PROJECT BASED LAB REPORT

On

UART RECIEVER

Submitted in partial fulfillment of

the Requirements for the award of

the Degree of Bachelor of

Technology

in

ELECTRONICS AND COMMUNICATION ENGINEERING

By

N. ESWAR	**180040256**
K.Y SAGAR	**180040257**
P. SRI VIDYA	**180040263**

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

K L University

Green Fields, Vaddeswaram , Guntur District-522 502

2020-2021

K L University

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

CERTIFICATE

This certificate is to certify the project Report entitled

"UART RECIEVER"

which is being submitted by" out **N. ESWAR (180040256), K. Y SAGAR (180040257), P. SRI VIDYA (180040263)** submitted in fulfillment of **B. Tech** degree from **Department of ECE** to the **K L University** is a record of work carried out under your guidance and supervision. The results included in this report are not copied from any other departments or University or Institute.

Signature of the H.O. D

Dr. Suman Maloji

Head of the Department

Professor Dept. of ECE

Signature of Project Guide

T. Sanath Kumar sir

Assistant prof.

Dept. of Ece

External Examiner

K L University

DEPARTMENT OF ELECTRONICS AND COMMUNICATIONENGINEERING

DECLARATION

We hereby declare that this project-based lab report titled **" UART RECIEVER"** has been prepared
by us in partial fulfilment of the requirements for the award of degree "**BACHELOR OF TECHNOLOGY in ELECTRONICS AND COMMUNICATION ENGNEERING**" during
the Academic year 2020-2021.

We also declare that this project-based lab report is of our own efforts and it has not been submitted to any other university for the award of any degree.

N. ESWAR	180040256
K.Y SAGAR	180040257
P. SRI VIDYA	180040263

ACKNOWLEDGEMENTS

It is great pleasure for me to express my gratitude to our honorable President **Sri. Koneru Satyanarayana**, for giving the opportunity and platform with facilities in accomplishing the project-based laboratory report.

I express the sincere gratitude to our principal **Prof Dr. N. Venkataram** for his administration towards our academic growth.

I express sincere gratitude to HOD **Mr. M Suman** his leadership and constant motivation provided in successful completion of our academic semester. I record it as my privilege to deeply thank for providing us the efficient faculty and facilities to make our ideas into reality.

I express my sincere thanks to our project supervisor **sir** for his novel association of ideas, encouragement, appreciation and intellectual zeal which motivated us to venture this project successfully.

Finally, it is pleased to acknowledge the indebtedness to all those who devoted themselves directly or indirectly to make this project report success.

INDEX

S.NO	TITLE	PAGE NO
1	INTRODUCTION	5
2	METHODOLOGY	7
3	SOFTWARE USED	8
4	IMPLEMENTATION	9
5	TESTBENCH PROGRAM CODE	12
6	RESULT	14
7	APPLICATION	13
8	Conclusion	14

INTRODUCTION

VERILOG

Verilog is a Hardware Description Language; a textual format for describing electronic circuits and systems. Applied to electronic design, Verilog is intended to be used for verification through simulation, for timing analysis, for test analysis (testability analysis and fault grading) and for logic synthesis.

BEHAVIORAL VERILOG MODELING

Higher levels of abstraction within an HDL improve the productivity of a hardware developer by allowing him or her to simply describe the intended behavior of the digital circuit rather than the flow of digital data through logic gates. Compared to gate-level descriptions at the structural or dataflow level, behavioral modeling utilizes many of the constructs available in higher level programming languages to describe the algorithm the circuit designer would like to implement in hardware. One thing to keep in mind, however, is that most HDLs (including Verilog) are still concurrent languages, which means that behavioral statements you describe run in parallel. After all, it is hardware that you are describing! Behavioral modeling in Verilog makes use of two very different structured procedure statements, initial and always. Both of these statements use the begin and end keywords to group behavioral statements into a single block. For example, code in between a begin and end pair immediately following an initial statement constitutes an initial block. Similarly, an always block includes the grouped code immediately following an always statement. An initial block executes only once starting at the beginning of simulation, whereas an always block repeats whenever a given trigger condition is met. The code below describes a 1-bit, 2:1 MUX using behavioral Verilog. Please take a moment to examine the code.

always@ IN VERILOLG

always@ blocks are used to describe events that should happen under certain conditions. always@ blocks are always followed by a set of parentheses, a begin, some code, and an end.

