|  |  |  |
| --- | --- | --- |
| G:\nsu-logo.png  **North South University**  Department of Electrical & Computer Engineering    **LAB REPORT**  Course Name:CSE231L  Section: 01  Experiment Number: 05     |  | | --- | | Experiment Name: Binary Arithmetic |     Experiment Date: 21/03/2021  Report Submission Date:28/03/2021  Group Number: | |
| Student Name:Ishrat Jahan | Score |
| Student ID:1921909042 |  |
| Remarks: |

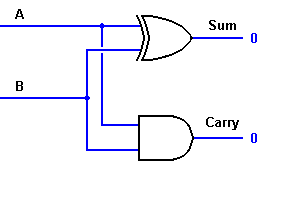
**Binary Arithmetic**

**Objectives**

* Understand the concept of binary addition and subtraction.
* Learn about half and full binary adders.
* Understand the concept of BCD addition

**Theory**

**Half Adder:** The addition of 2 bits is done using a combination circuit called Half Adder. The input variables are augend and addend bits and output variables are sum and carry bits.A and B are the two input bits. With the help of half adder, we can design circuits that are capable of performing simple addition with the help of logic gates.



**Full Adder**

Full Adder is the adder which adds three inputs and produces two outputs. The first two inputs are A and B and the third input is an input carry as C-IN. The output carry is C-OUT and the normal output is S which is SUM.A full adder logic is designed in such a manner that it can take eight inputs together to create a byte-wide adder and cascade the carry bit from one adder to another. We can create a Full Adder by using two Half Adders and connecting them by an OR gate. A single full adder performs the addition of two one bit numbers and an input carry. For performing the addition of binary numbers with more than one bit, more than one full adder is required and it depends on the number bits..



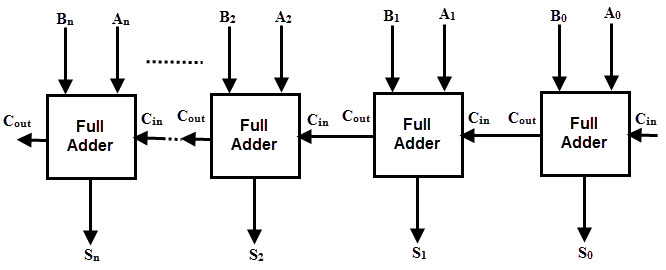
**Sub-Circuit**

Sub-circuits are self-contained circuits that appear as "boxes" in other circuits. A given sub-circuit may be used many number of times at multiple levels just as long as it does not become a sub-circuit of itself.

https://www.cs.oberlin.edu/~rms/dlsim.com/uguide/images/suby.gif

**Binary adder-subtractor**

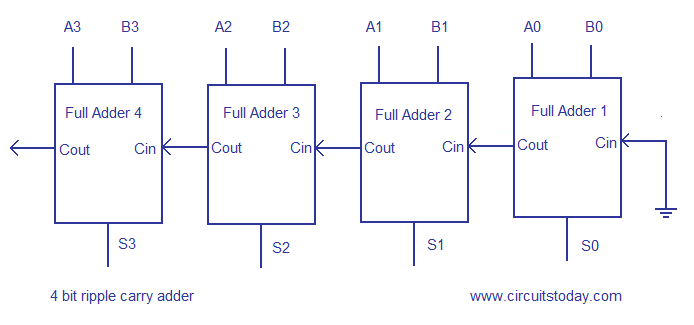
A **Binary Adder-Subtractor** is one which is capable of both addition and subtraction of binary numbers in one circuit itself. For a 4-bit Binary adder-subtractor the circuit consists of 4 full-adders since we are performing operation on 4-bit numbers.



For performing addition and subtraction we can create sub-circuit of this 4-bit full adder and use it to take 4-bit inputs. For subtraction, we need to make Cin as constant 0 and one input connected to a NOT gate and for addition Cin as constant 1. This Cin decides the type of operation, whether addition or subtraction. By varying the number of full adders we can decide the input number of bits.

**Ripple-Through Carry Adder**

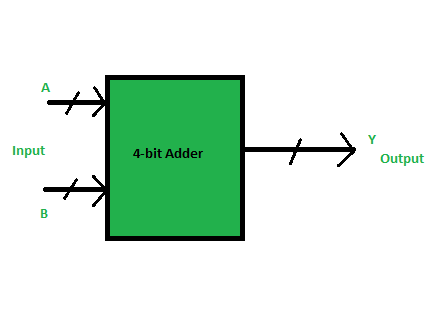
A ripple carry adder is a logic circuit in which the carry-out of each full adder is the carry in of the succeeding next full adder. It is called a ripple carry adder because each carry bit gets rippled into the next stage.



