Lab 5 – TFT LCD Display using the Soft Processor

1 Objectives

1. Use a soft processor core in the Cyclone V SoC Device.
2. Interface to a Serial Peripheral Interface (SPI) connected colour 128x128-pixel TFT LCD display.

2 Introduction

In this lab we will use the NIOS II soft processor with an Altera SPI IP block to control a TFT LCD 128x128 pixel display.

Note: This lab works well on version 18.1 of Quartus Prime. There are issues with the "Software" portion of the lab on later versions of Quartus.

3 Prelab exercises

Review the datasheet for the ST7735S LCD Controller chip. Note that we will be using the 4-line serial interface and only performing writes to the display (simplex connection). Important sections to review are:

- 8.4 (4-line serial interface characteristics) – Is the data steady on the rising or falling edge of the clock? What is the minimum serial clock cycle (write) period? What would the maximum clock frequency be?
- 9.4.4 – What is the state of the clock signal when no data is being transmitted (this will help you determine the polarity bit)?

Review the documentation for the Intel SPI Core IP block which you will use to complete this lab. The documentation can be sound online at: https://www.intel.com/content/www/us/en/programmable/documentation/sfo1400787952932.html#iga1401317569928

Review all of section 5. Compare the waveforms in section ***5.3.3 Timing Settings*** to those given in the ST7735S datasheet and determine the appropriate values for the clock polarity and clock phase options.

4 Lab Activities

4.1 Hardware

Create a lab 5 project. Add the **`lab5.sdc`**, **`lab5top.sv`** and import the pin assignments (.qsf). In the Project Navigator window, select the Files view and right-click on **`lab4top.sv`** and select it as the top-level entity.

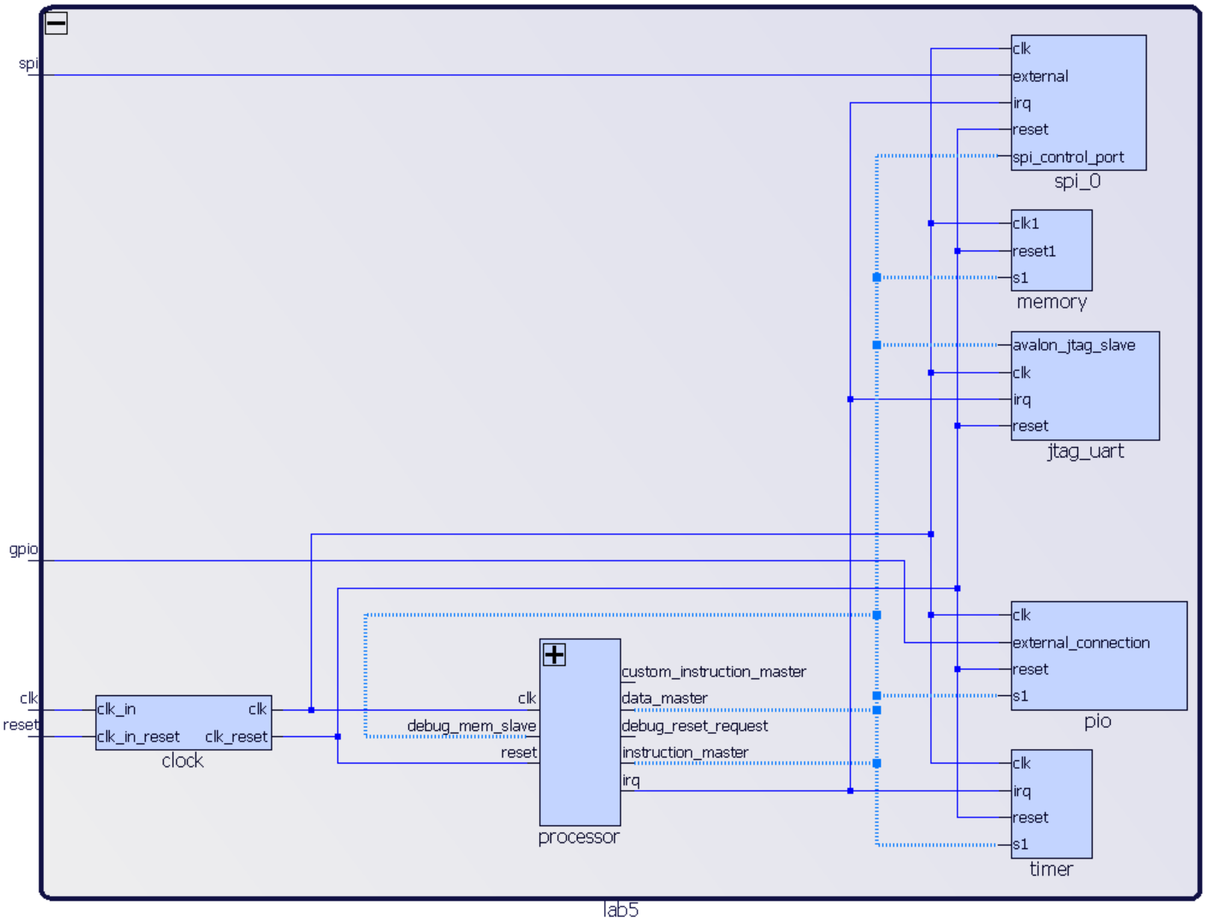
4.1.1 Generating the Soft Processor System

