

COE 405, Term 152

Design & Modeling of Digital Systems

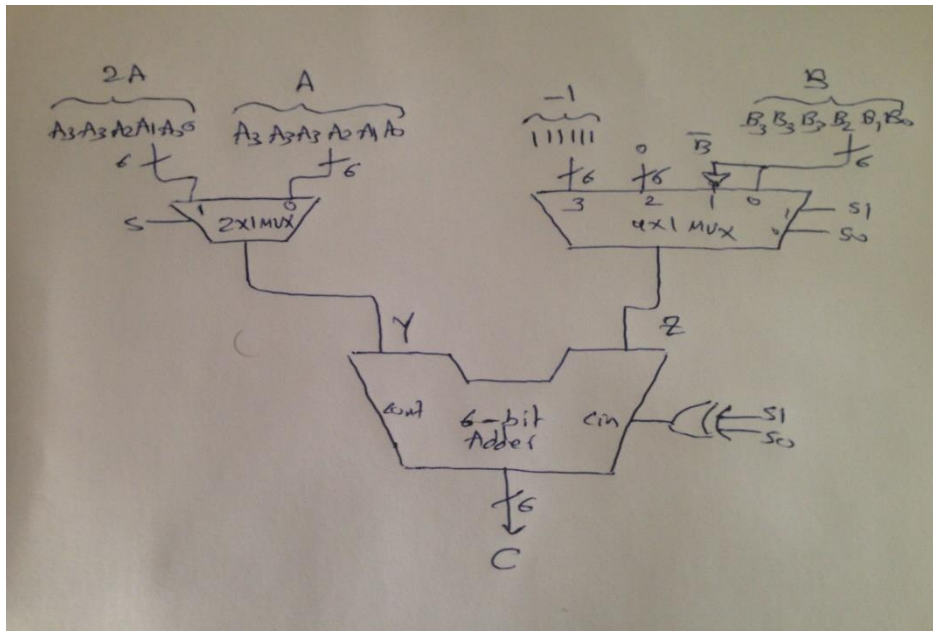
Quiz# 4

Date: Sunday, April 14, 2016

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 a **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: S_2, S_1 and S_0 .

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	$C = 2*A+B$
1 0 1	$C = 2*A-B$
1 1 0	$C = 2*A+1$
1 1 1	$C = 2*A-1$

(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 instantiate these components to model your circuit.

```
module Mux2x1 (output [5:0] y, input [5:0] a, b, input s);
```

```
    assign y = (s ? b : a);
```

```
endmodule
```

```
module Mux4x1(output reg [5:0] z, input [5:0] a, b, c, d, input s1, s0);
```

```
    always@(s1, s0, a, b, c, d) begin
```

```
        case ({s1, s0})
```

```
            2'b00: z=a;
```

```
            2'b01: z=b;
```

```
            2'b10: z=c;
```

```
            2'b11: z=d;
```

```
        endcase
```

```
    end
```

```
endmodule
```

```
module Adder (output [5:0] sum, input [5:0] a, b, input cin);
```

```
    assign sum = a + b + cin;
```

```
endmodule
```

```
module Quiz4 (output [5:0] C, input [3:0] A, B, input [2:0] S);
```

```
    wire [5:0] Y, Z;
```

```
    assign cin = S[0]^S[1];
```

```
    Mux2x1 M1 (Y, {A[3],A[3],A}, {A[3],A,1'b0}, S[2]);
```

```
    Mux4x1 M2 (Z, {B[3],B[3],B}, ~{B[3],B[3],B}, 6'b000000, 6'b111111,  
    S[1],S[0]);
```

```
    Adder M3 (C, Y, Z, cin);
```

```
endmodule
```

(iii) Write a Behavioral Verilog model to model the given circuit.

```
module Quiz4Behav (input [3:0] A, B, input s2, s1, s0, output reg[5:0]C);

wire [4:0] A5;
wire [5:0] A6, B6;
assign A5={ A[3], A };
assign A6={ A[3], A[3], A };
assign B6={ B[3], B[3], B };

always @* begin
    case ({s2,s1,s0})
        3'b000: C = A6 + B6;
        3'b001: C = A6 - B6;
        3'b010: C = A6 + 1;
        3'b011: C = A6 - 1;
        3'b100: C = 2*A5 + B6;
        3'b101: C = 2*A5 - B6;
        3'b110: C = 2*A5 + 1;
        3'b111: C = 2*A5 - 1;
    endcase
end
endmodule
```