Avalanche

The Serial Caravan

Design and Verification of an Enhanced UART Protocol using Verilog HDL

2nd year-

Design and implement a reliable and configurable **UART** Transmitter–Receiver system using Verilog HDL, incorporating robust error detection mechanisms and improved sampling accuracy to ensure correct asynchronous serial communication.

TEAM - LOTATRONS

Sathak Modi
2nd year ECE ~ 20244145
sarthak.20244145@mnnit.ac.in
7887294248

Varunveer Dubey
2nd year ECE ~ 20244172
varunveer.20244172@mnnit.ac.in
7089765126

Lewin Mariya
2nd year ECE ~ 20244508
lewin.20244508@mnnit.ac.in
8075062060

Palak Dhingra
2nd year ECE ~ 20244119
palak.20244119@mnnit.ac.in
7974139640

OBJECTIVE

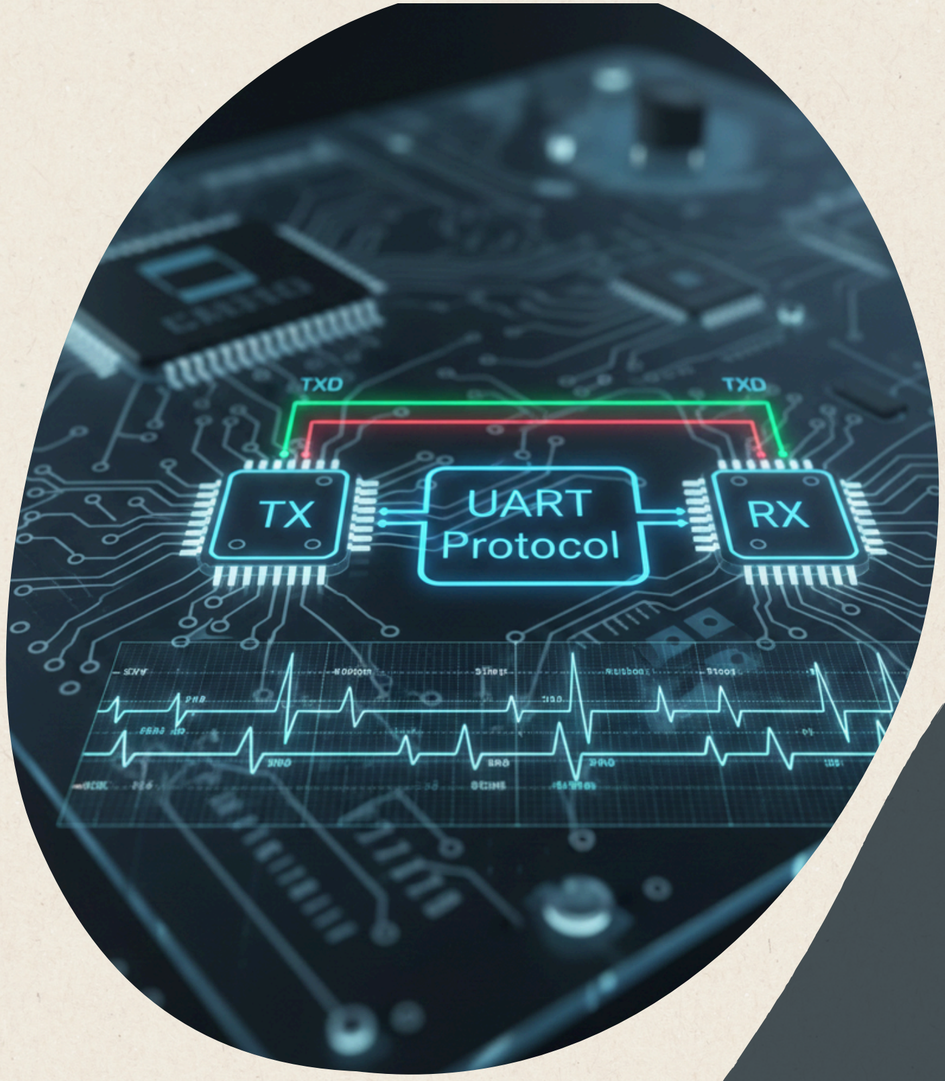
To design and simulate a reliable UART communication system using Verilog HDL.

