# Design of UART-Transmitter Using FSM in Verilog HDL

**Ahalya S[1], Alagulakshmi P[2], Jayasri P[3], Maha lakshmi S[4], Dr.R.Sudha[5]**
[1, 2, 3, 4, 5] Dept of Electronics and Communication Engineering
[1, 2, 3, 4, 5] Bachelor of Engineering,Sri Shakthi Institute of Engineering and Technology (Autonomous) ,Coimbatore-641 062.

***Abstract-*** *This project presents the design and implementation of a UART (Universal Asynchronous Receiver/Transmitter) transmitter using a Finite State Machine (FSM) in Verilog HDL. UART is a widely used serial communication protocol that enables data transfer between devices without the need for a shared clock signal. The proposed system converts parallel input data into a serial data stream by following the standard UART frame format, which includes a start bit, data bits, and a stop bit.The design utilizes an FSM to control the sequential transmission process through different states such as IDLE, START, DATA, and STOP. A shift register and bit counter are used to ensure correct data sequencing and timing. The implementation is verified using a testbench and simulated to observe correct waveform behavior.This project provides a simple and efficient approach for implementing UART communication in digital systems and can be extended to include features such as parity checking, baud rate generation, and full-duplex communication for real-time embedded applications.*

***Keywords:*** UART, Finite State Machine (FSM), Verilog HDL, Serial Communication, Parallel-to-Serial Conversion, Digital Design, Data Transmission.

## I. INTRODUCTION

The UART (Universal Asynchronous Receiver/Transmitter) is a widely used serial communication protocol that enables data transfer between digital systems without requiring a common clock signal. This project focuses on the design and implementation of a UART transmitter using a Finite State Machine (FSM) in Verilog HDL.In this system, parallel data is given as input and is converted into a serial data stream for transmission. The UART transmitter follows a predefined communication format consisting of a start bit, data bits, and a stop bit. The transmission process is controlled using an FSM, which ensures proper sequencing and timing of each bit during communication.The FSM operates through different states such as IDLE, START, DATA, and STOP. Initially, the system remains in the idle state until a transmission request is received. Once triggered, the transmitter sends a start bit, followed by 8-bit data (least significant bit first), and finally a stop bit to indicate the end of transmission.To ensure correct operation, the design includes components such as a shift register for serial data transmission and a bit counter to track the number of bits transmitted. The system is verified using a testbench, and simulation results confirm the correct working of the UART protocol.This project provides a basic understanding of serial communication and FSM-based digital design. It serves as a foundation for more advanced systems such as full UART communication (transmitter and receiver), baud rate generation, and embedded communication interfaces used in real-time applications.

**Objective:**

1. To design a UART transmitter using Verilog HDL.This project focuses on creating a UART transmitter module.Verilog HDL is used for hardware description and implementation.It helps in modeling digital circuits effectively.
2. To implement serial communication without a shared clock.UART allows data transfer without using a common clock signal.It uses asynchronous communication between devices.This makes it simple and widely applicable in systems.
3. To convert parallel data into serial data.The system takes multiple bits as parallel input.These bits are converted into a serial data stream.This process is essential for UART communication.
4. To develop a Finite State Machine (FSM) for control.FSM is used to manage different transmission stages.States like IDLE, START, DATA, and STOP are defined.It ensures proper sequence of data transmission.
5. To define the UART frame format.The frame includes start bit, data bits, and stop bit.Each part plays an important role in communication.This structure ensures accurate data transfer.
6. To use a shift register for data handling.A shift register is used to send data bit by bit.It shifts the data sequentially during transmission.This helps in smooth serial output generation.
7. To implement a bit counter for timing control.The bit counter keeps track of transmitted bits.It ensures

correct number of bits are sent.This improves accuracy in communication.

8. To verify the design using simulation.A testbench is created to test the design.Simulation helps observe waveform outputs.It confirms correct working of the system.
9. To achieve efficient and reliable data transmission.The design aims for stable and error-free output.It ensures proper communication between devices.

## II. LITERATURE SURVEY

UART is a widely used asynchronous serial communication protocol known for its simplicity and low cost. It enables data transfer between devices without requiring a shared clock.Verilog HDL is commonly used for UART design, especially in FPGA implementations. It allows easy modeling, simulation, and efficient hardware realization.

