Analysis of the baud rate of the UART to affects the data

Yichen Huang [1, a, †], Hao Lin [2, b, †], Di Zhang [3, *, †]

[1] School of Physics and Electronic Information Anhui Normal University, Wuhu, 241000, Country

[2] College of Electronic and Optical Engineering &College of Microelectronics Nanjing University of Posts and Telecommunications, Nanjing, 210023, China

[3] College of Electronic Science & Engineering Jilin University, Changchun, 130021, China

* Corresponding Author Email: zhangdi1919@mails.jlu.edu.cn, [a]2467865795@qq.com, [b]2274427443@qq.com

[†]These authors contributed equally

Abstract. This article explores the impact of different baud rates on Universal Asynchronous Receiver/Transmitter (UART) asynchronous serial communication. It discusses the frame format and calculation method of baud rate error, as well as the main reasons for error, such as clock error, sampling error, line noise, data bit error, and software delay. To ensure reliable communication, it is essential to test and debug the actual baud rate and adjust it accordingly. The study found that higher expected baud rates resulted in higher baud rate errors, affecting transmission distance, speed, reliability, and system complexity. The appropriate baud rate should be selected based on specific application scenarios and requirements. Higher baud rates are needed for high-speed transmissions, while lower baud rates are required for long transmission distances or high reliability requirements. However, further research is needed to determine the impact of unallocated baud rates on data. Overall, this article provides valuable insights into the development of UART.

Keywords: UART communication, frame format, baud rate, transmission distance, transmission speed.

1. Introduction

UART (Universal Asynchronous Receiver/Transmitter), that is, a universal transceiver, is a commonly used integrated serial bus in microprocessors, and it is also the most widely used full-duplex serial data communication interface [1, 2]. UART is widely used in serial communication, data acquisition and processing, debugging, and troubleshooting. The main function of UART is to convert parallel data into serial data and send it on the communication line, while being able to accept serial data and convert it to parallel data for system use [3]. The baud rate, as an indicator of the rate of serial communication, has an important impact on the waveform of UART. In the case of higher baud rates, on the one hand, since the recognition of data bits is determined by clock cycles, if the clock cycles are too short, data bits may not be recognized, resulting in errors. On the other hand, data at high baud rates change quickly and are susceptible to noise and crosstalk, which can deform the signal, cause the receiver to not correctly identify the data bits, and even cause data transmission failure. In this paper, the related principles of UART and the calculation and research methods of baud rate are introduced, and then the influence of different baud rates on the same waveform is analyzed by ModelSim, and the waveform graph is obtained and the data is calculated to analyze the error and its causes. And further optimization, draw relevant conclusions, aiming to provide certain reference significance for the development of UART.

2. Theoretical basis

2.1. Structure of the UART

The UART is asynchronous serial communication, comparing with parallel communication, though the serial communication has a low transfer efficiency, but it costs not very high and supports long-

distance transmission; comparing with synchronous serial communication, a synchronous serial communication does not require clock synchronization between the sending and receiving sides [4]. A synchronous serial communication frame format shown in the following figure 1.

Figure 1. A synchronous serial communication frame format

In asynchronous communication, the first bit of the data frame is the start bit, and it is in the logical "1" state when there is no data transmission on the communication line; If you are ready to send data, you first send a logical "0", that is, tell the receiver to start sending data, and this logical "0" is the start bit. When the receiver detects this low level, it is ready to receive data. After the start bit is followed by the data bit, generally there are 8 data bits by default, when sending data, the low bit is first, the high bit is behind, and it is sent bit by bit. The check digit is used to verify whether there is an error in the transmission of data, this check digit is one of the ways of limiting error detection agreed in advance by the sending and receiving parties, or it cannot use the check digit. The last bit of the data frame format stops the bit, the logical "1" is valid, and it occupies are 1/2 bits, 1 bit, 2 bits, and 2 bits used in this article. The stop bit is a sign of the end of transmitting a frame of data, and prepares for the next set of data to be passed later. The idle bit refers to the state when there is no data transmission on the communication line, that is, the idle state, the logic "1" is valid, the circuit is in the waiting state at this time, if the jump becomes a logical "0" means that there is new data about to be transmitted, the logical "0" is the start bit. In this loop, various data are transmitted on the communication line [5, 6].

