**Class Report 4:**

Task: 16.8.1 Inclination Sensing  
From: Oliver Jarvis and Ryan Morrison  
 Date: 2 November, 2023

The GitHub for this report is <https://github.com/OliverJarvis1/Class_Report_4> and a video of the working system can be found in the GitHub repository.

**Design Method**

The design for this project stems heavily from Pong Chu’s given code and example problem 16.8.1. We set up Chu’s timer, UART (for testing), LED GPO, Switch GPI and accelerometer SPI cores with the mmio\_sys\_vanilla and mcs\_top\_vanilla hardware. The LED GPO and accelerometer SPI cores are the main cores used in this assignment, as the accelerometer SPI core sets up the accelerometer system and the LED GPI core sets up the LED system.

**Implementation**

We used the GPO core and SPI core modules in Vitis given by Pong Chu. These files include the gpio\_cores.h, gpio\_cores.cpp, spi\_core.h and spi\_core.cpp. The main code was implemented in main\_vanilla\_test, which included the gpio\_cores.h and spi\_core.h files. The main code instantiated two cores: a GpoCore called led and SpiCore called spi. We used Pong Chu’s example code as our base, which set the frequency and mode of the spi core, checked the part id, and read the accelerometer data. The data came in three parts, the first being the raw x coordinates, then next being the raw y coordinates, and the last being the raw z coordinates. These values were then converted into a float value and divided by the maximum raw value. Which gives a value between -1 and 1. A series of if statements determine which of the four LEDs are turned on based on the x and y values given.

**Testing**

To test the system, we built the code and programmed the Nexys 4 board. To troubleshoot issues, different LEDs were turned on to determine which state the code was at (to see if any if statements were being entered or not). We also used UART to check the values we were receiving.