Many studies use a Finite State Machine (FSM) to control UART transmission. States like IDLE, START, DATA, and STOP ensure proper data flow. UART designs include components such as baud rate generators, shift registers, and counters. These ensure correct timing and accurate data transmission.Recent research focuses on optimizing UART designs for better speed, lower power consumption, and reduced hardware area, especially in FSM-based systems.Advanced UART designs include features like error detection, full-duplex communication, and configurable settings, making them suitable for modern embedded systems.

## III. METHODOLOGY

The methodology for this project begins with defining the UART transmitter specifications such as data length, stop bits, and baud rate. A Finite State Machine (FSM) is then designed with states like IDLE, START, DATA, and STOP to control the transmission process. The system is implemented in Verilog HDL by developing modules for the shift register and bit counter to handle parallel-to-serial data conversion and bit tracking. The FSM manages the sequence of data transmission according to the UART frame format. All modules are integrated to form the complete transmitter design. A testbench is created to apply input signals and validate functionality. Finally, the design is simulated, and the output waveforms are analyzed to ensure correct and reliable serial data transmission.

**EXISTING SYSTEM:**

The existing system for UART communication typically relies on pre-built hardware modules or microcontroller-based UART peripherals. These systems provide basic serial communication by converting parallel data into serial format using fixed internal logic. However, they offer limited flexibility for customization and optimization, as their design is predefined and not easily modifiable. In many cases, the internal working of the UART module is not visible to the user, making it difficult to study or improve the transmission process. Additionally, such systems may not efficiently support specific design requirements like custom baud rate generation or enhanced control over data transmission. Hence, there is a need for a more flexible and transparent design approach using Verilog HDL and FSM techniques.

**Disadvantages of Existing System:**

- Limited flexibility due to fixed hardware design.Existing UART modules come with predefined structures.Users cannot easily change their internal functionality.
- Difficult to modify or customize internal working.Built-in UART systems do not allow design-level changes.Modifying parameters or behavior is often restricted.This makes it less adaptable to new requirements.
- Less understanding of internal UART operation.Users only interact with high-level functions.Internal processes like state transitions are hidden.This reduces learning opportunities for students.
- Software-based methods may reduce performance speed.Some systems use software to handle UART tasks.This increases processing load on the CPU.As a result, communication speed may decrease.
- Higher complexity when using standard IP cores.IP cores can be complex to integrate and configure.They require detailed knowledge of system design.This may increase development time and effort.
- Not suitable for learning low-level hardware design.Pre-built systems hide the actual hardware logic.Users cannot practice designing from scratch.This limits practical understanding of digital design.

**PROPOSED SYSTEM**

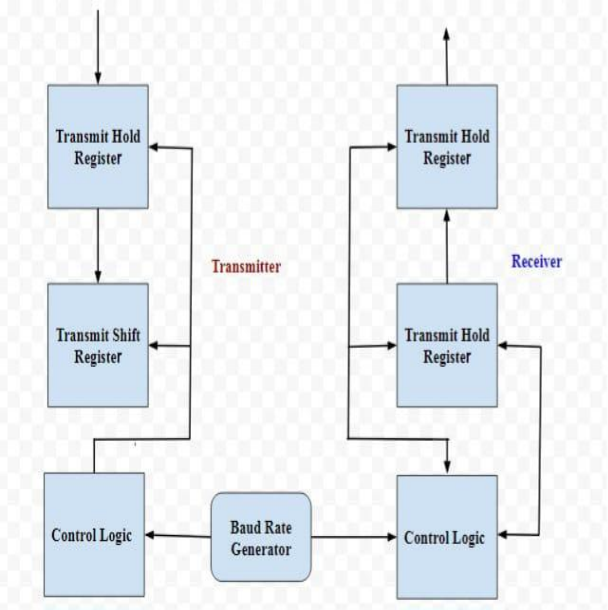
The proposed system is a UART transmitter designed using FSM in Verilog HDL. It enables reliable serial communication by converting parallel input data into serial output.The design includes key modules such as FSM controller, baud rate generator, shift register, and control logic. These components work together to manage data transmission efficiently.The Finite State Machine (FSM) controls the

transmission process. It operates through states like Idle, Start, Data, and Stop to ensure proper sequencing. A baud rate generator is used to produce precise timing signals. This ensures accurate and synchronized data transmission.The shift register converts parallel data into serial bits. It shifts data step-by-step for proper transmission over the communication line. The system is implemented and verified using a Verilog testbench. Simulation results confirm correct functionality and reliable performance.

