

Tyler Youk HW #10

//up_down_counter source code

```
module up_down_counter #(
    parameter N = 4
) (
    input logic    clk,
    input logic    en_b,
    input logic    load_b,
    input logic    up,
    input logic [N-1:0] load_in,
    output logic [N-1:0] q,
    output logic    rco_b
);
    always_ff @(posedge clk) begin
        if (~en_b) begin
            if (load_b) begin
                //using non-blocking assignment to assign q value
                q <= up ? q+1 : q-1;
            else
                q <= load_in;
            end
        else begin
            q <= load_b;
        end
    end
//end

    assign rco_b = ~(up & (&q)) | (~up & ~(|q));

endmodule
```

//testbench source code

```
module testbench_hw7 ();

    logic clk, en_b;
    logic [3:0] load_in4, q4;
    logic [4:0] load_in5, q5;
    logic rco_b4, rco_b5;

    up_down_counter #(
        .N(4)
    ) counter4 (
        .clk  (clk),
```

```

        .en_b (en_b),
        .load_b (load4_b),
        .up (up),
        .load_in(load_in4),
        .q (q4),
        .rco_b (rco4_b)
    );

    up_down_counter #(
        .N(5)
    ) counter5 (
        .clk(clk),
        .en_b(en_b),
        .up(up),
        .load_b(load5_b),
        .load_in(load_in5),
        .q(q5),
        .rco_b(rco_b5)
        .rco
    );

    initial begin
        clk = 1'b0;
        en_b = 1'b1;
        load4_b = 1'b0;
        load5_b = 1'b0;
        up = 1'b1;
        load_in4 = 4'b0000;
        load_in5 = 5'b00000;

        forever #5 clk = ~clk;
    end

    initial begin
        #10;
        load4_b = 1'b0;
        load5_b = 1'b0;
        #10;
        en_b = 1'b0;
        #10;
        load4_b = 1'b1;
        load5_b = 1'b1;
        #320;
        en_b = 1'b1;
        up = 1'b0;
        load4_b = 1'b0;
        load5_b = 1'b0;
        load_in4 = 4'b1111;
    end

```

```
load_in5 = 5'b11111;  
#10;  
en_b = 1'b0;  
#10;  
load4_b = 1'b1;  
load5_b = 1'b1;  
en_b = 1'b1;  
#10;  
en_b = 1'b0;  
#320;  
load4_b = 1'b0;  
load5_b = 1'b0;  
load_in4 = 4'b1010;  
load_in5 = 5'b01010;  
#10;  
load4_b = 1'b1;  
load5_b = 1'b1;  
#10;  
up = 1'b1;  
load4_b = 1'b0;  
load5_b = 1'b0;  
load_in4 = 4'b0101;  
load_in5 = 5'b10101;  
#10;  
load4_b = 1'b1;  
load5_b = 1'b1;  
#10;  
$finish();  
end
```

endmodule

Waveform Screenshot

