In the name of God

Computer Architecture Project Report

Students:

Salar Ghoochi: 98110446

Amirali Mostaghis: 98102324

Ahmadreza Tavana: 98104852

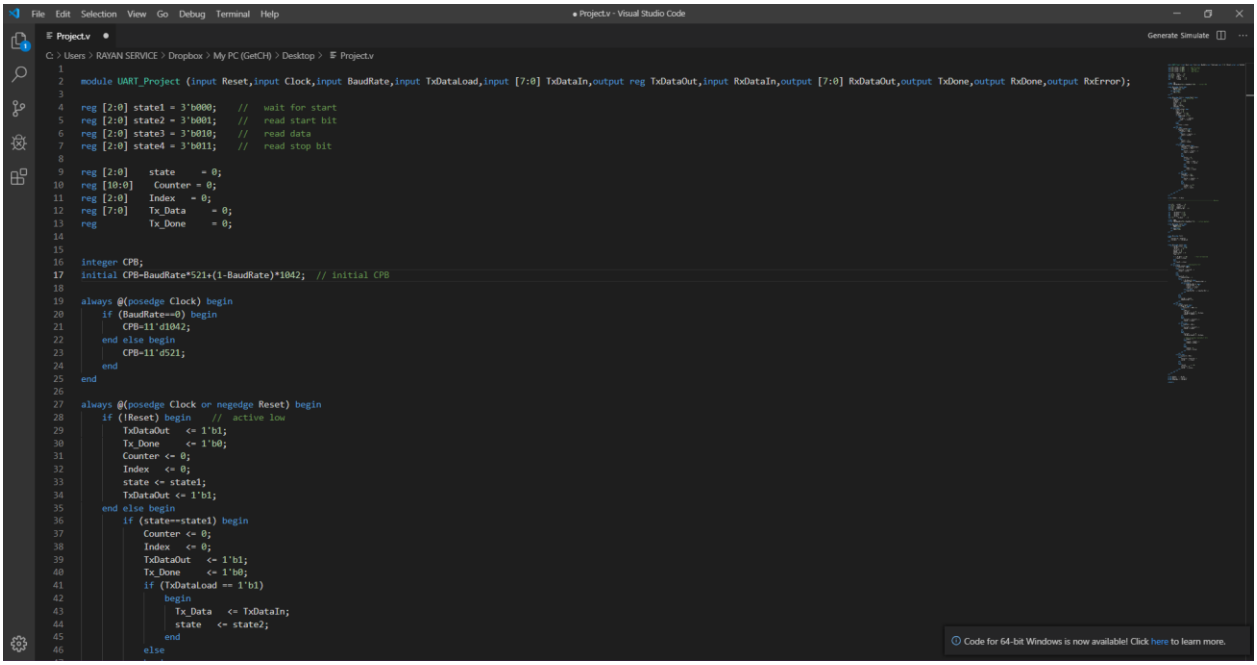
Professor: Dr.Hajsadeghi

Abstract

In this project we design a universal asynchronous receiver-transmitter (UART) with Verilog. UART is a hardware communication protocol that uses asynchronous serial communication with configurable speed. UART has two parts, sender part and receiver part. In this report we explain our code and modules generally and put block diagram and state diagram and also the results of our simulation.