To support configurable baud rates for flexible asynchronous data transmission.

To implement structured control logic with error detection for improved reliability.

To enhance data security using lightweight XOR-based encryption and decryption.

To verify functional and timing correctness through simulation and waveform analysis.

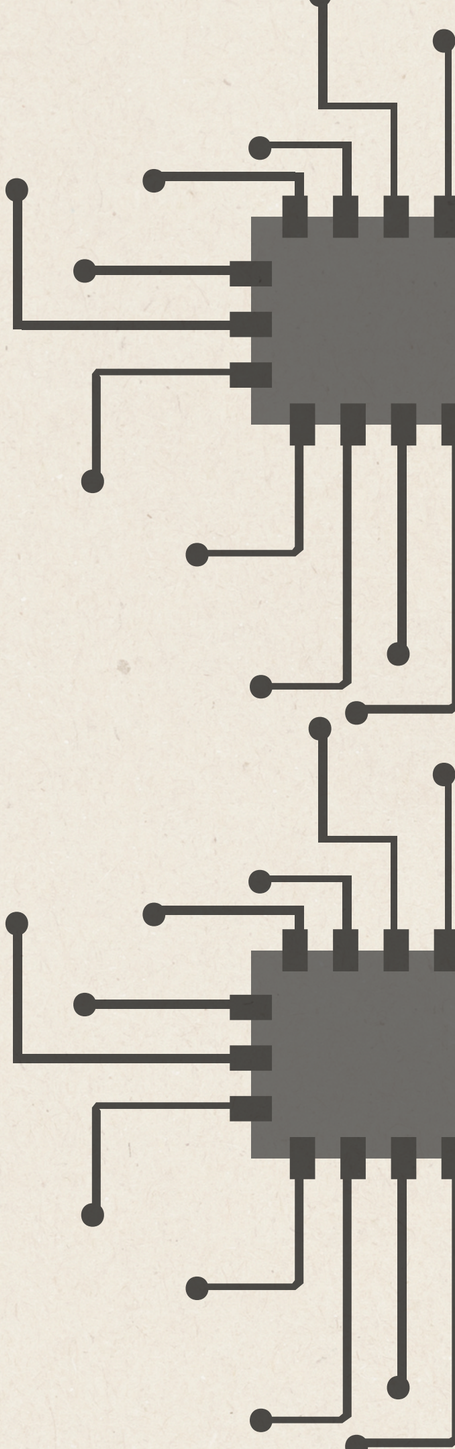


Brief Explanation

The project features an advanced UART Transmitter-Receiver system using Verilog HDL for reliable asynchronous serial communication. The transmitter and receiver use FSM control for proper sequencing of start, data, parity, and stop bits. A configurable baud rate generator from the system clock allows for various speeds. On the receiver side, 16× oversampling accurately samples incoming bits, reducing timing errors and enhancing noise immunity. The design includes parity checking and framing error detection for identifying corrupted data. Status flags aid in monitoring and debugging, while an optional XOR-based encryption provides basic data security with minimal hardware demands.

Why This Problem Was Chosen

UART is one of the most widely used communication protocols in embedded and digital systems, yet basic implementations often suffer from sampling inaccuracies and weak error detection. This project was chosen to address these limitations by enhancing data reliability, configurability, and robustness, while remaining hardware-efficient.



Design & Development Tools

Hardware Description Language:

- Verilog HDL (IEEE 1364 / 2001)

Simulation Tools:

- Icarus Verilog (iverilog), Xilinx Vivado Simulator

Waveform Analysis:

- GTKWave, Surfer

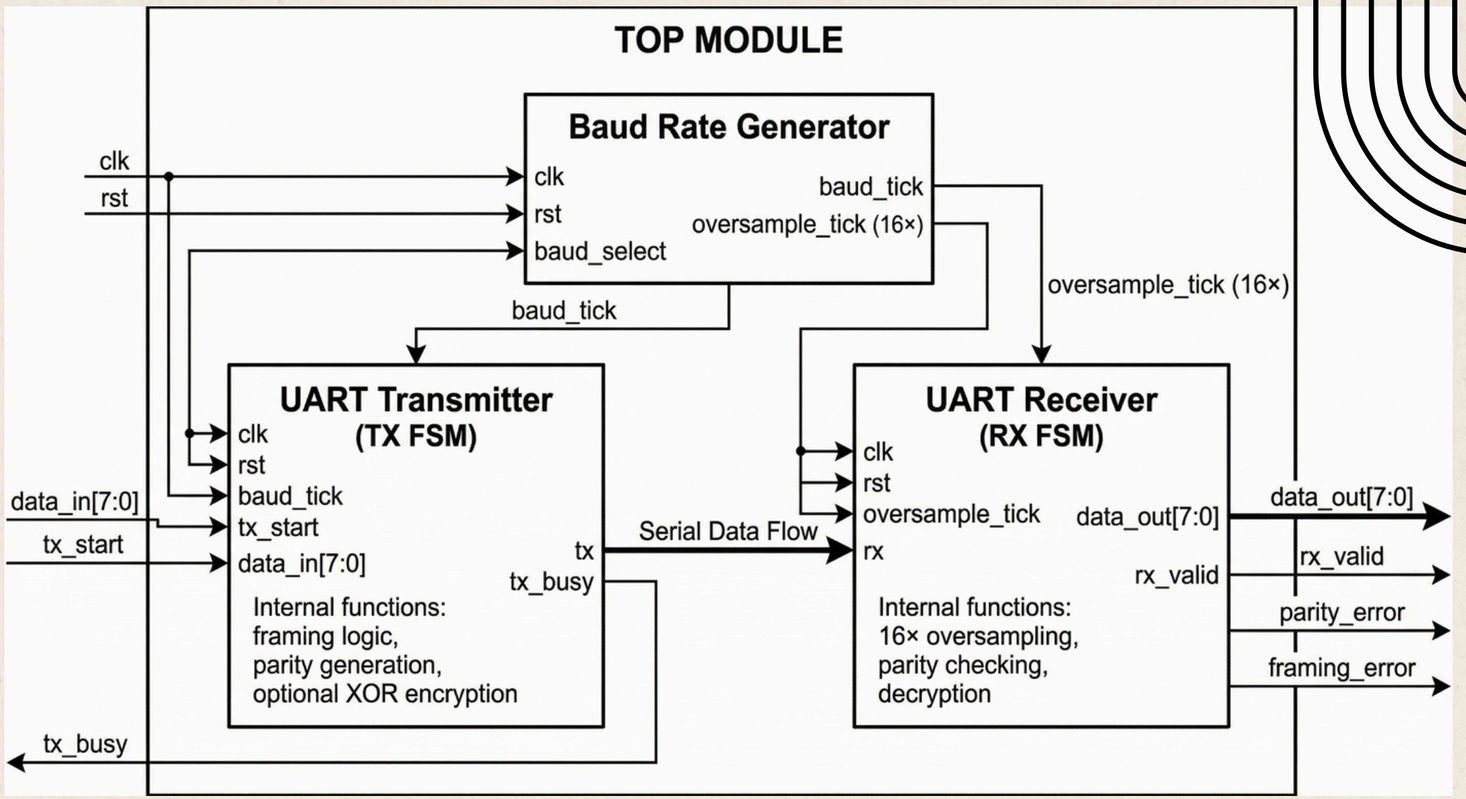
Code Editor / IDE:

- Visual Studio Code

Version Control System:

- Git (with GitHub for repository hosting)

Block Diagram



System Architecture & Features

System Architecture

- FSM-based UART Transmitter and Receiver
- Configurable baud rate derived from system clock
- Shared baud timing for synchronized operation

