International Research Journal on Advanced Engineering Hub (IRJAEH)
e ISSN: 2584-2137
Vol. 02 Issue: 05 May 2024
Page No: 1468 - 1477
https://irjaeh.com
https://doi.org/10.47392/IRJAEH.2024.0203

# Design and Implementation of Verilog Based High Speed Low Power UART

*Ms. Banti Kumari[1], Ms. Kanika Jindal[2], Mr. Amit Bindal[3]*
*[1,2,3] Noida Institute of Engineering and Technology, Greater Noida, India.*
***Emails:*** *singhbanti1994@gmail.com[1], kanika.jindal@niet.co.in[2], amit.kumar.ece@niet.co.in[3]*

## *Abstract*

*The most crucial component of serial communication is a microcircuit called a universal asynchronous receiver/transmitter (UART). Receive-transmitter asynchronous technology is known as UART, and it is widely used for device-to-device communication protocols. Using asynchronous serial communication at a speed that can be adjusted. A hardware communication technique called UART Asynchronous conditions occur when the output of the transmitting device and the receiving end are not in sync with a clock. In UART, receiving a signal is known as RxD, and transmitting a signal is known as TxD. In comparison to the existing conventional UART design, we were able to reduce delay by 29% and power usage by 33% using our approach. The effectiveness of the novel UART design is noticed with the reduction in delay and power consumption. Synthesis and simulation are done in Xilinx ISE and Modelsim and Verilog HDL is used to implement a unique UART design.*

***Keywords:*** *UART; RS-232C; Transmitter; Receiver; Asynchronous.*

## 1. Introduction

The term UART refers to the Universal Asynchronous Receiver Transmitter Protocol. By looking at the name alone, we can deduce that UART is used for serial transmission. Direct communication between two UARTs occurs during UART communication [1]. Parallel data is sent in serial form by a controlling device, like a CPU, to the sending UART. Following that in order to convert the data into its original parallelize form for the accepting device, it is transferred from the sending UART to the receiving UART. In UART, two cables are established, with one wire being used for transmission and the other for receiving. The main motive of this project is to provide a high-speed low power UART with RS232 standard signaling protocol that defines things like pin assignments, voltage levels and so forth and show the comparison results between existing conventional UART design on the basis of delay, power consumption, number of logic gates, number of IOBS etc [2]. Today, Numerous products use UART, including RFID-based software, Bluetooth modules, GSM and GPRS modems, GPS receivers, and wireless communication systems [17]. whereas in this project UART with RS232 standard is uses for transmission of data signal which gives some advantages over a conventional UART. UART with RS232 operates on the principle of asynchronous meaning that the data is transmitted individual bits without the need for continuous clock signal.RS232 is able to communicate up to 15 m at a rate of 1.492 kpbs without any interruption between two devices [15].

## 2. Implementations of UART

### 2.1. UART Data Frame

A universal asynchronous receiver/transmitter, or UART, enables full-duplex transmission and receiving by converting data between serial and parallel communication. It has a serial data stream that bit by bit transforms the input serial data into byte transmission and the computer's parallel data into output; data streams for input and output are checked for parity; Include operations in the output data stream, like removing start and stop tags. Asynchronous communication often uses the following frame format: 1 bit for the start, 5 to 8 bits for the data bit, optional parity bit, parity, or no parity, and stop bit (1, 1.5, or 2 bits). Format for UART data frames [3]. As seen in Below Figure 1.

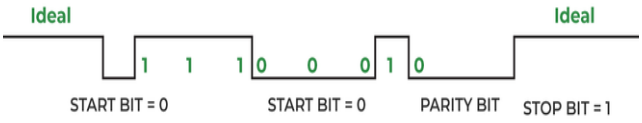


**Figure 1 UART Data Frame [31-36]**

To speed up the process, the parity bit is not set in this step. Eight bits of data, one bit for start, one bit for stop, and no parity bit make up the format of a data frame. Rs-232 is the standard communication interface, and full-duplex communication mode is used.[37]

### 2.2. Transmission Module

Sending the data is the sending module's primary job. Along with other modules, it includes FIFO, data sending logic unit, and monitoring logic unit. This module uses baud rate to determine the corresponding transmission frequency division coeffi-cient. TXFIFO receives the data to be delivered from the buffer and, while it is not empty, transmits a data frame, managed by the counter. Normally, stays high in the absence of data flow Second, the sending module will transform the sending of parallel data into serial data sending by transmitting the content in accordance with the data's format specification.

