iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 2 **Issue:** XI **Month of publication:** November 2014

DOI:

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

ISSN: 2321-9653
International Journal for Research in Applied Science & Engineering Technology (IJRASET)

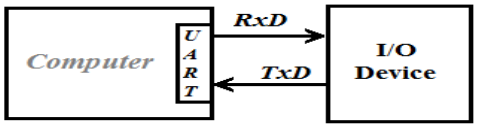
Implementation of UART Using VHDL

Indu Chauhan
M.Tech in VLSI and Microelectronics, Shobhit University Meerut

Abstract— UART (Universal Asynchronous Receiver Transmitter) is a kind of serial communication protocol; mostly used for short-distance, low speed, low-cost data exchange between computer and peripherals. During the actual industrial production, sometimes there is no need the full functionality of UART, but simply integrate its core part. UART includes three kernel modules which are the baud rate generator, receiver and transmitter. The UART implemented with VHDL language can be integrated into the FPGA to achieve compact, stable and reliable data transmission.
Keywords— UART, asynchronous serial communication, VHDL, simulation, Xilinx

I. INTRODUCTION

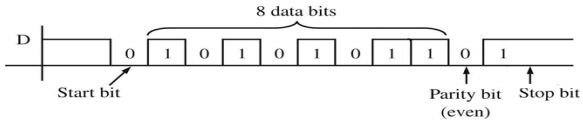
Universal Asynchronous Receiver Transmitter is a circuit that controls a computer's interface to its attached devices. It provides the computer interface so that computer can exchange data with devices. The UART takes bytes of data and transmits the individual bits in a sequential fashion. It performs all the tasks (e.g. parity checking etc.) needed for the communication.



When transmitting data UART, receives the data parallel from the application, and sends it serially on TxD pin, and when receiving, the UART receives the data serially on RxD pin, and provides the parallel data to the application.

A. Data Encoding

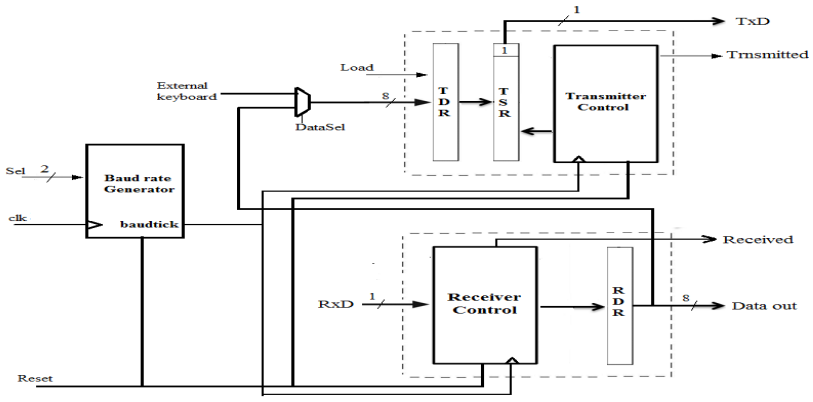
When no data transmitted D remains high. To signal a start of new transmission D goes low for 1 bit period, which is known as "Start bit". After the Start Bit, the individual bits of data are sent. Each bit in the transmission is transmitted for exactly the same amount of time as all of the other bits. After transmission of entire data has been sent then D again goes to high, the last bit known as "Stop bit". Transmission of next data can begin any time after that. In our project we have "even parity" bit for parity check.



II. IMPLEMENTATION OF UART

A UART is composed of three main component receiver transmitter and baud rate generator

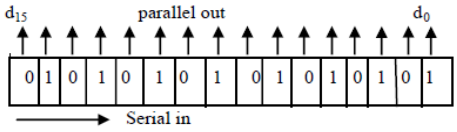
A. UART block diagram



B. UART Registers

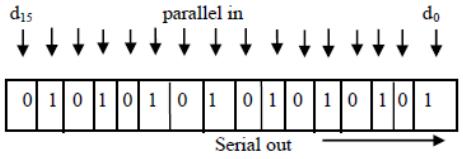
In this paper 16 bit registers are used in place of 8 bit registers. Different registers used in UART are discussed below:

1)RSR: Receiver Shift Register, it is a shift register used at the receiver end of UART .It receive the bits sequentially from RxD (receiving data pin) and shift them to right, one at each bit clock. As shown in Fig. serial in parallel out shift register is used in receiver.



serial in parallel out shift register

2)RDR: Receiver Data Register, it receives data from RSR and whatever data stored in RDR is placed on the data bus.
3) TDR: Transmit Data Register, it receives bytes of data from data bus to be transmitted and transfer it in TSR.
4) TSR: Transmit Shift Register, it is a shift register used at transmitter side it is used to transmit data bitwise by shifting each bit to right. Parallel in serial out shift register is used

