



e-ISSN: 2278-8875  
p-ISSN: 2320-3765

# International Journal of Advanced Research

in Electrical, Electronics and Instrumentation Engineering

Volume 15, Issue 5, May 2026



Impact Factor: 9.867

9940 572 462

6381 907 438

ijareeie@gmail.com

www.ijareeie.com



# Design & Verification of UART Communication using UVM

**Mr. Vallabh Karnawadi, Mrs. Roopa T C, Dr. Aijaz Ahamed Sharief**

M. Tech, Dept. of VLSI Design & Embedded Systems, Shridevi Institute of Engineering and Technology, Tumakuru, India

Assistant Professor, Dept. of ECE, Shridevi Institute of Engineering and Technology, Tumakuru, India

Professor & HOD, Dept. of ECE, Shridevi Institute of Engineering and Technology, Tumakuru, India

**ABSTRACT:** UART is a widely used serial communication protocol in embedded and digital systems for transferring data using fewer wires. It supports asynchronous communication, where the transmitter and receiver operate using a common baud rate instead of a shared clock signal. In this project, UART transmitter (Tx) and receiver (Rx) modules are designed for serial data communication. The transmitter converts parallel data into serial form, while the receiver converts the received serial data back into parallel data. A baud rate generator provides timing signals for accurate transmission and reception. UART communication includes a start bit, data bits, optional parity bit, and stop bit. The receiver detects the start bit, samples incoming data, and reconstructs the original information. Verification and simulation are performed to ensure correct data transmission, parity checking, and reliable communication.

## I. INTRODUCTION

In modern digital and embedded systems, communication between devices is essential for exchanging data and controlling operations. Microcontrollers, sensors, computers, and other electronic components often need an efficient method to transfer information reliably. One of the most widely used communication techniques for this purpose is serial communication, where data is transmitted sequentially over a single communication line, reducing wiring complexity and hardware requirements. Among the various serial communication protocols, the Universal Asynchronous Receiver/Transmitter (UART) protocol is one of the simplest and most commonly implemented interfaces. UART enables asynchronous communication between devices without requiring a shared clock signal. Instead, it relies on predefined baud rates, start bits, stop bits, and optional parity bits to ensure proper synchronization and error detection during data transmission. UART is extensively used in embedded systems for communication between microcontrollers, computers, GPS modules, Bluetooth devices, and many other peripherals. Due to its simplicity, reliability, and low hardware overhead, it is often implemented directly in hardware using digital logic design techniques. This project focuses on the design and verification of UART transmitter (Tx) and receiver (Rx) modules. The design includes key components such as baud rate generation, serial data transmission logic, and reception mechanisms that correctly interpret incoming serial data. Proper verification is performed to ensure accurate data transfer, correct frame structure, and reliable handling of start bits, stop bits, and parity bits. Through this implementation, the project demonstrates the fundamental working principles of UART communication and highlights its importance in digital system design and embedded applications.

## II. LITERATURE SURVEY

Various research works have been carried out in the field of UART communication focusing on design optimization, high-speed data transfer, low-power operation, and functional verification. The following studies provide important contributions related to UART transmitter and receiver implementation, FPGA realization, and verification methodologies.

1. Kumar et al. (2019) designed a UART transmitter and receiver using Verilog HDL on FPGA platforms. Their work focused on modular RTL architecture, resulting in efficient and reliable serial communication with reduced hardware complexity.
2. Patel and Mehta (2020) proposed a high-speed UART design using FPGA implementation and 16x oversampling techniques. The design improved timing accuracy and minimized communication errors during high-speed data transfer.



3. Zhang et al. (2021) developed a low-power UART architecture for embedded systems using clock gating techniques. Their approach significantly reduced dynamic power consumption while maintaining reliable communication performance.
4. Reddy et al. (2020) presented a SystemVerilog-based functional verification methodology for UART communication. By using assertions and coverage-driven verification, the design achieved improved reliability and ensured correct UART protocol functionality.

