# Lab title

your names

January 19, 2016

## 1 Introduction

Introduction with problem overview, your design procedure, and rationale. Be as brief as possible to let me know the big picture of the lab.

## 2 Interface

This section explains the input and output relationships of the design, so it can be treated as a black box. I have essentially provided you the inputs, outputs, and inouts, but I want to know you understand why. What inputs are used and how? What outputs are used and how? What sequence of commands need to be performed in order to operate the module? What form does the data get sent in (binary, two's compliment, excess code, etc.)? How are they grouped?

## 3 Design

This is the internal design of the item. Design description and explanation, including any pictures, charts, etc. This is the details of what you want to build, but not how to build it. In senior design terms, this is the specification.

## 4 Implementation

The Verilog code and explanations of why you implemented this way. There are many ways to implement a given design in Verilog. For instance why choose a case statement or ifs? Why did you trigger on a negedge verses any signal change? You should reference your code, for example, mine is in Listing 1 on page 1. Note you don't have to reference the page, I just wanted to show you how you could, and the power of a label in LaTeX .

Listing 1: Verilog code for implementing a register.

```
`include "definitions.vh"

module register(
```

```
    input clk,
    input reset,
    input  [`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
```

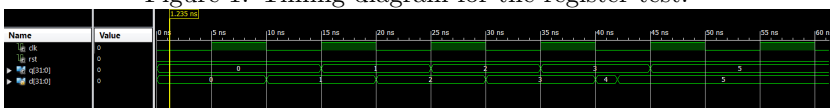
# 5 Test Bench Design

This is where you discuss the test benches you wrote, and what they were designed to test. You should discuss expected errors as well as random errors. Be sure to include your Verilog code of the testbench, for example, mine is in Listing 2 on page 2.

Listing 2: Verilog code for testing a register.

```
`include "definitions.vh"


module test_regs;

wire clk;
wire rst=0;
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
    d<=`WORD'd0; #`CYCLE;
```

Figure 1: Timing diagram for the register test.




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

# 6 Simulation

In this section you should show the results of your simulation, such as timing diagrams and explain any design issues you had to deal with. A sample timing diagram is in Figure 1 on page 3.

# 7 Conclusions

Overview the main points you want to stick in peoples minds and answer key questions you want to stick in peoples minds. Did it work? How well? What would you have done differently? What did you learn?