**COUNTER:**

module updown\_counter\_4bit (input clock, reset, count\_up, output reg [3:0]count);

always @(posedge clock or posedge reset)

begin

if (reset)

count = 4'b0000;

else

if (count\_up)

count = count + 1;

else

count = count - 1;

end

endmodule

**TESTBENCH:**

module Lab3\_Testbench();

reg clock, reset, count\_up;

wire [3:0] count;

updown\_counter\_4bit counter (clock, reset, count\_up, count);

initial begin

clock = 0;

reset = 1;

count\_up = 1;

forever #5 clock = ~clock;

end

initial begin

#10 reset = 0;

#120 reset = 1;

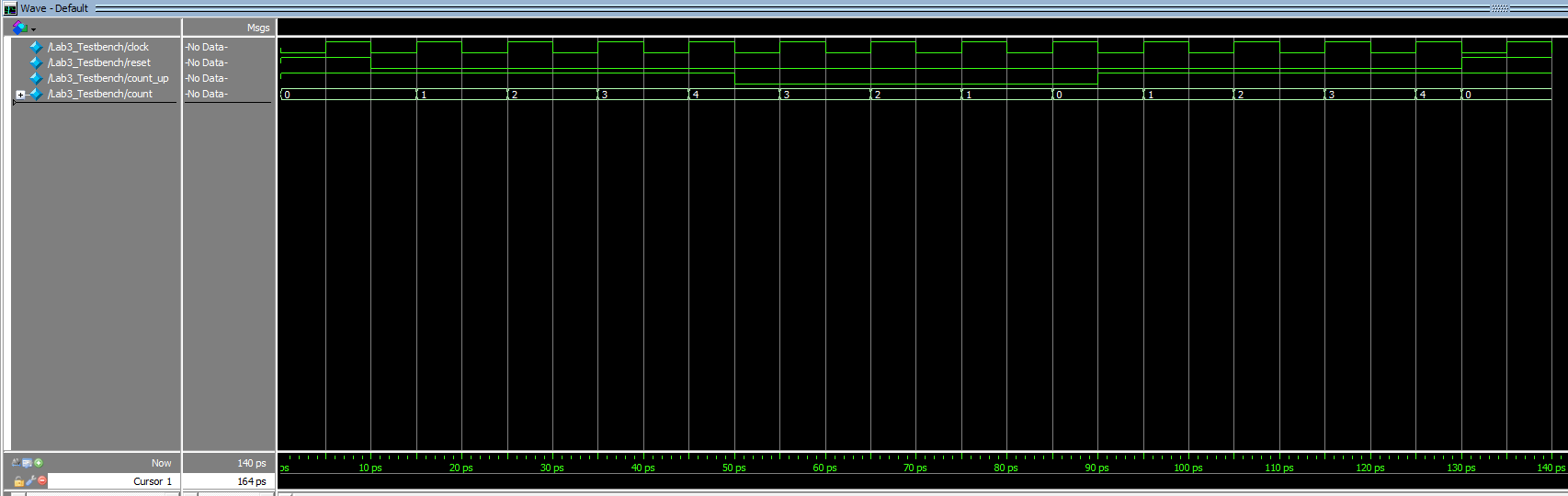
end

initial begin

#50 count\_up = 0;

#40 count\_up = 1;

end

endmodule

