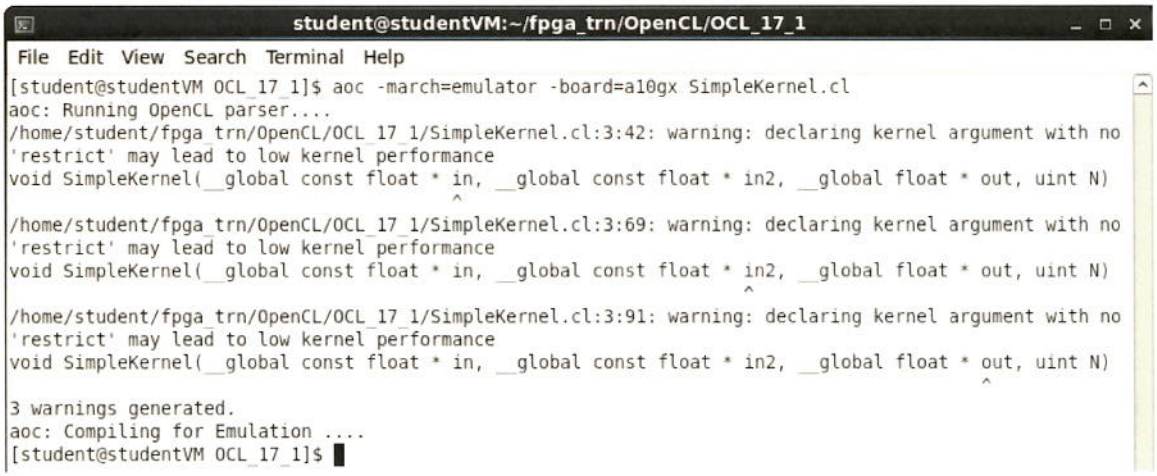

This is because the AOC compiler performs dependency analysis to optimize the pipelined circuit created. Multiple pointers can technically point to the same area, which prevents the compiler from creating the optimal circuit. Using the "restrict" keyword lets the compiler know that pointers won't point to the same area and that it can treat pointers as separate and make optimizations accordingly.

To resolve the warning, simply add the ***restrict*** *keyword to each of the pointers in the kernel.*

e.g. ***__global const float * restrict in***

Remember if you change the kernel code, you will need to rerun the aoc compiler on the kernel file.

Step 2. Write the host code that launches the kernel

____ 1. Reopen **main.cpp** in Eclipse.

____ 2. Find "#define EXERCISE1" around line 13 in **main.c** and comment it out.

Commenting enables the portion of the code that launches the kernel verifies the results.

____ 3. Find the comment "Exercise 2 Step 2.3". Right beneath it, write the code that would create the cl::Program object from the aocx binary. Use the context and devices list created from the first lab.

If we were not running in emulation mode, the variable named "mybinaries" would store the information from the .aocx file used to program the FPGA. However, in the emulation mode, the aocx file is just a software library that can be dynamically linked with the host code.

____ 4. After the comment "Exercise 2 Step 2.4", call the function that **builds** the program from the program created in the previous two steps.

For Intel FPGA, this is only done as a formality.