**Design Requirements**

In this lab, you will use performance counters to measure the performance of various aspects of your code from Lab 2.

First, make sure your CPU type is set to NIOS II/f and has support for hardware multiply and hardware floating-point. Next, add a performance counter to your system design.

Use performance counters to determine how many cycles are required on average to transform one pixel (using the rotation transform) using the following cache configurations for the NIOS II/f:

1. 4K instruction cache, no data cache
2. 4K instruction cache, 4K data cache
3. 4K instruction cache, 16K data cache

After this, replace all floating point operations with fixed point. Use at least four fractional bits and use rounding when converting to integers. After this, repeat the same tests as above.

Use preprocessor directives, such as `#define` and `#ifdef`, to make it convenient to switch between fixed point and floating point. For example:

```c
#define FIXED_POINT
...

#ifdef FIXED_POINT
    sin_val = (alt_32)(sin(angle) * 16.0);
    cos_val = (alt_32)(cos(angle) * 16.0);
#else
    sin_val = sin(angle);
    cos_val = cos(angle);
#endif
```

**Project Submission**

In addition, each group must submit a report that describes their performance results for each test case above.