Transmitter Part

In this part at first we define states and then in state1 we wait for start and also define TxDate for defining the series of data and then TxData for activation. This part has a clock that we named it TxClock that works with frequency of Baud Rate. The baud rate is the rate at which information is transferred in a communication channel. Baud rate is commonly used when discussing electronics that use serial communication. In the serial port context, "9600 baud" means that the serial port is capable of transferring a maximum of 9600 bits per second. Then we compare states with the general state and then base on the result we initialize TxDataOut and TxDone and repeat this process in posedge clock. That was a general explanation of our code and we put the main code blow to see the details of code:



```
module UART_Project (input Reset,input Clock,input BaudRate,input TxDataLoad,input [7:0] TxDataIn,output reg TxDataOut,input RxDataIn,output [7:0] RxDataOut,output TxDone,output RxDone,output RxError);

reg [2:0] state1 = 3'b000;    //   wait for start
reg [2:0] state2 = 3'b001;    //   read start bit
reg [2:0] state3 = 3'b010;    //   read data
reg [2:0] state4 = 3'b011;    //   read stop bit

reg [2:0]    state     = 0;
reg [10:0]   Counter = 0;
reg [2:0]    Index   = 0;
reg [7:0]    Tx_Data     = 0;
reg          Tx_Done     = 0;


integer CPB;
initial CPB=BaudRate*521+(1-BaudRate)*1042;  // initial CPB

always @(posedge Clock) begin
    if (BaudRate==0) begin
        CPB=11'd1042;
    end else begin
        CPB=11'd521;
    end
end

always @(posedge Clock or negedge Reset) begin
    if (!Reset) begin    //  active low
        TxDataOut   <= 1'b1;
        Tx_Done     <= 1'b0;
        Counter <= 0;
        Index   <= 0;
        state <= state1;
        TxDataOut <= 1'b1;
    end else begin
        if (state==state1) begin
            Counter <= 0;
            Index   <= 0;
            TxDataOut   <= 1'b1;
            Tx_Done     <= 1'b0;
            if (TxDataLoad == 1'b1)
                begin
                  Tx_Data   <= TxDataIn;
                  state   <= state2;
                end
            else
```

File Edit Selection View Go Debug Terminal Help ● Project.v - Visual Studio Code

Project.v ● Generate Simulate

C: > Users > RAYAN SERVICE > Dropbox > My PC (GetCH) > Desktop > Project.v

```
            end
        else
        begin
            state <= state1;
        end
    end else begin
        if (state==state2) begin
            TxDataOut <= 1'b0;
            if (Counter < CPB-1)
              begin
                Counter <= Counter + 1;
                state <= state2;
              end
            else
              begin
                Counter <= 0;
                state <= state3;
              end
        end else begin
            if (state==state3) begin
                TxDataOut <= Tx_Data[Index];
                if (Counter < CPB-1)
                begin
                    Counter <= Counter + 1;
                    state <= state3;
                end
                else
                begin
                    Counter <= 0;
                    if (Index < 7)
                    begin
                        Index <= Index + 1;
                        state   <= state3;
                    end
                    else
                    begin
                        Index <= 0;
                        state <= state4;
                    end
                end
            end else begin
                TxDataOut <= 1'b1;
                if (Counter < CPB-1)
                begin
                    Counter <= Counter + 1;
                    state  <= state4;
```

Code for 64-bit Windows is now available! Click here to learn more.

