IOSR Journal of VLSI and Signal Processing (IOSR-JVSP)
Volume 6, Issue 6, Ver. II (Nov. - Dec. 2016), PP 65-69
e-ISSN: 2319 – 4200, p-ISSN No. : 2319 – 4197
www.iosrjournals.org

Analysis of Universal Asynchronous Receiver and Transmitter for Reliable Data Transmission.

Yogeesh.K. V[1], Venkateshkumar.H[2],Rohith. S[3]
([1]ECE (PG student), Nagarjuna college of Engg. & Technology/VTU, Bengaluru India)
([2]ECE(Associative Professor), Nagarjuna college of Engg. & Technology /VTU, Bengaluru India)
([3]ECE(Assistant Professor), Nagarjuna college of Engg. & Technology /VTU, Bengaluru India)

Abstract: *The simultaneous reception and transmission of the data in parallel communication is cost effective and the complexity for the system resources increases. To overcome such drawback and to perform an effective transmission/reception, the serial communication protocols were came into existence. Universal Asynchronous Receiver and Transmitter (UART) is a kind of serial communication protocol. Transmission and reception techniques are discussed and analysed for reliable data transmission.*
Keywords: *UART, Transmitter, Receiver,ModelSim_Altera;*

I. Introduction

A universal asynchronous receiver andtransmitter (UART) is an integrated circuit, which play an important role in serial communication [1]. UART performs the conversion between serial-to parallel data in reception and parallel to serial data in transmission. UART has large internal buffer to store data coming from the input port until the CPU has time to process it. It consist a parallel-to serial converter for data transmitted from the computer and a serial-to parallel converter for data received from the computer.The signal distortion will be reduced in Serial communication, and hencedata can be transmitted over long distances. While converting data from parallel to serial for transmission, and serial to parallel on reception, a UART offers additional circuits for signals it can be used to indicate the state of the transmission media and to control the flow of data if the remote device is not ready to receive more data. UART has large internal buffer to store data coming from the input port until the CPU has time to process it.

The UART serial communication module is splits into three sub modules. The transmitter module, receiver module, and baud rate generator [2]. The baud generator is used to generate a local clock signal which is much higher than transmit and receive of UART control clock signal. The UART receiver module is used to receive the serial signals at RXD (receiver detection), and convert them into parallel data. The UART transmit module converts the bytes into serial bits according to the basic frame format and transmits those bits through TXD (transmitter detection).

II. Transmitter Module

The transmit module is used to convert 8-bit parallel data into serial data and then sends, it adds start bit,parity bit atMSB and stop bits at LSB of the data. When the UART transmit module is reset, the transmit module gets ready to send the data. The output appear as 1 start bit, 8 data bits, 1 parity bit and 1 stop bit. The parity bit is specified as the output. Finally stop bits displays logic 1.

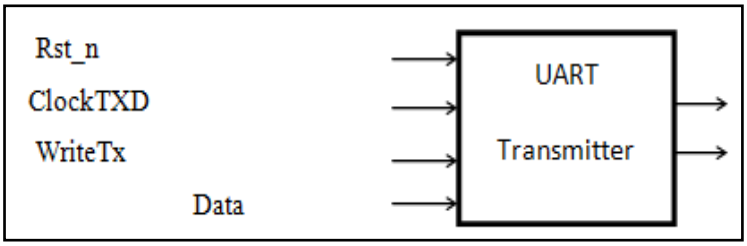

Figure 2.1 UART Transmitter Diagram

III. Receiver Module

In UART reception, serial data and receiving clock are asynchronous [3]. Therefore it is important to correctly determine the start bit of a frame data. RXD changes state, logic 0 to logic 1, it represents the beginning of a data frame. Once start bit is identified, next bit begin to count the rising edge of the baud clock, and then counting considers the sample RXD. Each value of the logic level stores in the register receiver buffer (7, 0) in order.

