Research of Improving Universal Asynchronous Receiver/Transmitter Speed

Yuanjin Li

Faculty of Science and Engineering, University of Nottingham Ningbo, Ningbo, 315000, China
ssyyl57@nottingham.edu.cn

Abstract. Due to the widespread implementation of UART (Universal Asynchronous Receiver-Transmitter), the data transmission speed has become a significant challenge, especially in applications requiring high performance. Improving the speed of UART communication is crucial to meet the growing demands of modern systems. This paper explores two main approaches to increasing the speed: protocol optimization and hardware implementation. The first approach, protocol optimization, involves examining and adopting advanced protocols to enhance efficiency. For instance, the 9-bit UART protocol can handle more data per frame, and the UVM (Universal Verification Methodology) environment protocol ensures robust communication with reduced overhead. These protocols aim to streamline data exchange and minimize latency. The second approach focuses on hardware implementation to boost performance. Techniques like utilizing DMA (Direct Memory Access) reduce CPU intervention during data transfer, while FSM (Finite State Machine) allows for efficient state-based control, improving overall processing speed. By combining these two approaches, it is possible to significantly enhance UART communication speed, making it suitable for high-speed applications.

Keywords: Universal Asynchronous Receiver/Transmitter, Finite State Machine, Direct Memory Access.

1 Introduction

Universal Asynchronous Receiver/Transmitter (UART) is a fundamental and widely adopted serial communication interface. At its core, UART converts parallel data, commonly used within microcontrollers and other digital devices, into serial data. This serial data is then transmitted over a communication line. This conversion allows for data transfer using just a few wires, typically a transmit (TX) line and a receive (RX) line, with the reverse conversion occurring at the receiving end. Emerging in the early days of digital electronics, UART has a long - standing history and was developed to enable simple and cost - effective data transfer between different devices.

J. Iqbal (ed.), *Proceedings of the 2025 2nd International Conference on Mechanics, Electronics Engineering and Automation (ICMEEA 2025)*, Advances in Engineering Research 272,
https://doi.org/10.2991/978-94-6463-821-9_42

As technology advances, the demand for UART has grown exponentially. In modern embedded systems, UART is frequently used to connect microcontrollers with an array of peripherals such as sensors, displays, and communication modules. For instance, in Internet of Things (IoT) devices, UART enables the connection of sensors that collect environmental data to the main processing unit. This facilitates seamless data collection and transmission [1]. In industrial control systems, it is used for communication between programmable logic controllers (PLCs) and other devices, ensuring efficient industrial processes [2].

However, the existing UART communication speed has become a bottleneck in many applications. With the increasing volume of data that needs to be transferred in real - time, such as high - resolution sensor data in IoT scenarios or rapid - response control commands in industrial systems, there is an urgent need to enhance the UART communication rate. Improving the UART communication speed can enhance system performance, reduce data transfer latency, and enable more complex and data-intensive applications. This paper aims to explore ways to boost the communication rate of UART to meet the growing demands of modern communication - intensive applications.

2 Improving Universal Asynchronous Receiver/Transmitter Speed

2.1 UART protocol optimization

9 bit UART. The UART transmission end has an address signal and a transmission signal. It uses an extra parity bit 1/0 to detect whether it is an address signal, or a transmission signal itself. If it is an address signal, for the slave end, it only needs to judge the extra parity bit. For the master end, the signal does not need to be judged. If it is a transmission signal, the master and slave end only need to judge this 1 bit. The 9bit UART is consist of 4 bits, start bit, data bits, parity bit and stop bit. Start Bit is a single low-level bit (0) that indicates the beginning of a data frame. Data Bits consists of 9 bits that carry the actual data. The extra bit can be used for different functions, such as extending the data range or parity checking. Parity Bit can be used to check for errors in the data transmission. It can be set to odd parity, even parity, or no parity. Stop Bit is an end signal of 1 or more high-level bits (1) that mark the end of a data frame [3]. The below Fig. 1 show the 9 bits UART.

