Lab title

your names

March 6, 2019

1 Executive Summary

The Executive Summary section should be a single concise paragraph that describes the following items. Please note that this should be a paragraph, not an itemized list like I have below. The list is just to tell you what I'm looking for...it is not there for you to fill it in on your lab report. See the Lab 1 Example Report.pdf on Canvas to see a good example of what I'm looking for.

1. The goal of the lab
2. What modules you created and what they do. This should describe how the modules function and how they fit into the overall processor that we are building. For instance, for Lab 1, you will want to describe the operation of the register and the fact that it is used to store the program counter, which is used to keep track of which instruction to execute next.
3. Whether your lab was successful. If not successful, please state what is not currently working.

2 Test Report

To verify operation of this/these module(s), this lab requires N test bench(es). (You can just copy this statement and fill it in with the correct number of test benches)

1. Register Test Bench

Figure 1: Expected Results of the register test.

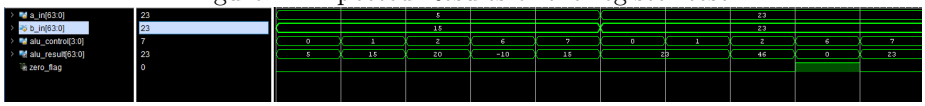
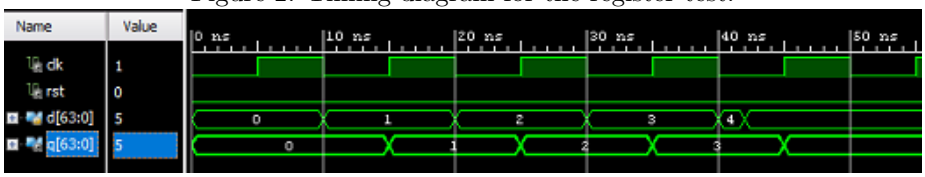


Figure 2: Timing diagram for the register test.



3 Code Appendix

Listing 1: Verilog code for testing a register.

```
`include "definitions.vh"
module ALU(
            input [`WORD-1:0] a_in,
            input [`WORD-1:0] b_in,
            input [3:0] alu_control,
            output reg [`WORD-1:0] alu_result,
            output reg zero_flag);

        always @(*)
         begin
              zero_flag <= (alu_result == 0);

              casex(alu_control)
                  `ALU_AND: begin
                       alu_result <= a_in & b_in;
                            end
                  `ALU_OR: begin
                       alu_result <= a_in | b_in;
                            end
                  `ALU_ADD: begin
                       alu_result <= a_in + b_in;
                            end
                  `ALU_SUB: begin
                       alu_result <= a_in - b_in;
                            end
```

```
                `ALU_PASS: begin
                    alu_result <= b_in;
                        end
            endcase
        end

endmodule
```

Listing 2: Verilog code for implementing a register.

```
`include "definitions.vh"

module register(
    input wire clk,
    input wire reset,
    input  wire [`WORD-1:0] D,
    output reg [`WORD-1:0] Q=`WORD'b0
    );

    always @(posedge(clk),posedge(reset))begin
        if (reset==1'b1)
            Q<=`WORD'b0;
        else
            Q <= D;
    end

endmodule
```