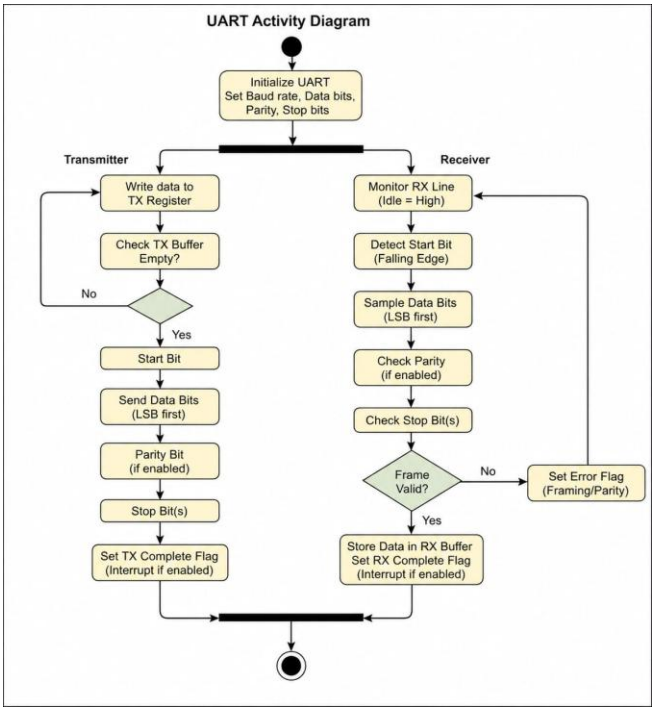
**Advantages of Proposed System**

- Provides reliable and accurate serial communication.Ensures proper data transmission without errors.Suitable for stable communication between devices
- Flexible design using FSM for easy modification.Allows changes in states and control logic.Can be customized based on system requirements
- Efficient parallel-to-serial data conversion.Converts input data into serial form effectively.Maintains correct data sequence during transmission.
- Precise timing control with baud rate generator.Generates accurate clock signals for transmission.Helps in synchronized and error-free communication
- Simple and modular design structure.Divides system into smaller functional blocks.Makes design easy to understand and implement
- Easy to implement using Verilog HDL.Uses simple coding techniques for hardware design.Suitable for beginners and educational purposes
- Supports clear understanding of UART operation.Demonstrates internal working of transmission process.Helpful for learning digital communication concepts
- Scalable for adding advanced features.Can include parity checking and error detection.Supports future enhancements and extensions
- Verified performance through simulation and testbench.Ensures correct functionality before implementation.Provides waveform analysis for validation

