UART(Universal Asynchronous Receiver Transmitter)

Video Lecture Link Part#01

Video Lecture Link Part#02

UART(Universal Asynchronous Receiver Transmitter)

UART Key Points:

- A Universal Asynchronous Receiver and Transmitter (UART) is a Digital Circuit that sends Parallel Data through a Serial Line.
- Main purpose of UART is to Transmit and Receive Serial Data.
- Two UARTs Communicate Directly in UART Communication System.
- UART Transmits data Asynchronously, which means there is no Clock signal to synchronize the output bits from the transmitting UART to the Sampling of bits by the receiving UART

UART(Universal Asynchronous Receiver Transmitter)

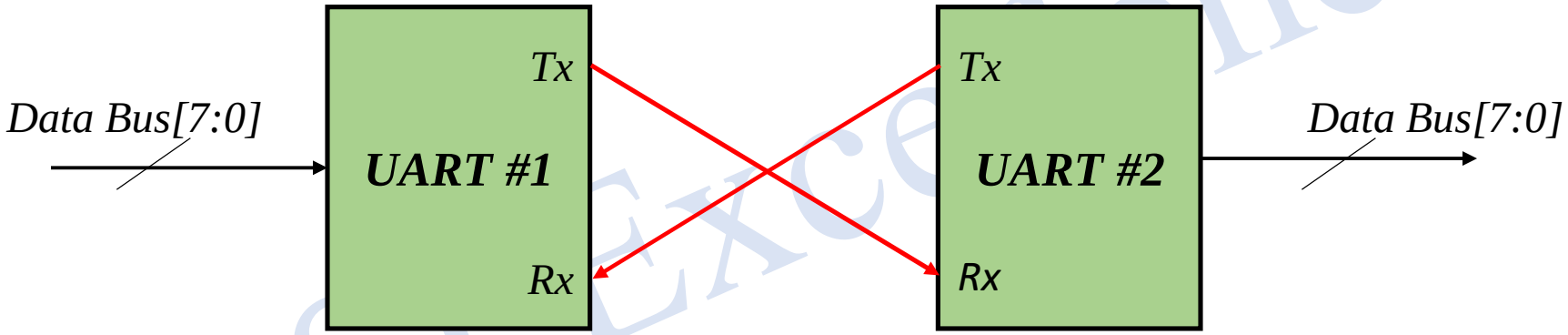

Figure #01: UART System

UART(Universal Asynchronous Receiver Transmitter)

UART Operating Principle:

- Transmitting UART converts the received parallel data (eg. from CPU, Memory) into serial form.
- The UART Transmitter adds START and STOP bits to the Data Packets being transferred.
- The Data Packet is then Serially Transferred to Receiving UART
- These bits defines the Start and Stop of the Data Packet, So the receiving UART knows when to sample the Data or Start reading the bits
- When the receiving UART detects a Start bit, it starts to read the incoming bits at a specific frequency known as the Baud Rate.
- The receiving device then removes the START bit, Parity bit and Stop bit, and converts the serial data into parallel form (which can be read by CPU)
- Baud Rate is the measure of the speed of the data transferred, expressed in Bits Per Second (BPS).
- Both Tx and Rx UART should operate at about the same baud rate
- Only One Master and One Slave device can participate in UART Serial Communication.

UART(Universal Asynchronous Receiver Transmitter)

Figure #2: UART Data Frame

Each UART Packet Contains 1 **START** bit, 5 to 8 **DATA** bits (depending on the UART), an optional **PARITY** bit and 1 or 2 **STOP** bits.

UART(Universal Asynchronous Receiver Transmitter)

In UART, the Data Frame is considered as a combination of four basic components namely

*1) **START Bit**: High to low-level voltage change represents the start of data transmission*
*2) **INPUT Data**: Data stored in the form of bits*
*3) **PARITY Bit**: It is used to evaluate the fault in received data*
*4) **STOP Bit**: Terminates data transfer*

UART(Universal Asynchronous Receiver Transmitter)

Advantages of UART Communication:

1) Fewer Signals (2)
2) No Clock Signals
3) Error Detection Using Parity Bit

UART(Universal Asynchronous Receiver Transmitter)

Dis-advantages of UART Communication:

1) Serial Communication (Less Throughput)
2) Data Limit (Limited Data Frame)
3) Only One Master and Slave Device is Supported
4) Both End Should have Similar Data Transfer Setting

