```
1   `timescale 1ns / 1ps
2   /**********************************************************************
3    * File Name: Tx_top.v
4    * Project: Tx Machine
5    * Designer: Marc Cabote
6    * Email: marcdominic011@gmail.com
7    * Rev. Date: 10 March, 2018
8    *
9    * Purpose:The transmit engine has the logic needed to communicate with a
10   *         computer using UART. For the terminal to have a stable connection
11   *         in order to output the proper characters, the program—Realterm—
12   *         should know what the setting is for the transmit engine—it should
13   *         know the baud rate, odd or even parity and number of bits.
14   *         If the program settings match the FPGA, it will then show
15   *         "CSULB CECS 460 - [LINECOUNTER] <CR> <LF>".
16   *
17   * Notes:   -  This module has an asynchronous reset input.
18   *
19   **********************************************************************/
20  module Tx_top(input clk, rst, eight, pen, ohel,
21                input [3:0] baud,
22               output [15:0] reads, writes,
23               output [15:0] leds,
24               output Tx);
25
26        wire load;
27        wire rst_out; //aiso wire
28        wire TxRdy;   //Tx wire
29        wire ped_out; //ped wire
30        wire sr_out;  //sr flop wire
31        wire interrupt_ack, write_strobe, read_strobe;//TB wires
32        wire [15:0] port_id, in_port;
33        wire [15:0] out_port;
34
35        // assign load
36        assign load = (port_id == 16'h0000) & (write_strobe);
37        //leds for debugging
38        assign leds = out_port[15:0];
39
40        aiso           aiso (.clk(clk),
41                             .rst(rst),
42                             .rst_out(rst_out));
43
44        transmitEngine   tx (.clk(clk),
45                             .rst(rst_out),
46                             .load(load),
47                             .eight(eight),
48                             .pen(pen),
49                             .ohel(ohel),
50                             .out_port(out_port[7:0]),
51                             .baud(baud),
52                             .TxRdy(TxRdy),
53                             .Tx(Tx));
54
55        ped              ped(.clk(clk),
56                             .rst(rst_out),
57                             .signal(TxRdy),
```

```
58                              .pulse(ped_out));
59
60          srflop          srflop(.clk(clk),
61                              .rst(rst_out),
62                              .s(ped_out),
63                              .r(interrupt_ack),
64                              .srOut(sr_out));
65
66          tramelblaze_top    TB(.CLK(clk),
67                              .RESET(rst_out),
68                              .IN_PORT(in_port),
69                              .INTERRUPT(sr_out),
70                              .OUT_PORT(out_port),
71                              .PORT_ID(port_id),
72                              .READ_STROBE(read_strobe),
73                              .WRITE_STROBE(write_strobe),
74                              .INTERRUPT_ACK(interrupt_ack));
75
76
77    endmodule
78
```