### 2.3. Receiver Module

#### 2.3.1. Receiver timing configuration

Every time the value changes from "0" to "1" the beginning bit of a frame of data can be considered to have arrived. However, to guarantee the accuracy of the information shared between the communication partners, Logical "1" can potentially shift to logic "0" due to transmission line noise. In this paper RS 232 is used that allows parallel data for serial communication between devices over relatively short distance. When we refer UART with RS 232 we discuss UART communication with the RS-232 electrical standard. In this configuration. The UART is responsible for serializing and deserializing data (converting between parallel and serial data formats) and handling the timing aspects of communication. RS-232 specifies the voltage levels used for signaling. To represent binary data (1s and 0s) on the transmit and receive lines, it specifies both high and low voltage levels. Usually, the voltage levels fall between -15V and +15V, where negative voltages represent logical 1s and positive voltages represent logical 0s. The receiving eventually comes to an end, Both the state and completed signals are pulled down and up, respectively. Once every data bit has been input, the sampling counter will count [16].

## 3. Literature Survey

İn paper [1], showed the implementation of the UART in the Virtex II Pro FPGA processor because of its quick time to market, high speed, low cost, and repro-grammable nature.The FPGA's high speed capabilities and register-rich architecture make it the ideal UART implementation. The Xilinx ISE8.2 software programme was used to implement the design. Finally, Using the Xilinx xst tool for simulation and synthesis, the RTL schematic of the filtering IC was produced.Author of [2] Has designed UART based on FPGA Verilog was used throughout the entire design process, and Xilinx ISE Webpack 14.7 was used to make it visible in the Spartan-3E FPGA. The design can operate at the maximum Spartan-3E FPGA clock frequency of 218.248 MHz, according to test results. The UART controller can handle bits at a maximum frequency of 192.773 MHz, which is why it only requires a tiny amount of storage. Author of [18] Use System Verilog (SV) to develop and validate a complete du-plex UART module. Without the need of a clock signal, systems can communicate with one another thanks to this serial communication protocol. Data from parallel streams is converted to serial format and sent. There has been a lot of re-searches in the literature author Jiajing Li suggests an asynchronous FIFO buffered APB bus-based UART communication interface. With this design, the AMBA bus may be used to set the UART controller in a flexible manner for baud rate ad-justment, transmission bits, parity mode configurability, and other features. Additionally, asynchronous FIFO buffering is implemented to facilitate communication between low-speed UART devices and high-speed devices [19]. This study elaborates on numerous elements linked to UART transceiver and focuses on the theoretical foundation and application of UART. Since UART was first invented fifty years ago, UART transceiver technology has advanced steadily and is now widely applied in many spheres of life. UART transceivers are crucial in many facets of contemporary life, including healthcare and the Internet of Things. UART transceivers also improve the application of currently available technology [20]. In Paper [5], author is concerned with improvements in PDP and total power in relation to clock period changes. The Cadence NCSIM Simulator and Compiler is used for simulation and synthesis, and 45 and 90 nm GPDK library files are used for UART implementation. Now a days A popular technique for testing digital circuits is scan-based testing, which improves the controllability of the circuit being tested, according to the paper's author [8]. The suggested technique divides a scan chain into numerous scan segments, therefore reducing the amount of test data. Circuit topology is used to determine how to partition the scan chain, which im-proves the number of scan segments that can be skipped. According to the results of the simulation, the suggested approach lowers the test power usage [4].

## 4. Motivation and Contribution

While UART is a physical communication method used for serial transmission between devices, RS-232 is a standard that describes the electrical and physical characteristics of the serial communication interface. Overall, the objective of a UART with RS-232 project is to create a stable and efficient serial communication interface that can be used in a wide range of applications, from simple data transfer between two microcontrollers to more complex scenarios like device control, monitoring, and data acquisition [6].

**UART (Universal Asynchronous Receiver/Transmitter):** It is a hardware component, or a communication protocol used for serial communication between computers or microcontrollers and peripheral devices. It handles the conversion of parallel data (inside the computer) to serial data that can be transmitted over a communication line (like a wire) and vice versa. As UART connection is asynchronous, data is transmitted without the use of a shared clock signal. In order for communication to be successful, the sender and recipient must agree on the baud rate, or data rate, as well as the data bits, stop bits, and parity settings [7].

**RS-232 (Recommended Standard 232):** The electrical and mechanical properties of serial communication between devices are described by the RS-232 standard [8].

