IOSR Journal of Engineering (IOSRJEN) **www.iosrjen.org**
ISSN (e): 2250-3021, ISSN (p): 2278-8719
Vol. 14, Issue 8, August 2024, ||Series -1|| PP 01-07

Design and Implementation of UART Transmitter and Receiver Using FPGA

SHILPA K C [1] MOHAN GOWDA H [2], NIDISH SHETTY [3], NIRANJAN REDDY [4], PRAFULL K S [5]

ECE Dept, Dr. Ambedkar Institute of Technology, Bengaluru, India,
Received 20 July 2024; Accepted 03 August 2024

Abstract: The project titled "Design and Implementation of UART Transmitter and Receiver Using FPGA" focuses on the development of a versatile and efficient solution for serial data communication employing Universal Asynchronous Receiver/Transmitter (UART) protocols within the context of Field-Programmable Gate Arrays (FPGAs). UART communication is a fundamental component in embedded systems, providing a reliable means of data exchange between different devices. This project aims to design, simulate, and implement a UART transmitter and receiver system on an FPGA platform, enabling robust and bidirectional serial communication.The primary objective of this project is to design, simulate, and implement a UART transmitter and receiver system on an FPGA platform, facilitating serial data communication between various devices. The project encompasses the following key components:1. UART Communication Protocol: The project includes the implementation of the UART protocol for serial data transmission and reception. This involves configuring baud rates, managing start and stop bits, and defining the data frame format.2. FPGA Platform:An FPGA development board is employed as the hardware platform for implementation. FPGAs are chosen for their reconfigurability and parallel processing capabilities, making them ideal for handling serial data communication.3. Hardware Description Language (HDL): To describe the UART transmitter and receiver modules, HDL, such as VHDL or Verilog, is utilized. This allows for efficient hardware synthesis and integration into the FPGA.4. Simulation and Verification: Prior to FPGA synthesis, the project employs simulation tools torigorously test and verify the functionality and reliability of the UART modules. This ensures that the system operates as intended and is free from errors.5. Baud Rate Configuration:The system is designed to allow users to configure baud rates, data bits, parity settings, and stop bits to accommodate a wide range of communication requirements..

I. Introduction

In the ever-evolving world of embedded systems, efficient and reliable data communication is paramount. Within this intricate ecosystem, Field-Programmable Gate Arrays (FPGAs) have emerged as powerful tools for designers, offering unparalleled flexibility and customization. One of the cornerstones of data exchange inFPGAs is the Universal Asynchronous

Receiver Transmitter, or UART. This document delves into the fascinating interplay between FPGAs and UARTs, exploring their functionalities, design considerations, and the unique advantages they offer for embedded system development.

The I/O Maestro: Orchestrating the Flow ofInformation

Imagine a bustling city – the FPGA – brimming with processing power and a multitude of logic elements. This city thrives on innovation and thrives on interacting with the outside world. However, to exchange vital information andinstructions, it needs a reliablecommunication channel. Enter the I/OProcessor (IOP) core, the unsung hero responsible for managing data flow between the FPGA and a Personal Computer (PC). In the context of UART communication, the IOP transforms into a skilled maestro, orchestrating the flow of serial data between the PC and the Design Under Test (DUT) residing within the FPGA.

The DUT could be a custom processor core, a complex algorithm implementation, or any other logic design undergoing testing or evaluation. The IOP, acting as the UART interface, facilitates the exchange of crucial information between the PC (the testengineer) and the DUT. This two-way communication is the lifeblood of embedded system development, enablingdebugging, configuration, data transfer, andultimately, the creation of robust andreliable systems.

Beyond the Parallel Universe:Demystifying UART

UART stands for Universal Asynchronous Receiver Transmitter, a hardware marvel that bridges the gap between the parallel data processing world of computers and theserial data transmission realm. While data

processing within devices occurs in parallel for sheer speed (imagine multiple lanes on a highway), communication between devices often takes the form of a single lane, with bits transmitted one after anotherlike cars on a single-lane road. This is where UART steps in, gracefullyconverting parallel data into a streamlined serial stream for transmission and performing the reverse operation (serial-to- parallel conversion) at the receiving end.The "Universal" aspect of UART refersto its adaptability. Unlike some communication protocols with rigid specifications, UART allows for configuration of the data frame format and the transmission speed. This flexibility caters to diverse communication needs, ensuring compatibility with various devices and applications. For instance, a UART canbeconfigured for a low baud rate (number of bits transmitted per second) for short- distance communication or a high baud ratefor high-speed data transfer.

