# AXI4-Stream UART IP Core - Design Specification Document

## 1. Introduction

The AXI4-Stream UART IP Core provides a bridge between an AXI4-Stream interface and a UART serial communication interface. This IP enables seamless integration between high-speed on-chip data buses and external serial communication links, supporting both transmission (TX) and reception (RX) of data. It follows AXI4-Stream handshaking (tvalid, tready, tdata) for reliable communication and uses configurable baud rate generation via a programmable prescaler.

## 2. Features

* Full-duplex asynchronous communication (independent TX and RX).
* Configurable data width (default: 8 bits).
* Programmable baud rate using 16-bit prescale value.
* AXI4-Stream compliant valid/ready interface for data flow.
* Error detection: Frame error and Overrun error in RX path.
* Single clock domain operation with synchronous reset.
* Easily portable for FPGA or ASIC design environments.

## 3. Architectural Overview

The AXI4-Stream UART IP consists of three main modules: UART\_TX, UART\_RX, and a top-level UART wrapper. The TX converts parallel data into serial form, while RX converts incoming serial data into parallel form. The top-level module integrates both and provides AXI4-Stream compliant interfaces.

## 4. Module Interface

The top-level UART module includes the following signals:

|  |  |  |  |
| --- | --- | --- | --- |
| Signal Name | Direction | Width | Description |
| clk | Input | 1 | System clock for synchronous operation. |
| rst | Input | 1 | Active-high synchronous reset. |
| s\_axis\_tdata | Input | DATA\_WIDTH | AXI4-Stream transmit data input. |
| s\_axis\_tvalid | Input | 1 | Indicates valid transmit data. |
| s\_axis\_tready | Output | 1 | Indicates transmitter ready to accept new data. |
| m\_axis\_tdata | Output | DATA\_WIDTH | AXI4-Stream receive data output. |
| m\_axis\_tvalid | Output | 1 | Indicates valid received data. |
| m\_axis\_tready | Input | 1 | Indicates receiver ready to accept data. |
| rxd | Input | 1 | UART serial data input line. |
| txd | Output | 1 | UART serial data output line. |
| tx\_busy | Output | 1 | Transmitter active status. |
| rx\_busy | Output | 1 | Receiver active status. |
| rx\_overrun\_error | Output | 1 | Indicates RX overrun error. |
| rx\_frame\_error | Output | 1 | Indicates RX frame error. |
| prescale | Input | 16 | Configures baud rate timing. |

## 5. UART Transmitter (UART\_TX)

The UART\_TX module converts AXI4-Stream input data into a serialized UART frame. It adds a start bit (logic 0), transmits data bits (LSB first), and ends with a stop bit (logic 1).

Operation Steps:

1. Wait for s\_axis\_tvalid and s\_axis\_tready handshake.
2. Load transmit data into the internal shift register.
3. Send start bit followed by data bits and stop bit based on prescale timing.
4. Assert busy during transmission; deassert when complete.

## 6. UART Receiver (UART\_RX)

The UART\_RX module monitors the serial line for start bit detection, samples incoming bits at the center of each bit period, and reconstructs the received byte into parallel form. It checks for framing and overrun errors.

1. Detect falling edge (start bit) on RX line.
2. Sample each bit in the middle of its period using prescale counter.
3. Shift in data bits into parallel register.
4. Check stop bit validity to ensure frame correctness.
5. Raise frame\_error or overrun\_error when applicable.

## 7. Baud Rate Configuration

The baud rate is configured through the 'prescale' input. The baud rate is determined by the formula:  
Baud Rate = Clock Frequency / (Prescale × 8)  
For example, if the system clock is 100 MHz and prescale = 1301, the baud rate is approximately 9600 bps.

## 8. Error Handling

The receiver reports two key error types:

1. Frame Error – Indicates missing stop bit or invalid frame end.

2. Overrun Error – Occurs when a new frame arrives before previous data is read.

## 9. Design Guidelines

* Use a single synchronous clock for TX and RX to avoid metastability.
* Ensure prescale value is configured before data transmission.
* Handle tvalid/tready handshake strictly to prevent data loss.
* Monitor status flags for debugging and reliable data flow.
* Perform loopback testing by connecting TXD to RXD during simulation.

## 10. Revision History

Version 1.0 – October 2025 – Initial version of AXI4-Stream UART IP specification.