CoE 168 Machine Problem

LCD Interface

1 Introduction

The aim of this machine problem is to implement a system that integrates the LCD interface.

2 Specifications

As described in the provided user manual, driving an LCD requires complex timing of data and control signals. An LCD also has different operating modes which can be set during its initialization phase. For this machine problem, the LCD should be configured in 4-bit mode. That is, commands and data, which are 8-bit wide, are sent in 4-bit (nibble) format; the upper nibble is transferred first, followed by the lower nibble. To facilitate the issue of commands or data, the following signals are used by the LCD display module:

1. Data lines [7:4] (DB7-DB4)- These are input/output lines which corresponds to the upper half of a byte. These are the only lines used when the LCD is operating in 4-bit mode.
2. Data lines [3:0] (DB3-DB0) - These are input/output lines which corresponds to the lower half of a byte. These lines are not used and should be grounded when the LCD is operating in 4-bit mode.
3. Enable (E) - This is an input signal for starting a read or write operation.
4. Read or Write (R/W) - This is an input signal to indicate the direction of operation. This should be set to "0" when writing to the LCD, and "1" when reading from the module.
5. Register Select (RS) - This is an input signal to indicate whether the type of input that will be sent to the LCD is a command or data. Commands are used to perform different functions, such as clearing the display and moving the cursor to a different address. The specific command is identified using an 8-bit value and is accompanied by RS set to 0. In contrast, data are used to display characters on the LCD. To write a character, the 8-bit ASCII encoding of the character is sent and is accompanied by RS set to 1.

2.1 Initialization

The 100MHz clock generator in the Artix-7 35T FPGA Arty Evaluation Kit will be used to provide timing signals to the system. A global, low-asserted reset signal will be mapped to the reset button of the board. Upon assertion of the reset signal, the initialization procedure of the LCD will start. Immediately after initialization, the LCD should display your full name, with your given name on the upper part of the display and your last name on the lower part, for two seconds before clearing. The system should enter type mode afterwards.

2.2 Type Mode

In this mode, the user can display printable characters (characters with ASCII value between `0x20` and `0x7E`) on the LCD display. Upon entering the type mode, the blinking cursor should be positioned at the top line, leftmost position.

When slide switch SW0 is DOWN or in the OFF position, pressing either BTN3 or BTN2 will allow the user to browse the printable characters at the current position of the cursor, and pressing BTN0 will allow the user to write the current/selected character and move the cursor to the next space on the LCD display. Pressing BTN2 browses the characters from `0x20` to `0x7E`. Pressing BTN3 browses the characters in the opposite direction; that is, from `0x7E` to `0x20`. Upon reaching the character with ASCII value of `0x7E` (or `0x20`), pressing BTN2 (or BTN3) will enable the user to move to the character with ASCII value of `0x20` (or `0x7E`). The system is expected to write at the top line of the LCD first before proceeding to the bottom line once the top line is full. If the next space to be written is out of the visible area of the LCD, the LCD should automatically scroll so that the space will be visible. Upon reaching the final space, the cursor should wrap-around and move to the first space of the other line on the LCD. Pressing BTN1 should erase (i.e., overwrite with a blank space) the character written before the cursor and move the cursor back by one space.

When slide switch SW0 is UP or in the ON position, pressing either BTN3, BTN2, or BTN0 will allow the user to move the cursor. Pressing BTN3 or BTN2 should move the cursor left or right, respectively. If the user is already at the rightmost space of a line on the LCD, pressing BTN2 should move the cursor to the first space on the other line. If the user is already at the leftmost space of a line on the LCD, pressing BTN3 should move the cursor to the last space on the other line. Pressing BTN0 should move the cursor between the top and the bottom lines of the LCD. If the user is at the top (or bottom) line, pressing BTN0 should move the cursor to the bottom (or top) line.

If the cursor is on a space with a written character, pressing either BTN3 or BTN2 (while slide switch SW0 is DOWN) will browse the printable characters from the currently written character, and pressing BTN0 (while slide switch SW0 is DOWN) will overwrite the currently written character with the intended character. If the user did not press BTN0 (while slide switch SW0 is DOWN), the currently written character should not be overwritten.

3 Submission and Deliverables

The grading of this machine problem is as follows:

- LCD initialization (20%)
  - The LCD is initialized properly and is able to display your full name.
- Type mode (70%)
  - The system supports writing of printable characters. (40%)
  - The system supports both cursor movement and scrolling. (30%)
- Documentation and source code submission (10%)
  - Submissions will be done through UVLê. Deliverables include all source codes for the RTL model and a three-page documentation (in PDF format) of how your design works. Place all required files in a single archived *zip*, *tar*, or *tar.gz* file. Make sure that all the files within the archive are located in a flat structure, with no subdirectories/folders containing the required files. The filename format for the archive should be *CoE168_MP_<Last_name>_<Initial_of_given_name>.<zip*, *tar*, or *tar.gz>*. For example, *CoE168_MP_Galapon_F.tar.gz* or *CoE168_MP_Tuso_K.tar*.

Final checking of this machine problem will be done on December 16. No late checking will be done. Your score will be based on what milestones you achieve on the checking date. The documentation and all source codes should be submitted on December 17.