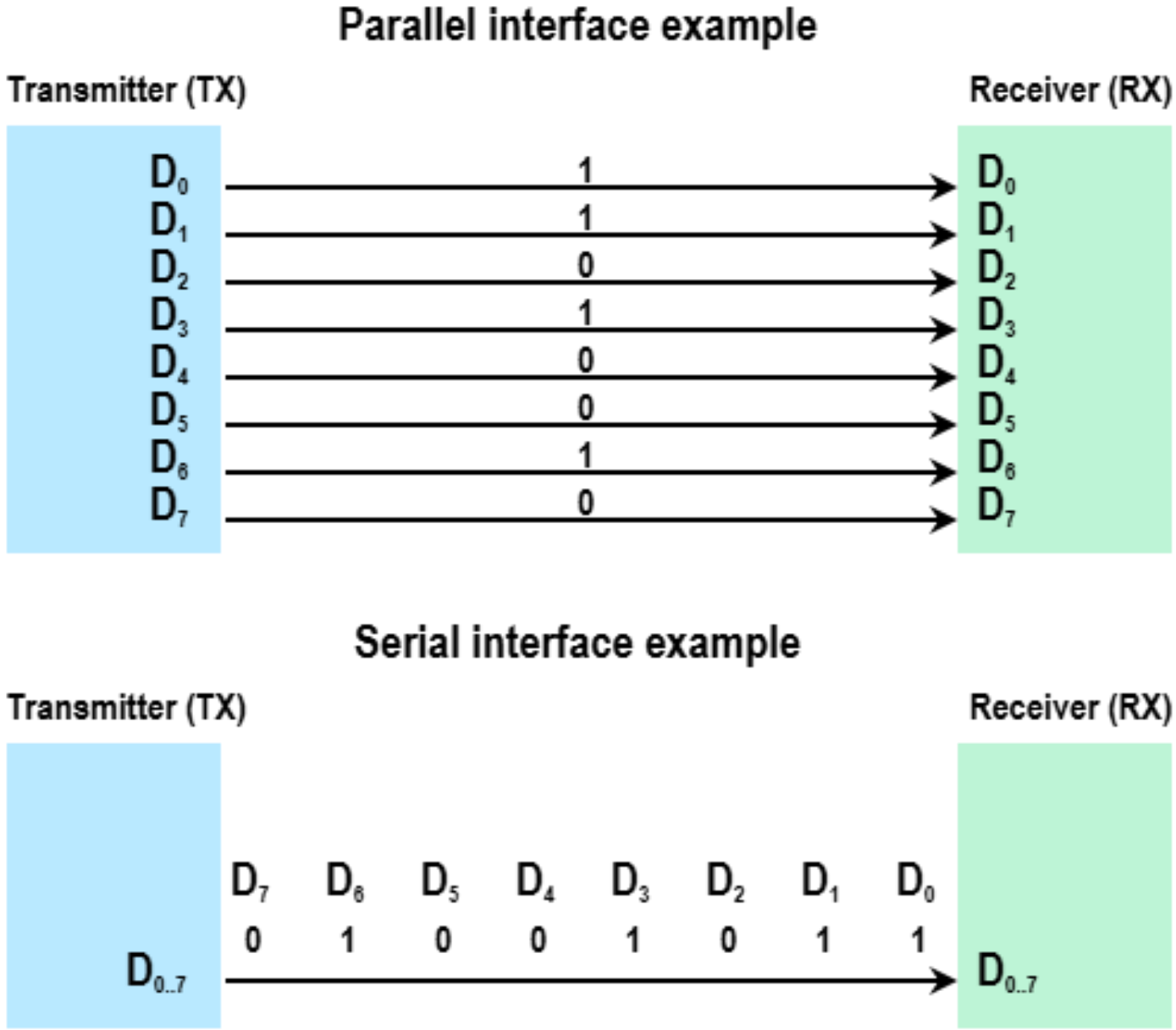
RTL Design of UART

Parallel vs. Serial Communication

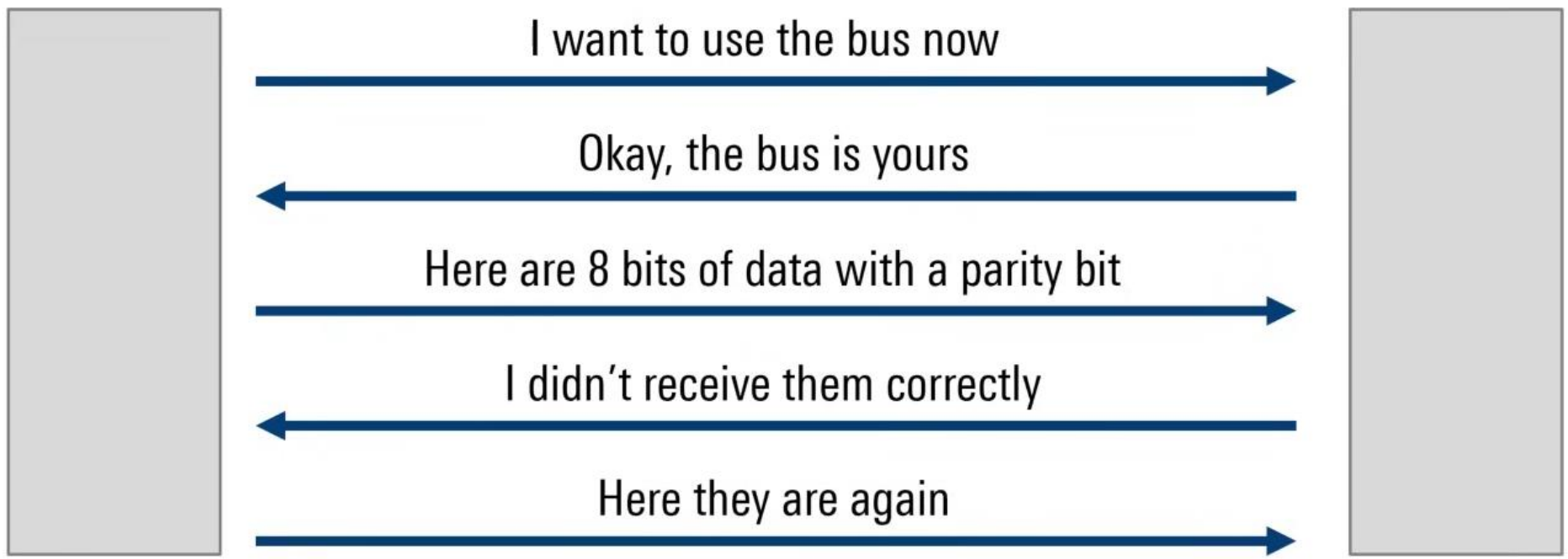


Feature	Parallel Communication	Serial Communication
