CSE141 Problem Set #2 Solutions

January 26, 2002

1 Itanium vs. Pentium

To calculate how much faster the Itanium is than the Pentium, one can use the formula at the bottom of page 62:

$$\frac{\text{CPU performance}_\text{I}}{\text{CPU performance}_\text{P}} = \frac{\text{Execution time}_\text{P}}{\text{Execution time}_\text{I}} = \frac{25\text{ sec}}{15\text{ sec}} = 1.67$$

Therefore, the Itanium machine is 1.67 times faster than the Pentium machine. In order to calculate "percent faster", we use the following formula from Larry's notes:

$$\begin{aligned}
\text{speed(I)} = (1 + \frac{\text{x}}{100})\text{speed(P)} \quad &\Rightarrow \quad \frac{\text{speed(I)}}{\text{speed(P)}} = (1 + \frac{\text{x}}{100}) \\
&\Rightarrow \quad \frac{\text{Execution time}_\text{P}}{\text{Execution time}_\text{I}} = (1 + \frac{\text{x}}{100}) \\
&\Rightarrow \quad 1.67 = (1 + \frac{\text{x}}{100}) \\
&\Rightarrow \quad .67 = (\frac{\text{x}}{100}) \\
&\Rightarrow \quad \text{x} = 67
\end{aligned}$$

Therefore, the Itanium machine is 67% faster than the Pentium machine.

2 Comparing Growth Rates

The easiest way to solve this problem is to find the earliest common year during which both growth rates provide a data point. Since DRAM data points occur every three years, and workstation performance data points occur every 10 years, the earliest common year that lies on both paths is 30. Given this, we need only calculate how much both attributes will grow during a 30 year period to determine which has the faster growth rate.

If the initial DRAM capacity is given by x, then after three years it will be $4x$. After 6 years it will be $16x$. After 9 years, $64x$, and so on. Extrapolation of this pattern could be simplified by looking at the growth as follows:

3 years $\rightarrow 4^1x$
6 years $\rightarrow 4^2x$
9 years $\rightarrow 4^3x$
.
.
.
30 years $\rightarrow 4^{10}x$

Therefore, after 30 years, DRAM capacity will be 4^{10} times greater. Now let us take a look at workstation performance. Since we are given the statistic that over a 10 year period performance has grown from 9 to 1140, we can say that every 10 years, performance increases by a factor of $(1140/9) = 126.7$. So, given this grown rate, if the original workstation performance is given by x, after 10 years performance will be $126.7x$. After 20 years, 126.7^2x, and after 30 years, 126.7^3x.

We now need to get the two values we have obtained into similar forms for comparison. In the tradition of Larry's "Back of the Envelope Estimates", one can make the observation that 126.7 is awfully close to 128, or 2^7. That being the case, instead of having to deal with 126.7^3 as one of our terms, we can now estimate with $(2^7)^3$ or $\mathbf{2^{21}}$. When compared with the DRAM capacity figure, 4^{10} or $\mathbf{2^{20}}$, it is clear that the growth rate of the workstation performance is indeed higher that that of the DRAM capacity.

Once could also have solved this problem by calculating an average yearly growth rate for the two statistics and performing a direct comparison. To say that DRAM capacity grows by a factor of 4 every 3 years is the same as saying there exists some factor f such that if x is the initial DRAM capacity:

$$x * f * f * f = 4x$$

$$f^3 = 4$$

$$f = 1.59$$

Similarly, for workstation performance, $f^{10} = 126.7$ and therefore, $f = 1.623$. Thus, we arrive at the same conclusion that the growth rate of workstation performance is greater than that of DRAM capacity.

3 MIPS with update addressing (exercise 3.13)

Let us begin the solution to this exercise by calculating the CPI for the original, unmodified architecture. Taking into account the percentages from page 189 and assuming a program of 100 instructions:

$$\frac{(1.0\frac{\text{cycles}}{\text{inst}} * 48\frac{\text{inst}}{\text{prog}}) + (1.4\frac{\text{cycles}}{\text{inst}} * 33\frac{\text{inst}}{\text{prog}}) + (1.7\frac{\text{cycles}}{\text{inst}} * 17\frac{\text{inst}}{\text{prog}}) + (1.2\frac{\text{cycles}}{\text{inst}} * 2\frac{\text{inst}}{\text{prog}})}{100\frac{\text{inst}}{\text{prog}}}$$

$$\Rightarrow \quad \frac{(48\frac{\text{cycles}}{\text{prog}}) + (46.2\frac{\text{cycles}}{\text{prog}}) + (28.9\frac{\text{cycles}}{\text{prog}}) + (2.4\frac{\text{cycles}}{\text{prog}})}{100\frac{\text{inst}}{\text{prog}}}$$

$$\Rightarrow \quad \frac{125.5\frac{\text{cycles}}{\text{prog}}}{100\frac{\text{inst}}{\text{prog}}} = \mathbf{1.255\frac{cycles}{inst}}$$

