UART - An Asynchronous Serial Interface
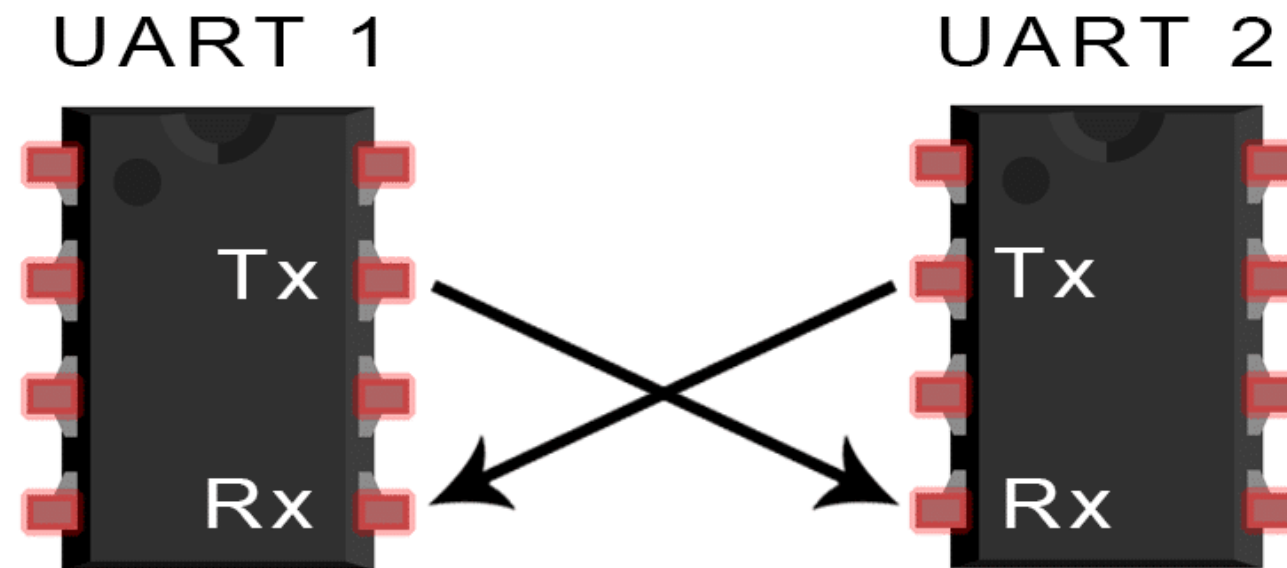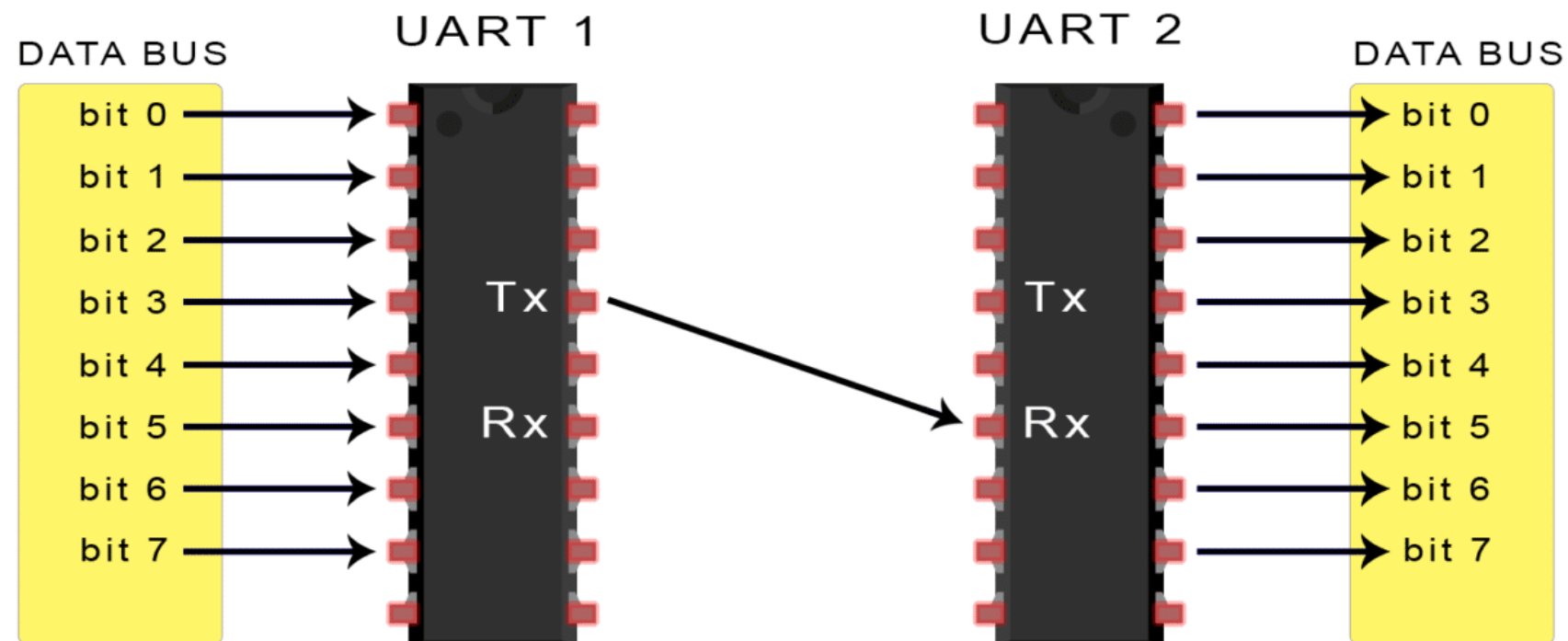
- UART stands for Universal Asynchronous Receiver Transmitter
- Used ubiquitously in serial ports such as RS-232 and in embedded systems