2.2. The calculation method of the band rate

UART transmits data based on the selected baud rate, so an accurate baud rate is critical to the success of the data transfer. The baud rate error refers to the difference between the actual baud rate and the desired baud rate, usually expressed as a percentage or partial thousandth (PPTH). Here is the general formula for calculating the UART baud rate error:

In the case that the stop bit sampling error is allowed, the data bit is eight bits, the parity bit is not set, and the tolerance margin is 10%, the error in the transmission bit is calculated, that is, when the last stop bit is just 90%.

The bit period of a Tx device is defined as follows in formula (1) [7].

$$\Delta Trx = Ttx(\frac{D}{100}) \tag{1}$$

Where D means received baud rate, Ttx means transmit baud rate, D means the percentage difference between the received baud rate and the sent baud rate.

The calculation process is simplified from the starting bit sampling to 50% of the initial work, and the last data bit 8ΔTrx turns off RX as the sampling time. Therefore, we arrive at the calculation formula if it is accurate enough:8ΔTrx<0.4Ttx, D<5

It is concluded that the communication can be considered valid when the baud rate error between the sending end and the receiving end is less than 5%.

2.3. Baud rate deviation adjustment

The actual baud rate of UART may produce errors in testing, so adequate testing and debugging is required in practical applications to ensure reliable and stable communication. The main reasons for the error are as follows:

- Clock error: UART uses clock signals to send and receive data bits at regular intervals. If the frequency of the clock signal is inaccurate, it can cause an error between the actual baud rate and the expected baud rate. Clock errors can be caused by factors such as crystal oscillators, temperature, voltage changes, and more[8].

- Sampling error: UART needs to sample the data when receiving it to determine the state of the data bits. If the sampling timing is inaccurate, it will result in the wrong bit of data received. Sampling errors can be caused by the UART receive clock being out of sync with the transmit clock, or by noise interference[7].
- Line noise: UART communication uses serial communication, and data is transmitted through a single line. If there is noise interference in the line, it will cause the received data bit to be wrong, resulting in an error between the actual baud rate and the expected baud rate[1].
- Data bit error: The data bits in UART communication can be configured as 5, 6, 7 or 8 bits. If the data bit settings of the receiver and sender are inconsistent, it will cause the received data bit error, resulting in an error between the actual baud rate and the expected baud rate[1].
- Software delay: In UART communication, the CPU needs to perform certain processing after sending or receiving data. If the CPU processing time is insufficient or too long, it will cause an error between the actual baud rate and the expected baud rate[1].

In summary, the actual baud rate error of UART may be caused by a variety of factors and needs to be analyzed and solved according to the specific situation. This paper compares the actual baud rate with the set theoretical baud rate, if there is a large deviation between the two, it needs to be adjusted, and the adjustment principle is as follows:

- Expected baud rate refers to the average baud rate expected during communication. The expected baud rate can be calculated according to the following formula (2) [7].

Expected baud rate = (*data bit* + *stop bit* + *check digit*)

$$* \textit{ transfer rate} \tag{2}$$

- To correctly calculate the actual baud rate, it is necessary to measure the time when the UART is sent and received and compare it with the theoretical baud rate. The steps to calculate the actual baud rate are: set the theoretical baud rate through the registers of the UART controller, generally using standard UART baud rate settings. Send a piece of data of a known length, then start a timer to record when data is sent. Receives data sent by UART and records the time it takes to receive the data.

To calculate the actual baud rate, you can use the following formula (3):

Actual baud rate = *length of data sent* /

$$\textit{time required to send data} \tag{3}$$

3. Baud rate simulation design and result analysis

3.1. Introduction to the ModelSim platform

ModelSim is a simulation software for HDL language developed by Mentor, which provides a friendly simulation environment for those who need it, and is currently the only simulation software that supports single-core VHDL and Verilog hybrid simulation. It not only compiles simulations quickly, but also supports cross-platform simulation, and supports system and description languages such as System Verilog, System C, PSL. And because its compilation code is platform-independent, it is more convenient to protect IP cores [9]. It also features a personalized graphical and user interface that provides a powerful means for users to speed up error tuning, making it the simulation software of choice for FPGA/ASIC designs [10]. In this paper, the Verilog language is used to program and debug the transmitting module and receiving module of UART on this platform, and the waveform results of the experiment are obtained.