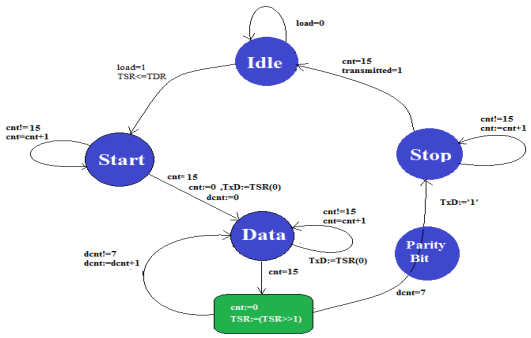


parallel in serial out shift register

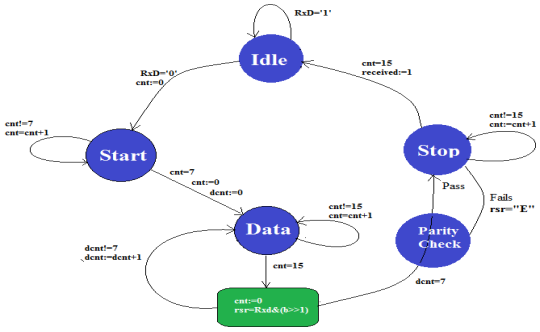
Inputs:
Sel :Selects baud rate(4 options are available)
RxD: Data coming from PC serial port.
DataSel: Selects whether data to be transmitted is coming from external keyboard attached to fpga or data coming from pc serial port.
Reset: Resets all components.
External Keyboard

Outputs:
TxD: Data is sent to pc serial port and displayed in Tera Term.
Data Out: Data received is shown on LED's on fpga. In case parity check fails, then output is always shown as "E".

1) *Transmitter Module:* Transmitter takes parallel data and sends it serially on the TxD pin. The transmitter consists of TDR (Transmit Data Register), TSR (Transmit Shift Register) and controller. As load signal goes high transmitter transfers data from TDR to TSR and outputs start bit "0" to the TxD pin then shifts TSR right eight times to transmit 8 bits. When eight data bits transmitted ,transmitter sends parity bit and finally outputs stop bit"1" to the TxD pin and signal "transmitted" goes high.



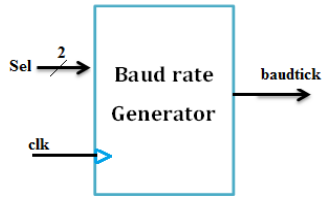
2) *Receiver Module:* Receiver takes data serially in RxD pin, and provides the parallel to the Data out pin. UART receiver consists of RDR (Received Data Reg.) and controller. When the UART detects start bit receiver reads and shifts 8 data bits serially into a temporary register. When 8 data bits has been received and parity check passes then after stop bit has been received controller transfers data from temporary register to RDR and received signal goes high. If parity check fails then , output of receiver is always shown as **"E"**.



Receiver state machine

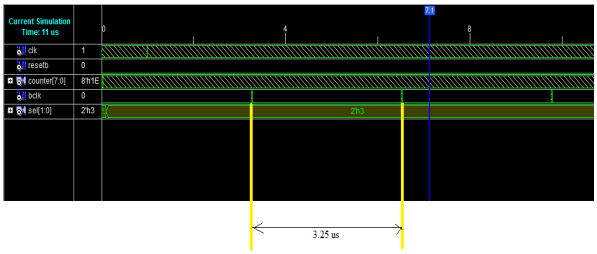
3) *Baud Rate Generator:* The baud rate generator generates a sampling signal whose frequency is exactly 16 times UART's baud rate. If the baud rate is X, the sampling rate has to be 16*X ticks per second. Assume the system clock rate is 50 MHz the baud generator needs a mod-m ($50*10^6/16*X$) counter, in which 1 clock-cycle-tick asserted once every m clock cycle.

The baud generator has 2-select bit to decide baud rate, since we are using two bits, we have the choice of four baud rates.



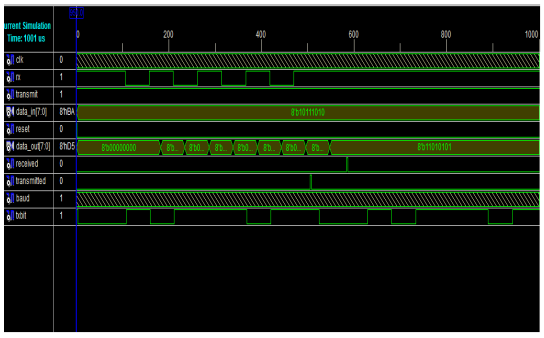
III. SIMULATION OF MODELS

Baud Rate: -19200



TestBench (Receiver and Transmitter):

1)Receiver and Transmitter are working independently(No Parity Check Here Receiver and Transmitter are working independently(No Parity Check Here).