We can ensure that, when the count is equals 8, all the data bits are received, and converted into a byte to parallel data.The receiver module receives data from Rxd pin, Rxd jumps into logic 0 from logic 1can be regarded as the beginning of data frame. When the UART module is reset, it has been waiting the Rxd level to jump. The start bit is identified by detecting rxd level changes from high to low. In order to avoid the maladjustment[4] of start bit caused by noise, we use error detection methods. UART resynchronize on the falling edge of the start bit only, and thenread the centre of each expected data bit, and this works if the broadcast data rate is accurate for allowance of the stop bits for reliability.

Three errors [5] of signals detection are commonly used in UART

- Parity bit used by receiver of UART for check the transmission errors
- Overrun Error is used to determine whether data is overwritten than expected.
- A framing error used to determine when the chosen start and stop bits are valid not.

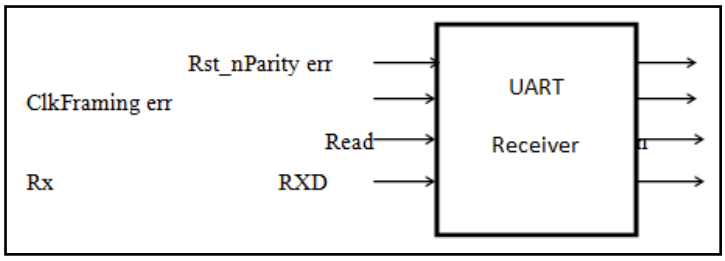

Figure 3.1 UART Receiver Diagram

IV. Baud Rate Generator

Baud rate generator is a frequency divider. The frequency clock produced by the baud rate generator is not exactlythe baud rate clock, but 16 times the baud rate clock. For sampling there exists the ideal time at the middle point of serial data bit. This output clock generated can be used as the receive reference clock by the receiver UART.

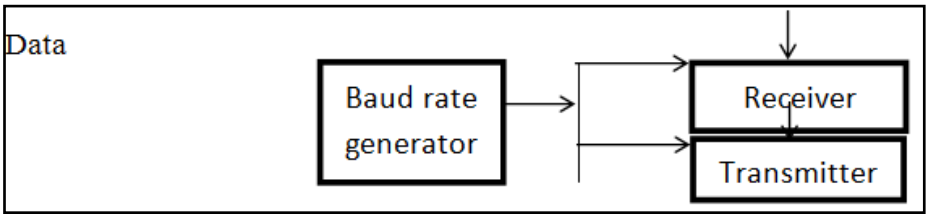

Figure 4.1 UART Module

V. UART Function

UART requires only two signal lines (RXD, TXD) for the full-duplex data communication [6]. TXD is the transmit side, it is output of UART. RXD is in the receiver side, it is used as the input of UART [7]. The two states in the signal line can be distinguished using logic 1 (high) and logic 0 (low). The start bit is used for indicates the receiver that a word of data is about tosent, and hence the clock in the receiver gets synchronized with transmitter clock.

When the transmitter is idle state, the data line in the logic high state. After the first bit, the individual bits are sent, with the least significant bit being sent first. Each bit in the transmission has to maintain accurate same amount of time and the receiver concentrates at approximately half way for determining if bit is 0 or 1. The sender does not make guarantee when the receiver expected at the value of bit.When the receiver receives all bits of data word is concentrates on parity bit and then generates the stop bit. If the stop bit does not appear then it results in framing error. This is due to the variations in clocks of receiver and transmitter module.

Irrespective of the received data the UART itself eliminates the start, parity and stop bits. If there is new word ready for transmission the system receive self- synchronised. If there is no data to transmit, the transmission line can be idle.

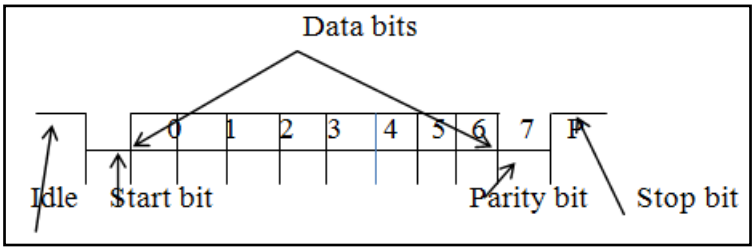

Figure 5.1 UART Frame Format

VI. Error Detection Code Using Parity Bit

In Asynchronous UART [7] Parity bit is a bit added to a string of binary code that detectswhether the number of 1-bits in the string is even or odd. Parity bits [8] are used simplest form of error detecting code. In communication and computing parity refers to evenness or oddness of the number of bits with value one with in a given set of bits, and thus determined by the value of all the bits. It can be calculated via an XOR sum of bits, yielding 0 for even parity and 1 for odd parity. This property of being upon all the bits and changing value if any bit changes and allows for its use in error detection schemes.