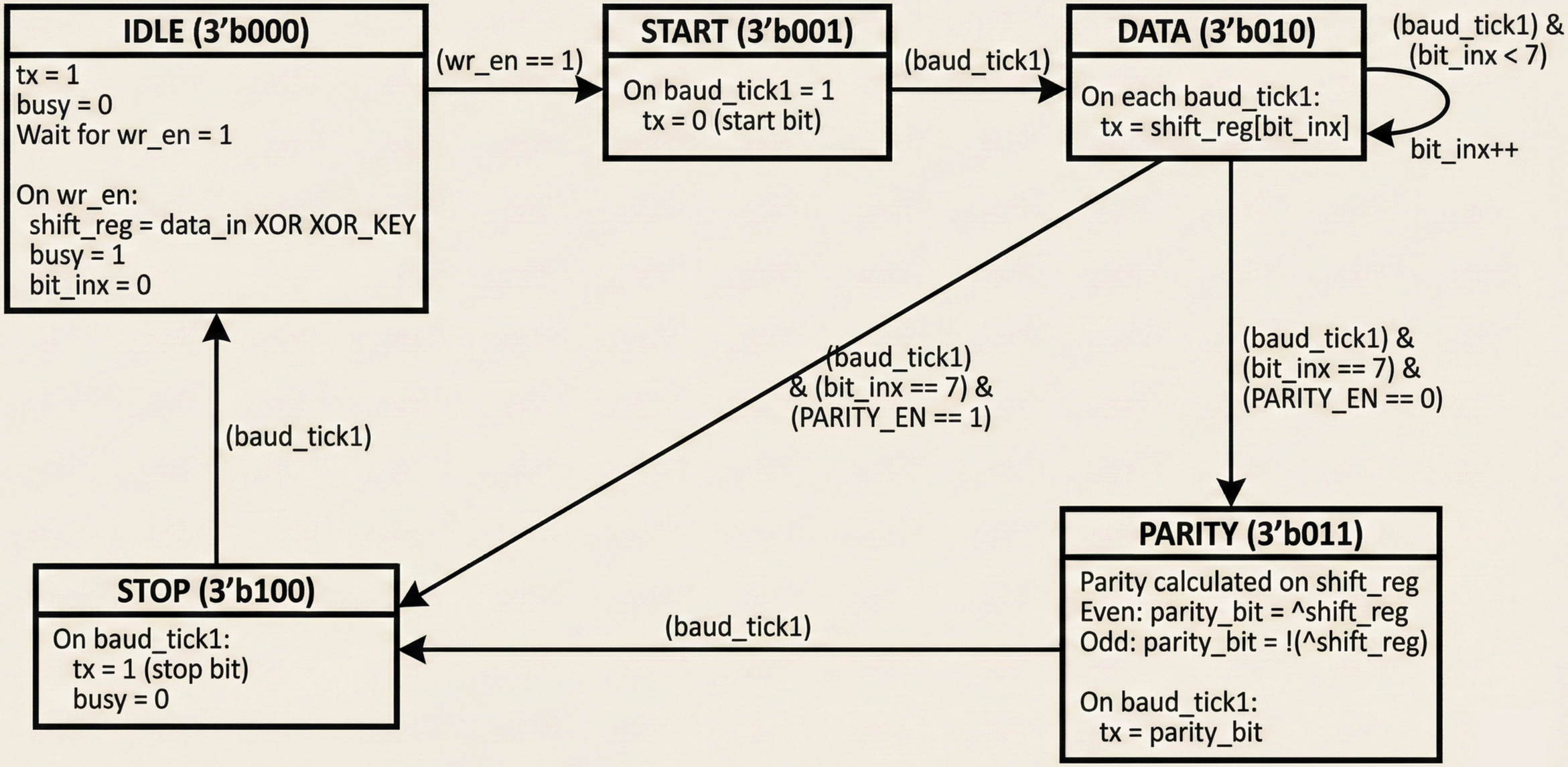
Key Features

- Configurable baud rate
- UART TX & RX using FSMs
- Parity generation and checking
- Framing error detection
- Status flags for TX/RX state monitoring
- 16× oversampling receiver for accurate bit sampling
- Optional XOR-based data encryption for secure communication

