Project UART

UART

1. What is UART protocol ?

- A set of rules and regulations is called a protocol.
- UART stands for a Universal Asynchronous Receiver and Transmitter.
- UART Protocols is a serial communication with two wired protocols.
- The data cable signal lines are labelled as Rx and Tx.
- Serial communication is commonly used for transmitting and receiving the signal.
- It transfers and receives the data serially bit by bit without class pulses

UART

2. What Is the Frame Format of UART Protocol?

- UART is a half-duplex protocol.
- Half duplex means transferring and receiving the data but not at the same time.
- The transmitter receives the 8-bit data from input and stores in some register.
- The transmitter then adds some special bits like start bit, stop bit and parity bit(if selected) to make the complete data frame.
- The data frame then sent out serially by the transmitter at the predefined clock rate (baud rate).

UART

- The receiver is by default at high logic state which indicates idle state of receiver and keeps looking for the high to low transition i.e. start bit.
- As soon as the start bit is detected, the receiver observe the start bit for 50% of the receiving baud rate, if it is the receiver start sampling other data bits at the middle of each bit otherwise receiver set flag for framing error.
- After detecting the 8-bit data, the receiver then looks for the parity bit which is generated by the transmitter for the single bit error detection.
- If the parity bit is detected properly, the receiver looks for the stop bit to stop the reception of data.
- After the successful detection of stop bit the receiver line goes high logic state to indicate idle state and start looking for the next start bit.

UART

3) How Does Communication Occur Between Transmitter and Receiver in UART?

- UART operates without a clock signal, relying on a predefined baud rate for synchronization.
- The transmitter serializes parallel data and appends a start bit, optional parity bit, and stop bit to create a complete data frame.
- The frame is transmitted bit by bit over the TX line.
- The receiver detects the start bit to synchronize with the incoming frame. It sequentially reads the data bits, checks parity for errors (if enabled), and identifies the stop bit.
- The receiver then reconstructs the original parallel data by stripping the additional bits

UART

4) Flowchart representing Transmitter and Receiver Communication in UART

Transmitter converts parallel data to serial.

↓

Start bit added to signal the start of data transmission.

↓

Data bits sent sequentially over the TX line.

↓

Parity bit (optional) added for error checking.

↓

Stop bit added to mark the end of the frame.

↓

Receiver detects the start bit and synchronizes.

↓

Receiver reads the data bits and checks the parity (if present).

↓

Receiver removes start, parity, and stop bits.

↓

Receiver reconstructs the original parallel data.

UART

5)What Is a Baud Rate Generator in UART and How Does It Work?

- The baud rate generator ensures that the transmitter and receiver
- operate at a common communication speed.
- It takes a high-frequency input clock signal and divides it by a
- predefined divisor value.
- The divisor value is chosen to match the desired baud rate (e.g., 9600,
- 115200 bps).
- The resulting lower-frequency output clock synchronizes the timing for
- data transmission and reception in UART.
- Both devices must use the same baud rate for reliable communication.

UART

6) Synchronization Process:

1. **Baud Rate Agreement**

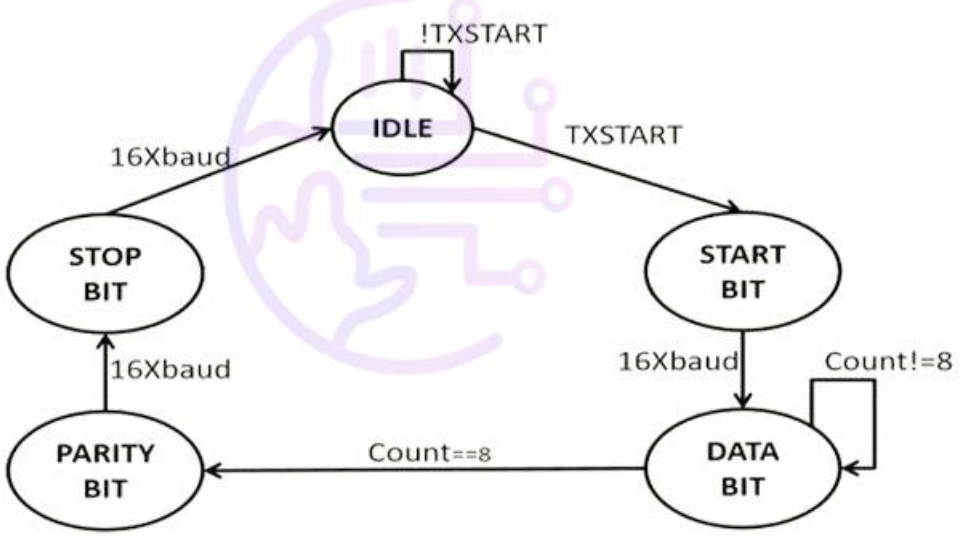
• Both devices are pre-configured to use the same baud rate (e.g., 9600, 115200 bps).
• Each device uses its own internal clock to match that baud rate.