If an odd number of bits are transmitted incorrectly, the parity bit will be incorrect, and indicating that a parity error occurred in the transmission. The data must be discarded and re-transmitted from scratch. On a noisy transmission medium, successful transmission can therefore take a long time, or even occur, however parity has the advantage that it uses only a single bit and requires only a number of XOR gates to generate.

The detection of error code by parity bit is shown below.

Types of bit parity	Successful transmission scenario
Even parity	A needs to transmit : 10000001 A computes parity bit value: 1+0+0+0+0+0+0+1 (mod2) = 0 A adds parity bit and sends: 100000010 B receives: 100000010 B computes parity: 1+0+0+0+0+0+0+1+0 (mod 2) = 0 B reports correct transmission after observing expected even result.
Odd parity	A needs to transmit: 10000001 A computes parity bit value: 1+0+0+0+0+0+0+0+1+1 (mod 2) =1 A adds parity bit and sends 100000011 B receives : 100000011 B computes overall parity : 1+0+0+0+0+0+0+0+1+1 (mod 2) = 1 B reports correct transformation after observing odd result

This mechanism is capable of detecting the single bit errors, because if one bit gets flipped due to line noise, there will be an incorrect number of ones in the received data. The above table shows the successful scenario.

- Consider the following example with a transmission error in the second bit using XOR.

Type of bit parity error	Failed transmission scenario
Even parity	A needs to transmit : 10000001 A computes parity bit value :1^0^0^0^0^0^0^1 = 0 A adds parity bit and sends : 100000010 ……Transmit Error……. B receives : 11000001 B computes overall parity : 1^1^0^0^0^0^0^1 = 1 B compares in correct transmission after observing odd result.
Odd parity	A needs to transmit : 11000001 A computes odd parity value : 1^1^0^0^0^0^0^1 = 1 A sends : 11000001 ……...Transmit Error……. B receives : 10000001 B computes overall parity :1^0^0^0^0^0^0^1 = 0 B reports incorrect transmission after observing expected even result

VII. Results

The output snapshot shows using modelsim_altera 6.6d mentor graphics software.

Simulated Output of Transmitter

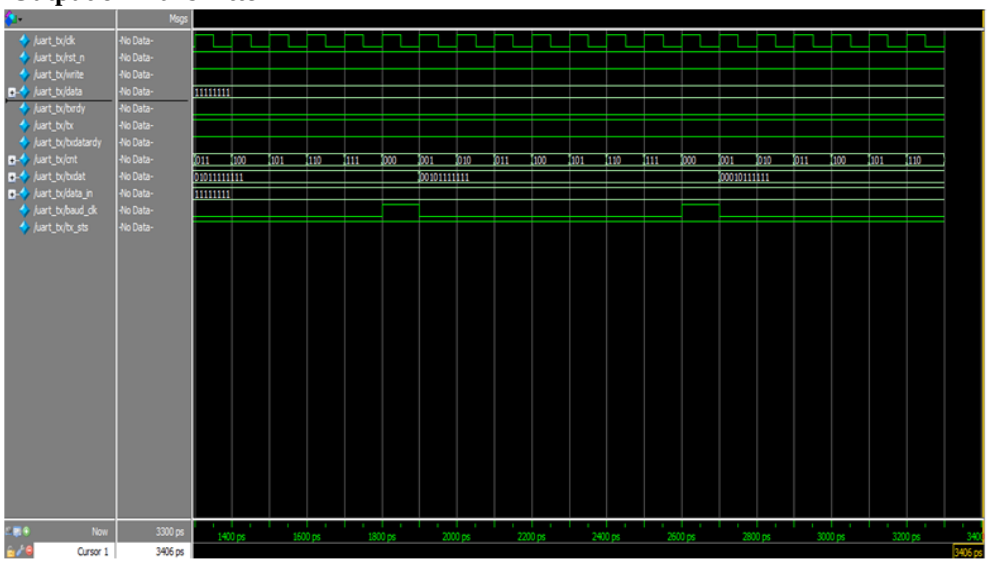

Figure 7.1 output of UART transmitter.

