COE 405, Term 152

Design & Modeling of Digital Systems

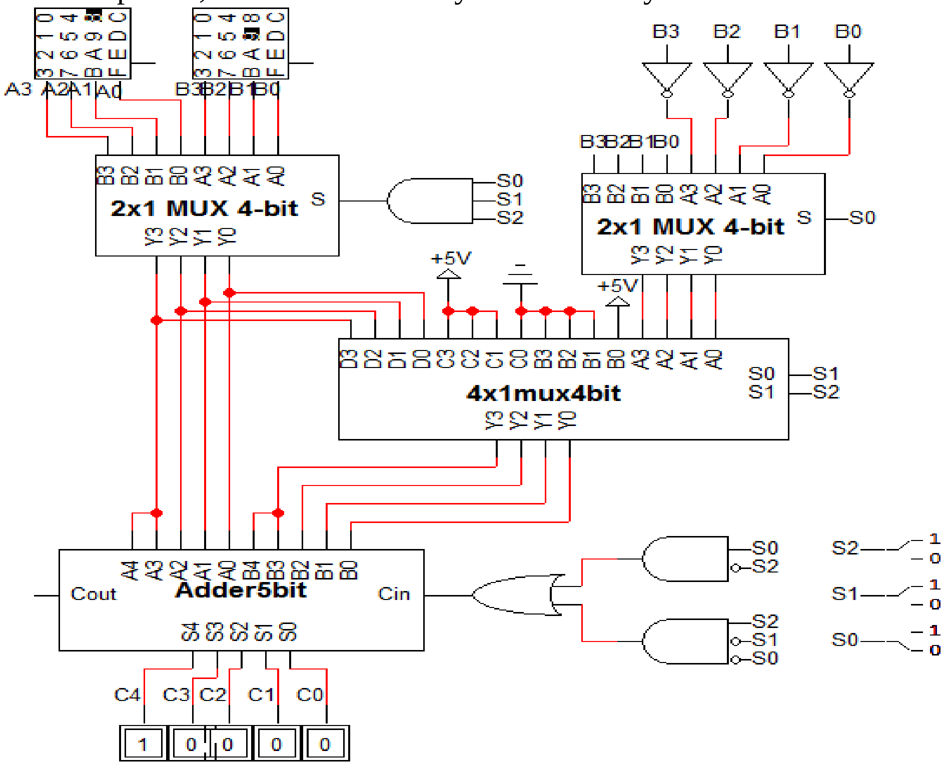
HW# 5 Solution

Due date: Thursday, April 7

Q.1. It is required to design a circuit that receives two **4-bit** signed numbers in 2's complement representation **A=$A_3A_2A_1A_0$, B=$B_3B_2B_1B_0$** and produces **<u>5-bit output</u> C= $C_4C_3C_2C_1C_0$**. The circuit implements the following functions based on the values of the three selection inputs: S1, S1 and S0.

S2 S1 S0	Function
0 0 0	C = A + B
0 0 1	C = A - B
0 1 0	C = A + 1
0 1 1	C = A + 2
1 0 0	C = A - 1
1 0 1	C = A - 2
1 1 0	C = 2A
1 1 1	C = 2B

(i) Show the block diagram design of your circuit using MSI components like Adder, Multiplexor, as needed. Use only one adder in your solution.



(ii) Model your design in Verilog by modeling each component separately i.e. adder, MUX, etc. and then instantiating these components to model your circuit.

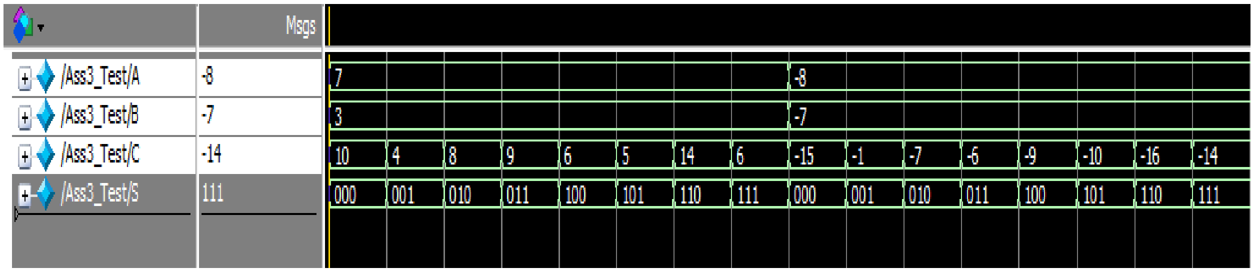
```
module Ass3 (input [3:0] A, B, input s2, s1, s0, output
[4:0]C);

wire sel;
wire [3:0] T1, T2, T3;

assign sel = s2 & s1 & s0;
mux2x1 #(4) M1 (A, B, sel, T1);
mux2x1 #(4) M2 (B, ~B, s0, T2);
mux4x1 #(4) M3 (T2, 4'b0001, 4'b1110, T1, s2, s1, T3);
assign Cin = s0 & ~s2 | s2 & ~s1 & ~s0;
adder  #(5) M4 (Cout, C, {T1[3], T1}, {T3[3], T3}, Cin);

endmodule

module mux4x1 #(parameter n = 1)(input  [n-1:0] a, b, c, d,
input s1, s0,  output reg [n-1:0] y);
      always@(s1, s0, a, b, c, d) begin
     case ({s1, s0})
          2'b00: y=a;
          2'b01: y=b;
          2'b10: y=c;
         2'b11: y=d;
      endcase
    end
endmodule

module mux2x1 #(parameter n = 1)(input  [n-1:0] a, b, input
select,  output [n-1:0] c);
     assign c = (select ? b : a);
endmodule

module adder #(parameter n = 4)
(output cout, output [n-1:0] sum, input [n-1:0] a, b, input
cin);
           assign {cout, sum} = a + b + cin;
endmodule
```