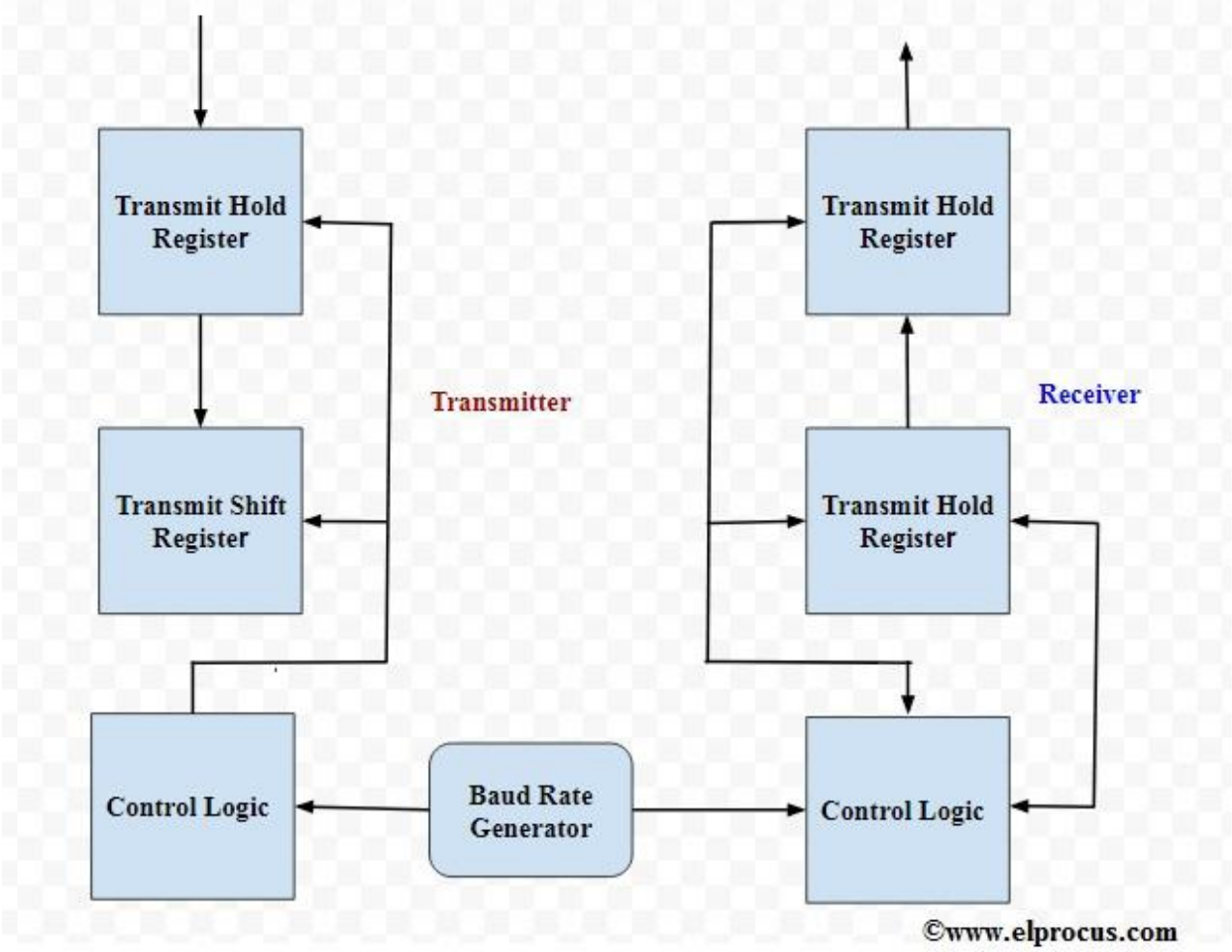
PROGRAM CODE

```
always @ ( ... sensitivity list ... )
        begin
       ...elements ...
        end
```

*UART, which stands for Universal Asynchronous Receiver/Transmitter is a circuit for sending parallel data through a serial line.

