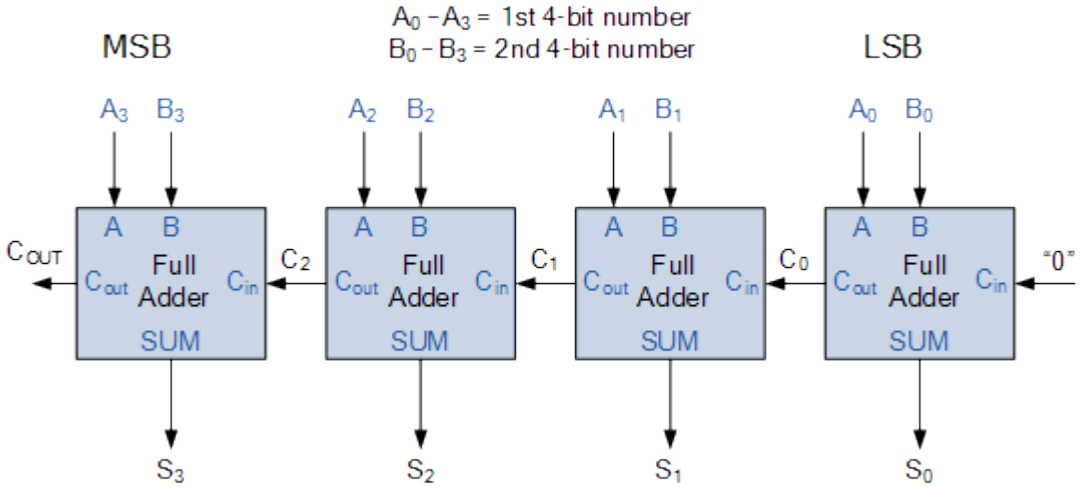
5. N-Bit Adder / Subtractor using Full Adders

Block Diagram:



***Adder*:**

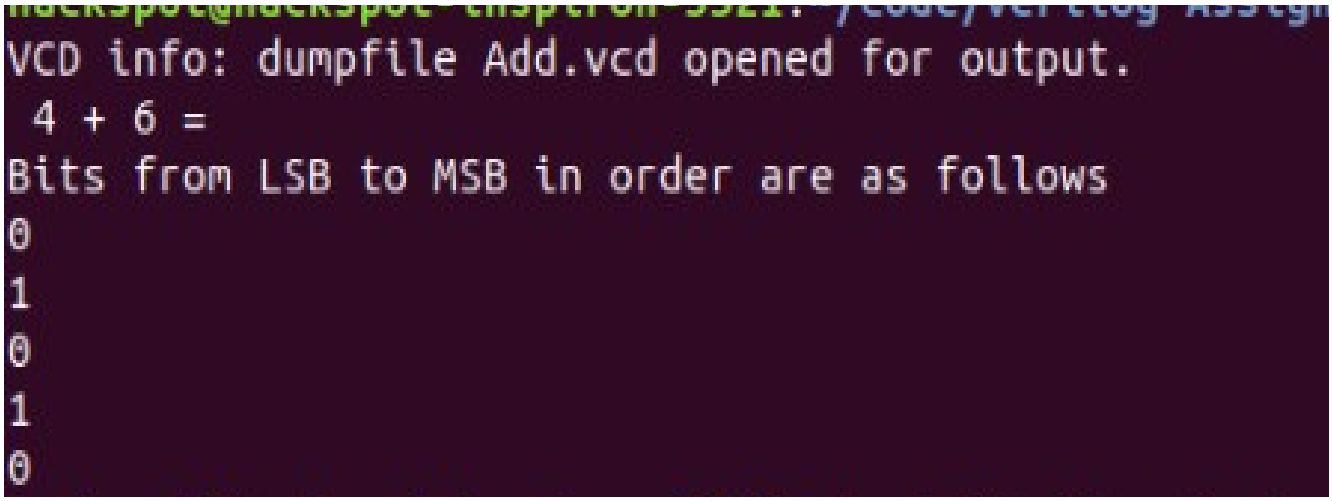
The circuit is used to add two n-bit values to give corresponding sum and carry outputs. The full adder starts adding bits (starting from the least significant bit) and the carry is stored in an intermediate variable which is fed to the next full adder. Hence, ‘n’ full adders can add two n-bit values to give a n-bit sum value and 1-bit carry value.