2. **Start Bit Detection**
• Receiver monitors **RxD** line for a falling edge (logic high → low), which signals a start bit.
• This falling edge is used to align the receiver's internal sampling timer.

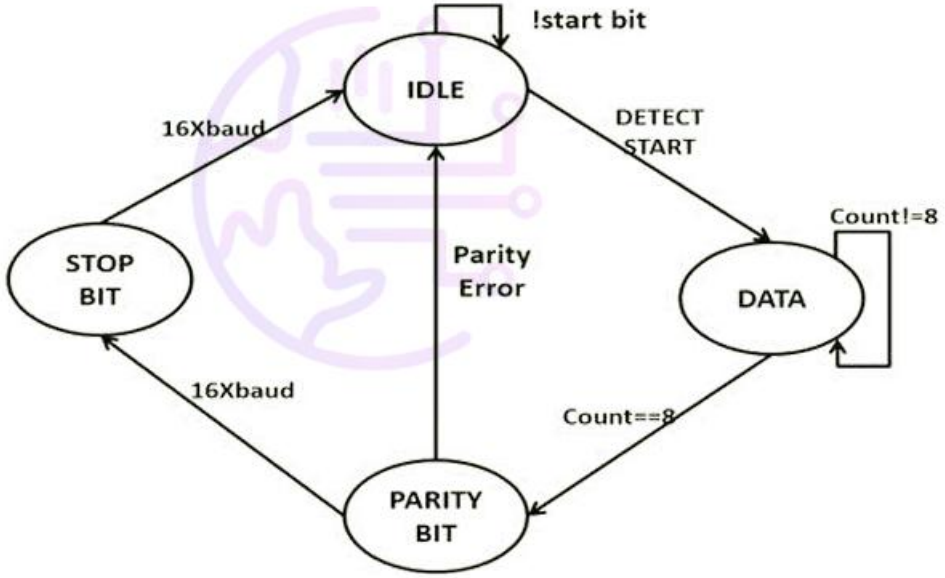
3. **Oversampling for Accuracy**
• Receiver often samples at 16× the baud rate to ensure correct timing.
• It samples the center of each bit using a mid-sample point (e.g., 8th tick out of 16).

4. **Bit-by-Bit Sampling**
• After detecting the start bit, the receiver samples the data bits at precise intervals based on its local clock.
• Stop bit confirms the end of transmission and validates synchronization