UART(Universal Asynchronous Receiver Transmitter)

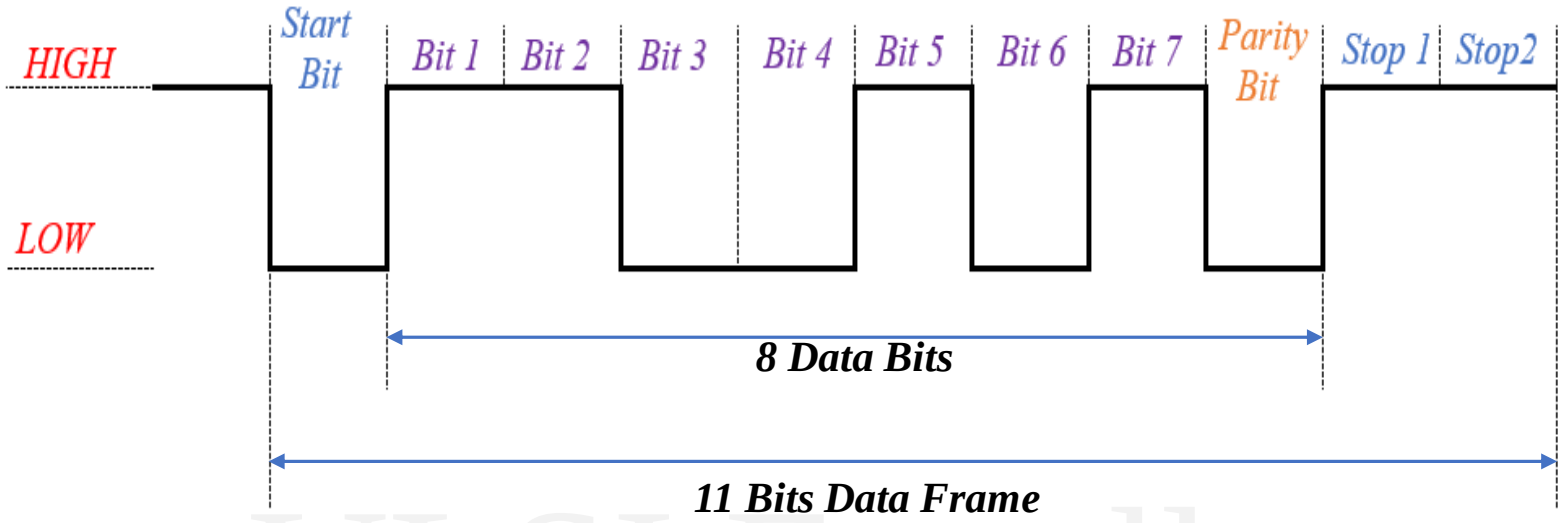

Figure #3: UART Data Frame Format

UART(Universal Asynchronous Receiver Transmitter)

