International Journal of Advanced Engineering and Technology
ISSN: 2456-7655; Impact Factor: RJIF 5.54
Received: 26-11-2020; Accepted: 11-12-2020; Published: 25-12-2020
www.newengineeringjournal.com
Volume 4; Issue 2; 2020; Page No. 36-38

Universal asynchronous receiver transmitter

Dinesh Sheoran
Assistant Professor, Department of ECE, MSIT, C-4, Janakpuri, New Delhi, India

Abstract
Almost all computers and microcontrollers have several serial data ports used to communicate with serial input/output devices such as keyboards and serial printers. By using a modem connected to a serial port serial data can be transmitted to and received from a remote location via telephone line. The serial communication interface, which receives and transmits the serial data is called a UART (Universal Asynchronous Receiver-Transmitter). RXD is the received serial data signal and TXD is transmitted data signal. In this project UART is implemented using verilog due to low cost, high speed, reprogram ability and fast time to market.

Keywords: UART, serial communication, baud rate, TXD, RXD, baud rate

Introduction

To meet the modern operation microcontroller and digital signal processor we need desired system performance. But in actual process, it is very difficult to attain desired result, since it depends on various factors. Communication is vital factor which affect the performance of system. A universal asynchronous receive/transmit (UART) is an integrated circuit which plays the most important role in serial communication. It handles the conversion between serial and parallel data. Serial communication reduces the distortion of a signal, therefore makes data transfer between two systems separated in great distance possible. It contains a parallel-to serial converter for data transmitted from the computer and a serial to parallel converter for data coming in via the serial line. The UART also has a buffer for temporarily storing data from high speed transmissions. In addition to the basic job of converting data from parallel to serial for transmission and from serial to parallel on reception, a UART will usually provide additional circuits for signals that can be used to indicate the state of the transmission media and to regulate the flow of data in the event that the remote device is not prepared to accept more data. UART must have a larger internal buffer to store data coming from the modem until the CPU has time to process it. The UART serial communication module is divided into three submodules: the baud rate generator, receiver module and transmitter module Therefore, the implementation of the UART communication module is actually the realization of the three sub-modules. The baud rate generator is used to produce a local clock signal which is much higher than the baud rate to control the UART receive and transmit; The UART receiver module is used to receive the serial signals at RXD, 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

Serial transmission

Serial transmission is used in transmitting a bit in UART. Serial data transmission is a form of data transmission where bits of characters are sent one at a time along a communication path. Serial data transmission travel over a single wire in one direction. Two types of serial data transmission are there:

Synchronous serial communication

Synchronous serial communication is a serial communication protocol where data is sent in a continuous stream at a constant rate. Synchronous Communication requires that the clocks in the transmitting and receiving devices are synchronized, running at the same rate so the receiver can sample the signal at the same time intervals used by transmitter. No start or stop bits are required. In synchronous communication data is not sent in individual bytes, but as frames of large data blocks as shown in Fig. 1.

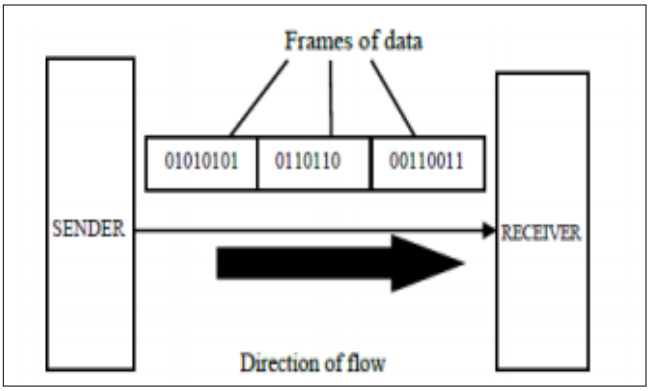

Fig 1: Synchronous Serial Transmission

Asynchronous Serial Communication

Asynchronous means "no synchronization", and thus does not require sending and receiving idle characters. However, the beginning and end of each byte of data must be identified by start and stop bits. The start bit indicates when the data byte is about to begin and the stop bit signals when it ends. The requirement to send these additional two bits cause asynchronous communications to be slightly slower than synchronous however it has the advantage that the processor does not have to deal with the additional idle characters. As shown in Fig. 2, one start bit and stop bit are added to the whole byte.