The Sum of S3 of the Full-Adder 4 is valid only after the joint propagation delays of Full Adder 1 to Full Adder 4. In simple words, the final result of the ripple carry adder is valid only after the joint propagation delays of all full adder circuits inside it.

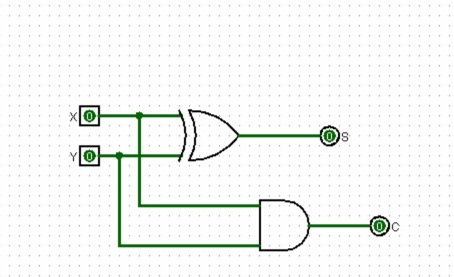
**BCD Adder**

BCD stands for binary coded decimal. Suppose, we have two 4-bit numbers A and B. The value of A and B can varies from 0(0000 in binary) to 9(1001 in binary) because we are considering decimal numbers.

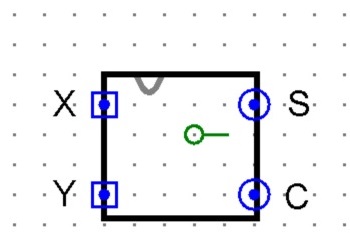


If the sum of two number is less than or equal to 9, then the value of BCD sum and binary sum will be same otherwise they will differ by 6(0110 in binary).

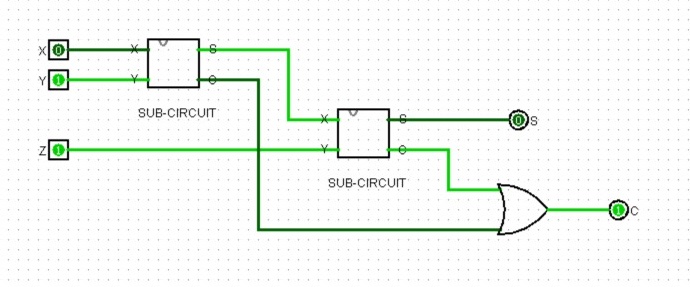
**Circuit Diagrams**

****

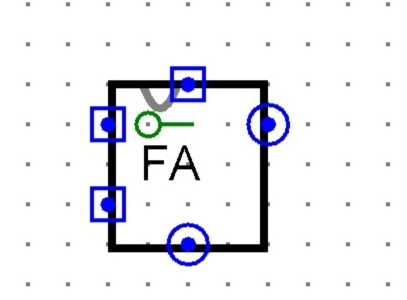
**Half adder**



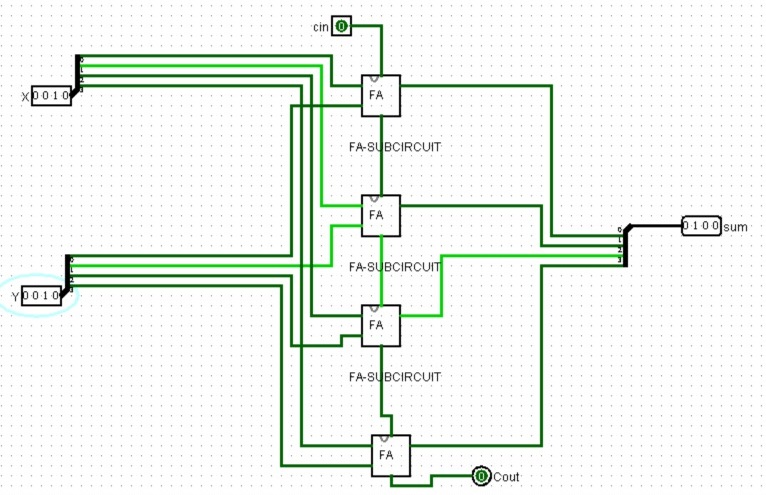
**Half adder sub-circuit**



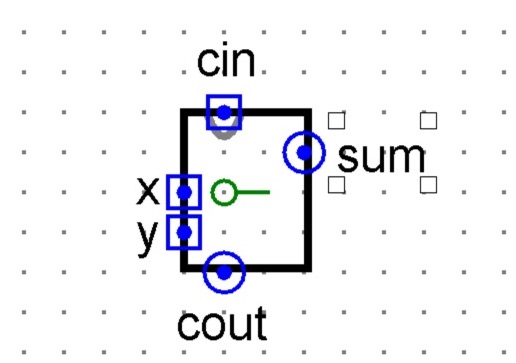
**Implementation of full adder with two half adders and an OR gate**



