ENGR xD52: Extra credit b000

This is a purely optional assignment. It presents an opportunity to showcase your skills in pipelining, fixed point math, and datasheet grokking.

Background

The TMS320C674x from Texas Instruments is a floating-point/fixed-point multi-core DSP with several execution units per core. It contains two DSP-style multipliers and six ALUs, which allow it to execute up to 8 instructions per clock cycle.

The Task

The DSP has a buffer full of I16 sensor readings that need to be normalized and converted to floating point. The C pseudo-code for this operation follows:

```
Int16 raw[];
float processed[];
float calibration;
for(int i=0;i<length;i++) {
     processed[i] = raw[i]/32767.0f – calibration;
}
```

As written, this C code takes very poor advantage of the DSPs capabilities. Find opportunity to better optimize this code, and explain

Hints

There is more room for improvement than hours left in your schedule for the semester. You do not need to find all of them for credit. Pick 1-3 optimizations and explain what their timing effect is and how you would do them. English is an acceptable substitute for assembly. **Credit for thought process.**

Unrolling the loop (and interleaving) reduces the impact of delay slots in your instructions. Your ability to do this is limited in turn by the functional unit latencies of the instructions.

There is no floating point division opCode in this architecture. Integer division is done in a conditional subtract loop. Doing an integer shift introduces errors past the log2(1-32767/32768) = 15^{th} bit and reduces the range of values to no longer include 1.0f. Can this be done with an integer multiply?

INTSP will convert an integer to a IEEE754 compliant float.

LDW and STW are their equivalent of LW and SW. LDW has 4 delay slots. You may assume a memory alignment of your choosing: Packed or Aligned.

Acknowledgement: This assignment is adapted from UCONN's CSE5095.