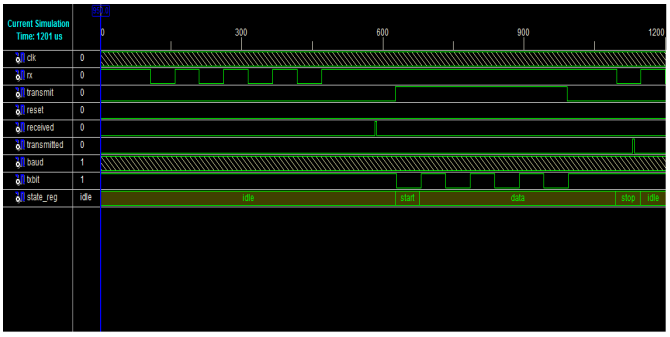
Receiver:

RxD(rx)= START_BIT 1 0 1 0 1 0 1 1 STOP_BIT
Received data(Data_out)= "11010101"

TRANSMITTER:

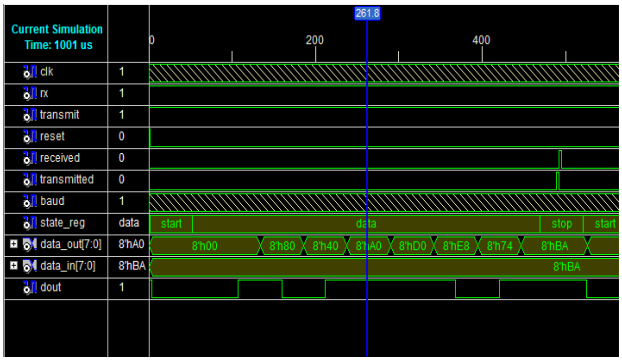
Transmitted data(Data_in)= "10111010"
TxD(txbit)= START_BIT 0 1 0 1 1 1 0 1 STOP_BIT

1) Output of receiver is going to input of transmitter(No Parity Check Here)

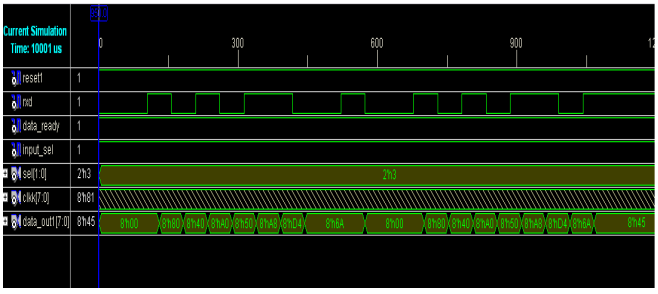


RxD(rx) = **START_BIT 1 0 1 0 1 0 1 1 STOP_BIT**
Trasmit signal is given at around 600 us.
TxD(txbit)= **START_BIT 1 0 1 0 1 0 1 1 STOP_BIT**

3)Output of transmitter is going to input of receiver(No Parity Check Here).



4)Parity check(even parity).



Here I am sending same data two times , first time with wrong parity , and 2nd time with correct parity.
Rxd = START_BIT 0 1 0 1 0 1 1 0 Parity_bit(0) STOP_BIT
Hence first time output is h6A (since parity is correct).

Ans 2nd time output is h45(**"E"** since parity bit is wrong).

IV. CONCLUSION

In this paper, a design of UART has been proposed.It internally consists of transmitter, receiver and baud rate generator. The design is successfully simulated using Xilinx ISE 8.2i software. The results are stable and reliable and show the correct functionality.

V. ACKNOWLEDGEMENT

I would like to acknowledge certain people who have encouraged, supported and helped me complete my thesis at Shobhit University.

I am very greateful to my advisor Prof. R.P Aggarwal for his guidance, patience and understanding throughout this work. His suggestions, discussions and constant encouragement has helped me to get a deep insight in the field of VLSI design. I would like to thank Mr. Ritish Kumar and Mr.Hamid Ali for sparing their time to be a part of my thesis. I would also like to thank all my friends here who made my stay at SU an enjoyable and a memorable one. I extend a special thank-you to Srinivas, my constant companion and beloved friend.