Data Lines	Multiple	One or Two
Speed	Faster	Slower
Cost	High (more wires)	Low (fewer wires)
Distance	Short	Long
Complexity	High	Low

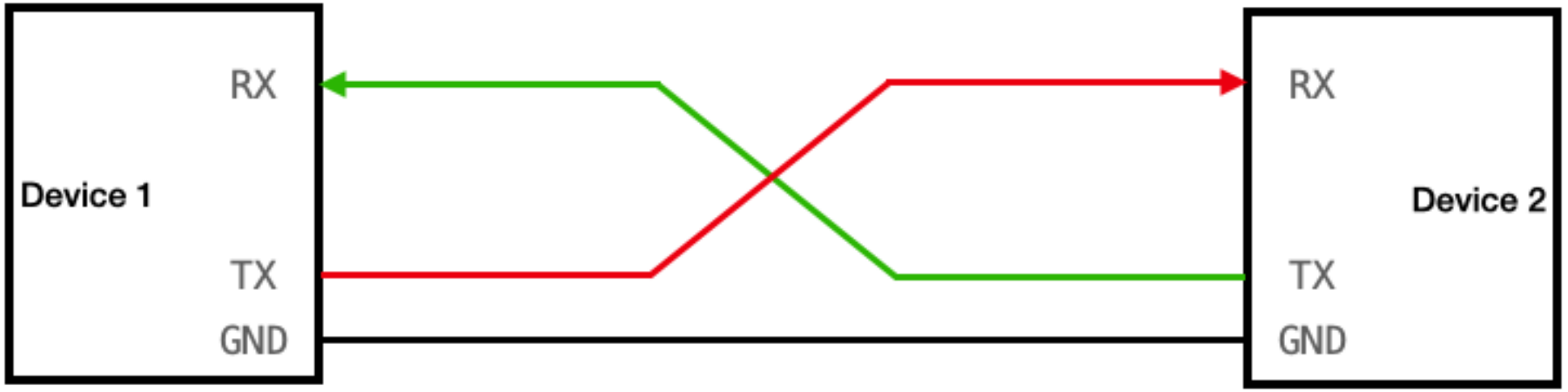
UART is a **serial** communication protocol.

