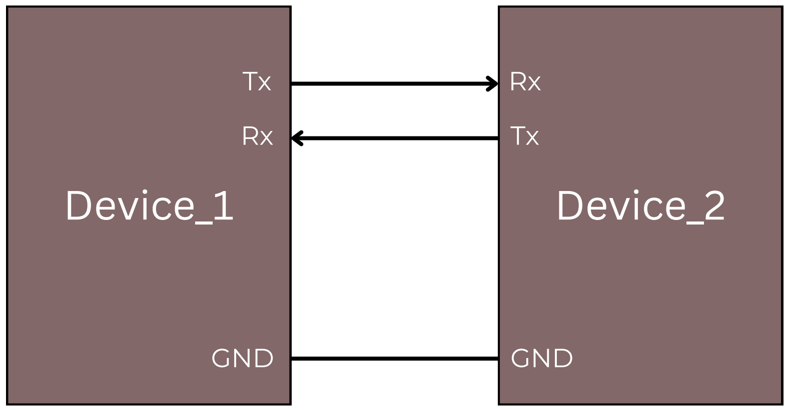
Author: Mudassir Ahmed



UART CORE

DESIGN

www.https://github.com/mudassir-dv/UART_Core.com

Introduction

The Universal Asynchronous Receiver-Transmitter (UART) communication protocol plays a crucial role in facilitating serial communication between devices. Its widespread use in various applications stems from its simplicity and flexibility.

Below are five important points regarding the UART protocol that highlight its key features and advantages.

1. **Asynchronous Communication:** UART is a widely used protocol for asynchronous serial communication between devices. This means that it does not require a shared clock signal between the transmitting and receiving devices, allowing for flexibility in connecting different systems with varying clock speeds.
2. **Simple Hardware Interface:** The simplicity of UART's hardware interface is one of its key strengths. It typically requires only two main signal lines for data transmission: TX (transmit) and RX (receive). This minimal wiring makes it an attractive option for connecting microcontrollers, sensors, and other peripherals in embedded systems.

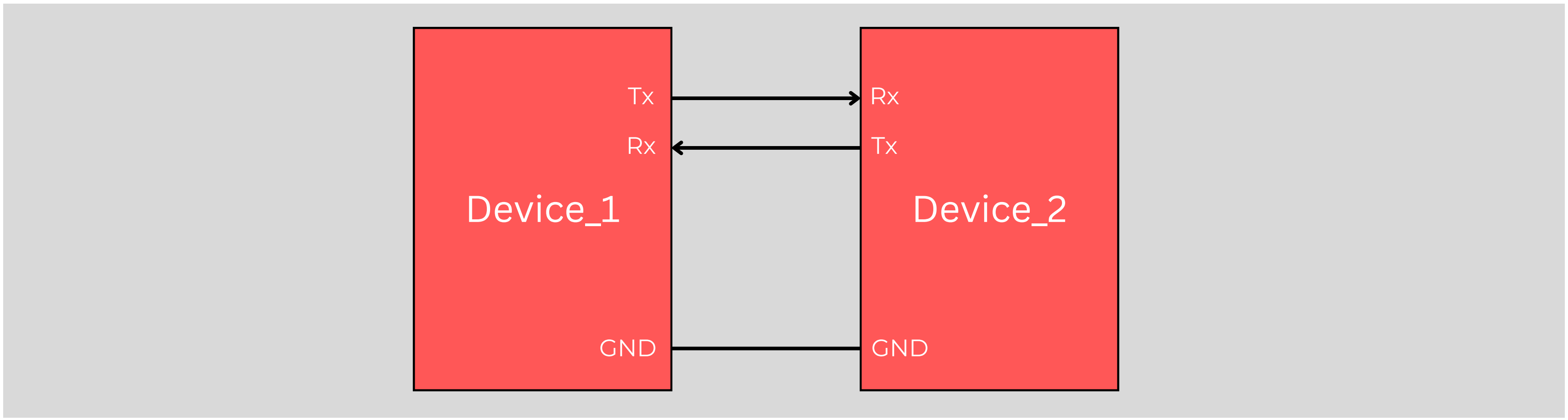
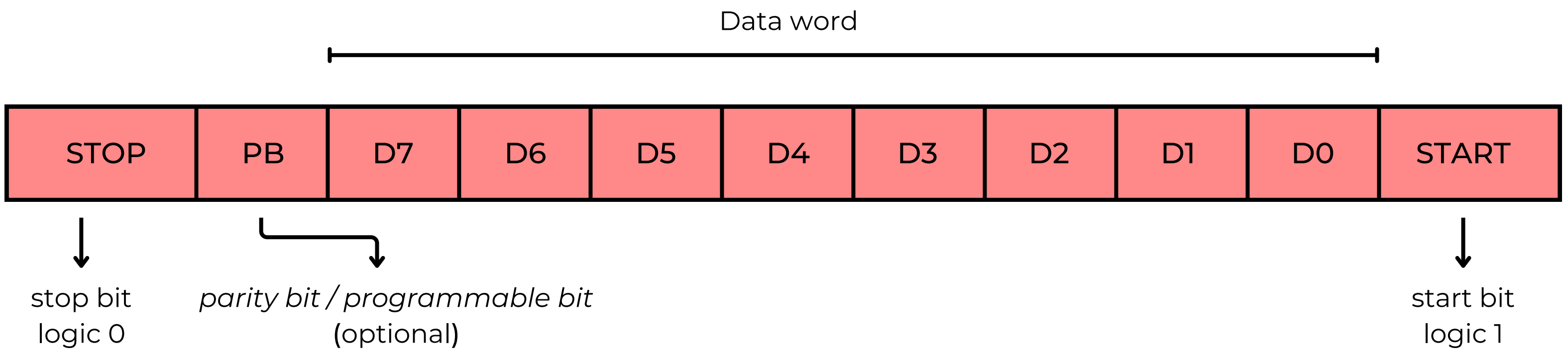