3.2. Code flow

This design requires that the UART that transmits data receives data from the data bus and transmits it to the receiving data bus which is shown in the following figure 2. This design takes the system clock to 50MHZ and calculates the clock cycle from the system clock and the set baud rate variable. In the transmit module: we first define the relevant signal, then collect the data and use the transmission process starting with low bytes. Falling edge detection is performed in the receiver module and the two-bit start and two-bit stop positions are detected. If the validation succeeds and continues running, otherwise re-receive the data and verify. After successful verification, the data is received into registers and latched, and finally the data is tested and a waveform graph is generated.

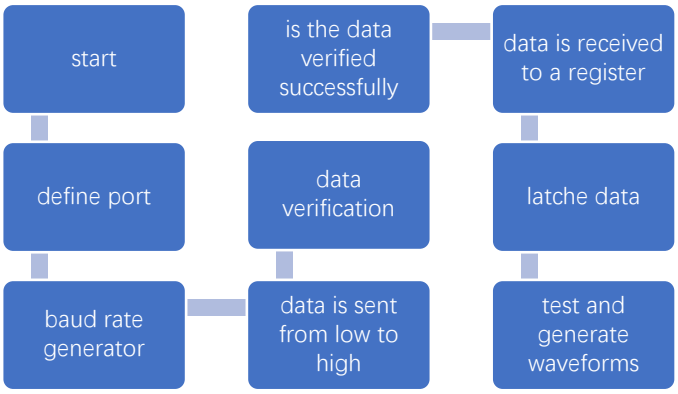

Figure 2. Experimental code flowchart

3.3. Simulation results

The expected baud rate is shown in the figure 3, 4, 5 below.

