

Universal Asynchronous Receiver Transmitter

Date

UART

- High bandwidth protocol :- It is a parallel communication like → APB.
→ AHB.
→ AXI.
- But APB is a less bandwidth protocol.
- Protocol :- A set of rules and regulations is called a protocol.
- Different types of protocols are available.
 1. Bus Protocol - APB, AHB, AXI, ASB, ACE.
 2. Peripheral Protocol (Serial comm. Protocol)
 - High speed : PCIe, SATA, SAS, Ethernet, USB, MIPI.
 - Low speed : UART, SPI, I₂C.
- APB → In system verilog
- AHB & AXI → In UVM.
- Exchange of information from one system to another system with a medium is called communication.
- ~~The~~ Universal Asynchronous receiver - transmitter (UART) is one of the simplest and oldest forms of device - to device digital communication.

UART

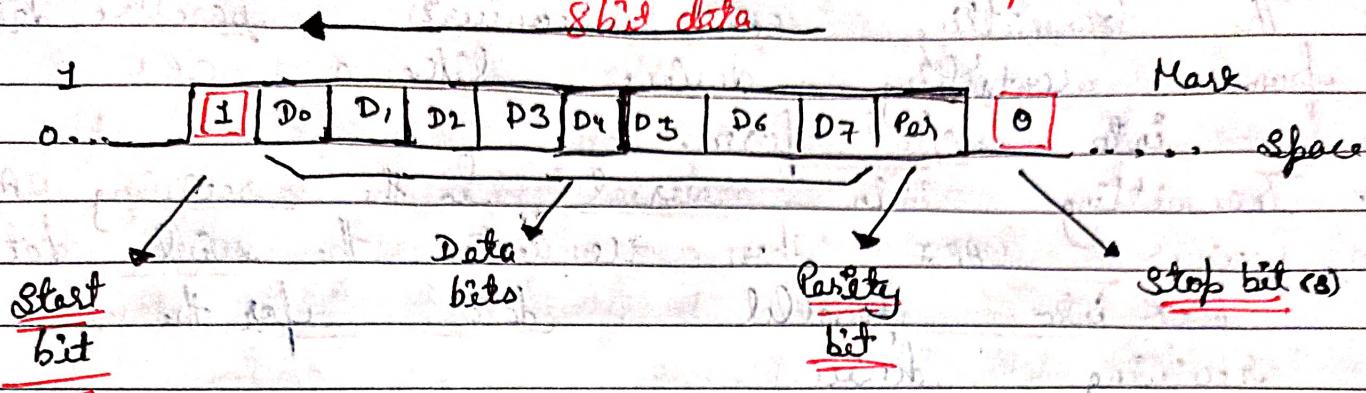
- UART stands for a universal asynchronous transmitter and receiver.
- (asynchronous means → no clock)
- UART protocols is a serial communication with two wired protocols.
- The data enable signal lines are labelled as Rx & Tx.
- Serial comm. is commonly used for transmitting and receiving the signal.
→ (Serial comm. means one by one.)
→ (bit by bit)
- It is transferred and receives the data serially bit by bit without clock pulses.
- The UART takes byte of data and sends the individual bits in a sequential manner.
- UART is a half-duplex protocol.
→ (half duplex means transfer & receive the data but not at the same time.)
- Half-duplex means transferring and receiving the data but not at the same time.
- It uses a single date line for transmitting and receiving the date.
- It has one start bit, 8-bit date and one stop bit means the 8-bit date transfer and its signal is high to low.

Frame Format

- UART Protocol Data Flow

Date

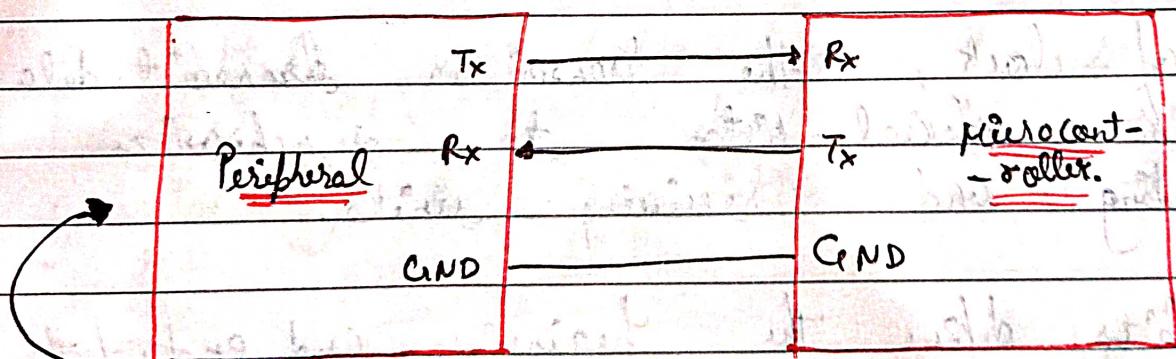
11 bit format



Master \rightarrow Slave
 (Transmitter) \rightarrow (Receiver)
 Sending data \rightarrow regard the data.

- Parity bit :- It check our receiving data is correct or not.

UART COMMUNICATION



(Peripheral contains port)
 (SOC)

through these
 ports

connect to

Microcontroller.

- Two UARTs communicate directly with each other.
- The transmitting UART converts parallel data from a controlling device like a CPU into serial form.
- Transmitting UART then converts the serial data to the receiving UART.
- Receiving UART then converts the serial data back into parallel data for the receiving device.
- Only two wires are needed to transfer data between two UARTs.
- Data flows from Tx pin of the transmitting UART to the Rx pin of the receiving pin.

o UART (asynchronous Tx & Rx).

no need to transmit clock signal with the transmitted data.

- Instead of clock, the transmitter transmits data with some special bits to synchronize the sending and receiving units.
- These bits define the beginning and end of the data packet so the receiving UART knows when to start and stop reading the bits.
- These special bits are:-
 - Start bit
 - Priority bit
 - Stop bit

Date

- Working

1. The transmitter receives the 8-bit - data from input and stores it in same register.
2. The transmitter then adds some special bits like start bit, stop bit and parity bit (if selected) to make the complete data frame.
3. The data frame then sent out serially by the transmitter at the predefined clock rate (baud rate).
4. The receiver is by default at high logic state which indicates idle stat. of receiver and keeps looking for the high to low transition i.e. start bit.
5. As soon as the start bit detected, the receiver observe the start bit for 50% of the receiving baud rate, if it is the receiver start sampling and other data bits at the middle of each bit otherwise receiver set flag for framing error.
6. 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.

Verilog Implementation.

- This project is divided into 3 main modules for the easy and clear understanding and also for the ease in further development of project.
 - Baud rate generator
 - UART-transmitter
 - UART-receiver
 - These 3 module are further divided into sub modules.
 - All the modules are connected by instantiating each module in the main module.
- Black diagram
1. baud-rate-generator
 2. uart-receiver-without-ff
 3. uart-transmitter-without-ff.
- Baud rate generator
 1. Baud rate generator determines transmission speed in asynchronous communication.
 2. It is the number of symbols per second transferred.
 3. Each bit is $1/(\text{baud rate})$ wide.
 4. $\text{Baud rate} = \text{clock freq.} / (16 \times \text{divider})$.

some standard baud rate:-

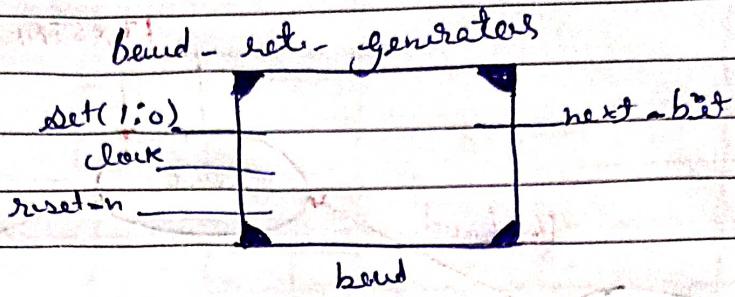
- 2400

- 9600

- 19200

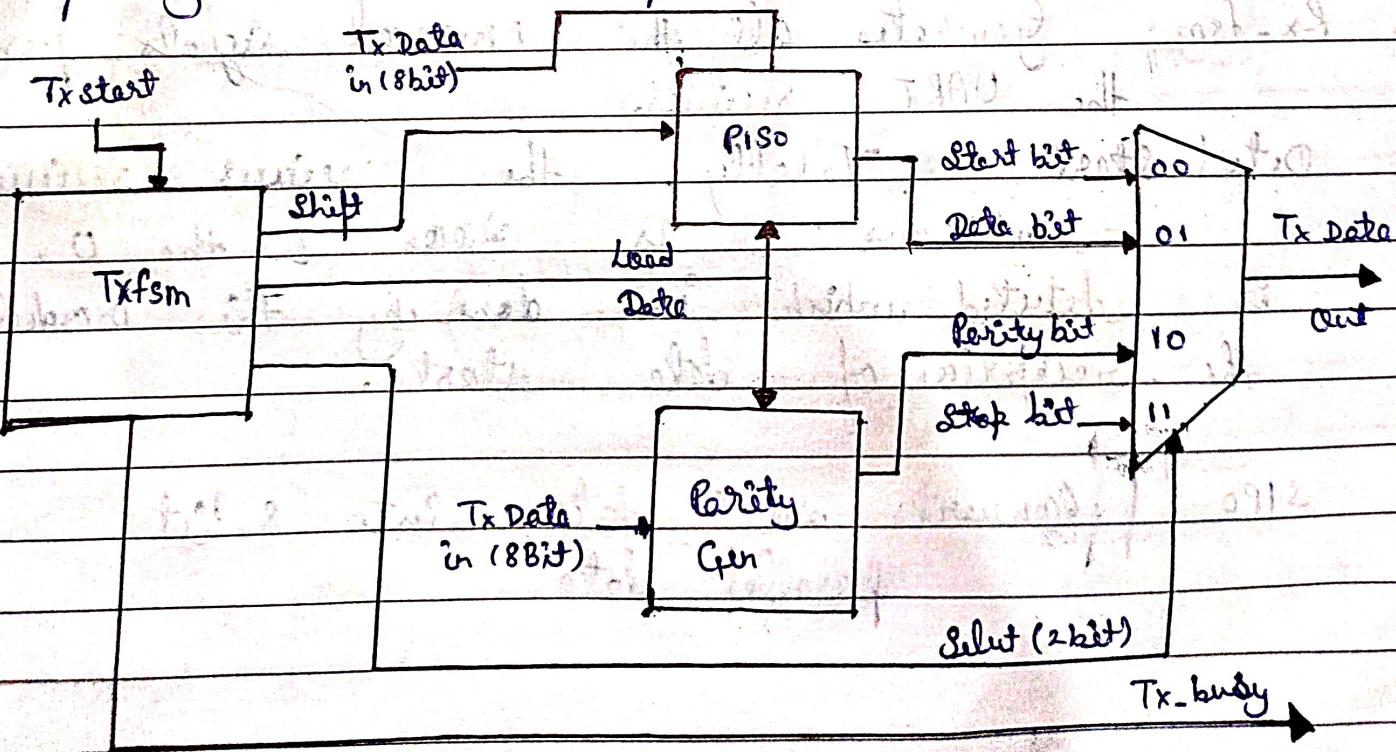
- 38400

- 115200



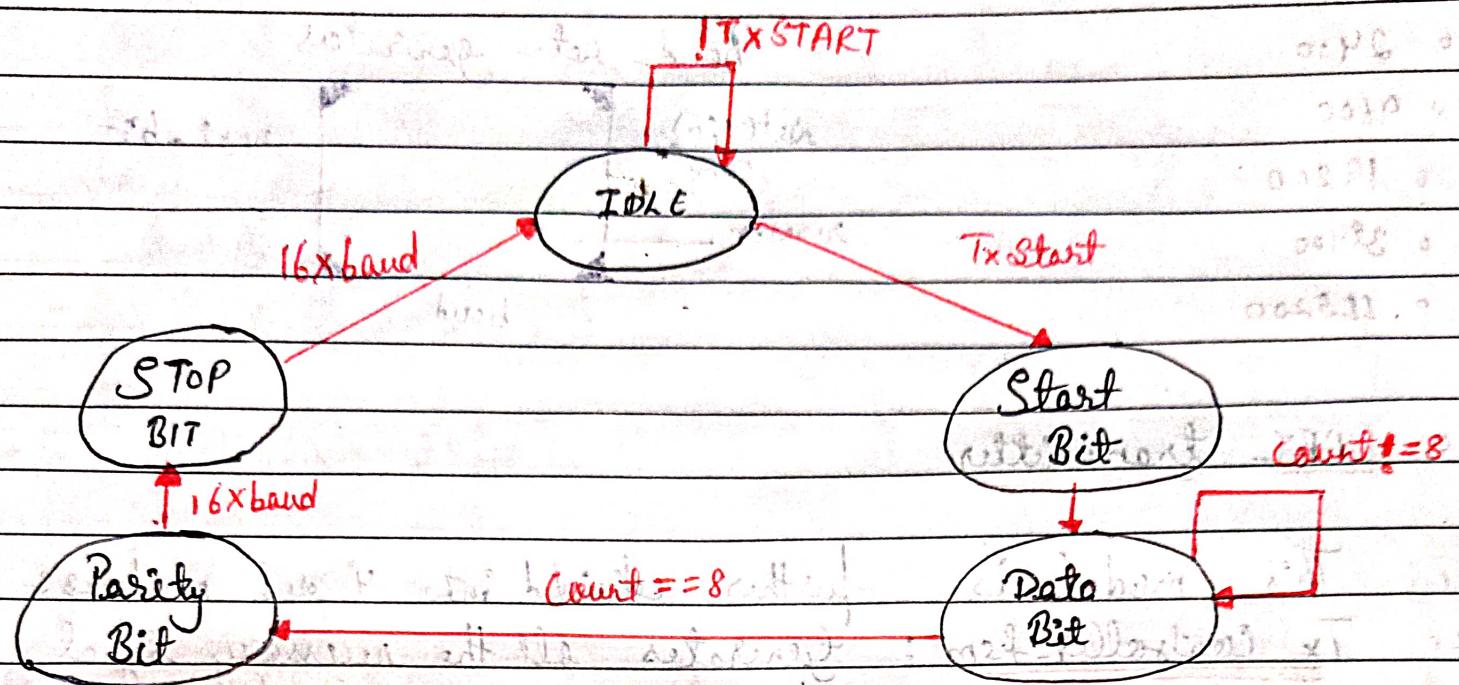
◦ UART-transmitter

1. This module is further divided into 4 sub modules:-
2. Tx Controller-fsm : Generates all the necessary signal required to transmit data at right time.
3. Parity-generator : Generate parity for the 8 bit input data.
4. PISO (Parallel In Serial Out) :- Takes the 8 bit input binary data and convert it into 1 bit serial data.
5. Txmux : It is 4×1 mux to transmit 4 different type of data viz., start bit, data bit, parity bit and the stop bit.



Transmitter State Machine

Date



- UART Receiver
- This module is further divided into the following sub modules:-
- 1. Rx-fsm : Generates all the control signals for the UART receiver.
- 2. Detect start : Ideally, the receiver receives continuous 1's as soon as the 0, is detected which is done by this module, the receiver of data starts.
- 3. SISO : Converts serial data into 8 bit parallel data.

Date

