**EEL 4930/5934 - System-on-Chip Design**

**Spring 2025**

**Home Assignment 4 Report**

**GitHUb Link:** [**https://github.com/21BMcDaniel/SoC\_Project4\_BMcDaniel**](https://github.com/21BMcDaniel/SoC_Project4_BMcDaniel)

**Problems Faced:**

At first, I experienced difficulty using the OpenCV and SystemC libraries because I was trying to implement the OpenCV library in Visual Studio. However, after asking for some help from my classmates, I found that it would be easier to work with these libraries in VSCode and wsl. So, after I was able to download and set up these programs onto my computer, I was able to smoothly run OpenCV and systemC libraries. Throughout the project, several challenges arose that required problem-solving and adaptation. One of the primary issues was tuning the parameters for edge detection and line detection, which was critical for accurately identifying road lines in the image. The Canny edge detection algorithm had to be carefully adjusted to avoid both under- and over-detection of edges, and finding the optimal thresholds took multiple iterations. Additionally, selecting the correct Region of Interest (ROI) proved to be a challenge since the road lines were expected to be in the lower half of the image. Ensuring that the ROI was properly defined and processed required careful handling of the image’s dimensions and structure. Another problem encountered was dealing with the noise in the image, which was addressed through blurring techniques before applying edge detection. The line detection process using the Hough Transform also needed fine-tuning, particularly with the angle filtering, as it required correctly identifying and excluding irrelevant lines while preserving road lines. Lastly, managing the coordinates of the detected lines to properly adjust them to the full image’s scale was tricky, as it involved precise mathematical adjustments to maintain accurate positioning. These obstacles, while challenging, were eventually overcome with careful attention to detail and iterative testing.

**What you learned:**

Throughout this project, I not only gained hands-on experience in image processing and computer vision techniques, but also learned valuable skills in Transaction-Level Modeling (TLM). TLM was essential for simulating communication between different components in the system, allowing me to model interactions between the processor and image data effectively. Additionally, I became familiar with using Windows Subsystem for Linux (WSL) to run Linux-based tools on a Windows machine, which was crucial for managing dependencies and executing commands smoothly. Learning how to use Visual Studio Code (VSCode) was another significant part of my workflow. VSCode provided a user-friendly and efficient environment for coding, debugging, and managing project files. By integrating TLM with image processing, and utilizing tools like WSL and VSCode, I enhanced both my software engineering and system-level design skills, ultimately leading to a better understanding of how to develop and test complex systems.

**Other:**N/A