Pep_Lab_06

In this Task, we're going to design an ALU module and write a test module for FPGA borad.

```
module alu(input [7:0] A, input [7:0] B, input [2:0] s,output reg [7:0] G, input cin, output reg cout);
    wire [3:0] ALU_Sel;
    reg [8:0] sol;
    assign ALU_Sel = {s,cin};
    reg [7:0] tmp;
always @(*) begin
    G = 7'b00000000;
    sol = 8'b000000000;
    cout = 0;
    if(ALU_Sel[3]) begin
        case(ALU_Sel[1:0])
            2'b00: G = A & B;
            2'b01: G = A | B;
            2'b10: G = A ^ B;
            2'b11: G = ~A;
            default:  G = 7'b00000000;
        endcase
    end else begin
        case(ALU_Sel[2:1])
            2'b00: tmp = 7'b0000000;
            2'b01: tmp = B;
            2'b10: tmp = ~B;
            2'b11: tmp = 7'b1111111;
            default: tmp = 7'b00000000;
        endcase
        sol = A + cin + tmp;
        cout = sol[8];
        G = sol[7:0];
    end
end
endmodule
```

```
module lab6(input [2:0] s,input cin,input[4:0] A,input[4:0] B, output [6:0] sibG,input clk, output reg [3:0] G, output reg M , output [6:0] sibM, output [6:0] sibcout);
    reg [7:0] tmpA, tmpB, tmpG;
    reg [3:0] tmpcout;
    wire [7:0] wire_G;
    reg cout;
    alu ab(tmpA, tmpB, s, wire_G, cin, cout);
    sib cd(G, sibG);
    minussib mi(M, sibM);
    sib aj(tmpcout, sibcout);
always @(*) begin
    //pre
    tmpA = {A[4],A[4],A[4],A};
    tmpB = {B[4],B[4],B[4],B};
    tmpcout = {3'b000,cout};
    //nach
    tmpG = wire_G;
    M = 1'b0;
    if(wire_G[4]) begin
        tmpG = (~wire_G)+1;
        M = 1'b1;
    end
    G = tmpG[3:0];
end
endmodule
```

In this design, we cannot use any register or latch for the ALU module, thus only one way we can store value is write in the test module and send it to the seven segment v file to render out the LED light.

The most difficult part I thought was let the mux do not generate a latch because when we didn't set the all case for determination in multiplexer, after compiling this module will generate a latch to store the status, which was undefined.

After finished this task, I learned how to design a ALU module with high-efficient way and how to combine 2 different part of module with the best way. I think that will be quite helpful for the next task to demonstrate a simple computer design.