- UART is **asynchronous** — *no explicit clock signal sent from Tx to Rx*
- UART is **point-to-point** (i.e., *one Tx device and one Rx device*)
- UART data is sent in **data frames with specific frame format**

- Prerequisite for UART communication is that Tx and Rx must agree on:
    - Data rate (baud rate in bps)
    - Data frame format

- **Key Idea:** Use the pre-agreed upon data rate and frame format to do **local clock recovery** (i.e., generate a clock signal at the Rx to sample bits on the line)

UART Basics

UART 1   UART 2

Tx   Tx

Rx   Rx

Tx pin of device 1 connected to Rx pin of device 2

DATA BUS   UART 1   UART 2   DATA BUS

bit 0   bit 0
bit 1   bit 1
bit 2   bit 2
bit 3   Tx   Tx   bit 3
bit 4   bit 4
bit 5   Rx   Rx   bit 5
bit 6   bit 6
bit 7   bit 7

Parallel data serialized at device 1 and de-serialized at device 2

UART Basics - Data Frame Format

Packet

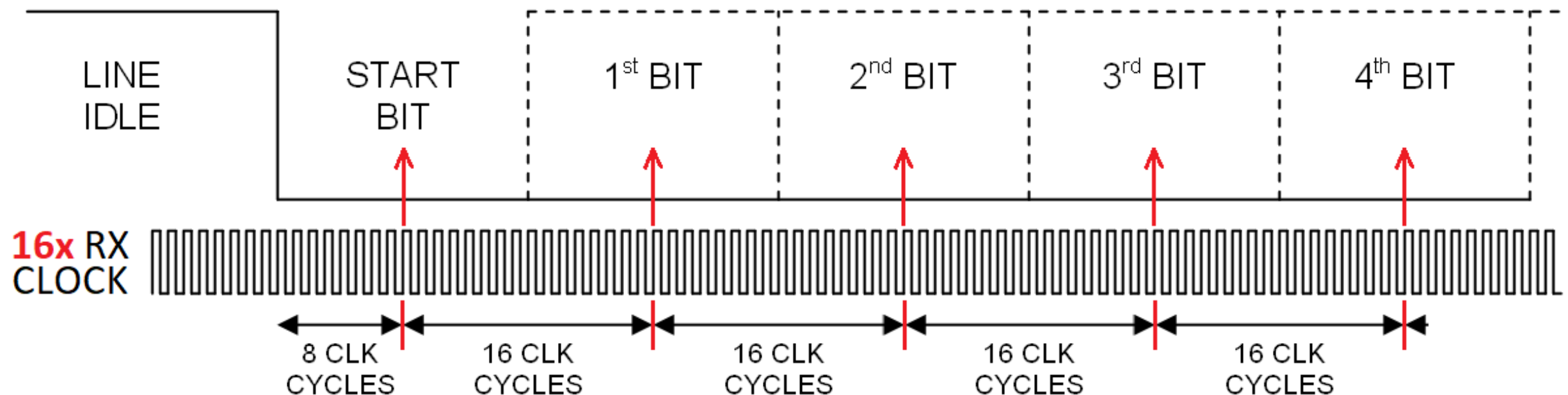| 1 start bit | 5 to 9 data bits | 0 to 1 parity bits | 1 to 2 stop bits |

Data Frame

Always '0'

