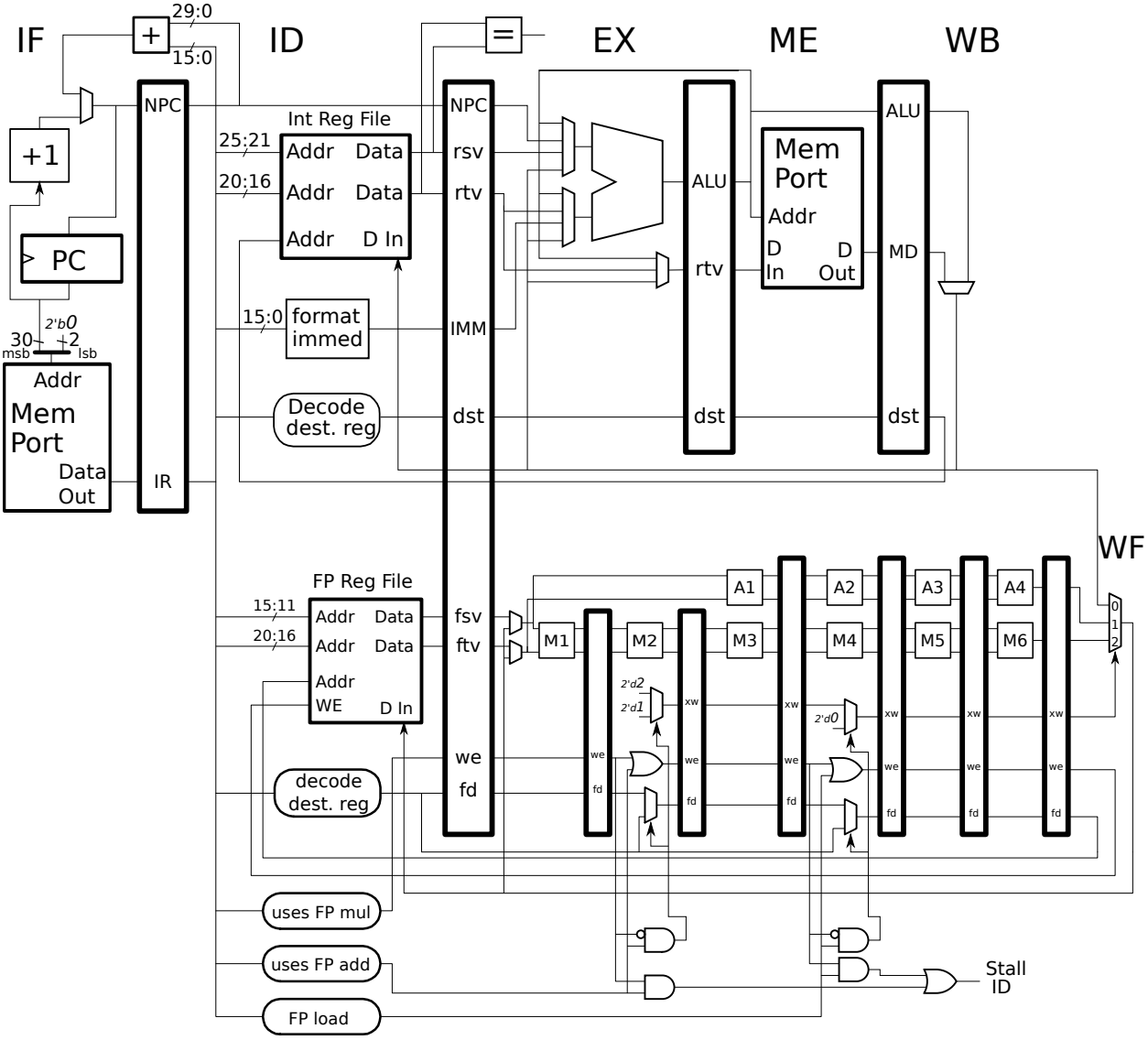
LSU EE 4720 **Homework 6** Solution **Due: 5 April 2017**

Attention Perfectionists: An Inkscape SVG version of the illustration used in the final exam and this assignment can be found at: `https://www.ece.lsu.edu/ee4720/2017/mpipei_fp.svg`.

Problem 1: Answer Spring 2016 Final Exam Problem 2b and 2c, which ask about the execution of FP MIPS code. The solution to these problems are available. **Make a decent attempt to solve these problems on your own, without looking at the solution.** Only peek at the solution for hints and use the solution to check your work.

Problem 2: Appearing below are two MIPS code fragments and the MIPS implementation from the final exam. The fragments execute on the illustrated implementation with the addition of the datapath needed for the store instructions that was provided in Final Exam Problem 2c. The fragments are labeled `Degree 1` and `Degree 2`, these refer to an optimization technique called *loop unrolling*, which has been applied to the `Degree 2` loop.



