# MC48 Homework 2

Max Hailperin

Due: September 20, 1995

Remember that your answers to these exercises should not be straight assembly language code or numbers or formulas but rather English text that includes code or numbers and formulas where appropriate. See the answers to the examples in chapters 2 and 3 for illustrations of this.

1. Do exercise 3.11 from page 157. This exercise tells you to use at most one branch or jump each time through the loop; as a clarification, this means that you are allowed to use as many branches or jumps as you like, so long as the number of times all but one of them are executed doesn't scale with the number of loop iterations. That is, if you write the number of branches or jumps executed as a linear function of the number of loop iterations, $n$, in the form $An + B$, then the coefficient $A$ must be 1, but the constant term $B$ can be anything you like.

2. Some computer architectures otherwise similar to the MIPS have auto-incrementing and auto-decrementing loads and stores, more or less as described on pages 133–134. In this exercise, we consider what the performance impact of this feature might be.

   The table on page 150 shows the percentage of the instructions executed by the `gcc` program that fall into each category. Some unknown fraction of the data transfer instructions could be changed into the auto-incrementing or auto-decrementing version. For each instruction changed in this way, one arithmetic instruction would be eliminated.

   To find out the impact of this elimination of arithmetic instructions on the number of clock cycles the program takes to run, we need to known the CPI for each class of instructions. Exercise 3.22 on page 158 has a table you can use for this purpose.

   We also need to know how much the clock cycle time would need to change by; assume that if me modified the MIPS architecture to accommodate auto-increment and auto-decrement instructions, the cycle time would need to be increased by 10%.

   (a) What if 50% of the data transfer instructions can be changed; which machine (MIPS or MIPS modified to have auto-increment and auto-decrement) would be faster for `gcc`? How much faster?

   (b) What if only 25% of the data transfer instructions can be changed; which machine is faster? How much?