UNIVERSAL ASYNCHRONOUS RECEIVER AND TRANSMITTER (UART) USING VERILOG

Submitted in partial fulfilment of
the requirements for the Award of the degree of
Bachelor of Technology
In

ELECTRONICS & COMMUNICATION ENGINEERING
by

B. SOWBHAGYA (R141013)
D. RAJANI (R141023)
E. PAVITHRA (R141256)

Under the guidance of

B. BHASKAR Sir

Department of Electronics & Communication Engineering
Rajiv Gandhi University of Knowledge Technologies
RK Valley-516 330, Idupulapaya, Kadapa (dist),
Andhra Pradesh

03

A.P.IIIT-RGUKT-R.K.VALLEY,KADAPA
April- 2019

Rajiv Gandhi University of Knowledge Technologies
Rkvalley-516330

CERTIFICATE

This is to certify that the thesis entitled " UART USING VERILOG " submitted by

D.RAJANI (R141023)

in the department of "ELECTRONICS AND COMMUNICATION ENGINEERING" in the partial fulfillment of requirements for the award of degree of Bachelor of Technology in Electronics and Communication Engineering for the Academic year 2018-2019 carried out the work under the supervision of

GUIDE Name	Head Of the Department
B. BHASKAR Sir	A. SREEKANTH REDDY Sir (Assistant Professor)

ACKNOWLEDGEMENT

The satisfaction that accompanies the successful completion of any task would be incomplete without the mention of the people who made it possible and whose constant guidance and encouragement crown all the efforts success. We are extremely grateful to our respected Director, Prof.H.SUDARSHAN RAO SIR for fostering an excellent academic climate in our institution. We also express my sincere gratitude to our respected Head of the Department Mr. A.SREEKANTH REDDY SIR for his encouragement, overall guidance in viewing this project a good asset and effort in bringing out this project.We would like to convey thanks to our Internal guide at college Mr. B. BHASKAR SIR for his guidance,encouragement, co-operation and kindness during the entire duration of the course and academics.

Our sincere thanks to all the members who helped me directly and indirectly in the completion of project work. We express my profound gratitude to all our friends and family members for their encouragement.

DECLARATION

I hereby declare that the project work entitled “ UART Using Verilog ” is a bonafide work carried out by me under the guidance of B. BHASKAR Sir , in partial fulfillment for the award of the degree of “ Bachelor of Technology ” in Electronics and Communication Engineering.

ABSTRACT

UART stands for Universal Asynchronous Receiver / Transmitter. It's not a communication protocol like SPI and I2C, but a physical circuit in a microcontroller, or a stand-alone IC. A UART's main purpose is to Transmit and Receive Serial data. One of the best things about UART is that it uses only Two wires to transmit data between PC and it's peripherals.

CONTENT

FIGURES REPRESENTATION

1. PROJECT STATEMENT

Develop a UART (Universal Asynchronous Receiver and Transmitter) using Verilog and Analysing the serial communication through UART transmitter and receiver.

2.VLSI AND IT'S IMPORTANCE

Let us understand the basic knowledge from our ancestors. In the early 19 th century the first vacuum tube was invented and then a computer is made from several vacuum tubes. As the vacuum tubes occupies more area, the computer was huge, more power is dissipated and the operation is also slow but the computer which is designed is successful. After the invention of vacuum tubes, “transistors” were invented in the mid 19 th century so the revolution has began. The main advantage of the transistors compared to the vacuum tubes is their size and power dissipation. So people began to make transistors in the size of micrometers as the technology is developed. So a number of large circuits are integrated in a single chip which is very small and controlled by several set of voltages. So the complex operations are being handled by a very small device which can do several set of operation with in a span of nano seconds. So the journey of Very Large Scale Integration is started which revolutionized everything in the 20th century.

If we clearly observe the growth of vlsi from the past, there are only 100's of transistors which implements some logic and then thousands and then lakhs and then crores. So as the technology is grows on increasing we have a scope of including large set of transistors in a small area. Vlsi circuit designers, researchers every one are trying in various domains to reduce the power, time delay, area etc.,.. So VLSI plays a major role in our circuit world.

