Lab5:Binary Arithmetic

A. Objectives

- Understand the concept of binary addition and subtraction.
- Learn about half and full binary adders.
- Perform binary addition and subtraction using IC74283.
- Understand the concept of BCD addition and implement a BCD adder using IC74283

B. Theory

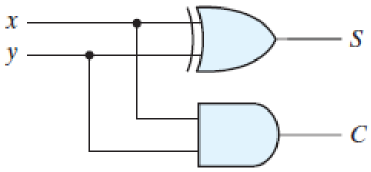
Digital computers perform a variety of information-processing tasks. Among the functions encountered are the various arithmetic operations. The most basic arithmetic operation is the addition of two binary digits. This simple addition consists of four possible elementary operations: 0 + 0 = 0, 0 + 1 = 1, 1 + 0 = 1, and 1 + 1 = 10. The first three operations produce a sum of one digit, but when both augend and addend bits are equal to 1, the binary sum consists of two digits. The higher significant bit of this result is called a ***carry***. When the augend and addend numbers contain more significant digits, the carry obtained from the addition of two bits is added to the next higher order pair of significant bits.

A combinational circuit that performs the addition of two bits is called a ***half adder***. One that performs the addition of three bits (two significant bits and a previous carry) is a ***full adder***. The names of the circuits stem from the fact that two half adders can be employed to implement a full adder.

In practice, binary addition is usually performed using ICs that contain several full adders chained together and can be used to add together groups of bits. These ICs themselves can be chained to form even larger adders. Since binary subtraction is performed by complement addition, the adder ICs can also be used for subtraction by using some extra logical operations to perform the complement calculation.

Half Adder

x	y	C	S
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

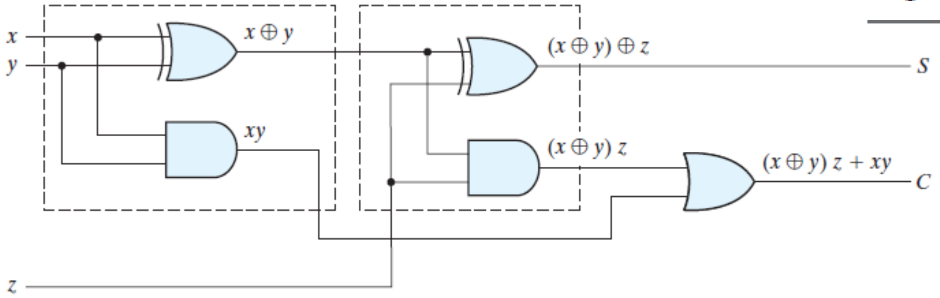


$S = x \oplus y$
$C = xy$

Figure B.1: Logic Diagram and Truth Table of a half adder

Full Adder

x	y	z	C	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



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

Figure B.2: Logic Diagram and Truth Table of a full adder

Experiment 1: Binary Adder Subtractor

C.1 Apparatus

- Trainer Board
- 1 x IC 742834-bit binary adder
- 2 x IC 7486 quadruple 2-input XOR gates

New Apparatus:
IC 74283: The 16-pin 74283 IC is a 4-bit full adder. That means, it can take two 4-bit binary numbers (**$A_4A_3A_2A_1$** and **$B_4B_3B_2B_1$**) and calculate the sum (**$S_4S_3S_2S_1$**). The input carry (if any) is connected to C_{in} and the output carry is obtained from C_{out}.

Two 74283 ICs can be cascaded to form an 8-bit ripple-through-carry adder. The lower 4 bits of each number is used as input for the first 74283 and the output carry is connected to the input carry of the next 74283. The higher 4 bits of each number is used as inputs for the second 74283. The first IC provides the lower 4 bits of the sum and the second one provides the upper 4 bits.

