```
1   `timescale 1ns / 1ps
2   /*********************************************************************************
3    * File Name: transmitEngine_tf.v
4    * Project: Tx Machine
5    * Designer: Marc Cabote
6    * Email: marcdominic011@gmail.com
7    * Rev. Date: 10 March, 2018
8    *
9    * Purpose: To verify the transmit engine
10   *
11   * Notes:   -  This module has an asynchronous reset input.
12   *
13   *********************************************************************************/
14  module transmitEngine_tf;
15     // Inputs
16     reg clk;
17     reg rst;
18     reg eight;
19     reg pen;
20     reg ohel;
21     reg load;
22     reg [7:0] out_port;
23     reg [3:0] baud;
24
25     // Outputs
26     wire TxRdy;
27     wire Tx;
28
29     // Instantiate the Unit Under Test (UUT)
30     transmitEngine uut (
31        .clk(clk),
32        .rst(rst),
33        .eight(eight),
34        .pen(pen),
35        .ohel(ohel),
36        .load(load),
37        .out_port(out_port),
38        .baud(baud),
39        .TxRdy(TxRdy),
40        .Tx(Tx)
41     );
42
43     //Generate clock
44     always #5 clk = ~clk;
45
46     initial begin
47        // Initialize Inputs
48        clk   = 0;
49        rst   = 1;         //exercise reset
50        eight = 0;
51        pen   = 0;
52        ohel  = 0;
53        load  = 0;
54        out_port = 8'hA5;  //test for 1010_0101
55        baud     = 4'b1011;//test for fastest baud
56
57        //Wait 100 ns for global reset to finish
```

```
 58          #100;
 59          rst = 0;            //exercise reset
 60
 61          //=====================================
 62          // case = 3'b000
 63          // output = 11_0100101_01
 64          //=====================================
 65          eight = 0;
 66          pen   = 0;
 67          ohel  = 0;
 68          load  = 1;
 69
 70          //=====================================
 71          // case 3'b001
 72          // output = 11_0100101_01
 73          //=====================================
 74          #100;
 75          eight = 0;
 76          pen   = 0;
 77          ohel  = 1;
 78          load  = 1;
 79
 80          //=====================================
 81          // case 3'b010
 82          // output = 11_0100101_01
 83          //=====================================
 84          #100;
 85          eight = 0;
 86          pen   = 1;
 87          ohel  = 0;
 88          load  = 1;
 89
 90          //=====================================
 91          // case 3'b011
 92          // output = 10_0100101_01
 93          //=====================================
 94          #100;
 95          eight = 0;
 96          pen   = 1;
 97          ohel  = 1;
 98          load  = 1;
 99
100          //=====================================
101          // case 3'b100
102          // output = 11_0100101_01
103          //=====================================
104          #100;
105          eight = 1;
106          pen   = 0;
107          ohel  = 0;
108          load  = 1;
109
110          //=====================================
111          // case 3'b101
112          // output = 11_0100101_01
113          //=====================================
114          #100;
```

```
115         eight = 1;
116         pen   = 0;
117         ohel  = 1;
118         load  = 1;
119
120         //=====================================
121         // case 3'b110
122         // output = 01_0100101_01
123         //=====================================
124         #100;
125         eight = 1;
126         pen   = 1;
127         ohel  = 0;
128         load  = 1;
129
130         //=====================================
131         // case 3'b111
132         // output = 11_0100101_01
133         //=====================================
134         #100;
135         eight = 1;
136         pen   = 1;
137         ohel  = 1;
138         load  = 1;
139
140      end
141
142   endmodule
143
144
```