Lab title

your names

August 20, 2018

1 Executive Summary

The Executive Summary section should be a single concise paragraph that describes:

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 number of test benches.

1. Register Test Bench

2.1 Register Test Bench

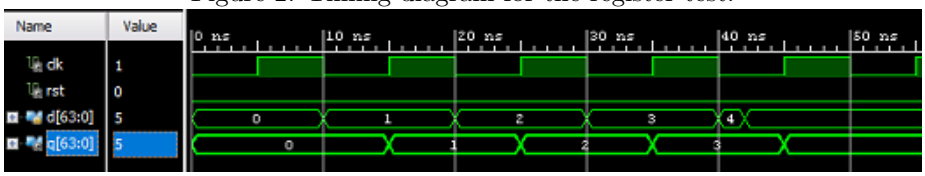
The register test bench contains:

1. Inputs
   (a) name_of_input - phrase describing the input
   (b) name_of_input - phrase describing the input
2. Outputs
   (a) name_of_output - phrase describing the output

Figure 1: Expected Results of the register test.

Time(ns)	0-10	10-15	15-20	20-25	25-30	30-35	35-40	40-42	42-45	45+
rst	0	0	0	0	0	0	0	0	0	0
d	0	1	1	2	2	3	3	4	5	5
q	0	0	1	1	2	2	3	3	3	5

Figure 2: Timing diagram for the register test.



Briefly describe what the testbench does and what it is testing. In something as simple as Lab 1, this might be as short as a few sentences sentences.

Operation of the testbench is verified by comparing the Simulation Results with the Expected Results Table (you can just include this sentence). Next you should state whether the module works as expected. If not, please describe what does not work and why it does not work.

3 Code Appendix

Listing 1: 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
```

Listing 2: Verilog code for testing a register.

```
`include "definitions.vh"


module register_test;

wire clk;
reg rst;
reg[`WORD - 1:0] d;
wire[`WORD - 1:0] q;

oscillator clk_gen(clk);

register UUT(
    .clk(clk),
    .reset(rst),
    .D(d),
    .Q(q)
    );

initial
begin
    rst = 0;
```

```
    d<='WORD'd0; #`CYCLE;
    d<='WORD'd1; #`CYCLE;
    d<='WORD'd2; #`CYCLE;
    rst = 1;
    d<='WORD'd3; #`CYCLE;
    d<='WORD'd4; #(`CYCLE/5);
    rst = 0;
    d<='WORD'd5; #(`CYCLE*4/5);
end

endmodule
```