3. ABOUT XILINX SOFTWARE

Xilinx ISE is a Software tool produced by Xilinx for synthesis and analysis of HDL designs,and also used for to design FPGA products . The Xilinx ISE is primarily used for circuit synthesis and design, where as ModelSim logic simulator is used for system-level testing. By using this Software we can do the verilog programming easily when we compare to the model sim software because of the presence of extra features in Xilnx Software. Xilinx's patented algorithms for synthesis allow designs to run up to 30% faster than competing programs, and allows greater logic density which reduces project time and costs. And also by using the Xilinx software we can dumb the verilog code into the FPGA kit where as in the model sim we can't do it.So,we use mostly Xilinx software for the desigining a FPGA product. We used Xilinx software to implement our project successfully by write verilog code.

4. HISTORY:

Some early telegraph schemes used variable-length pulses (as in Morse code) and rotating clockwork mechanisms to transmit alphabetic characters. The first seria communication devices(with fixed-length pulses) were rotating mechanical switches (commutators). Various character codes using 5, 6, 7, or 8 data bits became common in teleprinters and later as computer peripherals. The teletypewriter made an excellent general-purpose I/O device for a small computer.

Gordon Bell of DEC designed the first UART, occupying an entire circuit board called a line unit, for the PDP series of computers beginning with he PDP- According to Bell, the main innovation of the UART was its use of sampling to convert the signal into the digital domain, allowing more reliable iming than previous circuits that used analog timing devices with manually adjusted potentiometers. To reduce the cost of wiring, backplane and other components, these computers also pioneered flow control using XON and XOFF characters rather than hardware wires.

DEC condensed the line unit design into an early single-chip UART for their own use. Western Digital developed this into the first widely available single-chip UART, the WD1402A, around 1971. This was an early example of a medium-scale integrated circuit. Another popular chip was the SCN2651 from the Signetics 2650 family.

An example of an early 1980s UART was the National Semiconductor 8250 used in the original IBM PC's Asynchronous Communications Adapter card. In the 1990s, newer UARTs were developed with on-chip buffers. This allowed higher transmission speed without data loss and without requiring such frequent attention from the computer. For example, the popular National Semiconductor 16550 has a 16 byte FIFO, and spawned many variants, including the 16C550, 16C650, 16C750, and 16C850.

Depending on the manufacturer, different terms are used to identify devices that perform the UART functions. Intel called their 8251 device a "Programmable Communication Interface". MOS Technology 6551 was known under the name "Asynchronous Communications Interface Adapter" (ACIA). The term "Serial Communications Interface" (SCI) was first used at Motorola around 1975 to refer to their start-stop asynchronous serial interface device, which others were calling a UART. Zilog manufactured a number of Serial Communication Controllers or SCCs.

After most IBM PC compatible computers removed the RS-232 COM port in the 2000s, you could instead use an external USB-to-UART bridge. FTDI is one supplier of these chips.

5. INTRODUCTION TO UART

The UART can be used to control the process of breaking the parallel data from the PC down into serial data that can be transmitted vice versa for receiving data. The UART allows the device to communicate without the need to be synchronized.

UART is a popular method of Serial Asynchronous Communication typically, the UART is connected between a processor and peripheral. To the processor, vice versa for the peripheral.

The implementation of UART the serial communication is done with high data rate and no interrupts. Baud rate generator provides high data rate and interrupts controller handles all interrupts.

The UART serial communication interface device receives data and converts data from serial to parallel, where as the transmitter performs parallel to serial conversion.

The ability to convert data from serial to parallel and vice versa using shift registers. On-chip bit rate(baud rate) generators to control transmit and receive data rate.

6.WHAT IS UART ?

Tha UART full form is “ Universal Asynchronous Receiver and Transmitter “ and it is an in-built IC with in a microcontroller but not like a communication protocol (I2C and SPI). The main function of UART is to serial data communication. In UART the communication between two devices can be done in two ways namely Serial Communication and Parallel Communication.