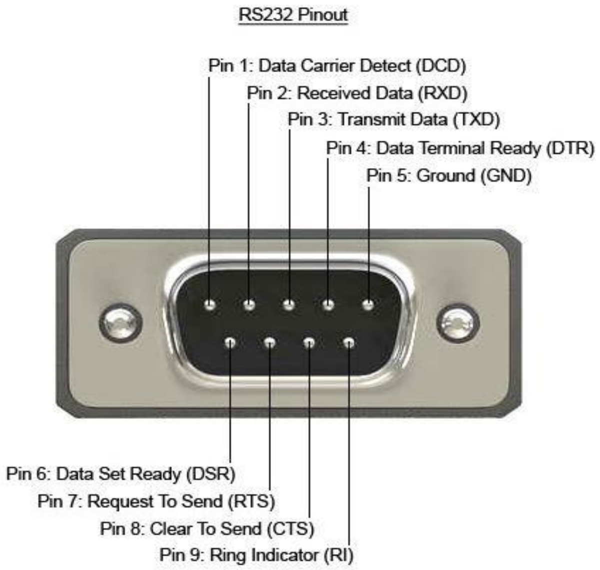

RS232 Pinout
Pin 1: Data Carrier Detect (DCD)
Pin 2: Received Data (RXD)
Pin 3: Transmit Data (TXD)
Pin 4: Data Terminal Ready (DTR)
Pin 5: Ground (GND)
Pin 6: Data Set Ready (DSR)
Pin 7: Request To Send (RTS)
Pin 8: Clear To Send (CTS)
Pin 9: Ring Indicator (RI)

**Figure 2 Pin Diagram of Rs232 [30]**

For serial communication, it specifies voltage levels, signal timing, and connection pin assignments. RS-232 is often used for connecting devices over short to moderate distances, typically with DB-9 or DB-25 connectors. The physical connections between devices might involve DB-9 or DB-25 connectors and cables with specific pin assignments for transmit (TX), receive (RX), ground (GND), and other control signals. The primary goal of this project is to create a modified UART architecture [9]. To do this, a 232-communication protocol must be implemented in the UART, and the results should be compared to the conventional UART method. comparison has been shown based on the factors like speed, delay, power consumption, number of IOBs, number of logic gates, etc. The main objective of a UART (Universal Asynchronous Receiver/Transmitter) with RS-232 project is typically to provide a communication interface between two electrical devices or systems using these protocols. My contribution to this work is that we use the RS 232 signaling proto-col to develop a novel architecture for a high-speed, low-power UART. We then compare the speed and power of this new architecture to those of the conventional UART that is currently in use [10] in Figure 2.

## 5. Organization

The paperwork is structured into several section. Section 2 provides the implementation of UART and literature survey. Section 3 provides motivation and my contribution in this project. section 4 is the organization of this paper. section 4 show system model and problem formulation. whereas section 6 has a simulation results section 7 discusses the results and section 8 provide the conclusion and future scope of this project [11].

## 6. Experiment Codes

### 6.1. UART Transmitter Module

Parallel data is available for transmission at [7:0] data can be parametric. The status is set at the elevated level, and r_data stores information from the data signal. when the input start signal shows high level. For every bit_flag high level, bit count rises by 1 in accordance with the baud rate setting. In r_data, the transmitter rs232_tx transmits serial data based on bit count [12]. A done signal is issued when the transmission is stopped by setting the state to low level when the bit count reaches 10 and the bit flag is at a high level [21-23].

```
module test_uart_tx;
// Inputs
reg clk = 0;
reg rst = 0;
reg [7:0] current_test = 0;
reg [7:0] s_axis_tdata = 8'd0;
reg s_axis_tvalid = 1'b0;
reg [15:0] prescale = 0;
wire s_axis_tready;
wire txd;
wire busy;
```

To avoid errors resulting from the fusion of two data sets, the data in data signal will be kept in r_data. A low state level denotes an idle state, a high level denotes a working state, and state signifies the working state [13]. The send flag is represented by bit_flag, the bit counter by bit_cnt, and the baud rate counter by baud_cnt.

```
always @ (posedge clk) begin
if (rst) begin
s_axis_tready_reg <= 0;
txd_reg <= 1;
prescale_reg <= 0;
bit_cnt <= 0;
busy_reg <= 0;
end else begin
if (prescale_reg > 0) begin
s_axis_tready_reg <= 0;
prescale_reg <= prescale_reg - 1;
end else if (bit_cnt == 0) begin
s_axis_tready_reg <= 1;
busy_reg <= 0;
```

The bit counter bit_cnt begins to count when the send flag bit_flag is high.Bit_cnt = 0 causes rs232_tx to transmit the initial bit (low level). Eight bits of r_data are sent from low to high by rs232_tx when bit_cnt is between 1 and 8. RS232_tx sends the stop bit (high level) when bit_cnt = 9 [24-25].