REFERENCES

[1] FANG Yi-Yuan CHEN Xue-Jun, "Design and Simulation of UART Serial Communication Module Based on VHDL' Shanghai University of Engineering Science, 2011

[2] Amanpreet Kaur, Amandeep Kaur, "An approach for designing a universal asynchronous receiver transmitter (UART)", International Journal of Engineering Research and Applications (IJERA) ISSN: 2248-9622 www.ijera.com Vol. 2, Issue 3, May-Jun 2012, pp.2305- 2311

[3] en.wikipedia.org/wiki/synchronous_serial_communication.

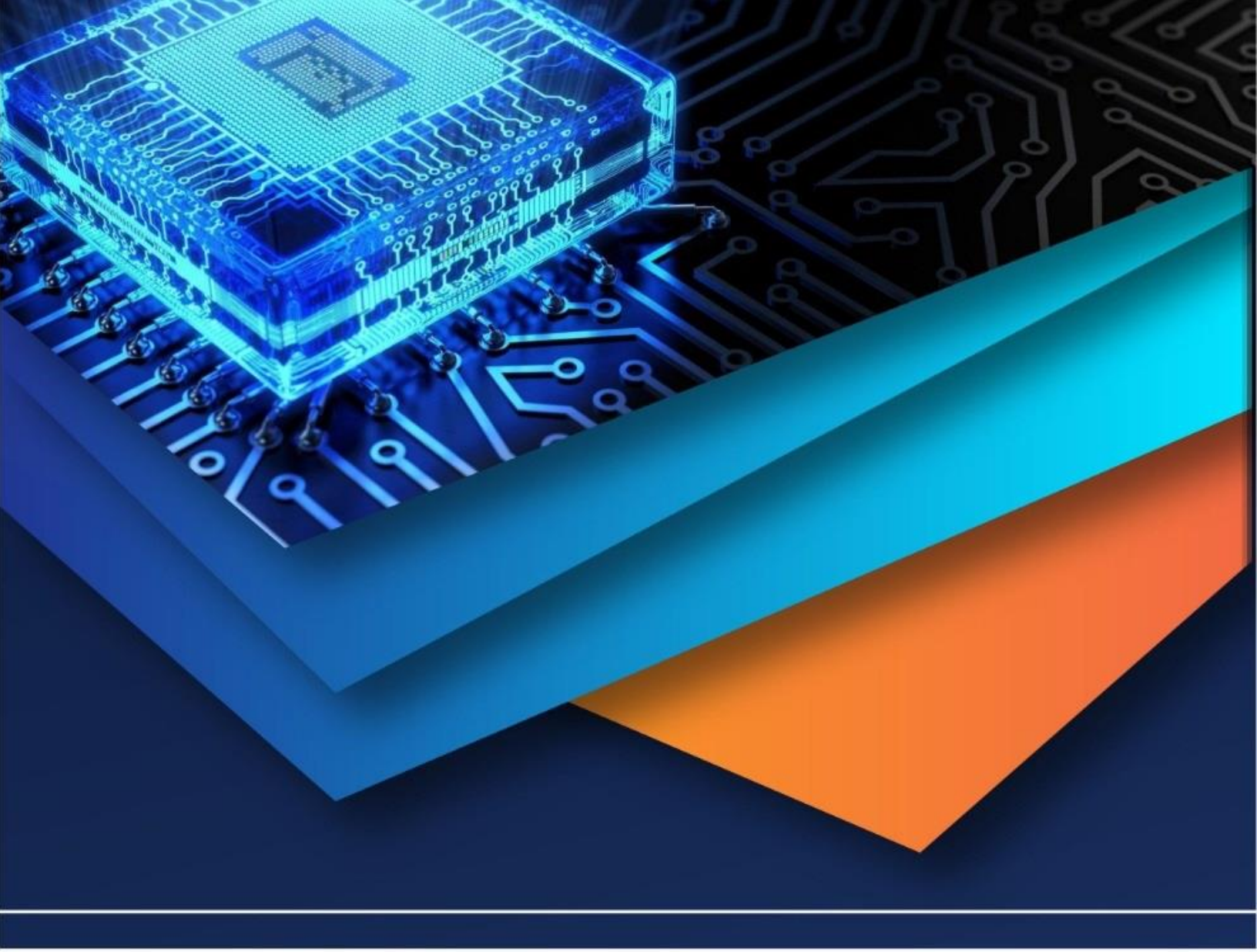
[4] Cowley, John (2007) communication and networking, A introduction, Springer, ISBN 9781846286452

[5] www.pccompi.com

[6] Shiva, S.G University of Alabama in Huntsville, AL, proceedings of the IEEE (volume 67, Issue: 12)

[7] Comparison of VHDL, Verilog and System verilog, by Stephen Bailey, www.model.com

[8] C. H. Roth, "Digital System Design Using VHDL", PWS Publishing Company, 2008cited.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY