

Basics of

UART

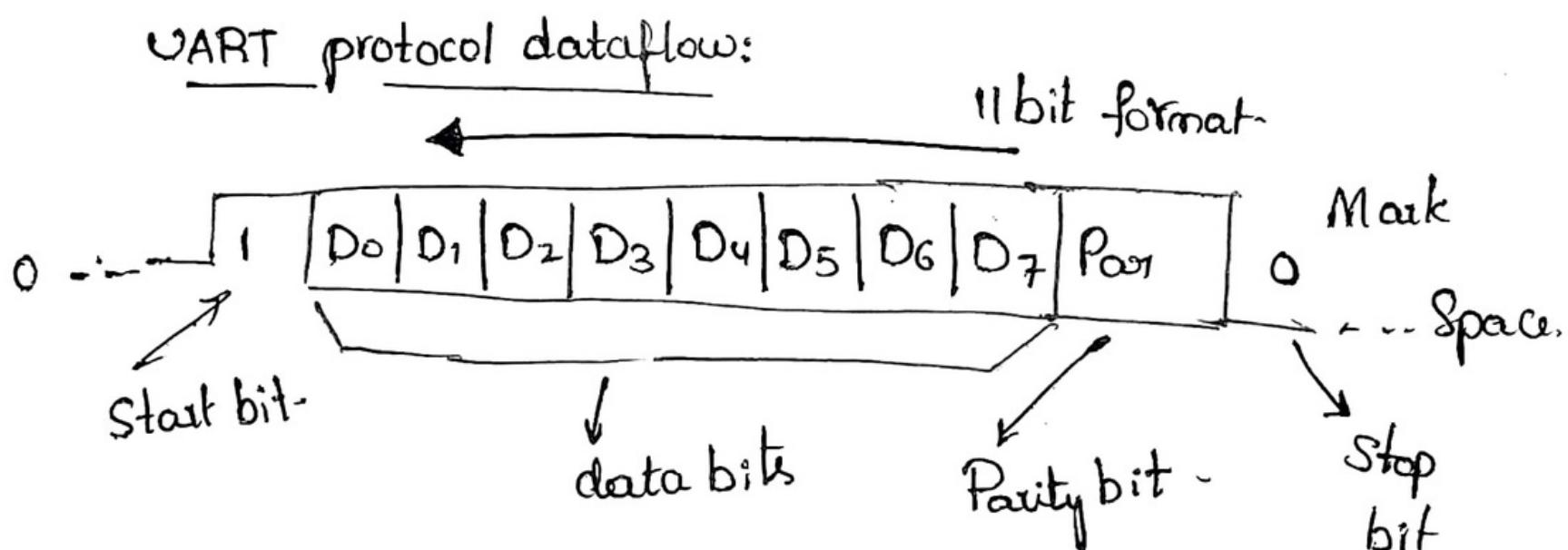
Communication

Ummidi Chandrika

Universal Asynchronous

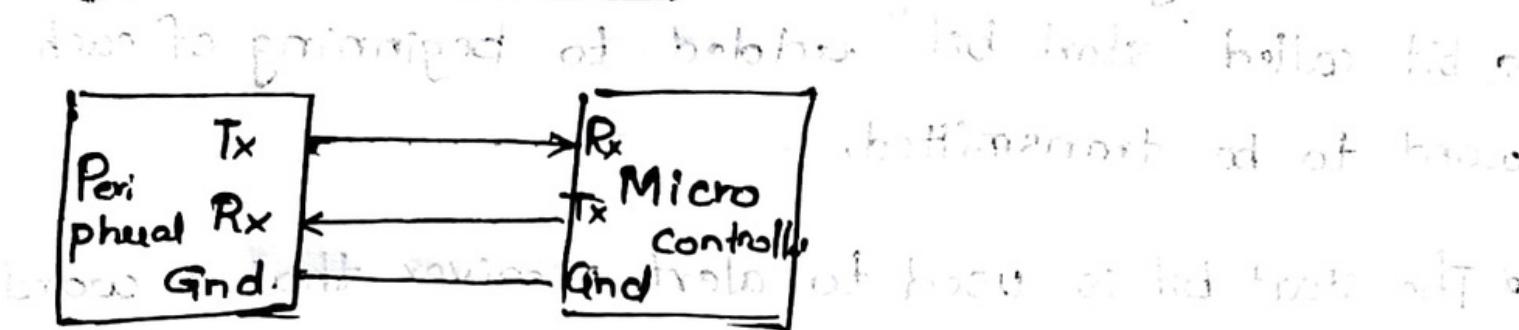
Receiver Transmitter (UART)