From the reviewed literature, it is observed that UART communication systems can be enhanced through efficient RTL design, FPGA implementation, power optimization techniques, and advanced verification methodologies. These studies provide a strong foundation for the proposed work on the design and verification of UART communication systems

### III. UART PROTOCOL OVERVIEW

UART is an asynchronous serial communication protocol where transmission and reception occur without a common clock signal. Both communicating devices are configured with the same baud rate. A UART frame typically consists of:

- Start bit
- Data bits (5–8 bits)
- Optional parity bit
- Stop bits (1 or 2 bits)

The implemented UART design supports configurable parameters:

- Baud rates: 4800, 9600, 19200, 38400, 57600, 115200
- Data lengths: 5–8 bits
- Even/Odd parity
- Single or double stop bits

### IV. PROPOSED UART DESIGN

The UART DUT consists of three major modules:

#### 4.1 Clock Generator

The clock generator produces transmission and reception clocks (tx\_clk and rx\_clk) based on the selected baud rate. Counters divide the system clock to generate accurate timing signals.

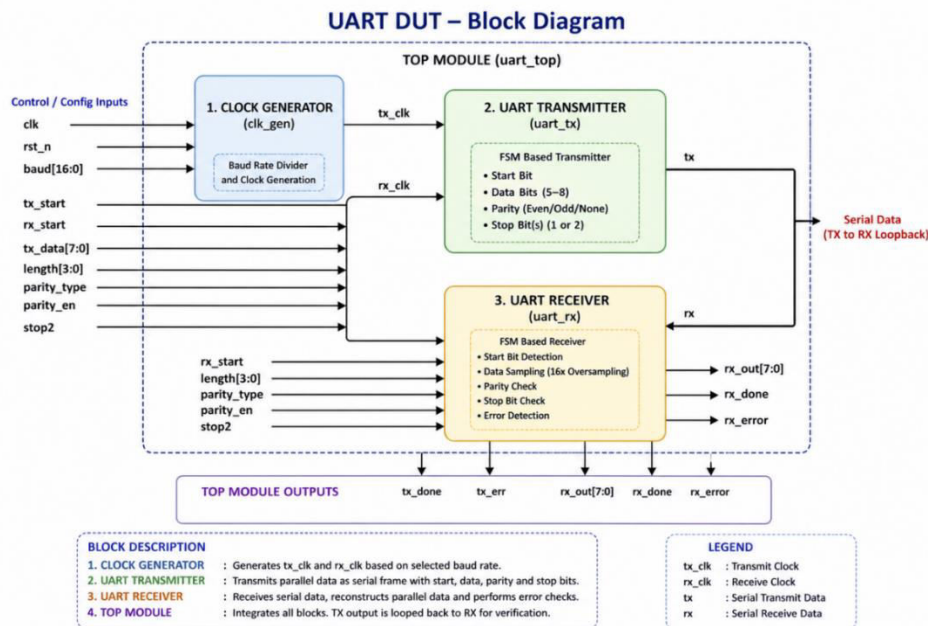


Fig.4.1 Block Diagram of UART DUT



4.2 UART Transmitter

The transmitter converts parallel input data into serial format using an FSM-based design. The transmission process includes:

- Start bit generation
- Serial data transmission
- Parity generation
- Stop bit transmission

The FSM states include:

- Idle
- Start Bit
- Send Data
- Send Parity
- Stop Bit
- Done State

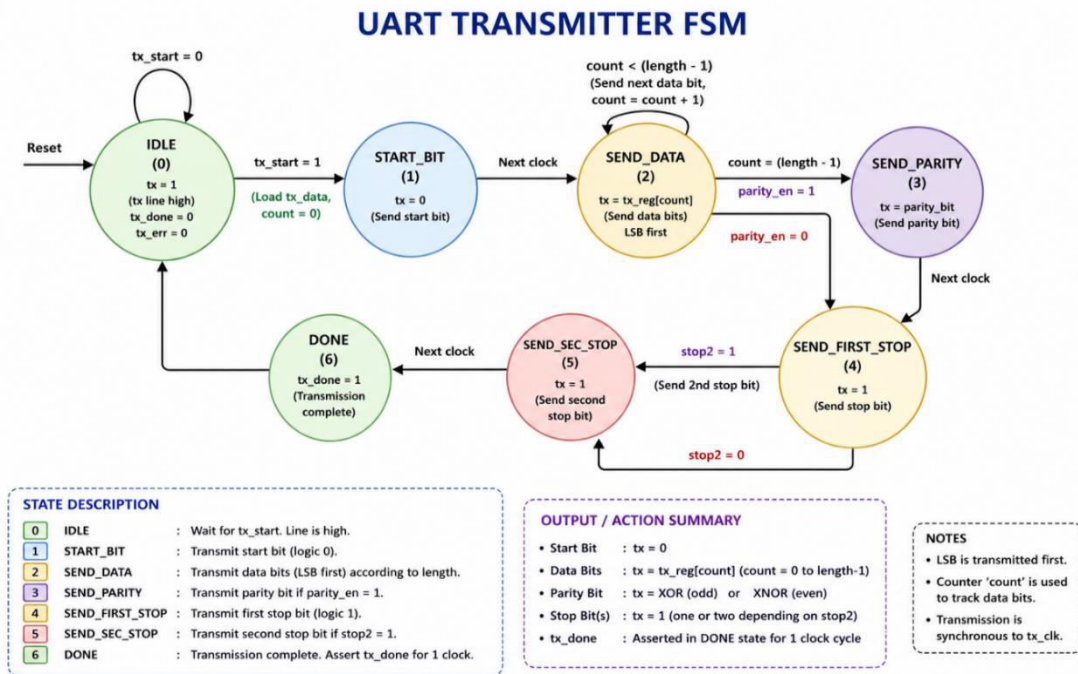


Fig.4.2 Block Diagram of UART TRANSMITTER FSM

4.3 UART Receiver

The receiver reconstructs serial data into parallel form using oversampling techniques. The receiver performs:

- Start bit detection
- Data sampling
- Parity verification
- Stop bit checking
- Error detection

The receiver FSM includes:

- Idle
- Start Detection
- Receive Data
- Check Parity
- Stop Bit Verification
- Done State

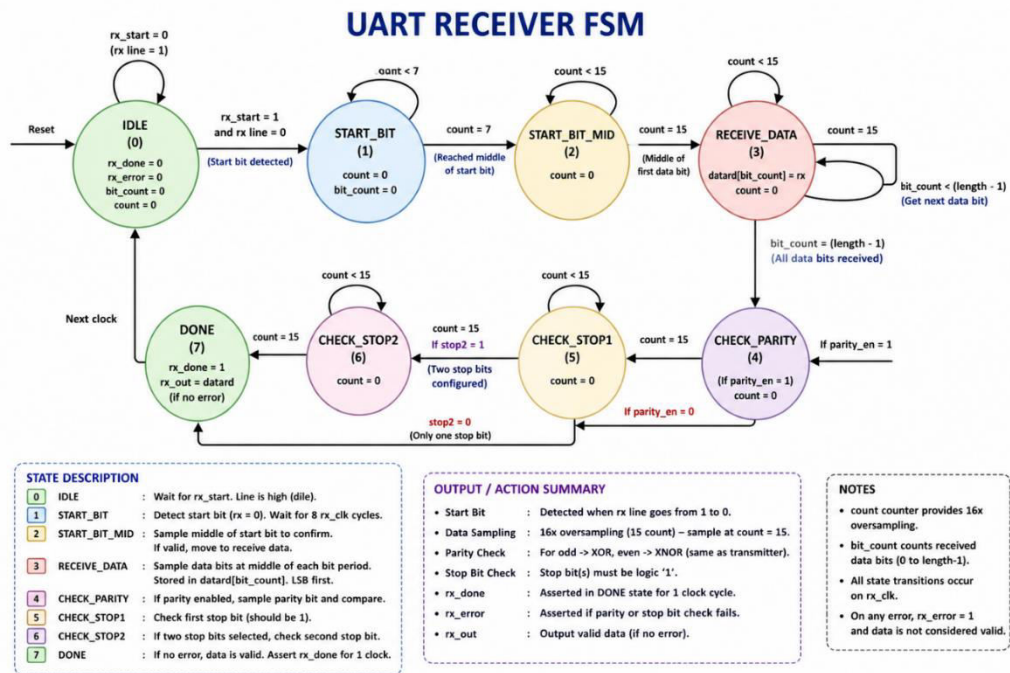


Fig.4.3 Block Diagram of UART RECEIVER FSM

V. VERIFICATION USING UVM

To ensure functional correctness, a UVM-based verification environment is developed. UVM provides reusable and scalable verification components such as:

- Driver
- Monitor
- Sequencer
- Scoreboard
- Environment
- Test Classes

5.1 UVM Testbench Architecture

The verification environment uses constrained random testing and functional checking. The scoreboard compares transmitted data (tx\_data) with received data (rx\_out) to determine pass/fail conditions.