About protocols

- Describe the **format and meaning of the bits** and **the rules for exchanging information**
  - Also often includes bus access / arbitration, error handling, etc.



Introduction to UART



UART (Universal Asynchronous Receiver Transmitter) is a hardware communication protocol

- **Asynchronous**
- **Serial**
- **Peer-to-Peer**
- **Low speed, Short distance**

UART is widely used in embedded systems, microcontrollers, and FPGA-based designs due to its simplicity and reliability.

Where is UART used?

- UART was one of the earliest serial protocols
  - Serial (COM) ports, RS-232, modems, etc.
- Popularity of UART is however decreasing:
  - SPI and I^2C between components
  - Ethernet and USB between computers and peripherals
- Still important for lower-speed, low-throughput applications

How UART Works

UART operates using two primary signals:

1. **TX (Transmit)**: Sends data out of the device.
2. **RX (Receive)**: Receives data into the device.

A UART transmitter (TX) sends bits sequentially, while the receiver (RX) reads them at the same baud rate. Both devices must **agree on the same communication settings** (baud rate, data format, parity, stop bits) for reliable data exchange.

About timing / synchronization

- UART is asynchronous – the transmitter and receiver do not share a common clock
- The transmitter and receiver therefore must:
  - Transmit at the same (known) speed
  - Use the same frame structure / parameters

Common UART baud rates
4800
9600
19200
57600
115200

The **baud rate** is the speed at which data is transmitted in UART, measured in **bits per second (bps)**. Common baud rates include:

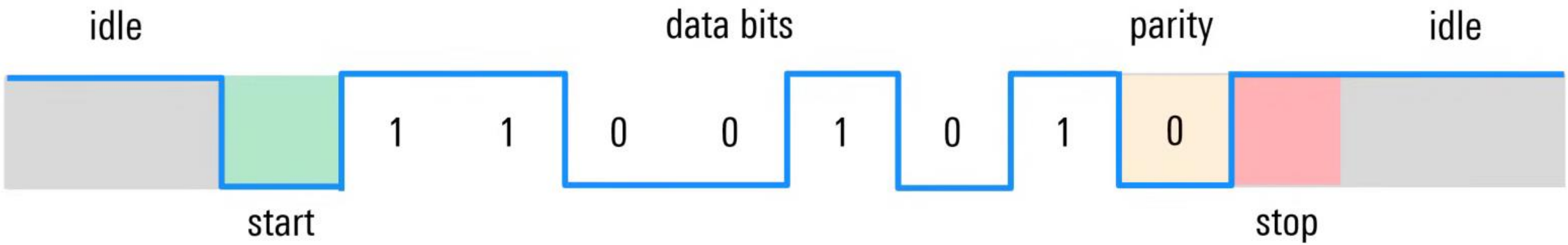
- **9600 bps**
- **115200 bps** (common in modern embedded systems)

Timing Constraints:

- The **baud rate generator** in UART must produce the correct **clock signal** to sample each bit at the right moment.
- If the receiver's baud rate is **not exactly the same** as the transmitter's, it may **misinterpret bits** and cause data corruption.

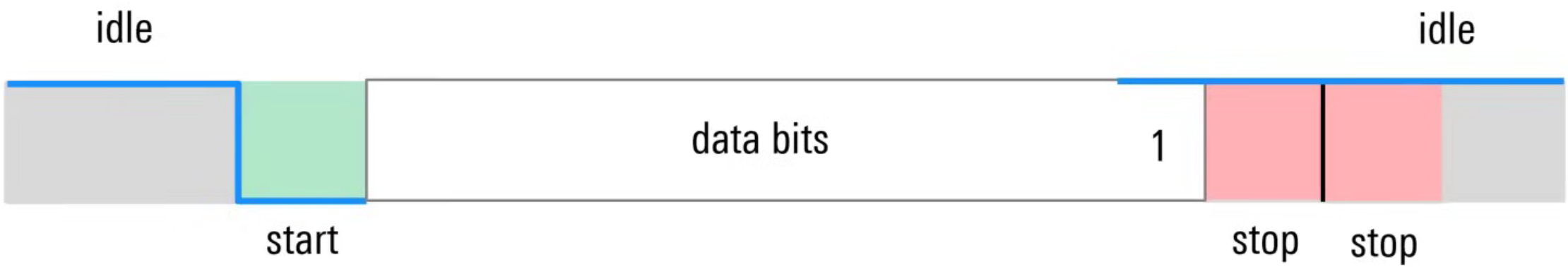
UART frame format

- UART frames consist of:
  - Start / stop bits
  - Data bits
  - Parity bit (optional)
- High voltage (“mark”) = 1, low voltage (“space”) = 0
- In the idle state, the line is held high