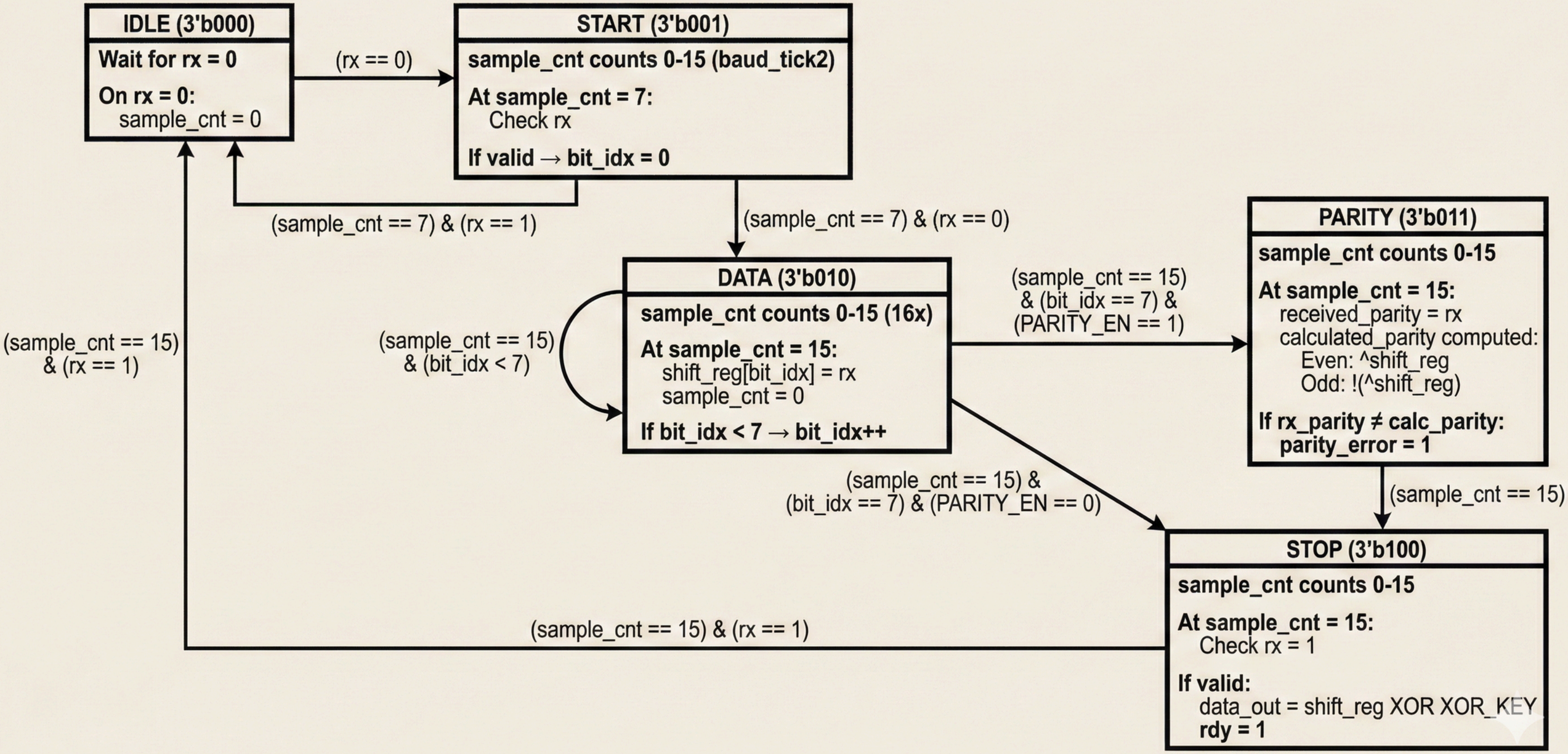
Verification

- Functional verification using Verilog testbench
- Timing and functional correctness validated using waveform analysis

UART Transmitter FSM (Verilog HDL)



UART Receiver FSM (Verilog HDL, 16x Oversampling, Parity, XOR Decryption)



Baud Rate Calculation

The UART system uses a 50 MHz system clock and supports a configurable baud rate of 9600 bps.