The "Asynchronous" nature of UARTsignifies the absence of a shared clock signal to synchronize the transmitter and receiver. This might seem counterintuitive, but it actually simplifies design and reduces complexity. In asynchronouscommunication, the transmitter andreceiver rely on embedded control bits within the data stream to achieve synchronization. This makes UART suitable for a wide range of scenarios, particularly those where a dedicated clock line might be impractical or unnecessary.

II. Literature Survey

In the intricate world of electronics, communication with the external environment is paramount. Whether transferring data to another device, sendingor receiving commands, or facilitatingdebugging, a reliable communication interface is essential. Enter the Universal Asynchronous Receiver Transmitter, or UART – a ubiquitous protocol that has stood the test of time.

Serial Communication: A Streamlined Approach

UARTs operate on the principle of serial communication, a method where data istransmitted one bit at a time, contrasting with parallel communication wheremultiple bits travel simultaneously. Thisapproach offers several advantages. Firstly, it reduces the number of wires required, simplifying cable design and lowering costs. Secondly, it allows for easier integration into compact devices where space is a premium.

From RS-232 to USB: A Journey of Adaptability

In the early days of computing, the RS-232 interface reigned supreme for serial communication. This standard defined the physical and electrical characteristics forconnecting devices – the familiar serial portfound on older personal computers. However, with the advent of miniaturization and the rise of laptops, the bulky and power-hungry RS-232 graduallygave way to the more versatile and user- friendly Universal Serial Bus (USB). Today, while many modern PCs lack dedicated serial ports, UARTs can be implemented using USB converters, seamlessly bridging the gap between legacydevices and contemporary systems.

The Inner Workings of a UART: A Symphony of Conversion

At its core, a UART performs three criticaltasks:

1. Parallel-to-Serial Conversion: When data arrives from a device's internal data bus (typically in parallel format), the UARTmeticulously converts it into a serial streamof bits, ready for transmission.
2. Serial Transmission: The converted serialdata stream is then transmitted over a singlecommunication line.
3. Serial-to-Parallel Conversion: Uponreceiving the serial data stream at the destination device, the UART reverses the process, converting the data back into its original parallel format for further processing.

A Historical Perspective: The Evolution of UART TechnologyThe story of UARTs is a fascinating journey of innovation. The first documented UART design emerged in 1971, credited to Western Digital. However, some historical accounts attribute the conceptual groundwork for UARTs toGordon Bell, who envisioned their use in the PDP series of computers. A key innovation of these early UARTs was the use of sampling to convert analog signals tothe digital domain, offering improved timing accuracy compared to traditional telegraphic circuits.

Driven by the desire to reduce costs and simplify designs, companies like DigitalEquipment Corporation (DEC) further refined UART technology. They achieved this by consolidating the line unit design into a single integrated circuit (IC) – a milestone that marked the birth of the first commercially available UART.

The Rise of Programmable Baud Rate Generators and FIFO Buffers

The 1980s witnessed a significant advancement with the introduction of National Semiconductor's 8250 UART. This groundbreaking chip incorporated an on-chip programmable baud rate generator. This innovation allowed for dynamic adjustment of the transmission speed (baud rate) to adapt to various communication needs. This flexibility proved crucial for interfacing with diverse devices operating at different data transfer rates. The 8250 UARTs found widespread adoption in the iconic IBM PC5150, further solidifying their place in the personal computinglandscape.

Recognizing the potential for increased efficiency and smoother data flow, Intel entered the scene with their 82510 UART. A key feature of this design was the inclusion of two independent 4-byte FIFOs (First-In-First-Out) buffers. These buffers acted as temporary storage locations, mitigating the effects of potential mismatches between data transmission andreception rates. This innovationsignificantly reduced the occurrence of interrupts

within the system, ultimately leading to more robust and efficient communication.
Following the success of Intel's 82510, the door was opened for further advancements. Numerous companies embarked on developing UARTs with varying FIFO buffer capacities and transmission speeds to cater to the diverse requirements of an ever-growing embedded systems market.
Beyond the Limits: Dual, Quad, and Octal UARTs
The quest for more communication channels led to the development of Dual UARTs (DUARTs) by NXP
Semiconductors. These ingenious chips integrated two independent communication channels within a single package, each equipped with dedicated control registers and counter/timers. This design allowed for simultaneous transmission and reception on two separate channels, significantly enhancing communication capabilities for multitasking applications. Notably, some popular DUART models include theSCC2692, SC26C92, and SC28L92.

III. PRE-REQUIREMENTS OF PROJECT

Concept of Project Before starting the project, one should look upon the pre-requirements needed for their project. In this project, pre-requirementsfor UART Implementation using FPGAincludes the study of serial communication, how data is transferred from peripheral devices to computer and vise-versa. UART transmitter starts the transmission by taking a data frame in parallel form and advising the UART totransmit the data frame in a serial form. Similarly, the receiver must detect transmission and receive the data in serial form, remove the start bit and stop bit, and store the data frame in parallel form. As the project states, UART is Universal Asynchronous Receiver Transmitter, we should know the difference between synchronous and asynchronous system. There are some main terms that we should study before starting the project work and the main terms includes UART, baud rate, data frame, etc. We should study the main block of UART separately - transmitter, receiver and baud rate generator. As thiswill have registers, flip flops, latches, counters and others one should know Area optimization technique so that less amount of area must be used on theboard. Now, we will study some concepts that we need in this project.

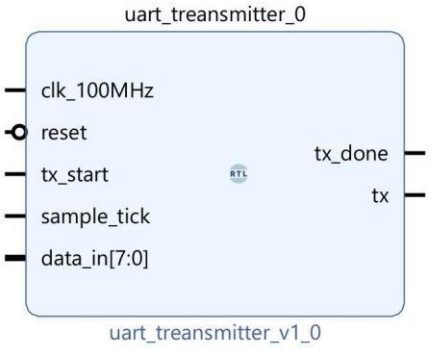
IV. PROPOSED WORK

Top level diagram

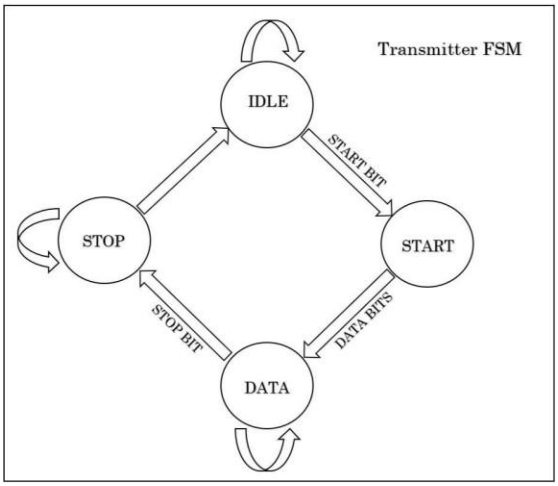
The Universal Asynchronous Receiver Transmitter, or UART, serves as a cornerstone for data communication in embedded systems. This ubiquitousprotocol facilitates the exchange of information between a device and the external world, enabling functionalities likedata transfer, command exchange, and debugging. At the core of a UART lies a fascinating interplay between three fundamental blocks: the Transmitter, the Receiver, and the Baud Rate Generator. Each of these blocks plays a critical role inthe seamless transmission and reception of data. This comprehensive exploration delves into the intricate micro-architecture and operational principles of these modules, unveiling the magic behind UART communication.

Transmitter Module

The operations in transmitter are with respect to clock which runs on multiple baud rates. The transmitter converts the 8- bit parallel data into serial data and adds start bit at start of the data frame and stop bit at the end of the data frame.
In transmitter, the 8-bit data from data bus is loaded to transmitter hold register (THR),it is an 8-bit register to store the data parallelly through d_in pin. Then after all the data is loaded in THR then data is loaded in Transmitter shift Register (TSR) parallelly. In TSR, start bit and stop bit is added to the data bits, transmitter shift register is of 10-bit size. TSR convert parallel data into serial form and then data is transmitted serially bit by bit through.
tx_out. While transmitting data LSB of the data bit is transmitted first and then remaining data is transmitted and last stop bit is transmitted to stop the transmission. The clock given to transmitter block is the baud clock (bclk) on which the baud rate is set. Counter is used in TSR to count the databits. Until and unless all the data is transmitted no other data can be stored in hold register.

