UART Transmitter
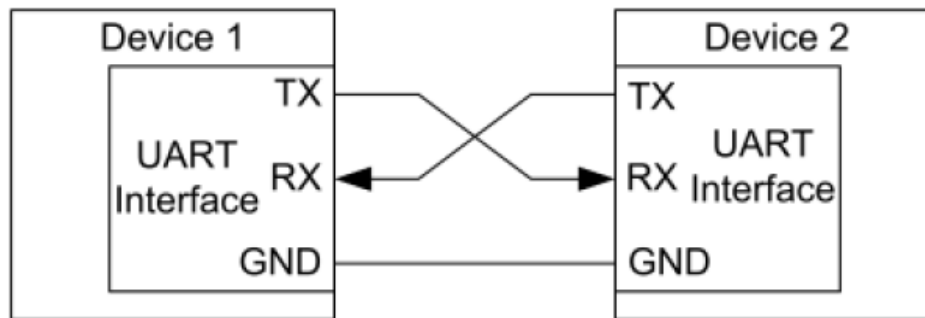
Introduction :-

- There are many serial communication protocols such as I2C, UART and SPI.
- A **U**niversal **A**synchronous **R**eceiver/**T**ransmitter (UART) is a block of circuitry responsible for implementing serial communication.
- UART is Full Duplex protocol (data transmission in both directions simultaneously)

- **Transmitting UART** converts parallel data from the master device (eg. CPU) into serial form and transmit in serial to receiving UART.
- **Receiving UART** will then convert the serial data back into parallel data for the receiving device.
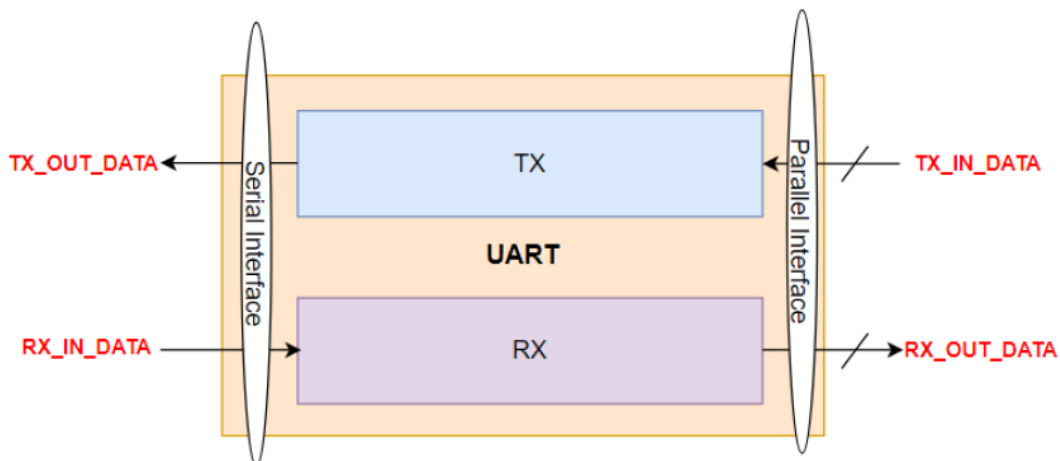
Block Interface (Top Module) :-
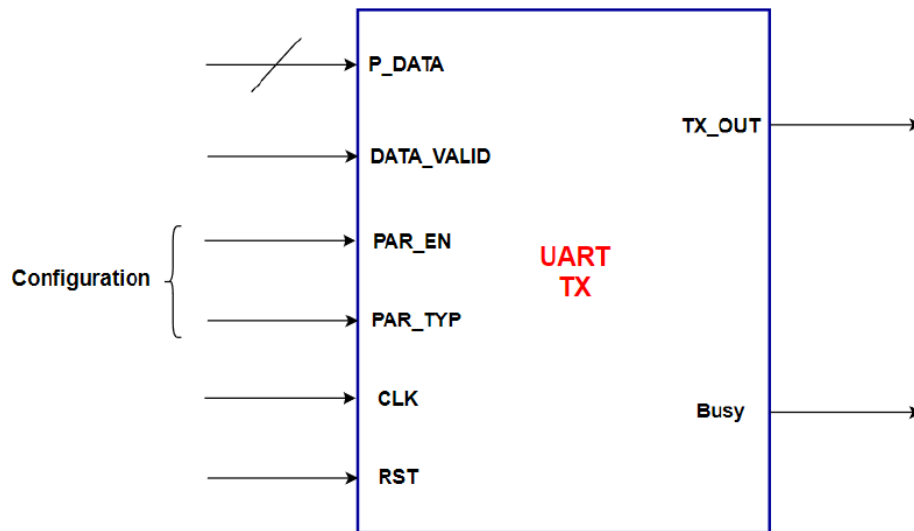
Port	Width	Description
CLK	1	UART TX Clock Signal
RST	1	Synchronized reset signal
PAR_TYP	1	Parity Type
PAR_EN	1	Parity_Enable
P_DATA	8	Input data byte
DATA_VALID	1	Input data valid signal
TX_OUT	1	Serial Data OUT
Busy	1	High signal during transmission, otherwise low

Specifications :-

- UART TX receive the new data on **P_DATA** Bus only when **Data_Valid** Signal is **high**.
- Registers are cleared using asynchronous active low reset
- **Data_Valid** is high for only 1 clock cycle
- **Busy** signal is **high** as long as UART_TX is transmitting the frame, otherwise **low**.
- UART_TX couldn't accept any data on **P_DATA** during UART_TX processing, however **Data_Valid** get high.
- **TX_OUT** is high in the **IDLE** case (No transmission).

