

# Lab 6: Interfacing a Hitachi LCD Controller with the 8051

CpE 214: Digital Engineering Lab II

Last revised: March 18, 2013 (CAC)

In this lab, you will be given the datasheet of the Hitachi 16x2 LCD panel available on Altera DE-2 FPGA board. The datasheet contains all the details of the set of commands to be executed to initialize the display and to write specified characters to it. It also contains the delay information of different signals needed to communicate with the LCD controller. You will also be provided with the pin description and addresses of the IO's of the LCD controller on DE-2 board. Using all this information and your knowledge of the board and Quartus II, you will write a C program for 8051 to display your name on the LCD panel.

## 0.1 Outline and Concepts

1. Introduction to the LCD controller
2. Creating Top Level Design
3. Writing the C program
4. Testing with Cyclone II FPGA

## 1 Introduction to the LCD controller

How the LCD controller works, and more simply how to work with it, can sometimes be confusing. Here we will break down step by step what needs to be done to get it ready to write data to. First let's have a look at the primary lines that connect to the LCD itself in Figure 1.

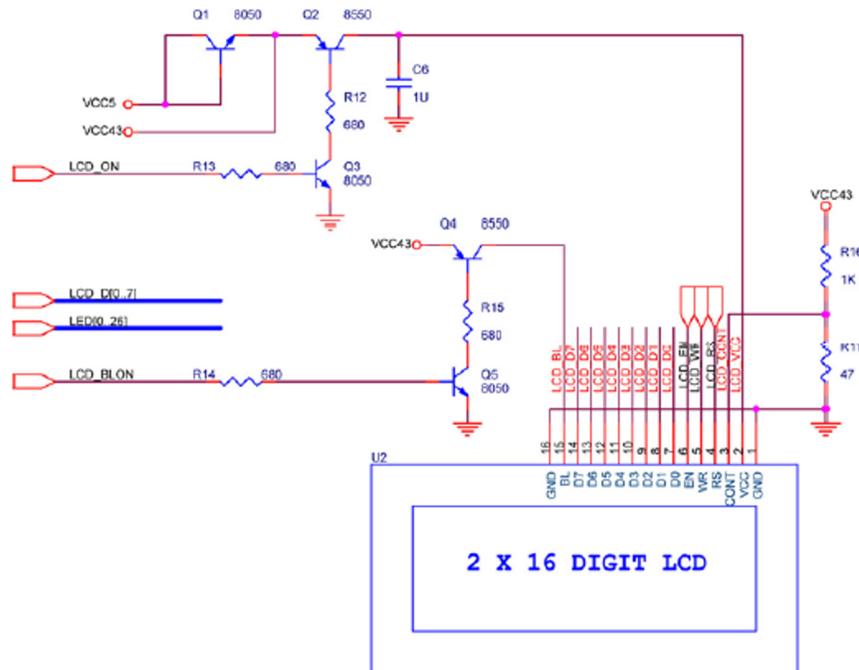


FIG. 1: LCD Controller electrical pin connections

Additionally, we can look more closely at a timing diagram for how the LCD controller actually uses the control lines to latch in data. This is shown in Figure 2 below.

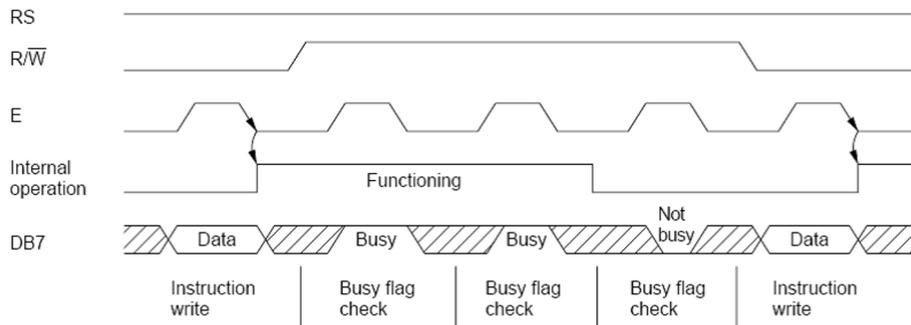


FIG. 2: Timing for LCD control lines and databus (DB)

Okay, so what exactly are these? The LCD controller has two main registers: the Control Register and the Data Register. All the commands are to be written to the Control Register and the characters to be displayed are to be written to the Data Register. The Register Select (RS) pin in the LCD controller selects the appropriate register. When RS is de-asserted, it writes to Control Register. All the initialization sequences are therefore written with RS=0. Other pins include Enable (E), Read/Write (R/W') and Data Bits (DB [7..0]).

- *In a nutshell, upon reset, you FIRST CONFIGURE THE LCD TO THE DESIRED MODE, then SECONDLY SEND DATA CHARACTERS TO IT.*

Configuration steps that need to be done are:

1. Enable power to the LCD, and Backlight (these are discrete pins that control Vcc to the chip with BJT's)
2. Turn on the LCD controller with the on/off instruction
3. Set the LCD mode (sets single or double line entry), and then LCD entry mode (sets cursor on/off, blinking, and which direction to shift the cursor as you write data).
4. Clear the LCD screen

After this, data can be sent/printed on the screen freely.

