Final report

On

Universal Asynchronous Receiver Transmitter

Submitted by

Anand Kumar

MOTILAL NEHRU NATIONAL INSTITUTE OF TECHNOLOGY, ALLAHABAD

CONTENT

ABSTRACT

Universal Asynchronous Receiver Transmitter (UART) is widely used in data communication processes especially for its advantages of high reliability, long distance and low cost. In this paper, we present the design of 8-bit UART modules based on Verilog HDL.We have implemented the VLSI design of the module and pass data between the proposed 8-bit UART with another 8-bit UART.

Here we have defined three control bits. They are Start bit, Stop bit and Transmission bit. Using these bits, we control the data transmission between one UART and another UART. We are transmitting 8-bit data when Start =0 and Stop =1. In order to avoid data collision, we are using a transmission bit which tells the data transmission path and which UART is busy. The testing and verification of the output is done using Xilinx ISE Project Navigator (14.7)

1.0. INTRODUCTION TO UNIVERSAL ASYNCHRONOUS RECEIVER TRANSMITTER

1.1. INTRODUCTION

Verification of any IC ’s in VLSI design requires a lot of effort and measuring the efficiency is tedious.As the devices design becomes more complex,verification also becomes more complex.An UART stands for Universal Asynchronous Receiver and Transmitter. The UART mainly intends in exchanging information between two devices with high reliability and capability to transmit any data over a long distance..An UART is a device which is used for both the transmission and reception of data or any information in serial manner and in asynchronous way..UART uses asynchronous serial data between devices.In asynchronous transmission the sender and receiver doesn’t have any clock signal compared to the synchronous transmission. It is mainly used in device to device communication.

1.2. OVERVIEW OF UART

UART is one of the most widely used interfaces.For UART we use two UART’s modules namely transmitter module and receiver module. The transmission UART module converts parallel information into serial information.The reception UART module converts serial information into parallel information.The main good point of using UART is only two wires will be needed for transmission and reception process between two UART’s.Since uart is asynchronous there is no need for an external clock to synchronize the data. The frame format for UART is given in Figure

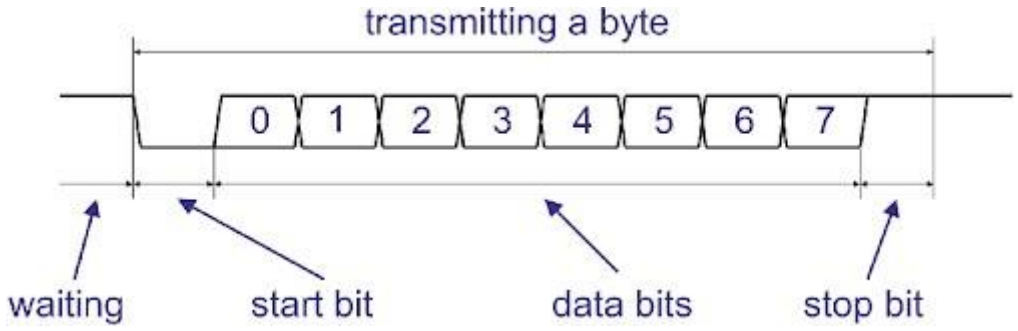

Fig1.1 UART Bit Format

If the start bit is found the UART reads the bits at some frequency which is called as Baud rate which has the unit as bits per second.Baud rate is mainly for measuring the transmission speed.The baud rate should be same for both transmitting and the receiving UART.Information is transmitted in parallel from the transmitter module to receiver module.Then data frame will be created by adding start,parity,stop bits.The parity bit is used as a checking bit at the receiver side.In that two types of parity bits are used they are odd parity and even parity bits.Data frame is sent in serial manner on the transmitter pin and receiver reads the data bit on its other end.Then again it will be converted into parallel form by eliminating the start bit,stop bit and parity bit.The data that has to be transmitted will bordered in frames.