PAR_EN (Configuration)

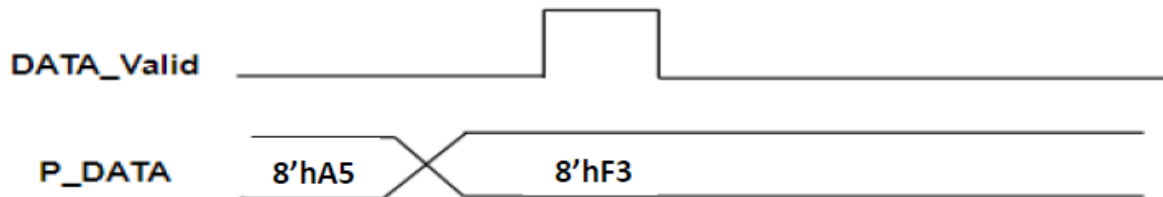0: To disable frame parity bit

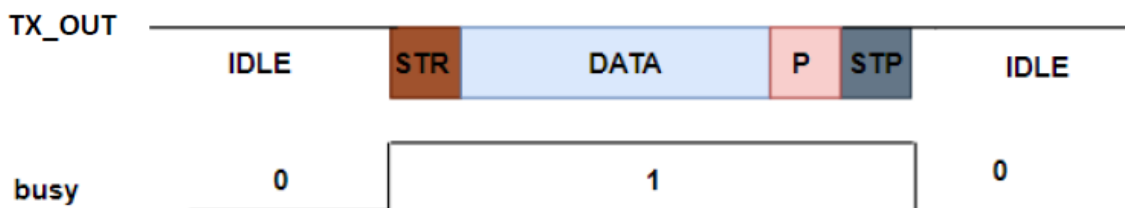1: To enable frame parity bit

PAR_TYP (Configuration)

0: Even parity bit
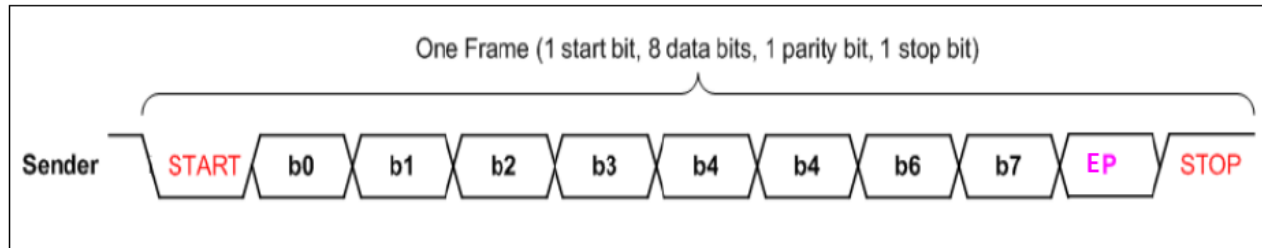
1: Odd parity bit

Waveform :-

Expected Input:-

Expected Output :-
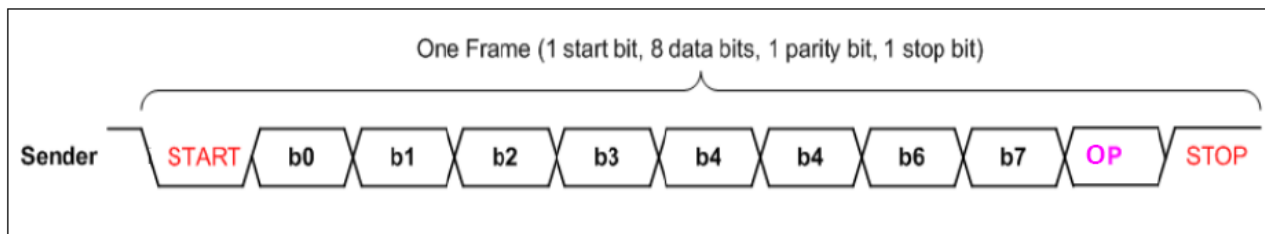
All Expected Output Frames

1. Data Frame (in case of Parity is enabled & Parity Type is even)

– One start bit (1'b0)

– Data (LSB first or MSB, 8 bits)

