EE 3610 Digital Systems

Lab 1

Title: Asynchronous Serial Receiver.

Objective: The student will become familiar with integrated design tools and how to use them to design a component of a digital system. The student will also gain experience with asynchronous serial communication.

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

Overview: In this exercise, you will create a digital system that receives data over a serial port and displays it on the LEDs and the LCD display. Fortunately, the digital circuitry that handles the LEDs and LCD already exists, so it is only necessary for you to design the component that receives the serial data.

Software: You will be using an application called ISE to design your circuits this semester. This application is available in the lab (Room 418), but it will be to your advantage to download it on your own computer if possible. The application is enormous, so make sure you have at least 20 GB free on your hard disk before you start. You can find the download page for version 13.4 (the same as in the lab) at:

http://www.xilinx.com/support/download/index.html/content/xilinx/en/downloadNav/design-tools/v13_4.html

You will be directed to create an account and after the download is complete you will need to request a Webpack license. (The Webpack license is free and never expires, but it does have limitations). This process takes time, so it is recommended that you start several days before lab.

Preliminary: When you start ISE, you will need to create a new project. This will create a project file and, if necessary, a folder to put it in. You will use this project file and folder for the rest of the laboratory exercises this semester. When you create your project you specify the hardware to be used. In this case, select the Spartan 3E Starter Board (or individually select Spartan3E, XC3S500E and FG320). Also, make sure the top level source type is HDL and the preferred language is VHDL.

For this exercise, you will design just the component that receives bytes over a serial port. The rest of the design is provided on the course website. Download lab1.vhd, lcd.vhd and lab1.ucf from the course website and add them to your project (use Project→Add Source). Then create a new VHDL

module called receiver.vhd (use Project→New Source). In order too fit with the other modules, the entity declaration must be as shown below:

```
entity receiver is
    port ( clk : in std_logic;                          -- clock input
           reset : in std_logic;                        -- reset, active high
           sdata : in std_logic;                        -- serial data in
           pdata : out std_logic_vector(7 downto 0);    -- parallel data out
           ready : out std_logic);                      -- ready strobe, active high
end receiver;
```

The clock (clk) has a frequency of 50MHz (for now), the reset (reset) is active high, the serial data in (sdata) will be discussed presently.

Each time a byte of serial data is received, the data must be placed on the parallel data out (pdata) and the ready strobe (ready) should be driven high for one clock cycle. (Note that for cycles when ready is not asserted, pdata may be anything.)

In order to know what to do with the serial data input, you must understand asynchronous serial communication. This means of communication often uses only one signal line and data bits are transmitted one at a time at a predetermined baud (bit) rate. When no data are being transmitted, the line is idle and is said to be in a “mark” state. To send data, the transmitter sends a start bit (which puts the line in a “space” state for one bit time). After that, data bits are sent one at a time, least significant first, where space = 0 and mark = 1. Finally one or more stop (mark) bits are sent (leaving the line in the mark state). Once it was common to transmit fewer than 8 bits and to send a parity bit prior to the stop bit(s), but that is almost never done today. For this lab, you will use a baud rate of 9600 (bits per second), 8 data bits and no parity.

On the 9-pin D-sub connector, the mark is a negative voltage and a space is a positive voltage, but since the Spartan 3E board uses a MAX3232 to drive and receive these signals, the polarity is reversed on the pins of the FPGA (mark = high, space = low)

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 diagram 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 MAX3232, the DB-9 connector, the LEDs and the LCD module (See, for example, sheets 2 and 13 of the Spartan 3E Starter Board Schematic.) Include pin numbers or coordinates. Omit the power supply.

Design a VHDL architecture to implement the entity declaration, above. Your design should monitor the serial input (sdata), and once a space (0) is detected it should wait ½ a bit time and re-sample the line. If the line has returned to the mark (1) state, your design should ignore the earlier space

and return to monitoring the line. Otherwise, it should sample the line nine more times (once per bit time) and shift those bits into a shift register. There will be 8 data bits and a stop bit. (The stop bit should be 1. If not, the system must wait for sdata to return to a mark state before looking for another start bit.) The shift register may be connected directly to pdata, but care must be taken to assert the ready strobe at a time that pdata is correct.

You will need to make use of the 50MHz clock. It will not divide by 9600 exactly, but if the bit time is within 1% it will be close enough.

Write a test bench for your module that verifies that it can receive at least two characters, Simulate the design and affix the simulation waveform to your lab book.

Making sure lab1.vhd is the top-level module, synthesize your design and verify that there are no errors. Generate a programming file.

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 (e.g. puTTY). Configure the serial protocol to be 9600 baud and to 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 appropriate ASCII character is displayed on the LEDs and on the LCD. Debug if necessary.

Demonstrate your system to the lab instructor.

Print copies of receiver.vhd, lab1.vhd and lab1.ucf and affix them to your lab book.

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