**Full adder sub-circuit**

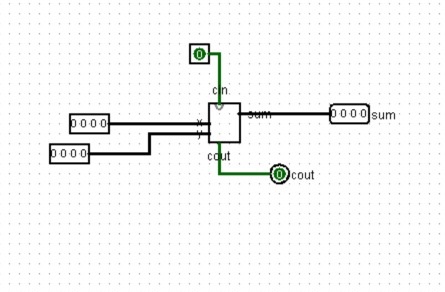


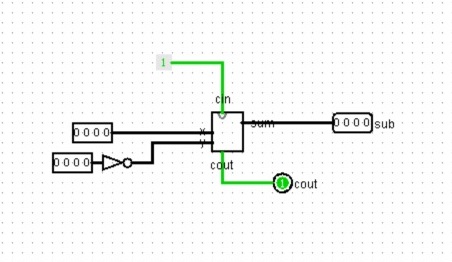
**4-bit full-adder**



**4-bit full-adder sub-circuit**

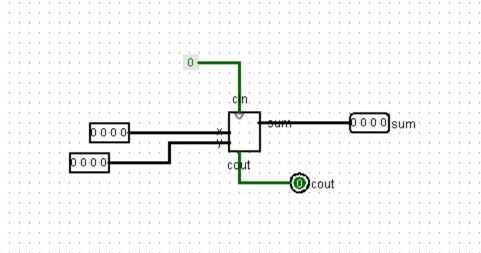
**Experiment-1**





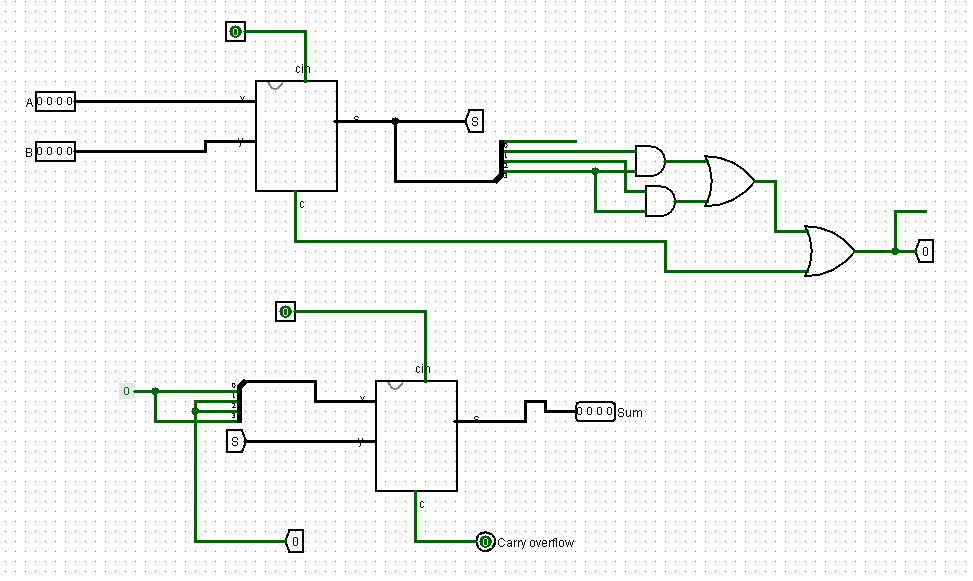
**Figure D.1.1 4-bit adder-subtractor**

**Experiment-2**



**Fig D.2: Ripple-Through Carry Adder**

**Experiment-3**



**Figure D.3.1**

**Truth Table**

**F.1 Experimental Data (Binary Adder-Subtractor):**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Operation** | **M** | **A** | **B** | **C4** | **S4 S3 S2 S1** |
| **7+5** | **12** | **0111** | **0101** | **0** | **1100** |
| **4+6** | **10** | **0100** | **0110** | **0** | **1010** |
| **9+11** | **20** | **1001** | **1011** | **1** | **0100** |
| **15+15** | **30** | **1111** | **1111** | **1** | **1110** |
| **7-5** | **2** | **0111** | **0101** | **1** | **0010** |
| **4-6** | **-2** | **0100** | **0110** | **0** | **1110** |
| **11-2** | **9** | **1011** | **0010** | **1** | **1001** |
| **15-15** | **0** | **1111** | **1111** | **1** | **0000** |