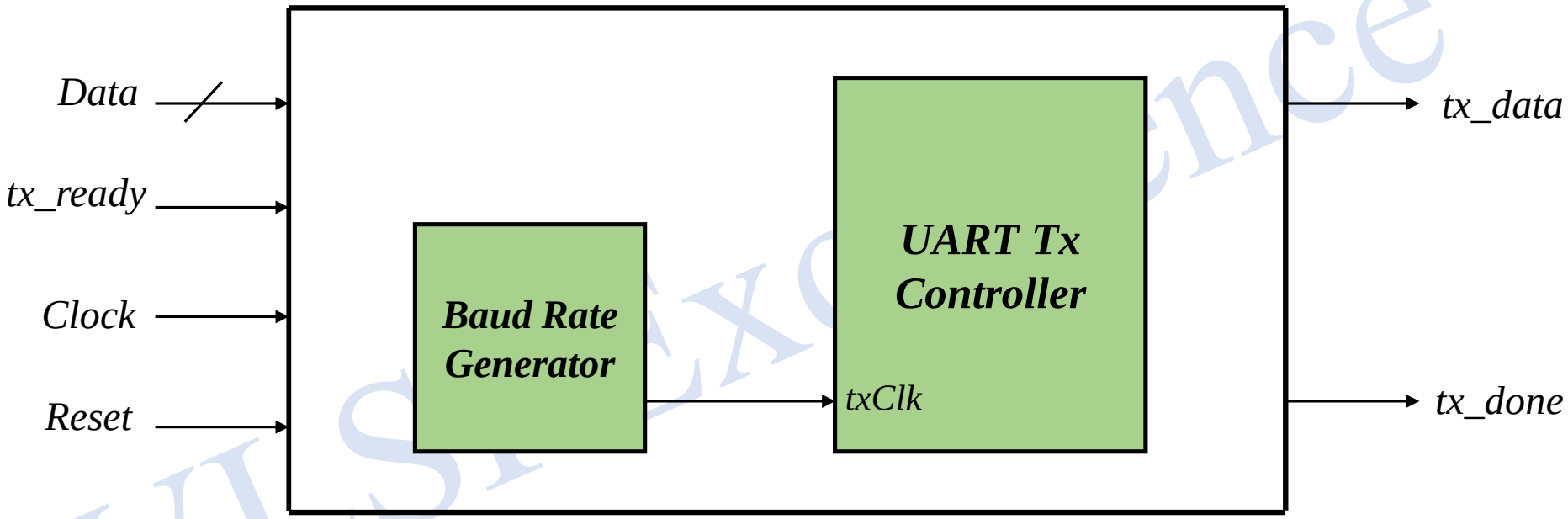

Figure #04: UART Tx Block Diagram

UART(Universal Asynchronous Receiver Transmitter)