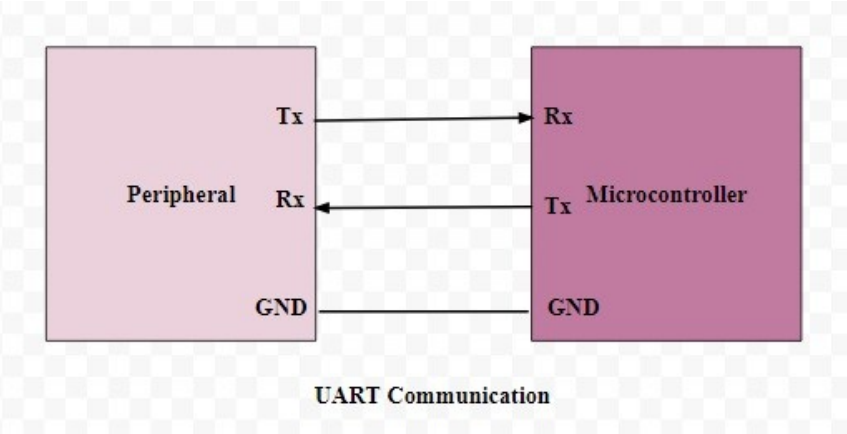


fig 6.a UART Communication

6.1 Serial and Parallel Communication

In serial data communication, the data can be transferred through the single cable or line in a bit by bit form and it requires two cables. Serial data communication is not an expensive when we compared with parallel communication. It requires very less circuitry as well as wires. Thus, this communication is very useful in compound circuits compared with parallel communication.

In parallel data communication, the data can be transferred through multiple cables at once. Parallel data communication is expensive as well as very fast, as it requires additional hardware and cables. The best examples for this communication are old printers, PCI, RAM, etc.

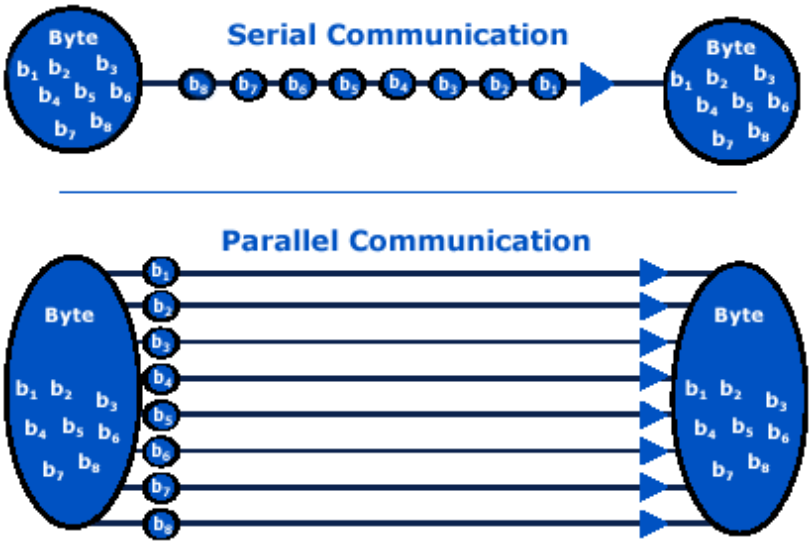


Fig 6.1.a. Serial vs Parallel Communication

7. UART BLOCK DIAGRAM

The UART block diagram consists of two components namely the UART transmitter and UART receiver which is shown in fig. 6.a . The transmitter section include three blocks namely-

Transmit Hold Register ,
Shift Register and
Control logic .

Likewise, the receiver section includes -

Receive Hold Register ,
Shift Register and
Control logic .

These two sections are commonly provided by a BAUD RATE generator. This generator is used for generating the speed when the transmitter section and the receiver section has to transmit or receive the data.

The hold register in the transmitter comprises the data byte to be transmitted. The shift registers in the transmitter and the receiver move the bits to the right or left till a byte of data is transmitted or received. A read or write control logic is used for telling when to read or write.

The baud rate generator among the transmitter and receiver generates the speed that ranges from 110 bps to 230400 bps. Typically the baud rates of microcontrollers are 9600 to 115200.