## 2 Creating Top Level Design

The top level design will be probably the simplest you have done so far. The standard 8051 modules and ROM symbol you are used to are all that is needed. Two ports will be used to send the control/setup information to the LCD controller, and one is used exclusively to send the Data, as shown in Figure 3.

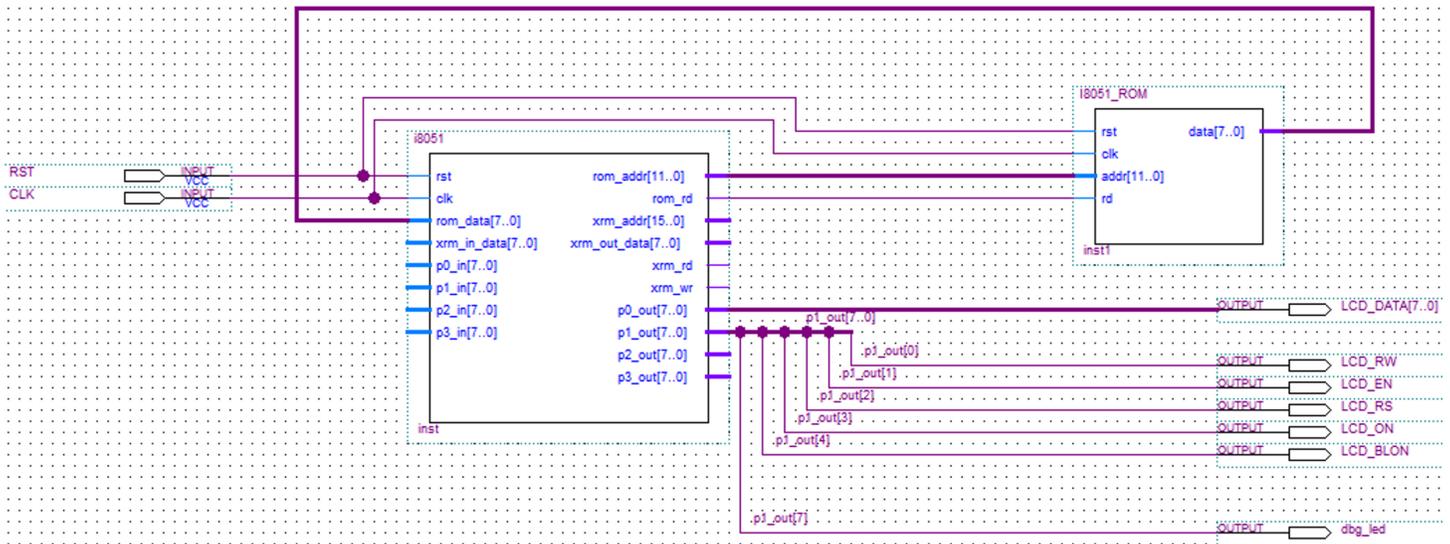


FIG. 3: Top level design schematic

When you finish, you should compile your design to check for any errors. *If the only error you receive is about I8051\_ROM, then everything is GOOD and you can proceed to writing your code.*

### 3 Writing the C program

As usual, a template to get started is provided on Blackboard. As you would guess, all of the control for the LCD is done in software. In the embedded world, it is crucial to develop the skill of being able to read datasheets to know how to write your control programs. A solid foundation in analog circuits/electronics, part of the CpE program's upper-level requirements, will help you in this endeavour.

- *Okay, but how do I know what I need to send to the display?*

You need to look at the Hitachi Datasheet (Appendix E) on page 24-25 which has the instructions you need for each function that is outlined in the previous section.

Take for example the Display on function: RS=0, R/W'=0, and for a blinking cursor we would write to the DB bus (P0 in our case) 0x0F. Enable is understood to always be on when you are doing something. It's just a latch for the data.

- *Once I get the LCD configured, how do I print characters to the screen?*

This is really simple once you've made it this far. On page 18 of the datasheet, you will find a character table (sort of like ASCII, but it's not the same) that has a rubric for what byte value to write to P0. There is a write function in the instruction set than is executed by setting RS=1 (Data Register selected), and R/W'=0 (write mode enabled).

## 4 Testing with Cyclone II FPGA

Before you test on the FPGA, you should compile your design first after you import your VHDL code back to Quartus. This way we can fix any errors first before you get to the board and something doesn't work.

1. Assign input pins N25 as your reset, N2 as your clock, and lastly your output pins and bus to the LCD display. Compile once more. Then download the design onto the Altera DE-2 FPGA board using the Programmer tool in Quartus II.
2. Observe the output and show the hardware implementation to your lab instructor. You should see your name sequentially printed, then the screen will clear as the program starts over.

- *What if it doesn't seem to be working?*

Almost always the issue is a simple fix. You probably have made a mistake decoding a configuration instruction (function set, entry mode set) into hex, or have altogether not included them. Show your code to the instructor and they can help you check.

## 5 Questions (attach at end of your report)

1. Describe the important control pins and initialization instructions of the LCD module.
2. Is the LCD controller edge triggered? Check the datasheet. If yes, what type?
3. What modifications would you make in your code in case you to make it scrolling (check the instruction set)?
4. Describe the issues you had in getting your name to display, and your solutions to them.
5. Explain the use of the bus DB[7..0].