2.0 PROJECT WORK

2.1 BLOCK DIAGRAM OF UART

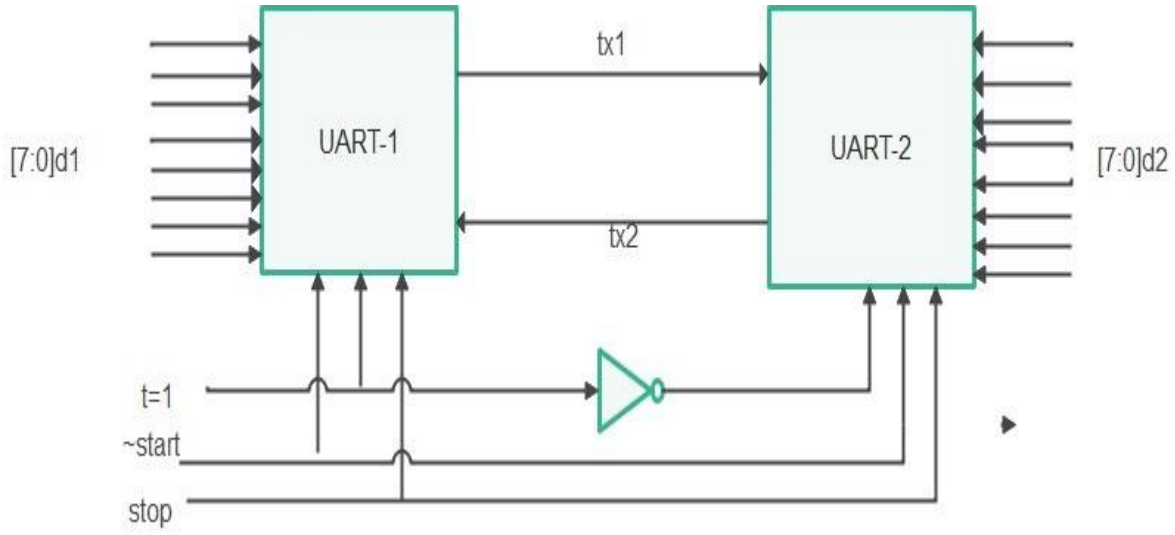

Fig:1.2. Block Diagram of UART

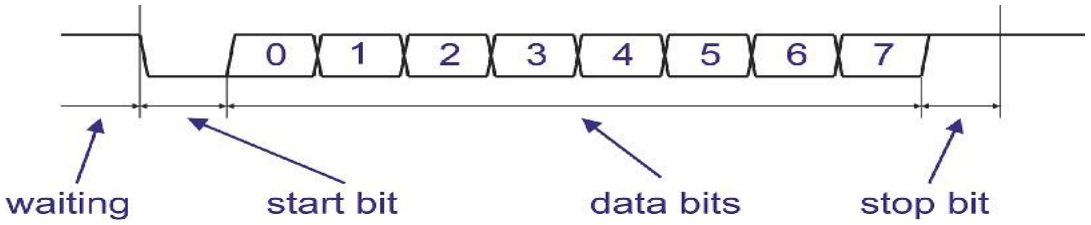

Fig:1.3.Bit format of UART

OPERATION:

Here, we are transferring 8-bit of data from UART-1 to UART-2 by considering three control signals i.e., Start, Stop and Transmission Bit. Initially UART will be in ideal state, when the start bit goes low, which is nothing but UART has received start bit, if transmission control t=1, then the data will be transmitted to UART-1 to UART-2 through transmission lines Tx1 in a serial manner. Tx2 will be in ideal mode. After transmitting 8-bit data, it will check for the stop bit. If it is high, the data transmission ends.

When t=0, UART-1 will receive the data from UART-2 through tx2 and tx1 will be in ideal mode. After transmitting 8-bit data, it will check for the stop bit. If it is high, the data transmission ends.

If Start bit is high, orelse Stop bit is low, there will not be any exchange of data between UART-1 and UART-2. And both the UARTs will be in its previous state.

2.2 VERILOG UART CODE

```
module  UART_f(input clk,t,input start,stop,output reg tx1,tx2);

reg [7:0]d1;//UART-1 data
reg [7:0]d2;//UART-2 data

always @(posedge clk)

Begin

   case({start,stop})// depends on start bit and stop bit
   2'b01 : //when start=0 and stop=1 then only data exchange takes place.

                        begin
                        if(t) //when t=1 data is transmitted from UART-1 to UART-2
                                begin
                                        tx1=1'b1;//Transmission line tx1 is busy.
                                        tx2=1'b0;//Transmission line tx-2 is idle.
                                        d2=d1;//d1 data is transmitted at d2 side.
                                 end
                        else //when t=0 data is transmitted from UART-2 to UART-1
                                begin
                                        tx1=1'b0;//Transmission line tx2 is idle.
                                        tx2=1'b1;//Transmission line tx2 is busy.
                                        d1=d2;//d2 data is transmitted at d1 side.
                                end
                        end

  default: //no data exchange takes place
                                begin
                                        tx1=1'b0;//Transmission line tx1 is idle.
                                        tx2=1'b0;//Transmission line tx1 is also idle.
                                        d1=d1;//next state is nothing but previous state itself
                                        d2=d2;
                                end
  endcase

end
```

