital Logic Design

Section: 8

Experiment Number: 5

Experiment Name:

Binary Arithmetic

Experiment Date: 24/7/2021

Date of Submission: 8/8/2021

Submitted By: Md. Rifat Ahmed - 1931725042

Course Instructor: Md. Shahriar Hussain

Submitted To: Md. Anisur Rahman Asif

Objectives:

- Our first objective is to understand the concept of binary addition and subtraction.
- Then we need to learn about half and full binary adders.
- Then we have to perform binary addition and subtraction using IC74283.
- And finally, we have to understand the concept of BCD addition and implement a BCD adder using IC74283.

Apparatus:

Experiment 1:

- 1 x IC 74283 4-bit binary adder
- 1 x IC 7486 quadruple 2-Input XOR gates
- Trainer Board
- Wires

Experiment 2:

- 2 x IC 74283 4-bit binary adder
- Trainer Board
- Wires

Experiment 3:

- 2 x IC 74283 4-bit binary adder
- 1 x IC 7408 quadruple 2-Input AND gates
- 1 x IC 7432 quadruple 2-Input OR gates
- Trainer Board
- Wires

Theory:

Adder Circuits:

There are multiple types of adder circuits. But a half adder and the full adder circuit are the basic ones. A half adder can add binary numbers of 1-bits so it can perform the addition up to 2-bits. On the other hand, a full adder can do the job up to 3-bits. Both these circuits has 2 outputs Carry and Sum. Now what is carry and what is sum, when we add 2 binary numbers of one bit for example, 1+0 we get the sum 1 but when we add 1+1, we get the sum 10 but here we can consider the sum only up to 1 bit so the least significant bit here is the sum which is 0 and the higher significant bit is called the carry which will be 1.

And then if we look at Figure 2, we can see that a full adder circuit can be implemented using 2 half adders and an OR gate.

Experimental Procedure:

Experiment 1:

At first, we have to construct the 4-bit adder-subtractor circuit of Figure 3. Then we have to complete the Table 1 where we'll convert the first operand to binary as A, and the second operand as B then write down the value of M as 0 when using as an adder and 1 when using it as a subtractor then we also have to note down the values of the output carry C4 and the data output or the sum S4-S1 & finally verify the results.

Experiment 2:

We have to start with deducing the circuit diagram of an 8-bit ripple-through-carry binary adder using two 4-bit adders and thus constructing the 8-bit adder. After that we have to complete the Table 2.

Experiment 3:

For this experiment we'll start by completing the Table 3.1 for the BCD sum then construct the circuit of Figure 4 and then verify all the outputs in Table 3.2.

Question/Answer:

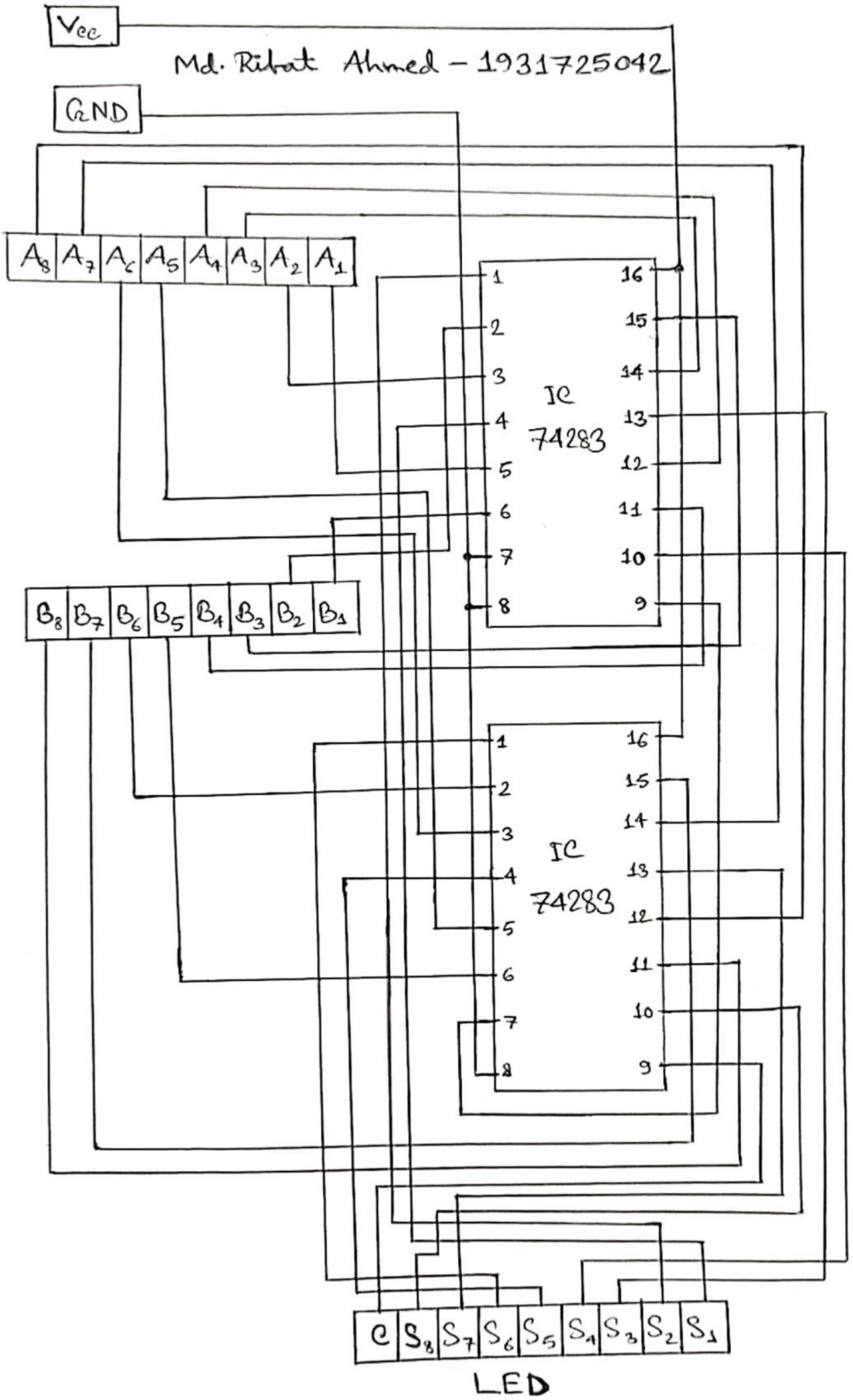
Answer to the Question No. 1 of Experiment 1:

Commenting on the use of the XOR gates and the M bit of the 4-bit adder-subtractor:

When subtracting we need to do a 2’s complement to complete a subtraction and in the adder-subtractor circuit in all of the XOR gates one of the values is coming from M so if M is 0 whatever the other input is the output will also be the same and thus when working as an adder the XOR gates won’t make any difference but when the value of M is 1 the XOR gates are doing the work of 1’s complement and then the Carry In M=1 works as the 2’s complement thus making our adder-subtractor circuit working as a subtractor.

Answer to the Question No. 1 of Experiment 2:

IC diagram for the 8-bit ripple-through-carry adder:



Answer to the Question No. 1 of Experiment 3:

Explaining how Binary Sum is converted into BCD Sum:

In Table 3.1 we can see that the Binary Sum and the BCD sum are exactly the same until 9 after that it’s different and that’s because in BCD we need to add an extra 6 to Binary number if its bigger than 9. Now if we take a closer look at the Table, we can see that after 9 the BCD sum follows a pattern where the Carry bit, C is always 1. Now this carry bit C is only 1 when any of the following 3 conditions are met,

(i) When both S_4 and S_2 are 1 in the Binary Sum

(ii) Or, when both S_4 and S_3 are 1 in the Binary Sum

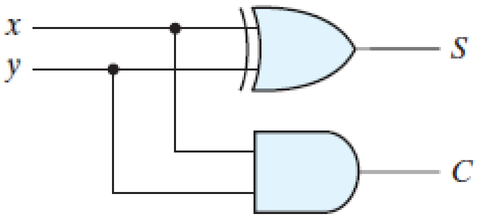
(iii) Or, when the carry bit of the Binary Sum, K=1.

Now if we look at Figure-4 we can see two AND gates that are used to fulfil our condition number (i) & (ii), and then the two OR gates are used to add our three conditions to the 2^{nd} adder circuit so that it can know when to add an extra 6 and when not to add the extra 6.