To save time and reduce errors, you will be supplied with a Platform Designer system configuration, **`lab5.qsys`**, that defines a simple Nios II processor system (the method to generate the configuration is shown in this tutorial https://www.youtube.com/watch?v=WGteIQqNfvY). You will edit this system to add your SPIr IP.

- Copy the **`lab5.qsys`** file to your project directory.

- Open Platform Designer from the Tools menu, and when asked to select a file to open, select the **`lab5.qsys`** file.
- In the “IP Calatalog” tab search field enter “SPI” and select the “***SPI (3 Wire Serial) Intel FPGA IP***” and click “+ Add…”.
- A window will popup allowing you to configure the IP that you are adding to your processor system. Set the number of slave selects, clock rate (use 5 MHz), data width, shift direction, clock polarity and clock phase and click Finish.
- In the System Components tab, click on the appropriate open circles in the “Connections” column to connect the clock, reset, and irq to the appropriate processor signals. The spi_control_port must be connected to the processor’s data_master bus.
- Double-click on the “Export” column on the conduit signal row for the SPI component and enter the name of the signal external to the processor (ie. `spi`); when finished there should be no error or warning messages left in the Messages tab (information messages are OK).
- Select System→Assign Base Addresses to generate a memory address for the SPI
- Select View→Schematic to see a schematic of the overall system. It should look similar to the figure below.
- Select File→Save to save your changes
- Select Generate→Generate HDL...; select an output directory in your project folder for the generated HDL files and click “Generate”; it will take a while to generate the Verilog files for the system’s IP components.
- Select File→Exit; you will be warned to add the .qip files to the project

Once you have exited the platform designer, select Project→Add/Remove Files from Project....; and add the .qip file which should be in the synthesis subfolder where the generated HDL files were placed.



4.1.2 Modify the Top-Level Entity

In the folder where Platform Designer generated the HDL files for the processor system, look for a file called **lab5_inst.v**. This file provides a template for instantiating the processor system generated by platform designer. Note that the template text <connected-to-...> must be replaced with a valid signal. Use the template to complete the **lab5top** module. Note that we will use the **s1** input as the reset signal for the CPU. The LCD signals should connect as follows:

LCD Signal	Connect to
lcd_sda	spi_MOSI
lcd_scl	spi_SCLK
lcd_cs	spi_SS_n
lcd_rs	gpio[0]
lcd_rst	gpio[1]

Be sure to connect the 8-bit internal **gpio** signal declared in the **lab5top** module to the **gpio_export** port of the processor.

Compile the project and program the FPGA.

4.2 Software

From the Windows Start menu run “Nios II Software Build Tools for Eclipse”. Select a subfolder of the project folder (e.g. software) as the workspace. It is very important that the path name does **not have any spaces or special characters in it**.

- select File → New → NiosII Application and BSP from Template
- Under Target hardware information select the name of the .sopcinfo file generated by Platform Designer in the Quartus project directory. (e.g. lab5.sopcinfo). The CPU name (processor) should get filled in.
- Enter a project name (e.g. lab5_software), select Blank Project as the Project Template, click Finish.
- Select File → Import → General → File System → Next and browse to the folder containing the C program provided (lab5.c and image.h). Under “Into folder” select the “lab5_software” folder and select ok and Finish.
- Select Project → Build All
- In the Project Explorer tab, right click on your project (**not** the project_bsp) and select Run As…→ Nios II Hardware.

You should now see an image on the display. Show the image to the lab instructor.

4.3 SignalTap

SignalTap is a debugging tool that allows you to monitor internal signals to assist with hardware debugging. Watch the following tutorial on SignalTap II Logic Analyzer tool which is part of Quartus Prime: https://www.youtube.com/watch?v=vhkzxCEXuaA

Enable SignalTap in your design such that it triggers on an **lcd_scl** edge when the **lcd_rs** signal is high. Include all the lcd signals in your capture (**lcd_rst, lcd_sda, lcd_cs, lcd_scl,** and **lcd_rs)**