### 6.2. UART Receiver Module

The transmitting module's sending data, or Rs232, is also assumed to be the data that the receiving module needs to receive in order to confirm that the receiving module's design is accurate. Comparable to the transmission module, bit_cnt counts determine which order in which the receiving end, rx_data, gets the eight-bit data from rs232_t2, from low to high [26-27] in Figure 3, 4.

```
module uart_rx #
(
parameter DATA_WIDTH = 8
)
(
input wire                      clk,
input wire                      rst,
output wire [DATA_WIDTH-1:0] m_axis_tdata,
output wire                 m_axis_tvalid,
input wire                  m_axis_tready,
input wire                      rxd,
output wire                     busy,
output wire                 overrun_error,
output wire                 frame_error,
input wire [15:0]               prescale
);
```

Set the clock signal to 50MHz after entering it. Low-level reset signal is represent-ed by rst_n. After successfully receiving the signal, the receiving end (rx_data) generates the end receiving signal done [14].

```
always @ (posedge clk) begin
        if (rst) begin
        m_axis_tdata_reg <= 0;
        m_axis_tvalid_reg <= 0;
        rxd_reg <= 1;
        prescale_reg <= 0;
        bit_cnt <= 0;
        busy_reg <= 0;
        overrun_error_reg <= 0;
        frame_error_reg <= 0;
        end
        else begin
        rxd_reg <= rxd;
        overrun_error_reg <= 0;
        frame_error_reg <= 0;
```

Bit_cnt is the bit counter that begins to count when bit_flag, the receiving flag, is high. The eight-bit rs232_t2 data is received by rx_data from low to high when bit_cnt is between 1 and 8.

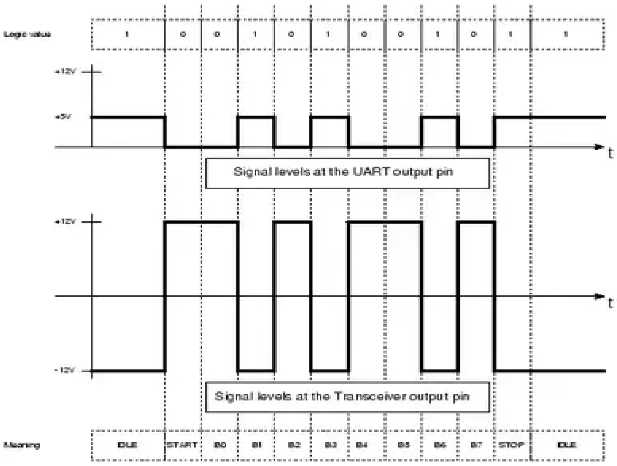


**Figure 3 Data Transmission of UART With Rs232 [29]**

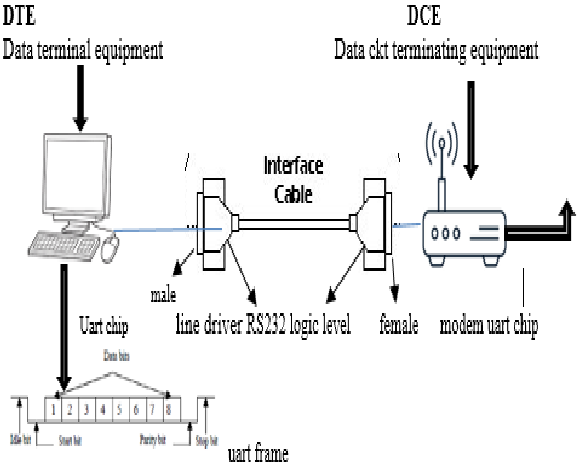
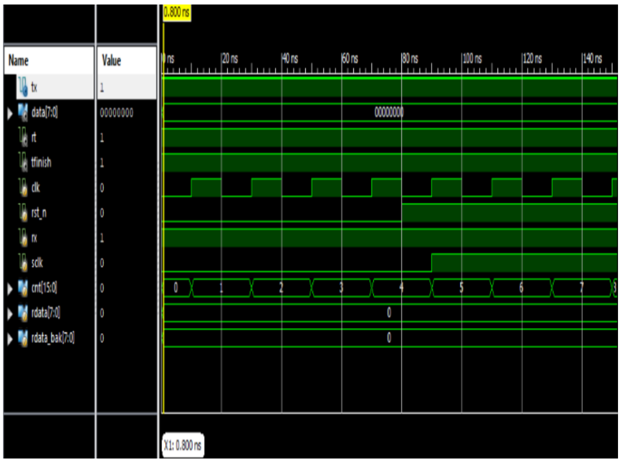


**Figure 4 Proposed Serial Data Exchange Between PC and Device Using RS232 Protocol**

## 7. Simulations Results

After synthesizing the top module of UART in Xilinx software we get the schematic of the waveform as shown in below fig 5 and implemented on FPGA artix 7.



| Name | Value |
|---|---|
| tx | 1 |
| data[7:0] | 00000000 |
| rt | 1 |
| tfinish | 1 |
| clk | 0 |
| rst_n | 0 |
| rx | 1 |
| sclk | 0 |
| cnt[15:0] | 0 |
| rdata[7:0] | 0 |
| rdata_bak[7:0] | 0 |