Start and stop bits

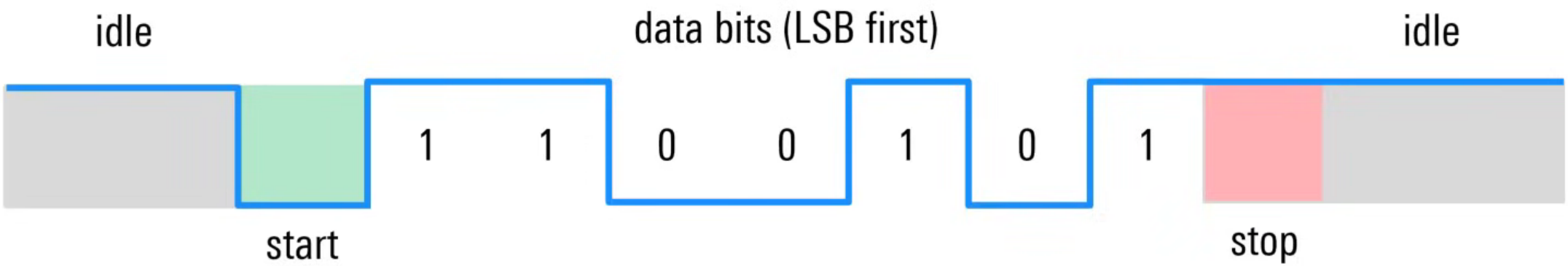
- The start bit indicates data is coming
  - Transition from idle (high) to low
- The stop bit(s) indicate data is complete
  - Stay / return to idle (high)
  - Second (optional) stop bit
    - Uncommon in practice



Data bits

- User (useful) data
- Length: 5 to 9 bits (usually 7 or 8)
- Data is typically sent with the least significant bit (LSB) first

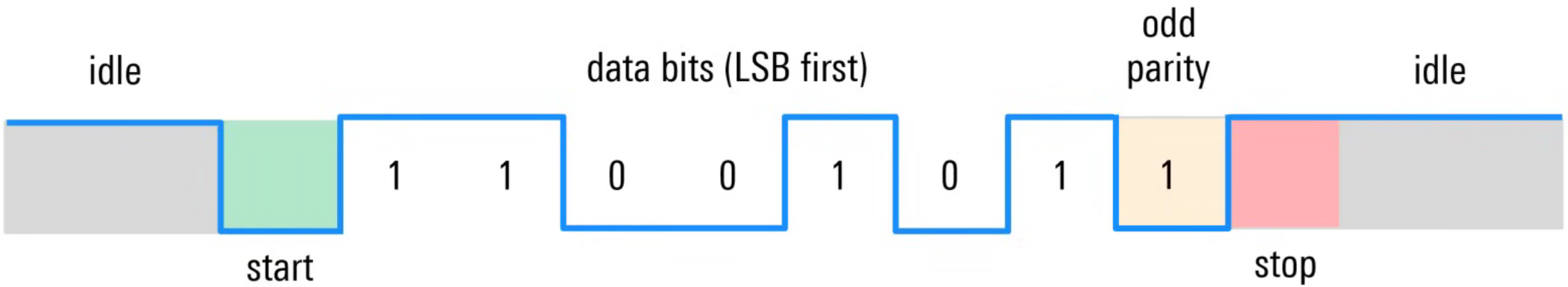
- Example:
  - 7-bit ASCII ‘S’ (0x53) = 1 0 1 0 0 1 1
  - LSB order = 1 1 0 0 1 0 1



Parity bit (optional)

