ieeecusb

Digital Design project

UART

2024-
2025

IEEE CAIRO UNIVERSITY
STUDENT BRANCH

UART

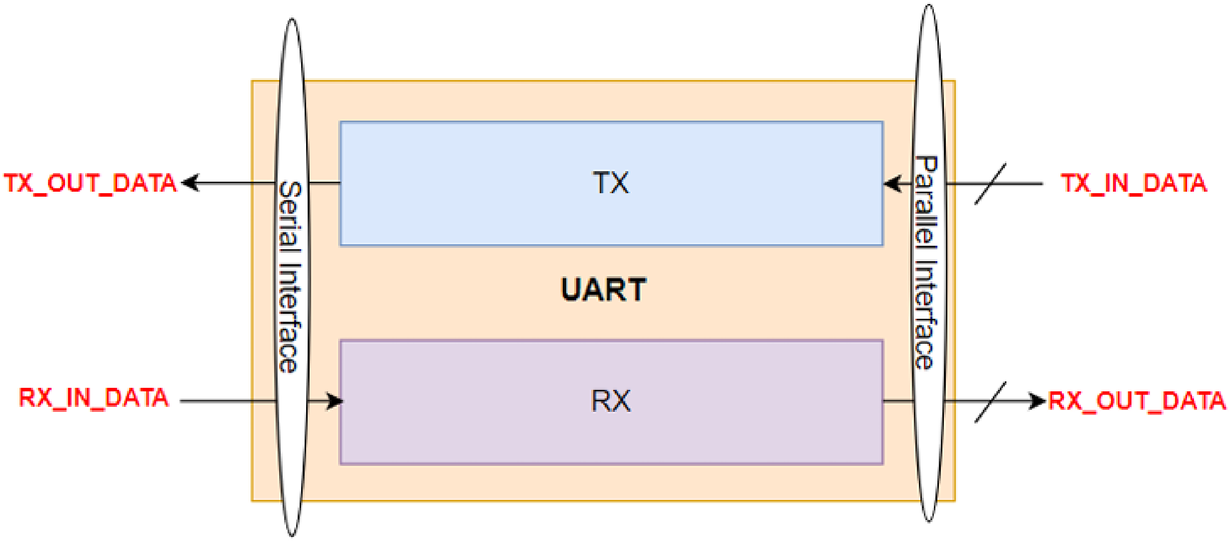
Introduction

This project details the design and implementation of a UART (Universal Asynchronous Receiver/Transmitter) Transmitter and Receiver using Verilog. UART is a full-duplex serial communication protocol that allows data transmission in both directions simultaneously. It consists of two main components:

1. UART Transmitter (TX) – Converts parallel data into serial data for transmission.
2. UART Receiver (RX) – Converts received serial data back into parallel form.

Objective

The goal of this project is to implement the UART receiver & transmitter referred to in the Figure, using Verilog for both Tx & Rx.

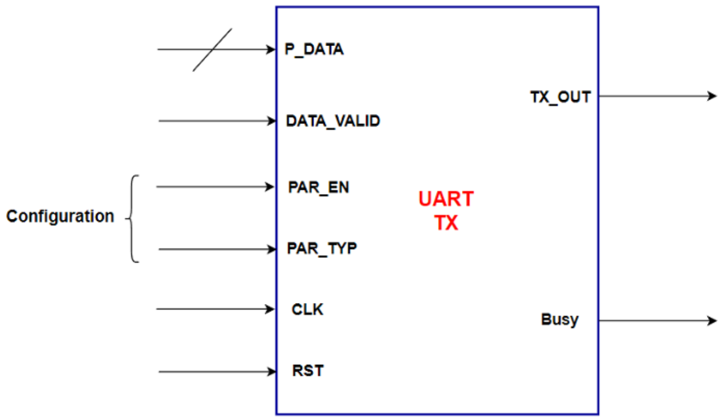


1 UART Tx & Rx

Main Modules

UART transmitter (Tx)

The UART Transmitter converts parallel data into serial format and transmits it to the receiver.