**Figure 5** **Waveform of The Proposed UART with Rs232**

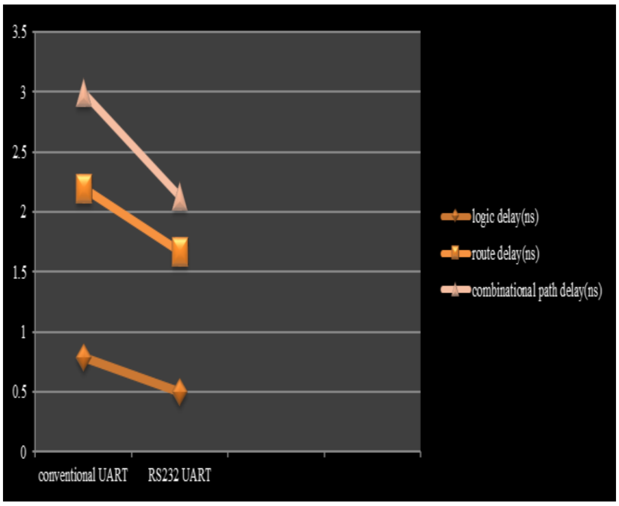


**Figure 6** **Comparison Graph of Delay**

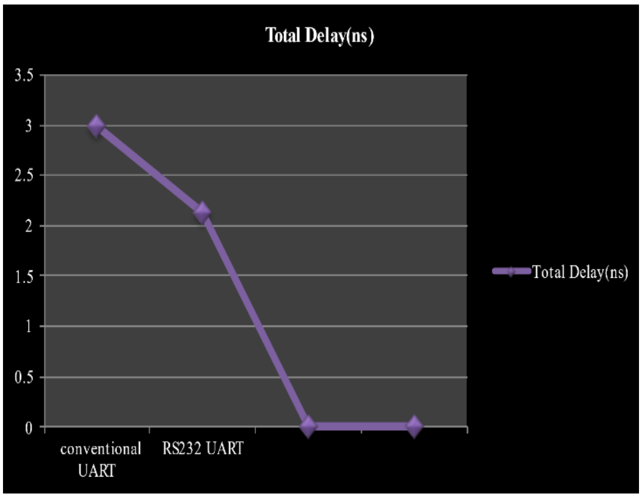


**Figure 7** **Overall Comparison Graph of Delay**

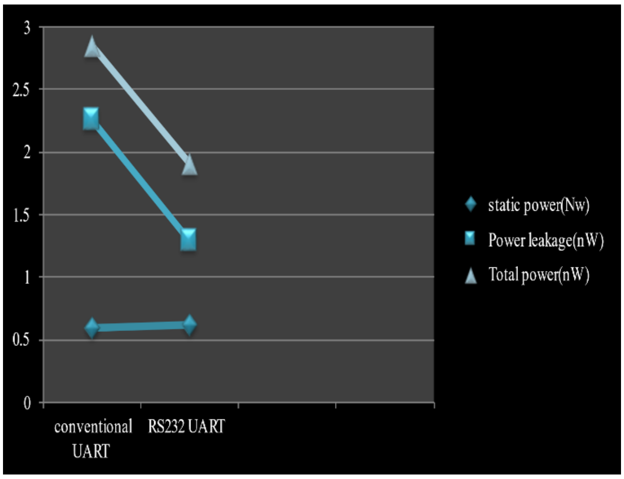


**Figure 8** **Power Consumption Graph**

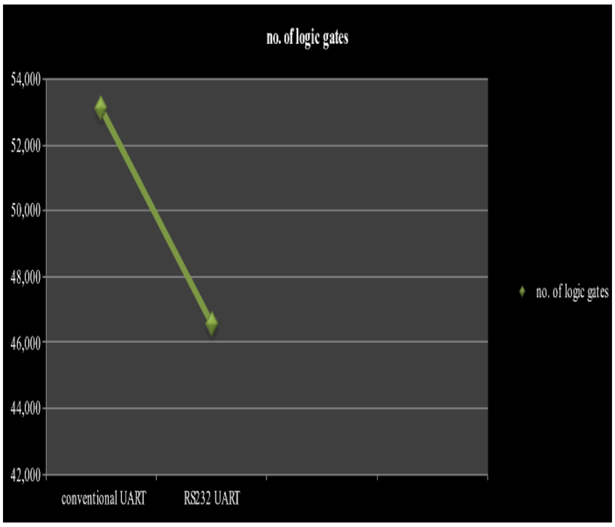


**Fıgure 9** **Graph of Number of Logic Gates Use**

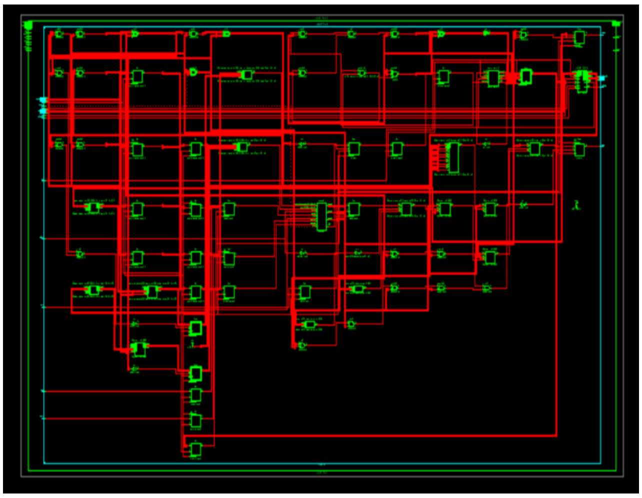


**Fıgure 10** **RTL View of Proposed Uart**

**Table 1 Performance Comparison of Existed and Proposed UART Design**

| PARAMETERS | CONVENTIONAL | UART WITH RS232 |
|---|---|---|
| **Logic delay(ns)** | 0.79 | 0.5 |
| **Route dela(ns)** | 2.2 | 1.67 |
| **Combinational delay delay(ns)** | 2.99 | 2.127 |
| **Total path delay (ns)** | 2.99 | 2.127 |
| **Static power (nW)** | 0.591 | 0.619 |
| **Leakage Power(nW)** | 2.26 | 1.293 |
| **Total power consumption(nW)** | 2.851 | 1.912 |
| **No of logic gates** | 53120 | 46560 |
| **No of flip flops** | 83 | 64 |

## 8. Result and Discussion

The above proposed work is detailed in Verilog HDL. Synthesis and implementation have been done using XILINX ISE 14.5 also simulation and waveform has been done on MODELSIM and the FPGA device used here is ARTIX -7. Fig7 shows the waveform of the proposed UART, where 8-bit data is transmitted and the other figure 5-10 shows the comparison analysis through graphical representation whereas comparison has been done on the basis of delay, power consumption, num. of logic gates use, num of IOBs etc. Table 1 illustrates the various above-mentioned parameters and show the comparison analysis between the conventional way of UART and the proposed UART design. from the above results we conclude that UART with RS232 transmit data more efficiently without any distortion in the signal data as compared to conventional way also conclude that proposed UART design consume less power and faster in speed as compared to the conventional way [28]. Proposed work also uses less num of logic gates and less IOBs which makes the overall speed of the UART faster. We also conclude that new proposed work successfully decreases the delay by 29% and power consumption by 33%. this can be possilble beacause of the RS232 communication which is relatively simple and doesn't require complex protocol handling or