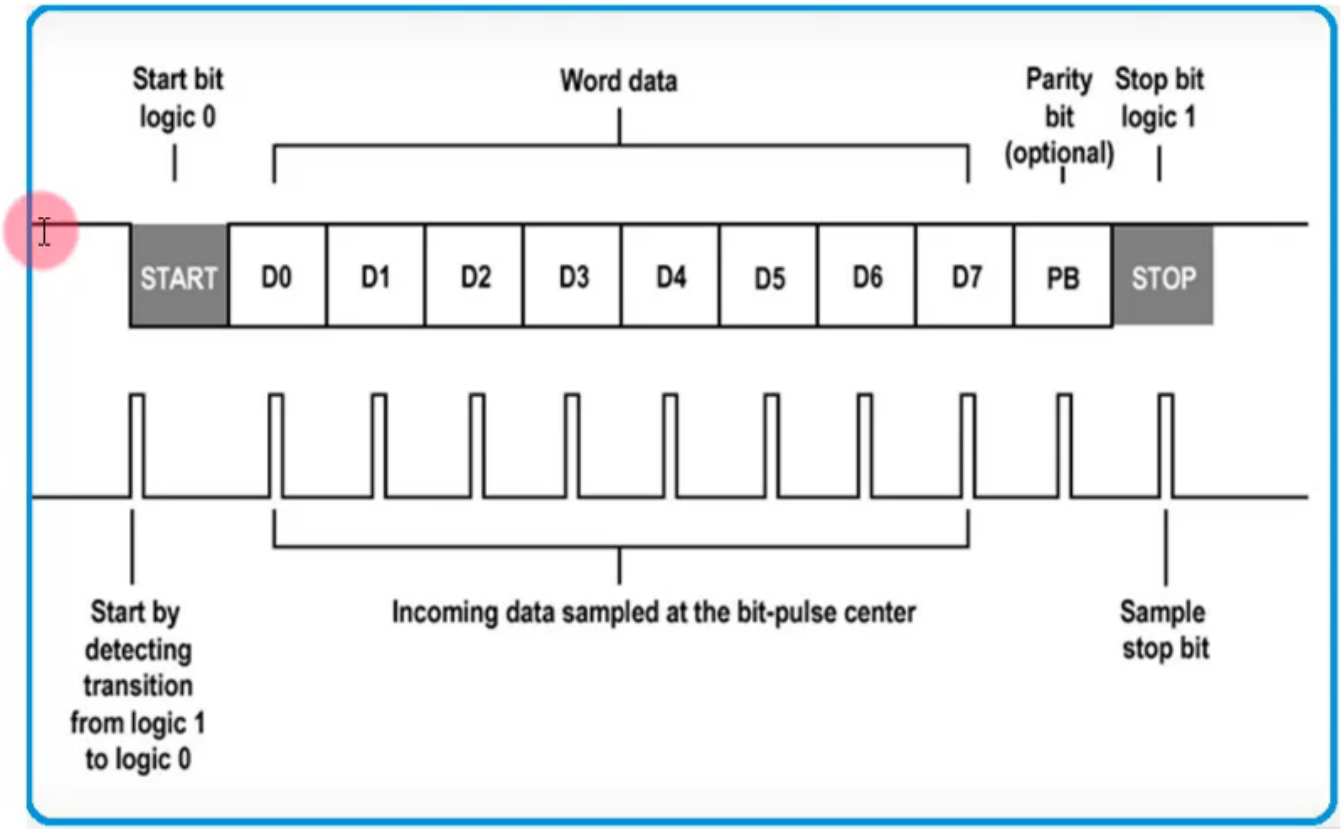
- Used for error detection
- **Even parity**: number of 1's must be even
- **Odd parity** : number of 1's must be odd
- Can detect a **single** flipped bit only

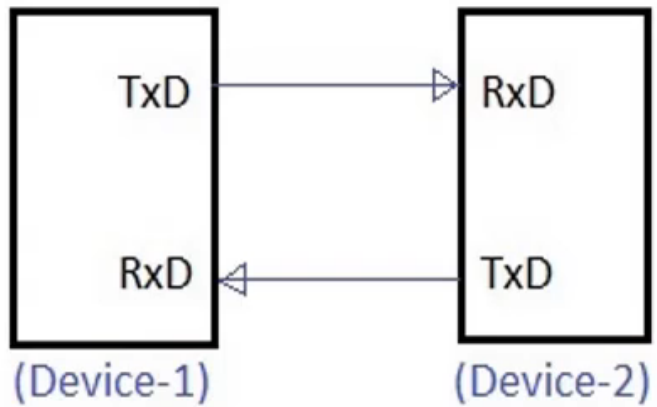
- Example:
  - ASCII 'S' (0x53 = 1 0 1 0 0 1 1) – four 1's
  - If even parity, parity bit is 0 (because number of 1's already even)
  - If odd parity, parity bit is 1 (to make the number of 1's odd)



