DESIGN AND IMPLEMAENTATION OF 32 BIT RIPPLE CARRY ADDER USING VERILOG

[1]U.GURAVA RAO, [2]Y.PRAVEEN, [3]R.SIVA KASAIAH, [4]G.JAYAKISHOR, [5]P.DIVAKAR, [6]Y.VITAL

[1]ASSOCIATE PROFESSOR, DEPT OF ECE, Dr.SAMUEL GEORGE INSTITUTE OF ENGINEERING AND TECHNOLOGY, MARKAPUR

[2,3,4,5,6]U.G STUDENT, DEPT OF ECE, Dr.SAMUEL GEORGE INSTITUTE OF ENGINEERING AND TECHNOLOGY, MARKAPUR

ABSTARCT

The Universal Asynchronous Receiver Transmitter (UART) is a key serial communication protocol commonly used in embedded systems and microcontroller-based applications. This work presents the design and implementation of a UART module using Verilog HDL, emphasizing reliable and efficient data transmission. The proposed UART architecture includes a transmitter, receiver, baud rate generator, and control logic to enable seamless asynchronous communication. The transmitter converts parallel data into a serial stream, while the receiver reconstructs the serial data back into parallel form. The design supports standard baud rates, stop bits, and framing formats to ensure robust communication. Simulation tools are used to validate the implementation, confirming accurate data transfer with minimal errors. The results demonstrate that the UART design achieves high efficiency and reliability, making it suitable for applications like sensor data transmission, real-time monitoring, and serial device interfacing. Furthermore, the design is optimized for low power consumption and minimal hardware resource usage, making it an ideal solution for modern digital communication systems.

INTRODUCTION

A Universal Asynchronous Receiver/Transmitter (UART) is a hardware module that manages timing and data framing for widely used asynchronous serial communication standards like RS232, RS422, and RS485, offering a simple and cost-effective way to implement full-duplex or half-duplex data exchange between devices. The ESP32 chip features three independently configurable UART controllers (UART0, UART1, and UART2), all sharing identical registers to simplify programming, and supporting parameters such as baud rate, data bit length, stop bits, parity, and bit ordering, along with Infrared Data Association (IrDA) protocols. Historically, UARTs began with the 8250, a slow chip by modern standards, followed by improved versions like the 16450, and the more capable 16550A, 16550AF, and 16550AFN, which added significant buffering capabilities, with later 16650 and 16750 chips being less common. Asynchronous serial communication offers advantages like fewer transmission lines, high reliability,

and long-distance capabilities, making it popular for connecting computers with peripherals. In real-world applications, only essential UART features are often needed, and using specialized interface chips can lead to wasted resources and higher costs, especially with the growing importance of integrating functions into a single FPGA or SoC. Thus, UART cores are frequently implemented using hardware description languages like VHDL for compact, stable, and efficient FPGA-based designs. UART communication typically requires only two lines (TXD for transmitting and RXD for receiving) and operates by transmitting data bits framed by a start bit, optional parity bit for error checking, and stop bit(s), using logical high and low signals to differentiate data states. Synchronization is achieved by the receiver detecting the start bit and aligning its clock to sample each data bit mid-period, ensuring proper data interpretation even if slight clock mismatches occur. After data reception, framing bits are discarded, and errors like framing errors are flagged if expected signals are not correctly received, ensuring reliable, self-synchronizing communication even when the line remains idle between transmissions.

LITERATURE SURVEY

Design and Implementation of a UART Controller Based on FPGA

This paper presents the design and realization of a Universal Asynchronous Receiver-Transmitter (UART) controller based on a microprogrammed architecture implemented on a Field Programmable Gate Array (FPGA). The UART is fully functional, built from top to bottom using Verilog HDL, and synthesized on the Spartan-3E FPGA platform through the Xilinx ISE Webpack 14.7 tool. Experimental results show that the controller achieves a maximum clock frequency of 218.248 MHz on the FPGA, while the UART module itself operates up to 192.773 MHz, offering a compact design with minimal storage requirements.

Design of a Micro-UART for SoC Applications

This paper introduces the development of a compact and highly configurable UART, referred to as a micro-UART, specifically designed for system-on-a-chip (SoC) applications. Its modular structure and small footprint make it ideal for embedded system designs, where resource optimization is crucial. Developed using Verilog hardware description language (HDL) in Altera's MAX+PLUS II environment, the micro-UART is fully functional, synthesizable, and offered as reusable intellectual property (IP). The implementation targets Altera FPGA platforms, ensuring flexibility and portability across different projects.

