**The University of Alabama in Huntsville**
**ECE Department**
**Homework #1 Solution**
**CPE 431/531 01/91/92**
**Fall 2015**

**1.5** Consider three different processors P1, P2, and P3 executing the same instruction set. P1 has a 3 GHz clock rate and a CPI of 1.5. P2 has a 2.5 GHz clock rate and a CPI of 1.0. P3 has a 4.0 GHz clock rate and has a CPI of 2.2.

a. Which processor has the highest performance expressed in instructions per second?

$$P_1 = \frac{3.0E9 cycles/s}{1.5 cycles/instruction} = 2.0E9 instructions/s$$

$$P_2 = \frac{2.5E9 cycles/s}{1.0 cycles/instruction} = 2.5E9 instructions/s$$

$$P_3 = \frac{4.0E9 cycles/s}{2.2 cycles/instruction} = 1.81E9 instructions/s$$

**$P_3$ has the highest performance expressed in instructions per second.**

b. If the processors each execute a program in 10 seconds, find the number of cycles and the number of instructions.

**ET = (IC x CPI)/CR**
**IC = (ET x CR)/CPI**
**$IC_{P1}$ = (10s x 3.0E9 cycles/s)/1.5 cycles/instruction = 20E9 instructions**
**$IC_{P2}$ = (10s x 2.5E9 cycles/s)/1.0 cycles/instruction = 25E9 instructions**
**$IC_{P3}$ = (10s x 4.0E9 cycles/s)/2.2 cycles/instruction = 18.2E9 instructions**
**$CC_{P1}$ = ET x CR = 10s x 3.0E9cycles/c = 30E9 cycles**
**$CC_{P2}$ = ET x CR = 10s x 2.5E9cycles/c = 25E9 cycles**
**$CC_{P3}$ = ET x CR = 10s x 4.0E9cycles/c = 40E9 cycles**

c. We are trying to reduce the execution time by 30% but this leads to an increase of 20% in the CPI. What clock rate should we have to get this time reduction?

**ET = 7 s, $CPI_{P1}$ = 1.2*1.5 = 1.8, $CPI_{P2}$ = 1.2*1.0 = 1.2, $CPI_{P3}$ = 1.2*2.2 = 2.64**
**ET = (IC x CPI)/CR**
**CR = (IC x CPI)/ET**
**$CR_{P1}$ = (20E9 instructions x 1.8 cycles/instruction)/7 s = 5.14E9 cycles/s**
**$CR_{P2}$ = (25E9 instructions x 1.2 cycles/instruction)/7 s = 4.28E9 cycles/s**
**$CR_{P3}$ = (18.2E9 instructions x 2.64 cycles/instruction)/7 s = 6.86E9 cycles/s**

**1.7** Compilers can have a profound impact on the performance of an application. Assume that for a program, compiler A results in a dynamic instruction count of 1.0E9 and has an execution time of 1.1 s, while compiler B results in a dynamic instruction count of 1.2E9 and an execution time of 1.5 s.

a. Find the average CPI for each program given that the processor has a clock cycle time of 1 ns.

**ET = IC*CPI*CT, CPI = ET/(IC*CT), CT = 1 ns**

**$CPI_A$ = 1.1 s/(1.00 E+09 instructions*1 E-09 s/cycle) = 1.1 cycles/instruction**
**$CPI_B$ = 1.5 s/(1.20 E+09 instructions*1.2 E-09 s/cycle) = 1.25 cycles/instruction**

b. Assume the compiled programs run on two different processors. If the execution times on the two processors are the same, how much faster is the clock of the processor running compiler A ;s code versus the clock of hte processor running compiler B's code ?

$$\frac{P_A}{P_B} = \frac{ET_B}{ET_A} = \frac{\frac{1.2E9 instructions \times 1.25 cycles / instruction}{CR_B}}{\frac{1E9 instructions \times 1.1 cycles / instructions}{CR_A}} = 1$$

$$\frac{1.5E9 cycles}{CR_B} = \frac{1.1E9 cycles}{CR_A}$$

$$1.5E9 cycles \times CR_A = 1.1E9 cycles CR_B$$

$$CR_B / CR_A = 1.5/1.1 = 1.37$$

c. A new compiler is developed that uses only 6.0$^E$8 instructions and has an average CPI of 1.1. What is the speedup of using this new compiler versus using compiler A or B on the original processor ?

$$\frac{P_{new}}{P_A} = \frac{ET_A}{ET_{new}} = \frac{\frac{1E9 instructions \times 1.1 cycles / instruction}{1E9 cycles / s}}{\frac{6E8 instructions \times 1.1 cycles / instructions}{1E9 cycles / s}} = \frac{1.0 \times 1.1}{0.6 \times 1.1} = 1.67$$

$$\frac{P_{new}}{P_B} = \frac{ET_B}{ET_{new}} = \frac{\frac{1.2E9 instructions \times 1.25 cycles / instruction}{1E9 cycles / s}}{\frac{6E8 instructions \times 1.1 cycles / instructions}{1E9 cycles / s}} = \frac{1.2 \times 1.25}{.06 \times 1.1} = 2.27$$

**1.9** Assume for arithmetic, load/store, and branch instructions, a processor has CPIs of 1, 12, and 5, respectively. Also assume that on a single processor a program requires the execution of 2.56E9 arithmetic instructions, 1.28E9 load/store instructions, and 256 million branch instructions. Assume that each processor has a 2 GHz clock frequency.

