

Universal Asynchronous Receiver Transmitter UART

Raj Kumar Laldev
Advance VLSI Lab
Silicon institute of technology Bhubaneswar, India
rajbihar5364@gmail.com

Abstract—All most all computers and microcontrollers have several serial data ports used to communicate with serial input/output devices such as keyboards and serial printers. By using a modem connected to a serial port serial data can be transmitted to and received from a remote location via telephone line. The serial communication interface, which receives and transmits the serial data is called a UART (Universal Asynchronous Receiver-Transmitter). Rx is the received serial data signal and Tx is transmitted data signal. In this project UART is implemented in Virtex II pro FPGA chip due to low cost, high speed, reprogram ability and fast time to market

Index Terms—

1 INTRODUCTION

Protocol: A set of rules and regulations is called a protocol. Different type of protocols are available.

1. Bus protocol: APB,AHB,AXI,ASB,ACE
2. Peripheral protocol.

- High speed: PCIe, SATA, SAS, Ethernet, USB, MIPI.
- Low speed :UART,SPI,I2C .

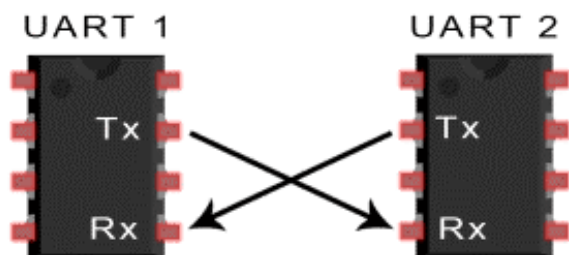


Fig. 1. Single wire connection

- One of the biggest challenge in SOC design is the on chip communication between the different components.
- The different bus protocols used for interconnection .
- Most of the times, the IP cores are designed with many different interfaces and communication protocols and this can be a problem while integrating into an SOC.
- To avoid this problem, standard on-chip bus structures and protocols were developed. Exchange of information from one system to another system with a medium is called communication.

- A set of rules and regulations that allow two electronic devices to connect to exchange the data with one and another.
- Universal asynchronous receiver-transmitter (UART) is one of the simplest and oldest forms of device-to-device digital communication. You can find UART devices as a part of integrated circuits (ICs) or as individual components.
- UART stands for a universal asynchronous transmitter and receiver.
- 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.

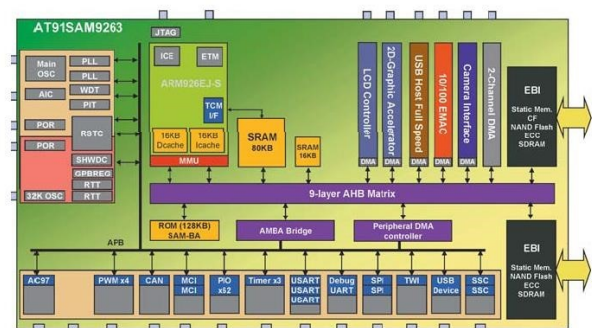
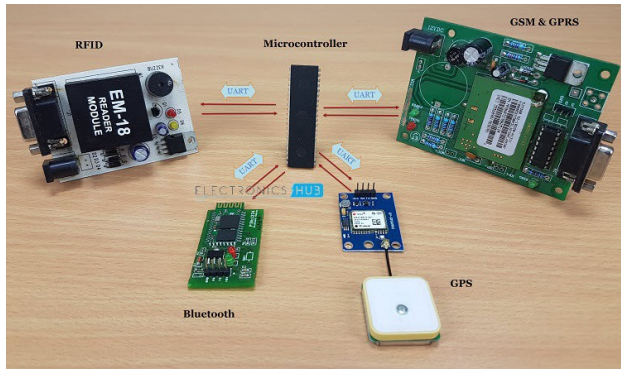


Fig. 2. System on Chip

- It is transferred and receives the data serially bit by bit without clock pulses.
- The UART takes bytes of data and sends the individual bits in a sequential manner.

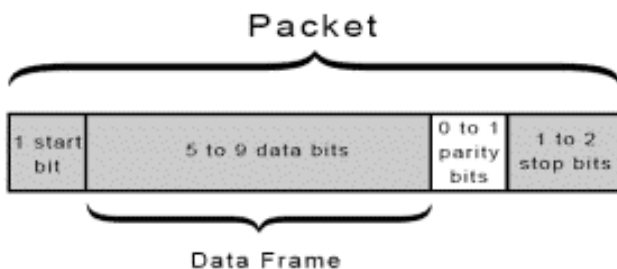
- UART is a half-duplex protocol.
- Half-duplex means transferring and receiving the data but not at the same time.
- It uses a single data line for transmitting and receiving the data. It has one start bit, 8-bit data and one-stop bit mean the 8-bit data transfer one's signal is high to low.

