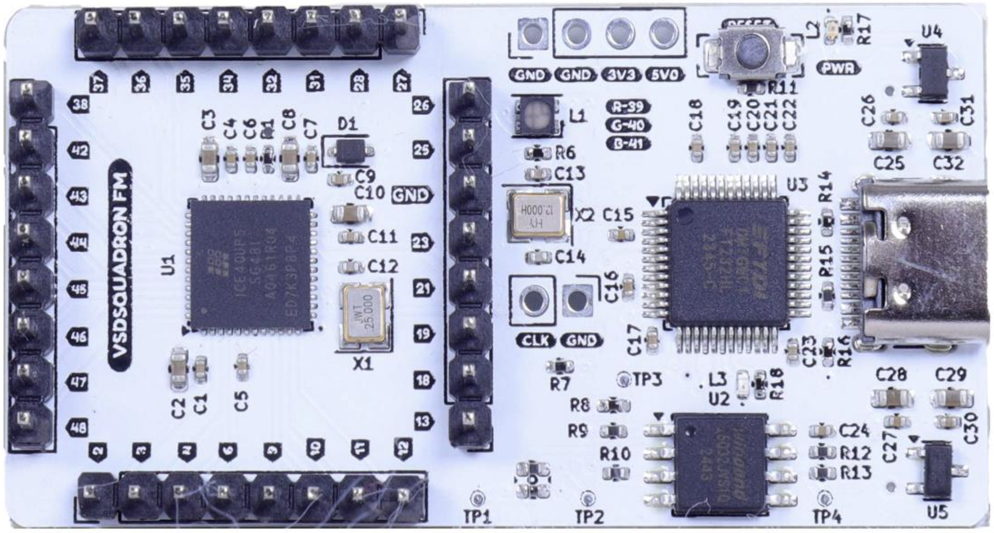


VSDFPGA Task 2

SUBMITTED BY

ISHA SINGH

Introduction

Universal Asynchronous Receiver Transmitter (UART) is one of the most widely used serial communication protocols in embedded systems due to its simplicity and reliability. UART enables full-duplex communication between digital systems using only two signal lines: Transmit (TX) and Receive (RX), along with a common ground.

This project focuses on implementing a UART loopback mechanism on the VSDSquadron FPGA Mini platform. In a loopback configuration, the data transmitted by the UART transmitter is immediately routed back to the UART receiver. This technique is extremely useful for functional verification, debugging, and validation of UART modules without requiring external peripherals.

1. UART FUNCTIONALITY

UART Communication Basics

UART communication involves:

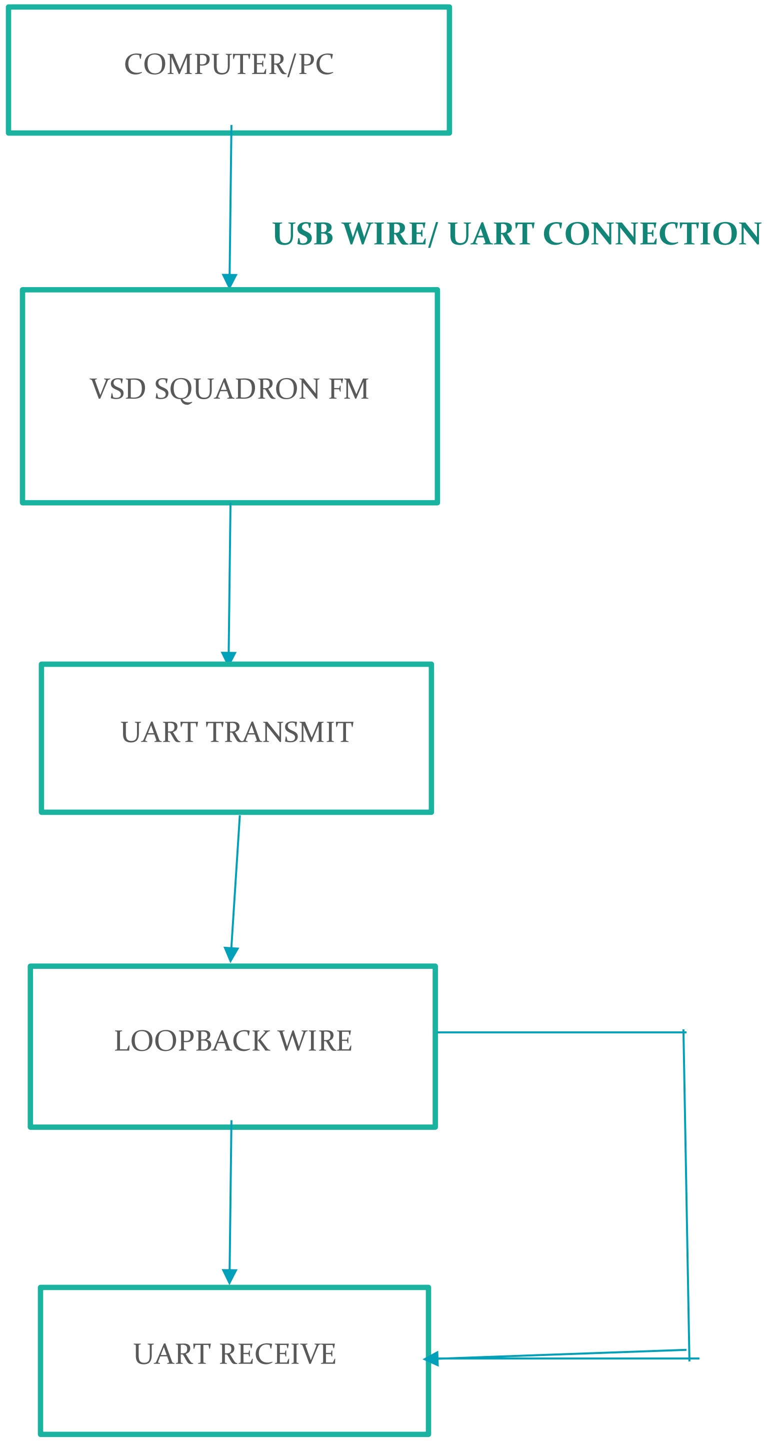
- **TX (Transmit)**: Sends serial data bit-by-bit
- **RX (Receive)**: Receives serial data
- **Baud Rate**: Speed of data transmission
- **Frame Format**:
    - I. 1 Start Bit (LOW)
    - II. 8 Data Bits (LSB first)
    - III. 1 Stop Bit (HIGH)

2. LOOPBACK UART LOGIC

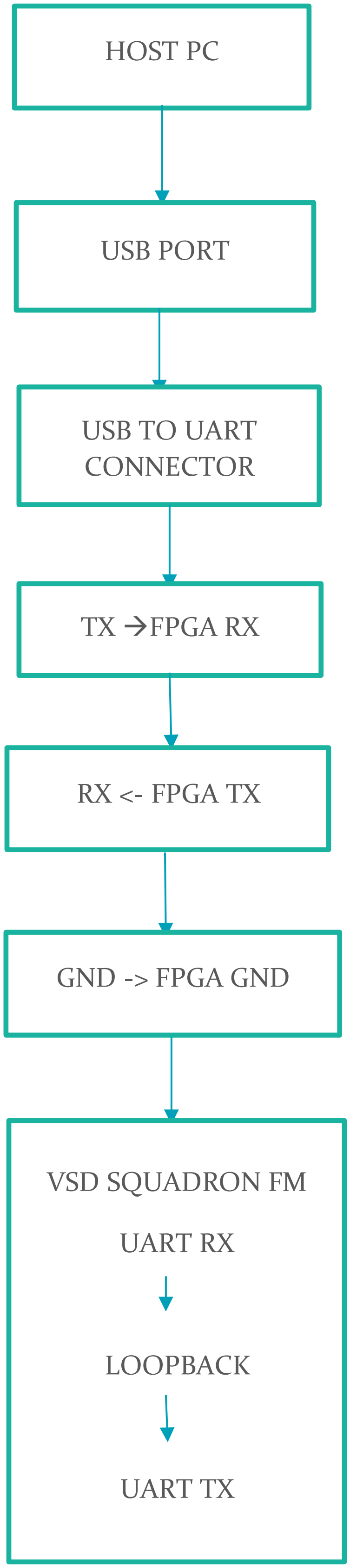
In the UART loopback design, the transmit (TX) output of the UART transmitter is internally connected directly to the receive (RX) input of the UART receiver within the FPGA. When a user sends data from a host computer using a serial terminal, the serial data stream first enters the FPGA through the UART RX pin. The UART receiver module samples the incoming bits according to the configured baud rate, detects the start and stop bits, and reconstructs the original parallel data byte. This received data is then immediately routed through the loopback path without any modification or processing. The UART transmitter takes this same data and converts it back into a serial bit stream, appending the required start and stop bits. Finally, the data is transmitted back to the host computer through the UART TX pin. Observing the same data echoed on the serial terminal confirms that both the UART transmission and reception mechanisms are functioning correctly and are properly synchronized.