Bit Duration Calculation

Bit Time (Tbit) = 1 / Baud Rate

Tbit = 1 / 9600
Tbit = 104.167 microseconds

Clock Period (Tclk) = 1 / 50,000,000
Tclk = 20 nanoseconds

Number of clock cycles per bit:

baud_div1 = clk_freq / baud
baud_div1 = 50,000,000 / 9600
baud_div1 ≈ 5208 clock cycles

✓ **One UART bit = 5208 system clock cycles**
✓ **Generates baud_tick1 (used by Transmitter)**

16× Oversampling for Receiver

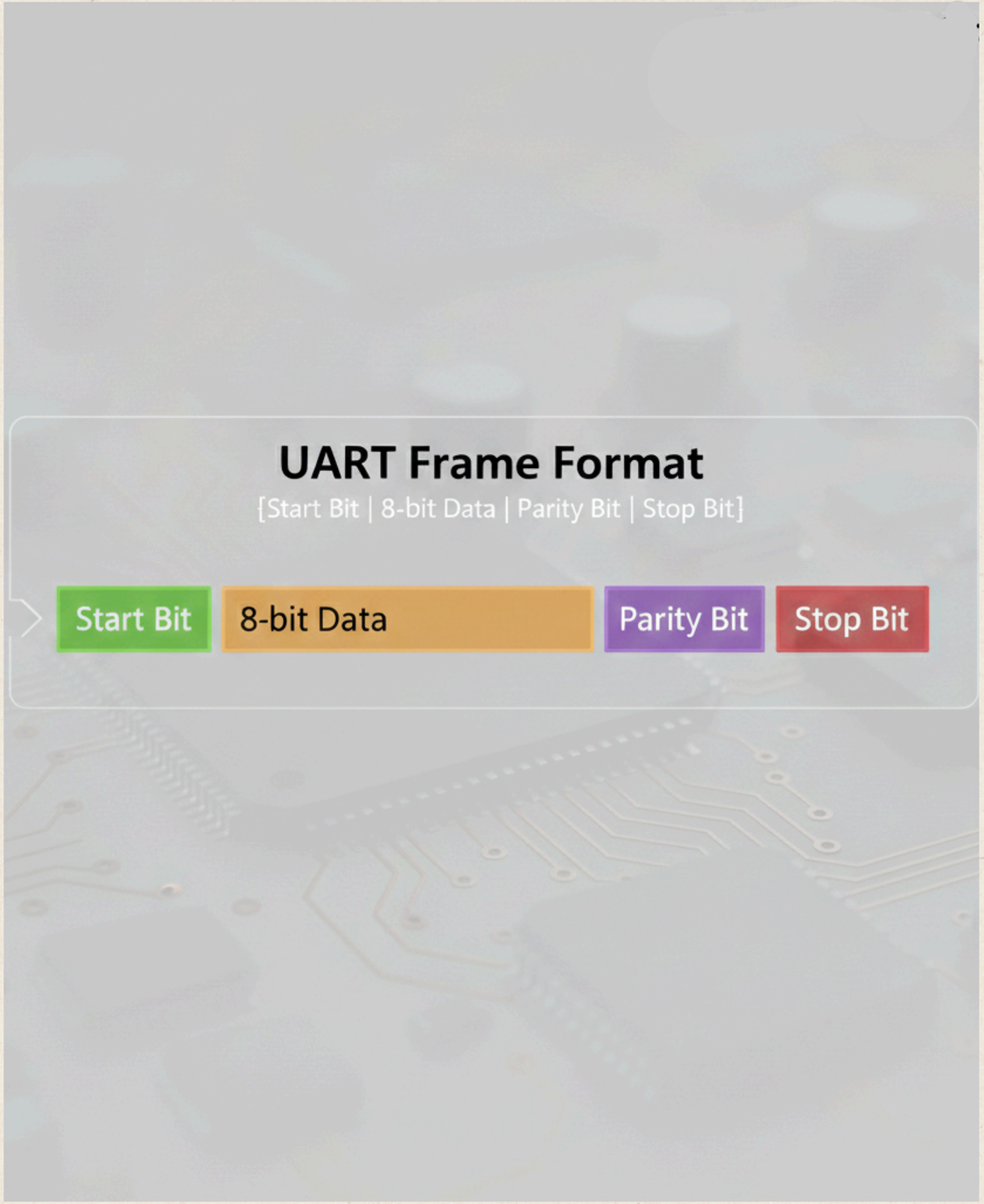
Receiver samples each bit 16 times for accuracy.

baud_div2 = clk_freq / (baud × 16)
baud_div2 = 50,000,000 / (9600 × 16)
baud_div2 ≈ 325 clock cycles

✓ **One oversample tick = 325 clock cycles**
✓ **Generates baud_tick2 (used by Receiver)**

Why 16× Oversampling?

