

```

1  module UART_TX_RX_MODULE
2  #(
3      parameter UART_BAUD_RATE           = 9600, //битрейт передачи
4      parameter CLOCK_FREQUENCY          = 50000000, //частота тактового сигнала IN_CLOCK
5      parameter PARITY                   = 1, //Параметр бита четности/нечетности
6      parameter NUM_OF_DATA_BITS_IN_PACK = 8, //число информационных бит в пакете
7      parameter NUMBER_STOP_BITS         = 2 //число стоп-битов а пакете
8  )
9  (
10     input          IN_CLOCK,                //тактовый сигнал
11     input          IN_TX_LAUNCH,            //сигнальная линия для
инициализации передачи
12     input [NUM_OF_DATA_BITS_IN_PACK-1:0] IN_TX_DATA, //вектор данных для передачи при
следующей транзакции
13     //данный вектор защелкивается во внутреннем регистре модуля при инициализации передачи.
14
15     output OUT_TX_ACTIVE,                    //сигнальная линия активности
передаточного узла TX
16     output OUT_TX_DONE,                      //сигнальная линия окончания
передачи пакета
17     output OUT_TX_STOP_BIT_ACTIVE,           //сигнальная линия передачи стоп
бита модулем TX
18     output OUT_TX_START_BIT_ACTIVE,          //сигнальная линия передачи
старт бита модулем TX
19     output OUT_RX_DATA_READY,                //сигнальная линия готовности
данных OUT_RX_DATA приемника
20     output [NUM_OF_DATA_BITS_IN_PACK-1:0] OUT_RX_DATA, //вектор данных, принятый
приемником RX
21     output OUT_RX_ERROR,                     //сигнальная линия ошибки приема
последнего пакета
22
23
24     input          IN_RX_SERIAL,             //RX
25     output         OUT_TX_SERIAL            //TX
26 );
27
28 localparam NUM_OF_DATA_BITS_IN_PACK_LOG_2=$clog2(NUM_OF_DATA_BITS_IN_PACK) ;
29 localparam CLKS_PER_BIT_LOG_2=$clog2(NUMBER_STOP_BITS*CLOCK_FREQUENCY/UART_BAUD_RATE);
30
31 UART_FPGA_TX #(
32     .UART_BAUD_RATE(UART_BAUD_RATE),
33     .CLOCK_FREQUENCY(CLOCK_FREQUENCY),
34     .PARITY(PARITY),
35     .CLKS_PER_BIT_LOG_2(CCLKS_PER_BIT_LOG_2),
36     .NUM_OF_DATA_BITS_IN_PACK(NUM_OF_DATA_BITS_IN_PACK),
37     .NUM_OF_DATA_BITS_IN_PACK_LOG_2(NUM_OF_DATA_BITS_IN_PACK_LOG_2),
38     .NUMBER_STOP_BITS(NUMBER_STOP_BITS)
39 )
40 TX
41 (
42     .IN_CLOCK(IN_CLOCK),
43     .IN_TX_LAUNCH(IN_TX_LAUNCH),
44     .IN_TX_DATA(IN_TX_DATA),
45     .OUT_TX_ACTIVE(OUT_TX_ACTIVE),
46     .OUT_TX_SERIAL(OUT_TX_SERIAL),
47     .OUT_TX_DONE(OUT_TX_DONE),
48     .OUT_TX_STOP_BIT_ACTIVE(OUT_TX_STOP_BIT_ACTIVE),
49     .OUT_TX_START_BIT_ACTIVE(OUT_TX_START_BIT_ACTIVE)
50 ); //подключение модуля TX
51
52 UART_FPGA_RX #(
53     .UART_BAUD_RATE(UART_BAUD_RATE),
54     .CLOCK_FREQUENCY(CLOCK_FREQUENCY),
55     .PARITY(PARITY),
56     .CLKS_PER_BIT_LOG_2(CCLKS_PER_BIT_LOG_2),
57     .NUM_OF_DATA_BITS_IN_PACK(NUM_OF_DATA_BITS_IN_PACK),
58     .NUM_OF_DATA_BITS_IN_PACK_LOG_2(NUM_OF_DATA_BITS_IN_PACK_LOG_2)
59 )
60 RX
61 (
62     .IN_CLOCK(IN_CLOCK),
63     .IN_RX_SERIAL(IN_RX_SERIAL),
64     .OUT_RX_DATA_READY(OUT_RX_DATA_READY),
65     .OUT_RX_DATA(OUT_RX_DATA),
66     .OUT_RX_ERROR(OUT_RX_ERROR)
67 ); //подключение модуля RX
68
69 endmodule
70

```