– Even Parity bit
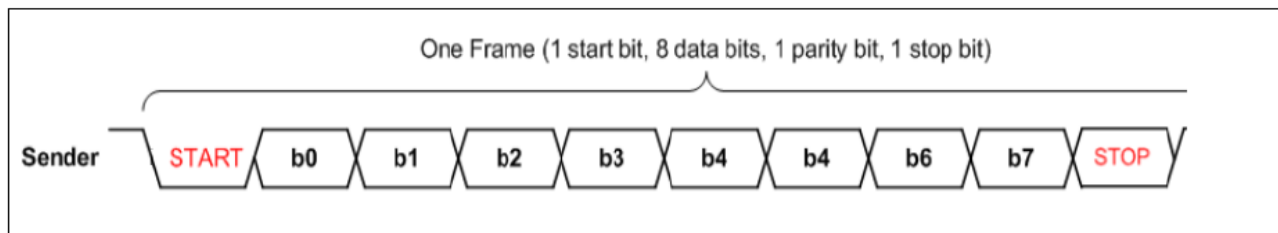
– One stop bit

2. Data Frame (in case of Parity is enabled & Parity Type is odd)

– One start bit (1'b0)

– Data (LSB first or MSB, 8 bits)
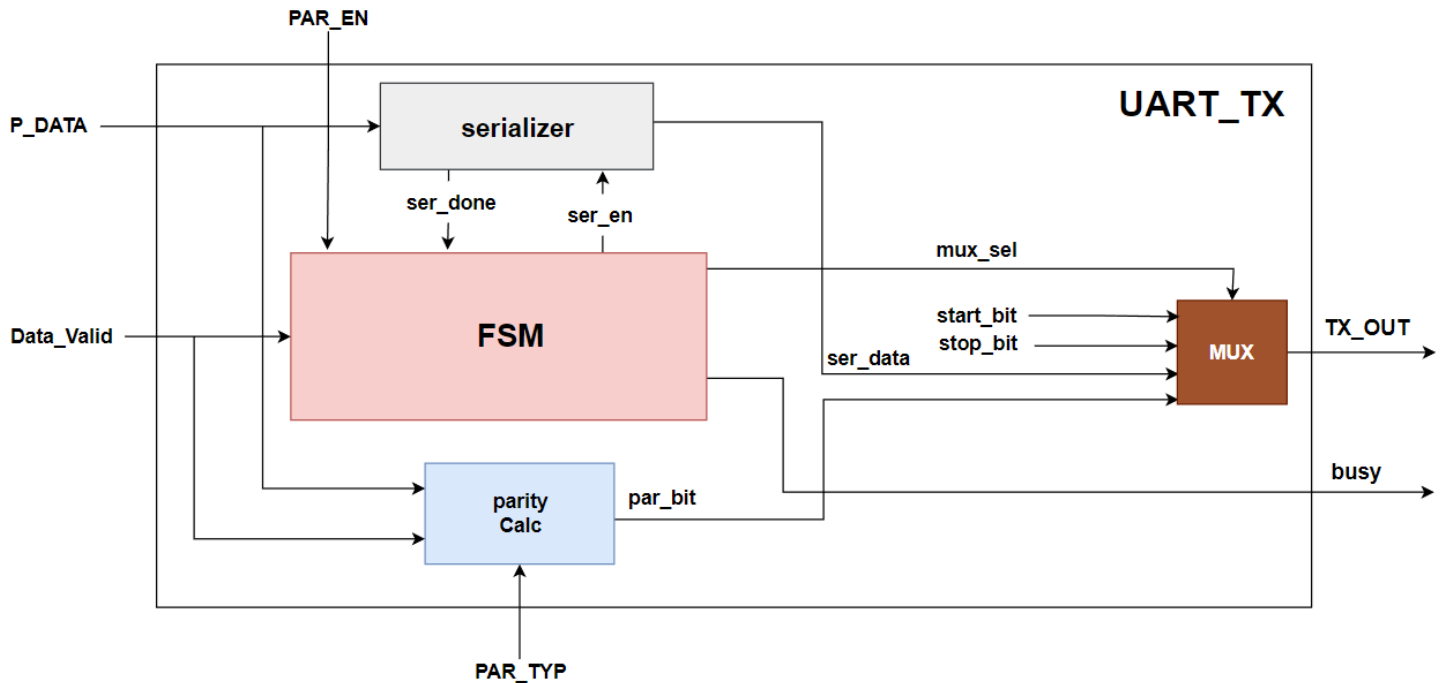
– Odd Parity bit

– One stop bit

3. Data Frame (in case of Parity is not Enabled)

– One start bit (1'b0)

– Data (LSB first or MSB, 8 bits)

– One stop bit

Block Diagram :-

Notes :-

1. Look at the frame from the receiver view, then you will notice that the start bit should be transmitted first (LSB) then transmitting data as it is, parity, stop bit.

2. This TX is implemented to be used inside a system so the signals at the input ports of the top module will come from system controller, like testbench.

3. There are two .txt files used in testbench to take inputs and check the outputs from them.

4. There is a Do file that can be used with QuestaSim or ModelSim tools to check the testbench cases faster.