Summary

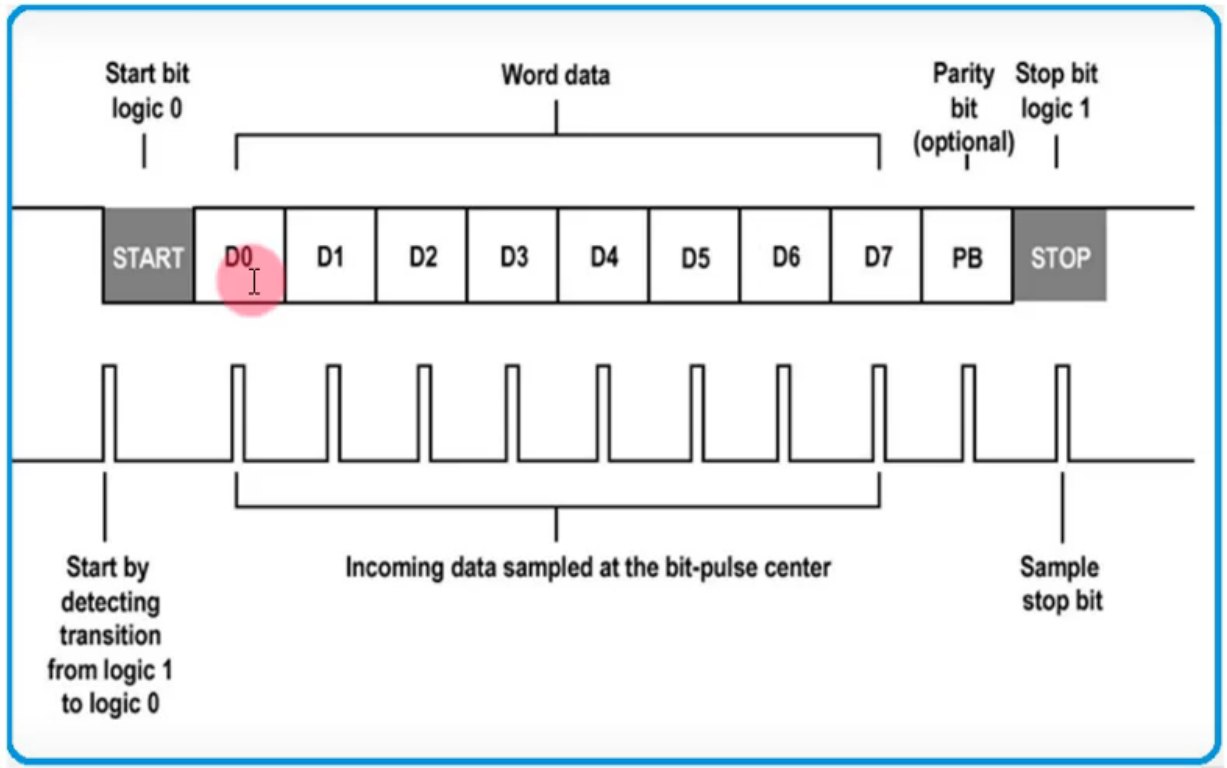
- UART = universal asynchronous receiver / transmitter
  - Simple, two-wire protocol for exchanging serial data
  - No shared clock – baud rate must be configured the same on both sides
  - Start / stop bits used to frame user data
  - Optional parity bit for detecting single-bit errors
- Widely used serial data protocol, but slowly being replaced by SPI, I^2C, USB and Ethernet