Assume that, as the program is parallelized to run over multiple cores, the number of arithmetic and load/store instructions per processor is divided by 0.7 x p (where p is the number of processors) but the number of branch instructions per processor remains the same.

**1.9.1** Find the total execution time for this program on 1, 2, 4, and 8 processors, and show the relative speedup of the 2, 4, and 8 processor result relative to the single processor result.

**1 processor: ET = CC*CT = (2.56E9*1 + 1.28E9*12 + 0.256E9*5)*0.5 ns = 9.6 s**
**2 processors: ET = CC*CT = (1.83E9*1 + 0.914E9*12 + 0.256E9*5)*0.5 ns = 7.04s**
**4 processors: ET = CC*CT = (0.912E9*1 + 0.457E9*12+ 0.256E9*5)*0.5 ns = 3.84 s**
**8 processors: ET = CC*CT = (0.457E9*1 + 0.229E9*12 + 0.256E9*5)*0.5 ns = 2.24 ss**

**Speedup 2-1 = 9.6/7.04 = 1.36**
**Speedup 4-1 = 9.6/3.84 = 2.5**
**Speedup 8-1 = 9.6/2.24 = 4.28**

**1.9.3** To what should the CPI of load/store instructions be reduced in order for a single processor to match the performance of four processors using the original CPI value?

**$ET_{4\ processors}$ = 3.84 s**
**3.84s = (2.56E9 x 1 + 1.28E9 x $CPI_{L/S}$ + 0.256E9 x 5)*0.5 ns**
**7.68E9 = 2.56E9 + 1.28E9 x $CPI_{L/S}$ + 1.28E9**
**3.84E9 = 1.28E9 x $CPI_{L/S}$**
**$CPI_{L/S}$ = 3**

**1.11** The results of the SPEC CPU2006 bzip2 benchmark running on an AMD Barcelona has an instruction count of 2.389E12, an execution time of 750 s, and a reference time of 9650 s.

**1.11.1** Find the CPI if the clock cycle time is 0.333 ns.

**ET = IC x CPI x CT**
**CPI = ET/(IC x CT)**
**CPI = (ET x CR)/IC**
**CPI = 750 s x 3 E9 cycles/s/(2.389E12 instructions) = 0.941 cycles/instruction**

**1.11.9** For a second benchmark, libquantum, assume an execution time of 960 s, CPI of 1.61, and clock rate of 3 GHz. If the execution time is reduced by an additional 10% without affecting the CPI and with a clock rate of 4 GHz, determine the number of instructions.

**$ET_{new}$ = 0.9 x 960 s, $CPI_{new}$ = $CPI_{old}$ = 1.61, $CR_{new}$ = 4 GHz**
**ET = (IC x CPI)/CR**
**IC = (ET x CR)/CPI**
**$IC_{new}$ = (0.9 * 960 s x 4 E9 cycles/s)/1.61 cycles/instruction**
**$IC_{new}$ = 2.146 E12 instructions**

**1.11.11** Determine the clock rate if the CPI is reduced by 15% and the CPU time by 20% while the number of instructions is unchanged.

**ET = (IC x CPI)/CR**
**CR = (IC x CPI)/ET**
**CPInew = 0.85 x CPIold, ETnew = 0.85 x ETold**
**ICnew = ICold**
**Using original information**
**IC = (960 s x 3 E9 cycles/s)/1.61 = 1.789 E 12 instructions**
**$CR_{new}$ = (1.789 E12 instructions x 0.85 x 1.61 cycles/instruction)/0.80 x 960 s**
**CRnew = 3.19 GHz**
**Using results of 1.11.9**
**IC = 2.146 E12 instructions**
**$CR_{new}$ = (2.146 E12 instructions x 0.85 x 1.61 cycles/instruction)/0.80 x 960 s**
**$CR_{new}$ = 3.82 GHz**

**1.12** Section 1.10 cites as a pitfall the utilization of a subset of the performance equation as a performance metric. To illustrate this, consider the following two processors. P1 has a clock rate of 4 GHz, average CPI of 0.9, and requires the execution of 5.0 E9 instructions. P2 has a clock rate of 3 GHz, an average CPI of 0.75, and requires the execution of 1.0 E9 instructions.