File Edit Selection View Go Debug Terminal Help ● Project.v - Visual Studio Code

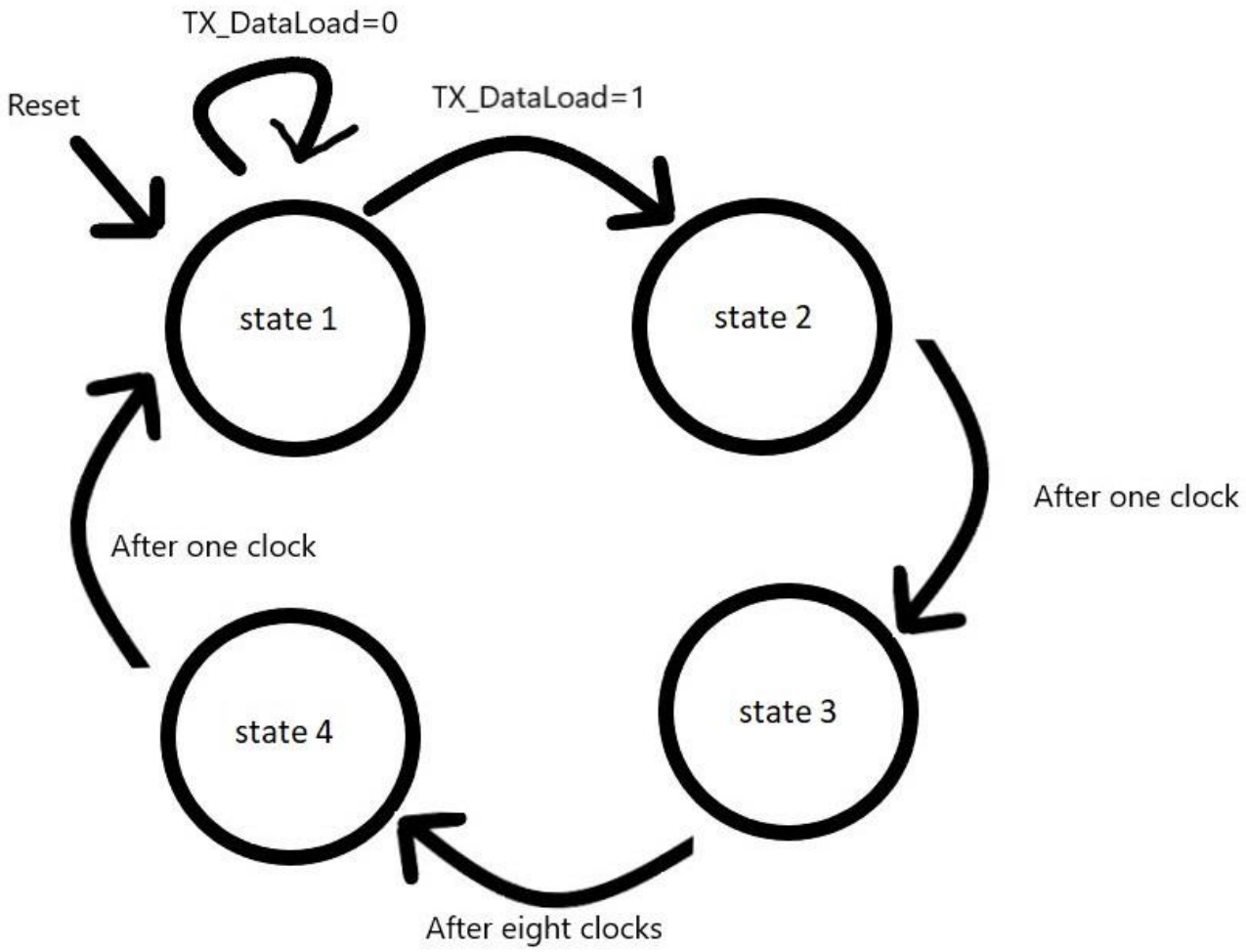
Project.v ● Generate Simulate

C: > Users > RAYAN SERVICE > Dropbox > My PC (GetCH) > Desktop > Project.v

```
                Counter <= 0;
                state <= state3;
              end
        end else begin
            if (state==state3) begin
                TxDataOut <= Tx_Data[Index];
                if (Counter < CPB-1)
                begin
                    Counter <= Counter + 1;
                    state <= state3;
                end
                else
                begin
                    Counter <= 0;
                    if (Index < 7)
                    begin
                        Index <= Index + 1;
                        state   <= state3;
                    end
                    else
                    begin
                        Index <= 0;
                        state <= state4;
                    end
                end
            end else begin
                TxDataOut <= 1'b1;
                if (Counter < CPB-1)
                begin
                    Counter <= Counter + 1;
                    state  <= state4;
                end
                else
                begin
                    Tx_Done  <= 1'b1;
                    Counter <= 0;
                    state <= state1;
                end
            end
        end
    end
  end
end

assign TxDone  = Tx_Done;
```

