UART Component Functional Description

Ridge Redding

December 10, 2025

UART Component Descriptions

UART Transmitter

The UART transmitter is responsible for converting 8-bit parallel data from the CPU into a serial stream for transmission on the `TxD` line. Transmission begins when the CPU asserts `WR` while `CE_UART` is high, causing the lower 8 bits of `WriteData[31:0]` to be latched into an internal transmit buffer. A control FSM then sets `TxRDY` low to indicate that the transmitter is busy, and the transmission begins with a logic-low start bit. The buffered data is loaded into an 8-bit shift register, which serializes the data starting from the least significant bit (LSB). A parity generator concurrently computes the even parity of the data stream. After all 8 bits are transmitted, the parity bit is sent, followed by a stop bit (logic-high). The `TxD` line reflects these states. Finally, the FSM reasserts `TxRDY` to indicate readiness for a new transmission cycle.

UART Receiver

The UART receiver monitors the `RxD` line for a falling edge, which signals the start of an incoming transmission. A dedicated start bit detector samples the line using a 16x oversampled `BaudClock`, confirming the presence of a valid start bit if 8 consecutive low samples are received. The receiver then begins sampling data bits every 16 baud ticks and shifts them into a register, LSB first. After all 8 bits are received, a parity bit is sampled and checked against the internally computed parity. If a mismatch occurs, the `RxParityErr` flag is set. Upon completion, the FSM sets `RxRDY` high to signal that valid data is available. The CPU can then read from the receiver by asserting `RD` with `CE_UART` high, which places the received byte into the lower 8 bits of `ReadData[31:0]` and clears `RxRDY`. If `CE_UART` is low, the output is tri-stated.

UART Status Register

The status register is a 32-bit register that provides visibility into the current state of the UART system, specifically the transmitter and receiver. It contains three important flags in its least significant bits: `TxRDY` (bit 0), `RxRDY` (bit 1), and `RxParityErr` (bit 2). These signals are sourced from the transmitter and receiver FSMs and are updated asynchronously. On the rising edge of the clock, if `Enable` is asserted, the register latches the current status values. When `OE` (output enable) is high, the contents are driven onto the `ReadData[31:0]` bus; otherwise, the output is tri-stated. All upper bits `[31:3]` are permanently zero, simplifying status decoding by the CPU.

UART Baud Generator

The baud generator is a timing-critical component that provides clock signals for controlling the rate of data transmission and reception. Internally, it uses a 5-bit counter that increments on every system clock cycle. The most significant bit (MSB) of the counter is assigned to the signal `BaudClock`, which is used by both transmitter and receiver FSMs to time state transitions, such as shifting and sampling bits. The least significant bit (LSB) is assigned to `BaudClock16`, used for finer-grain oversampling or detecting transitions such as the start bit. On system reset, the counter is initialized to zero. This modular and scalable design allows UART communication to operate at a fraction of the system clock frequency, which is essential for matching baud rates between devices without shared clocks.

I/O Signal Summary

The UART module uses several key input and output signals:

- `Clock, Reset`: Standard synchronous interface.
- `WR, RD`: Indicate write or read operations by the CPU.
- `CE_UART, CE_SR`: Enable signals for the UART and status register respectively.
- `WriteData[31:0]`: Input data from CPU; only `[7:0]` is used by transmitter.
- `ReadData[31:0]`: Output data bus; reflects received data or status register.
- `TxD`: Serial output line from transmitter.
- `RxD`: Serial input line to receiver.

The UART is memory-mapped at `0x3000` for data transmit/receive and `0x3004` for status. Correct usage requires polling the appropriate status flags before writing to or reading from the UART.