Heaven's Light is Our Guide

Rajshahi University of Engineering & Technology

Department of Electrical & Computer Engineering

LAB REPORT

- *Course title(Sessional) : Digital Techniques Sessional*
- *Course Code : ECE-2112*
- *Date of submission : 22/01/2025*

Submitted By :	Submitted To :
Name : Sanumong Marma	***Md. Omaer Faruq Goni***
Roll : 2210061	***Assistant Professor***
Reg no. : 1115	***Department of ECE,***
Series : 22	***RUET***
Department of ECE, RUET	

Lab no. : 02

Experiment name : Study of 4-bit adder subtractor using Verilog and DEEDS simulator

Theory: In Digital Circuits, A Binary Adder-Subtractor is can do both the addition and subtraction of binary numbers in one circuit itself. The operation is performed depending on the binary value the control signal holds. It is one of the components of the ALU (Arithmetic Logic Unit). A Binary Adder is an electronic circuit that sums two binary numbers. The most basic types are half adders and full adders. A half adder takes in two one-bit numbers, producing a sum and a carry bit. Full adder has three input bits-two actual bits and an incoming carry from the preceding operation. This makes the Full Adder to add multi-bit binary numbers as the carry can propagate through a number of Full Adders in a ripple-carry adder configuration. A Binary Subtractor is used to perform the subtraction of two binary numbers. Just as in the case of addition, this also can be done using an array of logic gates. The subtraction is done by using two's complement, in which the circuit first negates the binary number to obtain its complement and then adds one to the result before imputing it into a binary adder circuit together with the original number. Both addition and subtraction are handled by this type of circuit.

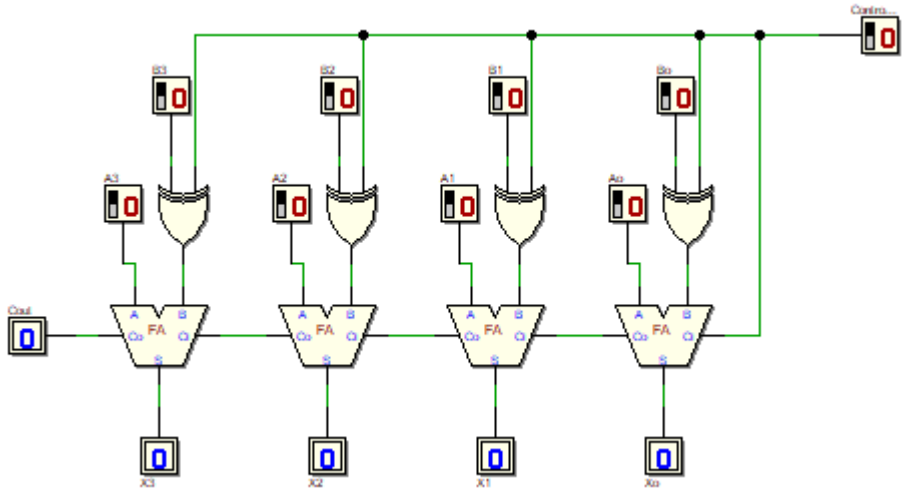

Figure 01: 4-bit adder/subtractor

Code :

```
module adder_subtractor (
    input [3:0] A,      // 4-bit Input A
    input [3:0] B,      // 4-bit Input B
    input SUB,          // Subtraction Control: 0 for Add, 1 for Subtract
    output [3:0] Result,// 4-bit Result
    output CarryOut     // Carry or Borrow Indicator
);
```

```
    wire [3:0] B_complement; // Complement of B for subtraction
    wire Carry;              // Carry output

    // Generate 2's complement of B if SUB is 1
    assign B_complement = B ^ {4{SUB}};

    // Perform addition
    assign {CarryOut, Result} = A + B_complement + SUB;
endmodule
`timescale 1ns / 1ps

module testbench;
    reg [3:0] A, B;      // 4-bit inputs
    reg SUB;             // Subtraction control
    wire [3:0] Result;   // 4-bit result
    wire CarryOut;       // Carry out

    // Instantiate the module under test (adder_subtractor)
    adder_subtractor uut (
        .A(A),
        .B(B),
        .SUB(SUB),
        .Result(Result),
        .CarryOut(CarryOut)
    );

    initial begin
        // Monitor the outputs
        $monitor("Time=%0t | A=%b B=%b SUB=%b => Result=%b CarryOut=%b", $time, A,
B, SUB, Result, CarryOut);

        // Test Cases
        A = 4'b0101; B = 4'b0011; SUB = 0; #10; // Add
        A = 4'b0101; B = 4'b0011; SUB = 1; #10; // Subtract
        A = 4'b1111; B = 4'b0001; SUB = 0; #10; // Add with carry
        A = 4'b1000; B = 4'b1000; SUB = 1; #10; // Subtract with borrow
        A = 4'b0000; B = 4'b0001; SUB = 1; #10; // Edge case: 0 - 1
        $finish;
    end
endmodule
```

Output :

```
Time=0 | A=0101 B=0011 SUB=0 => Result=1000 CarryOut=0
Time=10000 | A=0101 B=0011 SUB=1 => Result=0010 CarryOut=1
Time=20000 | A=1111 B=0001 SUB=0 => Result=0000 CarryOut=1
```

Time=30000 | A=1000 B=1000 SUB=1 => Result=0000 CarryOut=1
Time=40000 | A=0000 B=0001 SUB=1 => Result=1111 CarryOut=0

Discussion :

The 4-bit Adder-Subtractor module performs both addition and subtraction based on a control input SUB. When SUB is 0, it adds the inputs A and B; when SUB is 1, it subtracts B from A by inverting B and adding 1 (two's complement). The result is a 4-bit output Result, and a CarryOut signal is generated for overflow detection. This design allows for efficient arithmetic operations in digital systems.