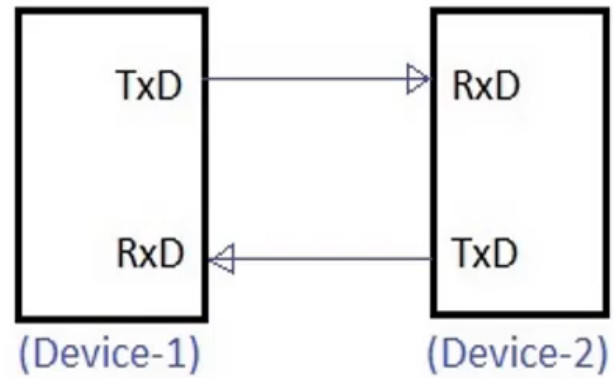




UART Interface Diagram

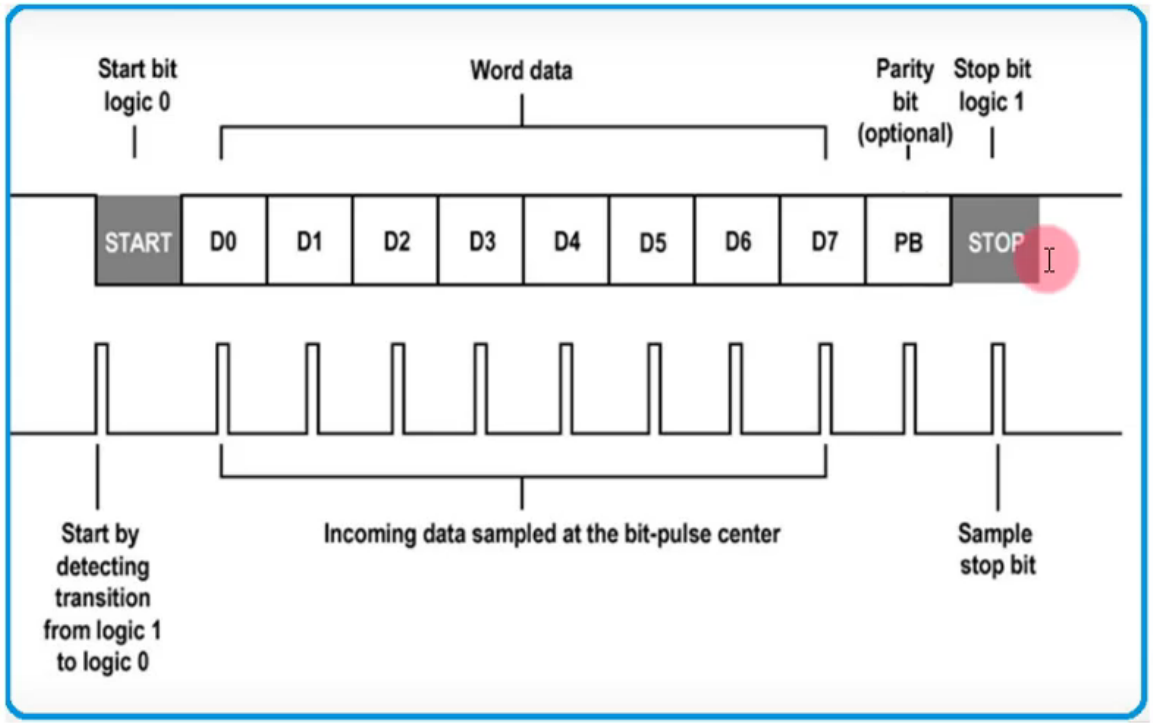
IDLE value for tx is Active High

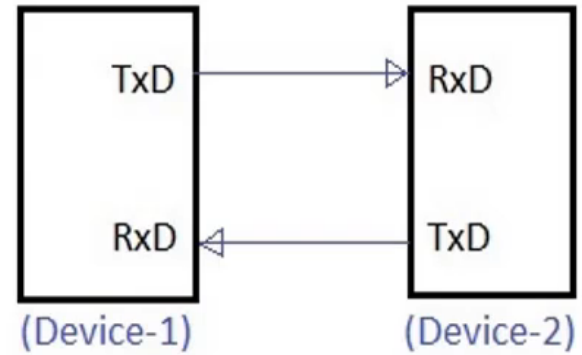




UART Interface Diagram

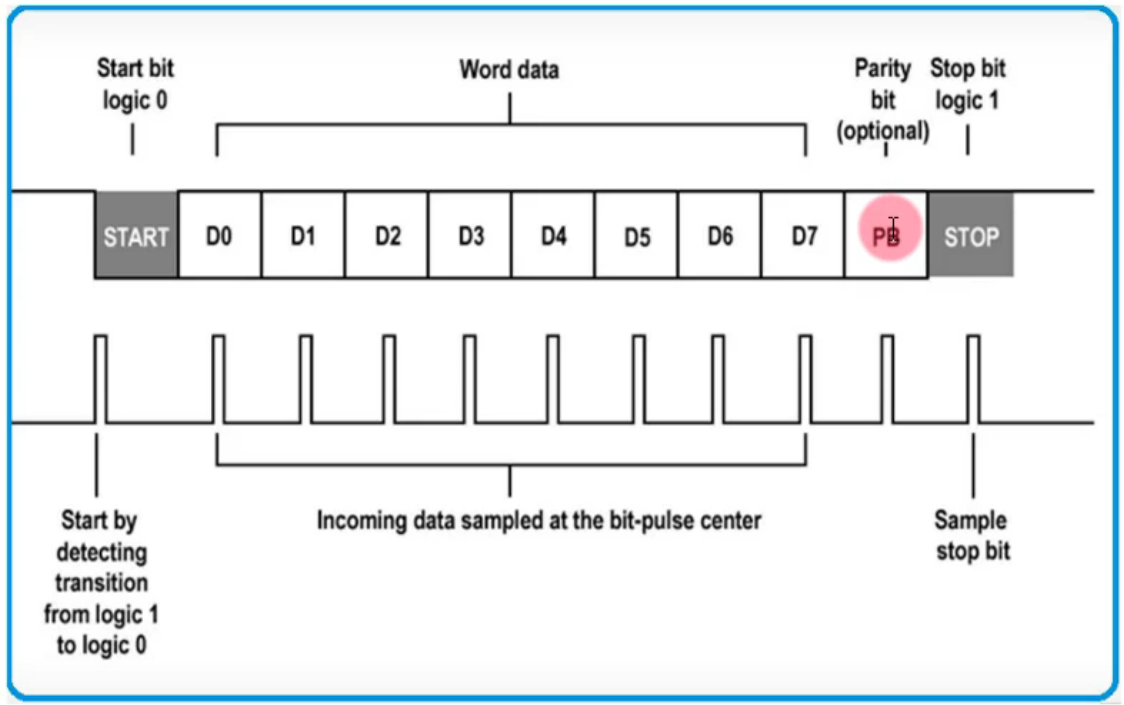
Data length could be varied between 5 to 9
Common Data length = 8

