Department of Electrical Engineering and Technology

College of Engineering and Technology

Mindanao State University – Iligan Institute of Technology

.

Laboratory Report

In partial fulfillment for the course

ECE 134.1 —Introduction to Digital VLSI Design Laboratory

Submitted to

Prof. Kevin O. Maglinte

Faculty, DEET

COET, MSU – IIT

Submitted by

Bryan Jeff C. Saycon

2018-0807

BS ECE – IV

INTRODUCTION

Verilog is a Hardware Description Language (HDL). It is a language used for describing a digital system like a network switch or a microprocessor or a memory or a flip−flop. It means, by using a HDL we can describe any digital hardware at any level. Designs, which are described in HDL are independent of technology, very easy for designing and debugging, and are normally more useful than schematics, particularly for large circuits. In this particular activity, we design a simple Up/Down Counter using Quartus Prime Lite Edition using the lessons discussed in the Laboratory.

OBJECTIVES

Use Quartus Prime Lite Edition and ModelSim in coding (in Verilog HDL format), compiling, and simulation of a system.
Properly design an up/down counter.
Discuss the simulation process using screen record and voice-over.

COUNTER VERILOG CODE

```
1   /*************************/
2   /* Laboratory 1 - Counter */
3   /*************************/
4
5   module counter_top(clk,
6                      rst_n,
7                      mode,
8                      in,
9                      flag,
10                     flag_count,
11                     count);
12  input clk;
13  input rst_n;
14  input mode;
15  input [7:0] in;
16
17  output [7:0] count;
18  output flag;
19  output [7:0] flag_count;
20
21  wire clk;
22  wire rst_n;
23  wire mode;
24  wire [7:0] in;
25  wire flag;
26  wire zero;
27  wire flag_reset;
28
29  reg [7:0] flag_count;
30  reg [7:0] count;
```

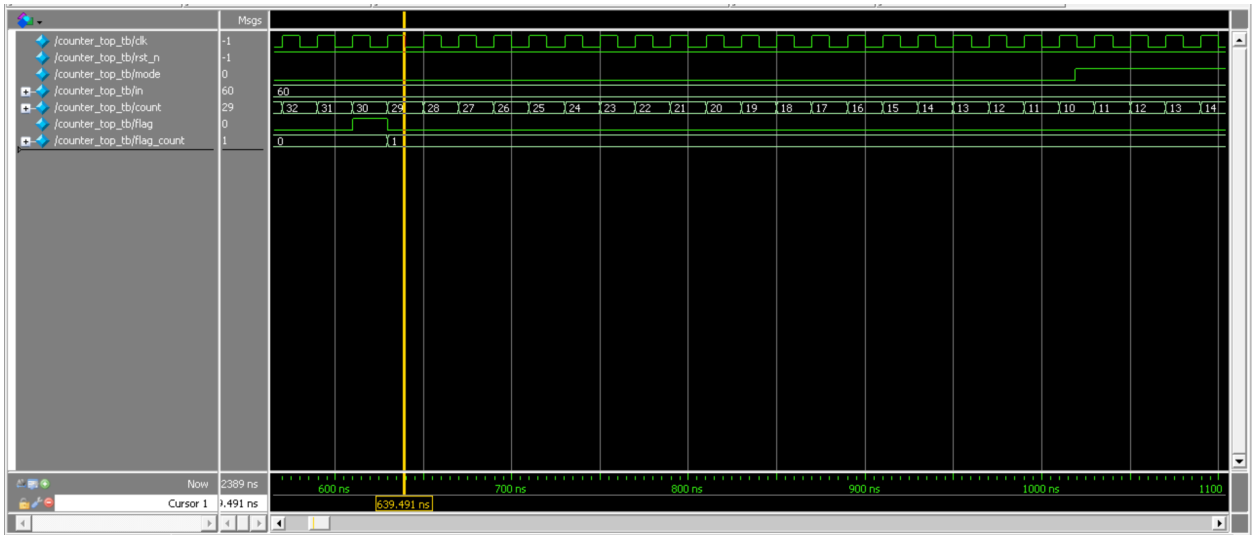
```
31
32  always @(posedge clk)
33    begin
34      if (!rst_n)
35        count <= in;
36      else
37        if (mode==1) //Up counter mode
38          if(count==60)
39            count <= 0;
40          else
41            count <= count+1;
42        else //Down counter mode
43          if(zero)
44            count<=in;
45          else
46            count <= count-1;
47    end
48  always @(posedge clk)
49    begin
50      if (!rst_n)
51        flag_count <= 0;
52      else
53        if (flag_reset)
54          flag_count <= 0;
55        else
56          flag_count <= flag_count + flag;
57    end
58
```

```
58
59  assign flag = (count == 8'd30);
60  assign zero = (count == 8'd0);
61  assign flag_reset = (flag_count == 8'd100);
62
63  endmodule
```

