CS223 – Digital Design

Ahmet Faruk Ulutaş

21803717

Section 1

Lab 5 Preliminary Work

A)

```
module RegisterFile(
    input clk,
    input logic writeEnable,
    input logic [2:0] writeAddress,
    input logic [2:0] readAddress1,
    input logic [2:0] readAddress2,
    input logic [3:0] writeData,
    output logic [3:0] readData1,
    output logic [3:0] readData2);

    logic [3:0] rf[7:0];

    always_ff @ (posedge clk)
        if (writeEnable)
            rf[writeAddress] <= writeData;

    assign readData1 = rf[readAddress1];
    assign readData2 = rf[readAddress2];
Endmodule
```

```
module ALU(
    input logic [1:0] Sel,
    input logic [3:0] a,
    input logic [3:0] b,
    output logic [3:0] Res);

    always_comb begin
        case (Sel)
            2'b00: Res = a + 1;
            2'b01: Res = a + b;
            2'b10: Res = a - b;
            2'b11: Res = a | b;
        endcase
    end
endmodule

module mux2to1_( input logic [3:0] d0, [3:0] d1, s, output logic [3:0] y);
    assign y = s ? d1 : d0;
Endmodule
```

B)

```
module testbench();
```

```
    logic clk;
    logic writeEnable;
    logic [2:0] writeAddress;
    logic [2:0] readAddress1;
    logic [2:0] readAddress2;
    logic [3:0] writeData;
    logic [3:0] readData1;
    logic [3:0] readData2;

    RegisterFile
dut(clk,writeEnable,writeAddress,readAddress1,readAddress2,write
Data,readData1,readData2);
    always
    begin
        clk = 1; #5; clk = 0; #5;
    end

initial begin
     writeEnable = 1;
     writeAddress = 3'b000; writeData = 4'b0000;
     readAddress1 = 3'b000; readAddress2 = 3'b000;
     #10;
     writeAddress = 3'b001; writeData = 4'b0001;
     readAddress1 = 3'b000; readAddress2 = 3'b001;
     #10;
     writeAddress = 3'b010; writeData = 4'b0010;
     readAddress1 = 3'b001; readAddress2 = 3'b010;
     #10;
     writeAddress = 3'b011; writeData = 4'b0011;
     readAddress1 = 3'b010; readAddress2 = 3'b011;
     #10;
     writeAddress = 3'b100; writeData = 4'b0100;
     readAddress1 = 3'b011; readAddress2 = 3'b100;
     #10;
     writeAddress = 3'b101; writeData = 4'b0101;
     readAddress1 = 3'b100; readAddress2 = 3'b101;
     #10;
     writeAddress = 3'b110; writeData = 4'b0110;
     readAddress1 = 3'b101; readAddress2 = 3'b110;
     #10;
     writeAddress = 3'b111; writeData = 4'b0111;
     readAddress1 = 3'b110; readAddress2 = 3'b111;
     #10;
    end
Endmodule

module alu_testbench();
    logic clk;
    logic [1:0] Sel;

    logic [3:0] a;

    logic [3:0] b;

    logic [3:0] Res;


    ALU dut( Sel, a, b, Res);


    always

    begin

       clk = 1; #5; clk = 0; #5;

    end


    initial begin

       a = 3'b000; b = 3'b000;

       Sel = 2'b00;

       #10;

       a = 3'b000; b = 3'b001;

       Sel = 2'b00;

       #10;

       a = 3'b001; b = 3'b000;

       Sel = 2'b00;

       #10;

       a = 3'b001; b = 3'b001;

       Sel = 2'b00;
```

```
#10;
a = 3'b000; b = 3'b000;
Sel = 2'b01;
#10;
a = 3'b000; b = 3'b001;
Sel = 2'b01;
#10;
a = 3'b001; b = 3'b000;
Sel = 2'b01;
#10;
a = 3'b001; b = 3'b001;
Sel = 2'b01;
#10;
a = 3'b000; b = 3'b000;
Sel = 2'b10;
#10;
a = 3'b000; b = 3'b001;
Sel = 2'b10;
#10;
a = 3'b001; b = 3'b000;
Sel = 2'b10;
#10;
a = 3'b001; b = 3'b001;
Sel = 2'b10;
```

```
        #10;
        a = 3'b000; b = 3'b000;
        Sel = 2'b11;
        #10;
        a = 3'b000; b = 3'b001;
        Sel = 2'b11;
        #10;
        a = 3'b001; b = 3'b000;
        Sel = 2'b11;
        #10;
        a = 3'b001; b = 3'b001;
        Sel = 2'b11;
        #10;
    end
Endmodule
```

C)

```
module toppDesign( input logic [3:0] EXTDATA, input logic isExternal,
input logic [2:0] AddrSrc1, input logic [2:0] AddrSrc2, input logic [2:0]
AddrDest, input logic writeEnable, input logic clk, input logic [1:0]
ALUSel, output logic [3:0] Rel);
   // logic writeEnable;
    logic [3:0] muxOut;
    logic [3:0] readData1;
   logic [3:0] readData2;


   mux2to1_ dut3( Rel, EXTDATA, isExternal, muxOut);
   //debouncer dut4( clk, pushButn, writeEnable);
   RegisterFile
dut(clk,writeEnable,AddrDest,AddrSrc1,AddrSrc2,muxOut,readData1
,readData2);
   ALU dut2( ALUSel, readData1, readData2, Rel);
   //SevenSegmentDisplay sevseg(clk, Rel, seg, dp, an);
Endmodule


module topDesignTestBench();
logic [3:0]EXTDATA;
logic isExternal;
logic [2:0]AddrSrc1;
logic [2:0]AddrSrc2;
logic [2:0]AddrDest;
logic pushButn;
logic clk;
logic [1:0]ALUSel;
logic [3:0] result;


toppDesign tD2 (EXTDATA, isExternal, AddrSrc1, AddrSrc2, AddrDest,
pushButn, clk, ALUSel, result);
```

```
always

begin

clk = 1; #5; clk = 0; #5;

end


initial begin

   pushButn = 1; isExternal = 1; EXTDATA = 4'b0011; AddrSrc1 = 3'b000; AddrSrc2 = 3'b000; AddrDest = 3'b000; ALUSel = 2'b01; #20;

   pushButn = 1; isExternal = 1; EXTDATA = 4'b0011; AddrSrc1 = 3'b001; AddrSrc2 = 3'b001; AddrDest = 3'b001; ALUSel = 2'b10; #20;

end


endmodule
```