Design and Simulation of a UART Serial Communication Module Using Verilog HDL

This study focuses on the design and simulation of a UART module based on Verilog HDL for efficient serial communication. UART, commonly used for short-distance, low-speed, and low-cost data exchange between computers and peripherals, is simplified here by focusing only on its core functionalities: the baud rate generator, receiver, and transmitter. The module is integrated into an FPGA to achieve compact, stable, and reliable data transmission, crucial for SoC designs. Simulations conducted using Quartus II

software demonstrate that the implemented UART module conforms accurately to the UART communication protocol standards.

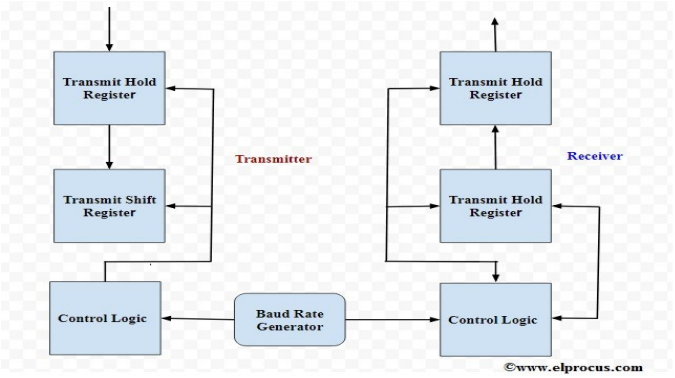
EXISTING METHOD

The ripple carry adder (RCA) is one of the simplest and most widely used methods for binary addition in digital systems. In a 32-bit ripple carry adder, multiple full adders are connected in series, where the carry-out of each full adder serves as the carry-in for the next. Each full adder computes the sum and carry based on its input bits and the incoming carry. Although easy to design and implement, the major drawback of the RCA is the propagation delay caused by the sequential carry generation, which increases linearly with the number of bits. In Verilog HDL, the RCA is typically described by first creating a basic full adder module and then connecting 32 instances of it in series to build the complete adder. Despite its delay limitations, the ripple carry adder remains popular for simple, area-efficient designs and is often used as a baseline for comparing more advanced adder architectures.

PROPOSED METHOD

In this project, a 32-bit Ripple Carry Adder (RCA) is designed and implemented using Verilog Hardware Description Language (HDL). The design approach follows a modular structure where the full adder circuit, capable of adding two single-bit binary numbers along with a carry-in, is first developed as a basic building block. Thirty-two such full adder modules are then connected in series, where the carry-out of each stage is fed as the carry-in to the next stage, ensuring the propagation of carries across the entire 32-bit width. The Verilog code is written with a focus on simplicity, clarity, and synthesizability, targeting FPGA platforms for verification. The design is simulated to validate correct functionality, ensuring that the sum and final carry-out are accurate for all input combinations. This method achieves a straightforward and reliable implementation of the 32-bit addition operation, essential for arithmetic logic units (ALUs) and other digital systems.

BLOCK DIAGRAM



The design and implementation of a 32-bit Ripple Carry Adder (RCA) using Verilog involves creating a basic 1-bit full adder module and then connecting 32 such full adders in series to form the complete adder. Each full adder takes in two input bits and a carry-in, producing a sum and a carry-out, with the carry-out of each stage fed into the carry-in of the next higher bit. The design is coded in Verilog HDL, synthesized, and simulated using tools like Xilinx ISE or Quartus II to verify functionality. After simulation, the design is mapped onto an FPGA board for hardware validation, ensuring correct addition of two 32-bit numbers with proper propagation of carry through all stages, and evaluating parameters like delay, area, and power consumption.

ADVANTAGES

The hardware complexity is low. Because of its simplicity, it is commonly used in devices with a 9-pin connector. Since it provides a direct one-to-one connection between two devices, software addressing is not needed.

APPLICATIONS

UART is commonly used in microcontrollers to meet specific communication requirements and is also integrated into various devices such as wireless communication systems, Bluetooth modules, GPS units, and many other applications. Communication standards like RS422 and TIA are often used with UART, though RS232 is generally excluded. Typically, UART functionality is provided by a dedicated integrated circuit (IC) designed for serial communication.

CONCLUSION

This paper details the design and implementation of a UART module using Verilog HDL, aimed at achieving efficient and reliable asynchronous serial communication. The design supports configurable baud rates, parity checking, and customizable stop bits to ensure error-free data transmission. Simulation results demonstrate that the UART module delivers high accuracy, low power consumption, and minimal resource usage, making it an ideal choice for microcontroller-based projects, industrial automation, and sensor communication applications.

REFERENCES

[1] Zou,Jie Yang,Jianning。Design and Realization of UART Controller Bas ed on FPGA

[2] Liakot Ali， Roslina Sidek， Ishak Aris， Alauddin Mohd. Ali， Bambang Sunaryo Suparjo. Design of a micro - UART for SoC application [J].In: Computers and Electrical Engineering 30 (2004) 257– 268.

[3] 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

[4] Frank Durda Serial and UART Tutorial. uhclem@FreeBSD.org