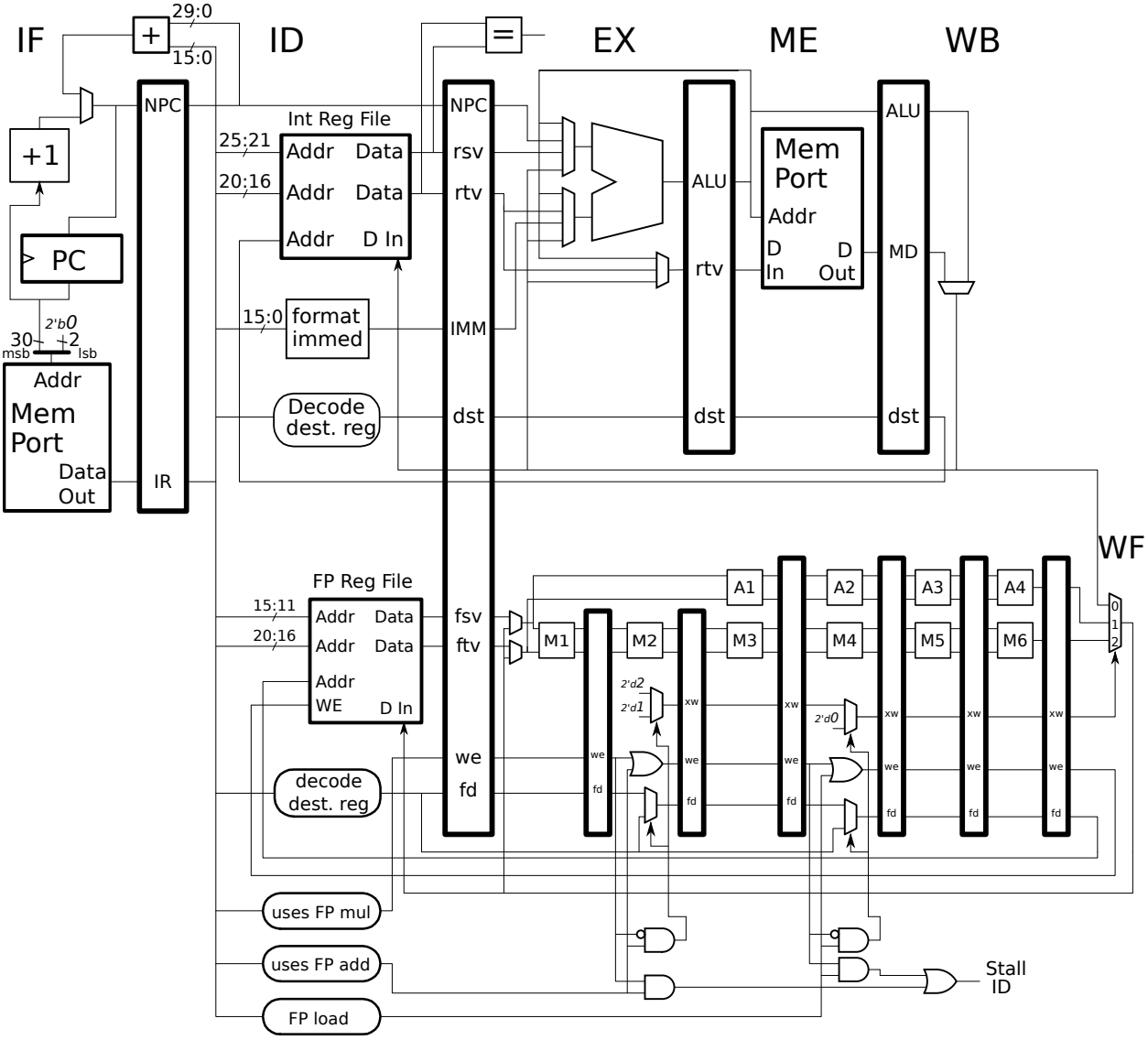
LSU EE 4720 **Homework 6** **Due: 5 April 2017**

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.



```
LOOP:  # Degree 1
 lwc1 f0, 0(r1)
 add.s f0, f0, f1
 swc1 f0, 0(r1)
 bne r1, r3 LOOP
 addi r1, r1, 4
```

```
LOOP:  # Degree 2
 lwc1 f0, 0(r1)
 lwc1 f1, 4(r1)
 add.s f0, f0, f9
 add.s f1, f1, f9
 swc1 f0, 0(r1)
 swc1 f1, 4(r1)
 bne r1, r3 LOOP
 addi r1, r1, 8
```

(*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.

(*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.

(*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.

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