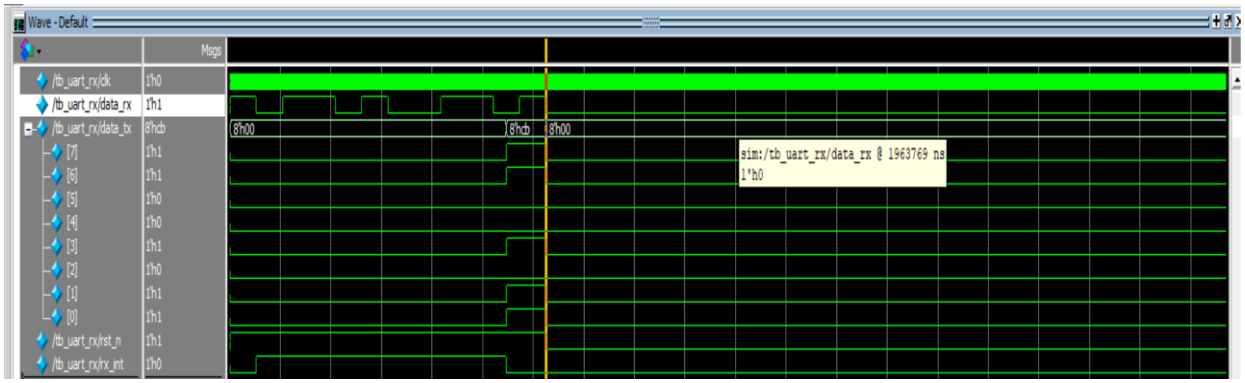

Figure 3. Experimental simulation results 1

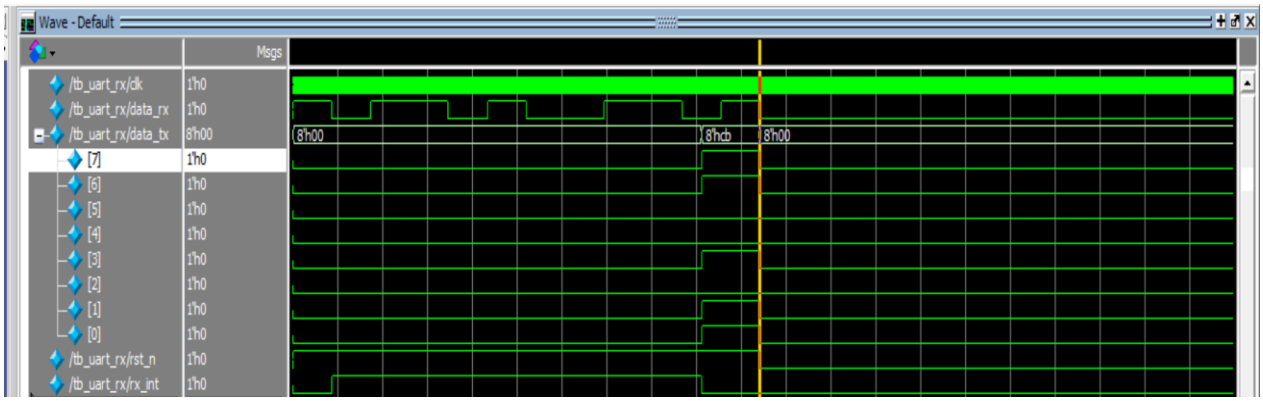

Figure 4. Experimental simulation results 2

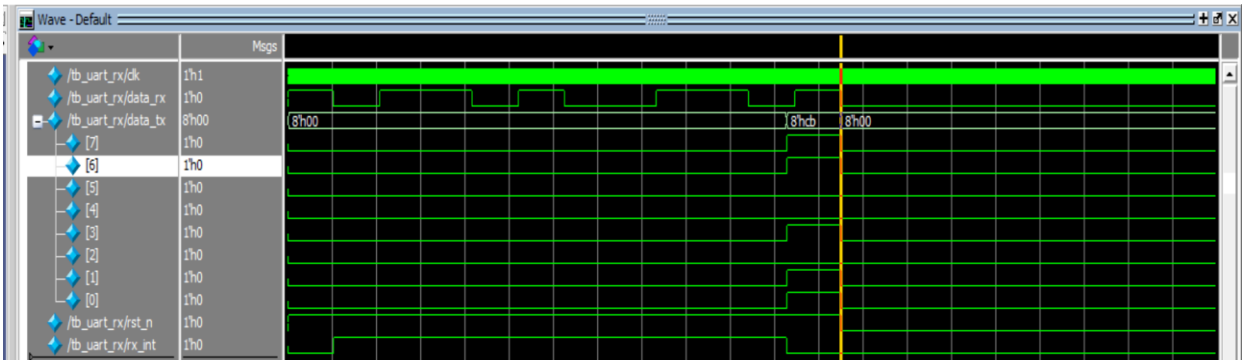

Figure 5. Experimental simulation results 3

The figure 3 is a waveform plot with an ideal baud rate of 9600. The waveform plot shows that the time required to receive data is 1249380ns. The sent data is 12 bits, the input is a binary number 11010011, the output is a binary number 11001011, the start flag is the binary number 10, and the end flag is the binary number 01.

The figure 4 shows the waveform with an ideal baud rate of 115200. The time required to receive data from the waveform plot is 1041740ns. The sent data is 12 bits, the input is a binary number 11010011, the output is a binary number 11001011, the start flag is the binary number 10, and the end flag is the binary number 01.

Figure 5 is a waveform plot with an ideal baud rate of 19200. The waveform plot shows that the time required to receive data is 625016ns. The sent data is 12 bits, the input is a binary number 11010011, the output is a binary number 11001011, the start flag is the binary number 10, and the end flag is the binary number 01.

3.4. Analysis of the results

The baud rate error in Figure 1 is 16.625%, and the baud rate error in Figure 2 is 16.669%, Figure 3 The baud rate error is 16.672%.