Always '1'
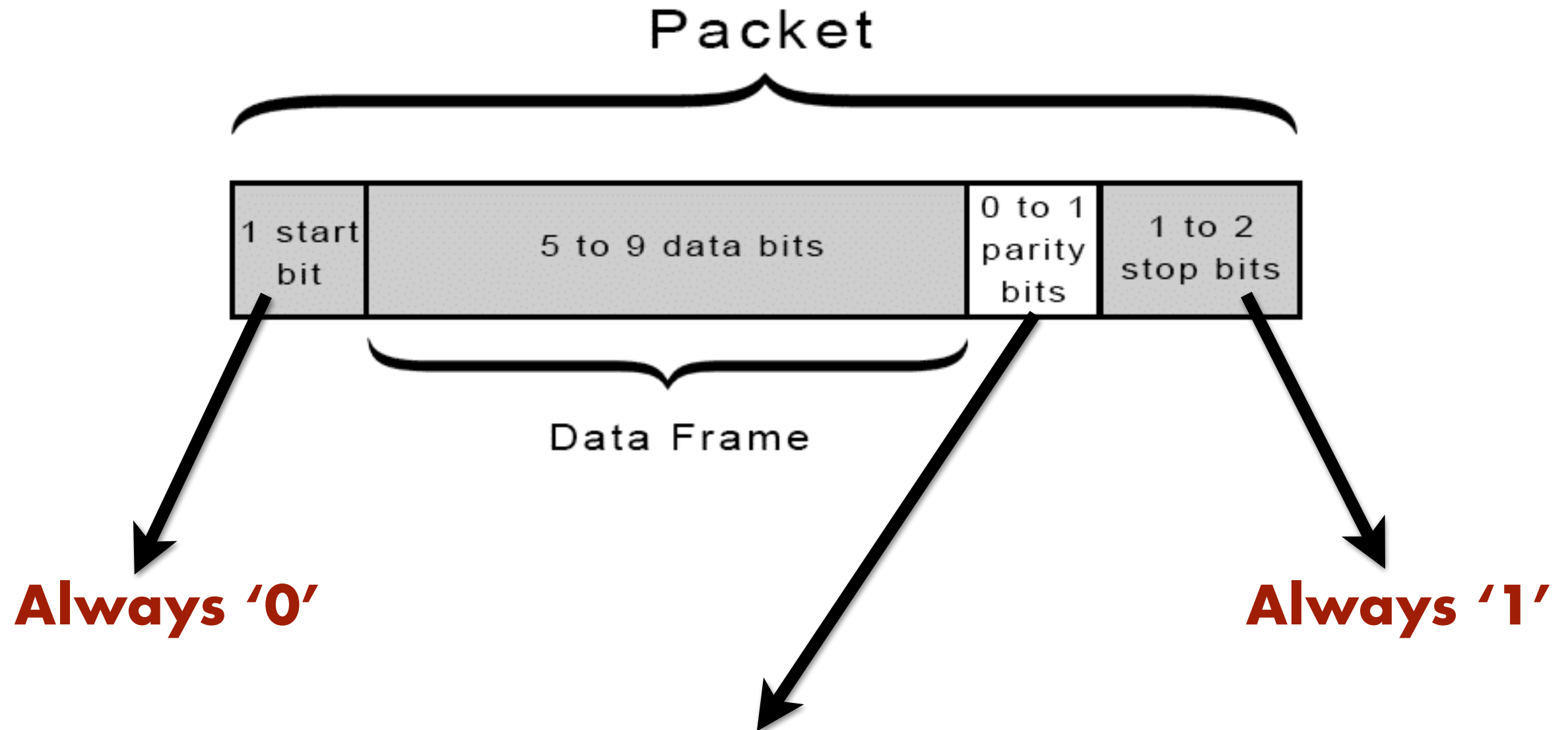
Odd parity (1): Total number of 1's in data frame should be odd
**Even parity (0):** Total number of 1's in data frame should be even
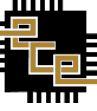
UART Basics - Synchronization

- Receiver device generates a local clock with frequency that is a multiple of the baud rate (typically 8x the baud rate or 16x the baud rate)

- **<u>Key step for UART to work is that the data line is held HIGH (i.e., logic 1) when there is no data being sent</u>**

- When a START bit is sent (remember, a START bit is always 0), there will be a 1 -> 0 transition (i.e., negative edge) on the data line

LINE IDLE	START BIT	1st BIT	2nd BIT	3rd BIT	4th BIT

16x RX CLOCK

8 CLK CYCLES	16 CLK CYCLES	16 CLK CYCLES	16 CLK CYCLES	16 CLK CYCLES

- After locking on to the middle of a the start bit, the Rx device samples every 16 bits to get to the middle of the next bit
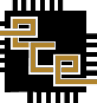
UART Basics

- A common data format is 8-N-1 (8 data bits, No parity bits, 1 stop bit), commonly used to transmit ASCII character data
- With this format, character transmission rate is the 1/10 of baud rate

- Can be full-duplex (independent transmit chain from device 2 to device 1)

Advantages

- Only 2 wires (Tx and Rx) for two-way communication
- No clock signal needed
- Has basic error checking

Disadvantages

- Needs clock recovery
- Needs serialization and de-serialization of data
- Only point to point (no multiple masters, no multiple slaves)
- Clocks on the devices need to be close to each other (~5%)