Reference link: https://github.com/thesourcerer8/VSDSquadron_FM/blob/main/uart_loopback/uart_trx.v

3. BLOCK DIAGRAM



4. CIRCUIT DIAGRAM



When data is typed in the serial terminal on the host computer, it travels through the USB-to-UART converter and enters the FPGA via the RX pin. The FPGA's UART receiver decodes this data and immediately forwards it to the UART transmitter through the internal loopback connection. The transmitter then sends the same data back to the host computer, where it appears on the terminal. This confirms correct wiring, proper UART timing, and successful FPGA operation.

FPGA TX Pin → USB-UART RX Pin
The TX (Transmit) pin of the FPGA is connected to the RX (Receive) pin of the USB-to-UART converter. This path carries serial data transmitted from the FPGA back to the host computer.

FPGA RX Pin → USB-UART TX Pin
The RX (Receive) pin of the FPGA is connected to the TX (Transmit) pin of the USB-to-UART converter. This allows the FPGA to receive serial data sent from the host computer.

FPGA GND → USB-UART GND
A common ground connection is mandatory to ensure that both the FPGA and the USB-to-UART interface share the same voltage reference. Without this, reliable data communication cannot occur.

FPGA VCC → Power Supply
The FPGA is powered using the specified operating voltage (typically 3.3 V). This power may be supplied via USB or an onboard voltage regulator depending on the board design.

The UART Receiver (RX module) captures incoming serial data from the RX pin and converts it into parallel data.

This parallel data is directly routed to the UART Transmitter (TX module) without any modification.

The transmitter then sends the same data back as a serial stream.

Integration Steps and Observations While Working with the VSDSquadron FPGA Mini Board

1. The VSDSquadron FPGA Mini board was connected to the host computer using a USB Type-C cable, as specified in the datasheet. Verified that the USB cable supported data transfer (not power-only)

2. Ensured the board's power LED was ON

3. Confirmed proper physical seating of the USB-C connector

4. After connecting the board, the system was checked to ensure the FPGA Mini board was detected by the operating system.

5. lsusb command was used

6. The output confirmed the presence of an **FTDI device**, indicating that the onboard programmer was recognized successfully.

7. As specified in the VSDSquadron FPGA Mini datasheet, the open-source iCE40 FPGA toolchain was installed. This included:

- Yosys (for Verilog synthesis)
- nextpnr-ice40 (for place and route)
- icestorm utilities (icepack, iceprog)

8. The provided Makefile was used to automate the build and flash process.

make clean: This command removed previously generated files such as .json, .asc, and .bin, ensuring a clean build environment.

make build: This step performed the Verilog synthesis, place and route, bitstream generation.

9. Successful execution resulted in the generation of the FPGA programming file (.bin).

10. Once the build completed successfully, the bitstream was flashed onto the FPGA Mini board using:

```
sudo make flash
```

11. Root privileges were required to access the USB programming interface.

12. The flashing process wrote the bitstream into the SPI flash memory on the board, enabling non-volatile storage of the design.

13. After programming, a serial screen terminal was installed to test UART communication with the FPGA.
14. The serial terminal was launched using the command:

```
sudo screen /dev/ttyUSB0 115200
```

15. Data typed in the serial terminal was transmitted to the FPGA, and due to the UART loopback implementation, the same data was received back and displayed, confirming successful UART operation.

