Implementation of Serial Communication using UART in FPGA

Naga Bala Subrahmanyam Gollapalli¹, Gopi Krishna Goli², Prem Chand Govathoti³, John Spurgeon Athyala⁴

¹⁻⁴ Undergraduate Students, Department of ECE, Vasireddy Venkatadri Institute of Technology, Nambur, Guntur, Andhra Pradesh, India

ABSTRACT

Many computers and microcontrollers include several serial data ports that allow them to communicate with serial input/output devices like keyboards and printers. A modem attached to a serial port allows for the transmission and receiving of serial data over telephone lines to and from remote destinations. The UART (Universal Asynchronous Receiver-Transmitter) is the serial communication interface that receives (RxD) and transmits (TxD) serial data. This technique enables asynchronous communication, allowing devices to exchange data without relying on a shared clock signal. The project develops into the UART-TxD and UART-RxD module's architecture, synchronization methods and hardware settings, emphasizing the FPGA's crucial role in efficient implementation and adaptability for various data transmission and receiving needs. Additionally, this project demonstrates practical applications for the FPGA-based UART-TxD and UART-RxD modules, including real-time data transfer and receive in embedded systems, IoT, and sensor interfaces. This resource caters to engineers and developers interested in utilizing FPGA boards for tailored serial communication solutions, providing insights into both theoretical and practical aspects of UART-TxD and UART-RxD serial communication within an FPGA framework implementation.

Keyword: - UART(Universal Asynchronous Receiver-Transmitter), FPGA(Field Programmable Gate Array), Debouncing, Ideal state, Transmission State, Receiver State, etc...

1. INTRODUCTION

In the fast changing field of digital electronics, Field Programmable Gate Arrays (FPGAs) have evolved as adaptable and powerful devices that provide a customized hardware foundation for a wide range of applications. One of the main reasons for FPGAs' appeal is their ability to efficiently implement complex communication protocols. In this project, we analyze serial communication, including the Universal Asynchronous Receiver/Transmitter (UART) protocol, and its implementation using FPGA technology.

1.1 FPGA Overview:

Field-Programmable Gate Arrays (FPGAs) are semiconductor devices that can be programmed and reprogrammed after manufacturing, allowing for very flexible digital circuit design and implementation. Unlike Application-Specific Integrated Circuits (ASICs), FPGAs strike a compromise between performance and reconfigurability, making them excellent for prototyping, quick development, and applications requiring adaptability.

FPGAs are made up of a variety of configurable logic blocks, interconnects, and programmable input/output (I/O) pads. The logic blocks can be programmed to perform specific operations, allowing designers to develop unique digital circuits. Because of their flexibility, FPGAs are ideal for a wide range of applications, including digital signal processing, embedded systems, and communication interfaces.

1.2 Serial Communication and UART:

Serial communication is a way of sequentially delivering data between two devices using a single communication connection. In contrast, parallel communication involves sending several bits over distinct lines at the same time.

The UART protocol, short for Universal Asynchronous Receiver/Transmitter, is a popular serial communication standard. It permits asynchronous communication between devices, which means that data is delivered without the use of a shared clock signal between the transmitter and receiver. Instead, both devices agree on a baud rate, which determines the rate of data transfer.

UART communication uses two pins: transmit (TxD) for sending data and receive (RxD) for receiving data. Each data frame normally consists of a start bit, a configurable number of data bits (usually 8 bits), an optional parity bit for error checking, and one or more stop bits to indicate the end of the data frame.

2. EXISTING SYSTEM

In the previous existing system we may observed that all the devices are accepting and transferring the data from the various FPGA boards like Xilinx Spartan-7 FPGA module, VIRTEX 5 FPGA board, etc.. with various methods like synchronized methods and structure method. Uses the more logic units to perform any one of the module either transmission or reception.

Using 8-bit data, we can transmit or receive data. We can swap states in the form of existence via registers. In our system, we operate both the transmitter and the receiver in parallel. This demonstrates correct serial data transfer while also operating the UART port in parallel.

