Complex Engineering Problem

UART Transmitter: Datapath and Controller

Course Code and Title: EE-272L Digital Systems
Semester: Spring 2023
Instructors: Dr Ubaid Ullah Fayyaz, Dr Muhammad Muzamal Rafique, Miss Shehzeen Malik
Total Marks: 30
Deadline: Last week of semester

CLOs and PLOs for Complex Engineering Problem

CLOs		**Description**	**Domains & Levels**	**PLOs, Levels**
CLO1	Lab	Generate combinational and sequential circuits description in System Verilog	Cognitive	3. Apply
CLO2	Lab	Simulate and synthesize the design in software	Psychomotor	4. Articulation

Problem Statement

UART is one of the serial communication protocols used in microcontrollers, embedded systems, and other electronic devices. This complex engineering problem requires design, implementation, demonstration, and documentation of a UART Transmitter module to communicate with the computer.

Objectives

The objective of this complex engineering problem is to familiarize students with the design process of a serial communication protocol – UART, following datapath and controller paradigm. In the process, enhancing their understanding of state machines.

Overview of UART

UART (Universal Asynchronous Receiver-Transmitter) is a simple and reliable method of communication that uses a **start bit**, **data bits**, optional **parity bit**[1], and **stop bit** to transmit data in a sequential fashion. UART, as it names suggest, is a form of asynchronous communication, meaning that no clock signal is transmitted between devices. Instead, data is transmitted on established **baud rates** (generally taken as bits per second), which determine the speed of communication.

[1] ignore parity bit for this lab

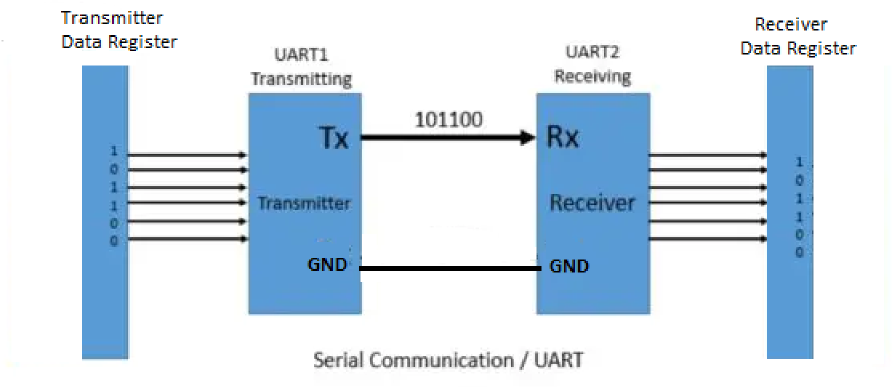

Fig. 1.1: UART Transmission between two Devices

Data on the transmitting end is provided in parallel to the data register, after which start, parity and stop bits are appended, then the entire data frame is transmitted serially via the transmitter module. Similarly, data is received bit by bit, by the receiver module, then the data bits are stored in a parallel fashion in the receiving side data register. (Fig. 1.1, Fig. 1.2)

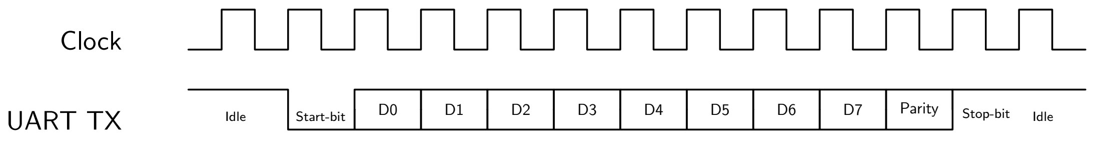

Fig. 1.2: UART Data Frame for 8-data bits and 1-stop bit

UART Data Frame

- Data Line is kept **high** when transmitter is idle. Therefore, when no data is being transmitted, it will transmit only **1s**.
- So to inform the receiver that we are going to start transmission, we send a logic **low** named as **start bit** for one clock cycle.
- Next, data bits are sent in a serial format (LSB first).
- Parity bit is an optional feature which counts the number of 1s in the data-bits. If odd parity is enabled, then if the number of 1s in the data are odd, then parity-bit is 1, otherwise zero.
- After completing transmission, the receiver needs to know that the whole data has been sent. This is done by simply transmitting **stop bit**(s). Transmitter sends one or two stops bits which are logic **high**. Hence, transmission is completed.

Baud Rate decides a transmission rate that is suitable for both transmitter and receiver. Both transmitter and receiver generate their independent clocks based on specified baud rate.

Tasks

Design and implement Datapath and Controller of a UART Transmitter module with the following specifications.

1. UART Data frame constituting one start-bit, 8-data bits and one stop-bit.

2. Baud-rate of 9600. (For simulation purpose, select a very low number for baud rate).
3. Data bits will be provided by the user which will be only available when a **Load** signal has been asserted.
4. User will indicate start of transmission by asserting **Tx_start** input.

Note:

Use of latches are not allowed in the design of synchronous circuits for the lab. Synchronous circuits must be designed with flip-flops.

Deliverables

1. A report containing the following items:
   (a) Top Level Diagram of the design.
   (b) The designed datapath circuit along-with control signals. The design should also show the labels of wires, along with the label for the width of the line for multi-bit wires.
   (c) State Machine for the controller of UART transmitter Circuit.
   (d) Implementation maximum combinational delay: Read the post implementation static timing report and identify the path with the maximum combinational delay?
   (e) Resource Utilization Report: Read the synthesis report and identify how many resources in the FPGA such as lookup tables (LUTs), input/output (IOs), etc., has been utilized.
   (f) Contribution of each member in the entire process.
2. Simulation of your designed module on QuestaSim. Show the simulation to the instructor. (No CEP will be considered without proper simulation.)
3. FPGA Implementation: Synthesize the circuit for Nexys A7 FPGA available in the lab. Tie the inputs to switches and output to Pmod Header connector. the data can be observed on the computer terminal putty using the UART to USB converter.

The collaboration between students is encouraged, but blind code sharing/copying is not allowed. If you are unable to explain anything in your code, it will be assumed you have copied it. **Copied code will be given ZERO marks.** So make sure you know every thing you have written in your code. We are least concerned about how you have learnt something as long as you have learnt it well.

Acknowledgments

The manual has been written by Ms. Shehzeen Malik and Mr. Muhammad Hammad Bashir.