5.2 Verification Scenarios

The following test cases are implemented:

- Random baud rate testing
- Variable data length testing
- Parity enabled/disabled testing
- Single and double stop bit testing
- Continuous transmission testing
- Reset condition testing
- Unsupported baud rate testing

VI. RESULTS AND DISCUSSION

Simulation results confirmed correct UART functionality across all supported configurations. The scoreboard consistently reported successful data matching between transmitted and received data.



Snapshots of results:

```
# KERNEL: ** Report counts by severity
# KERNEL: UVM_INFO : 43
# KERNEL: UVM_WARNING : 0
# KERNEL: UVM_ERROR : 0
# KERNEL: UVM_FATAL : 0
```

```
UVM_INFO @ 0: reporter [RNTST] Running test test_uart_rand_baud_with_stop...
UVM_INFO /home/runner/monitor.sv(66) @ 0: uvm_test_top.e.age.mon [MON] Monitor build_phase completed
UVM_INFO /home/runner/monitor.sv(70) @ 0: uvm_test_top.e.age.mon [MON] Monitor run_phase started
UVM_INFO /home/runner/driver.sv(16) @ 0: uvm_test_top.e.age.drv [DRV] System Reset : start of Simulation
UVM_INFO /home/runner/scoreboard.sv(85) @ 1198305: uvm_test_top.e.sb [SB] PASS: DATA match. EXP=0x122 ACT=0x122 (BAUD:9600 LEN:8 PAR_EN:0 PAR_T:1 STOP2:1)
UVM_INFO /home/runner/scoreboard.sv(85) @ 2657105: uvm_test_top.e.sb [SB] PASS: DATA match. EXP=0x47 ACT=0x47 (BAUD:9600 LEN:8 PAR_EN:1 PAR_T:0 STOP2:1)
UVM_INFO /home/runner/scoreboard.sv(85) @ 2900705: uvm_test_top.e.sb [SB] PASS: DATA match. EXP=0x20 ACT=0x20 (BAUD:57600 LEN:8 PAR_EN:1 PAR_T:1 STOP2:1)
UVM_INFO /home/runner/scoreboard.sv(85) @ 3239745: uvm_test_top.e.sb [SB] PASS: DATA match. EXP=0x71 ACT=0x71 (BAUD:38400 LEN:8 PAR_EN:0 PAR_T:0 STOP2:1)
UVM_INFO ./uvm-1.2/src/base/uvm_objection.svh(1271) @ 3239745: reporter [TEST_DONE] 'run' phase is ready to proceed to the 'extract' phase
UVM_INFO /home/runner/scoreboard.sv(112) @ 3239745: uvm_test_top.e.sb [SB] SCOREBOARD SUMMARY total=4 pass=4 fail=0 skipped=0
UVM_INFO ./uvm-1.2/src/base/uvm_report_server.svh(869) @ 3239745: reporter [UVM/REPORT/SERVER]
```

```
UVM_INFO @ 0: reporter [RNTST] Running test test_uart_rand_baud_len5wp...
UVM_INFO /home/runner/monitor.sv(66) @ 0: uvm_test_top.e.age.mon [MON] Monitor build_phase completed
UVM_INFO /home/runner/monitor.sv(70) @ 0: uvm_test_top.e.age.mon [MON] Monitor run_phase started
UVM_INFO /home/runner/driver.sv(16) @ 0: uvm_test_top.e.age.drv [DRV] System Reset : start of Simulation
UVM_INFO /home/runner/scoreboard.sv(85) @ 885705: uvm_test_top.e.sb [SB] PASS: DATA match. EXP=0x29 ACT=0x29 (BAUD:9600 LEN:5 PAR_EN:1 PAR_T:1 STOP2:0)
UVM_INFO /home/runner/scoreboard.sv(85) @ 2031905: uvm_test_top.e.sb [SB] PASS: DATA match. EXP=0x23 ACT=0x23 (BAUD:9600 LEN:5 PAR_EN:1 PAR_T:0 STOP2:1)
UVM_INFO /home/runner/scoreboard.sv(85) @ 2223305: uvm_test_top.e.sb [SB] PASS: DATA match. EXP=0x10 ACT=0x10 (BAUD:57600 LEN:5 PAR_EN:1 PAR_T:1 STOP2:1)
UVM_INFO /home/runner/scoreboard.sv(85) @ 2484105: uvm_test_top.e.sb [SB] PASS: DATA match. EXP=0x3 ACT=0x3 (BAUD:38400 LEN:5 PAR_EN:1 PAR_T:0 STOP2:0)
UVM_INFO ./uvm-1.2/src/base/uvm_objection.svh(1271) @ 2484125: reporter [TEST_DONE] 'run' phase is ready to proceed to the 'extract' phase
UVM_INFO /home/runner/scoreboard.sv(112) @ 2484125: uvm_test_top.e.sb [SB] SCOREBOARD SUMMARY total=4 pass=4 fail=0 skipped=0
UVM_INFO ./uvm-1.2/src/base/uvm_report_server.svh(869) @ 2484125: reporter [UVM/REPORT/SERVER]
```

```
UVM_INFO @ 0: reporter [RNTST] Running test test_uart_parity_check...
UVM_INFO /home/runner/monitor.sv(66) @ 0: uvm_test_top.e.age.mon [MON] Monitor build_phase completed
UVM_INFO /home/runner/monitor.sv(70) @ 0: uvm_test_top.e.age.mon [MON] Monitor run_phase started
UVM_INFO /home/runner/driver.sv(16) @ 0: uvm_test_top.e.age.drv [DRV] System Reset : start of Simulation
UVM_INFO /home/runner/scoreboard.sv(85) @ 1198305: uvm_test_top.e.sb [SB] PASS: DATA match. EXP=0x248 ACT=0x248 (BAUD:9600 LEN:8 PAR_EN:1 PAR_T:0 STOP2:0)
UVM_INFO ./uvm-1.2/src/base/uvm_objection.svh(1271) @ 1198325: reporter [TEST_DONE] 'run' phase is ready to proceed to the 'extract' phase
UVM_INFO /home/runner/scoreboard.sv(112) @ 1198325: uvm_test_top.e.sb [SB] SCOREBOARD SUMMARY total=1 pass=1 fail=0 skipped=0
UVM_INFO ./uvm-1.2/src/base/uvm_report_server.svh(869) @ 1198325: reporter [UVM/REPORT/SERVER]
```

