```
1   `timescale 1ns / 1ps
2
3   ////////////////////////////////////////////////////////////////////////////////
4   // Company:
5   // Engineer:
6   //
7   // Create Date:   17:04:33 04/15/2018
8   // Design Name:   receiveEngine
9   // Module Name:   F:/Spring2018/CECS460/project3/FullUART/receiveEngine_tf.v
10  // Project Name:  transmitEngine
11  // Target Device:
12  // Tool versions:
13  // Description:
14  //
15  // Verilog Test Fixture created by ISE for module: receiveEngine
16  //
17  // Dependencies:
18  //
19  // Revision:
20  // Revision 0.01 - File Created
21  // Additional Comments:
22  //
23  ////////////////////////////////////////////////////////////////////////////////
24
25  module receiveEngine_tf;
26
27     // Inputs
28     reg clk;
29     reg rst;
30     reg Rx;
31     reg eight;
32     reg pen;
33     reg reads0;
34     reg even;
35     reg [18:0] k;
36
37     // Outputs
38     wire RxRdy;
39     wire perr;
40     wire ferr;
41     wire ovf;
42     wire [7:0] data;
43
44     // Instantiate the Unit Under Test (UUT)
45     receiveEngine uut (
46        .clk(clk),
47        .rst(rst),
48        .Rx(Rx),
49        .eight(eight),
50        .pen(pen),
51        .reads0(reads0),
52        .even(even),
53        .k(k),
54        .RxRdy(RxRdy),
55        .perr(perr),
56        .ferr(ferr),
57        .ovf(ovf),
```

```
58          .data(data)
59      );
60
61      //Generate clock
62      always #5 clk = ~clk;
63
64      initial begin
65          // Initialize Inputs
66          clk = 0;
67          rst = 1;
68          Rx = 0;
69          eight = 0;
70          pen = 0;
71          reads0 = 0;
72          even = 0;
73          k = 109; //maximum baud rate
74
75          // Wait 100 ns for global reset to finish
76          #100;
77          rst = 0;
78
79          //DATA to be receieved 0xA5
80          wait(uut.btu == 1);
81          wait(uut.btu == 0);
82          Rx = 1;
83
84          wait(uut.btu == 1);
85          wait(uut.btu == 0);
86          Rx = 0;
87
88          wait(uut.btu == 1);
89          wait(uut.btu == 0);
90          Rx = 1;
91
92          wait(uut.btu == 1);
93          wait(uut.btu == 0);
94          Rx = 0;
95
96          wait(uut.btu == 1);
97          wait(uut.btu == 0);
98          Rx = 0;
99
100         wait(uut.btu == 1);
101         wait(uut.btu == 0);
102         Rx = 1;
103
104         wait(uut.btu == 1);
105         wait(uut.btu == 0);
106         Rx = 0;
107
108         wait(uut.btu == 1);
109         wait(uut.btu == 0);
110         Rx = 1;
111
112         //stop bit
113         wait(uut.btu == 1);
114         wait(uut.btu == 0);
```

```
115         Rx = 1;//stop
116 
117         wait(uut.done ==1);
118         #200;
119         $finish;
120 
121         // Add stimulus here
122 
123      end
124 
125   endmodule
126 
127 
```