```
1   `timescale 1ns / 1ps
2   /**********************************************************************
3    * File Name: counterTopLevel.v
4    * Project: Counter using TramelBlaze
5    * Designer: Marc Cabote
6    * Email: marcdominic011@gmail.com
7    * Rev. Date: 8 February, 2018
8    *
9    * Purpose: This project is a 16-bit counter that either counts up or down
10   *          depending on the uphdnl switch(sw0) on the board. The counting
11   *          is then displayed on the seven segment display(only using the
12   *          4 least significant, and the 4 most significant set to
13   *          constantly display zeros. The reset button is asynchronous
14   *          for the aiso but is synchronous throught the rest of the
15   *          module.
16   *
17   *          This project is a modification of the first project from last
18   *          semester. This time, the TramelBlaze replaced the counter.
19   *          Tramelblaze ISR reads switch to determine if the counter should
20   *          count up or down. Then write the updated count to the register
21   *          driving the seven segment display.
22   *
23   * Notes:   -  This module has an asynchronous reset input.
24   *          -  Count up when switch is high count down when switch is low
25   *          -  Debounce button is located at button down
26   *          -  Reset is button up
27   *          -  TramelBlaze uses the PicoBlaze ISA
28   *
29   **********************************************************************/
30  module counterTopLevel(input rst , clk ,dbnSw, uphdnl,
31                         output a7, a6, a5, a4, a3, a2, a1, a0,
32                                a, b, c, d, e, f, g);
33
34    wire rstOut, dbn, pulse, int_ack, srOut, wen, ren, load;
35    wire [15:0] port_id, outPort_count, count;
36
37    assign load =(port_id == 16'h0001) && wen;//to load reg16
38
39    aiso              m0(.clk(clk), .rst(rst), .rst_out(rstOut));
40                      //rstOut to wire all resets from AISO
41
42    debounce          m1(.clk(clk), .rst(rstOut), .sw(dbnSw),
43                         .db(dbn));
44                         //dbn to wire debounce to PED
45
46    ped               m2(.clk(clk), .rst(rstOut), .signal(dbn),
47                         .pulse(pulse));
48                         //pulse to wire ped to srflop
49
50    srflop            m3(.clk(clk), .rst(rstOut), .s(pulse), .r(int_ack),
51                         .srOut(srOut));
52                         //flop to tramelblaze
53
54    tramelblaze_top   m4(.CLK(clk), .RESET(rstOut), .IN_PORT({15'b0,uphdnl}),
55                         .INTERRUPT(srOut), .OUT_PORT(outPort_count),
56                         .PORT_ID(port_id), .READ_STROBE(ren),
57                         .WRITE_STROBE(wen), .INTERRUPT_ACK(int_ack));
```

```
58                          //ren and wen to write enable
59                          //srOut to wire sr
60                          //outport_count to wire tramelblazeTop to reg16
61
62      reg16             m5( .clk(clk), .rst(rstOut), .ld(load),
63                            .d(outPort_count), .q(count));
64                          //count to wire reg16 to displayController
65
66      displayController m6(.clk(clk), .rst(rstOut),
67                          .seg7(4'b0), .seg6(4'b0), .seg5(4'b0), .seg4(4'b0),
68                          .seg3(count[15:12]), .seg2(count[11:8]),
69                          .seg1(count[7:4]), .seg0(count[3:0]),
70                          .a7(a7), .a6(a6), .a5(a5), .a4(a4),
71                          .a3(a3), .a2(a2), .a1(a1), .a0(a0),
72                          .a(a), .b(b), .c(c), .d(d), .e(e), .f(f), .g(g));
73
74    endmodule
75
```