1. As part of our study of the Smotherman machine, we will implement that machine in Verilog. Here are the top two modules in a Hierarchial implementation of the Smotherman machine. Your job in this assignment is to implement the controller using the interface provided below. We will work on this assignment in class on Friday.

```
module processor #(parameter wordSize=8, opSize=2, addrSize=6)(
    input clk, reset);

wire [wordSize-1:0] MDR;
wire [addrSize-1:0] MAR;
wire memReadEnable, memWriteEnable;

// CPU
CPU toyCPU(clk, reset, memWriteEnable, memReadEnable, MAR, MDR, MDR);

// memory
memory toyMem(clk, reset, memReadEnable, memWriteEnable, MAR, MDR, MDR);
endmodule

// Smotherman CPU
module CPU #(parameter wordSize=8, opSize=2, addrSize=6)(
    input clk, reset,
    output memwrite,
    output memread,
    output [addrSize-1:0] address,
    output [wordSize-1:0] storeData,
    input [wordSize-1:0] loadData);

wire [opSize-1:0] opCode;
wire [2:0] counterOut;

// single wire control signals
wire ACCin, ACCout, aluadd, IRin, IRout, MARin, MDRin, MDRout, acceq0;
wire PCin, PCout, pcinc, TEMPOut, timerReset;

// controller inputs: counter output, instruction op_code, zero_flag
// controller outputs: control signals in Fig 3
controller c(acceq0, opCode, counterOut,
    ACCin, ACCout, aluadd, IRin, IRout, MARin, MDRin, MDRout, acceq0,
    PCin, PCout, pcinc, memread, TEMPOut, memwrite, timerReset);

// contains: registers, ALU, and counter
datapath dp(clk, reset, acceq0, ACCin, ACCout, aluadd, IRin, IRout, MARin,
    MDRin, MDRout, PCin, PCout, pcinc, memread, TEMPOut, memwrite,
    timerReset, address, opCode, counterOut, storedata, loaddata);
```
module controller #(parameter opSize=2)(
    input  acceq0,
    input  [opSize-1:0] opCode,
    input  [2:0] counter,
    output  reg ACCin, ACCout, aluadd, IRin, IRout, MARin, MDRin,
    output  reg MDRout, PCin, PCout, pcinc, memread, TEMPout,
    output  reg memwrite, timerReset);

// your implementations goes here

endmodule

2. Write a testbench that verifies that the control signals produced by your controller are correct. Submit your verilog testbench module and and transcript file to Moodle.