# EE 3610 Digital Systems

## Lab 7

Title: Final Project.

Objective: The student will gain experience integrating component modules to produce a working digital system.

Equipment:
Spartan 3E Starter Board
PC Keyboard with a PS/2 Interface
VGA monitor that accepts the 1024x768 XGA format
Computer Configured as a Terminal Emulator

Background: In this lab, you are to design a module that interprets ASCII characters and interfaces to the modules you created in Lab 1 and Lab 6. Then you will integrate all the modules you have designed this semester together to construct a “dumb” terminal.

A dumb terminal maintains a “cursor”, which denotes the next location in character memory to write. Usually the position of the cursor on the screen is shown with a flashing underline or a rectangle the size of one character. (On the puTTY terminal emulator, for example, the cursor is displayed as a green rectangle.)

ASCII defines some 33 control characters, but we will interpret only those that affect the position of the cursor:

Table 1. Control Characters that must be Interpreted.

| Character | Code (hex) | Description |
|---|---|---|
| CR<br>(CTRL+M) | 0D | Carriage Return – move cursor to the start (leftmost character) of the line. |
| LF<br>(CTRL+J) | 0A | Line Feed – move cursor to the next line down. Scroll the screen if necessary. |
| BS<br>(CTRL+H) | 08 | Backspace – Move the cursor one character to the left unless the cursor is already at the start of the line. |
| FF<br>(CTRL+L) | 0C | Form Feed – Clear the screen and move the cursor to the start of the first line. |

All other control characters may be ignored. Printable characters should, of course, be placed into character memory and the cursor should then be moved one space to the right.

If a printable character is written to the last column on a line, normally a dumb terminal will move the cursor to the beginning of the next line (as though CR and LF had been received), but to simplify this lab, you may elect not to move the cursor after depositing the last character on a line.

Also, you are encouraged (but not required) to display a visible cursor. The easiest way to generate a cursor is to modify your character generator from Lab 5 to take the cursor location as input and to generate an output that is asserted when the VGA interface is outputting the character at that location. You can then use that signal to, for example, drive the “green” output or to invert all the color signals.

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 crystal, the MAX3232, the DB-9 connector, the PS/2 connector and the VGA connector (See the Spartan 3E Starter Board Schematic.) Include pin numbers or coordinates. Omit the power supply.

Design an ASCII interpreter module that interfaces to your RS-232 receiver and your character memory module. Decide which inputs and outputs you will need for the module. Internally, the module must have a register that holds the current cursor location. When a printable character arrives, your module should write it to character memory at the current cursor location then move the cursor. If instead one of the control characters in Table 1 arrives, the module should move the cursor accordingly (along with any scrolling or clearing that is necessary). If any other control character arrives, your module should ignore it.

Write a test bench for your module. Verify that the reception of a printable ASCII character causes the appropriate signals to be sent to the character memory module (which should not be in the test bench). Verify the control characters in Table 1 cause the appropriate change to the cursor location and generate the scroll and clear signals as needed. Affix your simulation results to your lab book.

Design a top-level module that contains all the modules you have designed this semester. A good place to start is by merging the top level modules from Lab 3 and Lab 6. Include your RS-232 receiver from Lab 1 and the ASCII interpreter module you designed for this lab. The RS-232 receiver should be connected to the ASCII interpreter, which should then be connected to the character memory module. The other connections should be the same as in Lab 3 and Lab 6. 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, PC keyboard, VGA monitor and power supply to the Spartan board. Connect the serial cable from the terminal emulator to the DB-9 connector.

Turn on power to the monitor and the Spartan board.

Procedure: Download your code. Type characters on the PC keyboard and verify they are displayed on the terminal emulator. Particularly check the control characters in Table 1 and make sure LF eventually scrolls the screen. Type characters on the terminal emulator (including the control characters from Table 1) and verify they are displayed correctly on the VGA monitor. Demonstrate your system to the lab instructor.

Affix the final copy of your ASCII interpreter module and your top-level module (in VHDL) to your lab book.

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