COE 202, Term 142

Digital Logic Design

Assignment# 3 Solution

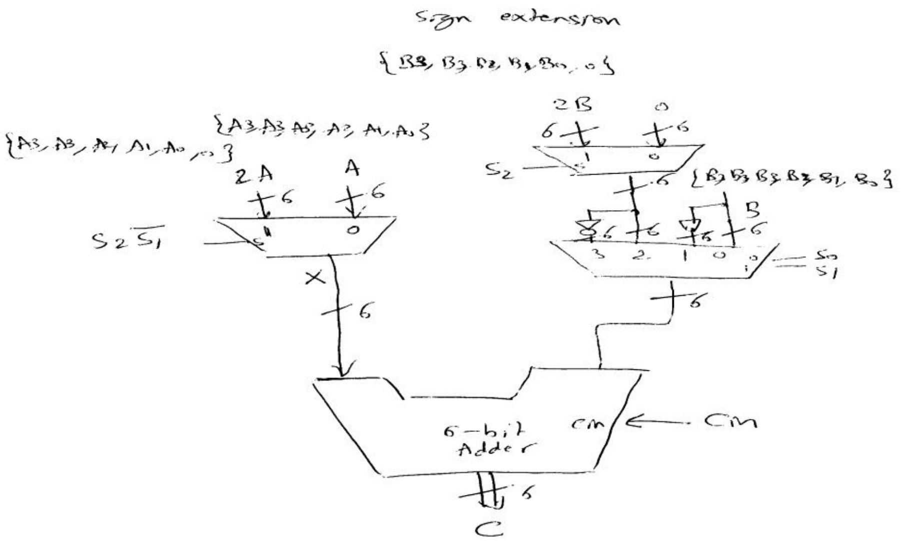
Due date: Thursday, April 16

Q.1. It is required to design a circuit that receives two 4-bit unsigned numbers **$A=A_3A_2A_1A_0$**, **$B=B_3B_2B_1B_0$** and produces **6-bit output** **$C= C_5C_4C_3C_2C_1C_0$**. 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 - 1
1 0 0	C = 2A+B
1 0 1	C = 2A-B
1 1 0	C = A+2B
1 1 1	C = A-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.

we assume the use of signed numbers



$S_2 \backslash S_1S_0$	00	01	11	10
0	0	1	0	1
1	0	1	1	0

$$C_{in} = \bar{S_1}S_0 + S_2S_0 + \bar{S_2}S_1\bar{S_0}$$

(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.

We show below the Verilog models of the various components needed in the solution:

2x1 MUX:

```
module mux2x1 #(parameter n=6) (input [n-1:0] a, b, input select,  output [n-1:0]  y);
             assign y = (select ? a : b);
endmodule
```

4x1 MUX:

```
module mux4x1 #(parameter n=6) (input [n-1:0] a, b, c, d, input [1:0] select,  output reg
[n-1:0] y);
      always@(select or a or b or c or d) begin
             case (select)
                  2'b00: y=d;
                  2'b01: y=c;
                  2'b10: y=b;
                  2'b11: y=a;
              endcase
     end
endmodule
```

Adder:

```
module adder #(parameter n = 6)
(output [n-1:0] sum, input [n-1:0] a, b, input cin);
             assign  sum = a + b + cin;
endmodule
```

Next we show the Verilog model for the whole ALU:

```
module ALU #(parameter n=4) (output [n+1:0] C, input [n-1:0] A, B, input [2:0] S);

wire [n+1:0] X, Y, Z, zero;
assign zero = 0;

assign cin = !S[1] & S[0] | S[2] & S[0] | !S[2] & S[1] & !S[0];

mux2x1 M1 ({A[n-1], A,1'b0},{A[n-1],A[n-1],A}, S[2] & !S[1], X);
mux2x1 M2 ({B[n-1],B,1'b0},zero, S[2], Y);
mux4x1 M3 (~Y, Y, ~{B[n-1],B[n-1],B}, {B[n-1],B[n-1],B}, S[1:0], Z);
adder  M4 (C, X, Z, cin);

endmodule
```

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

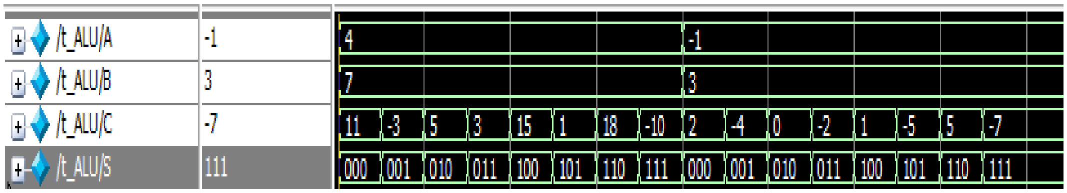
```
module t_ALU();

  wire   [5:0] C;
  reg     [3:0] A,  B;
  reg   [2:0] S;

   ALU  M1 (C, A, B, S);


initial begin
    A=4'b0100; B=4'b0111; S=3'b000;
    #10   S=3'b001;
    #10   S=3'b010;
    #10   S=3'b011;
    #10   S=3'b100;
    #10   S=3'b101;
    #10   S=3'b110;
    #10   S=3'b111;
    #10 A=4'b1111; B=4'b0011; S=3'b000;
    #10   S=3'b001;
    #10   S=3'b010;
    #10   S=3'b011;
    #10   S=3'b100;
    #10   S=3'b101;
    #10   S=3'b110;
    #10   S=3'b111;
  end
endmodule
```

The simulation snapshot is given below which demonstrates that the ALU is working properly:



/t_ALU/A
-1
4
-1
/t_ALU/B
3
7
3
/t_ALU/C
-7
11 -3 5 3 15 1 18 -10 2 -4 0 -2 1 -5 5 -7
/t_ALU/S
111
000 001 010 011 100 101 110 111 000 001 010 011 100 101 110 111