2 BACKGROUND



UART is one of the most simple and most commonly used Serial Communication techniques. Today, UART is being used in many applications like GPS Receivers, Bluetooth Modules, GSM and GPRS Modems, Wireless Communication Systems, RFID based applications etc.

2.1 UART Protocol Data Flow



These special bits are:

Start bit, Priority bit, Stop bit.

START BIT: When a word is given to UART for asynchronous transmission, a bit called "START BIT" is added to the beginning of each word that is to be transmitted.

The start bit is used to alert the receiver that a word of data is about to sent, and to force the clock in the receiver into synchronization with the clock in the transmitter.

DATA BIT OR DATA FRAME:- After the start bit, the individual bits of data are sent, with the least significant Bit(LSB) being sent first. Each bit in the transmission is transmitted for exactly the same amount of time as all of the other bits. And the receiver looks at the wire at approximately halfway through the period assigned to each bit to determine

if the bit is 1 or 0.

For example, if it takes 2 second to send each bit, the receiver will track the signal after 1 second has passed.

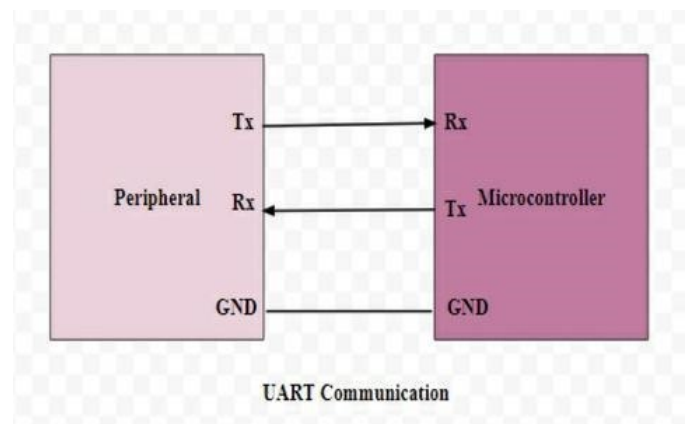
PARITYBIT: remove the problem of loss of some bits during the transmission of a signal, error correction mechanism must be added to the transmitted data. Parity bit error checking mechanism is one of the simplest methods to detect any error in received data. In asynchronous serial communication, a parity bit is added at the end of data bits to check the number of 1's.

STOP BIT: At the end of each data packet, stop bit i.e. 1 is added to indicate the end of one data packet. At the receiver end, this stop bit is used to stop the reception of Data.

3 HOW UART WORKS

Two UARTs communicate directly with each other.

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



- Transmitting it in serial to the receiving UART
- Receiver UART then converts the 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 the transmitting UART to the Rx pin of the receiving UART.
- UART transmit data asynchronously, which means there is no need to transmit clock signal with the transmitted data.
- Instead of clock, the transmitter transmit 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.
- As soon as the start bit is detected, the receiver observe the start bit for 50[percent]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.

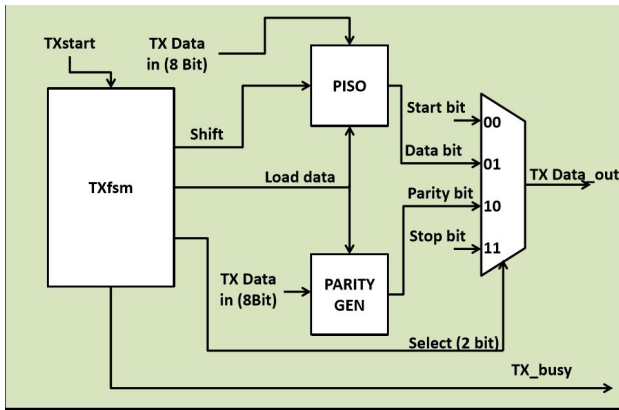


Fig. 6. Block Diagram of UART_Transmitter

- 4) **Tx_mux:** It is 4x1 Mux to transmit 4 different type of data viz.start bit, data bit, parity bit and the stop bit.

4.1 TRANSMITTER STATE_MACHINE

finite-state machine (FSM) or finitestate automaton (FSA) It is an abstract machine that can be in exactly one of a finite number of states at any given time. The FSM can change from one state to another in response to some inputs; the change from one state to another is called a transition. An FSM is defined by a list of its states, its initial state, and the inputs that trigger each transition.