**TableF.1.1**

**F.2 Experimental Data (Ripple-Through-Carry Adder):**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Operation** | **A** | **B** | **Overflow carry** | **Sum** |
| **7+5** | **0111** | **0101** | **0** | **1100** |
| **18+19** | **1000** | **1001** | **1** | **0001** |
| **72+83** | **0010** | **0011** | **0** | **0101** |
| **129+255** | **1001** | **0101** | **0** | **1110** |

**TableF.2.**

**F.3 Experimental Data (BCD Adder):**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **Binary Sum** | | | | | **BCD Sum** | | | | |
| **Decimal Value** | **K** | **Z3** | **Z2** | **Z1** | **Z0** | **C** | **S3** | **S2** | **S1** | **S0** |
| **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** | **1** | **1** | **0** | **1** | **0** | **0** | **1** | **0** | **1** |
| **6** | **0** | **1** | **1** | **1** | **0** | **0** | **0** | **1** | **1** | **0** |
| **7** | **0** | **1** | **1** | **1** | **1** | **0** | **0** | **1** | **1** | **1** |
| **8** | **0** | **1** | **0** | **0** | **0** | **0** | **1** | **0** | **0** | **0** |
| **9** | **0** | **0** | **0** | **0** | **1** | **0** | **1** | **0** | **0** | **1** |
| **10** | **0** | **0** | **0** | **1** | **0** | **1** | **0** | **0** | **0** | **0** |
| **11** | **0** | **0** | **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 F.3.1**

**F.3 Experimental Data (BCDAdder):**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Operation** | **A** | **B** | **Overflow Carry** | **Sum** |
| **9+0** | **1001** | **0000** | **0** | **1001** |
| **9+1** | **1001** | **0001** | **1** | **0000** |
| **9+2** | **1001** | **0010** | **1** | **0001** |
| **9+3** | **1001** | **0011** | **1** | **0010** |
| **9+4** | **1001** | **0100** | **1** | **0011** |
| **9+5** | **1001** | **0101** | **1** | **0100** |
| **9+6** | **1001** | **0110** | **1** | **0101** |
| **9+7** | **1001** | **0111** | **0** | **0110** |
| **9+8** | **1001** | **1000** | **0** | **0111** |
| **9+9** | **1001** | **1001** | **0** | **1000** |

**Table F.3.2**

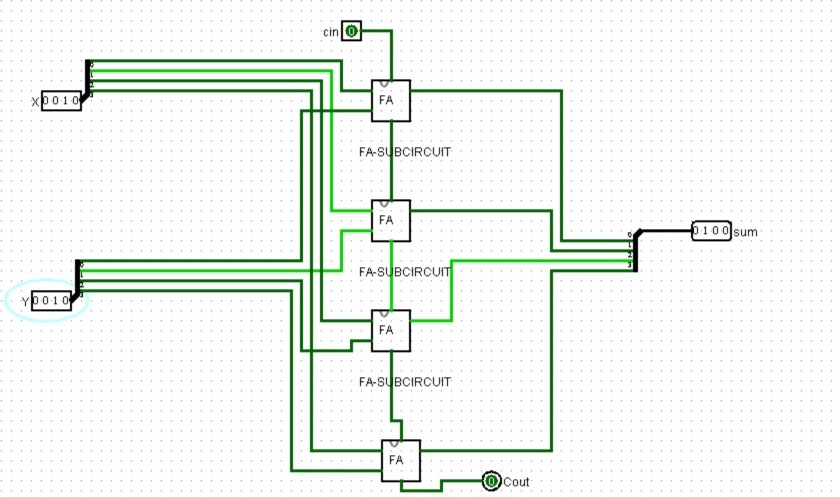
**Questions and Answers**

**E.1**

**1)** XOR gate will only produce an output “1” when either input is at logic 1, so we need an additional output to produce the carry bit when both inputs A and B are at logic 1.

The M-bit is a control line that holds a binary value of either 0 or 1 which determines that the operation being carried out is addition or subtraction.

