ECE 3829: Advanced Digital System Design with FPGAs
A Term 2020

Lab 3: Light Sensor
Sign-off and Report due Sunday October 4th at 4pm

Use the provided Ambient Light Sensor module to create a light sensor monitor on the Basys3 board.

This project involves reading the light sensor using an SPI interface. The lab involves the use of multiple sequential circuits (counters, shift registers, and state machines), and the Xilinx Core Generator (for the MMCM). There are multiple parts to this project. To be successful will require a good design and debugging approach. Make simpler projects that you can test and debug separately and then combine them together.

This lab (and report) is to be completed individually – it will be worth 35% of your course lab grade.

Lab Signoff Deadline: during a lab section (during the week of September 28th) – The TAs will use the signoff sheet that describes what you have working. Submit your Verilog listings (as .pdf) at the same time as the demo and sign-off. The lab report is due Sunday October 4th at 4pm (no late reports accepted)

Description

Part 1: Light Sensor Interface

- Use an MMCM to create a 10MHz internal clock (clk_10m)
- Add your seven-segment display from module 1
  - You should not need to make any changes to this module
  - Display the last two digits of your WPI ID number on two of the seven-segment displays
- Create an SPI interface to be able to read the 8-bits of light sensor information from the PmodALS module provided.
  - Use the 10MHz clock with a counter and clock enable signal to generate an effective 1MHz for the CS# and shift register for the data
  - Verify the SCLK and CS signals are correct with an oscilloscope or through simulation.
- Capture a new light sensor value every 500ms (2Hz)
  - Use a shift register to read in the 8-bits of ADC data
- Display the light sensor value in hexadecimal on two of the seven-segment displays (00 to FF).
- Use a state machine for overall control of the light sensor interface
  - For example: Reset state, Wait state for 500ms, Read Light Sensor, Display Light Sensor value, and repeat cycle

- Optional: Capture an SPI ADC 16-bit transfer using an oscilloscope (show the CS, SCLK, and SDO signals on the scope capture) and include this in your report along with a description
  - Note: For all ‘scope pictures, preferably take a screen capture with a USB flash drive rather than a camera picture. You should be able to clearly see all the signals and the timebase.

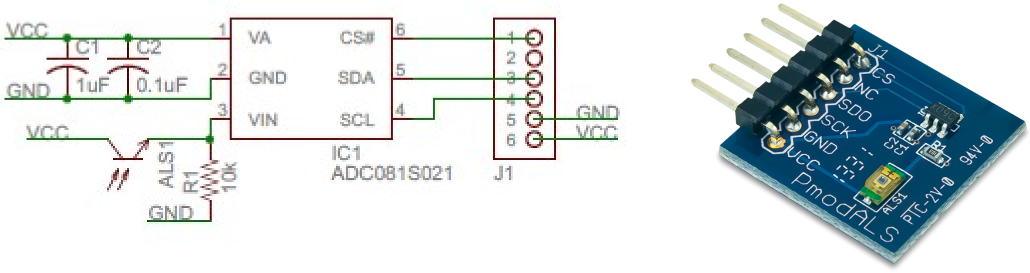
Part 2: Simulation and Testing (not required as part of demo and signoff but required for lab report)

- Create a test bench to show the following:
- Your state machine operation and an SPI cycle showing light-sensor data being transferred to the seven_segment displays (to speed up the simulation change the 500ms cycle time to 1ms)
- Important – describe and annotate the simulation waveforms