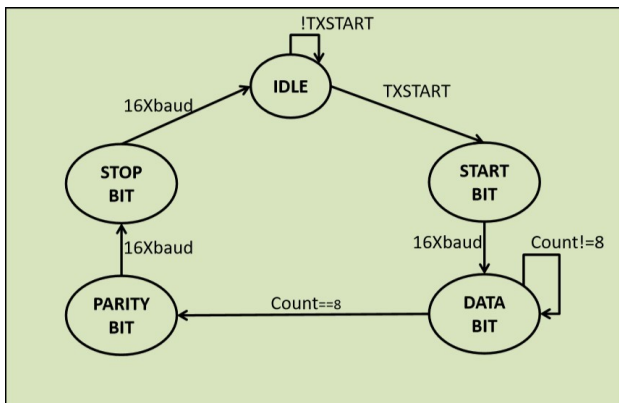


Fig. 7. Block Diagram of T_fsm

From the above figure we have to take two STATES for data shifting, given as follows

1.present_state and 2.next_state.

4.1.1 Verilog implementation from above figure.

<https://www.edaplayground.com/x/4jQu>

5 UART_RECEIVER

The receiver, on the other hand, shifts in data bit by bit and then reassembles the data. Likewise, the Receiver must detect transmission, receive the data in serial format, strip of the start and stop bits, and store the data word in a parallel format.

This module is further divided into following sub modules:-

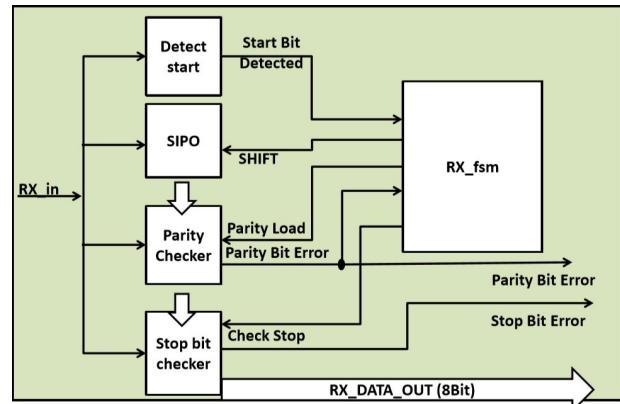


Fig. 8. Block Diagram of UART_Receiver

1. RX_fsm: Generates all the control signals for the UART receiver.

2. Detect start: Ideally the receiver receives continuous 1, as soon as the 0 is detected which is done by this module, the reception of data starts.

3. SIPO: Converts serial data into 8 bit parallel data. **4. Parity checker:** Checks the correctness of data by Xoring the 10th received bit with xor value of 8 bit received data.

5. Stop Bit checker: After the detection of valid parity bit, the stop bit is detected, if the stop bit is not detected then reception of data is terminated by setting Stop bit error signal high.

5.1 RECEIVER STATE_MACHINE

From the below fig.9, we have to take two STATES for data shifting, given as follows **1.present_state and 2.next_state.**

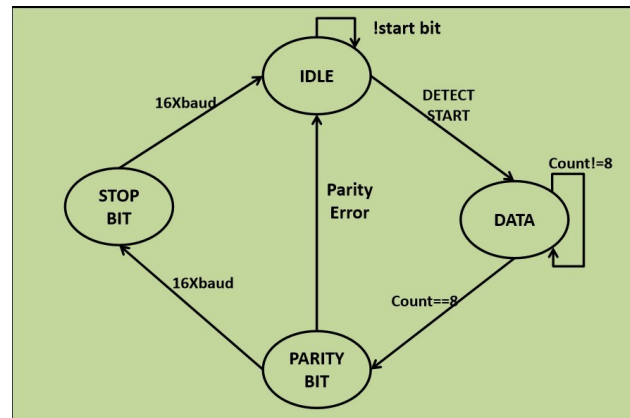


Fig. 9. Block Diagram of UART_Receiver fsm

5.1.1 Verilog implementation from above figure.

<https://www.edaplayground.com/x/6Mk4>

5.2 OVERSAMPLING PROCEDURE

- The most commonly used sampling rate is 16 times the baud rate, which means that each serial bit is sampled 16 times.
- Assume that the communication uses N data bits and M stop bits. The oversampling scheme works

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 N-1 more times 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 M more times to obtain the stop bits.