3.PROPOSED SYSTEM

The suggested UART Implementation for serial communication employs the structural methodology, which produces a set of values that precisely reflect what is happening inside the device. In the present structured and thorough strategy, we may use fewer LUT's than in other existing systems.



Fig-1: functional Block diagram of UART

3.1 UART TxD MODULE :

The transmitter module in Verilog represents a UART transmitter. It features a state machine that manages serial data transmission and data storage via a shift register. The module uses a positive clock edge and includes logic for baud rate generation, data loading, and bit shifting. It responds to a reset signal and a transmit trigger, switching

between idle and transmit modes. During transmission, it sends 10 bits of data, including start and stop bits, and updates the transmitter output (TxD) accordingly. The state machine ensures that transmission events are properly synchronized and sequenced, resets counters, and shifts data as needed.

On the other hand, the debounce module defines a button debounce method with a predefined threshold. It uses three flip-flops for signal synchronization and a 31-bit counter to keep track of the duration of button presses. The counter increases when the button is pressed and decreases when it is released. When the counter exceeds the defined threshold, the module outputs a new flipflop with the value 1 to indicate a stable button state. Then the flipflop2 and third flipflop are acted as defined then the debounce signal(transmit) is with value 1. This debounce technique serves to filter out noise and variations in the button signal, resulting in a reliable indication of button presses for external use, such as initiating the UART transmission in the associated transmitter module.







Fig-3 : SM chart for Transmitting module

3.2 UART RxD MODULE:

The receiver Verilog module is a UART receiver intended for usage in FPGA-based systems like the Basys 3 board. The receiver operates at a clock frequency of 100.1 MHz and uses a state machine to receive serial data via the RxD input. The module has internal variables for controlling bit counts, baud rate counters, and received data shifting. It has a parameterized setup that includes constants for clock frequency, baud rate, and division factors to meet various system requirements. The module uses a shift register (rxshift_reg) to store incoming 10-bit serial data, minus the start and stop bits. The RxD input causes a state machine to change from an idle state to a receiving state. The receiver synchronously samples incoming data at predetermined intervals and stores it in the register. Additionally, the module includes reset capability for counters and state variables. The output, RxData, contains the 8-bit data component of the received byte, which can be utilized for additional processing or display. This UART receiver module enables reliable serial communication within an FPGA system while taking into account synchronization and bit sampling requirements.

•





Fig-6: output of transmission

Fig-7: Tera term console

Receiver : provides the 8-bit binary value of character A(01000001)





Fig-7: output of receiving

5. CONCLUSION

The project focuses on the design and implementation of UART-TxD and UART-RxD modules, including their architecture, synchronization mechanisms, and hardware configuration. The total modules operate at the same baud rate (speed) of 9600.FPGAs are critical to this project because they allow for efficient implementation and customization of the UART-TxD and UART-RxD modules to meet a variety of data transmission and receiving requirements. The architecture was successfully implemented in FPGA.Finally, simulation, synthesis, and implementation with the Xilinx VIVADO tool are completed.

6. REFERENCES

[1]. Universal Asynchronous Receiver and Transmitter (UART) by Umakanta Nanda, Sushant Kumar Pattnaik. (IEEE Published)

[2]. A Review Paper on Design and Simulation of UART for Serial Communication by Vibhu Chinmay, Shubham Sachdeva (IJIRT)

[3]. Analysis of Universal Asynchronous Receiver and Transmitter for Reliable Data Transmission by Yogeesh K, Venkatesh Kumar ,Rohith (IOSR-JVSP)

[4]. UART-TxD & UART-RxD serial communication by Prof. Mughal (YouTube)

[5]. Design and implementation of UART Using FPGA Board by Anchal Govil, Anmol Karwal, Govind Sindhu, Ayush Singh, Dr. Shubham Shukla (IJRASET)

[6]. UART Implementation Using FPGA by Mukul lokhande, Rohit Prabhakar Mate (Research gate)