UART



Tx state machine



RX state machine

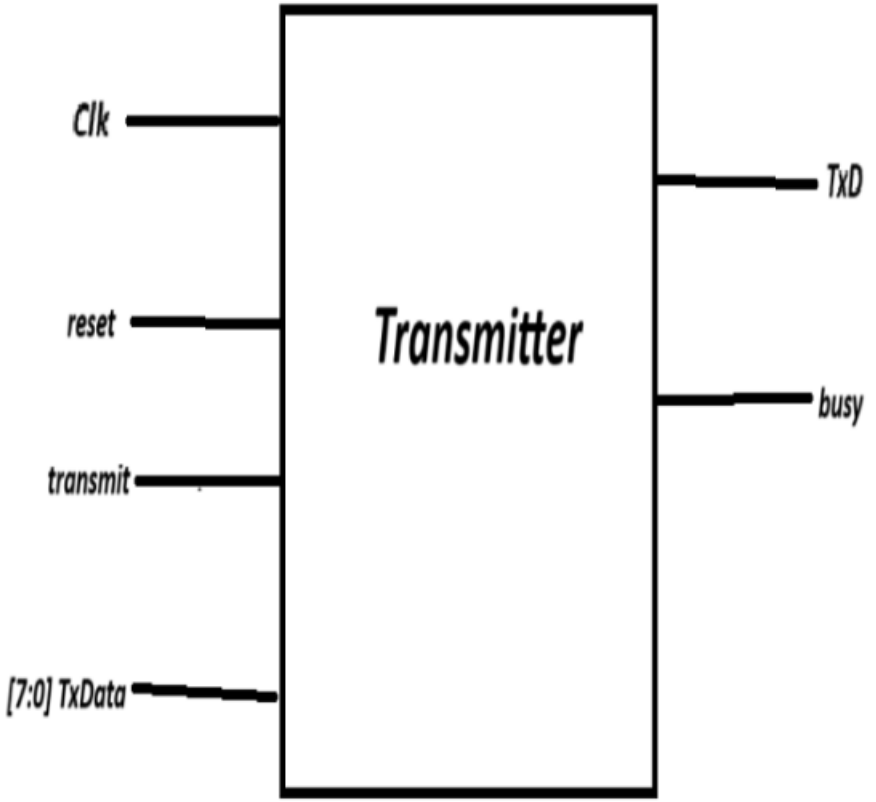
UART

1. Transmitter Design

- Input: 8-bit parallel data
- Adds: Start bit (0) , Data bits (LSB first) , Stop bit (1)
- Uses a shift register to serialize data.
- Tracks current bit index and asserts busy while transmitting.

2. Baud Rate Generator

- Generates timing pulses for both transmitter and receiver.
- Divides system clock to match the required baud rate.
- Receiver often samples at 16× baud for accuracy.

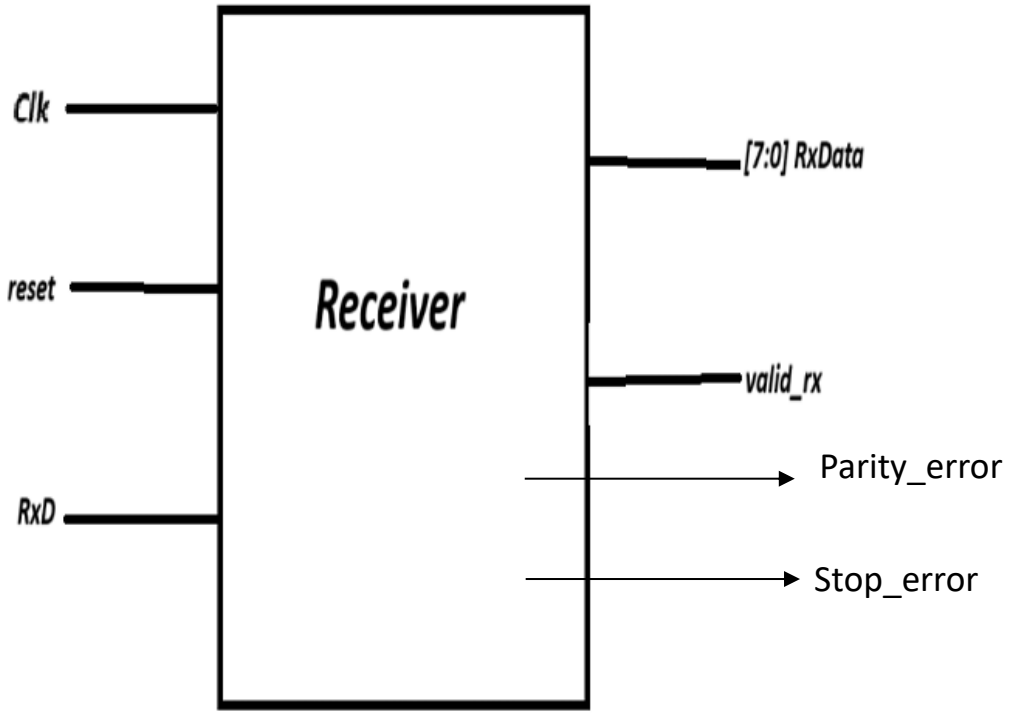


UART

3.Receiver Design

• Monitors the **RxD** line for the start bit.
• Samples data using a sample counter (e.g., 16× sampling).
• Stores incoming bits in a shift register.
• Asserts **valid_rx** when a full frame is received and validated.

- **parity_error** signal is high when the calculated parity bit does not equal the received frame parity bit as this means that the frame is corrupted.
- **stop_error** signal is high when the received stop bit does not equal 1 as
  this means that the frame is corrupted



4. UART Interface (Top Module)

- Integrates transmitter and receiver.
- Provides clean I/O to external system:
- Routes serial data through TxD and RxD lines.