6 RESULTS

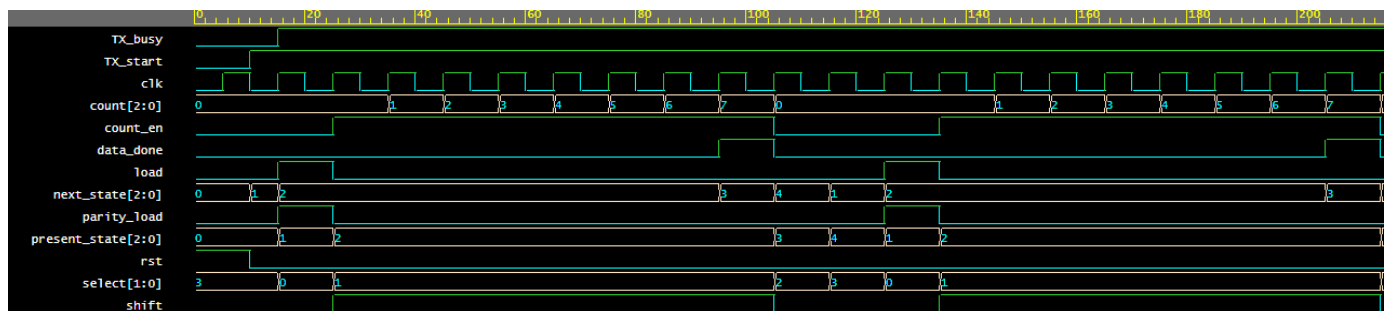


Fig. 10. *Output Of Transmitter State_machine*

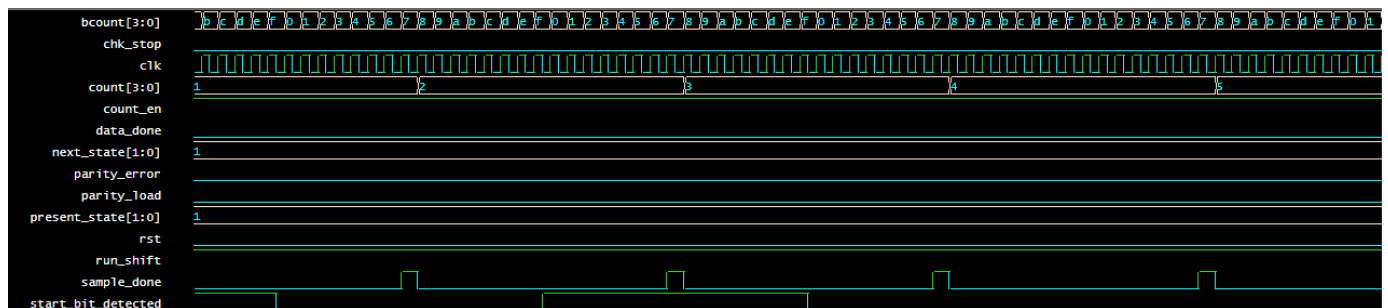


Fig. 11. *Output Of Receiver State_machine*

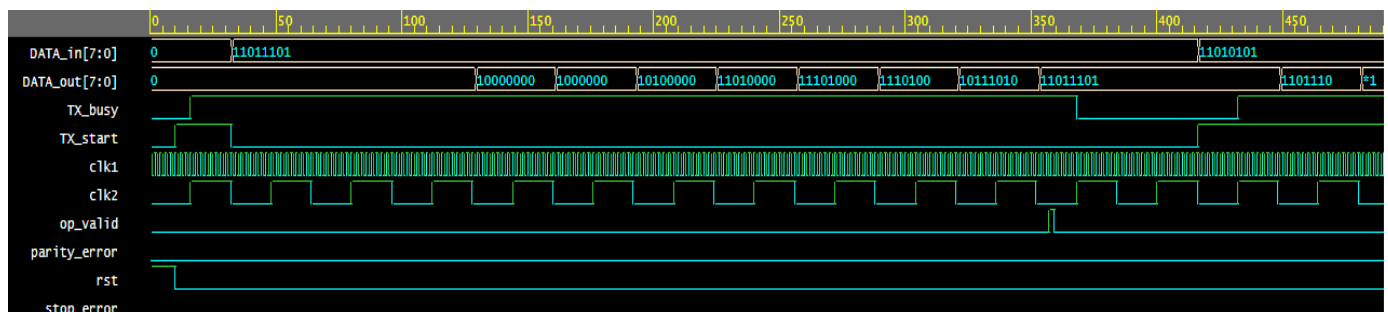


Fig. 12. *Final Output Of UART*

7 CONCLUSION

UART protocol perfectly fits for communication with low speed device,i.e.This UART is a configurable programmable logic component that accommodates communication through a simple asynchronous serial interface. It allows a user to specify the system clock, baud rate, data length, parity scheme, and oversampling rate.

8 ACKNOWLEDGMENT

I would like to thank our institution for equipping us with Advanced VLSI Lab facilities to successfully execute our project. I am also grateful to our mentor ,Dr.Saroj Rout ([linkedin.com/in/sroutk](https://www.linkedin.com/in/sroutk)) for his constant advice and guidance throughout the project duration.

REFERENCES

- [1] Semi Design [2] Electronic Hub [3]Wikipedia