**Lab Number: 7**

**Section Number: 001**

**Names: Barak Barclay**

**Assigned Date: 03/29/2016**

**Due Date: 04/07/2016**

**Introduction:**

The lab is to construct a 2-bit up/down counter in Verilog using ModelSim. The program will have a test bench that originally resets the counter then, starts counting up and down indefinitely. The module will also show the state the counter is in whenever it switches states in the transcript display.

**Part 1:**

// 2 bit up/down counter

module L7(output A,B, output [1:0]State, input dir,clock,reset);

reg[1:0]state; //state is neither an input or output and must not be declared in module line

parameter S0=2'b00, S1=2'b01, S2=2'b10, S3=2'b11; //assigns values of S0-S3 prior to the always statement

always @(posedge clock or negedge reset)

if(reset==0) state <=S0; //non-blocking syntax used here

else case (state)

S0: if(dir) state=S1; else state=S3;

S1: if(dir) state=S2; else state=S0;

S2: if(dir) state=S3; else state=S1;

S3: if(dir) state=S0; else state=S2;

endcase

assign A=state[1];

assign B=state[0];

assign State[1]=state[1];

assign State[0]=state[0];

endmodule

// 2 bit up/down counter testbench

module L7TB;

wire A,B;

wire [1:0]State;

reg dir,clock,reset;

L7 M1(A,B,State,dir,clock,reset);

initial

begin

$monitor ("AB=%b%b", A,B);

dir = 1'b1;clock = 1'b0;reset = 1'b0;

#5 dir = 1'b1;clock = 1'b0;reset = 1'b1;

forever begin

#25 dir = 1'b1;clock = 1'b1;reset = 1'b1;

#25 dir = 1'b1;clock = 1'b0;reset = 1'b1;

#25 dir = 1'b1;clock = 1'b1;reset = 1'b1;

#25 dir = 1'b1;clock = 1'b0;reset = 1'b1;

#25 dir = 1'b1;clock = 1'b1;reset = 1'b1;

#25 dir = 1'b1;clock = 1'b0;reset = 1'b1;

#25 dir = 1'b1;clock = 1'b1;reset = 1'b1;

#25 dir = 1'b0;clock = 1'b0;reset = 1'b1;

#25 dir = 1'b0;clock = 1'b1;reset = 1'b1;

#25 dir = 1'b0;clock = 1'b0;reset = 1'b1;

#25 dir = 1'b0;clock = 1'b1;reset = 1'b1;

#25 dir = 1'b0;clock = 1'b0;reset = 1'b1;

#25 dir = 1'b0;clock = 1'b1;reset = 1'b1;

#25 dir = 1'b0;clock = 1'b0;reset = 1'b1;

#25 dir = 1'b0;clock = 1'b1;reset = 1'b1;

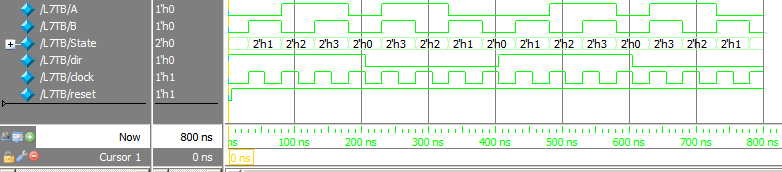
#25 dir = 1'b1;clock = 1'b0;reset = 1'b1;

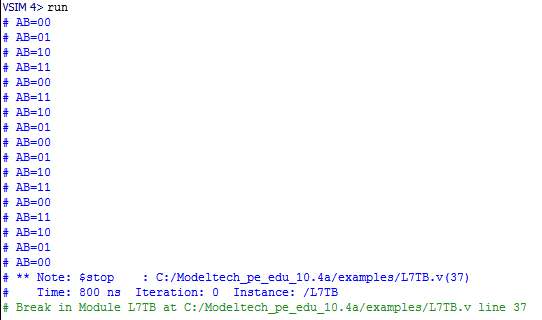
end

end

initial #800 $stop; //change # to length of above

endmodule





**Conclusion:**

The first module is the 2-bit counter like the module from lecture 14. The second module is the test bench which uses a forever loop to keep the counter counting up and down indefinitely until the module itself is told to stop at 800ns. The code $monitor ("AB=%b%b", A,B); monitors any change of the state and displays it to the transcript window when it notices a change.