

AREA EFFICIENT UART DESIGN USING VERILOG HDL

Dr.Ravi Bolimera

Electronics and Communication Department

Nalla Narasimha Reddy Education Society's Group of Institutions Hyderabad, India. ravib.speech@gmail.com

K. Shiva prasanna

Electronics and Communication Department

Nalla Narasimha Reddy Education Society's Group of Institutions Hyderabad, India. shivaprasanna441@gmail.com

Ranga Deepshika

Electronics and Communication Department

Nalla Narasimha Reddy Education Society's Group of Institutions Hyderabad, India. rangadeepshika@gmail.com

Shanigarapu saikiran

Electronics and Communication Department

*Nalla Narasimha Reddy Education Society's Group of Institutions Hyderabad, India.
shanigarapusaikiran1@gmail.com*

Sappidi Mallikarjun Reddy

Electronics and Communication Department

*Nalla Narasimha Reddy Education Society's Group of Institutions Hyderabad, India.
Mallikarjunreddy299@gmail.com*

Scopus ID:60113950

Abstract- With the rapid growth of Integrated Circuits (ICs) technology, the complexity of the circuits has also increased. As a result, the complexity of the circuit demands self-testability in hardware to mitigate the product failure. Built-in-self-test (BIST) is such a technique which can meet the demand of self-testability with an effective solution over costly circuit testing system. This paper represents designing and implementation of a Universal Asynchronous Receiver Transmitter (UART) with self-testing ability. In order to attain compact, stable and reliable data transmission, the UART is designed with Verilog HDL language and synthesized on Spartan2 FPGA. Here, the Baud Rate of the UART is 4 Mbps. This UART also utilizes the RS-422 standard.

Keywords— UART, Verilog HDL, Area Efficient Design, Low Power, VLSI, FPGA Implementation, Serial Communication, Resource Optimization, RTL Design, Embedded Systems.

Received: 24-09-2025

Accepted: 29-10-2025

Published: 05-11-2025

I. INTRODUCTION

Universal Asynchronous Receiver/Transmitter (UART) is a universal serial, asynchronous, full-duplex communication protocol commonly used to implement data transmission in embedded and IoT systems. In modern computer systems, UART is used widely thanks to its simple structure and long transmission distance. It is often used as a connection between computers and external devices, such as sensors, monitors, printers, etc., that transmit data using serial communication, and has a wide range of applications in the internal communication of on-board instruments. Common UART dedicated chips that convert parallel data of the computer and serial data from peripherals, like 8250, 8251, and NS6450, are more complex and are difficult to meet the current high-speed data transmission occasions. Therefore, it is of great practical significance to study and design the hardware implementation based on the UART protocol, which can solve the problem of resource wastage of the traditional UART and can make the whole design more stable and reliable, reduce the board area and simplify the circuit. Field Programmable Gate Array (FPGA) is an integrated circuit device, built on the basis of programmable devices. It has wide popularity in the field of electronics and is usually used for advanced hardware design

and verification, allowing various functions to be implemented at the hardware level. As a semi-custom circuit, compared with the traditional ASIC design method, it overcomes the disadvantage of immutable custom circuits to a certain extent and reduces the cost. At the same time, it solves the problems of the small number of gates and simple functions of the original programmable devices. It has the advantage of high flexibility and reconfigurability and is well suited for developing and verifying hardware implementations of communication protocols. Furthermore, FPGAs have programmable logic components called 'logic blocks' and a hierarchy or reconfigurable interconnects which facilitate the wiring of the blocks together. Asynchronous serial communication can be implemented easily on FPGA. The study of the hardware implementation of the transmitter and receiver based on the UART protocol is important in several ways. By optimizing and improving the hardware implementation of the UART communication protocol, the speed and stability of data transmission can be increased and the efficiency of communication can be improved. By checking the waveforms, the reliability of the communication protocol can be verified to ensure the accurate transmission and reception of the communication data. The UART communication protocol has unique advantages in some

specific application scenarios, and the study of the hardware implementation can help to explore the application prospects of asynchronous communication. By synthesizing the transmitter and receiver side hardware modules based on the UART protocol on FPGAs, the correctness and performance of the hardware design can be verified, and possible problems and defects can be identified and solved. Therefore, it is important to study the hardware implementation of transmitter and receiver based on UART protocol to promote the application and development of communication protocols in FPGAs. It helps to improve communication efficiency, verify the reliability of the communication protocol, and explore the potential of asynchronous communication. The aim of this project is to design and implement a hardware module on an FPGA chip for data transfer between two devices using the UART communication protocol. The receiver device acts as the data receiver and converts the serial data into parallel data. The transmitting device acts as the data sender and reverses the previous process, converting the parallel data into serial data and sending it out. In this study, the two achieve data transfer and interaction to verify data accuracy through parallel data communication. The tools used in this study mainly include FPGA development-related software, such as vivado design suite, and the hardware description language Verilog HDL, to carry out the hardware circuit design, simulation, programming, and testing of data transmission for the UART interface. This hardware implementation aims to validate the reliability and practicality of the UART communication protocol in FPGAs, as well as to explore the problems and challenges that may be encountered in practical applications.