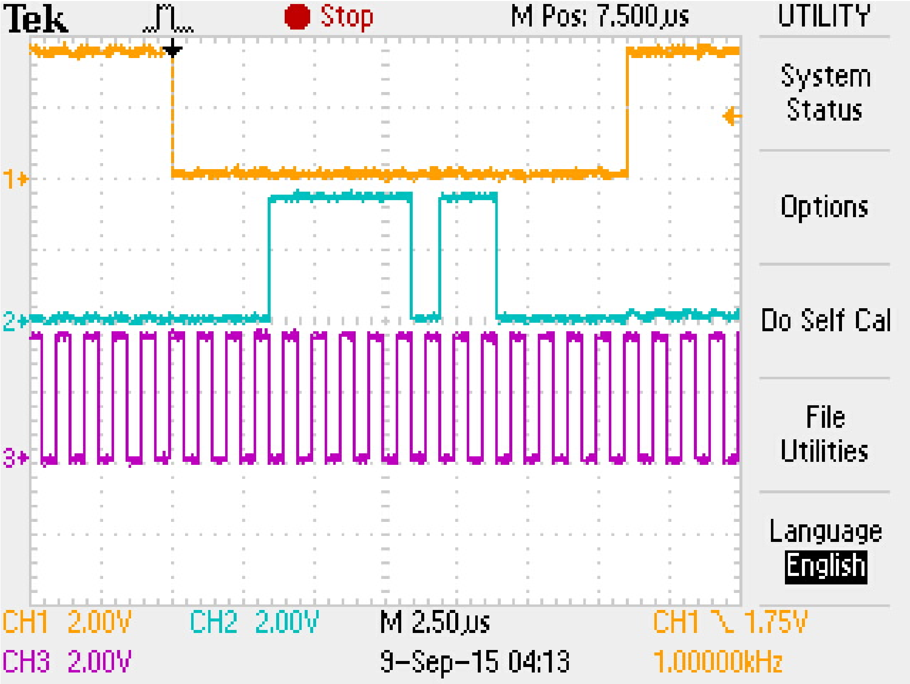
Extra credit

Up to 10% lab bonus points for any good improvements or enhancements to your design (must demo on board **and** describe in your report). For example, make a system that can count objects passing through a light beam.

Reference Material



PmodALS schematic and module from Digilent



CS#, SDO, SCK example SPI transfer (with CSK at 1MHz) - bright sensor value (0xFB)

Notes:

Demo your system during one of the lab sessions before the deadline. Have your Verilog source files ready so they can be uploaded to Canvas immediately after the demo (do not forget to include name, description, and comments).

Write a report including: an introduction, a description of your design including good block diagrams showing how you implemented the design, a section describing how many flip-flops your design used and why. Include part of the synthesis file that displays warnings – copy and explain these. Include a conclusion describing any problems or issues you had, and lessons learned. Include your Verilog test bench files in an appendix. Save the report as a PDF and upload to Canvas.

Grading Guidelines

- [40 pts] Implementation
    - [50 pts] Design works on board and meets requirements
- [15 pts] Source Code – (include Verilog Test Bench files in Appendix)
    - Code style and comments (well-commented and tab-indented code!)
    - Use of *case* vs. *if*, spaghetti code vs. structured, etc.
    - Recognizable implementation of "standard" elements (state machines, counters, shift registers, clock dividers, decoders)
    - Good modular design
    - No latches or other synthesis problems
- [45 pts] Lab Report including Test Bench
    - [5 pts] Brief Introduction / Problem Statement
    - [15 pts] General Overview of approach to solution and description and (include Block and State Diagrams with descriptions).
        - Optional: Include oscilloscope picture of DAC waveform.
    - [5 pts] FPGA Resource usage (# flip-flops with explanation) and listing and explanation of warning messages (do not copy all the Xilinx reports – just the relevant sections)
    - [15 pts] Test bench description and simulation waveforms (do not just copy the waveforms - annotate and explain well)
    - [5 pts] Conclusions
        - Problems faced in implementation
        - Solutions used to solve problems
        - Lessons learned from the project
        - Suggestions for further improvements and extensions
- [10 pts] Extra points
    - Possible extra points for good additional features or capabilities (need to demo on board and include description in report)

ECE 3829: Lab 3 sign-off sheet

Name: ____________________

Part 1 (Light Sensor)

WPI ID number on two seven-segment displays ______________________

The light sensor displays the correct value ______________________

on the seven segment displays (00 to FF, dark to light)

Extra Credit (describe)

____

____