CS M151B Homework 2

Charles Zhang

January 11, 2022

Problem 2.16

Assume that we would like to expand the MIPS register file to 128 registers and expand the instruction set to contain four times as many instructions.

a) How would this affect the size of each of the bit fields in the R-type instructions?

In order to adjust for the fact that there are four times as many instructions, we must allow for the instruction to store four times as many opcodes. As a result the `op` field must increase from 6 bits to 8 bits. In addition, the increase to 128 registers means we must now be able to identify 128 unique registers in the `rs`, `rt`, and `rd` fields, which requires 7 bits each. This requires 29 bits. Since we cannot reduce the size of the `func` field, as we do not want to reduce the number of R-type instructions, and we should reserve at least 1 bit for the `shamt` field to allow shifting, this kind of change would require 36-bit instructions.

b) How would this affect the size of each of the bit fields in the I-type instructions?

For the same reasons as in part a) above, the `op` field must increase from 6 bits to 8 bits and the `rs` and `rt` fields much increase from 5 bits to 7 bits each. Since MIPS has fixed-length instructions, we would then expand the immediate to have 14 bits rather than 10 to make I-type instructions 36 bits long.

c) How could each of two propose changes decrease the size of an MIPS assembly program? On the other hand, how could the proposed change increase the size of an MIPS assembly program?

Expanding the register file may reduce register spilling, resulting in fewer loads and stores. Expanding the instruction set may help to condense multiple basic instructions into a single, complex instruction, reducing the instruction count of the program.

The size of the program may increase because all instructions are now required to be an extra 4 bits long, increasing the amount of memory they each require.

Problem 2.39

Assume for a given processor the CPI of arithmetic instructions is 1, the CPI of load/store instructions is 10, and the CPI of branch instructions is 3. Assume a program has the following instruction breakdowns: 500 million arithmetic instructions, 300 million load/store instructions, 100 million branch instructions.

a) Suppose that new, more powerful arithmetic instructions are added to the instruction set. On average, through the use of these more powerful arithmetic instructions, we can reduce the number of arithmetic instructions needed to execute a program by 25%, and the cost of increasing the clock cycle time by only 10%. Is this a good design choice? Why?

$$\text{Execution Time} = \text{Instruction Count} + \text{CPI} + \text{Cycle Time}$$
$$\text{ET}_i = 900,000,000 \times (\frac{5}{9} \times 1 + \frac{3}{9} \times 10 + \frac{1}{9} \times 3) \times \text{CT}_i$$
$$\text{ET}_i = 3,600,000,000 \times \text{CT}_i$$

$$\text{\# of arithmetic instructions} = 0.75 \times 500,000,000 = 375,000,000$$
$$\text{IC}_f = 775,000,000$$
$$\text{CT}_f = 1.1 \times \text{CT}_i$$
$$\text{ET}_f = 775,000,000 \times (\frac{375}{775} \times 1 + \frac{300}{775} \times 10 + \frac{100}{775} \times 3) \times (\text{CT}_i \times 1.1)$$
$$\text{ET}_f = 4,042,500,000 \times \text{CT}_i$$

This would be a bad design choice because the increased clock time along with the increased CPI would increase the execution time, offsetting any benefits obtained from lowering the instruction count.

b) Suppose that we find a way to double the performance of arithmetic instructions. What is the overall speedup of our machine? What if we find a way to improve the performance of arithmetic instructions by 10 times?

Neither of these improvements will speed up or slow down the machine at all. This is because arithmetic operations already take a single cycle to execute, and therefore, we will gain no benefits to instruction count or CPI. In addition, since we have no knowledge about the other instructions, we can't safely reduce the cycle time, so execution time would remain the same.

Problem 2.40

Assume that for a given program 70% of the executed instructions are arithmetic, 10% are load/store, and 20% are branch.

a) Given the instruction mix and the assumption that an arithmetic instruction requires 2 cycles, a load/store instruction takes 6 cycles, and a branch instruction takes 3 cycles, find the average CPI.

$$\text{Avg. CPI} = (0.7 \times 2) + (0.1 \times 6) + (0.2 \times 3)$$
$$\boxed{\text{Avg. CPI} = 2.6}$$

b) For a 25% improvement in performance, how many cycles, on average, may an arithmetic instruction take if load/store and branch instructions are not improved at all?

$$\frac{\text{ET}_i}{\text{ET}_f} = 1.25$$
$$\frac{\text{IC} \times \text{CPI}_i \times \text{CT}}{\text{IC} \times \text{CPI}_f \times \text{CT}} = 1.25$$
$$\frac{\text{CPI}_i}{\text{CPI}_f} = 1.25$$
$$\frac{2.6}{\text{CPI}_f} = 1.25$$
$$\text{CPI}_f = 2.08$$
$$(0.7 \times C) + (0.1 \times 6) + (0.2 \times 3) = 2.08$$
$$\boxed{C = 1.26 \text{ cycles}}$$

c) For a 50% improvement in performance, how many cycles, on average, may an arithmetic instruction take if load/store and branch instructions are not improved at all?

$$\frac{2.6}{\text{CPI}_f} = 1.5$$
$$\text{CPI}_f = 1.73$$
$$(0.7 \times C) + (0.1 \times 6) + (0.2 \times 3) = 1.73$$
$$\boxed{C = 0.76 \text{ cycles, this is not possible}}$$