# Lab #1
# Creating a SRAM interface for the DE2 LCD

**EELE 466**
**Computational Computer Architecture**

**Assignment Date: 1/20/15**
**Lab Due Date: 2/3/15**

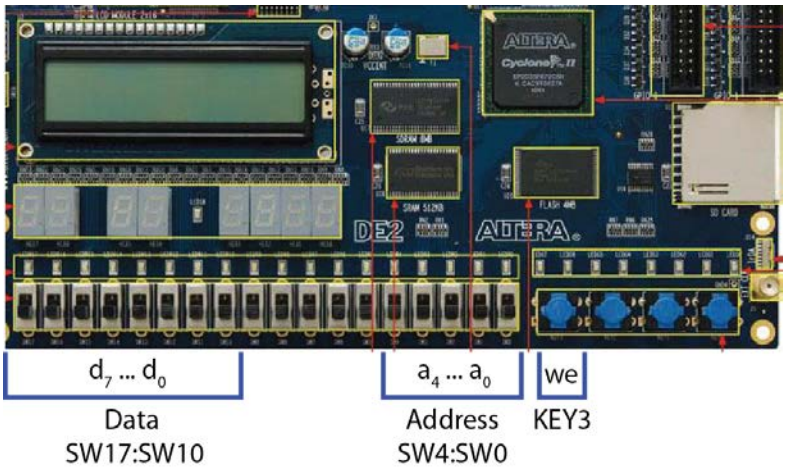
## Lab Description

The DE2 board has a LCD on it and the goal for the lab is to modify the file LCD_Display.vhd (which is one of the blocks found in chapter 5 of the text Rapid Prototyping of Digital Systems : SOPC Edition and is found on the D2L site) so that the LCD interface looks just like a SRAM.

Note: To use the DE2 board, you need to compile with Quartus II **v13.0** sp1 since later versions don't support the Cyclone II devices.

## Functional Goal

The functional goal of the lab is to:




1. **Data:** Enter a ***binary*** ASCII value on switches SW17:SW10. The ASCII ***decimal*** value should show up on the seven segment displays HEX2:HEX0.
2. **Address:** Enter the binary address location on switches SW4:SW0 as to where the character will show up. The first character on the first line should be associated with address 0. The last character on the second line should be associated with address 31. The address in ***hexadecimal*** should show up on the seven segment displays HEX7:HEX6. The address in ***decimal*** should show up on the seven segment displays HEX5:HEX4.
3. **Write:** A single write should take place when KEY3 is pressed and the character should show up on the LCD.

## Creating Dual Port SRAM Component

Use the Altera MegaWizard Plug-In Manager (Tools-> MegaWizard Plug-In Manager) to create a dual port memory. This can be found in the MegaWizard under Memory Compiler->RAM: 2-PORT, which is under the megafunction listings. Set the memory up with the following parameters:

1. One read port and one write port
2. 32 words of memory (8-bits wide)
3. Dual clock: use separate 'input' and 'output' clocks
4. Use a Memory Initialization File (.mif) to preload the memory contents with the following default string:

    a. First Line: Name of Lab Partner 1.
    b. Second Line: Name of Lab Partner 2.

    Matlab scripts are provided on the D2L site to create the .mif file. The Matlab scripts are (and will most likely require modifications):

    c. DE2LCDmeminit.m
    d. fixpt2alteramif.m

5. Select both the VHDL component declaration file and the Instantiation template file so that the VHDL code can be easily added to your design (added using Edit->Insert Template->Megafunctions).

## Key Press

You will need to ensure that the following occurs when KEY3 is pressed.

1. The asynchronous key KEY3 is synchronized.
2. The key press is debounced.
3. The input value is written *just once* when KEY3 is pressed (The write enable signal must be just one clock cycle long).
4. **The functionality listed in 1 to 3 must be encapsulated within a single VHDL component.**

The VHDL files/components that might be of help, which are described in chapter 5 of the text are:

- LCD_Display.vhd (the main file of interest)
- DEBOUNCE.VHD
- CLK_DIV.VHD
- ONEPULSE.VHD

## Submitting the Lab

On or before the lab due date have the instructor verification sheet signed off (given below). You will need to explain all the VHDL code that you wrote. If there are lab partners, each partner should be in charge of some aspect of the VHDL code. I will be asking each student what they contributed to the lab in terms of VHDL components. If your lab partner did all the work, they will get all the credit.

Upload your ***commented*** VHDL code to the D2L dropbox. Note: Your VHDL code should be distinct. I don't want to see code sections that look like they have been copied from someone else.

There is no lab report due for this lab.

**Instructor Verification Sheet**
Turn this sheet in to get credit for the lab.

# Lab #1
# Creating a SRAM interface for the DE2 LCD

**EELE 466**
**COMPUTER ARCHITECTURE AND SYSTEM ORGANIZATION**

**Assignment Date: 1/20/15**
**Lab Due Date: 2/3/15**

Name : _____

**Demo #1 :**

- Show that the LCD will display an ASCII character in the correct location when KEY3 is pressed.
- Show the names of the Lab Partners on each line at power-up.

Be prepared to explain your VHDL code.

- How did you synchronize your asynchronous inputs?
- How did you debounce your input signals?
- How did you ensure that the input value was written just once when KEY3 was pressed?
- How did you convert binary to hex/decimal for the seven segment displays?

Verified: _____ Date:_____