- Protocol → set of rules to transmit data.
- UART stands for Universal Asynchronous Receiver Transmitter.
No Synchronization, No clock, Communication possible.
- is a serial Communication with two wired protocol.
- The data cable lines are labelled as Rx and Tx.
- Serial communication commonly use for Transmitting & Receiving s/l.
- It transfers & receives data serially bit by bit without class pulses.
- The UART takes bytes of data & sends individual bits in a sequential manner.
- UART is a half duplex protocol.
- tx and Rx data not at same time
- use single data line for transmitting & ex data.
it has one start bit, ^{one parity bit} 8-bit data & one stop bit means
8-bit data transfer one's signal is high to low.



Intra protocol → UART.

UART Communication: TRRS → serial communication.



→ Two UARTs of host side of both of which Tx

→ Two UARTs communicate directly with each other.

→ The transmitting UART converts parallel data from controlling device like CPU into serial form.

→ Transmitting in serial to the receiving UART.

→ Receiving UART then converts serial data back into parallel data for the receiving device.

→ Only two wires are needed to transmit data between two UARTs.

→ Data flows from Tx pin of transmitting UART to the Rx pin of receiving UART.

→ UART transmit data asynchronously, which means there is no need to transmit clock signal with transmitted data.

→ instead of clock, the transmitter transmit data with some special bits to synchronize sending & receiving units.

→ These bits define beginning & end of the data packet so receiving UART knows when to start and stop reading bits.

→ These special bits are:

Start bit, Priority bit, Stop Bit.

Start Bit:

- When word given to UART for asynchronous transmission, a bit called "start bit" added to beginning of each word to be transmitted.
- The start bit is used to alert receiver that a word of data is about to sent, and to force the clock in receiver into synchronization with clock in Transmitter.
- The Tx receives 8-bit data from input and stores in some register.
- The Transmitter then adds some special bits like start bit, stop bit & parity bit (if selected) to make complete data frame.
- The data frame then sent out serially by transmitter at the predefined clock rate. (Baud Rate)
- The receiver is by default at high logic state, which indicates idle state of receiver. and keeps looking for high to low transition i.e., start-bit.

Data bit:

- After start bit, individual bits of data are sent, with LSB sent 1st.
- Each bit in Transmission is transmitted for exactly the same amount of time as all of other bits.
- And the receiver looks at wire at approximately halfway through period assigned to each bit to determine if bit is 1 or 0.

Parity Bit:

- To remove problem of loss of some bits during transmission of signals, error correction mechanism must be added to transmitted data.
- Parity bit error checking mechanism is one of the simplest methods to detect.
- Any error in received data.

Stop Bit:

- At end of each data packet, stop bit i.e. 1 is added to indicate end of one data packet.
- At rx end, this stop bit is used to stop the reception of data.

Verilog Implementation:

- This project is divided into 3 main modules for easy and clear understanding & also for ease in further development of project.
 - Band_rate_generator
 - UART_transmitter
 - UART_Receiver
- These 3 modules are further divided into Sub-modules.
- all the modules are connected by instantiating each module in main module.

Baud Rate Generator: (for synchronization of data transfer)

- Baud Rate Generator determines the speed in Asynchronous Communication.

- It is no. of symbols per second transferred.

- Each bit is $1 / (\text{baud rate})$ wide.

- Baud Rate = $\frac{\text{clock freq.}}{(16 \times \text{Divisor})}$ [Higher the baud rate, data transferred in less time]

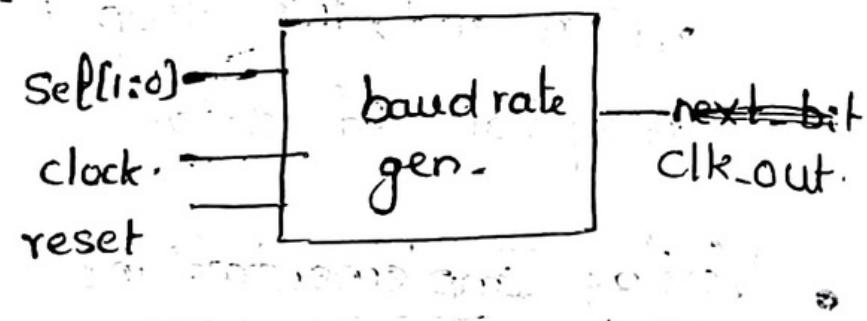
- Some of the standard Baud Rate are:

→ 2400

→ 9600 (mostly used)

→ 19200

→ 38400



UART Transmitter:

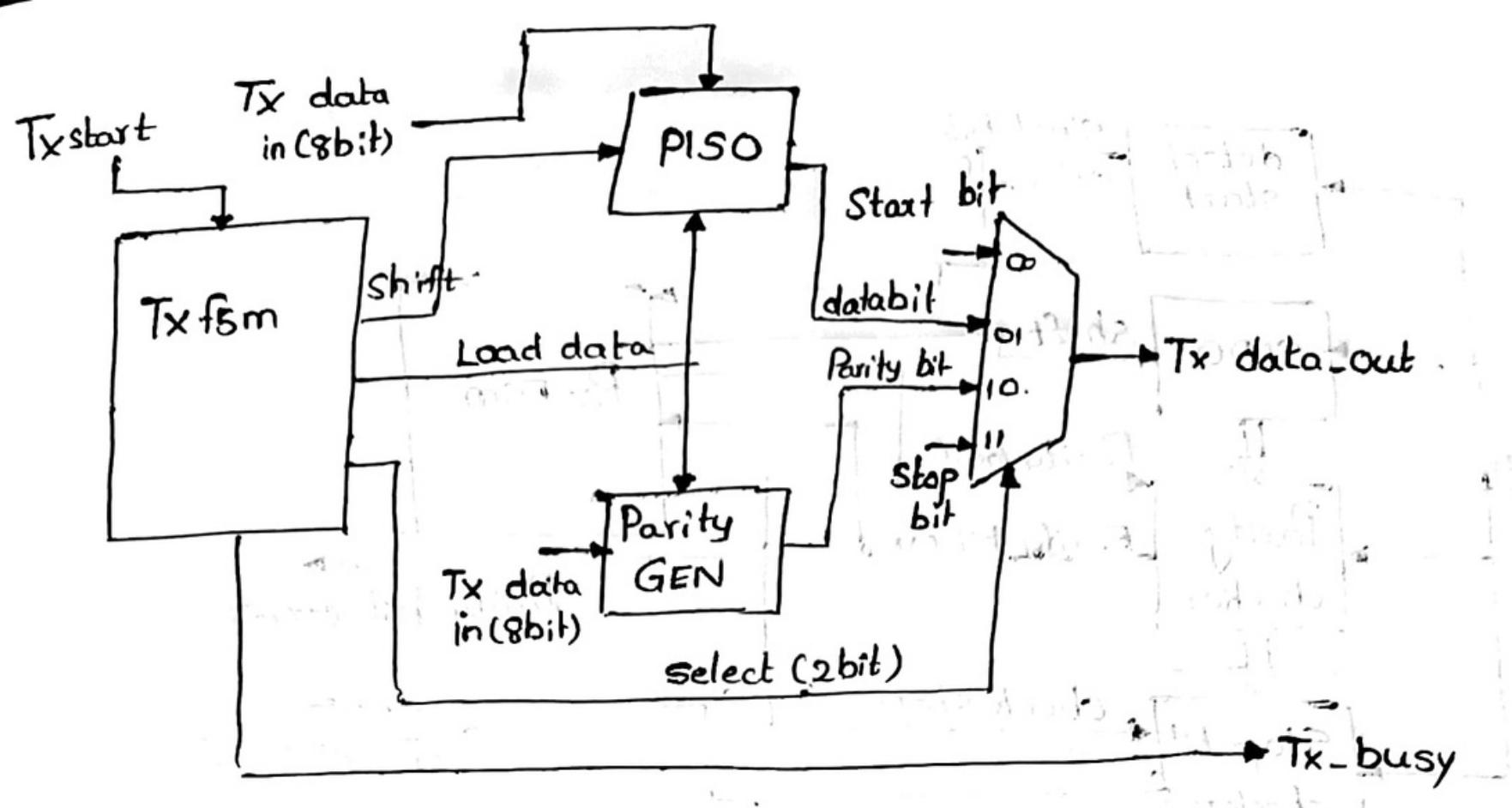
This module is further divided into 4 Sub-modules:

Tx Controller_fsm: Generates all the necessary signals required to transmit data at right time. Brain of Tx.

