UART Transmitter (uart_tx) Documentation

Introduction

The uart_tx project implements a UART (Universal Asynchronous Receiver-Transmitter) transmitter module using FPGA. This module facilitates serial communication by transmitting data over a UART interface. UART is widely used in embedded systems for device-to-device communication over serial ports. This document provides a comprehensive guide to understanding, implementing, testing, and verifying the uart_tx module.

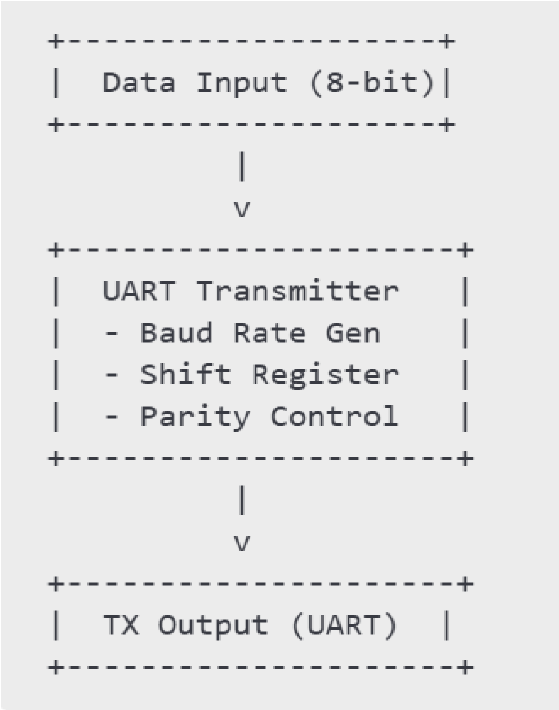
Steps for Implementation

1. Study the Existing Code

- **Repository Access:** Access the uart_tx project from the VSDSquadron_FM repository. This repository contains the Verilog source code required for implementing the UART transmitter.
- **Understanding Verilog Code:**
    - The code includes modules for handling the serial data transmission process.
    - Key components of the Verilog implementation:
        - **Baud Rate Generator:** Responsible for generating the clock signal at the required baud rate.
        - **Shift Register:** Shifts out data bits serially over the TX line.
        - **Start and Stop Bit Handling:** Ensures proper framing of data according to UART protocol.
        - **Parity Bit (Optional):** Used for error detection in some implementations.

2. Design Documentation

- **Block Diagram:** The following block diagram represents the structure of the uart_tx module:



```
+--------------------+
|  Data Input (8-bit)|
+--------------------+
          |
          v
+---------------------+
|  UART Transmitter   |
|  - Baud Rate Gen    |
|  - Shift Register   |
|  - Parity Control   |
+---------------------+
          |
          v
+---------------------+
|  TX Output (UART)  |
+---------------------+
```

- **Circuit Diagram:**
  - Develop a circuit diagram illustrating the connection between the FPGA's UART TX pin and the receiving device (such as a PC or another UART-compatible device).
  - Example of UART TX pin connection:
    - **FPGA TX Pin → USB-to-Serial Adapter RX Pin**
    - **GND Connection between FPGA and Serial Adapter**

3. Implementation

- **Hardware Setup:**
  - Components required:
    - FPGA development board (VSDSquadron Mini FPGA or any compatible FPGA)
    - USB-to-Serial adapter (such as FTDI) for interfacing with a PC
    - Wires and connectors for interfacing

    - Steps:
        1. Connect the FPGA TX pin to the USB-to-Serial adapter RX pin.
        2. Ensure the ground (GND) of the FPGA and the adapter are connected.
        3. Power up the FPGA board.
- **FPGA Programming:**
    - Synthesize the Verilog code using an FPGA development tool (such as Yosys, Quartus, or Vivado).
    - Generate the bitstream and upload it to the FPGA.
    - Verify the programming by checking the UART output.

4. Testing and Verification

- **Setup for Testing:**
    - Connect the FPGA to a PC using a USB-to-Serial adapter.
    - Open a serial terminal application (such as PuTTY, Tera Term, or RealTerm).
    - Set the correct baud rate (e.g., 9600, 115200 bps) matching the FPGA configuration.
- **Observing Transmitted Data:**
    - The FPGA should transmit predefined data over the UART TX line.
    - The serial terminal on the PC should display the received characters.
    - Verify proper data transmission by checking for any dropped or garbled characters.
- **Debugging Issues:**
    - If data is not received, check the TX-RX connections and ensure the FPGA is programmed correctly.
    - Verify baud rate settings in both FPGA and serial terminal.
    - Use an oscilloscope or logic analyzer to inspect TX signal timing.

5. Documentation

- **Report Preparation:**
    - Document the block diagram, circuit diagram, and step-by-step implementation.
    - Include Verilog code walkthrough with explanations.

  - Record test results and observations.
- **Video Demonstration:**
  - Create a short video showcasing the UART transmission in operation.
  - Explain the circuit setup and show live data transmission on a serial terminal.

Conclusion

The uart_tx module is a fundamental component for serial communication in FPGA-based projects. By following the implementation steps, one can successfully transmit data using UART and integrate it into larger embedded systems. This documentation provides detailed insights into both hardware and software aspects of the project, ensuring a smooth development process.