EE 3610 Digital Systems

Lab 2

Title: Asynchronous Serial Transmitter.

Objective: The student will understand basic buffering and how to design a digital system to transmit serial data.

Equipment: Spartan 3E Starter Board
9-pin D-Sub Cable.
Computer Configured as a Terminal Emulator

Background: In this lab, you are to design a system that receives a character from the serial port, displays it on the LED and LCD, and transmits it back to the sender over the RS-232 connection.

This system is very similar to lab1. There will be a module to receive a byte from the RS-232 connection (from lab 1), a module to display that byte to the LCD (provided in lab 1), a module that transmits a byte over the RS-232 connection (which you will write), and a top level module that ties them all together (which must be called lab2.vhd but can be largely be plagiarized from lab1.vhd).

The transmit module should contain a transmit buffer register (TBR) and a transmit shift register (TSR). The output of the TSR is connected to the transmit line (TXD). The TSR can only be loaded when it is idle (e.g. after a stop bit has been sent).

All characters to be sent are transferred first to the TBR then the TSR. If the TBR is holding a character when the TSR is idle, the character is transferred from the TBR to the TSR and transmission begins. The TBR then becomes empty until another character is written to it, and the TSR becomes busy until the character and its stop bit have been sent.

Preparation: Write the title and a short description of this lab in your lab book. Make sure the page is numbered and make an entry in the table of contents for this lab.

Draw a schematic that includes only the components you will be using for this lab. Specifically, include the FPGA (but show only the pins you are using), the LCD module, the MAX3232 and the DB-9 connector (Hint. copy your schematic from lab 1 and add the transmit signal.) Include pin numbers or coordinates. Omit the power supply.

Since this system is similar to lab1, you should use lab1's VHDL code as a starting point. <u>Do not modify lab1.vhd or lab1.ucf.</u> Instead, copy lab1.vhd to lab2.vhd and copy lab1.ucf to lab2.ucf. Add a constraint to lab2.ucf for the transmit signal (Pin M14 of the FPGA – See page 60 of the user's manual). You will also need to add this signal to the port list in lab2.vhd (and change its module name). Finally, modify lab2.vhd so it connects to the new transmitter module described below.

Design a VHDL module to drive the transmit data line of the serial port. Your module should take four inputs: asynchronous reset, clock, parallel data (an 8-bit vector) and a signal that loads the TBR. (Expect this signal to be asserted for one clock cycle when the parallel data is valid). Your module should output the transmit data (TXD) and a single status bit that indicates whether the TRB is empty or full.

You will need a flip-flop to keep track of whether or not the TBR is full. You will also need a controller that repeatedly waits for TBR to be full, transfers the data to the TSR (thus making the TBR empty), and shifts out the character (start bit, 8 data bits and a stop bit) at 9600 baud. Assume a 50MHz clock for now.

Write a test bench for your module that verifies (1) that it can transmit two characters back to back, and (2) that it reports the status of the TBR properly. Simulate the (transmitter) design and affix the simulation waveform to your lab book.

Make lab2.vhd the top level module and synthesize your design to verify that there are no syntax errors.

Bring your lab notebook and the Spartan board, above, to your lab period.

Set up: Connect the USB cable, serial cable and power supply to the Spartan board. Connect the other end of the serial cable to the computer and run a terminal emulation program. Configure the serial protocol to be 9600 baud and have 8 data bits, no parity and 1 stop bit.

Procedure: Download your code to the Spartan board. Verify that each time a key is pressed, the same character is echoed. Demonstrate your system to the lab instructor.

Conclusions: In the conclusion section, write a short summary of what you did, what you learned, and what could be done better.