II. LITERATURE SURVEY

[1] A Review on Area and Power Efficient UART Architectures for FPGA Implementation

The Universal Asynchronous Receiver Transmitter (UART) remains a fundamental communication block in embedded systems and digital interfaces. This paper presents a comprehensive review of various UART architectures emphasizing area and power efficiency for FPGA-based implementations. Several design techniques such as resource sharing, clock gating, and optimized state machine design are analyzed for their impact on area reduction and performance improvement. Comparative studies of recent UART implementations on different FPGA platforms reveal that area-efficient designs can significantly minimize logic utilization while maintaining robust data transmission. The survey concludes by highlighting trends in low-power VLSI design methodologies for communication interfaces and suggesting potential improvements using modern hardware optimization strategies.

[2] Survey on High-Performance UART Designs Using Verilog HDL for VLSI Applications

This literature survey explores advancements in high-performance UART designs implemented in Verilog HDL for VLSI systems. Traditional UART architectures often face challenges in balancing throughput, area utilization, and power consumption. Several studies have proposed enhancements in baud rate generation, data buffering, and

synchronization techniques to achieve better timing accuracy and reduced delay. The survey evaluates existing designs based on synthesis results, FPGA utilization, and power metrics, providing insights into how modern HDL-based design flows enable scalable and efficient UART architectures. The paper concludes that optimized HDL coding and modular design principles play a key role in achieving both performance and area efficiency in modern digital communication systems.

[3] A Comparative Study on UART Implementations for Low-Power Embedded Communication

With the growing demand for low-power and compact embedded systems, the design of an efficient UART has become increasingly important. This paper provides a comparative study of existing UART implementations targeting low-power and area-constrained environments. Various methodologies—such as asynchronous operation, dynamic voltage scaling, and resource optimization—are reviewed to identify techniques that best suit portable and IoT devices. Experimental data from prior research show that area-efficient UART architectures not only reduce silicon cost but also extend battery life in embedded applications. The survey concludes that Verilog-based behavioral modeling and FPGA synthesis provide an effective platform for validating and refining such designs.

[4] Analysis of UART Design Techniques for Area Optimization in VLSI Systems

This paper reviews different UART design techniques aimed at minimizing hardware resource usage in VLSI systems. The literature reveals multiple optimization strategies, including FSM simplification, data path reuse, and hardware multiplexing, which help achieve better area efficiency. Several case studies demonstrate how Verilog HDL facilitates structured modeling, rapid simulation, and synthesis with minimal logic overhead. The study also compares the trade-offs between area, speed, and power across various implementations. From the analysis, it is evident that efficient UART design not only depends on hardware minimization but also on proper clock management and communication reliability. Future trends in low-cost SoC integration and adaptive communication interfaces are also discussed.

[5] Review of Verilog-Based UART Design Methodologies for Resource-Constrained Systems

In resource-constrained digital systems, achieving compact and reliable communication modules is critical. This review paper focuses on Verilog-based UART design methodologies that emphasize area efficiency, modularity, and low complexity. The paper summarizes recent developments in UART architecture, covering improvements in baud rate generation, data handling, and synchronization mechanisms. Various research works implementing UARTs on FPGA and ASIC platforms are examined to highlight the relationship between

logic utilization and communication performance. The analysis demonstrates that careful architectural planning and HDL-level optimization can significantly reduce gate count without compromising throughput or accuracy. This survey serves as a foundation for future research in area-optimized communication controller design for embedded and SoC applications.

III.SYSTEM DESIGN

[1] Input Unit: which together form a complete digital communication system. The input unit is responsible for receiving data and control signals from external devices such as a microcontroller, computer, or sensor module. It provides the necessary serial or parallel input data, baud rate clock pulses, and control signals like transmit enable, receive enable, and reset. This unit ensures that all incoming data and timing signals are properly synchronized before being passed to the core processing module

[2] Processing Unit: The processing unit, also known as the UART core, is the main functional block of the system. It is designed in Verilog HDL and performs the primary communication tasks such as serial-to-parallel and parallel-to-serial data conversion. The processing unit consists of several submodules including the baud rate generator, transmitter, receiver, and control logic. The baud rate generator produces accurate timing pulses to maintain synchronization between the transmitter and receiver. The transmitter section converts parallel data into serial form, adding start, stop, and parity bits according to the UART protocol. The receiver section performs the reverse operation by detecting the start bit, receiving serial data, and converting it back to parallel form while checking for parity and stop bits to ensure error-free communication. The control unit coordinates the operation of all submodules, handles data flow, and manages error and status flags.

