Digital Design and Computer Architecture LU

Lab Exercise II

Robert Najvirt, Thomas Polzer[1]
{rnajvirt, tpolzer}@ecs.tuwien.ac.at
Department of Computer Engineering
University of Technology Vienna

Vienna, October 25, 2016

[1]Original version authored by Jakob Lechner and Thomas Polzer

1 Overview

In this exercise you have to design your own state machines for controlling the 2-line text LCD module which is available on the DE2-115 FPGA board. Designing decent state machines is a key issue for many hardware circuits.

Note that this document is not the complete assignment. Please view the protocol template for all required measurements, screenshots and questions to be answered.

2 Required Reading

- Design flow tutorial
- VHDL modeling slides
- State machine slides
- IP core documentation
- DE2-115 manual
- LCD module manuals
- SignalTap documentation

3 Task Description

Task 1: External Interface The LCD is controlled by sending specific commands, which are then executed by the internal controller chip of the display. After power-up the LCD-module first needs to be configured with a certain sequence of commands. Use the commands shown in Listing 1 to set-up the display in 2-line mode. Since the FPGA is connected to all data inputs of the LCD module, you can use the 8-bit interface. This allows you to send a command word (8 bits long) in a single write access.

```
1. "00110000" -- Function set (8-bit interface)
2. "00110000" -- Function set (8-bit interface)
3. "00110000" -- Function set (8-bit interface)
4. "00111100" -- Function set (8-bit interface, 2 lines, 5x8 font
5. "00001111" -- Display on
6. "00000001" -- Display clear
7. "00000110" -- cursor moving right, display shift off
```

Listing 1: Initialization sequence.

Design and implement a state machine (3-process method Moore FSM recommended), which is able to perform write accesses in order to send commands to the LCD module. Typically the busy flag of the module should be checked before sending a new command. This, of course, would also require you to implement support for read access on the LCD. For sake of simplicity, you can skip the read out of the busy flag and simply use a timeout mechanism to wait for the execution of the last command to complete before sending a new one. The execution times for all commands can be found in the LCD module manual. After reset this state machine should send the initialization commands given in Listing 1 and then wait for user commands for printing text on the LCD.

Task 2: Internal Interface Section 9 of the IP core manual describes the interface of the video controller, which is used for printing text on the 4.3" touchscreen of the lab boards. Your task for this lab exercise is to design and implement a controller for the 2-line text LCD display, which provides the same interface as the existing video controller. More specifically, this means that the interface signals need to be identical on bit-level (see Listing 17 of IP core manual) and the controller needs to implement the same interface behavior (see Section 9.6 of IP core manual). With an identical interface, the existing video controller and your LCD controller can be used interchangeably.

As presented in the IP core manual the existing controller implements a simple instruction interface (Table 23). Your LCD controller should support all of the specified instructions. For some instructions, however, we introduce minor exceptions from the original behavior, due to the limited capabilities of the 2-line text LCD. Instructions with differing behavior are listed in Table 1.

Design and implement a state machine (3-process method Moore FSM recommended), which processes internal instructions and translates them into one or multiple LCD commands. Use the state machine designed in Task 1 to send these commands to the LCD module.

Task 3: Integration Add the new LCD controller to the top-level design of the project of Lab Exercise I. Connect the *instr*, the *instr_data* and the *wr* inputs to the existing signals, so that the characters produced by the keyboard input will be displayed both on the touchscreen and on the text LCD. Obviously, the *busy* outputs of both display controllers need to be joined by an OR. Synthesize the whole design and download it onto the FPGA board for testing.

Task 4: Simulation Write a testbench for the controller you have designed and perform a behavioral simulation. Simulate displaying one character (INSTR_SET_CHAR) and clearing it again (INSTR_DELETE).

Instruction	Description
INSTR_SET_CHAR	The attributes bits in the instruction data input can be ignored. Furthermore it is not necessary to implement scrolling up the displayed text, if the end of the second line is reached. In this case the cursor can be returned back to the beginning of the first line. Whenever the cursor is moved to the beginning of the next line, clear all characters of this line.
INSTR_DELETE	Bits for background color can be ignored.
CLEAR_SCREEN	Ignore all parameters of the instruction data input. Simply empty the LCD display.
INSTR_MOVE_CURSOR_NEXT	Ignore background color parameter. Whenever the cursor is moved to the beginning of the next line, clear all characters of this line. If the cursor is at the last position of the second line, simply jump to the first position of the first line (no text scrolling).
INSTR_NEW_LINE	Ignore background color parameter. Clear all characters of the new line. If the cursor is in the second line, simply jump to the first position of the first line (no text scrolling).
INSTR_CFG	No operation.

Table 1: Adapted LCD controller instructions.

For the protocol, take a screenshot showing the signals holding the states of both state machines as well as all signals of the internal interface, the external interface and the signals connecting the two state machines. The screenshot should span from the first command on the internal interface to the last command on the external interface.

Task 5: Measurement Use a SignalTap II Logic Analyzer to analyze the behavior of your design in run-time. Observe the same set of signals as in Task 4 and trigger on the first command on the external interface (to the LCD module) after the second instruction (from the beginning of the measurement) on the internal interface has arrived, not knowing which instruction it will be.

For the protocol, start the measurement and issue the same internal interface instructions as for the simulation in Task 4 (INSTR_SET_CHAR followed by INSTR_DELETE). Take a screenshot spanning the delete instruction on the internal interface and the first LCD command on the external interface.

4 Submission Specification

The results have to be submitted via myTI. The deadline is **November 18$^{\text{th}}$ 2016, 23:59**. Upload a tar.gz or zip archive containing the following items:

- Your lab protocol as PDF.

- The complete VHDL source code of the system (don't forget the test-bench).
- Your Quartus project (don't forget a cleanup!) or a TCL script creating the project (including the pin mappings!).

Make sure the submitted Quartus project is compilable. All submissions which can not be compiled will be graded with zero points!