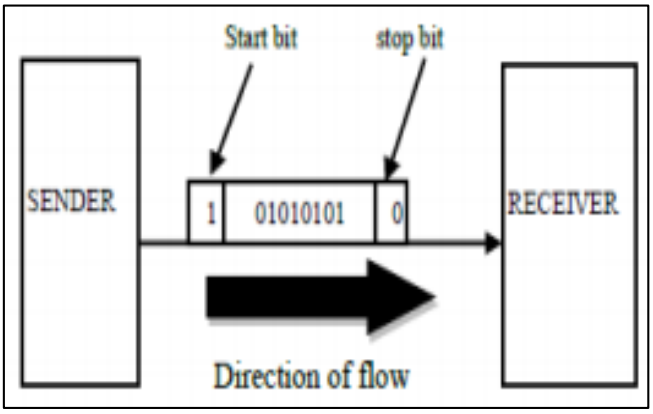

Fig 2: Asynchronous serial transmission

Just as software designers use high level languages (HLL) to express the algorithms in terms of language statements, digital hard-ware designers use hardware description languages (HDL) to describe the system they are designing. Although HDL's were originated as a medium of precise yet concise description of digital hardware, they have found a variety of applications such as generating user manuals, teaching logic design, acting as an input medium for an automatic design system, VHDL and Verilog are widely used HDL's. Hardware description languages such as Verilog are similar to software programming languages because they include ways of describing the propagation time and signal strengths (sensitivity). Verilog is a strongly and richly typed language. We are using VHDL as our programming language.

Technology used

Xilinx ISE

Xilinx ISE (integrated software environment) is a software tool produced by Xilinx for synthesis and analysis of HDL designs, which enables the developer to synthesize ("compile") their designs, perform timing analysis, examine RTL diagrams, simulate a design's reaction to different stimuli, and configure the target device with the programmer. Xilinx ISE 8.2i is a version of ISE, which we are used in our project for simulation.

There are many conventional methods used for the optimisation of our ELD problem and they are: Lambda Iteration Method, Newton's Method, Base point and Participation factor Method and Gradient Method.In this, we will be using one of these conventional methods as our base and that is the lambda iteration method.

Baud rate

BAUD rate defines how many number of bits can be transferred in 1 second e.g. BAUD 9600 means 9600 bits can be transferred in 1 sec.

Baud rate is a measure of the speed of data transfer, expressed in bits per second (bps). Both UARTs must operate at about the same baud rate. The baud rate between the transmitting and receiving UARTs can only differ by about 10% before the timing of bits gets too far off.

Both UARTs must also must be configured to transmit and receive the same data packet structure.

Operation of UART transmitter and receiver

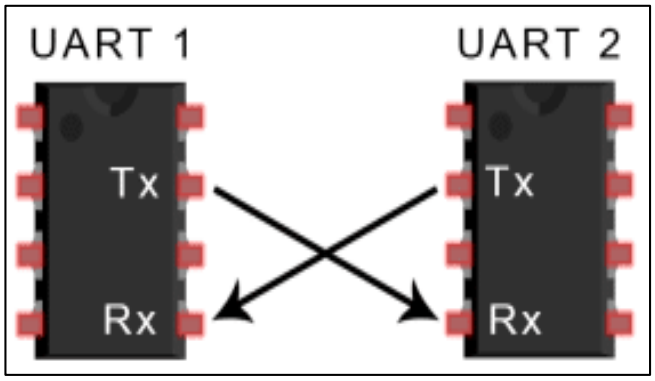

Fig 3

In UART communication, two UARTs communicate directly with each other. The transmitting UART converts parallel data from a controlling device like a CPU into serial form, transmits it in serial to the receiving UART, which then converts the serial data back into parallel data for the receiving device. Only two wires are needed to transmit data between two UARTs. Data flows from the Tx pin of the transmitting UART to the Rx pin of the receiving UART.

Steps of UART Transmission

1. The transmitting UART receives data in parallel from the data bus:

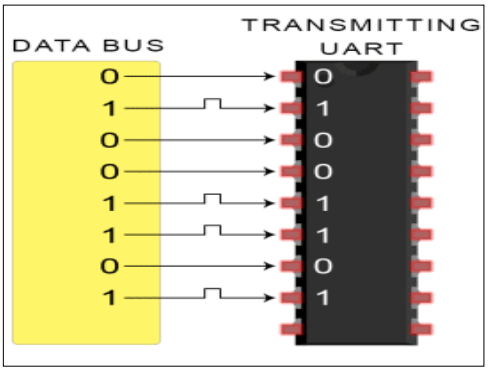

Fig 4

2. The transmitting UART adds the start bit, parity bit, and the stop bit(s) to the data frame:

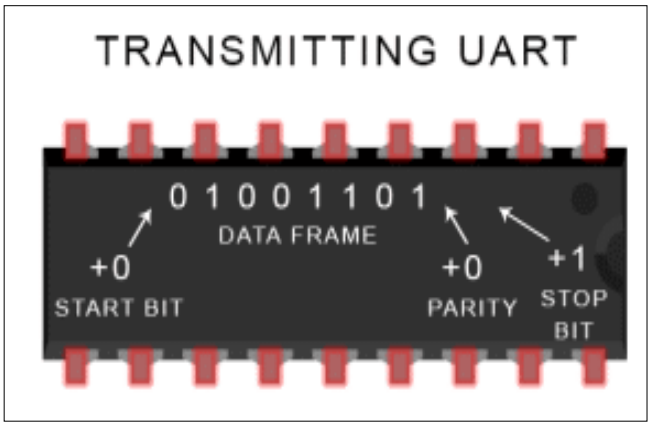

Fig 5

3. The entire packet is sent serially from the transmitting UART to the receiving UART. The receiving UART samples the data line at the pre-configured baud rate:

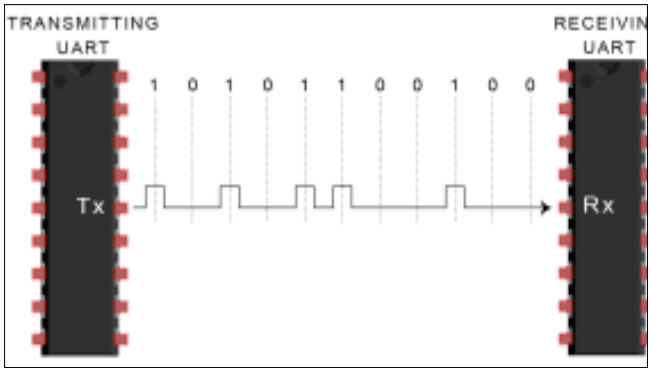

Fig 6

4. The receiving UART discards the start bit, parity bit, and stop bit from the data frame:

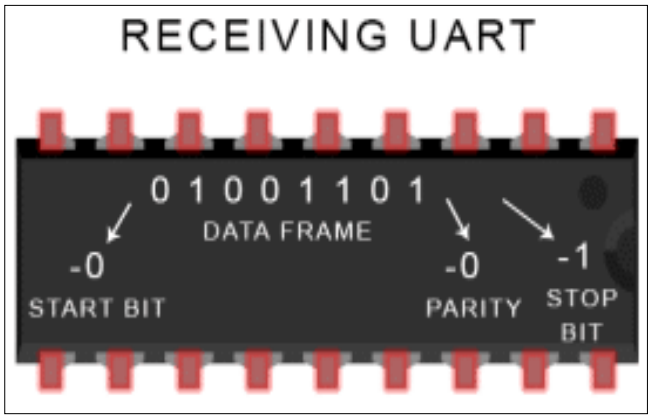

Fig 7

5. The receiving UART converts the serial data back into parallel and transfers it to the data bus on the receiving end:

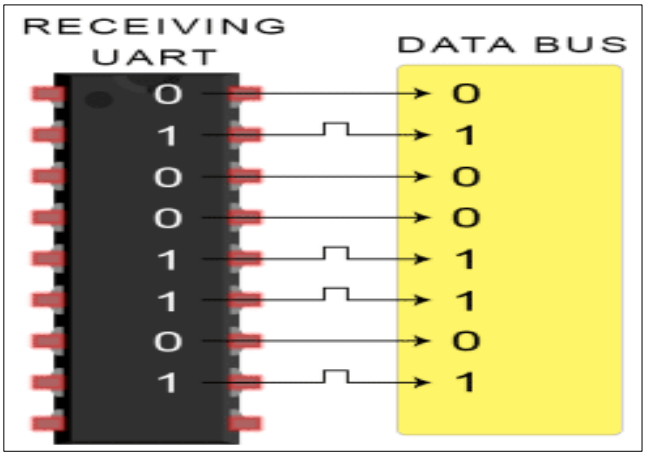

Fig 8

Advantages

- Only uses two wires
- No clock signal is necessary
- Has a parity bit to allow for error checking
- The structure of the data packet can be changed as long as both sides are set up for it

Conclusion

In this paper, we proposed a design of UART. It internally consists of transmitter, receiver and baud rate generator. The design is successfully simulated using Xilinx ISE software. The results are stable and reliable and show the correct functionality. Hence, we can improve the speed of UART using different baud rates.

References

1. Himanshu Patel, Sanjay Trivedi, Neelkanthan R, Gujraty VR. “a robust uart architecture based on recursive running sum filter for better noise Performance”.
2. Intel MCS-51 microcontroller Family User’s manual.
3. Synopsys Design Ware DW8051 Macro Cell Data book.
4. Delvai M, Eisenmann U, Elmenreich W. “Intelligent UART Module for Real-Time Applications” First Workshop on Intelligent Solutions in Embedded Systems (WISES), Vienna, Austria, 2002, 177-185.
5. Xilinx Virtex FPGA datasheets
6. Spartan-3 FPGA Family: Complete Data Sheet
7. Wilfried Elenmenriech, Marti Delvai.” Time triggered communication with UARTs.” 4th IEEE international workshop on factory communication system, Sweden, 2002.
8. Raffaele Gallo, Martin Delvai, Wilfried Elmenreich, Andreas Steininger. ”Revision and Verification of an Enhanced UART”, Austria, 2004.
9. Delvai M, Eiasenmann U, Elmenreich W. “Intelligent UART Module for Real-Time Applications.” First Workshop on Intelligent Solutions in Embedded Systems (WISES), Vienna, Austria, 2002, 177-185.
10. Mahat NF. “Design of a 9-bit UART module based on Verilog HDL”, in the proceedings of 10th IEEE International Conference on Semiconductor Electronics (ICSE), 2012. DOI: 10.1109/SMElec.2012.6417210,pp. 570-573