Fig.6.1 Verification data

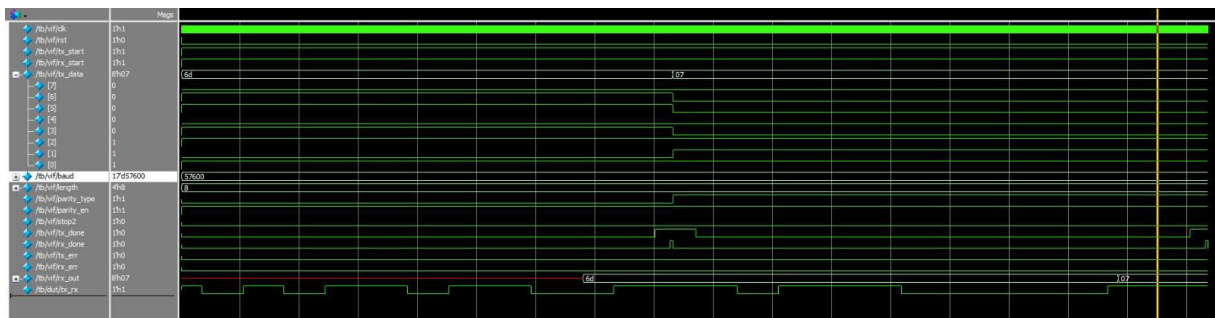


Fig.6.2 Sequence: rb, transactions 2, tr.baud = 57600

The given waveform shows UART transmission and reception for randomized data values. The tx\_data field provides the byte to be transmitted, while tx\_rx represents the serial UART line. A UART frame begins with a start bit, followed by 8 data bits, parity bit, and stop bit. After frame transmission completes, tx\_done is asserted. The receiver reconstructs the transmitted byte and updates rx\_out, after which rx\_done is asserted. The equality of tx\_data and rx\_out confirms successful UART communication, while tx\_err and rx\_err remain low, indicating error-free operation.