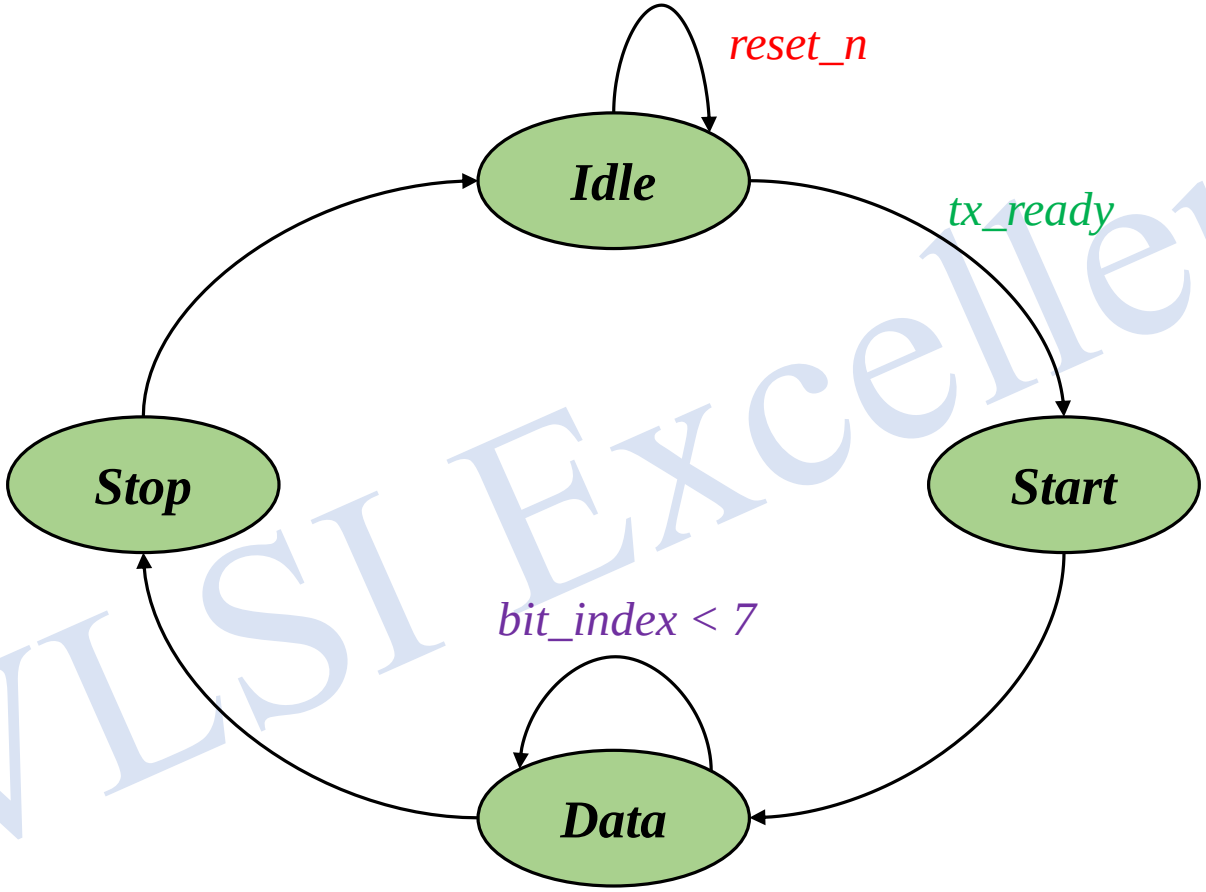

Figure #05: UART Tx FSM

UART(Universal Asynchronous Receiver Transmitter)

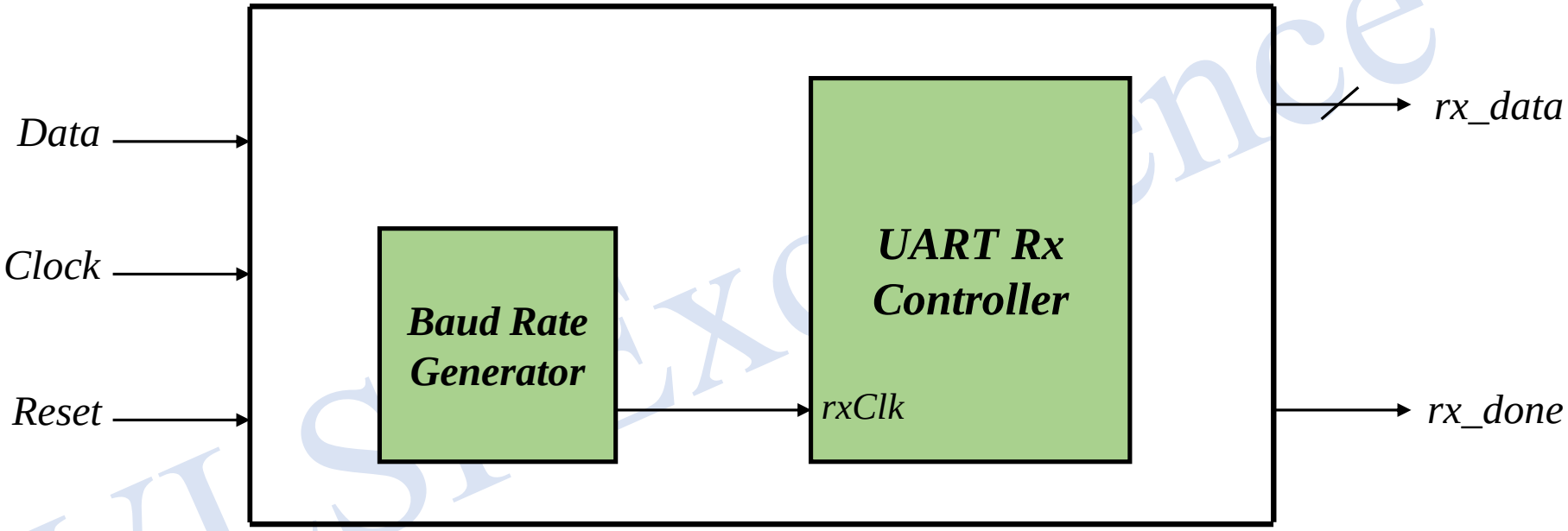

Figure #06: UART Rx Block Diagram

UART(Universal Asynchronous Receiver Transmitter)