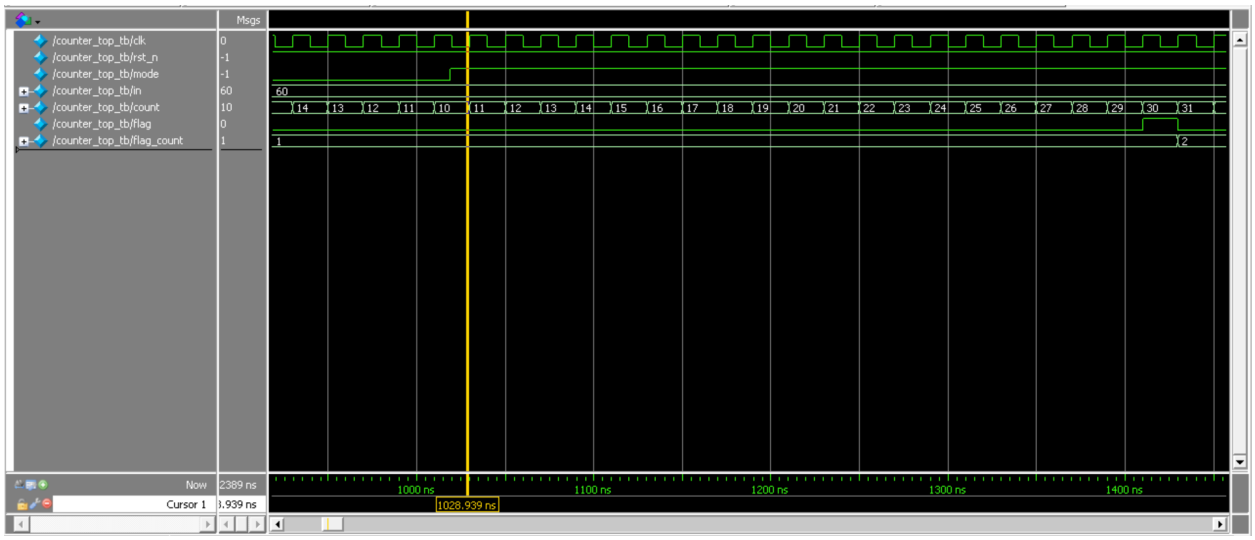
TEST BENCH CODE

```
/* TEST BENCH */
`timescale 1ns/1ps
module counter_top_tb ();

reg clk;
reg rst_n;
reg mode;
reg [7:0] in;

wire [7:0] count;
wire flag;
wire [7:0] flag_count;

initial
begin
clk = 0;
rst_n = 1;
mode = 0;
in = 8'd60;
#2 rst_n = 0;
#17 rst_n = 1;
#1000 mode = 1;
#1500 mode = 0;
#2500 rst_n = 0;
#150 rst_n = 1;
#3000 mode = 1;
#3050 rst_n = 0;
#170 rst_n = 1;
#5000 mode = 0;
#5000 mode = 0;
#6000 $finish;
end

always #10 clk = ~clk;

counter_top dut(.clk(clk),
                .rst_n(rst_n),
                .mode(mode),
                .in(in),
                .flag(flag),
                .flag_count(flag_count),
                .count(count));
endmodule
```

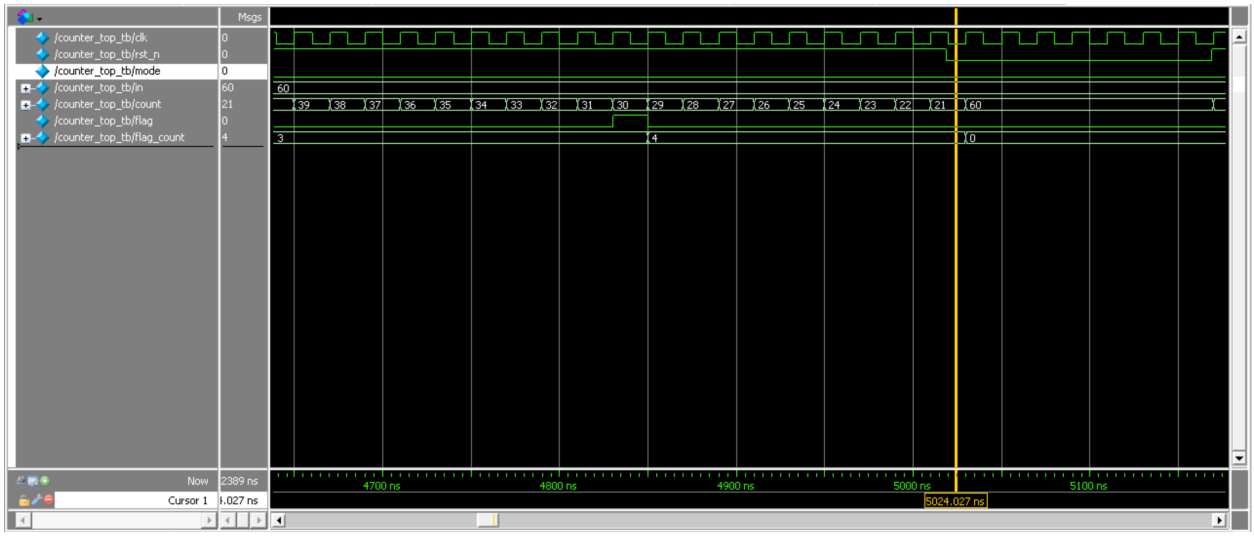
RESULTS



The screenshot above shows the first flag count of the clock since the count touches 30 that raises the flag count to 1.



The screenshot above shows the transition from a down counter to up counter since we wrote in the test bench to raise the mode to logic 1 after 1000 time units where the count transitioned from decreasing to increasing.



The screenshot above shows the code triggers rst_n function that resets the flag count and count counters.

CONCLUSION

This activity is about running an Up/Down Counter Verilog HDL using Quartus Prime Lite and ModelSim. The program starts at a Down counter where the count starts from 60 and the flag starts to count when it passes 30 and just goes from 0 to 60 when there is no command in the clock and switches to Up counter when the mode is turned on where it then starts counts in increasing manner. And the flag resets to zero whenever the rst_n function is triggered. The program functioned alright and there are no errors in the code.