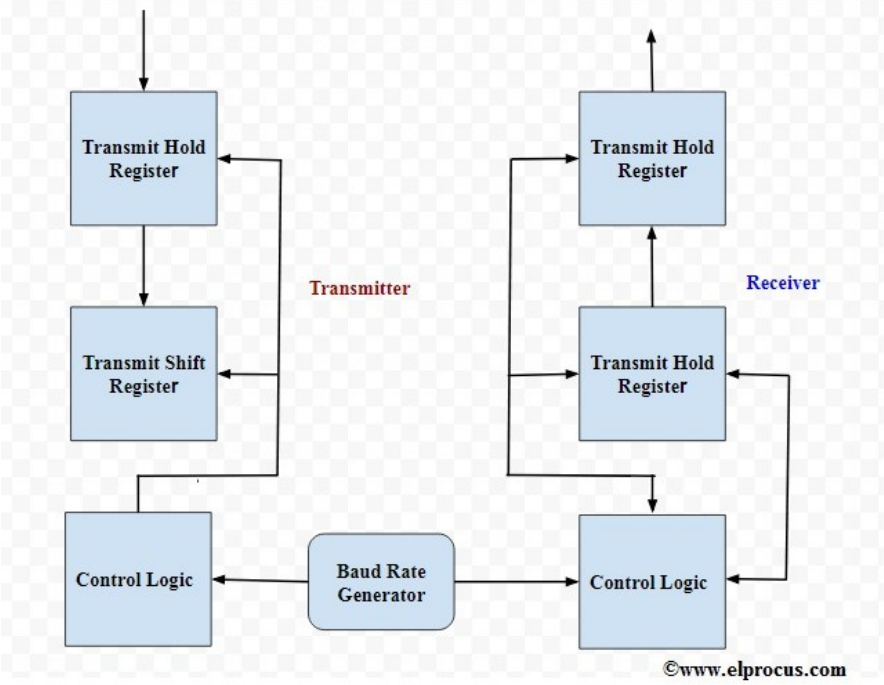


fig 7.a UART block diagram

8. WORKING OF UART

The UART that is going to transmit data receive the data from a data bus. The data bus is used to send data to the UART by another device like CPU, memory or microcontroller. The data is transferred from the data bus to the transmitting UART in a parallel form. After the transmitting UART gets the paralle data from the data bus, it adds a start bit, a parity bit and a stop bit, creating the data packet. Next, the data packet is output serially, bit by bit at the TX pin. The receiving UART reads the data packet bit by bit as its RX pin. The receiving UART then converts the data back into parallel form and the removes the start bit, parity bit and stop bits. Finally, the receiving UART transmits the data packet in parallel to the data bus on the receiving end.

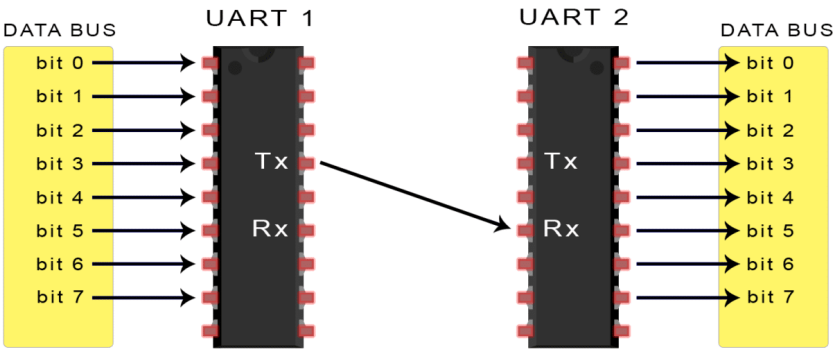


fig 8.a UART Transmission

UART transmitted data is organized into packets. Each packet contains 1 start bit, 5 to 9 data bits (depending on the UART), an optionalparity bit and 1 or 2 stop bits.