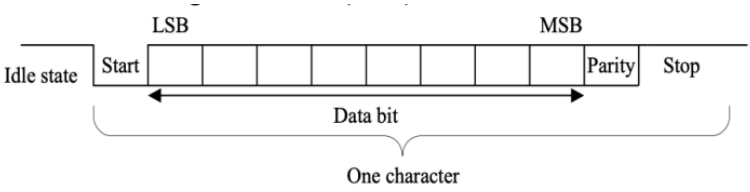

Fig. 1. The 9 bits UART [3]

Shared channel bidirectional serial communication. For multiple master-slave circuits, such as a system with 256 slave terminals but all performing the same function, relying on the ordinary UART communication protocol will require 512 lines for two-way communication [4]. The result will be very complicated and cannot handle the problems of more slave terminals. The innovative point is that due to repeated functions, the communication protocol can be simplified and divided into only the transmitter and the receiver, so that multiple slaves share a channel [4]. Since each slave port has a different address, detailed data of each port can be obtained when receiving the response, which is equivalent to all slaves receiving signals from the master, but only the slave with a matching address signal will execute the instruction.

UVM environment protocol. Firstly, it innovatively uses Verilog HDL for UART design and conducts verification in the UVM environment with System Verilog code. This not only streamlines the verification process but also takes full advantage of UVM's reusability, scalability, and interoperability, greatly reducing verification time [5]. Second, it offers a comprehensive understanding of the UART protocol. It details the asynchronous communication mechanism, frame structure, and communication modes, and expounds on the advantages of the protocol, such as simple wiring and error - detection capabilities. Third, it meticulously analyses the functions of various UART registers, clarifying their bit - widths and access types, which is crucial for understanding UART operation. Finally, it validates the UART protocol through multiple test cases like Full Duplex, Half Duplex, Parity Error, and over - run tests. The presented simulated waveforms and UVM reports prove the successful verification of the UART's functionality, ensuring its reliable communication performance.

Low Power UART based on Asynchronous Techniques. The UART is a communication protocol for serial data transfer between peripherals and embedded systems. The authors explore the implementation of the UART protocol using asynchronous design methods, aiming to reduce power consumption. First, the background knowledge including relative timing and asynchronous CAD tool flow is introduced [6]. Then, the synchronous UART implementation with its baud generator, transmitter, and receiver blocks is described. Next, the asynchronous UART design is presented. It has a similar datapath to the synchronous one but uses asynchronous handshaking controls. For example, the baud generator can be paused during idle time, and the transmitter and receiver blocks operate based on asynchronous handshaking networks. The experimental results show that the asynchronous UART design has a significant power advantage. When compared with synchronous designs using different clock-gating methods, the asynchronous UART design demonstrates a significant power reduction. In standard operating modes, especially with 90% idle time, it consumes only about one-fourth of the power. In conclusion, the asynchronous UART design is more suitable for slow peripherals that are idle most of the time, providing an effective solution for low - power UART design [6].

2.2 UART Design and Verification in hardware

Based serial communication. By summarizing different researchers' work, such as UART with BIST capability, multi - channel UART controllers, DMA - based UART

IP soft cores, and high - speed UART designs, it provides a broad perspective on the evolution and diversity of UART design approaches [7]. Through this review, this helps researchers and engineers understand the advantages and limitations of different methods. The paper points out the characteristics of different UART designs. For example, the paper highlights several key aspects: the importance of testability in UART with BIST, the reconfigurability of multi-channel UART controllers, and the performance improvements of DMA-based UART IP soft cores. It also discusses the trade - offs between using on - chip UART IP hard cores (high performance but low flexibility) and FPGA - based soft cores (high flexibility but different performance characteristics), which is valuable for making informed design decisions. Based on the reviewed work, the authors propose an application design using UART. They plan to implement serial communication between a PC hyper - terminal and an LCD display through a UART controller, with the hardware implementation in VHDL [7]. This design aims to leverage UART's advantages, such as flexibility and cost-effectiveness, showcasing its practical applications in electronic systems, particularly in optimizing SOC communication efficiency.

