ECE 574: Modeling and synthesis of digital systems using Verilog and VHDL
Fall Semester 2019

Design and implementation (in VHDL) of a VGA Display and Light Sensor to run on the Nexys4DDR board
Report and Signoff due Week 7 (October 9)

This project involves the design of a number of interfaces to peripheral devices. It includes reading a light sensor ADC using an SPI, and also driving a VGA display monitor. The project also uses the Xilinx Core Generator (for the MMCM) and use of existing IP (Digilent VGA Controller). There are multiple parts to this project. To be successful will require a good design and debugging approach. Make multiple simpler projects that you can test and debug separately and then combine them together.

Preliminary:

- Modify the simple seven segment display from the tutorial to create a seven_seg module that can display from "0000" to "FFFF" on four of the seven segment displays. The input to the module should be a 16-bit wide bus, with four bits used to indicate the value to be displayed on each of the seven segments.
- Make this a separate module – you will use this module in this and later projects.
- Test this out by using the slide-switches to enter various numbers.

Part 1: VGA display

- Use a MMCM to create a 25MHz clock required for the VGA pixel clock.
    - (see the MMCM tutorial for how to add this IP to your design).
    - Note: only connect the 100MHz FPGA clock to the MMCM (nothing else)
    - Add a period constraint to your XDC to match the Nexys4DDR board 100MHz clock frequency.
    - Use this 25MHz clock signal for all the sequential logic in this lab
- Create a VGA display using the VGA controller provided by Digilent (just the 640 by 480 version) – see information at end of this doc.
- Use the slider-switches to select and display the following patterns
    - Complete red display
    - Vertical bars of alternating blue and yellow colors with each horizontal bar 32 pixels wide

  (These should be relatively easy once you start working with the VGA controller provided by Digilent)
- Assume the VGA screen is divided into blocks, with each block 24 pixels high by 32 pixels wide, and the top left corner block is at block x,y position (0,0) and the bottom right block is at x,y position (19, 19)
    - Using the slider switches display the following:
        - A white colored block and place it at a starting position of (2, 8)
        - A red colored block and place it at a starting position of (11, 11)
        - A yellow colored block and place it at a starting position of (14, 3)
        - A blue colored block and place it at a starting position of (19, 18)

- Use two of the seven-segment displays and display the current ‘x’ position of the block in decimal (00 to 19)

Part 2: Light Sensor Interface

- Create an SPI interface to be able to read the 8-bits of light sensor information from the PmodALS module provided.
- Use the 25MHz clock with a counter and clock enable signal to generate the ADC SCLK at 1MHz
- Use a counter or shift register to create the ADC CS signal.
  - Verify the SCLK and CS signals are correct with an oscilloscope.
- Capture a new light sensor value every 100ms (10Hz)
  - 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 approx. FF).
- 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.

Combine all the parts into one project that you can demo to the TA.

As usual, this is not a complete description – make whatever additions or changes you think are necessary.

Prepare a sign-off sheet and demo your system and write your report before the deadline.

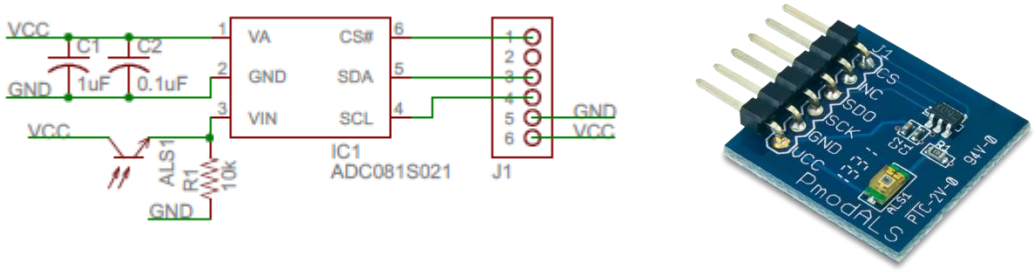
Reference Material

Note: To stop two warnings from being generated when you create a bit stream add the following two statements to the XDC constraints file:

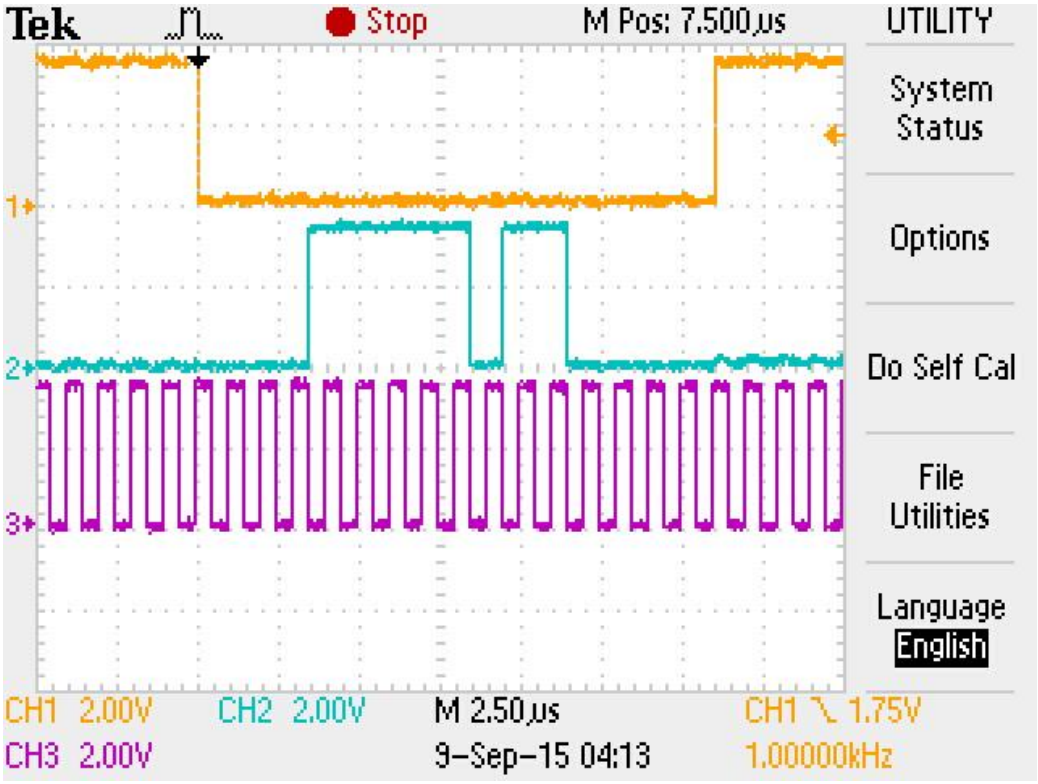
```
# Configuration bank voltage select (CFGBVS) must be set to VCCO or GND, and
CONFIG_VOLTAGE must be set to the correct configuration voltage,
# in order to determine the I/O voltage support for the pins in bank 0.
set_property CFGBVS VCCO [current_design]
#where value1 is either VCCO or GND
set_property CONFIG_VOLTAGE 3.3 [current_design]
#where value2 is the voltage provided to configuration bank 0
```

Read the *Seven Segment and VGA Port* section in the Nexys4DDR Reference Manual.

Read the Digilent PmodALS Reference Manual and the Texas Instruments ADC081S021 ADC data sheet.



PmodALS schematic and module from Digilent



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

Download the VGA Component Reference Design from Digilent from my website. This design is in VHDL. Do NOT modify this code, just instantiate it in your top-level.

```
----------------------------------------------------------------------------
-- vga_controller_640_60.vhd
----------------------------------------------------------------------------
-- Author : Ulrich Zoltán
--          Copyright 2006 Digilent, Inc.
----------------------------------------------------------------------------
-- Software version : Xilinx ISE 7.1.04i
--                    WebPack
-- Device              : 3s200ft256-4
----------------------------------------------------------------------------
-- This file contains the logic to generate the synchronization signals,
-- horizontal and vertical pixel counter and video disable signal
-- for the 640x480@60Hz resolution.
----------------------------------------------------------------------------
--  Behavioral description
----------------------------------------------------------------------------
-- Please read the following article on the web regarding the
-- vga video timings:
-- http://www.epanorama.net/documents/pc/vga_timing.html

-- This module generates the video synch pulses for the monitor to
-- enter 640x480@60Hz resolution state. It also provides horizontal
-- and vertical counters for the currently displayed pixel and a blank
-- signal that is active when the pixel is not inside the visible screen
-- and the color outputs should be reset to 0.
```

Grading Guidelines

- [50 pts] Implementation
  - [50 pts] Design works on board and meets requirements
- [20 pts] Source Code – VHDL 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 (for example: state machines, counters, shift registers, decoders)
  - Use of packages and other VHDL code structuring elements (sensible use of modularity)
  - No latches or other synthesis problems
- [30 pts] Lab Report
  - [5 pts] Brief Introduction / Problem Statement
  - [15 pts] General Overview of approach to solution and description and oscilloscope pictures (also include good Block and State Diagrams with descriptions)
  - [5 pts] FPGA Resource usage (# flip-flops with explanation) and listing and explanation of warning messages (don't copy all the Xilinx reports – just the relevant sections)
  - [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)