```
`timescale 1ns / 1ps
/**********************************************
**********************************
*
* Module: debounce_top
*
* Author: Eric Christie
* Class: ECEN 220, Section 1, Winter 2021
* Date: 3/16/21
*
* Description: Synchronizer fed to
debouncer to One Shot to the final counter
*                      Also has an
un-debounced line from the synchronizer
for contrast
*
**********************************************
**********************************/

`default_nettype none
module debounce_top(
    input wire logic clk, btnu, btnc,
    output logic [3:0] anode,
    output logic [7:0] segment);
```

```
    logic FF1, FF2; //synchronizer flip
flops
    logic debounced; //output from
debouncer
    logic dOS_FF, dOS, rOS_FF, rOS; //One
shot flip-flop and output for debouncer
and raw
    logic [7:0] debCount, rawCount;
//counters

    //Synchronizer
    always_ff @(posedge clk) begin
        FF1 <= btnc;
        FF2 <= FF1;
    end

    //debouncer - clk, reset, noisy,
debounced (output)
    debounce db(clk, btnu, FF2, debounced);

    //One Shot
    always_ff @(posedge clk) dOS_FF <=
debounced;
```

```
    assign dOS = !dOS_FF && debounced;

    //debounced counter
    always_ff @(posedge clk) begin
        if(btnu) debCount <= 0;
        else if(dOS) debCount <= debCount
+1;
    end

    //non-debounced One Shot and Counter
    always_ff @(posedge clk) rOS_FF <= FF2;

    assign rOS = !rOS_FF && FF2;

    always_ff @(posedge clk) begin
        if(btnu) rawCount <= 0;
        else if(rOS) rawCount <= rawCount
+1;
    end


    //Seven Segment Display clk, reset,
dataIn, digitDisplay, digitPoint, anode,
```

```
segment
    SevenSegmentControl SSC(clk, btnu,
{rawCount, debCount}, 4'b1111, 4'b0000,
anode, segment);

endmodule
```