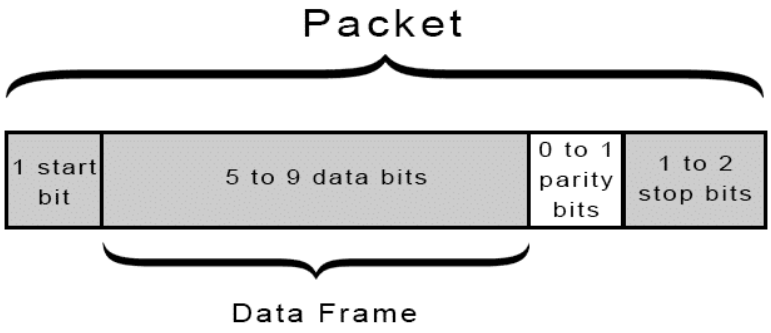


Fig 8.b Data packet of UART

8.1 START BIT

The UART transmission line is normally held at a high voltage level when it's not transmitting data. To start the transfer of data. The transmitting UART pulls the transmission line fron high to low for one clock cycle. When the receiving UART detects the high to low voltage transition, it begins reading the bits in the data frame at the frequency of the baud rate.

8.2 DATA FRAME

The data frame contains the actual data being transferred. It can be 5 bits up to 8 bits long of a parity bit is used. If no parity bit is used, the data frame can be 9 . In most cases, the data is sent with the least significant bit first.

8.3 PARITY BIT

Parity describes the evenness or oddness of a number. The parity bit is a way for the receiving UART to tell if any data has changed during the transmission. Bits can be changed by electromagnetic radiation, mismatched baud rates, or long distance data transfers. After the receiving UART reads the data frame, it counts the number of bits with a value of 1 and checks if the total is even or odd number. If the parity bit is a 0 it is even parity, the 1 bits in the data frame should total to an even number. If the parity is a 1 i.e, odd parity, the bits in the data frame should total to an odd number. When the parity bit matches the data, the UART knowns that the transmission was

free of errors. But if the parity bit is 0, and the total is odd or the parity bit is a 1 , and the total is even, the UART knows that bits in the data frame have changed.

8.4 STOP BIT

To signal the end of the data packet, the sending UART drives the data transmission line from a low voltage to a high voltage for at least two bit duration.

9. STEPS OF TRANSMISSION

1. The transmitting UART receives data in parallel from the data bus.

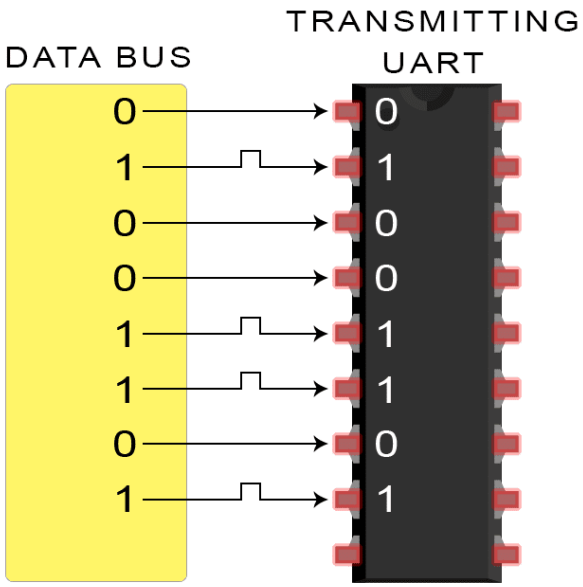


Fig.9.a UART Transmission

2. The transmitting UART adds the start bit , parity bit and the stop bits to the data frame:

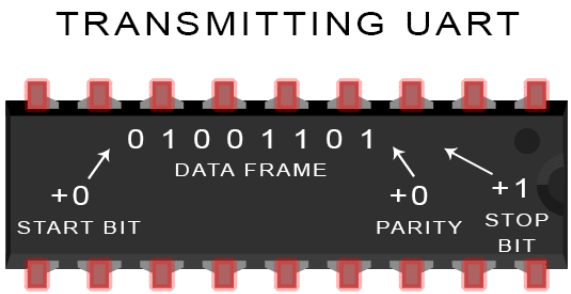


fig 9.b Adding start, parity and stop bits to the data frame

3. The entire packet is sent serially from the transmitting UART to the receiving UART. The receiving UART samples the data line at the pre- configured baud rate.