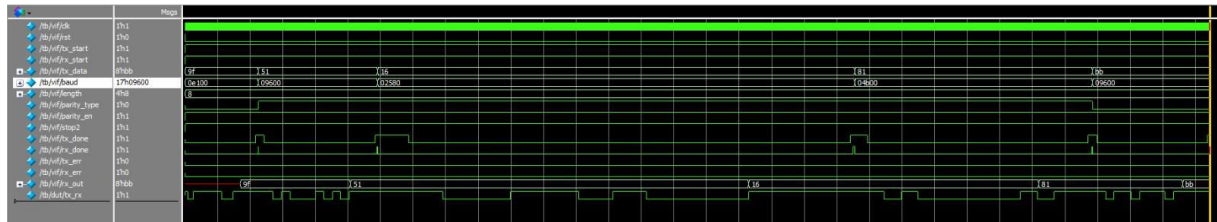


Fig.6.3 Sequence: rand\_baud\_with\_stop

For configuration: Start + 8 Data + Parity + Stop1 + Stop2 = 1 8 1 1 1



Fig.6.4 Sequence: rand\_baud\_len5p

UART frame format becomes : Start + 5 Data Bits + Parity + Stop = 1 5 1 1  
 Start + 8 Data + Parity + Stop = 11 bits

The following observations were made:

- Correct synchronization for all baud rates
- Accurate parity generation and checking
- Reliable stop bit handling
- No data loss during continuous transmission
- Proper reset behavior
- Correct error detection functionality

The implemented UVM testbench successfully verified:

- Data transmission accuracy
- FSM state transitions
- Error handling mechanisms
- Functional correctness under random conditions

### VII. APPLICATIONS

The UART communication system can be used in:

- Embedded systems
- IoT devices
- Processor debugging interfaces
- Bluetooth communication
- GPS modules
- Industrial automation
- SoC communication interfaces