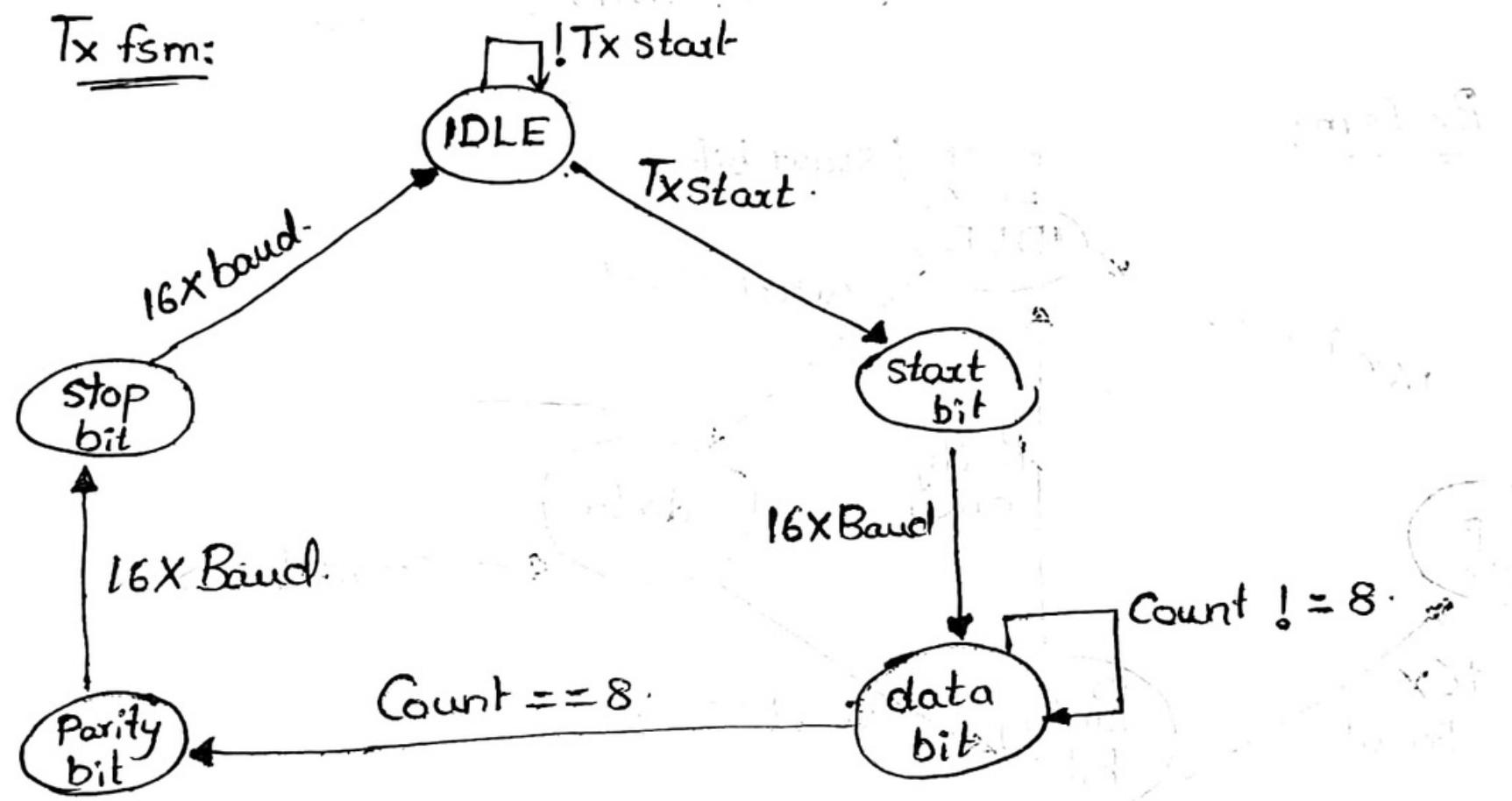
Parity Generator: Generate parity for 8 bit input data.

PISO (Parallel In Serial Out): Takes 8-bit input binary data & convert it into 1bit serial data.

Tx Mux: it is 4×1 mux to transmit 4 different type of data viz. start bit, data bit, parity bit & stop bit.