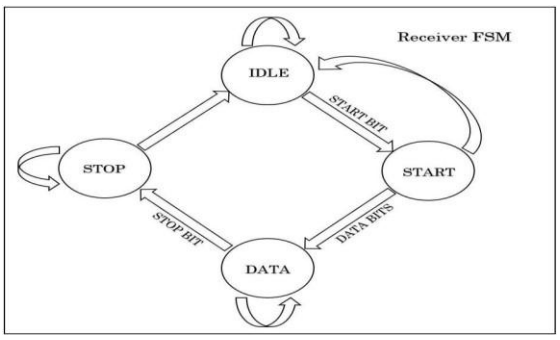


Transmitter is designed based of FSM used asgiven below:



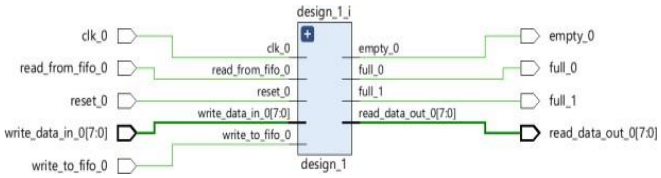
Receiver Module

The receiver module is complex than the transmitter module. It runs 4 X baud rate faster than transmitter. The serial data is received from transmitter through rx_data pin 1-bit at a time serially. Then, rx_data pin jumps into logic 0 from logic 1 indicating beginning of the data frame. Thestart bit is identified by change in level fromhigh to low level and stop bit by change in the level from low to high. After identifyingstart bit, all the data bits are sampled and counted using the bit counter generated in the receiver. The counter counts the positive edge of the clock at every riding edge of the clock 1 bit is counted. When count equals to 8 then it says all bits are received and stored in an 8 bit Receiver Shift Register (RSR) and the data bits are converted into parallel data. Then it sends the data bits to Receiver Hold Register (RHR) which is also an 8 bit register. RSR send only data bits to RHR, start bit and stop bit is not sent. Then the parallel data issent to the data bus or peripheral device.



V. RESULTS

RTL Schematics

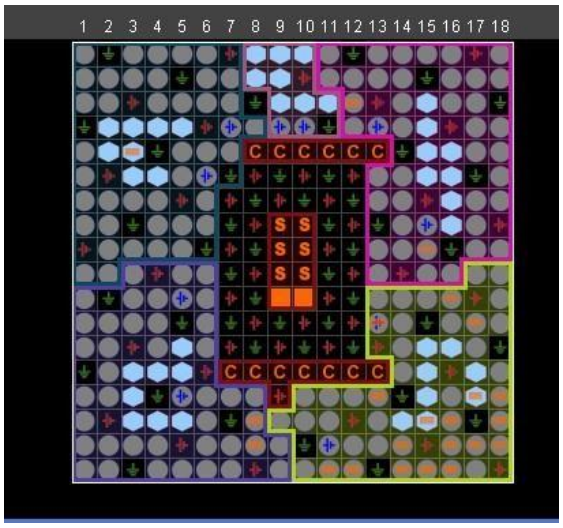


Top Level Simulation

FIFO Simulation

Baud Rate Generator Simulation

Device Package



VI. CONCLUSION

This project showcases a configurable baud rate UART design implemented in Verilog for reliable serial data communication. The design leverages finite state machines to manage the Transmitter and Receiver modules.

Successfully synthesized using Xilinx ISE 14.7, the UART has been comprehensively simulated and verified for various baud rates within the tool suite. This configurable baud rate capability makes the design suitable for a wide range of applications with varying data transmission speed requirements.

To delve deeper, the Verilog code describes the logic for both the Transmitter and Receiver sections. The Transmitter module is responsible for framing the data to be sent by adding start and stop bits, and potentially parity bits for error checking.

The baud rate configuration determines thetiming for these bits, ensuring accurate data transmission at the chosen speed. TheReceiver module waits for the start bit, captures the data bits, and verifies any parity bits before presenting the received data. Finite state machines manage the intricate logic for these operations within both Transmitter and Receiver, ensuring reliable data exchange.

The configurable baud rate functionality is a key strength of this design. It allows the UART to adapt to different communication protocols and devices. For instance, a low baud rate might be suitable for battery-powered sensor communication where minimizing power consumption is crucial. Conversely, high baud rates can beused for faster data transfer applications, such as video or audio transmission. The ability to configure the baud rate on-the-flyprovides flexibility in various scenarios.

When considering real-world applications, error handling mechanisms become vital. Techniques like parity checking or framingerror detection can be incorporated into thedesign to identify and potentially correct errors during data transmission. These techniques add an extra layer of robustness to the communication, especially for critical applications where data integrity isparamount.