- Samples at center of bit
- Improves noise immunity
- Tolerates small clock mismatch
- Industry standard UART technique



UART Frame Format

The implemented UART communication follows a fixed serial data frame structure to ensure reliable transmission and reception. Each data packet begins with a start bit to indicate the start of communication, followed by 8 data bits transmitted in sequence. A parity bit is then appended to support error detection during reception. Finally, a stop bit is added to mark the end of the data frame and allow proper synchronization before the next transmission.

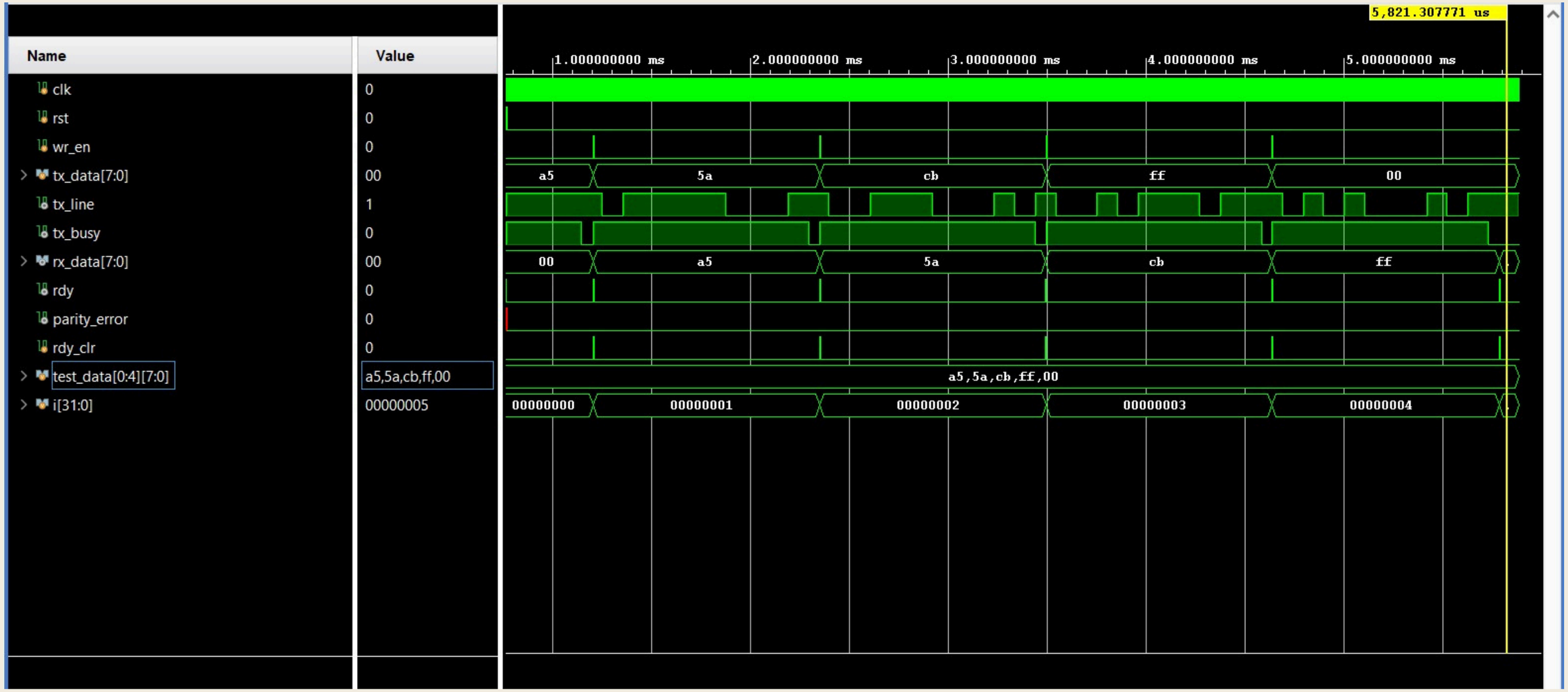
XOR Encryption

- UART transmits data without built-in security, making it vulnerable to interception
- XOR encryption is applied to enhance data confidentiality with minimal hardware complexity
- The 8-bit data field (D0–D7) is XORed with a fixed key before transmission
- At the receiver, the same XOR key is applied to recover the original data
- This method is fast, simple, and suitable for FPGA/Verilog-based UART systems