DMA. This part centers on the UART data reception problem when using DMA. Specifically, it addresses the limitations of fixed-length data transmission. Its primary academic contributions lie in. A novel method combining external interrupts and timer matching is designed. By connecting a GPIO pin to the UART RXD pin, an external interrupt is generated at the falling edge of the start bit [8]. The timer, set in "Retrigger" mode, clears its accumulated time when receiving an EVENT related to the external interrupt. When the timer times out, it indicates the end of data reception, solving the fixed - length data transmission limitation of DMA. Utilizing the Atmel ATSAME54N20A chip, the authors meticulously configure the hardware, including connecting UART pins to the RS - 485 chip and external interrupt pins [8]. In software, they initialize the crystal oscillator, clock, UART, DMA channel, external interrupt, and timer. This comprehensive hardware - software design ensures the smooth operation of the system. The performance of the proposed method is thoroughly analyzed through waveform studies. A 5 Mbps UART communication test environment is established. The test results, including the correct reception of data and the expected behavior of the timeout interrupt, verify the effectiveness of the scheme (Fig. 2 and Fig. 3).

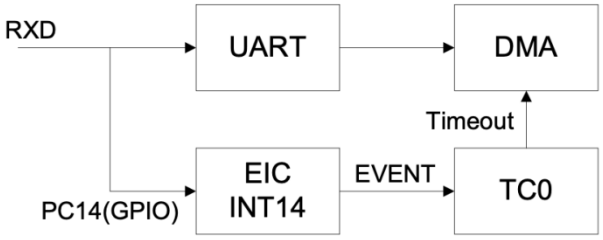

Fig. 2. DMA Design [8].

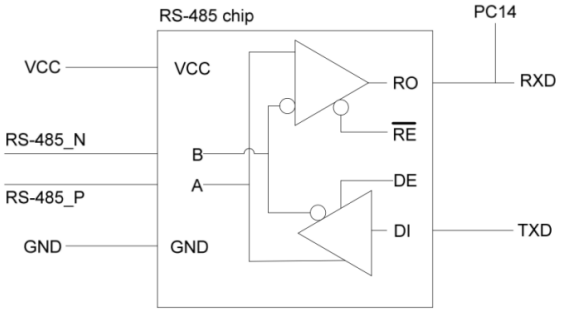

Fig. 3. RS485 chip [8].

FSM implement. In UART communication, FSM can be used to parse data frames to ensure the integrity and correctness of the data. FSM can be used at the transmitter and receiver. After receiving the complete UART signal, it can be checked through CRC to verify the integrity of the data. FSM, therefore, is suitable for short-distance signal transmission. This paper focuses on the implementation of a reduced FSM - based UART design for RF modules using different FPGA technologies. The core academic contributions are as follows [9]. First, a UART integrated circuit with receiver, transmitter, and clock divider circuitry for baud rate generation is designed. The receiver and transmitter are modelled with FSMs, which are crucial for stable operation. Second, the design is implemented on various FPGAs from Intel and Xilinx, and a comprehensive comparison is made in terms of maximum achievable speed, corresponding power consumption, and area. It is found that Xilinx Virtex 4 FPGA has the highest maximum frequency, while Intel MAX V has the lowest maximum power consumption [9]. These results provide valuable insights for choosing FPGA devices in UART - based RF module designs, guiding future research on optimizing power, speed and area trade - offs in such designs.

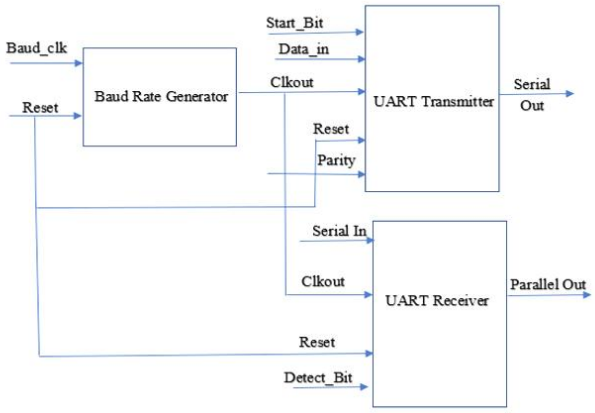

Fig. 4. The basic design of UART [10]

UART is widely used for serial data communication in digital circuits due to its high reliability. However, previous UART designs lacked the analysis of power and delay, which are crucial for low - power applications. In this study, the UART design consists of a Baud Rate Generator (BRG), transmitter, and receiver module [10]. The BRG determines the communication speed, and the FSM is employed to implement data conversion in the transmitter and receiver. The design was simulated using Cadence NCSIM and synthesized with Cadence RTL Compiler, with 45 nm and 90 nm General Process Design Kit (GPDK) library files [10]. The results show that as the clock period increases, the total power and the Power Delay Product (PDP) decrease. The 45 nm technology exhibits better performance in terms of power consumption compared to the 90 nm technology. Moreover, the proposed design shows a significant improvement in power consumption compared to existing designs. Overall, this research contributes to the development of efficient UART designs, meeting the demands of modern low-power applications (Fig. 4).