Discussion:

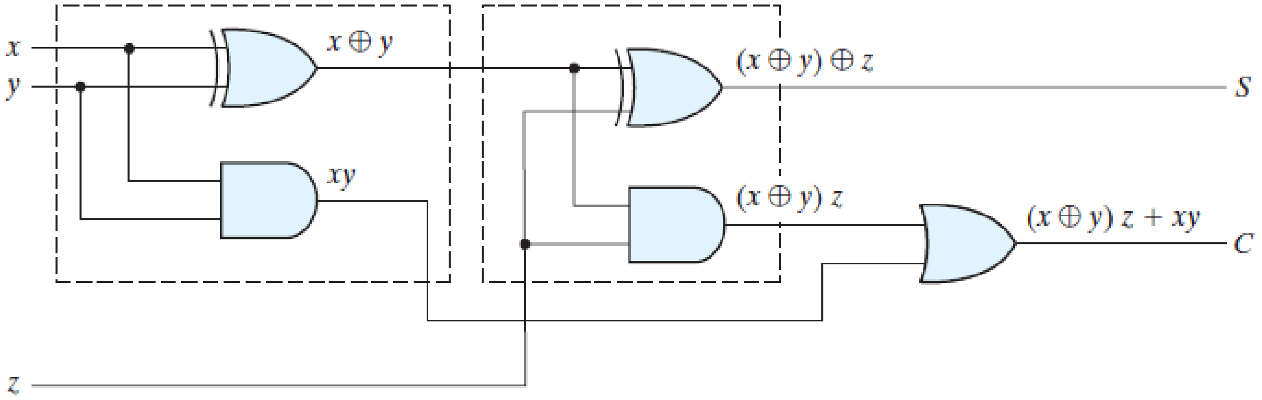
Through this lab we learned about 3 different types of circuits: binary adder-subtractor, 8-bit ripple-through-carry adder, BCD adder. Starting with the first one, we normally use adder or subtractor circuit but here we learned how we can combine them two into one single circuit. To simulate this experiment, we had to use a 4-bit adder and 4 XOR gates where the carry in (M) was connected to all the XOR gates which was kind of working as the indicator for the circuit to run as an adder or a subtractor. Then in the 2nd experiment we made an 8-bit ripple-through-carry adder using 2 4-bit adders where we first calculated the first 4 least significant bits then took their carry out into the 2nd adder as its carry in and then in this adder, we added the 4 higher significant bits (from A5 to A8 & B5 to B8) so the result from the 2nd adder is actually the sum from the 5th least significant bit (from S5 to S8) and then we connected all of the outputs into a LED matrix there we also connected the carry out as the 9th bit so that we can see if there's any carry overflow. And finally, we also learned about the BCD adder. We saw that Binary sum and BCD sum are same only until the sum is 9 but different when its more than 9 which is because whenever the binary number is more than 9, we need to add an extra 6 in BCD system. Then we needed to figure out how to make the circuit know when it needs to add an extra 6 and when it doesn't and for that we saw some patterns in Table 3.1 and then used two AND and two OR gates to follow that pattern and thus making our circuit know whenever the sum is more than 9 and add an extra 6 to it. However, we saw while simulating the circuit that it works perfectly only when the value of K is 0, whenever the value of K is 1 the circuit doesn't give us the carry value so it's safe to say that our circuit only works until the sum is 15. So overall we learned a lot of new things while doing this lab over 2 weeks and the only problem we encountered was with the BCD adder not showing its carry out correctly which we ultimately figured out.

Data Sheet & Circuit Diagrams:



$$S = x \oplus y$$
$$C = xy$$

Figure 1: Half Adder Circuit



Implementation of full adder with two half adders and an OR gate

Figure 2: Full Adder Circuit using 2 Half Adders and an OR gate

Data of Experiment 1 (Binary Adder-Subtractor):

Operation	M	A4 A3 A2 A1	B4 B3 B2 B1	C4	S4 S3 S2 S1
7 + 5	0	0 1 1 1	0 1 0 1	0	1 1 0 0
4 + 6	0	0 1 0 0	0 1 1 0	0	1 0 1 0
9 + 11	0	1 0 0 1	1 0 1 1	1	0 1 0 0
15 + 15	0	1 1 1 1	1 1 1 1	1	1 1 1 0
7 – 5	1	0 1 1 1	0 1 0 1	1	0 0 1 0
4 – 6	1	0 1 0 0	0 1 1 0	0	1 1 1 0
11 – 2	1	1 0 1 1	0 0 1 0	1	1 0 0 1
15 – 15	1	1 1 1 1	1 1 1 1	1	0 0 0 0

