

```

timescale 1ns / 1ps
/*****
* Module: debounce_top
*
* Author: Noah Hanks
* Class: ECEN 220, Section 3, Fall 2020 - ECEN 220, Section 1, Winter 2020 * Date: 3
November 2020
*
* Description: Uses the debounce module to connect the buttons and displays to the
counters
*
*
*****/
`default_nettype none

module debounce_top(
    input wire logic clk, btneu, btnc,
    output logic[3:0] anode,
    output logic[7:0] segment
);

    logic[3:0] digit0, digit1, digit2, digit3;
    logic pressed, ffPressed, undebounced, debounced, notDebounced;

    //these flip flops synchronize the input
    always_ff @(posedge clk)
        pressed <= btnc;
    always_ff @(posedge clk)
        ffPressed <= pressed;

    debounce DB0(.debounced(debounced), .clk(clk), .reset(btneu), .noisy(ffPressed));

    logic oneShotDebounced, oneShotUndebounced;
    // this always ff block detects the one shot
    always_ff @(posedge clk) begin
        notDebounced <= debounced;
        undebounced <= ffPressed;
    end

    assign oneShotDebounced = (debounced && ~notDebounced);
    assign oneShotUndebounced = (ffPressed && ~undebounced);

    logic inc1, inc2, inc3, inc4;
    mod_counter #(10, 4) MC1(.clk(clk), .reset(btneu), .increment(oneShotDebounced),
.rolling_over(inc1), .count(digit0));
    mod_counter #(10, 4) MC2(.clk(clk), .reset(btneu), .increment(inc1),

```

```
.rolling_over(inc3), .count(digit1));
    mod_counter #(10, 4) MC3(.clk(clk), .reset(btnu),
.increment(oneShotUndebounced), .rolling_over(inc2), .count(digit2));
    mod_counter #(10, 4) MC4(.clk(clk), .reset(btnu), .increment(inc2),
.rolling_over(inc4), .count(digit3));

    SevenSegmentControl SSC0 (.segment(segment), .anode(anode), .reset(btnu),
.clk(clk), .dataIn({digit3, digit2, digit1, digit0}), .digitPoint(4'b0000),
.digitDisplay(4'b1111));

endmodule
```