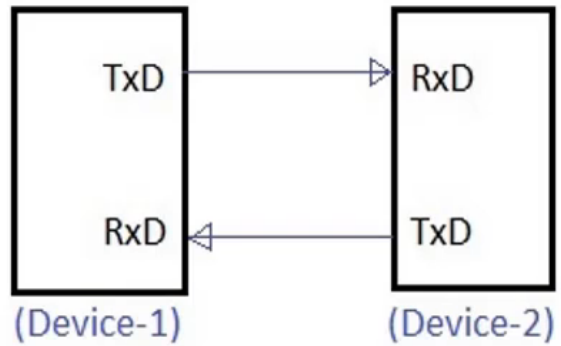




UART Interface Diagram

Stop bit could be one or two bit long.
Common stop bit length : single bit

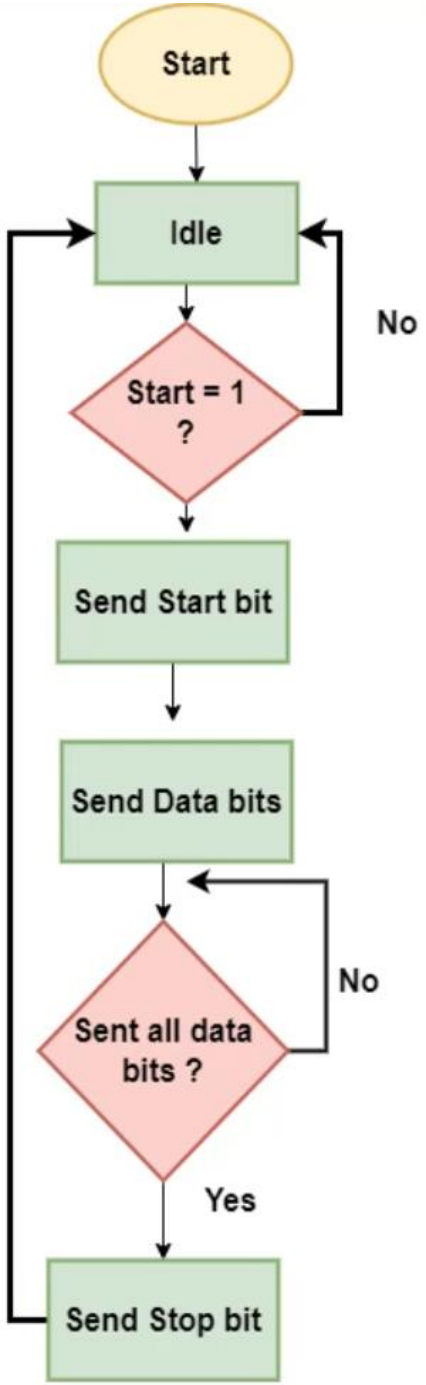




UART Interface Diagram

Parity bit is optional. We could also choose type of parity between even or odd.
Common parity : odd

Tx Algorithm



Rx Algorithm