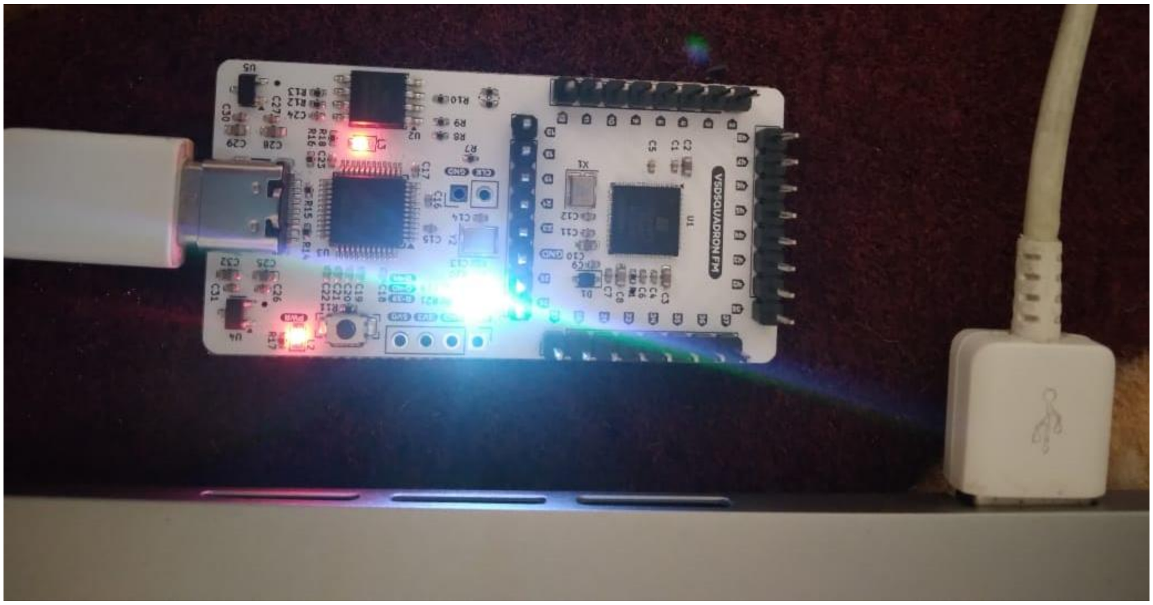
In the implemented design, the UART loopback is achieved through a direct hardware connection between RX and TX pins, without internal baud-rate generation or UART protocol decoding. Therefore, the effective baud rate is governed entirely by the external USB-to-UART interface and the serial terminal configuration.

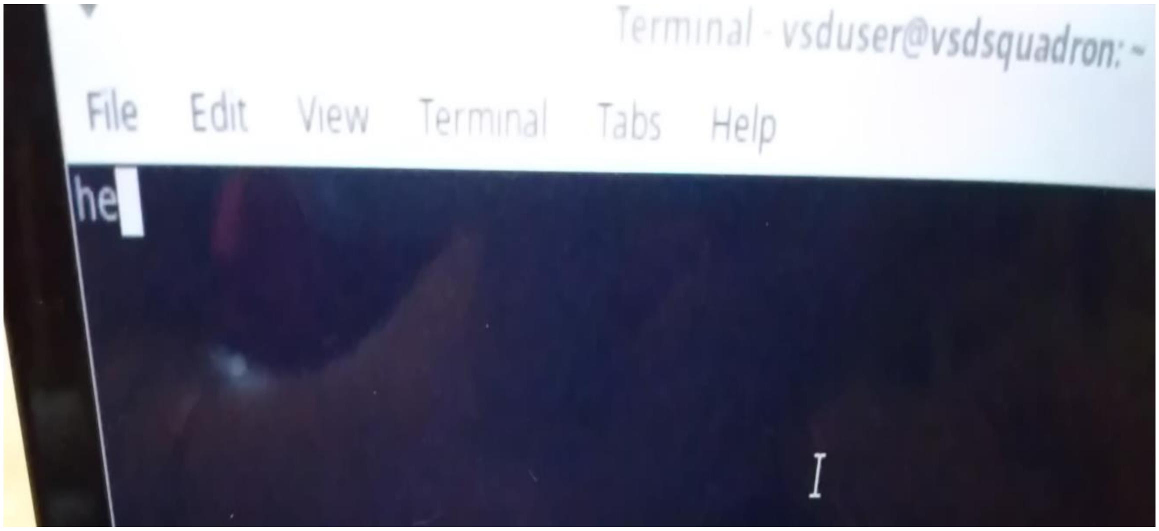
Observations:

- When characters are typed in the serial terminal, the same characters are immediately displayed back on the screen, indicating successful UART loopback operation.
- This confirms that the FPGA correctly routes the incoming UART data from the RX pin directly to the TX pin without data loss or distortion.

OUTPUT

The serial terminal displayed the same data that was transmitted, confirming correct UART hardware loopback operation on the VSDSquadron FPGA Mini board.




Terminal - vsduser@vsdsquadron: ~
File
Edit
View
Terminal
Tabs
Help

```
he
```