ECE 593 2025 WINTER

Design Specifications for UART Protocol Project TEAM 13

Git link: https://github.com/Defuse-cfg/UART-PROTOCOL

1. Overview

The UART system is designed to facilitate asynchronous serial communication with FIFO buffering and clock management. It consists of the following key components:

- uart_top: Integrates all modules
- uart_tx: Handles data transmission
- uart_rx: Handles data reception
- FIFO: Buffers data to manage flow control
- b_clk: Generates baud rate clock

2. Functional Requirements

Data Transmission

- uart_tx accepts 8-bit parallel data, adds a Start bit (1-bit), Parity bit (1-bit), and Stop bit (1-bit), then transmits data serially.

Data Reception

- uart_rx detects Start bit and Stop bit, collects 8-bit data, verifies parity, and outputs received byte.

FIFO Buffering

- Temporary storage for transmitted and received data, integrated in uart_top for efficient flow control.

Clock Management

- Baud rate clock generation based on divisor ensures synchronized transmission and reception.

3. Design Overview

Key modules of the design include:

- uart_top: Top-level module integrating all components.
- uart_tx: Converts 8-bit parallel data into serial format.
- uart_rx: Receives serial data, verifies Start, Stop, and Parity bits.
- FIFO: 16-entry buffer to store incoming/outgoing data.
- b_clk: Generates baud rate clock for precise timing.

4. Design Parameters

Packet Format	Data (8 bits) + Start (1 bit) + Stop (1 bit) + Parity (1 bit)
FIFO Size	16 entries (8-bit each)
System Clock Frequency	50 MHz

5. Module Specifications

• uart_top: Controls data flow between transmitter, receiver, and FIFO buffers.

• uart_tx: Converts 8-bit parallel data into serial format, adds Start, Parity, and Stop bits, and manages state transitions.

• uart_rx: Receives serial data, detects Start and Stop bits, verifies parity, and outputs received data.

• FIFO Buffer: Implements a 16-entry FIFO queue for buffering data with flow control.

6. Timing Requirements

System Clock Frequency: 50 MHz (Period = 20 ns)

Baud Rate Calculation:

Divisor (dvsr) = System Clock Frequency / (Baud Rate × Oversampling Factor)

For 115200 bps baud rate: dvsr = 50 MHz / (115200 × 16) = 27

Baud rate = 50 MHz / 27 = 1.85 MHz

Transaction Time for One Frame = 10 / 1.85 MHz = 5.40 µs

7. Performance Analysis

• Maximum Data Rate: 1.85 million bits per second.

• FIFO Buffer Capacity: 16 entries of 8-bit each (Total = 128 bits).

• Latency: Single Data Transmission Cycle = 5.4 µs.

8. Conclusion

This document outlines the design specifications, calculations, and performance metrics for the UART system. The architecture ensures asynchronous data transfer, reliable buffering, and precise timing with FIFO and baud rate clock generation.

Team Responsibilities:

Benarji and Vedant - UVM Testbench Infrastructure

Develop UVM Environment, including:

- uvm_env, uvm_agent, uvm_scoreboard
- uvm_sequencer, uvm_driver, uvm_monitor
- Implement the virtual interface for DUT communication.

Sahil and Raghu- UVM Sequences & Test Cases

Write UVM testcases for:

- Basic UART functionality (TX & RX)
- Edge Cases (FIFO full, FIFO empty, Parity errors)
- Stress Testing (Random data bursts)
- Develop reusable UVM sequences with randomization.

Vedant and Sahil- Coverage & Assertions

Implement functional coverage:

- Different baud rates
- FIFO buffer states
- Start/Stop bit detection
- Add SystemVerilog Assertions (SVA) for protocol compliance.

Benarji and Raghu - Debugging & Report Generation

- Run UVM simulations and debug failures.
- Optimize the testbench for better performance.
- Generate coverage reports and summarize test results