(iii) Write a Verilog test bench to test your design and verify its correctness by simulation. For each function, test at least 2 input combinations of your choice to demonstrate correct functionality. Show snapshots of your simulation to demonstrate its correctness.

```
module Ass3_Test();

     wire [4:0] C;

     reg [3:0] A, B;

  reg [2:0]  S;
```

```
    Ass3 M1 (A, B, S[2], S[1], S[0], C);

        initial begin

            A = 7; B = 3; S = 0;

            #10 S = 1;

            #10 S = 2;

            #10 S = 3;

            #10 S = 4;

            #10 S = 5;

            #10 S = 6;

            #10 S = 7;

            #10 A = -8; B = -7; S = 0;

            #10 S = 1;

            #10 S = 2;

            #10 S = 3;

            #10 S = 4;

            #10 S = 5;

            #10 S = 6;

            #10 S = 7;

        end

    endmodule
```

Simulation Results :



```
                Msgs
/Ass3_Test/A    -8      7                                                 -8
/Ass3_Test/B    -7      3                                                 -7
/Ass3_Test/C    -14     10   4    8    9    6    5    14   6    -15  -1   -7   -6   -9   -10  -16  -14
/Ass3_Test/S    111     000  001  010  011  100  101  110  111  000  001  010  011  100  101  110  111
```

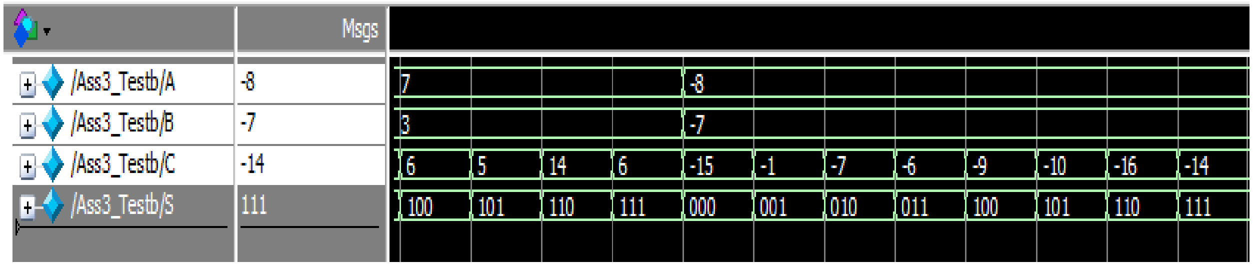
(iv) Write another Verilog model that models the circuit behaviorally. Make your model parametrizable where n is the inputs width in bits.

```
module Ass3Behav  #(parameter n=4)(input [n-1:0] A, B, input s2, s1, s0, output reg[n:0]C);

wire [n:0] A5, B5;
assign A5={A[n-1], A};
assign B5={B[n-1], B};

always @(s2, s1, s0, A, B) begin
case ({s2,s1,s0})
3'b000:    C = A5 + B5;
3'b001:    C = A5 - B5;
3'b010:    C = A5 + 1;
3'b011:    C = A5 + 2;
3'b100:    C = A5 - 1;
3'b101:    C = A5 - 2;
3'b110:    C = A5 + A5;  // or 2*A5
3'b111:    C = B5 + B5;  // or 2*B5
endcase
end
endmodule
```

(v) Use the test bench you developed in (iii) to test the correctness of your behavioral model. Show snapshots of your simulation to demonstrate its correctness.

```
module Ass3_Testb();

        wire [4:0] C;

        reg [3:0] A, B;

        reg [2:0]  S;


        Ass3Behav M1 (A, B, S[2], S[1], S[0], C);


        initial begin

                A = 7; B = 3; S = 0;

                #10 S = 1;

                #10 S = 2;

                #10 S = 3;

                #10 S = 4;

                #10 S = 5;

                #10 S = 6;

                #10 S = 7;

                #10 A = -8; B = -7; S = 0;
        #10 S = 1;

        #10 S = 2;

        #10 S = 3;

        #10 S = 4;

        #10 S = 5;

        #10 S = 6;

        #10 S = 7;

    end

endmodule
```

<u>Simulation Results</u> :



```
                   Msgs
/Ass3_Testb/A      -8     7                          -8
/Ass3_Testb/B      -7     3                          -7
/Ass3_Testb/C      -14    6    5    14   6    -15  -1   -7   -6   -9   -10  -16  -14
/Ass3_Testb/S      111    100  101  110  111  000  001  010  011  100  101  110  111
```

(vi) Implement your behavioral model on FPGA and demonstrate its correct functionality for n=2. Include a link for a video snapshot to demonstrate correct functionality of your circuit on FPGA.