Lab 1

Steve Potter

August 28, 2019

1 Executive Summary

The goal of this lab was to build and test the Program Counter register for a 64-bit computer using the hardware description language Verilog. The Program Counter register is used to store the address of the next instruction that the computer will execute. The register takes a 64-bit input D and stores it in a 64-bit output Q on the rising edge of the clock signal. If the reset wire is set to 1, the value of Q will be forced to 0.

2 Test Report

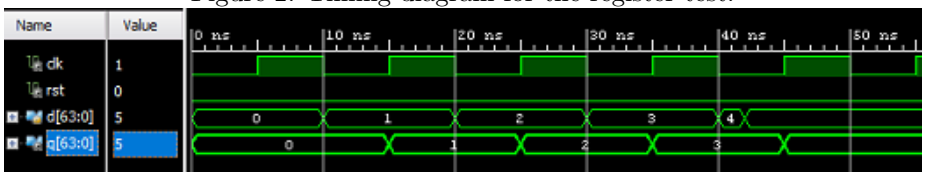
To verify operation of this module, this lab requires 1 test bench.

1. Register Test Bench

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.



3 Code Appendix

Listing 1: 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
```

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
```