processing compared to some other communication standards. This can lead to lower power consumption in the sense that less computational power is required to manage communication. Also, it provides a wide voltage range which makes the transmission less prone to noise and distortion. so, in this paper we are trying to show the new implementation of UART design where we replace conventional way of UART and use a UART with RS232 and try to enhance the speed and lower the power consumption and we have done successfully. The results are shown in table 1.

## Conclusions

In this, we compare the implementation of UART design in terms of power, delay, No. of transistor, No. of bonded IOBS etc. and we conclude that the implementation of the proposed UART with RS232 gives the better result in terms of power, delay, and other parameters. It is achieved a 29% reduction in delay as compared to the Existing conventional UART and power consumption is reduced by 33%. if we compare the overall results, proposed work gives the better result and the proposed implementation comes with many features like wide range of voltage which makes transmission less prone to noise, less distortion, less complexity which leads to the lower power consumption. İn future we can use there are several alternatives to RS-232, for more better result we can use RS485 which is a serial communication standard that allows for longer cable lengths and greater noise immunity than RS-232. It is often used in industrial and automation settings. Other we can use USB (Universal Serial Bus) it is a widely adopted interface for connecting peripherals to computers. It offers higher data transfer rates than RS-232. RS-232 has been mainly displaced by more sophisticated communication standards like USB, Ethernet, Wi-Fi, Bluetooth, and others in the majority of modern applications, particularly those involving high-speed data transfer, long-distance communication, or wireless networking. The use of RS-232 will probably become even more specialized and restricted to a small number of specialist applications as technology develops.

## References

[1]. Umakanta Nanda, Sushant Kumar Pattnaik” Universal Asynchronous Receiver and Transmitter” 3rd International Conference on Advanced Computing and Communication Systems (ICACCS -2016), Jan. 22 – 23, 2016, Coimbatore, INDIA.