### VIII. CONCLUSION

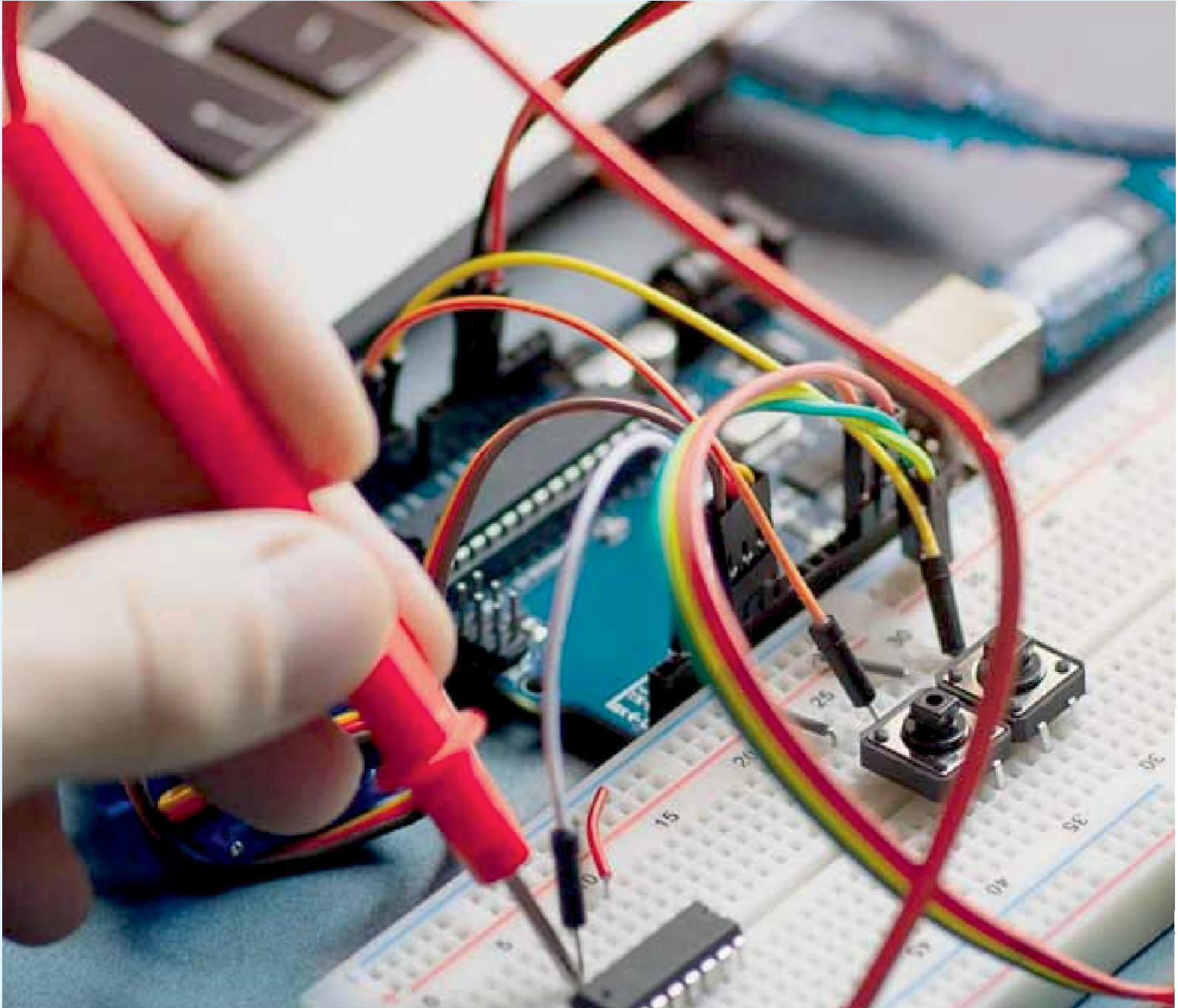
This work successfully demonstrates the design and verification of a configurable UART communication system using Verilog HDL and UVM. The UART DUT supports multiple baud rates, configurable frame formats, parity generation, and error detection mechanisms. FSM-based transmitter and receiver architectures ensure reliable serial communication.



The UVM verification environment validates the design using reusable and scalable components. Multiple test cases confirm accurate data transfer and robust operation under different configurations. The project provides a strong foundation in RTL design and advanced verification methodologies used in modern VLSI systems.

#### REFERENCES

- [1] A. Kumar, S. Verma, and R. Gupta, “Design and Implementation of UART Using Verilog HDL,” IEEE Int. Conf. on Communication Systems, pp. 120–124, 2019.
- [2] S. Patel and R. Mehta, “FPGA Based High-Speed UART Design and Verification,” IEEE Int. Conf. on VLSI Design, pp. 45–49, 2020.
- [3] M. Zhang, Y. Liu, and H. Wang, “Low Power UART Design for Embedded Systems,” IEEE Trans. on VLSI Systems, vol. 28, no. 6, pp. 1452–1460, 2021.
- [4] J. Lee, K. Park, and S. Kim, “Design of UART Controller with FIFO Buffer,” Int. J. Embedded Systems, vol. 12, no. 2, pp. 89–95, 2019.
- [5] K. Reddy, M. Rao, and V. Prasad, “Functional Verification of UART Using SystemVerilog,” Int. Conf. on Advanced Computing, pp. 233–237, 2020.
- [6] L. Wang and X. Zhou, “Design Optimization of UART for High Reliability Systems,” Int. J. Electronic Systems, vol. 18, no. 3, pp. 200–207, 2022.



INNO  SPACE  
SJIF Scientific Journal Impact Factor

  
doi<sup>®</sup>  
cross ref

 INTERNATIONAL  
STANDARD  
SERIAL  
NUMBER  
INDIA



# International Journal of Advanced Research

in Electrical, Electronics and Instrumentation Engineering

 9940 572 462  6381 907 438  [ijareeie@gmail.com](mailto:ijareeie@gmail.com)



[www.ijareeie.com](http://www.ijareeie.com)

Scan to save the contact details