Verilog code

module UART\_TX\_RX(

input clk,

input start,

input [7:0] txin,

output reg tx,

input rx,

output [7:0] rxout,

output rxdone, txdone

);

parameter clk\_value = 100\_000;

parameter baud = 9600;

parameter wait\_count = clk\_value / baud;

reg bitDone = 0;

integer count = 0;

parameter idle = 0, send = 1, check = 2;

reg [1:0] state = idle;

///////////////////Generate Trigger for Baud Rate

always@(posedge clk)

begin

if(state == idle)

begin

count <= 0;

end

else begin

if(count == wait\_count)

begin

bitDone <= 1'b1;

count <= 0;

end

else

begin

count <= count + 1;

bitDone <= 1'b0;

end

end

end

///////////////////////TX Logic

reg [9:0] txData;///stop bit data start

integer bitIndex = 0; ///reg [3:0];

reg [9:0] shifttx = 0;

always@(posedge clk)

begin

case(state)

idle :

begin

tx <= 1'b1;

txData <= 0;

bitIndex <= 0;

shifttx <= 0;

if(start == 1'b1)

begin

txData <= {1'b1,txin,1'b0};

state <= send;

end

else

begin

state <= idle;

end

end

send: begin

tx <= txData[bitIndex];

state <= check;

shifttx <= {txData[bitIndex], shifttx[9:1]};

end

check:

begin

if(bitIndex <= 9) ///0 - 9 = 10

begin

if(bitDone == 1'b1)

begin

state <= send;

bitIndex <= bitIndex + 1;

end

end

else

begin

state <= idle;

bitIndex <= 0;

end

end

default: state <= idle;

endcase

end

assign txdone = (bitIndex == 9 && bitDone == 1'b1) ? 1'b1 : 1'b0;

////////////////////////////////RX Logic

integer rcount = 0;

integer rindex = 0;

parameter ridle = 0, rwait = 1, recv = 2, rcheck = 3;

reg [1:0] rstate;

reg [9:0] rxdata;

always@(posedge clk)

begin

case(rstate)

ridle :

begin

rxdata <= 0;

rindex <= 0;

rcount <= 0;

if(rx == 1'b0)

begin

rstate <= rwait;

end

else

begin

rstate <= ridle;

end

end

rwait :

begin

if(rcount < wait\_count / 2)

begin

rcount <= rcount + 1;

rstate <= rwait;

end

else

begin

rcount <= 0;

rstate <= recv;

rxdata <= {rx,rxdata[9:1]};

end

end

recv :

begin

if(rindex <= 9)

begin

if(bitDone == 1'b1)

begin

rindex <= rindex + 1;

rstate <= rwait;

end

end

else

begin

rstate <= ridle;

rindex <= 0;

end

end

default : rstate <= ridle;

endcase

end

assign rxout = rxdata[8:1];

assign rxdone = (rindex == 9 && bitDone == 1'b1) ? 1'b1 : 1'b0;