[2]. S. Keerthi Nayani, G. Nikhilesh, S. Shiva Tej Kumar “Desıgn and Implementatıon Ofuart Protocol For Avıonıcs Applıcatıons” International Conference on Intelligent Computing and Control Systems (ICICCS 2019) IEEE Xplore Part Number: CFP19K34-ART; ISBN: 978-1-5386-8113-8

[3]. R, Y., & V, R. M. Design and Verification of UART using System Verilog. International Journal of Engineering and Advanced Technology, 2020,9(5), 1208–1211.

[4]. Thai, N. P., Ngoc, B. H., Duy, T. D., Truong, P. Q., & Phan, V. C” A novel multichannel UART design with FPGA-based implementation” International Journal of Computer Applications in Technology, 2021, 67(4), 358.

[5]. Kamath, A., Mendez, T., Ramya, S., & Nayak, S.” Design and Implementation of Power-Efficient FSM based UART” Journal of Physics: Conference Series, 2022, 2161(1), 012052.

[6]. Keshav Kumar, Shabeer Ahmad, Bishwajeet Pandey, Amit K Pandit, Deepa Singh and Akbar Hussain D.M “Power Efficient Frequency Scaled and Thermal-Aware Control Unit Design on FPGA,” Int. J. of Innovative Technology and Exploring Engineering (IJITEE), 2019, 8(9 Special Issue 2), pp. 530-533. DOI: 10.35940/ijitee.I1111.0789S219.

[7]. Ashok Kumar Gupta, Ashish Raman, Naveen Kumar and Ravi Ranjan, “Design and Implementation of High-Speed Receiver and Transmitter (UART),” 2020 7th Int. Conf. On Signal Processing and Integrated Networks (SPIN), Noida, 27 - 28 Feb. 2020, pp. 295-300

[8]. H. Kim, H. Oh, S. Lee and S. Kang, "Low Power Scan Chain Architecture Based on Circuit Topology," 2018 International SoC Design Conference (ISOCC), 2018.

[9]. H. Woo, S. Jang and S. Kang, "A Secure Scan Architecture Protecting Scan Test and Scan Dump Using Skew-Based Lock and Key," in IEEE Access, 2021.

[10]. Kyunghwan Cho, Jihye Kim, Hyunggoy

Oh, Sangjun Lee, and Sungho Kang "A New Scan Chain Reordering Method for Low Power Consumption based on Care Bit Density" in IEEE International SoC Design Conference (ISOCC) 2019.

[11]. D Manasa Manikya, Marala Jagruthi1, Rana Anjum, Ashok Kumar K, "Design of Test Compression for Multiple Scan Chains Circuits" International Conference on System, Computation, Automation and Networking (ICSCAN), 2021.

[12]. Nagesh B, Nikhil Chandra B S, "Design of Efficient Scan Flip-Flop", 6th International Conference on RecentTrends on Electronics, Information, Communication & Technology (RTEICT), 2021.

[13]. Yucong Zhang, Xiaoqing Wen, Stefan Hols, Kohei Miyase, Seiji Kajihara, Hans-Joachim Wunderlich, and Jun Qian, "Clock-SkewAware Scan Chain Grouping for Mitigating Shift Timing Failures in Low-Power Scan Testing “, IEEE 27th Asian Test Symposium, 2018.

[14]. K. -J. Lee, C. -A. Liu and C. -C. Wu, "A Dynamic-Key Based Secure Scan Architecture for Manufacturing and In-Field IC Testing," in IEEE Transactions onEmerging Topics in Computing, vol. 10, no. 1, pp. 373-385, 1 Jan.-March, 2022.

[15]. Yucong Zhang, Xiaoqing Wen, Stefan Hols, Kohei Miyase, Seiji Kajihara, Hans-Joachim Wunderlich, and Jun Qian, "Clock-SkewAware Scan Chain Grouping for Mitigating Shift Timing Failures in Low-Power Scan Testing “, IEEE 27th Asian Test Symposium, 2018.

[16]. V. Shivakumar, C. Senthilpari and Z. Yusoff, "A Low-Power and Area-Efficient Design of a Weighted Pseudorandom Test-Pattern Generator for a Test-Per-Scan Built-in SelfTest Architecture," in IEEE Access, vol. 9, pp. 29366-29379, 2021.

[17]. G. Maanasa, K. Raghava Rao, L. Anjali, P. Satyannarayana”An Inter-active GPS And RFID Based Receptacle Security System” International Journal of Engineering and Advanced Technology (IJEAT) ISSN: 2249-8958, Volume-8 Issue-4, April 2019.