In conclusion, this configurable baud rate UART design offers a versatile and reliablesolution for serial data communication. Theimplementation in Verilog and successful synthesis using Xilinx ISE 14.7 demonstrate a well-structured approach. The design's adaptability to various baud rates makes it suitable for a broad spectrumof applications, from low-power sensor communication to high-speed data transfer.Further exploration could involve incorporating error handling mechanisms and exploring specific use cases in greater detail. This project serves as a solidfoundation for further development and customization to meet the specific needs ofreal-world serial communication applications.

Future Scope

The previously described UART design with configurable baud rates lays a strong foundation for serial communication. However, for robust data transmission, incorporating status registers and error logic significantly improves its functionality. These additions enhance reliability by providing feedback on the UART's state and potential errors. The transmitter's status register can include flags indicating buffer emptiness and transmission completion, allowing for optimized data flow. Additionally, error logic can detect parity or framing issues before transmission. The receiver benefits similarly with a status register flagging received data availability, overruns, and parity/framing errors. This error detection allows the controlling entity to take corrective actions or handle errors gracefully. Overall, these modifications transform the UART design into a

more reliable and efficient communication tool.

References:

[1]. Alexander Barkalov LT; Logic Synthesis for FSM- Based Control Units. Springer Berlin Heidelberg, 2009.
[2]. Ali L, Sidek R, Aris I, Ali AM, SuparjoBS; Design of a micro-uart for soc application. Computers & Electrical Engineering, 2004; 30(4):257-268.
[3]. Barkalov AA, Titarenko LA, EfimenkoKN; Optimization of circuits of compositional microprogram control units implemented on fpga. Cybernetics and Sys.Anal, 2011; 47(1):166–174.
[4]. Bomar BW; Implementation of microprogrammed control in fpgas.Industrial Electronics, IEEE Transactions on, 2002; 49(2): 415–422.
[5]. Digilent. Digilent Basys2 Spartan-3EFPGA Board.
[6]. HU Likun WQ; Uart-based reliable communication and performance analysis. In Computer Engineering, 2006; 32: 15–21.
[7]. Norhuzaimin J, Maimun H; The designof high speed uart. In Applied Electromagnetics, APACE 2005. Asia-Pacific Conference on, 2005.
[8]. Roth Jr. CH, John LK; Digital SystemsDesign Using VHDL. Thomson- Engineering, 2007.
[9]. Tomasi W; Advanced electronic communication systems, Third Edition.Prentice-Hall, 1994.
[10]. Wis´niewski R, Barkalov A, TitarenkoL, Halang W; Design of microprogrammedcontrollers to be implemented in fpgas. Int.
[11]. J. Appl. Math. Comput. Sci, 2011; 21(2):401– 412.
[12]. Yi-yuan F, Xue-jun C; Design and simulation of uart serial communication module based on vhdl. In Intelligent Systems and Applications (ISA), 2011 3rd International Workshop on, 2011; 1-4.
[13]. Ms.Neha R. Laddha, Prof. A. P. Thakare, "A Review on Serial Communication by UART", Volume 3, Issue 1, January 2013 International Journal of Advanced Research in Computer Science and Software Engineering.
[14]. Kavyashree S, "Design and Implementation of UART using Verilog", International Journal Of Engineering And Computer Science ISSN:2319-7242 Volume – 4 Issue - 12December, 2015 Page No. 15240-15245
[15]. FANG Yi-yuan, CHEN Xue-jun,"Design and Simulation of UART Serial Communication Module Based on VHDL", College of Electrical and Electronic EngineeringShanghai University of Engineering Science Shanghai, China
[16]. KeyStone Architecture Universal Asynchronous Receiver/Transmitter(UART) User Guide by Texas Instruments.
[17]. Hazim Kamal Ansari, Asad Suhail Farooqi, " Design Of High Speed Uart For Programming Fpga", International Journal Of Engineering And Computer Science Volume1 Issue 1 Oct 2012 Page No. 28-36
[18]. Virtex 5 FPGA Starter Kit Board UserGuide https://www.xilinx.com › products ›silicon-devices › fpga
[19]. Umakanta Nanda, Sushant Kumar Pattnaik, "Universal AsynchronousReceiver and Transmitter(UART)", 2016 3rd International Conference on Advanced Computing and Communication Systems (ICACCS -2016),Jan. 22 – 23, 2016, Coimbatore, INDIA