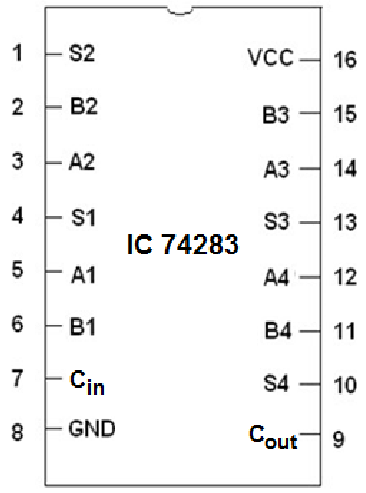


Figure C.1.1:
Pinout of IC74283

D.1 Procedure

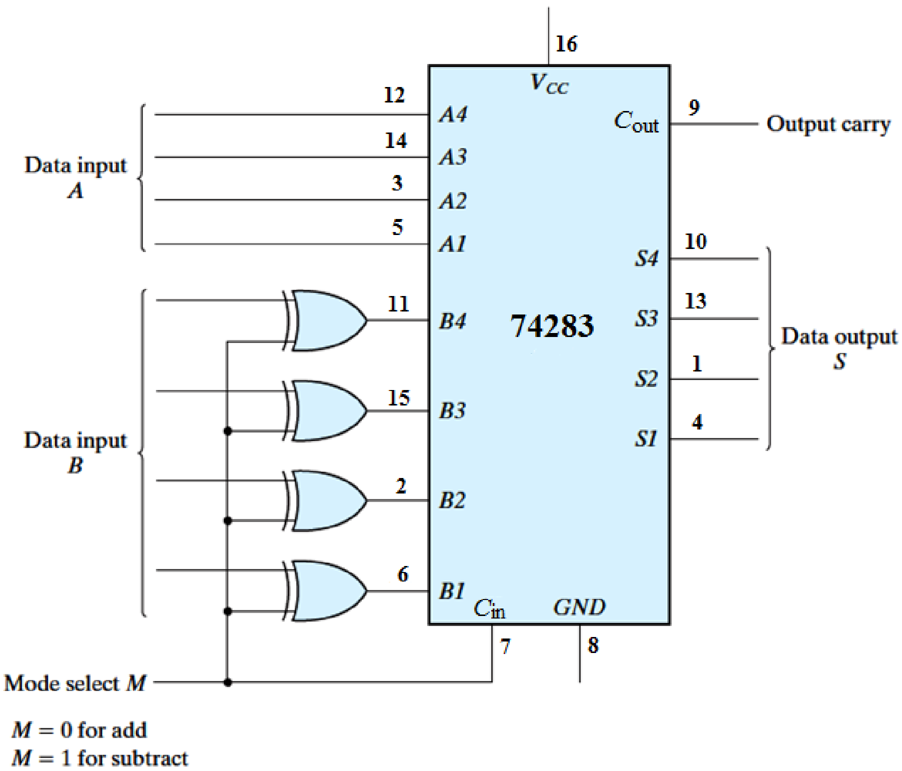


Figure D.1.1: 4-bit Adder-Subtractor

1. Construct the 4-bit adder-subtractor circuit of **Figure D.1.1** using 4-bit full adder and the XOR gates. Use four binary switches to represent the bits of input A and four more binary switches to represent the bits of input B. Use another switch for the mode select M. Use 4 LEDs to view the output S and another LED for the output carry C4.
2. Complete the operations in **Table F.1.1**.
   i. For each operation, convert the first operand to binary as A, and the second operand as B.
   ii. Write down the value of M required for the operation. M should be 0 for an addition operation and 1 for a subtraction operation.
   iii. Note down the values of the output carry C4 and data output S4-S1. Verify the results.

Experiment 2: BCD Adder

C.2 Apparatus

- Trainer board
- 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

D.2 Procedure

1. Complete **Table F.2.1** for the BCD sum. In BCD, a group of four bits can only represent the decimal values from 0 to 9, after which we need to use higher order bits. Here, ‘C’ represents that higher order bit.
2. Construct the circuit of **Figure D.2.1**. Unlike the previous circuit, this is a 4-bit adder despite the fact that two 74283 ICs are being used.
   i. The output of the first IC74283 (the upper one in the figure) is fed into the input of the second IC74283.
   ii. The output of the second IC74283 (the lower one in the figure) is connected to four LEDs.
   iii. The combinational circuit created with AND and OR gates does the work of converting the binary sum to the BCD sum.
3. Verify the outputs in **Table F.2.2**.