[18]. Yamini R”Design and Verification of UART using System Verilog” International Journal of Engineering and Advanced Technology (IJEAT) ISSN: 2249 – 8958 (Online), Volume-9 Issue-5, June 2020.

[19]. Jiajing Li; Lixin Yang “UART Controller with FIFO Buffer Function Based on APB Bus” IEEE International Workshop on Anti-counterfeiting, Security, Identification, ASID 2022.

[20]. Hang Zhang “A Systematic Analysis of The UART Transceiver Theory and Application” Science,engineering and technology vol 61 ,IFMPT 2023.

[21]. W. Zhang, Y. Hu, R. Ding, and B. Yang, “Research on High-Speed Asyn-chronous Serial Transmission Based on 8b10b,” Applied Informatics and Com-munication, pp. 586 – 592, Aug. 2011.

[22]. D. Bhadra, V. S. Vij, and K. S. Stevens, “A low power UART design based on asynchronous techniques,” IEEE Xplore, Aug. 01, 2013.

[23]. K. Gupta, A. Raman, N. Kumar, and R. Ranjan, “Design and Implementa-tion of High-Speed Universal Asynchronous Receiver and Transmitter (UART),” IEEE Xplore, Feb. 01, 2020.

[24]. E. Xie and J. Zhou, “Analysis and Comparison of Asynchronous FIFO and Synchronous FIFO,” 2023 IEEE 2nd International Conference on Electrical En-gineering, Big Data and Algorithms (EEBDA), Feb. 2023.

[25]. P. Sharma, A. Kumar, and N. Kumar, “Analysis of UART Communication Protocol,” 2022 International Conference on Edge Computing and Applications (ICECAA), Oct. 2022.

[26]. B. K. Dakua, M. S. Hossain, and F. Ahmed, “Design and Implementation of UART Serial Communication Module Based on

FPGA," International Confer-ence on Materials, Electronics & Information Engineering, ICMEIE-2015, Jan. 2015.

[27]. Gigantea, V. Mudeng, H. S. Natiand, and M. Izzudin Abdillah Afif, "Microcontroller Serial Communication to Analyze Bit Characters," IEEE Xplore, Dec. 01, 2018.

[28]. E. Peňa and M. G. Legaspi, "UART: A Hardware Communication Pro-tocol Understanding Universal Asynchronous Receiver/Transmitter | Analog Devices," www.analog.com, Sep. 2020. https://www.analog.com/en/analog-dialogue/articles/uart-a-hardware-communication-protocol.html.

[29]. V. Ryabchevsky and G. Nikonova, "Development of Human Mashine Inter-face for an Electromyogram Recorder," 2019 International Multi-Conference on Engineering, Computer and Information Sciences (SIBIRCON), Oct. 2019.

[30]. D. Elbaeva, K. M. Senov, Y. V. Bolgov, and A. D. Elbaev, "Modeling of the Instrument-Computer System for the Diagnostics of Atherosclerosis," 2020 International Conference Quality Management, Transport and Information Se-curity, Information Technologies (IT&QM&IS), Sep. 2020.

[31]. R. R. Pahlavi, A. G. Putrada S., and M. Abdurohman, "Fast UART and SPI Protocol for Scalable IoT Platform," 2018 6th International Conference on Information and Communication Technology (ICoICT), May 2018.

[32]. L. V. Bogdanov, "A Wi-Fi to UART Bridge for Firmware Updates of Mi-crocontrollers," 2019 IEEE XXVIII International Scientific Conference Electron-ics (ET), Sep. 2019.

[33]. B. Jeevan, P. Sahithi, P. Samskruthi, and K. Sivani, "Simulation and synthesis of UART through FPGAZedboard for IoT applications," 2022 International Conference on Advances in Computing, Communication and Applied Infor-matics (ACCAI), Jan. 2022.

[34]. M. A. Lotufo, L. Colangelo, C. Perez-Montenegro, E. Canuto, and C. Novara, "UAV quadrotor attitude control: An ADRC-EMC combined approach," Control Engineering Practice, vol. 84, pp. 13 – 22, Mar. 2019.

[35]. S. Keerthi Nayani, G. Nikhilesh, and S. S. Tej Kumar, "Design and Imple-mentation of Uart Protocol for Avionics Applications," 2019 International Con-ference on Intelligent Computing and Control Systems (ICCS), May 2019.

[36]. D. Chen, G. Cao, J. Wang, and H. Fan, "Research on Key Technologies of Detecting 1553B Avionics Data Bus Network," Defence Technology, vol. 9, no. 3, pp. 176 – 180, Sep. 2013.

[37]. Choi, J. I., Jain, M., Srinivasan, K., Levis, P., & Katti, S." single channel, full duplex wireless communication" Proceedings of the Sixteenth Annual Interna-tional Conference on Mobile.