(*a*) Show a pipeline execution diagram of each on the illustrated code fragments. Show enough iterations to compute the CPI. Note that the second loop should have fewer stalls than the first.

Solution appears below. Note that the state of the pipeline at the start of the second iteration, in cycle 9, is the same as the state of the pipeline at the start of the third iteration, in cycle 18. In both cases the **addi** is in **ID**, the **bne** is in **EX**, and the **swc1** is in **ME**. Therefore we can use the second iteration to compute CPI. (It just so happens that we could also the first iteration, but we didn't prove that we could with the argument above.)

```
#  Degree 1 -- SOLUTION
LOOP:  #  First Iteration
       #  Cycle    0  1  2  3  4  5  6  7  8  9  10 11 12 13 14 15 16 17 18 29 20
 lwc1 f0, 0(r1)    IF ID EX ME WF
 add.s f0, f0, f1     IF ID -> A1 A2 A3 A4 WF
 swc1 f0, 0(r1)          IF -> ID -------> EX ME WB
 bne  r1, r3 LOOP              IF -------> ID EX ME WB
 addi r1, r1, 4                            IF ID EX ME WB
LOOP:  #  Second Iteration
       #  Cycle    0  1  2  3  4  5  6  7  8  9  10 11 12 13 14 15 16 17 18 29 20
 lwc1 f0, 0(r1)                               IF ID EX ME WF
 add.s f0, f0, f1                                IF ID -> A1 A2 A3 A4 WF
 swc1 f0, 0(r1)                                     IF -> ID -------> EX ME WB
 bne  r1, r3 LOOP                                         IF -------> ID EX ME
 addi r1, r1, 4                                                       IF ID EX ME WB
       #  Cycle    0  1  2  3  4  5  6  7  8  9  10 11 12 13 14 15 16 17 18 29 20 21
LOOP:  #  Third Iteration
 lwc1 f0, 0(r1)                                                          IF ID EX ME..

#  Degree 2 -- SOLUTION
LOOP:   #  First Iteration
        #  Cycle    0  1  2  3  4  5  6  7  8  9  10 11 12 13 14 15 16 17 18 29 20
 lwc1 f0, 0(r1)     IF ID EX ME WF
 lwc1 f1, 4(r1)        IF ID EX ME WF
 add.s f0, f0, f9         IF ID A1 A2 A3 A4 WF
 add.s f1, f1, f9            IF ID A1 A2 A3 A4 WF
 swc1 f0, 0(r1)                 IF ID ----> EX ME WB
 swc1 f1, 4(r1)                    IF ----> ID EX ME WB
 bne  r1, r3 LOOP                           IF ID EX ME WB
 addi r1, r1, 8                                IF ID EX ME WB
        #  Cycle    0  1  2  3  4  5  6  7  8  9  10 11 12 13 14 15 16 17 18 29 20
LOOP:   #  Second Iteration
 lwc1 f0, 0(r1)                                    IF ID EX ME WF
 lwc1 f1, 4(r1)                                       IF ID EX ME WF
 add.s f0, f0, f9                                        IF ID A1 A2 A3 A4 WF
```

(*b*) Compute the execution efficiency of both loops in CPI. Remember that the number of cycles should be determined by looking at the same point in execution, usually IF of the first instruction, in two different iterations. Put another way, just because the custom car you will order after graduation will take two months to arrive, doesn't mean that the factory makes just one car every two months.

The Degree 1 loop has a CPI of $\frac{18-9}{5} = 1.8\,\text{CPI}$ based on the iteration start times of cycle 9 and cycle 18. The Degree 2 loop has a CPI of $\frac{10-0}{8} = 1.25\,\text{CPI}$, under the assumption that all iterations will execute in the same way as the first.

(*c*) Assume that both loops operate on N-element arrays (and that N is even). The Degree-1 loop operates on just one element per iteration, while the Degree-2 loop operates on two elements per iteration.

Devise a performance measure that can be used to compare the two loops based on the work that they do. The improvement of Degree-2 or Degree-1 should be higher with this work-based performance measure than the improvement computed using CPI.

A reasonable measure would be cycles per element or CPE. The Degree 1 loop computes one element of the array per iteration, and so it computes $\frac{18-9}{1} = 9$ CPE. The Degree 2 loop computes two elements per iteration, and so it computes at $\frac{10-0}{2} = 5$ CPE, almost twice as fast.

Based on CPI the Degree 2 loop is $\frac{1.8}{1.25} = 1.44$ times better than the Degree 1 loop. (Remember that lower CPI is better.) But based on CPE the Degree 2 loop is $\frac{9}{5} = 1.8$ times better.

Note that CPI is appropriate for comparing two implementations that run the same program, but it's not useful for comparing two different programs.

(*d*) Besides eliminating stalls, what makes `Degree 2` faster than `Degree 1` even when doing the same amount of work?

There are fewer instructions per element. For example, an `addi` is executed for each array element in Degree 1, but it is executed for every two array elements in Degree 2.