 A universal asynchronous receiver-transmitter is a computer hardware device for asynchronous serial communication in which the data format and transmission speeds are configurable

BLOCK DIAGRAM:



METHODOLOGY:

*Asynchronous transmission allows data to be transmitted without the sender having to send a clock signal to the receiver.

*In asynchronous transmission, the sender sends a Start bit, 5 to 8 data bits (LSB first), an optional Parity bit, and then 1, 1.5 or 2 Stop bits

SOFTWARE REQUIRED:

Xilinx ISE is a design Suite 14.7

IMPLEMENTATION

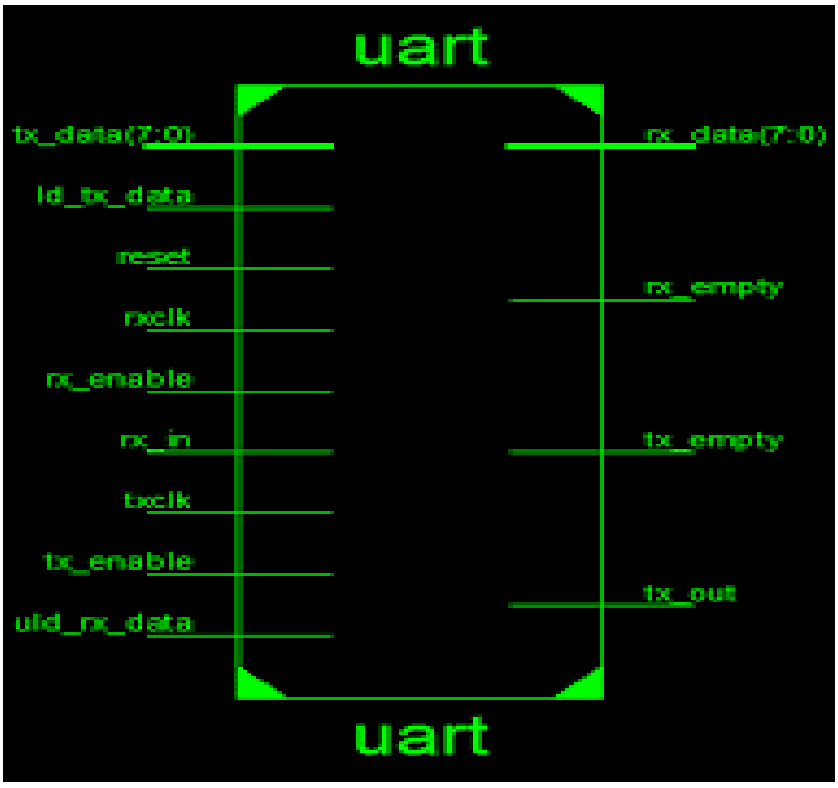
(UART RECEIVER) PROGRAM CODE:

```
module UART_rx (Clk,Rst_n,RxEn,RxData,RxDone,Rx,Tick,NBits);
input Clk, Rst_n, RxEn,Rx,Tick;
input [3:0]NBits;
output RxDone;
output [7:0]RxData;
parameter IDLE = 1'b0, READ = 1'b1;
reg [1:0] State, Next;
reg read_enable = 1'b0;
reg start_bit = 1'b1;
reg RxDone = 1'b0;
reg [4:0]Bit = 5'b00000;
reg [3:0] counter = 4'b0000;
reg [7:0] Read_data= 8'b00000000;
reg [7:0] RxData;


always @ (posedge Clk or negedge Rst_n)
begin
if (!Rst_n) State <= IDLE;
else State <= Next;
end


always @ (State or Rx or RxEn or RxDone)
begin
   case(State)
        IDLE:  if(!Rx & RxEn)          Next = READ;
               else                    Next = IDLE;
        READ: if(RxDone)               Next = IDLE;
               else                    Next = READ;
        default                        Next = IDLE;
   endcase
end


always @ (State or RxDone)
begin
   case (State)
        READ: begin
               read_enable <= 1'b1;
            end

        IDLE: begin
               read_enable <= 1'b0;
```