**SYSTEM ARCHITECTURE DIAGRAM**

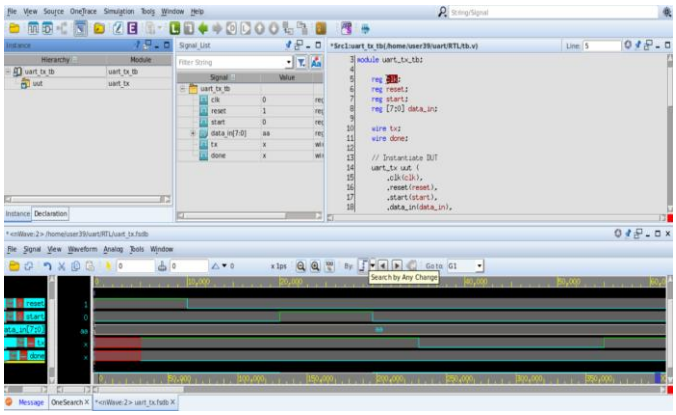


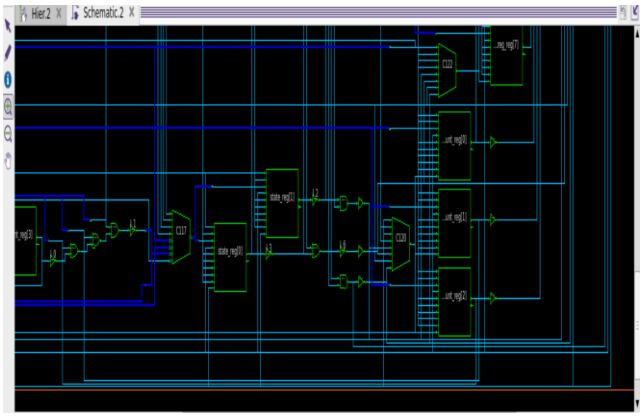

**ACTIVITY DIAGRAM**



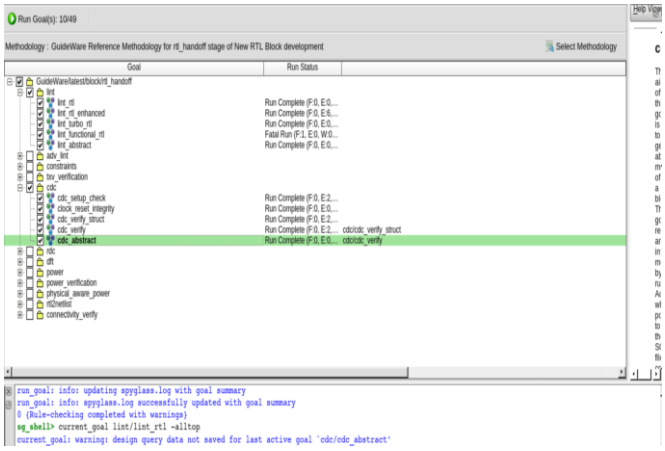

**IV. RESULT**

**VERDI PAGE**

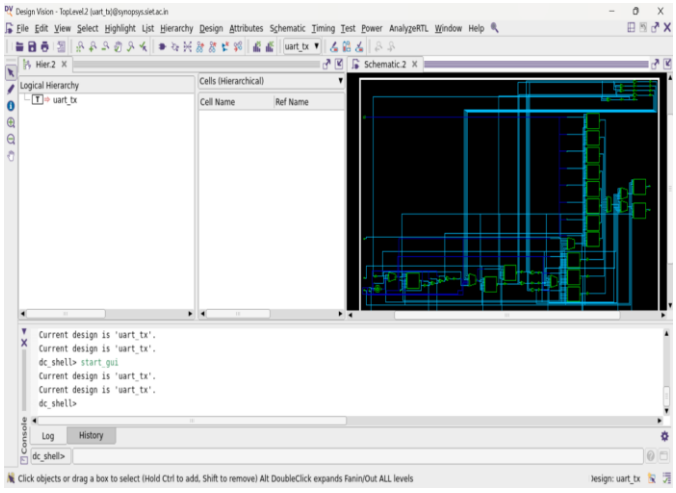




**SPYGLASS PAGE**



**SYNTHESIS PAGE**



**OUTPUT**

**SYSTEM REQUIREMENTS:**

**Software Requirements:**

1. Verilog HDL Coding Environment:
Used to write and develop the UART transmitter design.Helps in describing hardware behavior using code.Example: ModelSim editor, Vivado, or Quartus.

2. Simulation Tool (ModelSim):
Used to simulate and verify the Verilog design.Helps observe waveform outputs and timing behavior.Ensures the design works correctly before hardware use.

3. FPGA Design Software (Xilinx Vivado / Intel Quartus Prime):
Used for synthesis and implementation of the design.Converts Verilog code into hardware configuration.Supports FPGA-based development and testing.

4. Testbench Development Tool:
Used to create test cases for simulation.Helps apply input signals and verify output results.Essential for validating UART functionality.

5. Waveform Viewer Tool:
Used to analyze signal transitions visually.Displays timing diagrams of UART transmission.Helps in debugging and performance analysis.

6. Text Editor / IDE:
Used for writing and editing Verilog code.Provides syntax highlighting and error checking.Examples: VS Code, Notepad++, or built-in editors.

**Module Description:**

1. UART Transmitter Top Module

This is the main module that integrates all submodules of the design.It takes parallel input data and control signals from the user.The module manages the overall data flow during transmission.It connects FSM, shift register, and bit counter together.The output is a serial data stream through the TX line.This module ensures proper coordination of all operations.

