Homework 1 (100 points total)
EE282 Autumn 2001/2002
Solutions

Date due: Tuesday, October 9, 2001, 5:00pm

October 18, 2001

1 Making Games Fast (60 points total)

Assume that the total CPU time for processing all the tasks = 100s

Original Machine This implies 10 seconds were spent in TGAL, 40s in TT, 35s in PM, 10s in GP and 5s in others. Number of frames executed in this time = 15 * 100 = 1500.

1.1 Compiler Optimization (15 points)

Total CPU time	(60s * 80%) + (40s * 100%) = 88s
Number of frames executed in this time	1500

Therefore new frame rate = 1500frames/88s = 17.045 frames/s

1.2 Hardware Acceleration (15 points)

a. Mid-range Accelerator

Total CPU time	$(\frac{40s}{2}) + (60s) = 80s$

Therefore new frame rate = 1500frames/80s = 18.75 frames/s

b. I3 Accelerator

Total CPU time	$(\frac{40s}{5}) + (\frac{10s}{2} + 50s) = 63s$

Therefore new frame rate = 1500frames/63s = 23.81 frames/s

1.3 Hand code optimization (15 points)

To get 18 frames/sec, we will need

$$\frac{TotalFrames}{TotalTime} = frames/sec \tag{1}$$

$$\frac{TotalFrames}{frames/sec} = TotalTime \tag{2}$$

$$\frac{1500}{18} = 83.33333...s \tag{3}$$

$$83.33333...s = 10s + 40s + \frac{35s}{x} + 10s + 5s = 65s + \frac{35s}{x} \tag{4}$$

...where x is the speedup factor for Physical Modeling.

Solving for x,

$$18.33333...s = \frac{35s}{x} \tag{5}$$

$$x = \frac{35s}{18.333..s} = 1.9090... \tag{6}$$

This means that Physical Modeling must be done at 1.91x the original speed. In other words, Physical Modeling must have its execution time reduced by $(35 - 18.3333333)/35 = 47.619...\%$, or its new execution time must be 52.38% of its original. Credit should be given for answers stated in any of these three ways.

1.4 Putting things together (15 points)

a. Mid-range Accelerator

Again, let's calculate the new time as we did above:

$$\frac{1500}{40} = 37.5s \tag{7}$$

$$37.5s = (10s * 80\%) + \frac{40s}{2} + (\frac{35s}{x} * 80\%) + (10s * 80\%) + (5s * 80\%) \tag{8}$$

$$37.5s = 40s + (\frac{35s}{x} * 80\%)) \tag{9}$$

As you can see, it's theoretically impossible to reach 40 frames/sec via a combination of the mid-range accelerator, the compiler optimization, and any optimization of Physical Modeling.

b. I3 Accelerator

From the previous calculation, the new time for 40 frames/sec must be 37.5s . Thus:

$$37.5s = (\frac{10s}{2} * 80\%) + \frac{40s}{5} + (\frac{35s}{x} * 80\%) + (10s * 80\%) + (5s * 80\%) \tag{10}$$

$$37.5s = 24s + (\frac{35s}{x} * 80\%) \tag{11}$$

Thankfully, it's possible in this case. Solving for x:

$$13.5s = \frac{35s}{x} * 80\% \tag{12}$$

$$x = \frac{35s * 80\%}{13.5s} = 2.074074... \tag{13}$$

Again, the answer can come in the other forms. Note that we are giving speedup values for Physical Modeling separate from the contribution from compiler optimization. That is, the absolute speedup for Physical Modeling is a combination of the speedup values we provide here and the compiler optimization effects. Full credit should also be given for answers given with respect to the total speedup (hand-optimization and compiler optimization) applied to Physical Modeling.

$$\frac{13.5s}{80\%} = \frac{35s}{x} \tag{14}$$

$$16.875s = \frac{35s}{x} \tag{15}$$

So to summarize, Physical Modeling must be done at 2.07x the original speed. In other words, Physical Modeling must have its execution time reduced by $(35 - 16.875)/35 = 51.78...\%$, or its new execution time must be 48.21% of its original. Again, credit should be given for answers stated in any of these three ways.

2 DLX performance benchmarking (40 points total)

2.1 Software floating point performance (15 points)

To calculate this, we first figure out how much time the DLX machine spends on instructions not executed on behalf of emulating floating point.

$$(875,000 * 1.3) + (580,000 * 1.3) + (1,075,000 * 1.3) = 3,289,000 cycles \tag{16}$$

$$3,289,000 cycles * time/cycle = 3,289,000 * \frac{1}{25 * 10^6} = 0.13156s \tag{17}$$

Thus the DLX machine spends 5-0.13156 = 4.86844s emulating floating point instructions.

2.2 Floating point emulation (10 points)

We can find the number of instructions spent emulating by:

$$(\#instructions) * CPI * (time/cycle) = 4.86844s \tag{18}$$

Assuming that CPI for emulating instructions is 1.3, then:

$$\#instructions = \frac{4.86844s}{CPI * (time/cycle)} = \frac{4.86844s}{1.3 * \frac{1}{25*10^6}} = 93.62...millioninstructions \tag{19}$$

Dividing by 1,955,000 yields 47.889 integer instructions to emulate a floating-point instruction on average.

2.3 Cost versus performance (15 points)

The problem states that the cost/performance must be the same. Thus, since DLX-FP costs 10x as much, its performance must also be 10x, that is, the execution time must be 1/10th of DLX. In other words, the benchmark that took DLX 5 seconds to execute should execute in 0.5 seconds on DLX-FP.

Writing this out, we have

$$0.5 = (1.3 * 0.875 + 1.3 * 0.58 + 1.6 * 1.075 + x * 1.955) * 10^6 * \frac{1}{50 * 10^6} \tag{20}$$

... where x is the CPI for floating point instructions on DLX-FP.

$$25 = (3.6115 + x * 1.955) \tag{21}$$

$$x = \frac{21.3885}{1.955} = 10.94... \tag{22}$$