Table 1

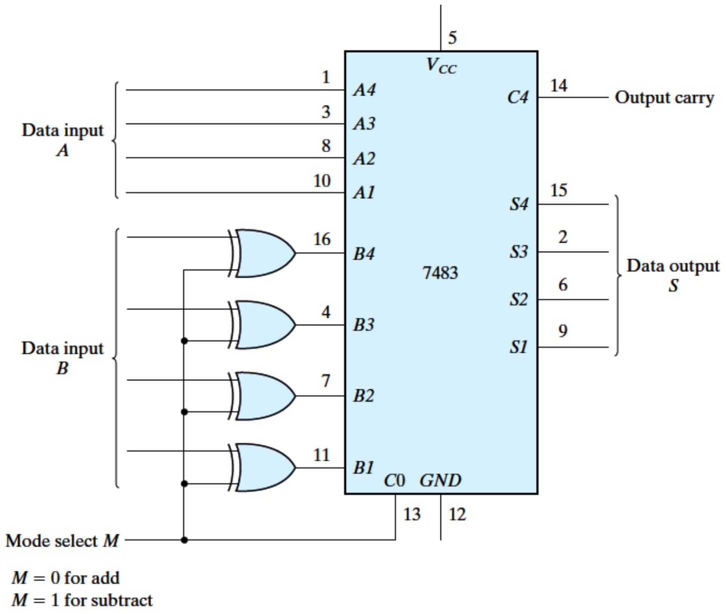


Figure 3: 4-bit Adder-Subtractor (Experiment 1)

Data of Experiment 2 (Ripple-Through-Carry Adder):

Operation	A8 A7 A6 A5 A4 A3 A2 A1	B8 B7 B6 B5 B4 B3 B2 B1	Overflow Carry	S8 S7 S6 S5 S4 S3 S2 S1
7 + 5	0 0 0 0 0 1 1 1	0 0 0 0 0 1 0 1	0	0 0 0 0 1 1 0 0
18 + 19	0 0 0 1 0 0 1 0	0 0 0 1 0 0 1 1	0	0 0 1 0 0 1 0 1
72 + 83	0 1 0 0 1 0 0 0	0 1 0 1 0 0 1 1	0	1 0 0 1 1 0 1 1
129 + 255	1 0 0 0 0 0 0 1	1 1 1 1 1 1 1 1	1	1 0 0 0 0 0 0 0

Table 2

Data of Experiment 3 (BCD Adder):

Decimal Value	Binary Sum					BCD Sum				
	K	S_4	S_3	S_2	S_1	C	S_4	S_3	S_2	S_1
0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	1	0	0	0	0	1
2	0	0	0	1	0	0	0	0	1	0
3	0	0	0	1	1	0	0	0	1	1
4	0	0	1	0	0	0	0	1	0	0
5	0	0	1	0	1	0	0	1	0	1
6	0	0	1	1	0	0	0	1	1	0
7	0	0	1	1	1	0	0	1	1	1
8	0	1	0	0	0	0	1	0	0	0
9	0	1	0	0	1	0	1	0	0	1
10	0	1	0	1	0	1	0	0	0	0
11	0	1	0	1	1	1	0	0	0	1
12	0	1	1	0	0	1	0	0	1	0
13	0	1	1	0	1	1	0	0	1	1
14	0	1	1	1	0	1	0	1	0	0
15	0	1	1	1	1	1	0	1	0	1
16	1	0	0	0	0	1	0	1	1	0
17	1	0	0	0	1	1	0	1	1	1
18	1	0	0	1	0	1	1	0	0	0
19	1	0	0	1	1	1	1	0	0	1

Table 3.1

Operation	A4 A3 A2 A1	B4 B3 B2 B1	Overflow Carry	S4 S3 S2 S1
9 + 0	1 0 0 1	0 0 0 0	0	1 0 0 1
9 + 1	1 0 0 1	0 0 0 1	1	0 0 0 0
9 + 2	1 0 0 1	0 0 1 0	1	0 0 0 1
9 + 3	1 0 0 1	0 0 1 1	1	0 0 1 0
9 + 4	1 0 0 1	0 1 0 0	1	0 0 1 1
9 + 5	1 0 0 1	0 1 0 1	1	0 1 0 0
9 + 6	1 0 0 1	0 1 1 0	1	0 1 0 1
9 + 7	1 0 0 1	0 1 1 1	1	0 1 1 0
9 + 8	1 0 0 1	1 0 0 0	1	0 1 1 1
9 + 9	1 0 0 1	1 0 0 1	1	1 0 0 0

Table 3.2

Decimal >> Binary >> BCD
0 >> 0000 >> 0000
9 >> 1001 >> 1001
10 >> 1010 >> 00010000
12 >> 1100 >> 00010010
34 >> Binary >> 00110100

