Homework 5

Due Date: May 2

All written homeworks are due at the beginning of class on the due date. There is no provision for homework turned in late. Partial credit will be given for work shown. Points are scaled relative to the "time" value given in the book (the number in square brackets next to the problem). If your homework is longer than one page, please staple or paper-clip the pages together. Be as neat as possible; if I can't read it, it's wrong.

It is not guaranteed that all problems will be graded. We will either cover answers in class and/or a solution sheet will be posted.

At the end of your homework, please write and sign the Honor Code pledge:

I have abided by the Wheaton College Honor Code in this work.

Do text problems:

4.16.1-4.16.3, 4.20*, 4.25.1, 4.25.2, 4.27.2 (see below), 5.5.1, 5.5.2, 5.5.4, 5.5.5

For 4.27.2: Change and/or rearrange the code **and** assume you have forwarding. By writing the five-stage pipeline, show how you can (or perhaps not) reduce the number of NOPs.

For 5.5.4, set up a table like this:

Ref.	Tag	Index	Offset	Hit/Miss	Replace
0	0-00	00000	00000	M	–
4	0-00	00000	00100	H	No
etc.					

In addition:

1. [5] Redo problem 4.20, but this time assume you have a datapath with a five-stage pipeline **with** forwarding.

2. [10] Consider the following sequence of instructions:

   ```
   and r1, r2, r3     # instruction 1
   and r2, r1, r4     # instruction 2
   and r1, r1, r2     # instruction 3
   ```

   Indicate the dependencies and their type (RAW, WAR, etc.) in a table as shown in the following example:

Dependency	Register	Instructions
RAW	r8	4 & 5
RAW	r8	4 & 6

3. [10] For each of the following references, identify the tag and index given a direct-mapped cache with 16 one-word blocks. Also list if each reference is a hit or a miss, assuming the cache is initially empty. Assume an 8 bit address (we are disregarding the byte portion).

   3, 180, 43, 3, 191, 35, 3, 3, 35, 190, 191, 174, 190, 35, 35, 190, 191, 3, 180

4. Good study questions (for final exam) but **not** to hand in:

   5.16.1–5.16.3

* `NOP` means "no operation;" that is, an "empty" instruction.