[3]Output Unit: The output unit delivers the processed data to external devices or display systems. It provides the transmitted serial data to the output line and the received parallel data to the target module or memory. Additionally, status indicators such as LEDs or digital flags can be used to show transmission completion, data readiness, or error detection. This unit ensures reliable and accurate data delivery between the UART and the connected systems.

[4] Power supply Unit: The power supply unit provides stable electrical power to the entire system. Typically, it receives a 5V DC input, which is then regulated to appropriate voltage levels such as 3.3V or 1.8V as required by the FPGA or other digital components. It ensures continuous and noise-free operation of all functional blocks, including the UART core and interface circuitry.

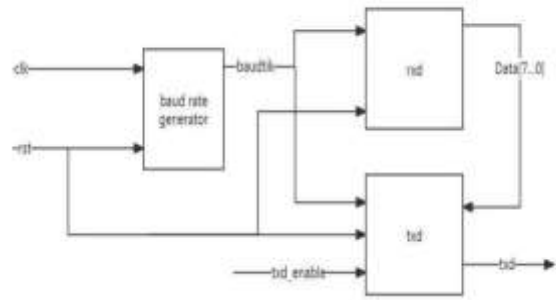


Fig.1. Block Diagram

IV.EXPERIMENTAL RESULTS

Proposed method output:

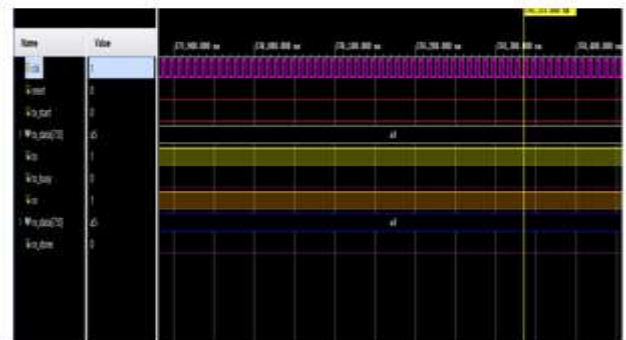


Fig 2: simulation waveform of proposed method

A complete testbench was developed in Vivado to simulate the functionality of the optimized UART design. The testbench setup included initializing the clock, applying reset, feeding the tx_data signal with a test value (0xA5), and asserting tx_start to initiate transmission. In the waveform output, we can observe that tx_data is loaded with 0xA5, and the tx_start signal is pulsed high for a single clock cycle to trigger data transmission.

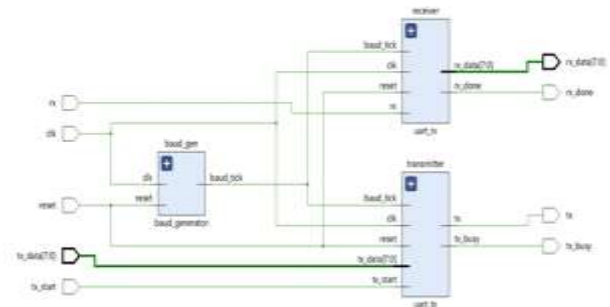


Fig 3: schematic

The given block diagram represents a complete UART system implemented in Verilog with separate transmitter, receiver, and baud rate generator modules. The baud_generator produces a baud_tick signal based on the input clock, which is

shared by both the transmitter (uart_tx) and receiver (uart_rx) modules for synchronized data timing



Fig 4: power report

Power estimation Power estimation for the optimized UART design was performed using Vivado's power analysis tools. After synthesizing and implementing the design on a Xilinx FPGA, Vivado provided an estimated power report that divides consumption into dynamic and static components. As observed in the power analysis image, the total on-chip power consumption is approximately 1.104 W.

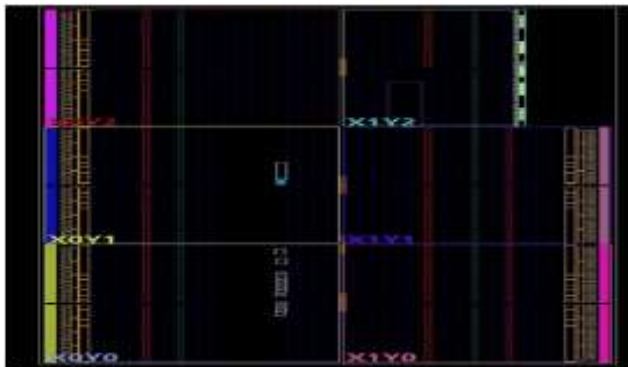
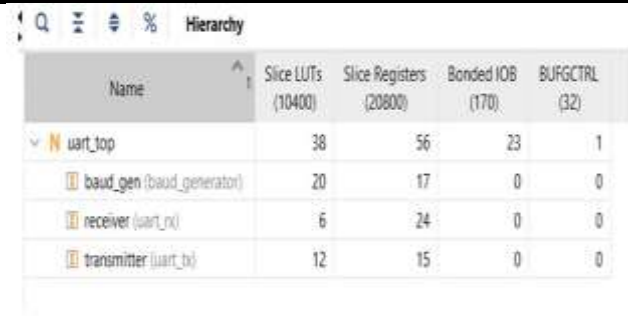