**1.12.3** A common fallacy is to use MIPS (millions of instructions per second) to compare the performance of two different processors, and consider that the processor with the largest MIPS has the largest performance. Check if this is true for P1 and P2.

$$MIPS = \frac{IC}{ET \times 10^6} = \frac{IC}{\frac{IC \times CPI}{CR} \times 10^6} = \frac{IC \times CR}{IC \times CPI} \times 10^{-6} = \frac{CR}{CPI} \times 10^{-6}$$

**$MIPS_{P1}$ = (4 E9 cycles/s/0.9) E-6 = 4.44 E3**
**$MIPS_{P2}$ = (3 E9 cycles/s/0.75) E-6 = 4.0 E3**
**By MIPS, P1 has higher performance than P2.**

$$\frac{P_{P2}}{P_{P1}} = \frac{\frac{1}{ET_{P2}}}{\frac{1}{ET_{P1}}} = \frac{ET_{P1}}{ET_{P2}} = \frac{\frac{IC_{P1} \times CPI_{P1}}{CR_{P1}}}{\frac{IC_{P2} \times CPI_{P2}}{CR_{P2}}} = \frac{IC_{P1} \times CPI_{P1} \times CR_{P2}}{IC_{P2} \times CPI_{P2} \times CR_{P1}} = \frac{5 \times 10^9 \times 0.9 \times 3 \times 10^9}{1 \times 10^9 \times 0.75 \times 4 \times 10^9} = 4.5$$

**The processor with the highest MIPS does not have the highest performance.**

**1.12.4** Another common performance figure is MFLOPS (millions of floating-point operations per second), defined as MFLOPS = Number of Floating Point operations/(execution time x 1E6) but this figure has the same problems as MIPS. Assume that 40% of the instructions executed on both P1 and P2 are floating-point instructions. Find the MFLOPS figures for the programs.

$$MFLOPS = \frac{FLOPS}{ET \times 10^6} = \frac{IC \times 0.4}{\frac{IC \times CPI}{CR} \times 10^6} = \frac{IC \times 0.4 \times CR}{IC \times CPI} \times 10^{-6} = \frac{0.4 \times CR}{CPI} \times 10^{-6}$$

**$MFLOPS_{P1}$ = ((0.4 FLOPS x 4 E9 cycles/s)/0.9) E-6 = 1.78 E3 MFLOPS**
**$MFLOPS_{P1}$ = ((0.4 FLOPS x 3 E9 cycles/s)/0.75) E-6 = 1.6 E3 MFLOPS**
**By MFLOPS, P1 has higher performance than P2, but P2 has higher performance than P1.**

**1.13** Another pitfall cited in Section 1.10 is expecting to improve the overall performance of a computer by improving only one aspect of the computer. Consider a computer running a program that requires 250 s, with 70 s spent executing FP instructions, 85 s executing L/S instructions, and 40 s spent executing branch instructions.

**1.13.2** By how much is the time for INT operations reduced if the total time is reduced by 20%

**$ET_{Total}$ = 250 s, $ET_{FP}$ = 70 s, $ET_{L/S}$ = 85 s, $ET_{B}$ = 40 s**
**$ET_{INT}$ = 250 s – 70s – 85 s – 40 s = 55 s**
**$ET_{Total\_New}$ = 250 s * 0.8 = 200 s, ETINT_New = 200 s – 70 s – 85 s – 40 s = 5 s**
**$ET_{INT\_New}/ET_{INT}$ = 5 s/55 s = 0.091**
**Looking at it another way % Reduction = (55 – 5)/55 x 100 = 90.9 %**

**1.14** Assume a program requires the execution of 50 E6 FP instructions, 110 E6 INT instructions, 80 E6 L/S instructions, and 16 E6 branch instructions. The CPI for each type of instruction is 1, 1, 4, and 2, respectively. Assume that the processor has a 2 GHz clock rate.

**1.14.1** Can the total time be reduced by 20% by reducing only the time for branch instructions?

**ET = (50 E6 instructions x 1 cycle/instruction + 110 E6 instructions x 1 cycle/instruction + 80 E6 instructions x 4 cycles/instruction + 16 E6 instructions x 2 cycles/instruction)/2 E9 cycles/s**

**$ET_{Total}$ = 256 ms**
**$ET_{branch}$ = (16 E6 instructions x 2 cycles/instruction)/2 E9 cycles/s = 16 ms**
**$ET_{branch}/ET_{Total}$ = 16 ms/256 ms = 0.051**
**Since only 5.1 % of the total execution time is spent executing branches, reducing only the time for branch instructions cannot decrease the total execution time by 20%.**