Testbench Strategy

The UART testbench is designed to verify reliable transmitter and receiver operation. A clock and reset are first applied to initialize the system into an idle state. When the write enable signal is asserted, 8-bit parallel data is loaded into the transmitter and sent in standard UART frame format with start bit, data bits, parity bit, and stop bit. The receiver samples incoming data using 16× oversampling at the center of each bit to ensure accurate reception. The tx_busy and rdy signals are monitored to confirm proper transmission and reception. XOR-based encryption is enabled to validate correct data encryption before transmission and successful decryption at the receiver, with no parity or framing errors observed.

WAVEFORM


Name
Value
clk
0
rst
0
wr_en
0
tx_data[7:0]
00
tx_line
1
tx_busy
0
rx_data[7:0]
00
rdy
0
parity_error
0
rdy_clr
0
test_data[0:4][7:0]
a5,5a,cb,ff,00
i[31:0]
00000005
5,821.307771 us
1.000000000 ms
2.000000000 ms
3.000000000 ms
4.000000000 ms
5.000000000 ms
a5
5a
cb
ff
00
00
a5
5a
cb
ff
a5,5a,cb,ff,00
00000000
00000001
00000002
00000003
00000004

RESULTS

- The UART transmitter and receiver were successfully implemented using **finite state machines (FSMs) with a configurable baud rate.**
- Simulation results confirm correct transmission and reception of multiple 8-bit data values (A5, 5A, CB, FF, 00) without data loss.
- The TX busy flag accurately indicates the active transmission period and returns low after stop-bit completion.
- The RX ready **(rdy)** signal asserts once a full frame is received, confirming successful reception.
- **Parity generation** and **checking** operate correctly, with no parity errors observed during valid data transmission.
- Framing error detection remains inactive, indicating **proper start and stop bit alignment**.
- The **16× oversampling receiver** ensures stable mid-bit sampling, improving noise immunity and timing accuracy.
- Optional **XOR-based encryption** correctly encrypts data before transmission and restores the original data after reception.
- Overall, the design demonstrates reliable, accurate, and secure **serial communication**, meeting all functional requirements.

WAVEFORM ANALYSIS

- The clk signal provides the system timing reference, while **rst initializes the UART into an idle state**.
- When **wr_en** is asserted, parallel input data (**tx_data[7:0]**) is **loaded** into the transmitter.
- The **tx_line** waveform shows:
- **Start bit (logic 0)**
- **8 data bits (LSB first)**
- **Parity bit**
- **Stop bit (logic 1)**
- The **tx_busy** signal **remains high** throughout the transmission frame, confirming proper FSM control.
- At the receiver side:
- The incoming serial data is sampled using **16× oversampling**, with **sampling performed at the center** of each bit.
- The recovered data appears correctly on **rx_data[7:0]**.
- The **rdy signal pulses high after each complete frame**, indicating valid data reception.
- No assertion of **parity_error** or **framing_error** signals is observed, verifying correct parity handling and frame integrity.
- The waveform confirms accurate timing, correct protocol sequencing, and robust UART operation.

CONCLUSION

The UART transmitter and receiver were successfully designed and verified using **FSM-based architecture** with a configurable baud rate. Simulation results confirm **reliable serial communication** with accurate data transmission and reception. The implementation of 16× oversampling in the receiver ensured precise **mid-bit sampling**, significantly improving timing accuracy and robustness against noise.

Parity generation and checking mechanisms functioned correctly, while framing error detection effectively validated start and stop bit integrity. Status flags such as **TX busy and RX ready** provided clear monitoring of transmission and reception states. Additionally, the optional **XOR-based encryption** enhanced data security without affecting communication reliability.

Overall, the waveform analysis and simulation results demonstrate that the proposed UART design is functionally correct, efficient, and robust, making it suitable for embedded systems and serial communication applications requiring reliability, flexibility, and error detection