2. Finite State Machine (FSM) Module

The FSM controls the sequence of UART transmission.It consists of states like IDLE, START, DATA, and STOP.Each state performs a specific function in data transfer.It changes states based on clock and control signals.FSM ensures correct order and timing of operations.This improves reliability and structured design.

3. Baud Rate Generator Module

This module generates the required transmission speed.It divides the system clock into a slower baud rate clock.The baud rate determines how fast data bits are sent.Common values include 9600, 115200 bits per second.It ensures synchronization between transmitter and receiver.Accurate timing is essential for correct communication.

4. Shift Register Module

The shift register is used to convert parallel data to serial form.It stores the input data temporarily before transmission.Data bits are shifted one by one on each clock cycle.This enables sequential transmission over the TX line.It works together with the FSM for proper control.Ensures smooth and continuous data flow.

5. Bit Counter Module

This module keeps track of the number of bits transmitted.It counts the data bits during the DATA state.Once all bits are sent, it signals the FSM to move forward.It helps in maintaining correct frame length.Prevents extra or missing bits during transmission.Ensures accuracy in UART communication.

6. Start Bit Generator Module

This module generates the start bit for UART frame.The start bit indicates the beginning of transmission.It is usually a logic low (0) signal.FSM activates this module during the START state.It prepares the receiver to accept incoming data.This is essential for proper synchronization.

7. Stop Bit Generator Module

This module generates the stop bit at the end of transmission.The stop bit indicates the completion of data transfer.It is usually a logic high (1) signal.FSM controls this module during the STOP state.It provides a small gap before the next transmission.Ensures clear separation between data frames.

8. Testbench Module

The testbench is used to verify the UART transmitter design.It provides input signals like clock, reset, and data.It simulates different transmission scenarios.Outputs are observed using waveform viewers.Helps detect errors and validate functionality.Ensures the design works correctly before implementation.

## V. CONCLUSION

The project successfully demonstrates the design and implementation of a UART (Universal Asynchronous Receiver/Transmitter) transmitter using a Finite State Machine (FSM) in Verilog HDL. The system converts parallel input data into a serial data stream by following the standard UART protocol, including the generation of start bit, data bits, and stop bit.The FSM-based approach ensures proper sequencing and control of the transmission process, resulting in a reliable and efficient design. The functionality of the system has been verified through simulation, and the waveform results confirm correct operation. The design is also synthesizable, making it suitable for implementation on FPGA or ASIC platforms.Overall, the project provides a strong understanding of serial communication, FSM design, and RTL-based digital system development

## VI. FUTURE SCOPE

The design can be enhanced to support higher baud rates for faster data transmission in advanced communication systems.It can be extended to include error detection techniques such as parity checking and framing error handling.Integration with UART receiver can form a complete full-duplex communication system.The system can be optimized for low power consumption, making it suitable for portable and embedded devices.It can be implemented on FPGA and ASIC platforms for real-time hardware applications.Advanced features like configurable data length, stop bits, and parity modes can be added for flexibility.

## REFERENCES

[1] “Design and Implementation of UART Using Verilog HDL” – International Journal of Engineering Research & Technology (IJERT). This paper explains UART transmitter and receiver design using Verilog HDL and FSM-based control for serial communication.

[2] “FPGA Implementation of UART Controller Using Verilog HDL” – International Journal of Computer Applications (IJCA). It describes UART controller implementation on FPGA and highlights efficient FSM-based serial data transmission.

[3] "Design of UART Serial Communication Module Using Verilog" – International Journal of Advanced Research in Electronics and Communication Engineering (IJARECE). This paper focuses on UART module design and data transmission using Verilog.
[4] "A Study of UART Communication Protocol for Embedded Systems" – IEEE Xplore Digital Library. It provides an overview of UART protocol, its working, and applications in embedded systems.
[5] "Finite State Machine Based UART Design on FPGA" – International Journal of VLSI Design & Communication Systems (VLSICS). This paper discusses FSM-based UART design and its hardware implementation.
[6] "Digital Design" – M. Morris Mano. This book provides fundamental concepts of digital logic design and finite state machines used in this project.
[7] "Verilog HDL: A Guide to Digital Design and Synthesis" – Samir Palnitkar. This book explains Verilog HDL concepts and coding techniques used for UART design.