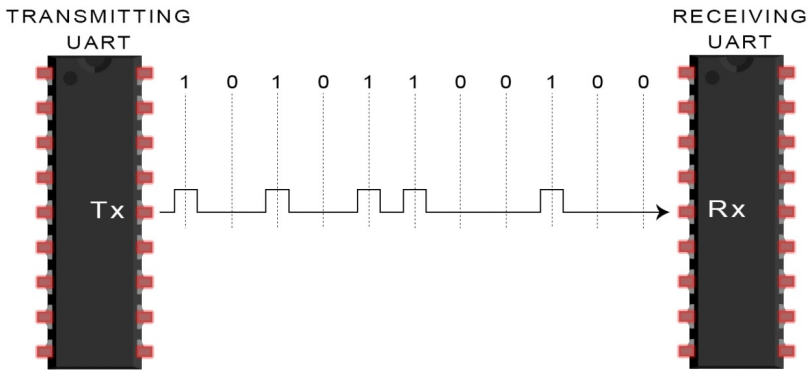


Fig 9.c Communication between TX UART and RX UART

4. The receiving UART converts the serial data back into parallel and transfers it to the data bus on the receiving end.

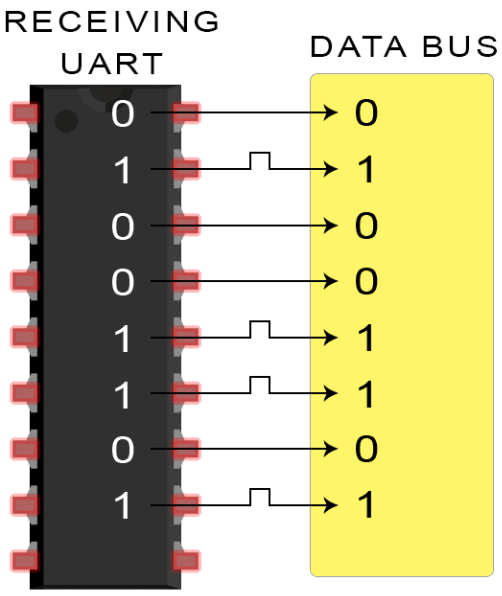


Fig 9.d Receiving UART

10. APPLICATIONS OF UART

Uart is normally used in microcontrollers for exact requirements and these are also available in various communication devices like wireless communication, GPS units, Bluetooth module and many other applications.

Uart is also used in FPGA Spartan 3e, Spartan 6e and Keyboard.

11.1 ADVANTAGES OF UART

- It uses only two wires
- It doesn't require clock signal
- It has a parity bit to allow for error checking
- Well documented and widely used method
- There is no complexity in the circuit
- Low cost
- Used widely for long distances

11.2 DISADVANTAGES OF UART

- The size of the data frame is limited to a maximum of 9 bits excluding parity bit.
- More time consuming process compare to parallel communication

12. VERILOG CODE FOR UART

12.1. TRANSMITTER UART

```
module uart (
input [7:0]data,
input wr,clk,ret,
output tx);
reg tx_start,buad_clk;
reg [10:0]data_frame;
reg [2:0]count;
always@(posedge clk)
begin
if(ret==0)
tx_start=0;
else
tx_start=1;
end
always@(posedge clk)
begin
if(tx_start==1)
count=count+1;
else
count=0;
end
always@(posedge clk)
begi
if(count==7)
buad_clk=1;
```

```
else
buad_clk=0;
end
always@(posedge clk)
begin
if(ret==0)
data_frame={1'b0, data, ^data, 1'b1};
if(buad_clk==1)
begin
data_frame=data_frame>>1;
end
else
data_frame=data_frame;
end
assign tx=data_frame[0]
endmodule
```

12.2. RECEIVER UART

```
module uart_rx (
input wr,clk,ret,rx,
output reg [7:0]data_out);
reg rx_start,buad_clk;
reg [10:0]data_frame;
reg [2:0]count;
always@(posedge clk)
begin
if(ret==0)
rx_start=0;
else if(wr==1)
rx_start=1;
else
rx_start=0
end
always@(posedge clk)
begin
if(ret==0)
count=0;
else if(rx_start==1)
count=count+1;
else
count=0;end
always@(posedge clk)
begin
if(ret==0)
buad_clk=0;
```

```
else
if(count==1)
buad_clk=1;
else
buad_clk=0;
end
always@(posedge clk)
begin
if(ret==0)
data_frame=11'b0;
else if(buad_clk==1)
begin
//data_frame=0;
data_frame={rx, data_frame[10:1]};
//data_frame=data_frame>>1;
end
else
data_frame=data_frame;
end
always@(posedge clk)begin
if(ret==0)
data_out=8'b0;
else if(data_frame[0]==1)
begin
data_out<=data_frame[9:2];
data_frame<=11'b0;
end
```