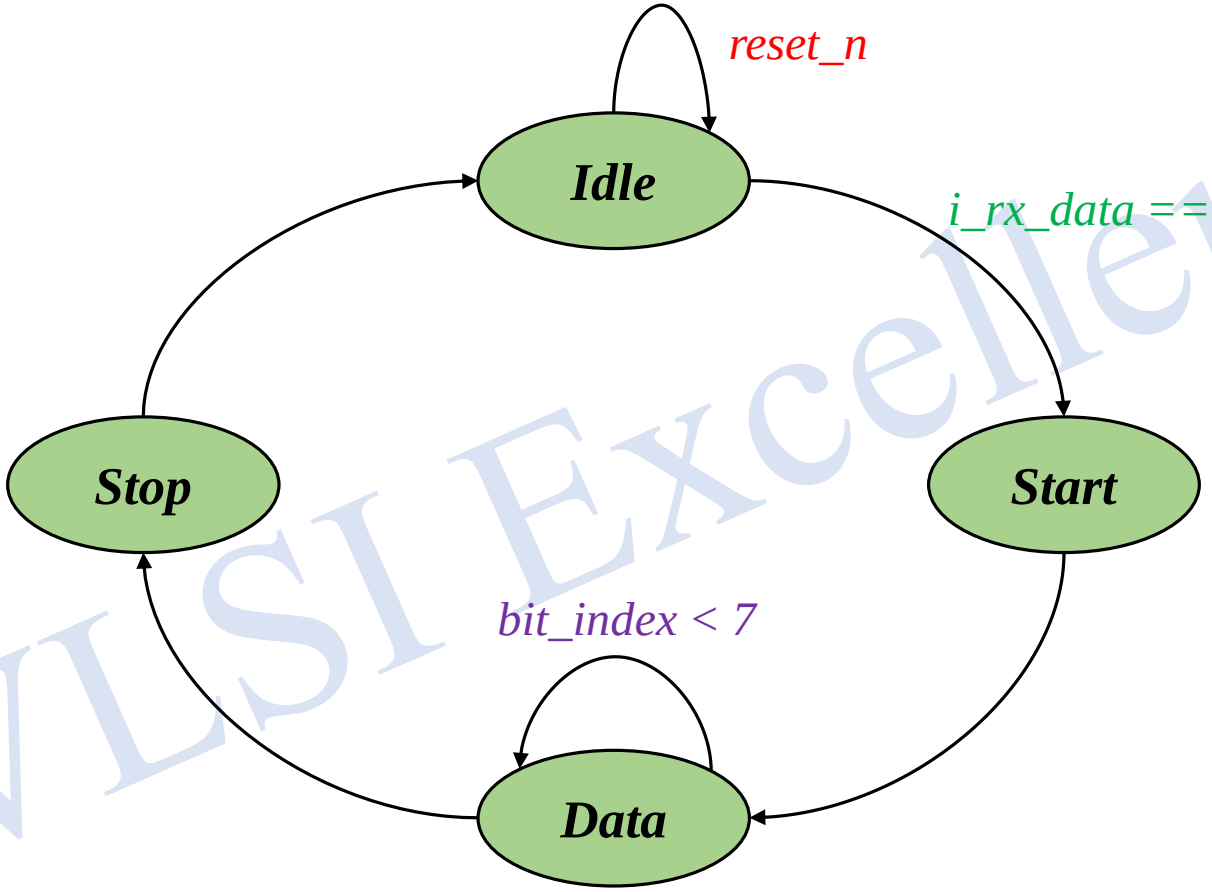

Figure #07: UART Rx FSM

UART(Universal Asynchronous Receiver Transmitter)

Oversampling:

The receiver of the UART peripheral implements different user-configurable oversampling techniques for data recovery by discriminating between the valid incoming data and noise.

the oversampling technique talks about the sampling frequency of the UART peripheral's receive engine to recover the data coming on the Rx line of the UART.

UART(Universal Asynchronous Receiver Transmitter)

The oversampling scheme works as follows: (16x Oversampling)

1) Wait until the incoming signal becomes 0, the beginning of the start bit, and then start the sampling tick counter.
2) When the counter reaches 7, the incoming signal reaches the middle point of the start bit. Clear the counter to 0 and restart.
3) When the counter reaches 15, the incoming signal progresses for one bit and reaches the middle of the first data bit. Retrieve its value, shift it into a register, and restart the counter.
4) Repeat step 3, to retrieve the remaining data bits.
5) If the optional parity bit is used, repeat step 3 one time to obtain the parity bit.
6) Repeat step 3, to obtain the stop bits.

Best Free VLSI Content

1. Verilog HDL Crash Course – Link
2. Static Timing Analysis (STA) – Theory Concepts – Link
3. Static Timing Analysis (STA) – Practice/Interview Questions – Link
4. Low Power VLSI Design – Theory Concepts – Link
5. Low Power VLSI Design (LPVLSI) – Practice/Interview Questions – Link
6. Digital ASIC Design Verilog Projects – Link

Please Like, Comment, Share & Subscribe My Channel in Order to Reach Out the Content to a Larger Audience.

Thanks !!