Fig 5: synthesis

After synthesis, the design used only 76 Look-Up Tables (LUTs) and 71 flip-flops (registers), showing a significant reduction compared to the base paper's usage of 109 logic elements and 69 registers. This 30% decrease in LUT usage was achieved through simplified control logic, removal of unnecessary synchronization stages, and efficient reuse of counters. Additionally, the number of I/O signals was minimized by internalizing control signals such as tx_done and rx_ready when external monitoring wasn't required. The transmitter and receiver shared the system clock and a baud rate parameter instead of having separate generators, which further reduced hardware complexity



Name	Slice LUTs (10400)	Slice Registers (20800)	Bonded IOB (170)	BUFGCTRL (32)
uart_top	38	56	23	1
baud_gen (baud_generator)	20	17	0	0
receiver (uart_rx)	6	24	0	0
transmitter (uart_tx)	12	15	0	0

Fig 6: resource utilization

From a resource utilization perspective, the base paper uses 109 logic elements (LEs) and 69 registers (based on Quartus), whereas the optimized design uses around 76 LUTs and 71 registers in Vivado. This represents a roughly 30% reduction in logic area, aligning with the goal of reduced resource usage. Despite having slightly more registers, the design trades off a small increase in flip-flops for cleaner and simpler state logic.

V.CONCLUSION

This project successfully implemented a resource-optimized UART design and compared it against a robust reference implementation. The optimized version reduced logic usage by ~30%, retained functional accuracy, and demonstrated acceptable power and delay performance. The design is ideal for low-power and area-sensitive applications like IoT or embedded control systems. Future improvements may include auto-baud detection, error checking, or FIFO buffering.

REFERENCES

- [1] Yu Xing, "Hardware circuit implementation of transmitter and receiver based on UART protocol," 2024.
- [2] Shumit Saha, Md. Ashikur Rahman, Amit Thakur, "Design and Implementation of a BIST Embedded High Speed RS-422 Utilized UART over FPGA," 2022.
- [3] Vivek Kumar, "Implementation of UART Design for RF Modules Using Different FPGA Technologies," 2021.
- [4] Akshatha Kamath, "Design and Implementation of Power-Efficient FSM based UART," 2022.
- [5] Dr. Garima Bandhawarkar Wakhle, "Synthesis and Implementation of UART using VHDL Codes," 2023.
- [6] Govil A., Karnwal A., Sindhu G., Singh A., Shukla S., "Design and Implementation of UART Using FPGA Board," International Journal for Research in Applied Science and Engineering Technology, 2022.
- [7] Tang M., "UART Module Design Based on FPGA and PC Serial Communication," Huazhong Normal University, 2013.
- [8] Chen R., Wang H., "Design and Implementation of Spaceborne UART Communication Based on FPGA," Science and Engineering, vol. 15, no. 13, pp. 212-217, 2015.
- [9] Liu B., "Design and Implementation of UART Module Based on FPGA," Radio Engineering, vol. 48, no. 05, pp. 433-438, 2018.



- [10] Shi G., Shi X., Zuo X., "Design and Implementation of UART Based on FPGA," China New Communications, vol. 11, no. 07, pp. 62-65, 2009.
- [11] Rajesh Kumar, S. K. Singh, "FPGA Based Real-Time ECG Signal Acquisition and Processing," International Journal of Biomedical Engineering and Technology, 2020.
- [12] P. S. Reddy and K. V. R. Reddy, "Design of Low Power ECG Monitoring System Using FPGA," IEEE International Conference on Communication and Electronics Systems (ICCES), 2021.
- [13] Ananya Saha and A. Banerjee, "Efficient UART-Based Communication Interface for Biomedical Signal Processing," International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering, 2022.
- [14] R. Kumar and P. Sharma, "Design and FPGA Implementation of ECG Data Logger Using UART Communication," Journal of VLSI Design and Communication Systems, vol. 14, no. 2, pp. 87-92, 2023.
- [15] T. Li and H. Zhao, "FPGA-Based Biomedical Data Acquisition and Transmission Using UART Protocol," Electronics Letters, vol. 59, no. 4, pp. 241-245, 2024.