Therefore, the CPU Execution time for the original architecture can be given by:

$$\mathbf{1.255 * IC_{orig} * Clock_{orig}}$$

Now with our modified architecture, we know that 25% of data transfer instructions are affected by the change, and that for each of these intsructions, one arithmetic operation can be eliminated. Data transfer instructions make up 33% of the total instruction mix, according to p.189, and therefore, $.25 * 33\% = 8.25\%$ of all instructions are affected by the change. If we again assume a 100 instruction program, this means we can eliminate 8.25 arithmetic instructions. Let us calculate our new CPI. Keep in mind that since 8.25 arithmetic instructions have been removed from the instruction mix, this number also needs to be subtracted from the total number of instructions we are considering (see denominator):

$$\frac{(1.0\frac{\text{cycles}}{\text{inst}} * (48 - 8.25)\frac{\text{inst}}{\text{prog}}) + (1.4\frac{\text{cycles}}{\text{inst}} * 33\frac{\text{inst}}{\text{prog}}) + (1.7\frac{\text{cycles}}{\text{inst}} * 17\frac{\text{inst}}{\text{prog}}) + (1.2\frac{\text{cycles}}{\text{inst}} * 2\frac{\text{inst}}{\text{prog}})}{(100 - 8.25)\frac{\text{inst}}{\text{prog}}}$$

$$\Rightarrow \quad \frac{(39.75\frac{\text{cycles}}{\text{prog}}) + (46.2\frac{\text{cycles}}{\text{prog}}) + (28.9\frac{\text{cycles}}{\text{prog}}) + (2.4\frac{\text{cycles}}{\text{prog}})}{91.75\frac{\text{inst}}{\text{prog}}}$$

$$\Rightarrow \quad \frac{117.25\frac{\text{cycles}}{\text{prog}}}{91.75\frac{\text{inst}}{\text{prog}}} = \mathbf{1.278\frac{cycles}{inst}}$$

Now we have everything we need to make a comparison. We know that the new instruction count is 91.75% of the original, the new clock rate is 10% greater or 1.1 times more than the original, and we know that our modified CPI is 1.278. Therefore, CPU Execution time for the modified architecture can be given by:

$$1.278 * (.9175)\mathrm{IC_{orig}} * (1.1)\mathrm{Clock_{orig}} = \mathbf{1.290 * IC_{orig} * Clock_{orig}}$$

If we compare this to our original architecture which has a CPU time of $\mathbf{1.255 * IC_{orig} * Clock_{orig}}$, it is clear that our original architecture is actually the faster one. How much faster?

$$\frac{1.290 * \mathrm{IC_{orig}} * \mathrm{Clock_{orig}}}{1.255 * \mathrm{IC_{orig}} * \mathrm{Clock_{orig}}} = \mathbf{1.028}$$

4 CPI for Gcc and Spice (exercise 3.16)

When we average the percentages given on p.189, we end up with the following instruction mix:

Arithmetic	49%
Data Transfer	37%
Conditional Branch	12.5%
Jump	1.5%

And now we simply calculate CPI as we have done before:

$$\frac{(1.0\frac{\mathrm{cycles}}{\mathrm{inst}} * 49\frac{\mathrm{inst}}{\mathrm{prog}}) + (1.4\frac{\mathrm{cycles}}{\mathrm{inst}} * 37\frac{\mathrm{inst}}{\mathrm{prog}}) + (1.7\frac{\mathrm{cycles}}{\mathrm{inst}} * 12.5\frac{\mathrm{inst}}{\mathrm{prog}}) + (1.2\frac{\mathrm{cycles}}{\mathrm{inst}} * 1.5\frac{\mathrm{inst}}{\mathrm{prog}})}{100\frac{\mathrm{inst}}{\mathrm{prog}}}$$

$$\Rightarrow \quad \frac{(49\frac{\mathrm{cycles}}{\mathrm{prog}}) + (51.8\frac{\mathrm{cycles}}{\mathrm{prog}}) + (21.25\frac{\mathrm{cycles}}{\mathrm{prog}}) + (1.8\frac{\mathrm{cycles}}{\mathrm{prog}})}{100\frac{\mathrm{inst}}{\mathrm{prog}}}$$

$$\Rightarrow \quad \frac{123.85\frac{\mathrm{cycles}}{\mathrm{prog}}}{100\frac{\mathrm{inst}}{\mathrm{prog}}} = \mathbf{1.239\frac{cycles}{inst}}$$

5 Load Elimination

The first thing to do in this exercise is determine how much the instruction count needs to be reduced in order to compensate for the 10% increase in clock rate. Note that we have the luxury of assuming that CPI is unaffected, unlike in question 3.13. Our initial problem essentially boils down to the following equation, where p is the desired percentage of the original instruction count.

$$\frac{\mathrm{p}}{100}\mathrm{IC_{orig}} * \mathrm{CPI} * (1.1)\mathrm{Clock_{orig}} = \mathrm{IC_{orig}} * \mathrm{CPI} * \mathrm{Clock_{orig}}$$

Since everything cancels out, we are left with:

$$\frac{p}{100} = \frac{1}{1.1}$$

$$\Rightarrow \quad = \mathbf{90.9}$$

Therefore, we want our new instruction count to be 90.9% of the original. In other words, we wish to eliminate 9.1% of our instructions. Since loads are what we are targeting, we should note that two-thirds of data transfer instructions, which make up 33% of the overall instruction mix are loads. It then follows that loads make up 22% of the instruction mix, since two-thirds of 33 is 22. Finally, if we are to remove all 9.1% of instructions from the loads, that would mean removing $(9.1/22) * 100$ or approximately **41.4%** of all loads.