Above graph 7.1 shows UART Transmitter
Set clock is high.
Set reset_n and write inputs are logic 0.
In write logic 0condition, assign 8'bit of data is 11111111. When set clock and write inputs are logic high, the transmitter output status TXD is 0 and input of data transmits serially at Tx output.

Simulated Output of Receiver

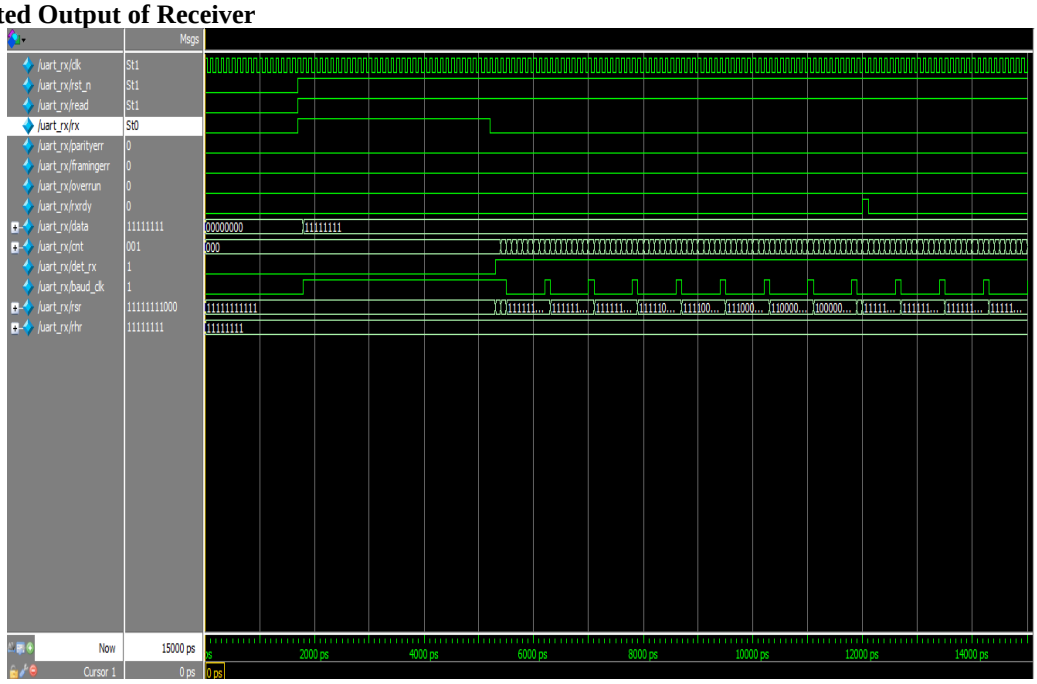

Figure 7.2 output of UART Receiver

The above figure 7.2 shows UART Receiver.
Set clock is high.
Set rst_n and Rx inputs are logic 0. Rx input receives serially transmitted data from transmitter when exactly half of the baud rate clock.
Set rst_n and Read inputs are logic 1. Read input reads the transmitted 8'bit of data 11111111 then RXDpin status becomes 1, and displays the output at Rx of receiver.