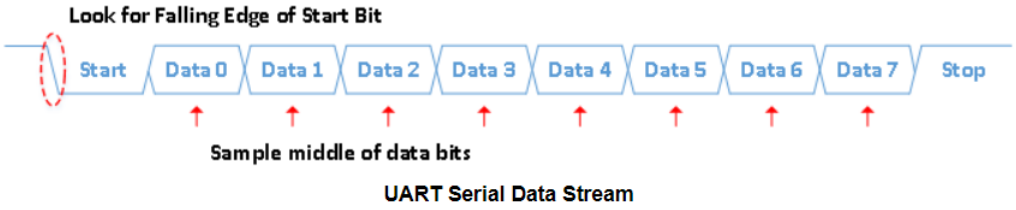
Code for 64-bit Windows is now available! Click here to learn more.

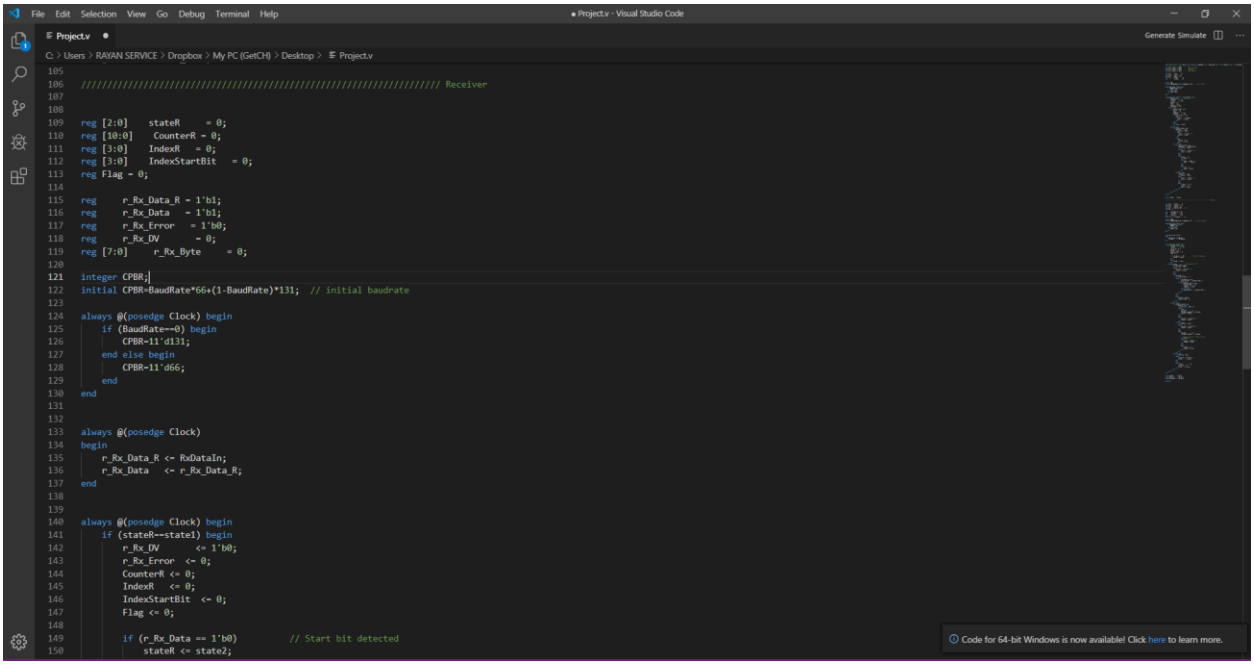
Transmitter State Diagram



Receiver Part

In this part at first we define stateR and RxDataIn for receiving data and RxDataOut for sending and RxError for the error part and contain the noise that defined in project document. And also we define the clock like the sender part that is equal to Baud Rate. In this part, as the previous part, we compare the states and then we check our counter and then shift it and also check the Rxdata and start bit and shift start bit if RxData is equal to 0 and then we check if start bit is less than 4 we make it 0 and then we put RxError equal to 1 and then change the state and do this process for other states and repeat this process in posedge of clock and finally we clarify RxDone and RxError and RxDataOut at the end of code. With completing the receiver part, our UART is ready and we can use it to sending data in a frame that we put in blow. Also we put the main code of receiver part to see the details of the code:





```
////////////////////////////////////////////////////////////////////////////// Receiver


reg [2:0]    stateR     = 0;
reg [10:0]    CounterR = 0;
reg [3:0]    IndexR    = 0;
reg [3:0]    IndexStartBit   = 0;
reg Flag = 0;

reg     r_Rx_Data_R = 1'b1;
reg     r_Rx_Data   = 1'b1;
reg     r_Rx_Error   = 1'b0;
reg     r_Rx_DV       = 0;
reg [7:0]     r_Rx_Byte     = 0;

integer CPBR;
initial CPBR=BaudRate*66+(1-BaudRate)*131;  // initial baudrate

always @(posedge Clock) begin
    if (BaudRate==0) begin
        CPBR=11'd131;
    end else begin
        CPBR=11'd66;
    end
end


always @(posedge Clock)
begin
    r_Rx_Data_R <= RxDataIn;
    r_Rx_Data   <= r_Rx_Data_R;
end


always @(posedge Clock) begin
    if (stateR==state1) begin
        r_Rx_DV       <= 1'b0;
        r_Rx_Error  <= 0;
        CounterR <= 0;
        IndexR   <= 0;
        IndexStartBit  <= 0;
        Flag <= 0;

        if (r_Rx_Data == 1'b0)          // Start bit detected
            stateR <= state2;
```

File Edit Selection View Go Debug Terminal Help • Project.v - Visual Studio Code

Project.v •

Generate Simulate

C: > Users > RAYAN SERVICE > Dropbox > My PC (GetCH) > Desktop > Project.v

```
            stateR <= state2;
        else
        begin
            stateR <= state1;
        end
    end else begin          //Checking Start bit
        if (stateR==state2) begin
            if (CounterR < CPBR-1)
            begin
                CounterR <= CounterR + 1;
                stateR <= state2;
            end
            else
            begin
                CounterR <= 0;
                if (IndexStartBit < 7)
                begin
                    if (r_Rx_Data==0) begin