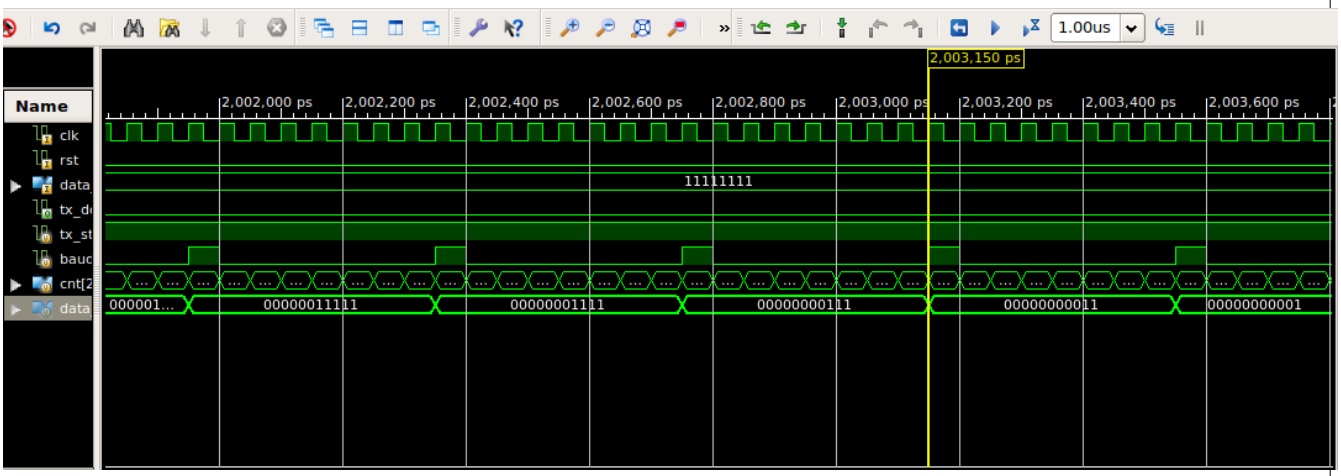
```
else
data_out=8'b0;
end
endmodule
```

12.3 UART TX-RX CODE

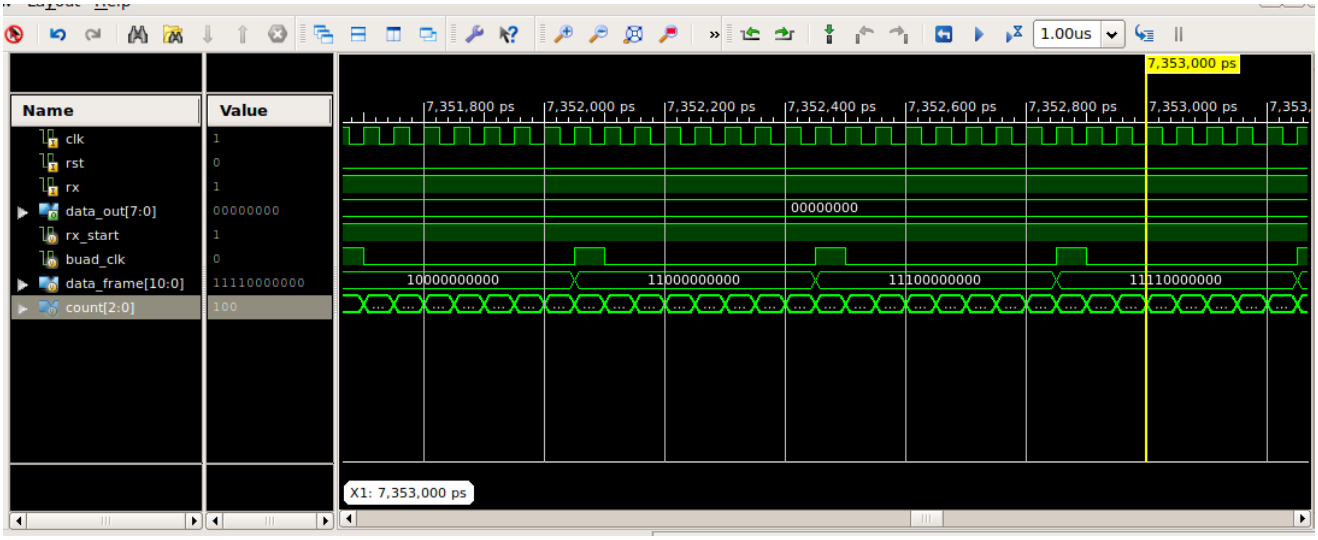
```
module uarttxrx (
input clk1,ret1,wr1,
input [7:0]data1,
output [7:0]data_out1);
wire tx1;
uart DUT (.data(data1),.wr(wr1),.clk(clk1),.ret(ret1),.tx(tx1));
uart_rx
dut1(.wr(~wr1),.clk(clk1),.ret(ret1),.rx(tx1),.data_out(data_out1)
);
endmodule
```