Fig (a) Basic UART communication

Introduction

- **Configurable Data Frame**: UART communication involves the transmission of data in the form of a frame, which can be configured to include various elements such as start bits, data bits, parity bits, and stop bits. This configurability allows users to tailor the protocol to meet specific data integrity and format requirements.



- **Baud Rate Flexibility**: The baud rate, which determines the speed of data transmission, can be adjusted in UART communication. This flexibility enables devices to communicate effectively over short or long distances, making UART suitable for a range of applications from short-distance device interfacing to long-distance communication in industrial settings.

- **Error Handling Capabilities**: While UART is simple, it includes basic error handling mechanisms, such as parity bits, to detect errors in transmitted data. These features help ensure data integrity, although additional error correction may be needed in more complex or noise-prone environments.

Signal List

NAME	TYPE	DESCRIPTION
clk	input	Rising edge of clk is used for all uart module operations
reset	input	Asynchronous active LOW reset signal
tx	input	Transmitter line used to transmit the data
w_data [7:0]	output	8 - bit data to be transmitted
wr_uart	input	Write Enable signal to write the data to be transmitted
tx_fifo_full	output	If HIGH indicates the Tx FIFO is full, If LOW indicates otherwise

Signal List

NAME	TYPE	DESCRIPTION
rx	input	Receiver line used to receive the data
r_data [7:0]	output	8 - bit data to be read once the rx_fifo_empty goes LOW
rd_uart	input	Read Enable signal to read the data received
rx_fifo_empty	output	If HIGH indicates the Rx FIFO is empty, If LOW indicates otherwise
rx_fifo_empty	output	If HIGH indicates the Rx FIFO is empty, If LOW indicates otherwise

Schematic

UART_TOP

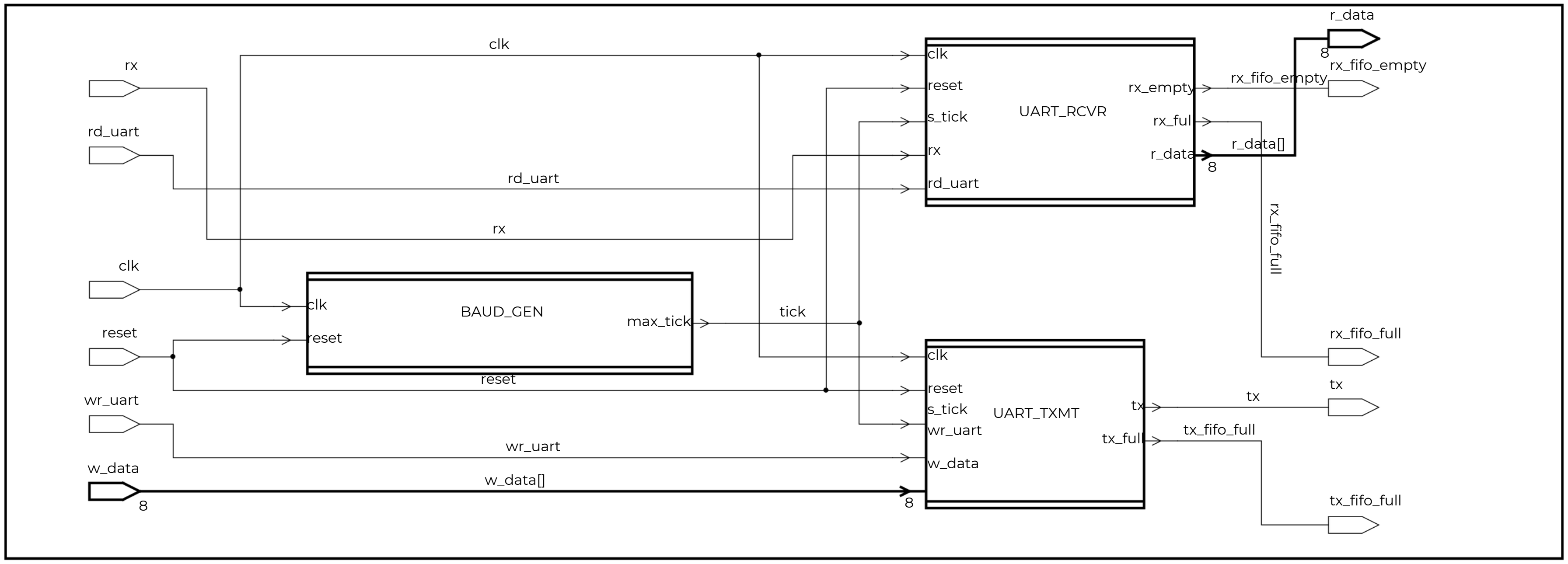


Fig (b) UART Top Module