                        IndexStartBit  <= IndexStartBit +1;
                    end else begin
                        if (IndexStartBit<4) begin
                            IndexStartBit  <=  0;
                            stateR   <= state1;
                            r_Rx_Error  <= 1;
                        end else begin
                            IndexStartBit  <= IndexStartBit +1;
                        end
                    end
                end
                else
                begin
                    stateR <= state3;
                    IndexStartBit <= 0;
                end
            end
        end else begin
            if (stateR==state3) begin
                if (Flag==0) begin
                  if (CounterR == (CPB-1)/2)
                  begin
                      Flag <= 1;
                      CounterR       <= 0;
                      r_Rx_Byte[IndexR] <= r_Rx_Data;
                      IndexR <= IndexR + 1;
                  end
                  else
```

Code for 64-bit Windows is now available! Click here to learn more.

File Edit Selection View Go Debug Terminal Help • Project.v - Visual Studio Code

Project.v •

Generate Simulate

C: > Users > RAYAN SERVICE > Dropbox > My PC (GetCH) > Desktop > Project.v

```
                  else
                  begin
                      CounterR <= CounterR + 1;
                      stateR <= state3;
                  end
                end else begin
                    if (CounterR < CPB-1)
                    begin
                        CounterR <= CounterR + 1;
                        stateR <= state3;
                    end
                    else
                    begin
                        CounterR       <= 0;
                        r_Rx_Byte[IndexR] <= r_Rx_Data;

                        // Check if we have received all bits
                        if (IndexR < 7)
                        begin
                            IndexR <= IndexR + 1;
                            stateR <= state3;
                        end
                        else
                        begin
                            IndexR <= 0;
                            stateR <= state4;
                        end
                    end
                end
            end else begin
                if (CounterR < CPB-1)
                begin
                    CounterR <= CounterR + 1;
                    stateR     <= state4;
                end
                else
                begin
                    r_Rx_DV       <= 1'b1;
                    CounterR <= 0;
                    stateR   <= state1 ;
                end
            end
        end
    end
end
```

Code for 64-bit Windows is now available! Click here to learn more.