13. SIMULATION

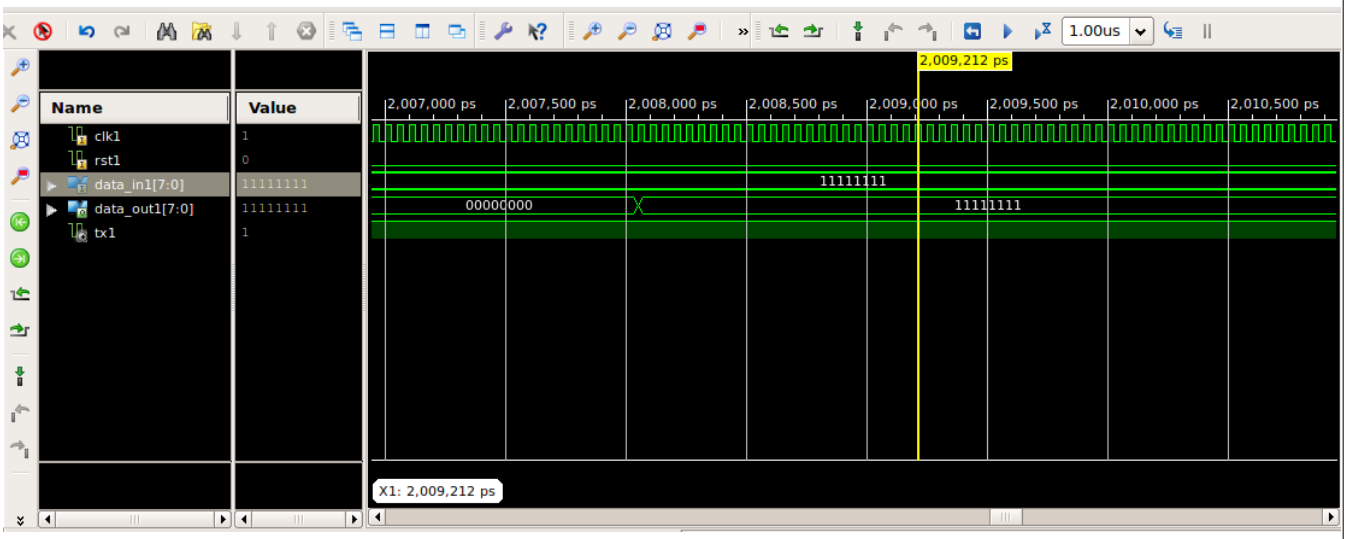
13.1. TRANSMITTER UART OUTPUT



13.2 RECEIVER UART OUTPUT



13.3. UART TX-RX OUTPUT



14. CONCLUSION

The design of UART has been done successfully by using Verilog code. The conversion of parallel data into serial data and serial data into parallel data for the communication is done. It shows that the transmitter transmits the data and receiver receives the data with the same baud rate. By this there is no loss of time as it transmit and receive at the same baud rate.


THANK YOU