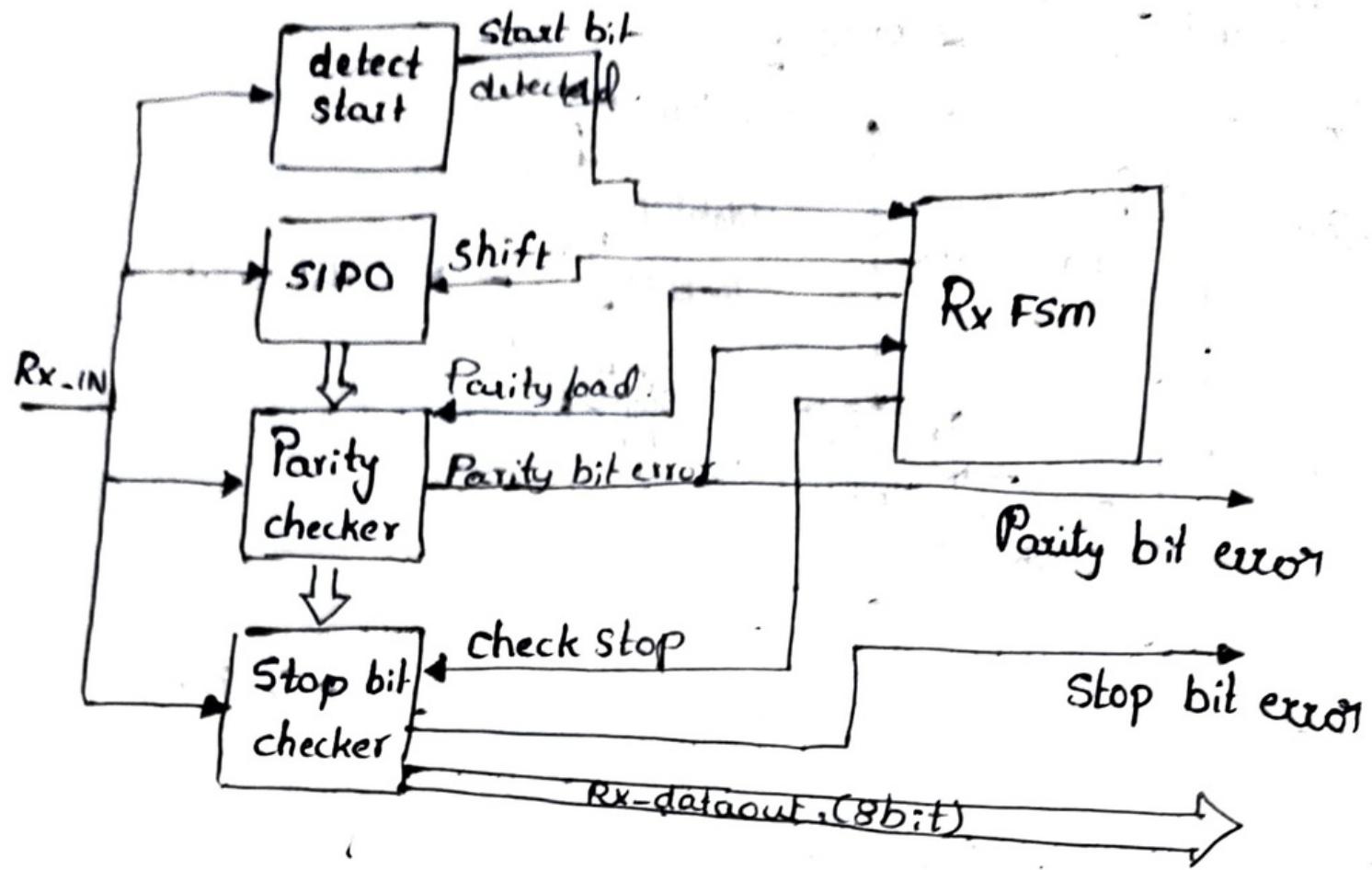


Tx fsm:

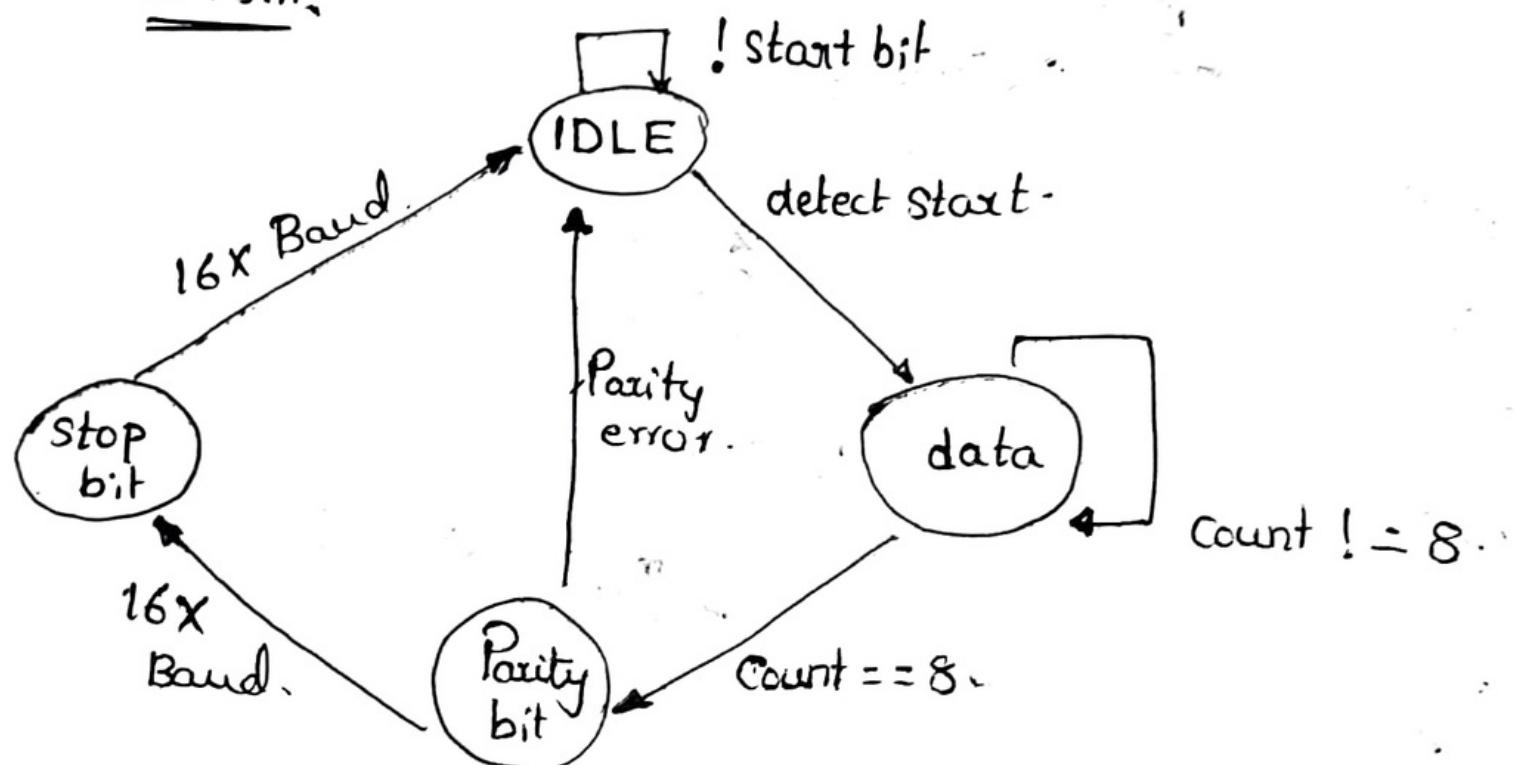


UART Receiver:

- This module is further divided into following submodules:
 - **Rx_fsm**: Generates all control signals for UART receiver.
 - **Detect start**: Ideally receiver receives continuous 1, as soon as 0 is detected which is done by this module, reception of data starts.
 - **SIPo**: Converts serial data into 8-bit parallel data.



Rx fsm:



Advantages and Disadvantages of UART:

No communication protocol is perfect, but UART's are pretty good at what they do.

Advantages:

- Only uses two wires.
- No clock signal is necessary.
- Has a parity bit to allow for error checking.
- The structure of data packet can be changed as long as both sides are set up for it.
- Well documented and widely used method.

Disadvantages:

- The size of data frame is limited to a maximum of 9 bits.
- Doesn't support multiple slave or multiple master systems.
- The baud rates of each UART must be within 10% of each other.