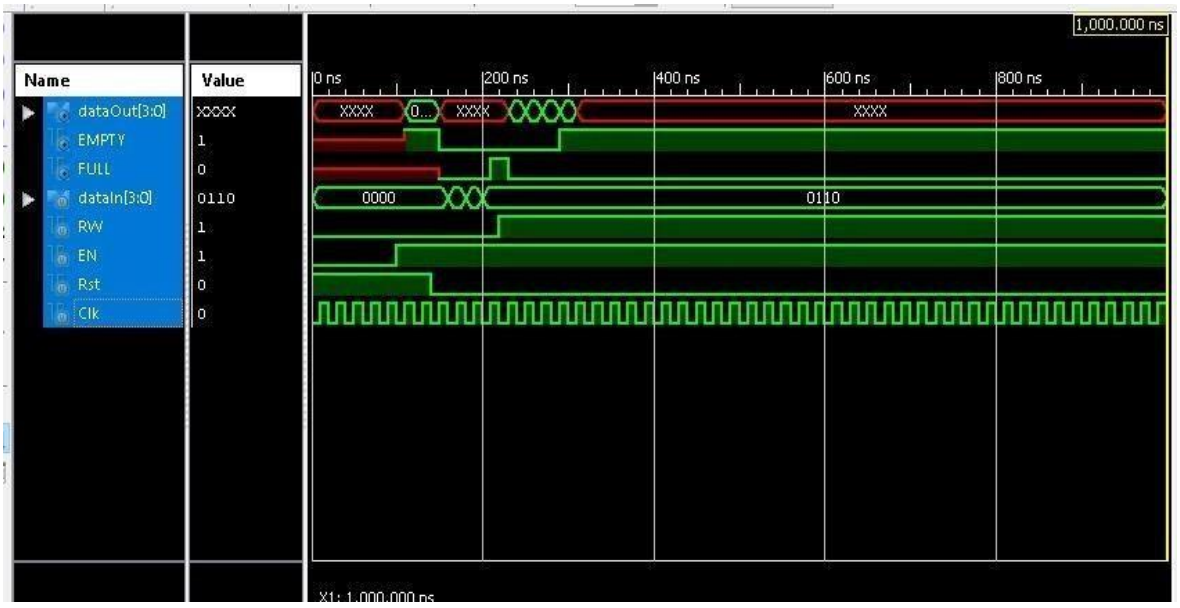
```
            end
   endcase
end


always @ (posedge Tick)

        begin
        if (read_enable)
        begin
        RxDone <= 1'b0;
        counter <= counter+1;


        if ((counter == 4'b1000) & (start_bit))
        begin
        start_bit <= 1'b0;
        counter <= 4'b0000;
        end

        if ((counter == 4'b1111) & (!start_bit) & (Bit < NBits))
        begin
        Bit <= Bit+1;
        Read_data <= {Rx,Read_data[7:1]};
        counter <= 4'b0000;
        end
        if ((counter == 4'b1111) & (Bit == NBits) & (Rx))
        begin
        Bit <= 4'b0000;
        RxDone <= 1'b1;
        counter <= 4'b0000;
        start_bit <= 1'b1;
        end
        end
end


always @ (posedge Clk)
begin
if (NBits == 4'b1000)
begin
RxData[7:0] <= Read_data[7:0];
end
if (NBits == 4'b0111)
begin
RxData[7:0] <= {1'b0,Read_data[7:1]};
end
if (NBits == 4'b0110)
begin
RxData[7:0] <= {1'b0,1'b0,Read_data[7:2]};
end
end
endmodule
```

RESULT

BLOCK DIAGRAM



SIMULATION OUTPUT:



APPLICATIONS

UART is one of the most simple and most commonly used **Serial** Communication techniques.

*Today, **UART** is being used in many **applications** like GPS **Receivers**, Bluetooth Modules, GSM and GPRS Modems, Wireless Communication Systems, RFID based **applications** etc

CONCLUSION

***This bit is usually used by receiver to perform simple error checking.**

***Lastly, Stop bit will be sent to indicate the end of transmission.**