Signal name	Direction	Description
CLK	Input	System clock signal
RST	Input	Synchronized reset signal
PAR_TYP	Input	Parity type: 0(even), 1(odd)
PAR_EN	Input	Parity enable: 0(Disabled), 1(enabled)
P_DATA	Input	8-bit parallel data input
DATA_VALID	Input	Data valid signal
TX_OUT	Output	Serial data output
Busy	Output	High when transmission is in progress

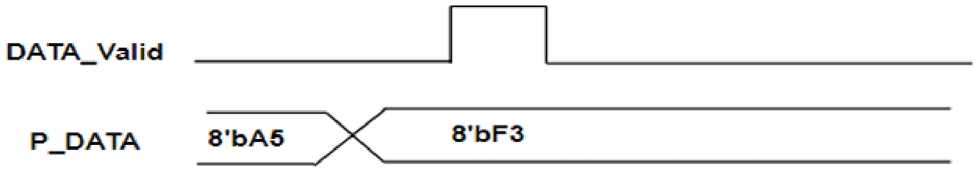
Main Modules

TX Functional Description

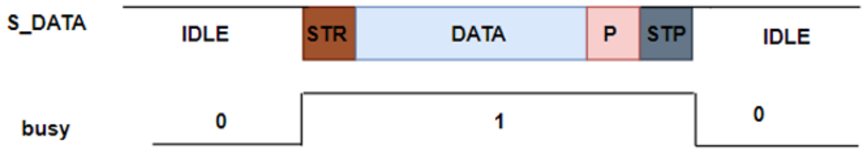
- Data is transmitted only when **DATA_VALID** is high.
- Registers are cleared using an active-low reset.
- **Busy** remains high while data is being transmitted.
- If **PAR_EN** is enabled, a parity bit is appended based on **PAR_TYP**.
- The transmitted frame format depends on the parity settings:
  TX Frame Formats
  1. Parity Enabled (Even/Odd)
     - Start bit (0)
     - 8-bit data (LSB first)
     - Parity bit (Even or Odd)
     - Stop bit (1)
  2. Parity Disabled
     - Start bit (0)
     - 8-bit data (LSB first)
     - Stop bit (1)

Waveforms

Expected input:

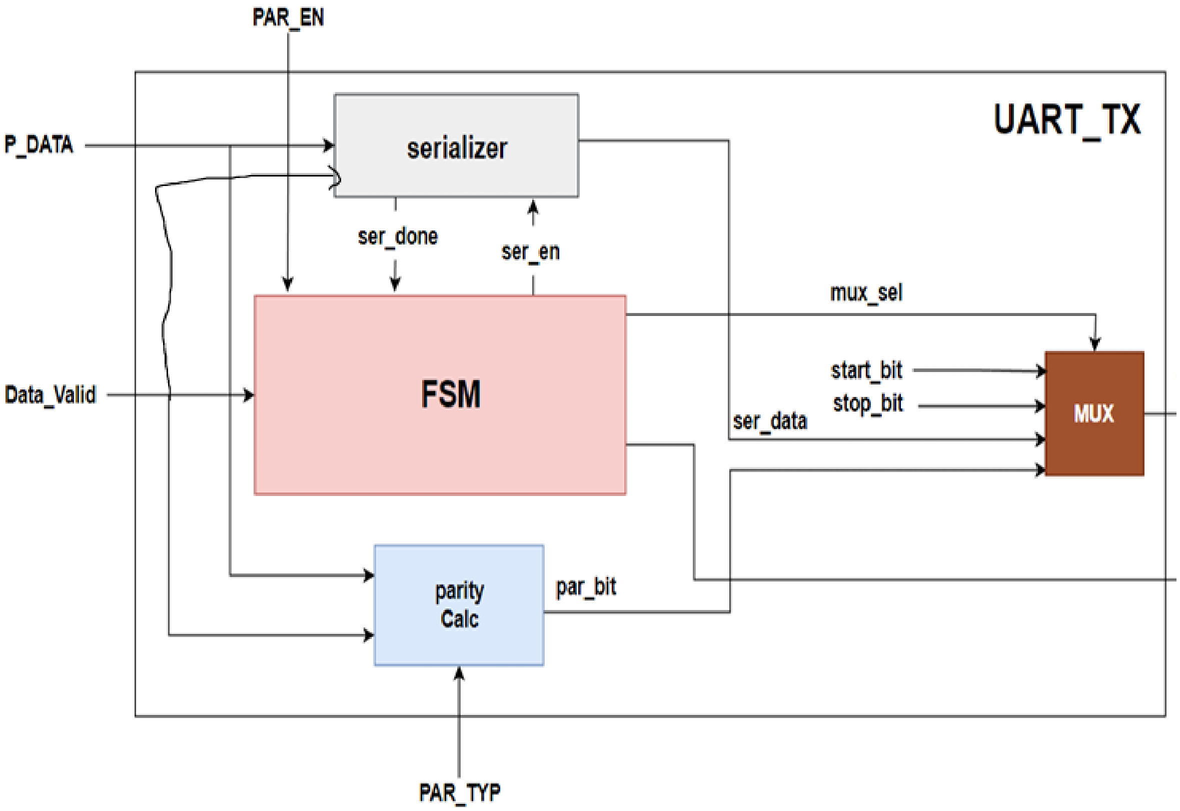


Expected Output:



Main Modules

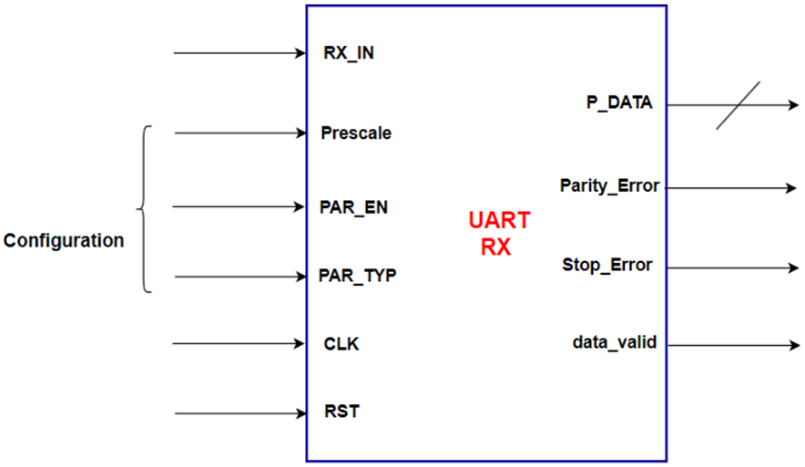
TX Block Diagram



Main Modules

UART receiver (Rx)

The UART Receiver converts serial data back into parallel form, checks for errors, and validates received data.



Signal name	Direction	Description
CLK	Input	System clock signal
RST	Input	Synchronized reset signal
PAR_TYP	Input	Parity type: 0 (Even), 1 (Odd)
PAR_EN	Input	Parity enable: 0 (Disable), 1 (Enable)
Prescale	Input	Oversampling prescaler (6-bit)
RX_IN	Input	Serial data input
P_DATA	Output	8-bit parallel data output
Data_valid	Output	High when valid data is received
Parity_Error	Output	High when a parity error is detected
Stop_Error	Output	High when a stop bit error is detected

Main Modules

RX Functional Description

- Receives and decodes UART frames from **RX_IN**.
- Supports oversampling rates of 8, 16, and 32.
- If **PAR_EN** is enabled, the parity bit is checked against the received data.
- Errors are flagged using **Parity_Error** and **Stop_Error**.
- Extracted data is sent to **P_DATA** with **Data_valid** asserted.

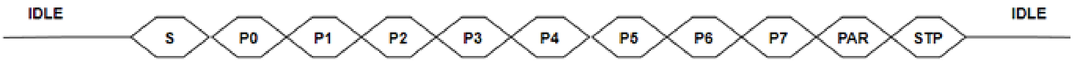
  RX Frame Formats

  1. Parity Enabled (Even/Odd)
     - Start bit (0)
     - 8-bit data (LSB first)
     - Parity bit (Even or Odd)
     - Stop bit (1)
  2. Parity Disabled
     - Start bit (0)
     - 8-bit data (LSB first)
     - Stop bit (1)

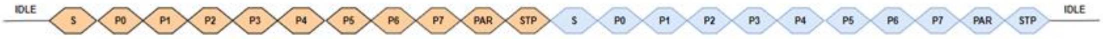
Waveforms

Expected input:

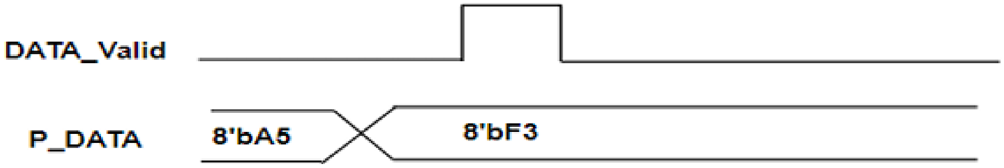
a. One Frame:



b. Consequent frames:



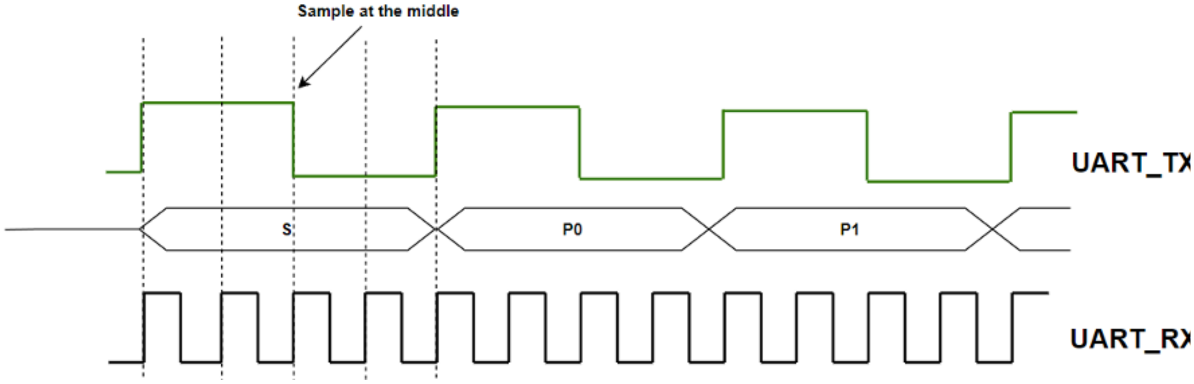
Expected Output:



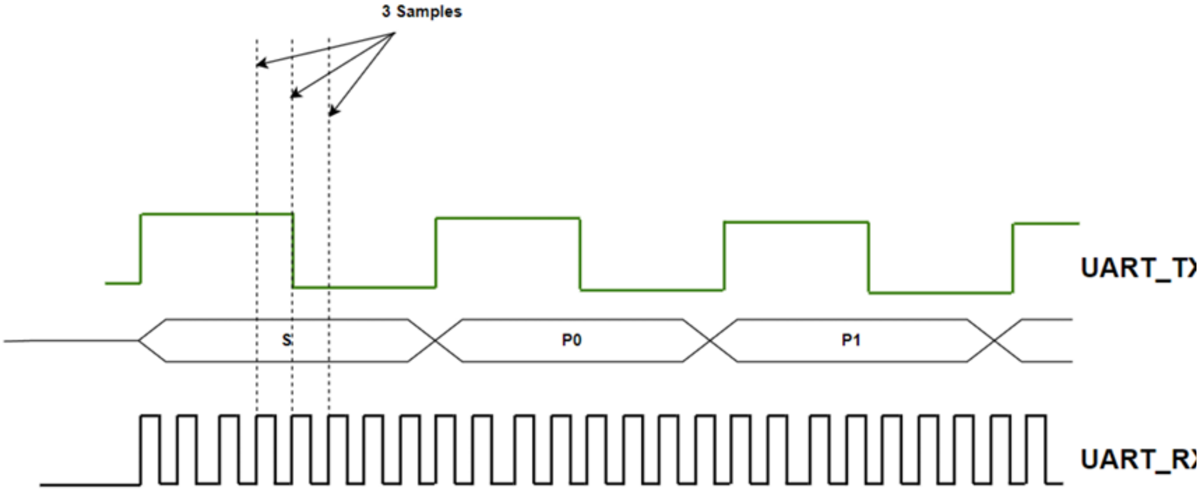
Main Modules

Oversampling

Oversampling by 4: This means that the clock speed of UART_RX is 4 times the speed of UART_TX.

