Problem Set 1

Justin Ely

605.411 Foundations of Computer Architecture

08 September, 2016

1a)

The clock rate is the inverse of the clock cycle. Thus, the clock rate for this implementation is $\frac{1}{t_{cycle}} = \frac{1}{250e-12} = 4GHz$.

1b)

The total number of clock cycles consumed is calculated by the total number of instructions multipled by the weighted sum of the cycles per instruction.

$N_{cycles} = 20e6(.1 \times 12 + .9 \times 5) = 1.14E8$

1c)

Making the divide instruction twice as fast means cutting the number of cycles in half from 12 to 6.

$N_{cycles\_new} = 20e6(.1 \times 6 + .9 \times 5) = 1.02E8$

Thus, the speedup is the ratio between the new and old number of cycles required.

$speedup = \frac{N_{cycles}}{N_{cycles\_new}} = 1.12$ or 12%.

2a)

$CPI = time \times \frac{rate}{n_{inst}} = 1.1 \times \frac{1e-9}{1e-9} = 1.1$

2b)

$T2 = 1.1 \times \frac{6e8}{\frac{1}{1e-9}} = .66$

Thus the speedup is: $\frac{T1}{T2} = \frac{1.1}{.66} = 1.67$ or 67%.

3a)

Ignoring constant terms, which cancel, and taking the fraction of multiply instructions and other instructions to be x and y respectively,

$$\begin{aligned} t_1 &= 4x + 4y && (1) \\ 2.85 &= 4x + 4y && (2) \end{aligned}$$

$$\begin{aligned} t_2 &= 3x + 4y && (3) \\ 2.28 &= 3x + 4y && (4) \end{aligned}$$

Re-ordering (3) and (4), and substituting variables gives us:

$$\begin{aligned} 2.85 - 4x &= 2.28 - 3x && (5) \\ 2.85 &= 2.28 + x && (6) \\ .57 &= x && (7) \end{aligned}$$

The multiply instruction accounts for 57% of the executed instructions.

3b)

Since speedup is $\frac{t1}{t2}$, constant-terms can again be ignored.

$$\begin{aligned} t_1 &= (12 \times .22 + 4 \times .78) && (8) \\ t_2 &= (3 \times .22 + 4 \times .78) && (9) \end{aligned}$$

Combining (8) and (9) gives $\frac{t_1}{t_2} = 1.52$, which is the speedup factor.

4a)

The total clock cycles for each mode will be the weighted sum of the CPI * instruction count.

- P1: $1E6(1 \times .1 + 2 \times .2 + 2 \times .5 + 3 \times .2) = 2.1e6$ cycles.
- P1: $1E6(2 \times .1 + 2 \times .2 + 2 \times .5 + 2 \times .2) = 2.0e6$ cycles.

4b)

The total running time will be the number of cycles times the clock cycle time ($\frac{1}{rate}$).

- P1: $2.1e6 \times \frac{1}{2.5e9} = .00084$ seconds.
- P2: $2.0e6 \times \frac{1}{3e9} = .00067$ seconds.

4c)

The CPI for each implementation is the weighted sum of the CPIs of the individual instruction classes.

- P1: $1 \times .1 + 2 \times .2 + 2 \times .5 + 3 \times .2 = 2.1$ CPI.
- P1: $2 \times .1 + 2 \times .2 + 2 \times .5 + 2 \times .2 = 2.0$ CPI.

5a)

The initial running time is 250s, which consists of 70s on FP, 85s on L/S, 40s on branching, and 55s on other instructions.

If FP instructions are reduced by 20%, the FP instructions will then take $70 * .8 = 56$. Thus, the new total time will be $56 + 85 + 40 + 55 = 236$, which is a reduction of 14 seconds from the original.

5b)

To reduce the total time by 20%, we would need to cut off $250 \times .2 = 50$ seconds. Since branching instructions currently only take 40 seconds, even if they took 0 seconds the goal time could not be reached.