Verilog Code:

```
        integer i;
        fullAdder inst (sum, carry, op1, op2, op3);     // Full Adder thread instantiated
        initial begin
                n1=4;   //random values n1 and n2
                n2=6;
                imm=0;
                op1=1'b0;
                op2=1'b0;
                op3=1'b0;
                $display("%d +%d = ",n1,n2);
                $display("Bits from LSB to MSB in order are as follows");
                //3 must be replaced by (n - 1)
                for (i=0;i<=3;i=i+1) begin
                #10     op1=n1[i];
                        op2=n2[i];
                        op3=imm;
                        assign imm=carry;
                #10     $display("%b",sum);     // Displaying the sum bit
                end

                $display("%b",carry);   // displaying the last carry bit
        end
endmodule
```

Output:



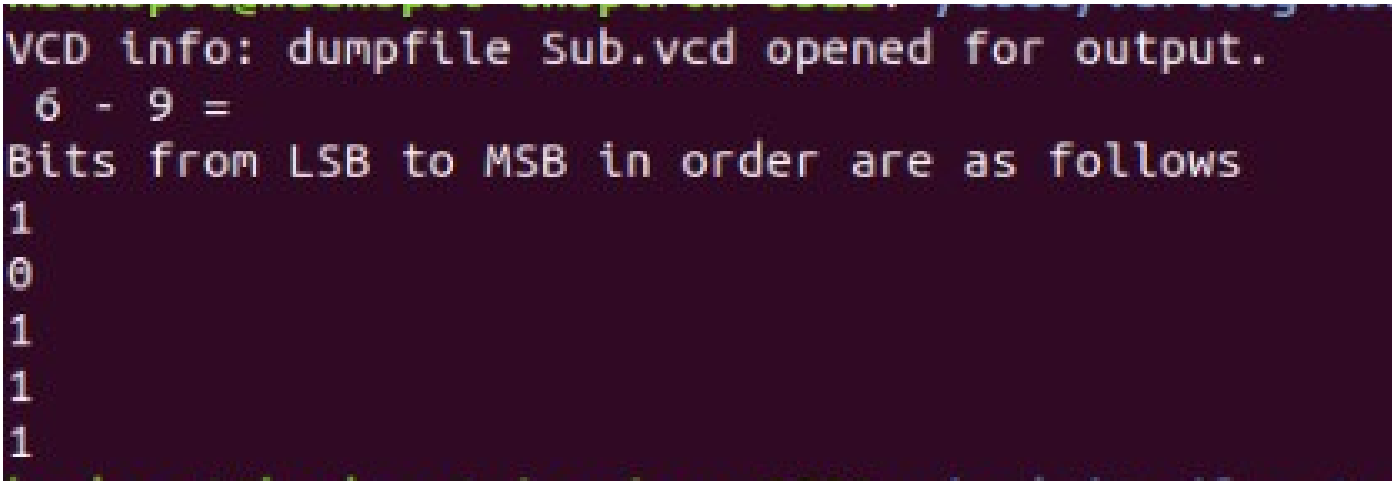
***Subtractor*:**

Adder can be modified into a subtractor by negating the input bits and changing the initial carry to 1.

```
        reg imm;                // immediate value
        integer i;
        fullAdder inst (sum, carry, op1, op2, op3);     // Full Adder thread instantiated
        initial begin
                n1=6;   //random values n1 and n2
                n2=9;
                imm=1;
                op1=1'b0;
                op2=1'b0;
                op3=1'b0;
                $display("%d -%d = ",n1,n2);
                $display("Bits from LSB to MSB in order are as follows");
                //3 must be replaced by (n - 1)
                for (i=0;i<=3;i=i+1) begin
                #10     op1=n1[i];
                        op2=~n2[i];
                        op3=imm;
                        assign imm=carry;
                #10     $display("%b",sum);     // Displaying the sum bit
                end

                $display("%b",~carry);  // displaying the last carry bit
        end
endmodule
```

Output:


VCD info: dumpfile Sub.vcd opened for output.
6 - 9 =
Bits from LSB to MSB in order are as follows
1
0
1
1
1