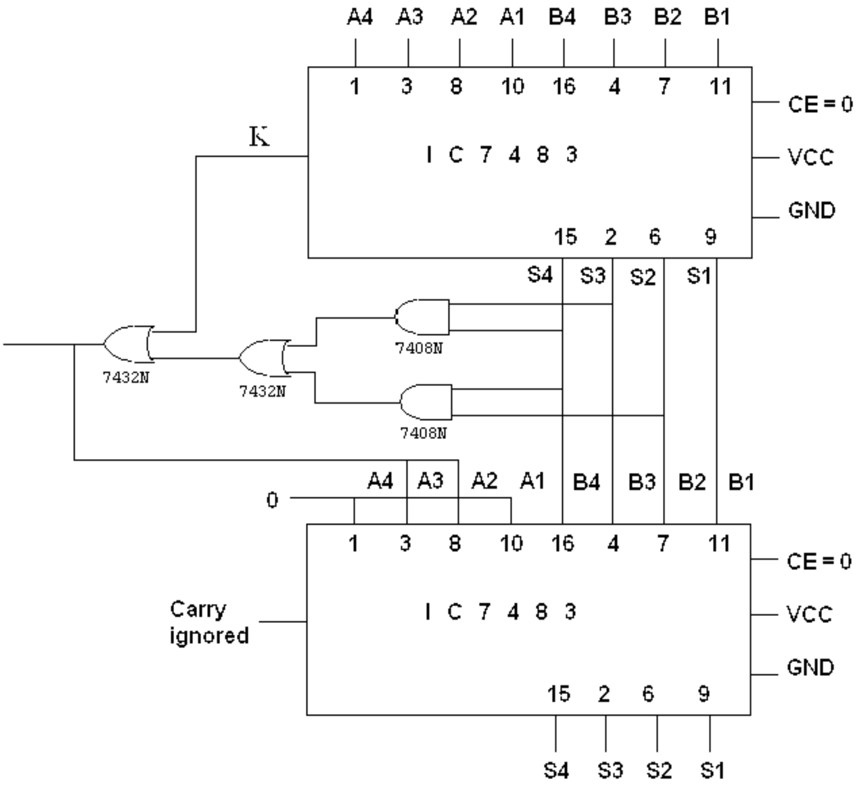
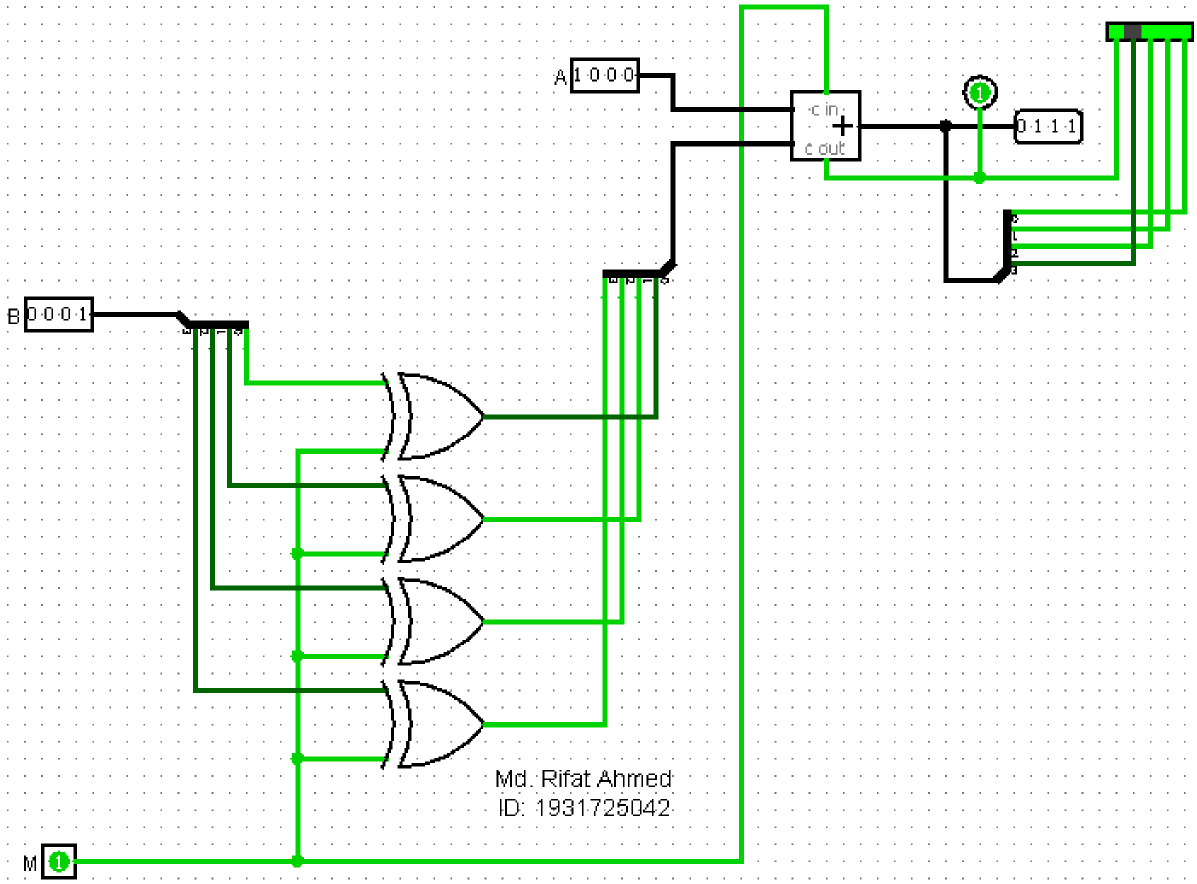