File Edit Selection View Go Debug Terminal Help • Project.v - Visual Studio Code

Project.v •

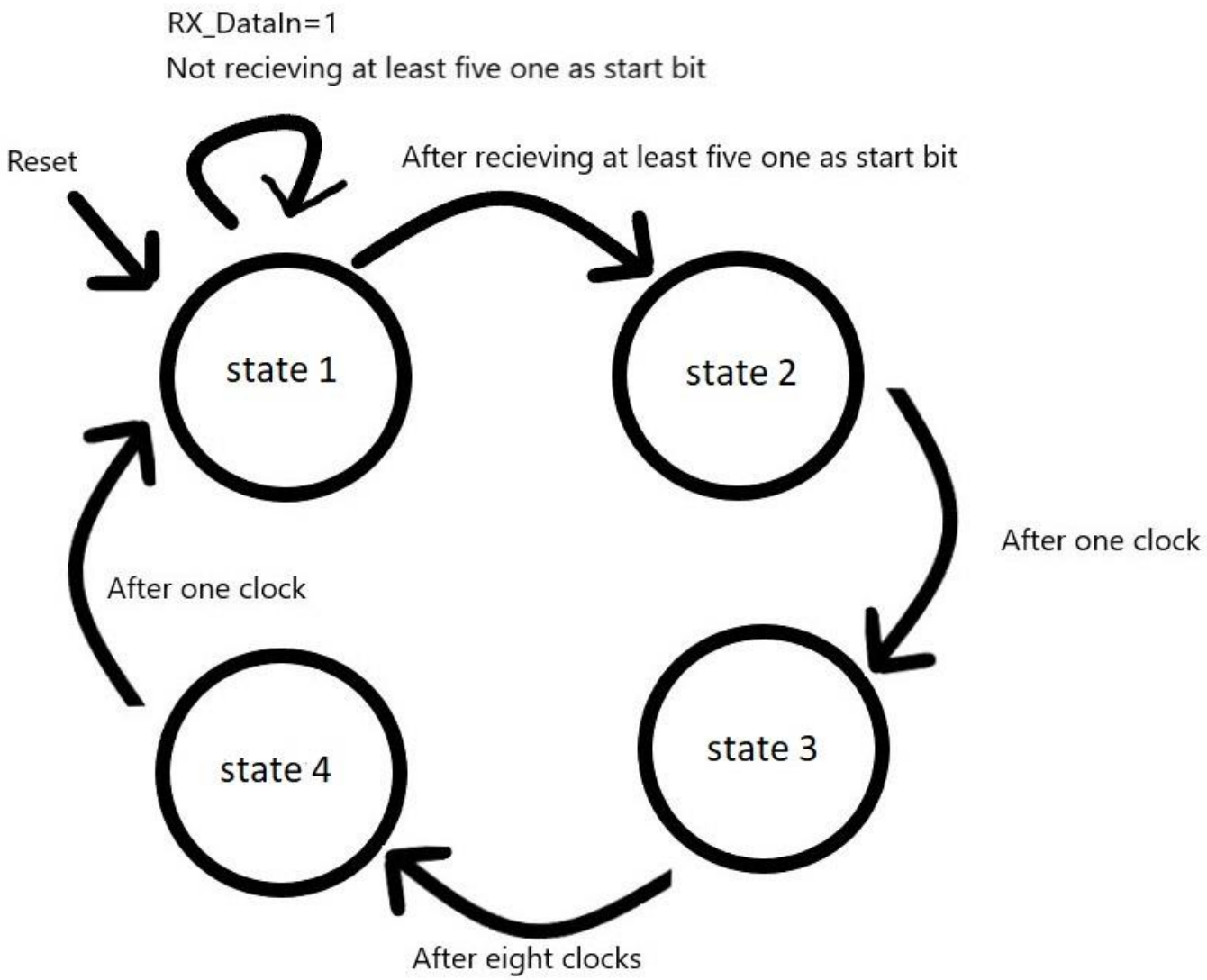
Generate Simulate

C: > Users > RAYAN SERVICE > Dropbox > My PC (GetCH) > Desktop > Project.v

```
                end else begin
                    if (CounterR < CPB-1)
                    begin
                        CounterR <= CounterR + 1;
                        stateR <= state3;
                    end
                    else
                    begin
                        CounterR       <= 0;
                        r_Rx_Byte[IndexR] <= r_Rx_Data;

                        // Check if we have received all bits
                        if (IndexR < 7)
                        begin
                            IndexR <= IndexR + 1;
                            stateR <= state3;
                        end
                        else
                        begin
                            IndexR <= 0;
                            stateR <= state4;
                        end
                    end
                end
            end else begin
                if (CounterR < CPB-1)
                begin
                    CounterR <= CounterR + 1;
                    stateR     <= state4;
                end
                else
                begin
                    r_Rx_DV       <= 1'b1;
                    CounterR <= 0;
                    stateR   <= state1 ;
                end
            end
        end
    end
end

assign RxDone   = r_Rx_DV;
assign RxError   = r_Rx_Error;
assign RxDataOut = r_Rx_Byte;

endmodule
```

Code for 64-bit Windows is now available! Click here to learn more.