endmodule

2.3 TESTBENCH FOR UART

```
module uarttest;

	// Inputs
	reg clk;
	reg  t;
	reg start;
	reg stop;

	// Outputs
	wire tx1;
	wire tx2;

	// Instantiate the Unit Under Test (UUT)
	UART_f uut (.clk(clk),.t(t),.start(start),.stop(stop),.tx1(tx1), .tx2(tx2));

	initial begin
		// Initialize Inputs
		clk = 1;

		t = 1;// data will be transmitted from UART-1 to UART-2
		start = 0;
		stop=1;
		uut.d1=8'b10010001; //As start =0 and stop=1 this data will be received at reg d2

		#100;uut.d1=8'b10011111;//after 100ns As start =0 and stop=1 this data will be received at reg d2

		#100;uut.d2=8'b10000111;// data given to reg d2
					t=0;//data will be transmitted from UART-2 to UART-1
					start= 0;
					stop=1;
		#100;uut.d2=8'b10100111;
					start = 0;//start bit is low
					stop=0;//instead of  high,  stop bit is low so no data will be exchange
					t=1'bx;//As no data will be exchanged t is dont care
		#100;uut.d2=8'b10010111;//data in reg d2
```

```
                                start= 1;//instead of low, start bit is low so no data will
be exchange
                                stop=1;
                                t=1'bx;//As no data will be exchanged t is dont care
                        #100;uut.d1=8'b10100101;//data in reg d1
                                start = 0;
                                stop=1;
                                t=1;// data will be transmitted from UART-1 to UART-2
                        #100;uut.d2=8'b00000000;//data in reg d2
                                start = 0;
                                stop=1;
                                t=0;//data will be transmitted from UART-2 to UART-1
                        #100;uut.d1=8'b11111111;//data in reg d1
                                start = 0;
                                stop=1;
                                t=1;// data will be transmitted from UART-1 to UART-2
                        #100;uut.d2=8'b00000000;//data in reg d2
                                start = 0;
                                stop=1;
                                t=0;//data will be transmitted from UART-2 to UART-1

            end
        always #50 clk=~clk;// the time period of the clock is 100ns
endmodule
```

2.4 OUTPUT WAVEFORM OF UART

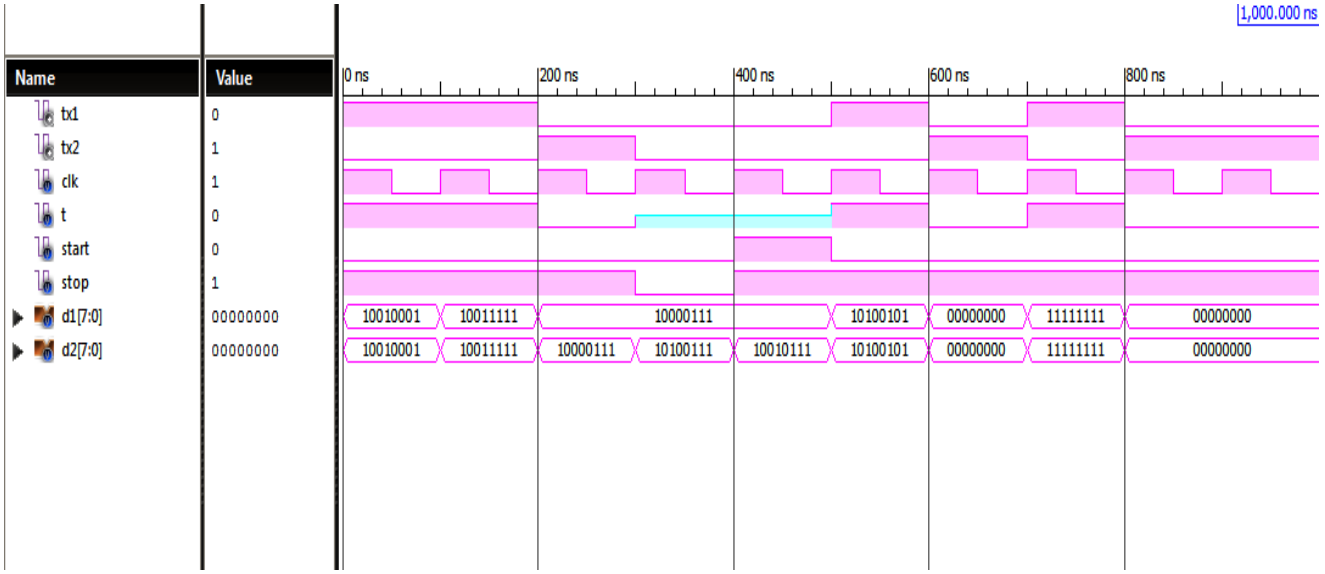

Fig 1.4 Output waveform of UART

3.0 ADVANTAGES & DISADVANTAGES AND APPLICATIONS OF UART

3.1 ADVANTAGES:

Following are the **advantages of UART**:

- Hardware complexity is low.
- As this is one to one connection between two devices, software addressing is not required.
- Due to its simplicity, it is widely used in devices having 9 pin connectors.

3.2 DISADVANTAGES:

Following are the **disadvantages of UART:**

- It is suitable for communication between only two devices.
- It supports fixed data rate between devices wanting to communicate otherwise data will be garbled.

3.3 APPLICATIONS:

Following are the **applications of UART**

- Achieving communication amongst distant computers around 900 meters.
- Transferring data through PC serial port.
- Baud rate generation for numerous applications that helps to determine the speed of data transmission.

REFERENCES:

1. Verilog HDL(A guide to Digital Design and Synthesis)-by Samir Palnitkar
2. [online] Available: www.wikipedia.org/uart.
3. IEEE paper- The Design of High Speed UART Ritesh Kumar Agrawal, Vivek Ranjan Mishra VLSI Department, VIT University
4. Design of UART Using Verilog And Verifying Using UVM B.Priyanka and Gokul.M