CSE 352 - Lab 8

Your lab 8 project is to develop a software program that reads from a serial input (UART) and displays the characters received on an LCD screen. We will give you a simulation and a bit-file for the y86e processor so that you can develop the software.

You will have until Final date at 2:00 PM Monday, June. 4, to complete this lab. You are allowed to work in group of 2.

Here is the outline of this lab:

Step 1 - Understanding the y86e embedded processor

The following document provides the specification for the y86e processor.

Specification for the y86e embedded processor

Step 2 - Compile your design to the FPGA

Here is a link to our solution to the y86e processor as an archived Aldec design. This design includes not only the y86e processor, but also the infrastructure around it that allows us to compile and run it on the FPGA.

y86ehw.zip

This is all described in the following document.

Using the hardware version of the y86e processor

Step 3 – Understand the software driver for the LCD screen

The following document describes the operation of the LCD and how your program needs to operate.

Developing the LCD driver program

You DO NOT have to implement this part. The solution is given here:

lcd.ys

Step 4 - Implementing the UART driver

The final piece of the project is the UART driver. This is the piece that you need to get working. This connects to a serial line which receives characters from any source that has a serial connection. Your program will display the characters received on the UART on the LCD screen. The following document describes how the UART works and how your program should implement the UART.

Developing the UART driver program

Demo/Turn In Deadlines

2:00 PM Monday, June. 4 - Demo your UART + LCD program running on your y86e processor
Turn in your UART + LCD assembly program to the Drop Box.

Extra Credit

There are a lot of interesting ways to extend this project - here are some ideas:

1. Use more of the functionality of the LCD - see the LCD documentation for details.
2. Implement the fastest receiver you can, e.g. 115K baud or even faster.
3. Implement an adaptive baud rate receiver. The receiver is sent the character 'U' as the first character, from which it determines the baud rate of the sender. (The LCD test fixtures sends the string "UHello!")
4. Implement a serial transmitter in addition to the receiver and connect it to another project.
5. Your own idea.