Simulated Output of UART

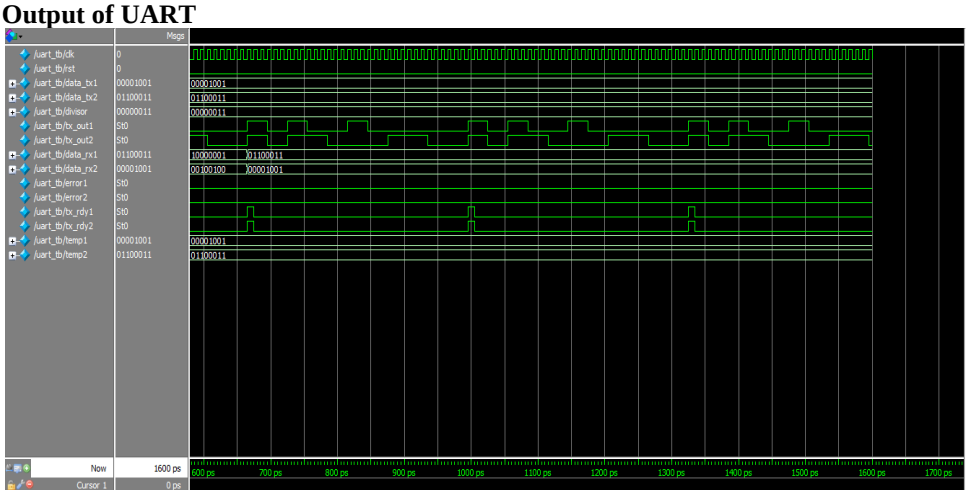

Figure 7.3 complete output of UART

The figure 7.3 shows complete transmission and reception of a UART.
Set rst_n and write inputs are logic 0. In logic condition 0 we assign 8'bit of data 00110011. TXD status pin becomes 0 it is ready to send. When set logic high rst_n and write inputs are logic 1, the 8'bit of data transmits serially at transmitter output of Tx pin.
If half of the baud rate the, Rx pin receives serially, when pin read is logic 0 it reads data and stop bits becomes 1, it displays 8'bit parallel data at receiver pin RXD.

VIII. Conclusion

The design of UART is studied in detail and simulated using modelsim_altera 6.6d mentor graphic software, and verified by transmitting and receiving data with error and without error. The results are stable and reliable shows functionality as expected. Further it can improve by sending 16 bit transmitting and receiving data obtained results can be checked whether it is stable, reliable and verified for correct functionalities.

References

[1] Hu Hua, BAI, feng-e. *Design and simulation of UART serial communication module based on Verilog –HDL [J]. J ISUANJ I YU XIANDA IHUA 2008 VOL.8 Frank Durda Serial and UARTTutorial.*

[2] R. J. Samuel, *Self-tuning baud rate generator for UART applications, USA: 7062003B2, 2006.*

[3] Amanpreet kaur, Amandeep kaur, *an approach for designing a universal asynchronous receiver transmitter "UART". International Journal of engineering research and application(IJCERA)ISSN. 2248-9962 www.ijera.com vol.2 issue 3, may-June 2012, pp-2 2305-2311.*

[4] Shamanth H. K. , Venkatesh Kumar. H "Digital Implementation Of 6-bit SAR ADC with Foreground Technology" *International Research Journal of Engineering & Technology [IRJET] volume 03, Issue 07 / July-2016 e-ISSN 2395-0056 p-ISSN 2395-0072 Nagarjuna College Of Engineering Bangaluru.*

[5] S. Muppalla and K. R. Vaddempudi, "A novel VHDL implementation of UART with single error correction and double error Detection capability," *Signal Processing And Communication Engineering Systems (SPACES), 2015 International Conference on*, Guntur 2015, pp. 152-156.

[6] G. B. Wakhle, I. Aggarwal and S. Gaba, "Synthesis and Implementation of UART Using VHDL Codes," *Computer, Consumer and Control (IS3C), 2012 International Symposium on*, Taichung, 2012, pp. 1-3.

[7] Umakanta Nanda, Sushant Kumar Pattnaik , department of electronics and communication silicon institute of technology Bhuvenshwar India"Universal*Asynchronous Receiver and Transmitter". 2013 international conference on advance and Communication systems (ICACCS-2016), Jun 22-23, 2016*CoimbatoreIndia.

[8] G. Aydos and G. Fey, "Exploiting error detection latency for parity-based soft error detection," *2016 IEEE 19th International Symposium on Design and Diagnostics of Electronic Circuits & Systems (DDECS)*, Kosice, 2016, pp. 1-6.