Receiver State Diagram



Test Bench

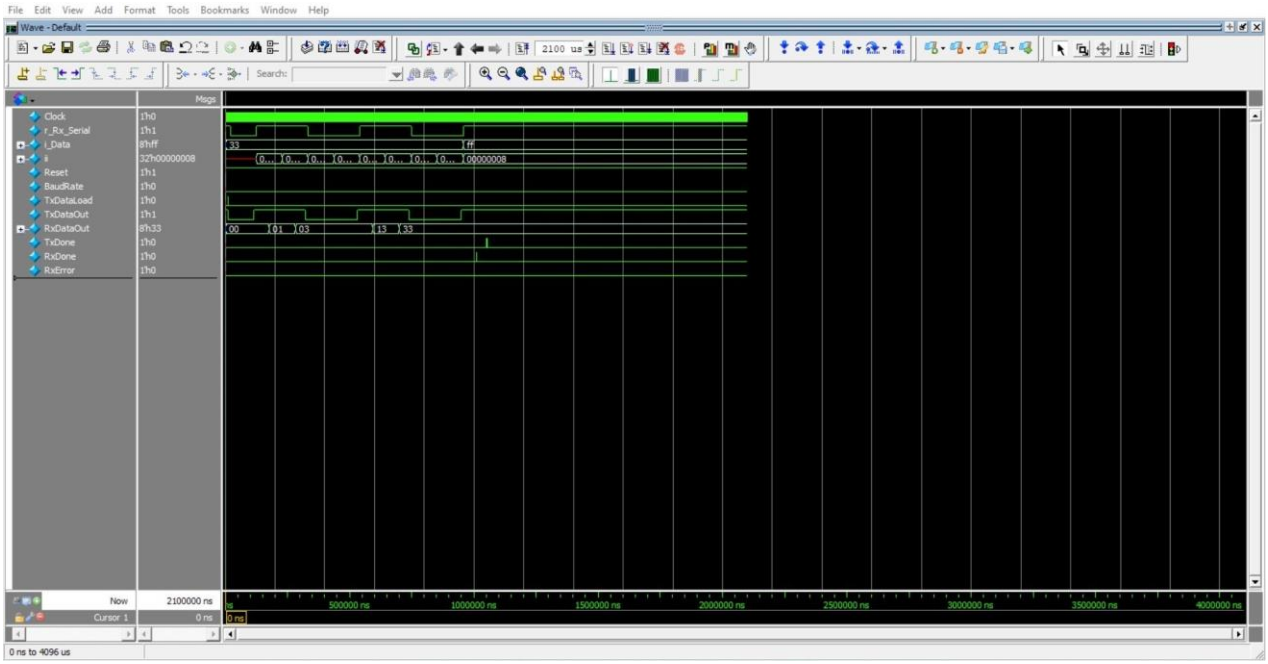
In our test bench at first we make the clock and after that we initialize i_Data that we set it 00110011and then we go at a loop to make i_Data equal to 11111111 and then send it to our UART to make serialize it and in this process we check start bit for sending and sending data byte and finally check the stop bit for ending module. We put the test bench of our code blow and then we put the result of our simulation on this test bench:

```
module uart_tb ();


parameter c_CLOCK_PERIOD_NS = 100;
parameter c_BIT_PERIOD      = 104167;

reg Clock = 0;
reg r_Rx_Serial = 1;
reg [7:0] i_Data = 8'b00110011;
integer     ii;

reg Reset=1;
reg BaudRate=0;
reg TxDataLoad=0;
wire TxDataOut;
wire [7:0] RxDataOut;
wire TxDone;
wire RxDone;
wire RxError;



 UART_Project  UART_INST
   (.Reset(Reset),
    .Clock(Clock),
    .BaudRate(BaudRate),
    .TxDataLoad(TxDataLoad),
    .TxDataIn(i_Data),
    .TxDataOut(TxDataOut),
    .RxDataIn(r_Rx_Serial),
    .RxDataOut(RxDataOut),
    .TxDone(TxDone),
    .RxDone(RxDone),
    .RxError(RxError)
    );

  always  begin
      #(c_CLOCK_PERIOD_NS/2) Clock <= !Clock;
  end

  always
  begin
      if (i_Data != 8'b11111111) begin
          #10000;
          TxDataLoad=1;
          #100;
```

```
    .RxDataIn(r_Rx_Serial),
    .RxDataOut(RxDataOut),
    .TxDone(TxDone),
    .RxDone(RxDone),
    .RxError(RxError)
    );

  always  begin
      #(c_CLOCK_PERIOD_NS/2) Clock <= !Clock;
