

```

1  `timescale 1ns/1ps
2  module UART_TX_RX_MODULE_TB3#(
3      parameter UART_BAUD_RATE          = 9600,
4      parameter CLOCK_FREQUENCY         = 38400,
5      parameter PARITY                   = 2,
6      parameter NUM_OF_DATA_BITS_IN_PACK = 8,
7      parameter NUMBER_STOP_BITS         = 1
8  );
9  localparam PERIOD_IN_CLOCK_NS=1000000000/CLOCK_FREQUENCY;
10 localparam STEP=64;
11 //Входы
12 reg IN_CLOCK_1, IN_CLOCK_2;
13 reg IN_TX_LAUNCH_1, IN_TX_LAUNCH_2;
14 reg [NUM_OF_DATA_BITS_IN_PACK-1:0] IN_TX_DATA_1, IN_TX_DATA_2;
15 //Выходы
16 wire OUT_TX_ACTIVE_1, OUT_TX_ACTIVE_2;
17 wire OUT_TX_DONE_1, OUT_TX_DONE_2;
18 wire OUT_TX_STOP_BIT_ACTIVE_1, OUT_TX_STOP_BIT_ACTIVE_2;
19 wire OUT_TX_START_BIT_ACTIVE_1, OUT_TX_START_BIT_ACTIVE_2;
20 wire OUT_RX_DATA_READY_1, OUT_RX_DATA_READY_2;
21 wire [NUM_OF_DATA_BITS_IN_PACK-1:0] OUT_RX_DATA_1, OUT_RX_DATA_2;
22 wire OUT_RX_ERROR_1, OUT_RX_ERROR_2;
23
24 wire BUS_TRANSMIT_1_TO_2, BUS_TRANSMIT_2_TO_1;
25
26
27 UART_TX_RX_MODULE #(
28     .UART_BAUD_RATE(UART_BAUD_RATE),
29     .CLOCK_FREQUENCY(CLOCK_FREQUENCY),
30     .PARITY(PARITY),
31     .NUM_OF_DATA_BITS_IN_PACK(NUM_OF_DATA_BITS_IN_PACK),
32     .NUMBER_STOP_BITS(NUMBER_STOP_BITS)
33 )
34 UTRM_1
35 (
36     .IN_CLOCK(IN_CLOCK_1),
37     .IN_TX_LAUNCH(IN_TX_LAUNCH_1),
38     .IN_TX_DATA(IN_TX_DATA_1),
39
40     .OUT_TX_ACTIVE(OUT_TX_ACTIVE_1),
41     .OUT_TX_DONE(OUT_TX_DONE_1),
42     .OUT_TX_STOP_BIT_ACTIVE(OUT_TX_STOP_BIT_ACTIVE_1),
43     .OUT_TX_START_BIT_ACTIVE(OUT_TX_START_BIT_ACTIVE_1),
44     .OUT_RX_DATA_READY(OUT_RX_DATA_READY_1),
45     .OUT_RX_DATA(OUT_RX_DATA_1),
46     .OUT_RX_ERROR(OUT_RX_ERROR_1),
47
48     .IN_RX_SERIAL(BUS_TRANSMIT_2_TO_1),
49     .OUT_TX_SERIAL(BUS_TRANSMIT_1_TO_2)
50 );
51
52 UART_TX_RX_MODULE #(
53     .UART_BAUD_RATE(UART_BAUD_RATE),
54     .CLOCK_FREQUENCY(CLOCK_FREQUENCY),
55     .PARITY(PARITY),
56     .NUM_OF_DATA_BITS_IN_PACK(NUM_OF_DATA_BITS_IN_PACK),
57     .NUMBER_STOP_BITS(NUMBER_STOP_BITS)
58 )
59 UTRM_2
60 (
61     .IN_CLOCK(IN_CLOCK_2),
62     .IN_TX_LAUNCH(IN_TX_LAUNCH_2),
63     .IN_TX_DATA(IN_TX_DATA_2),
64
65     .OUT_TX_ACTIVE(OUT_TX_ACTIVE_2),
66     .OUT_TX_DONE(OUT_TX_DONE_2),
67     .OUT_TX_STOP_BIT_ACTIVE(OUT_TX_STOP_BIT_ACTIVE_2),
68     .OUT_TX_START_BIT_ACTIVE(OUT_TX_START_BIT_ACTIVE_2),
69     .OUT_RX_DATA_READY(OUT_RX_DATA_READY_2),
70     .OUT_RX_DATA(OUT_RX_DATA_2),
71     .OUT_RX_ERROR(OUT_RX_ERROR_2),
72
73     .IN_RX_SERIAL(BUS_TRANSMIT_1_TO_2),
74     .OUT_TX_SERIAL(BUS_TRANSMIT_2_TO_1)
75 );
76 always
77 begin
78     #(PERIOD_IN_CLOCK_NS/2)
79     IN_CLOCK_1=!IN_CLOCK_1;

```

```
80     IN_CLOCK_2=!IN_CLOCK_2;
81 end
82 initial begin
83     IN_CLOCK_1=1'b1;IN_CLOCK_2=1'b0;
84     IN_TX_LAUNCH_1=0;IN_TX_LAUNCH_2=0;
85     IN_TX_DATA_1=8'bz;IN_TX_DATA_2=8'bz;
86     #(PERIOD_IN_CLOCK_NS*10)
87     IN_TX_DATA_1=8'b01010101;
88     #(PERIOD_IN_CLOCK_NS*12)
89     IN_TX_LAUNCH_1=1'b1;
90     #(PERIOD_IN_CLOCK_NS*12)
91     IN_TX_LAUNCH_1=1'b0; //убираем
92     #(PERIOD_IN_CLOCK_NS*20)
93     IN_TX_DATA_1=8'bz;
94
95 end
96
97 initial begin
98     forever
99     begin
100         @(posedge OUT_RX_DATA_READY_2)
101         begin
102             if(OUT_RX_DATA_2+STEP<2**NUM_OF_DATA_BITS_IN_PACK-1)
103             begin
104                 IN_TX_DATA_2=OUT_RX_DATA_2+STEP; //заряжаем принятые данные, чтоб
отправить обратно
105                 #(PERIOD_IN_CLOCK_NS*25)
106                 IN_TX_LAUNCH_2=1'b1;
107                 #(PERIOD_IN_CLOCK_NS*25)
108                 IN_TX_LAUNCH_2=1'b0;
109                 #(PERIOD_IN_CLOCK_NS*20)
110                 IN_TX_DATA_2=8'bz;
111             end
112             else
113             begin
114                 IN_TX_DATA_2=8'bz;
115             end
116         end
117     end
118 end
119
120
121 initial begin
122     forever
123     begin
124         @(posedge OUT_RX_DATA_READY_1)
125         begin
126             if(OUT_RX_DATA_1+STEP<2**NUM_OF_DATA_BITS_IN_PACK-1)
127             begin
128                 IN_TX_DATA_1=OUT_RX_DATA_1+STEP; //заряжаем принятые данные, чтоб
отправить обратно
129                 #(PERIOD_IN_CLOCK_NS*25)
130                 IN_TX_LAUNCH_1=1'b1;
131                 #(PERIOD_IN_CLOCK_NS*25)
132                 IN_TX_LAUNCH_1=1'b0;
133                 #(PERIOD_IN_CLOCK_NS*20)
134                 IN_TX_DATA_1=8'bz;
135             end
136             else
137             begin
138                 IN_TX_DATA_1=8'bz;
139             end
140         end
141     end
142 end
143
144 endmodule
```