From the simulation results, it can be obtained that as the expected baud rate increases, the baud rate error increases. The influence of different baud rates on UART is mainly manifested in the following aspects:

- Transmission distance: As the baud rate increases, the transmission distance decreases. This is because high-speed transmission requires higher voltages, and high-voltage signals are prone to distortion and interference, which reduces the transmission distance.
- Transfer speed: The higher the baud rate, the faster the transmission speed. Of course, the transmission speed is also affected by factors such as serial port bandwidth and system processing speed.
- Reliability: When the baud rate is low, the transmitted signal is more stable and reliable, and the probability of errors is smaller. However, if the baud rate is too low, it will not meet the actual demand.
- System complexity: High baud rates require higher processing speeds and more complex circuit designs, so the complexity of the system will increase accordingly.

In summary, the selection of the appropriate baud rate needs to be determined according to the specific application scenarios and requirements. In general, high-speed transmission requires a higher baud rate, while applications with long transmission distances or high reliability requirements require a lower baud rate. In this paper, the baud rate of this experiment is not assigned enough times, so only an approximate range of its impact on the data can be obtained. Only 11001011 data were used, and no other data was used to verify that the results were generalizable. Further research is needed on whether the above analysis results have an impact on the conclusions of the simulation plot by the code used in this paper.

4. Conclcusion

This article summarizes the structure and points of UART, and further explains the baud rate research principle that this article focuses on. After determining the algorithm and the experimental process, set different baud rates to simulate on the ModelSim platform. After obtaining the waveform diagram, the error is calculated and analyzed and compared. As the baud rate increases, the error becomes larger and the baud rate error is an important influence on data transmission. Improving the influence of baud rate error on UART and reducing UART error are very important for UART. This paper provides some help for the study of the tolerance range of UART communication baud rate error. Higher rate and wider application range: The data transmission rate of UART is usually between tens of Kbps and hundreds of Kbps, and the rate of UART may be further improved in the future. With the development of the Internet of Things, communication between devices needs better security and reliability. Future UART may add security and reliability features such as data encryption, error detection, and correction.

References

[1] Y. Fang and X. Chen, "Design and Simulation of UART Serial Communication Module Based on VHDL," *2011 3rd International Workshop on Intelligent Systems and Applications*, May 2011.

[2] E. G. Rajgure and A. P. Thakare, "'The Novel Technique for Channel Security using UART,'" *2014 International Conference on Green Computing Communication and Electrical Engineering (ICGCCEE)*, Mar. 2014.

[3] C. Zheng and Y. Liu, "Microcontroller principle and engineering applications - from MCS-51 to ARM," Wuhan: Huazhong University of Science and Technology Press, pp. 153 – 155, 2020.

[4] S. K. Ibrahim, H. M. Hathal, and R. A. Abdul Hussein, "Baud rate variations effect on virtual channel based on PIC microcontroller," *TELKOMNIKA (Telecommunication Computing Electronics and Control)*, vol. 17, no. 6, p. 2755, Dec. 2019.

[5] D. V. Gadre and S. Gupta, "Universal Asynchronous Receiver and Transmitter (UART)," Getting Started with Tiva ARM Cortex M4 Microcontrollers, vol. 10.1007/978-81322–37662_12, pp. 151 – 167, Oct. 2017.

[6] N. Patel, V. Patel, and V. Patel, "VHDL Implementation of UART with Status Register," *2012 International Conference on Communication Systems and Network Technologies*, May 2012.

[7] Peña, Eric, and Mary Grace Legaspi. "Uart: A hardware communication protocol understanding universal asynchronous receiver/transmitter." Visit Analog 54.4, 2020.

[8] R. Gallo, M. Delvai, W. Elmenreich, and A. Steininger, "Revision and verification of an enhanced UART," *IEEE Xplore*, Sep. 01, 2004.

[9] J. Hua, L. Zhou, C. Chen, and L. Jiang, "Synchronization for QDPSK —Costas loop and Gardner algorithm using FPGAs," *2014 IEEE/ACIS 13th International Conference on Computer and Information Science (ICIS)*, Jun. 2014.

[10] T. Guha, I. Pal, R. Basak, F. Khan, and S. A. Khan, "Design and Implementation of HDLC Protocol using VHDL technique," *2020 4th International Conference on Electronics, Materials Engineering & Nano-Technology (IEMENTech)*, Oct. 2020.