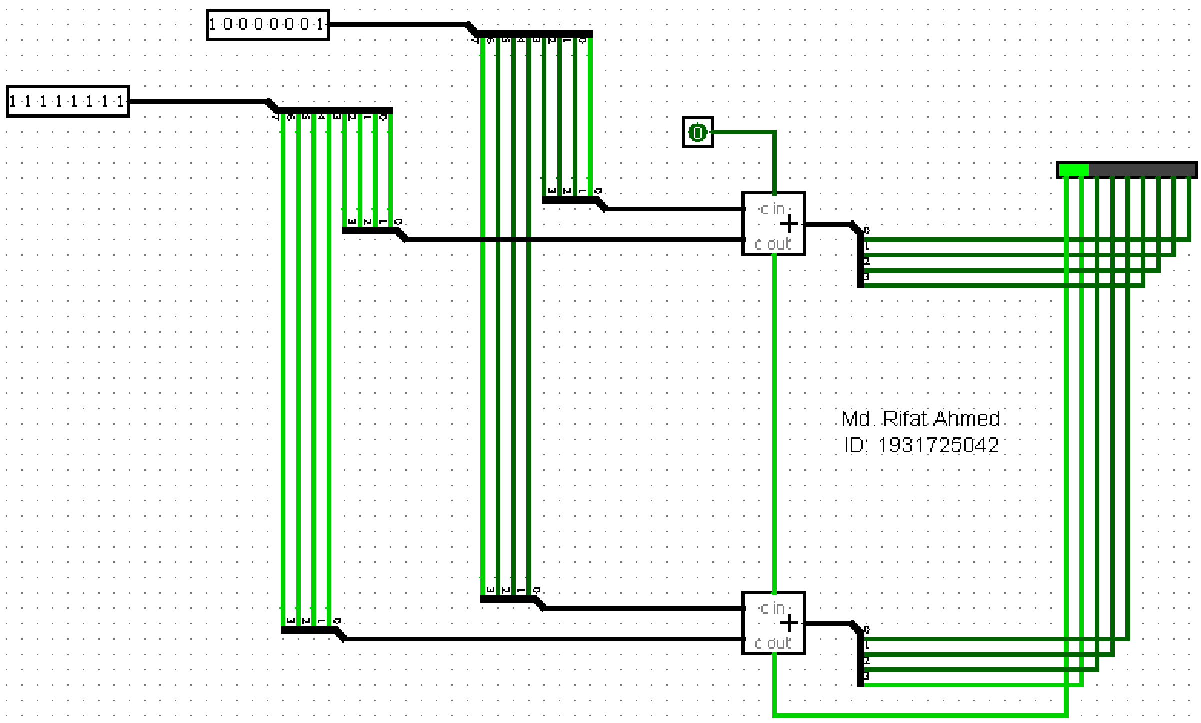
Figure 4: BCD Adder (Experiment 3)

Simulation:

Simulating the 4-bit Adder-Subtractor circuit from Experiment 1:



Simulating the 8-bit Ripple-through-carry Adder circuit from Experiment 2:


1 0 0 0 0 0 0 1
1 1 1 1 1 1 1 1
0
c in
c out
c in
c out
Md. Rifat Ahmed
ID: 1931725042