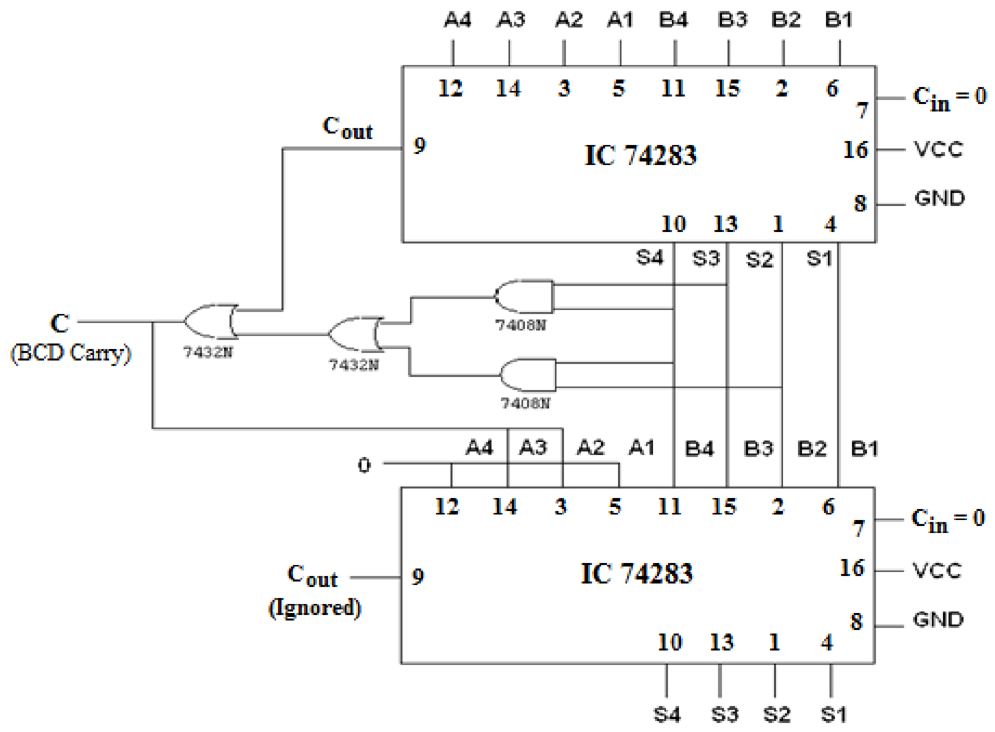


Figure D.2.1: BCD Adder Circuit

Questions:

1) Simulate a 4-bit adder in Logisim using **basic logic gates** (AND, OR, NOT, XOR). Provide a screenshot of the Logisim circuit schematic with your report with the two inputs set to your lab group number. (If your group number is 7, then the inputs should be 0111 and 0111).

2) Explain how the XOR gates and the M bit are being used in the 4-bit adder-subtractor to perform subtraction operations when the M bit is set to 1.

3) Explain the workings of the circuit for the BCD adder (Fig D.2.1 in the manual). Your explanation should cover the following points:
   a) The functions of the top and bottom 74283 4-bit adders.
   b) The inputs and outputs of the two adders.
   c) The function of the combinational circuit (AND and OR gates) between the two adders.
   d) The principles using which the binary sum is being converted to BCD.

EEE211L – Lab 5 – Binary Arithmetic

Data Sheet: Instructor’s Signature: ....

Section:	Group No.:	Date:

F.1 Experimental data (4-bit Binary Adder-Subtractor):

Operation	**M**	**A**	**B**	**C_{out}** (From LED)	**S4 S3 S2 S1** (From LEDs)
7 + 5					
4 + 6					
9 + 11					
15 + 15					
7 – 5					
4 – 6					
11 – 2					
15 – 15					

Table F.1.1

F.2 Experimental Data (BCD Adder):

Decimal Value	Binary Sum					BCD Sum				
	C_{out}	Z_3	Z_2	Z_1	Z_0	C	S_3	S_2	S_1	S_0
0	0	0	0	0	0					
1	0	0	0	0	1					
2	0	0	0	1	0					
3	0	0	0	1	1					
4	0	0	1	0	0					
5	0	0	1	0	1					
6	0	0	1	1	0					
7	0	0	1	1	1					
8	0	1	0	0	0					
9	0	1	0	0	1					
10	0	1	0	1	0					
11	0	1	0	1	1					
12	0	1	1	0	0					
13	0	1	1	0	1					
14	0	1	1	1	0					
15	0	1	1	1	1					
16	1	0	0	0	0					
17	1	0	0	0	1					
18	1	0	0	1	0					
19	1	0	0	1	1					

Table F.2.1

Operation	A	B	**Overflow Carry** (From LED)	**Sum** (From LEDs)
9 + 0				
9 + 1				
9 + 2				
9 + 3				
9 + 4				
9 + 5				
9 + 6				
9 + 7				
9 + 8				
9 + 9				

Table F.2.2