Nash Reilly

Snider/EELE 475

December 14, 2011

Final Project Report—Embedded Systems

**Introduction**

In the course of my senior design project, I have had to design a custom VHDL component to implement an algorithm for correcting data from an infrared camera. The need for this correction arises out of a material flaw with infrared cameras. Because of the nature of the optical sensing element within an infrared camera, the output of the camera can vary with temperature. While in some applications this is irrelevant, this particular camera is in use by the Optical Remote Sensing Laboratory at Montana State University. Since the data in question here is being gathered for the purposes of optical research, precise values of received radiance are needed for the purposes of the Laboratory’s research goals. This need initiated my senior design project, which in turn initiated my final project for EELE 475. To prove that the concept of correcting the data with a custom VHDL component is feasible, I developed a component to do the correction and paired it with a Nios II system to change values being fed into the calibration element.

**Hardware**

Included in the figure below is a block diagram of the component developed for the purposes of the project.

Figure 1: Calibration Block

(insert Visio drawing here)

The component’s purpose is to implement a fixed-point multiplication as well as a subtraction to the incoming pixel data, represented in the block diagram as the signal data. The algorithm’s output, data\_cal, is the result of the equation below, where D represents the digital number at the input, DC represents the calibrated data output, and G(T) and N(T) represent the gain and offset as functions of temperature.

This corrected value, ultimately, will be stored in a FIFO stack to await being transferred over a network connection. Given that this project is a proof of concept, it will serve as an output that will be captured by SignalTap or a physical logic analyzer for the purposes of verifying the output. The values for gain and offset displayed above as G(T) and N(T) will be fed into the system by the signals that share their name: gain and offset. Though the gain is treated as a 16 bit unsigned integer for the purposes of transmitting it throughout the circuit, it is actually used as a decimal value in a fixed-point calculation within the machine. Using Matlab’s fixed point toolbox, it was possible to determine that the gains needed for the system could be represented as a 16 bit value, which requires a radix point of 4 units above the most significant bit of the 16 bit integers. By programming this internally to