3 Conclusion

This paper focuses on enhancing the communication rate of UART and conducts in-depth research on protocol optimization and hardware implementation. In terms of protocol optimization, several innovative protocols have been proposed. The 9-bit UART adds a parity bit to distinguish signals, enhance functionality, and improve accuracy. The shared-channel bidirectional serial communication simplifies the protocol for multi-master/slave circuits, enhancing scalability. The UVM environment protocol streamlines the verification process and ensures reliable communication. The asynchronous - based low - power UART significantly reduces power consumption, being suitable for low - power applications.

Regarding hardware implementation, a review of different UART designs provides references for design decisions, and an application design is proposed. The method combines external interrupts and timer matching to overcome the limitations of DMA. This approach is particularly effective in UART data reception. The FSM - based UART design effectively parses data frames, and the comparison of different FPGA implementations offers guidance for device selection. The UART design considering power and delay demonstrates superior low-power efficiency.

In conclusion, the methods proposed in this paper have achieved good results. For future research, it is advisable to explore the integration of UART with emerging wireless communication technologies, such as 5G or 6G, to expand its application scenarios in high - speed and long - distance communication. Additionally, further optimization of UART in terms of security, especially in the context of an increasing number of security - sensitive applications, is essential. Research on more efficient error - correction algorithms and encryption mechanisms for UART communication can enhance data security. Finally, investigating the performance of UART in edge - computing environments, where real - time data processing and low - latency communication is crucial, may open up new application areas for UART.

References

1. Daraban, M., Corches, C., Taut, A., Chindris, G.: 'Protocol over UART for Real-Time Applications'. Proc. Int. Conf. System Integration Technology Management Electronics, Brasov, Romania, pp.1-6 (2021).
2. Ścisło, Ł.: 'Controller–sensing element communication using UDP protocol', Tech. Trans., Automatyka Zeszyt, **1-AC**(2), pp. 67-77 (2013).
3. Mahat, N.F.: 'Design of a 9-bit UART module based on Verilog HDL'. Proc. Int. Conf. Semiconductor Electronics, Kuala Lumpur, Malaysia, September pp.570-573 (2012).
4. Rodrigues, A.Z., Moreno, R.L., Crepaldi, P.C., Pimenta, T.C.: 'Implementation of a bidirecional serial communication protocol using shared channel'. Proc. Int. Conf. Microelectronics, Beirut, Lebanon, pp. 1-4 (2017).
5. Kashyap, B., Ravi, V.: 'Universal Verification Methodology Based Verification of UART Protocol', J. Phys. Conf. Ser., **1716**(1), 012040 (2020).
6. Bhadra, D., Vij, V.S., Stevens, K.S.: 'A low power UART design based on asynchronous techniques', IEEE Trans. Very Large Scale Integr. Syst., **21**(6), 1174-1180 (2013).
7. Laddha, N.R., Thakare, A.P.: 'A review on serial communication by UART', Int. J. Adv. Res. Comput. Sci. Softw. Eng., **3**(1), 24-28 (2013).
8. Zang, F., Niu, H.: 'Design and Implementation of High Speed UART Based on DMA'. Proc. Int. Conf. Electrical, Automation and Mechanical Engineering, Xi'an, China, pp.128-131 (2018).
9. Kumar, V., Rastogi, A., Tomar, V.K.: 'Implementation of UART Design for RF Modules Using Different FPGA Technologies', IOP Conf. Ser.: Mater. Sci. Eng., (2021).
10. Kamath, A., Mendez, T., Ramya, S., Nayak, S.G.: 'Design and Implementation of Power-Efficient FSM based UART', J. Phys. Conf. Ser., **2161**(1), 012052 (2022).