Name:
Computer Design with Programmable Logic
CST 231
Lab Assignment 5: UART Transmitter
Due: Check course website

This lab assignment is to be completed with a partner. You are allowed to use course materials, the internet, and ask the instructor or teaching assistant for limited assistance. You should read through the nentire lab before completing it.

You have one week(s) to complete this lab. It is your responsibility to manage your time in such a way that your lab is complete before the due date without requiring last minute instructor assistance or checkoff. While instructor will attempt to assist and check off the lab at the last minute, there is no guarantee.

1 Introduction

In this lab, you will build a UART transmitter. The UART transmitter shall adhere to the following specifications:

1. 9600 baud
2. 2 stop bits
3. 1 start bit
4. 8 data bits
5. Even parity bit

Your project should be organized in the following manner:

1. Transmitter module
2. Clock divider for transmitter

2 Instructions

2.1 Task 1:Develop the clock divider for transmitter

1. Your master clock on the board is 50 MHz or 100 MHz. use the clock divider class notes to determine the size of register required, as well as the count-up value.
2. Develop a 9600 Hz clock divider for the transmitter. The clock should have a 50 percent duty cycle.
3. After developing the clock divider, program it to the board and test the clock divider by taking a screenshot on the oscilloscope. Make sure to grab at least 2 cycles and perform a measurement to indicate that it is indeed operating at 9600 Hz. (2021- you can't do this, so have the instructor look at the numbers before continuing).

2.2 Task 2: Develop the transmitter

1. Note that you are free to design your own state machine and that the state machine diagram pictured may not be exactly the one you require.

2. The transmitter is easier to code than the receiver. Please implement the state machine diagram as listed below, noting that you can eliminate the stophalfbit state and the arc from stop1bit to start. You can also eliminate the parity state. This transmitter state machine diagram is from Lattice Semiconductor. Please use the implementation as listed below as a starting point.

   THR is the transmitter hold register, which holds data to be transmitted. This is the same as my 'data' signal below.

   The transmitter module should have the following items in the module declaration:

```
module tx(
input clk  // The 9600 Hz clock that you generated in task 1
input go // 1-bit input signal to tell the transmitter to transmit
input [7:0] data  // The 8-bit data to be transmitted
output reg dataout // 1-bit serial transmission output
);
```

3. Make a block model of the transmitter and show the inputs and outputs coming out of your block model of the transmitter. (1 block) This will need to be submitted.

   Start out in start state until you get the 'go' signal (This is different from the way they implemented it). In the start state, we keep outputting '1' to dataout to signify that the line is idle if we do not get the 'go' signal.

   When we receive the 'go' signal, take the data from the data bus and load it into a register. Also since we are transmitter, drop the output line from '1' to '0' to indicate a start bit.

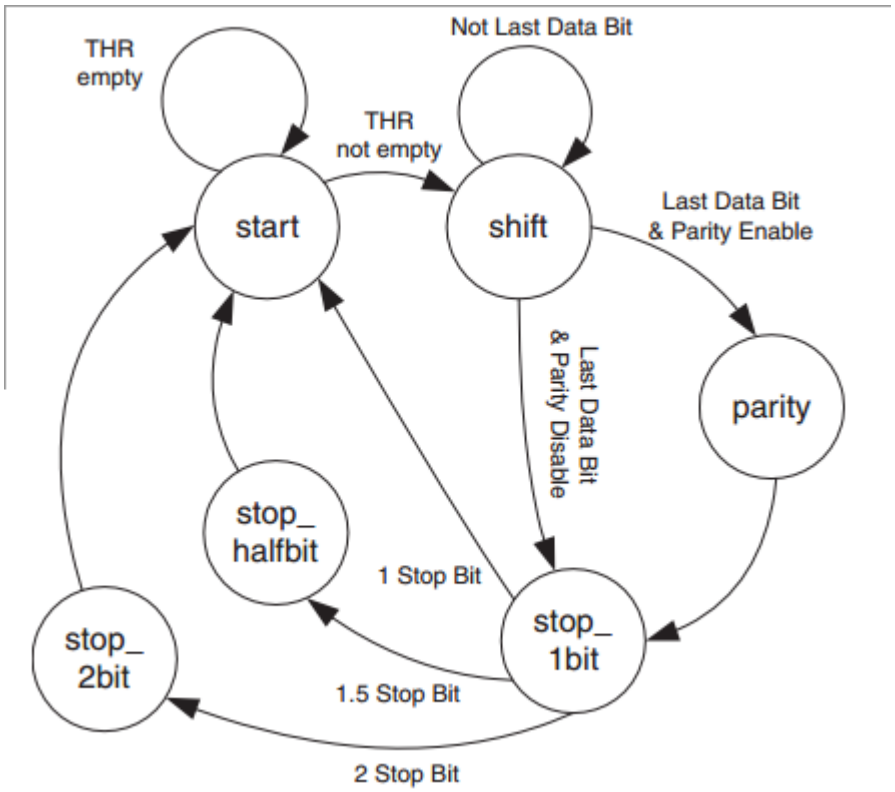
   Moving to shift state, keep shifting out data on each clock cycle until the counter indicates 8 data bits have all been transmitted.

   If all data bit have been transmitted, go to the first stop state and transmit a '1' to indicate a stop bit.

   Then move to the second stop state and transmit another '1' for two stop bits.

   Go back to start state and wait for idle.

4. Demonstrate to the instructor that your design can transmit one character from the FPGA through a USB to UART converter. It must be received by a computer terminal application such as Putty or HyperTerminal. If you do not know how to setup Putty or HyperTerminal, ask the instructor for assistance.



(Lattice UART application note)

2.3 Task 3: Transmit 5 characters at a time

Setup a module such that once you hit one of the buttons on the DE10-Lite, that you can transmit 5 different characters at a time with no glitching.

3 Submission

Electronically submit the lab.

1. Submit your cleaned and zipped project files. Remember to comment.
2. In a word document (upload this independently of the zipped file), include the following items.

   Complete calculations for your clock divider, bit size, etc.

   Block model of transmitter

   Your detailed state machine diagram for transmitter.

   State machine for the 5-character transmission module.

   Include your RTL schematic.

   Screenshot your resource allocation - how many Logic Elements did you use?