  end

  always
  begin
      if (i_Data != 8'b11111111) begin
          #10000;
          TxDataLoad=1;
          #100;
          TxDataLoad=0;
          #10000;
           // Send Start Bit
           r_Rx_Serial <= 1'b0;
           #(c_BIT_PERIOD);


          // Send Data Byte
          for (ii=0; ii<8; ii=ii+1)
              begin
              r_Rx_Serial <= i_Data[ii];
              #(c_BIT_PERIOD);
              end

          // Send Stop Bit
          r_Rx_Serial <= 1'b1;
          i_Data = 8'b11111111;
          #(c_BIT_PERIOD);
          end
          else
          begin
              #2000;
          end
  end



endmodule
```

Simulation Result

In our test we try to send number 33 to UART in some clocks and as we can see in blow we define the clock in test bench and then as we said we are trying to send 33 number and as we can see the number sent and received successfully and our module work as we want.



References

1: https://en.wikipedia.org/wiki/Universal_asynchronous_receiver-transmitter

2: https://www.setra.com/blog/what-is-baud-rate-and-what-cable-length-is-required-1

3: https://learn.sparkfun.com/tutorials/serial-communication/rules-of-serial

4: https://www.circuitbasics.com/basics-uart-communication/

5: https://www.analog.com/en/analog-dialogue/articles/uart-a-hardware-communication-protocol.html#

6: https://www.nandland.com/vhdl/modules/module-uart-serial-port-rs232.html

7: https://en.wikipedia.org/wiki/Baud