ECE437: Sensors and Instrumentation

Lab 7: Image Sensor – SPI Interface

Introduction

The next two labs will introduce the operational principles for the image sensor on the custom board. The image sensor is placed in the upper right corner of the sensor's board. We are using the grayscale version of the CMOSIS CMV300 image sensor. The image sensor has 648 by 488 pixels and can operate in two modes of operation: high read out mode achieving 480 frames per second and low read out mode of 120 frames per second. The low read out mode uses parallel data read out and we will focus on this mode of operation to acquire and display images. The datasheet for this sensor is located on the course website.

In this lab, we will focus on programming the imaging sensor using SPI serial interface. You will develop the finite state machine (FSM) for both reading and writing data to the image sensor from the FPGA. This will enable you to program the image sensor and setup the image acquisition process.

Relevant Documents for this Lab

Required reading material for this lab:

1. CMOSIS CMV300 image sensor data sheet is located on the course website. Read over the sensor's datasheet and get familiar with the serial interface protocol.

There is no additional reference material.

The Goals of This Lab Are:

The objective of this lab is to acquire video signal from the CMOS imaging sensor and display this information on the screen. The first step in this process is to program the image sensor via serial SPI interface. You will develop the FSM for reading and writing data to the image sensor.

Prelab Questions (30 pts):

1. What is the frequency range for the imager's master clock signal? (5 pts)
2. What is the maximum frequency for the SPI clock signal? (5 pts)
3. Draw the **SPI read block** diagram and describe how would you use this information to develop the FSM in Verilog? (10 pts)
4. Draw the **SPI write block** diagram and describe how would you use this information to develop the FSM in Verilog? (10 pts)

Checkpoint 1 (100 points)

You will write Verilog and Python code for serial (SPI) communication with the image sensor. The imaging sensor uses SPI communication to set various internal registers as well as to read the value stored in these internal registers. These internal registers control various parameters of the imaging sensor, such as number of pixels to be read out in the x and y direction (also known as the region of interest or ROI to be read out), analog to digital converter gain control, and others. Your FSM will communicate with Python and send/receive data to and from the PC. The timing information about the SPI protocol is provided in the imager's datasheet.

For example, from your Python code you will specify the register address, read or write mode of operation and value for this register. This information is sent to the FPGA, which runs the FSM for either reading or writing values to the specified register. If you are reading the register value from the image sensor located at address X, then the PC will send the register address X, followed by a read request to the FPGA. The FPGA will then run the read FSM and read the register data. The register data is sent to the PC and the results are displayed on the screen.

Here are the details for this milestone:

1. Once you complete the read and write FSM machine, program register Start1 located in memory address 3 and 4. Since the Start1 register is 16 bits wide, its data is split into two 8-bit registers. Program these two registers with an arbitrary value specified from Python.

2. Read the value that you programmed in register Start1 and show the results on the screen. Make sure that the data you wrote and read to the temperature registers is the same to get full credit. (50 points).

3. Read the temperature register data. Look up how to convert the temperature data from these two registers into readable data in degrees Celsius. To be able to read the temperature data, the image sensor will need to be clocked. Read over the documentation to figure out the imager clock pin and locate this pin in the XDC constrains file. You will have to generate a clock signal on this pin from your Verilog code. Choose two different clocking frequencies for the image sensor and demonstrate that your temperature result is the same for both of these operational frequencies. Furthermore, the temperature value should be around 30°C and your results should be around this value for full credit. (50 points).

4. Print out your Verilog and Python code and submit it as final report.