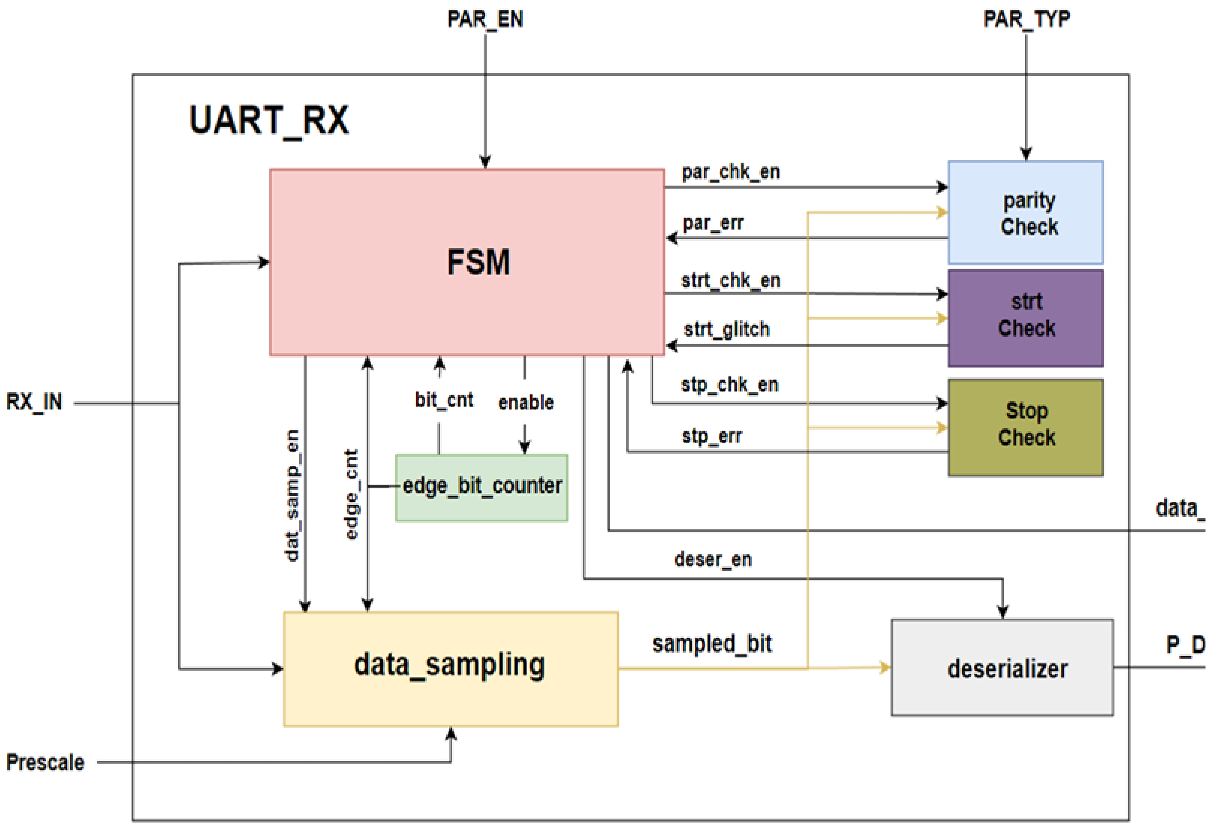


Oversampling by 8: This means that the clock speed of UART_RX is 8 times the speed of UART_TX



Main Modules

Rx Block Diagram



Simulation

Simulation and Verification

Simulation Steps:

1. **TX Module Testing:**
   - Send 8-bit data with different parity settings.
   - Verify the serial transmission matches the expected frame format.
2. **RX Module Testing:**
   - Receive frames and check data reconstruction.
   - Validate parity and stop bit error detection.
3. **Verify UART operation with continuous frames.**

Deliverables:

1. Snippets from the waveforms captured from QuestaSim for the design with inputs assigned values and output values visible.
2. Synthesis snippets for each encoding
   - Schematic after the elaboration & synthesis
   - Synthesis report showing the encoding used
   - Timing report snippet
   - Snippet of the critical path highlighted in the schematic
3. Implementation snippets for each encoding
   - Utilization report
   - Timing report snippet
   - FPGA device snippet
4. Snippet of the “Messages” tab showing no critical warnings or errors after running elaboration, synthesis, implementation and a successful bitstream generation.