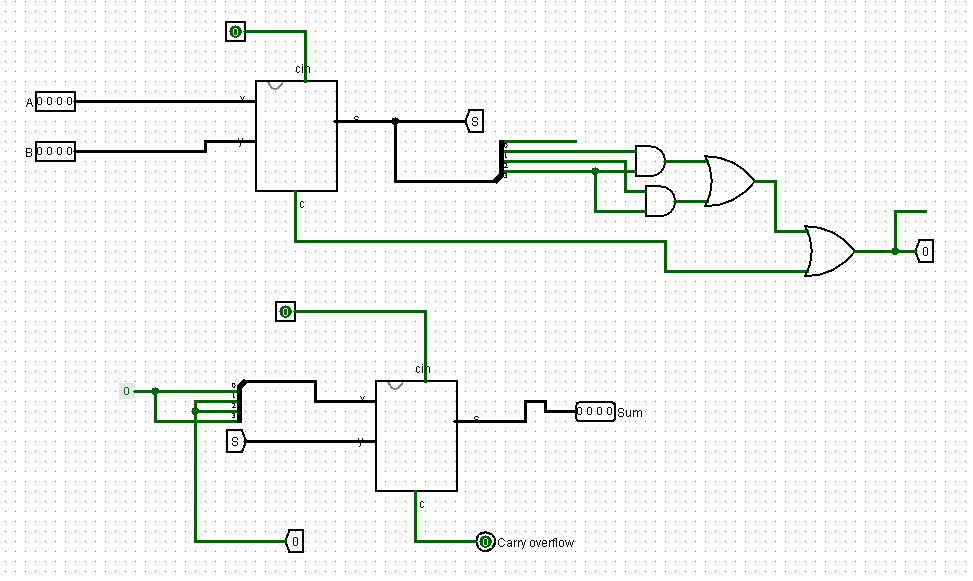
**2)**

****

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **X3** | **X2** | **X1** | **X0** | **Y3** | **Y2** | **Y1** | **Y0** | **Cout** | **S3** | **S2** | **S1** | **S0** |
| **1** | **0** | **0** | **0** | **0** | **0** | **1** | **0** | **0** | **1** | **0** | **1** | **0** |
| **1** | **0** | **0** | **0** | **1** | **0** | **0** | **0** | **1** | **0** | **0** | **0** | **0** |
| **0** | **0** | **1** | **0** | **1** | **0** | **0** | **0** | **0** | **1** | **0** | **1** | **0** |
| **0** | **0** | **0** | **1** | **0** | **1** | **1** | **1** | **0** | **1** | **0** | **0** | **0** |
| **1** | **0** | **1** | **0** | **1** | **0** | **1** | **1** | **1** | **0** | **0** | **1** | **0** |
| **1** | **1** | **1** | **0** | **1** | **1** | **1** | **1** | **1** | **1** | **0** | **1** | **0** |
| **1** | **0** | **1** | **0** | **1** | **1** | **0** | **1** | **1** | **0** | **1** | **1** | **1** |

**E.3**

**Circuit for BCD-Adder**

****

**Discussion**

The topic of Lab-5 was Binary Arithmetics.As the name suggests; the lecture contained the arithmetic operations of Binary numbers. Firstly, we studied about carries and sum. For the 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 to1, the binary sum consists of two digits. The higher significant bit of this result is called a carry. Then, we learned about Half-Adders and how they perform addition of two bits using them. We studied how to draw the half adder circuit and its respective sub-circuit. Using the knowledge of Half-Adder we drew the circuit of a Full-Adder with three inputs using two half-adders and connected them with a OR gate with an output S and carry out C .Then we designed its respective sub-circuit. Secondly, using the sub-circuit of the full-adder we designed a 4-bit full adder circuit where we used 4 full-adders, as the inputs were 4 bits and changed the output to 4 bit as well. Then we used the sub-circuit of the 4-bit full adder to carry out Experiment-1, where we had to design a 4-bit binary adder-subtractor. For designing the adder we have to make the Cin(Carry input) to 1 and for the subtraction the Carry input to 0 using the constant. As each of the inputs were 4-bits we used splitter to make the inputs 4 bits where the data bits and fan out were also changed to 4.Following the completion of the circuit, we completed Table F.1.1 where M=Sum of the numbers in Decimal, A=Binary value of first operand=Binary values of second operand , C4=Cout, S4 S3 S2 S1 =The sum/sub, by varying the inputs in the circuit according to A and B’s binary value. We proceeded on to Experiment-2 where we studied about ripple through carry adder and we drew a circuit of 8-bit ripple-through-carry binary adder using two 4-bit adders and where Cin=0. We completed Table F.2.1 where A=Binary value of first operand, B=Binary value of second operand, Overflow Carry=Cout and Sum=The sum. Lastly, we studied the BCD adder from the D.3.1 and derived the circuit for it.We used two full adders and a tunnel to connect one output to others input to complete the circuit. Then we completed table F.3.1 where we derived the BCD values of each

Decimal values from 0-19 and Table F.3.2 where we verified the outputs of the D.3.1 circuit.