Eric Paulz

CPSC 3300
Homework 2
Due before class 3:30PM on Thursday, February 1
Submit your answers to canvas

Please provide sufficient space on your homework solutions so that your calculations and answers are easily readable and so that grading will be easier. Furthermore, except for the simplest questions, giving only the answer without showing your work is not acceptable. For the best chance at partial credit, show the generic equation you are starting with and any derivations needed to handle the information as given in the question, then plug in the values from the question. You may, of course, use a calculator for the homework. (Unlike the exams, the values in the homework questions are not necessarily chosen for ease of hand calculation.)

1. (30pt) A processor P has a 4.0 GHz clock rate and has a CPI of 2.2.

(a) If the processor executes a program in 20 seconds, find the number of cycles and the number of instructions.

20s = CPU Clock Cycles / 4.0GHz
CPU Clock Cycles = 20s * 4.0GHZ = **$80*10^9$ cycles**

$80*10^9$ = Instruction Count * 2.2
Instruction Count = ($80*10^9$) / 2.2 = **$3.636363*10^{10}$ instructions**

(b) What is the MIPS rate for the processor?

MIPS = 4.0GHz / (2.2 * 10^6) = 4.0GHz / 2,200,000 = **$1.818181*10^{-6}$**

(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?

20s – (20*.3) = 14s (new execution time)

14s = ($3.636363*10^{10}$ * 2.2) / new clock rate
new clock rate = ($3.636363*10^{10}$ * 2.2) / 14s = **~5.71GHz**

Eric Paulz

2. (20pt) Consider two different implementation of the same instruction set architecture. The instructions can be divided into four classes according to their CPI (class A, B, C, and D). P1 has a clock rate of 2.5 GHz and CPIs of 1 (class A), 2 (class B), 3 (class C), and 3 (class D).

   Given a program with a dynamic instruction count of 1.0E6 instructions divided into classes as follows: 20% class A, 10% class B, 50% class C, and 20% class D.

   (a) What is the global CPI?

   1,000,000 * .2 = 200,000 (A,D)
   1,000,000 * .1 = 100,000 (B)
   1,000,000 * .5 = 500,000 (C)

   200,000*1 + 100,000*2 + 500,000*3 + 200,000*3 = 2,500,000
   global CPI = 2,500,000 / 1,000,000 = **2.5**

   (b) Find the clock cycles required to run the program on P1.

   clock cycles = 1,000,000 * 2.5 = **2,500,000**

Eric Paulz

3. (20pt) 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: 600 million arithmetic instructions, 250 million load/store instructions, 150 million branch instructions.

   (a) Suppose we find a way to double the performance of the arithmetic instructions. What is the overall speedup of our machine?

   Arithmetic = 600,000,000 / 1,000,000,000 = 60%
   Load/store = 250,000,000 / 1,000,000,000 = 25%
   Branch = 150,000,000 / 1,000,000,000 = 15%

   $\text{Speedup}_{\text{overall}}$ = 1/((1-.6)+(.6/2)) = 1/(.4+.3) = 1/.7 = **1.4286**

   (b) If we find a way to improve the performance of the arithmetic instructions by 10 times, what is the overall speedup of our machine?

   $\text{Speedup}_{\text{overall}}$ = 1/((1-.6)+(.6/10)) = 1/(.4+.06) = 1/.46= **2.1739**

Eric Paulz

4. (30pt) On machine newton, examine how compiler optimization levels and options change the number of instructions for the program whetstone and the number of CPU cycles to execute the program. Use gcc to compile your program.

   Examine the following levels/options:
   a. -O0
   b. -O1
   c. -O2
   d. -O3
   e. -O3 -funroll-loops

   Use a table to show the instruction count, #cycles, IPC, and time for each of the experiments, and calculate the speedup based on the execution time with -O0. Paste your screen shot at the end.

	IC	#Cycles	IPC	Time	Speedup
-O0	22,784,694,724	16,891,747,796	1.35	6.34551s	--
-O1	10,958,855,008	8,818,410,264	1.24	3.45466s	1.84
-O2	6,054,132,512	6,140,402,667	0.99	2.49257s	1.39
-O3	6,024,060,884	6,092,805,113	0.99	2.47805s	1.01
-O3 -funroll-loops	5,013,689,208	5,480,624,673	0.91	2.26039s	1.09

** screenshot on next page

```
[11:06:12] epaulz@newton:~/spring18/3300/hw2 [21] perf stat -e instructions -e cycles ./whetstone_O0 200000

Loops: 200000, Iterations: 1, Duration: 6 sec.
C Converted Double Precision Whetstones: 3333.3 MIPS

 Performance counter stats for './whetstone_O0 200000':

    22,784,694,724      instructions              #    1.35  insns per cycle
    16,891,747,796      cycles

       6.345514987 seconds time elapsed

[11:06:40] epaulz@newton:~/spring18/3300/hw2 [22] perf stat -e instructions -e cycles ./whetstone_O1 200000

Loops: 200000, Iterations: 1, Duration: 3 sec.
C Converted Double Precision Whetstones: 6666.7 MIPS

 Performance counter stats for './whetstone_O1 200000':

    10,958,855,008      instructions              #    1.24  insns per cycle
     8,818,410,264      cycles

       3.454661267 seconds time elapsed

[11:07:00] epaulz@newton:~/spring18/3300/hw2 [23] perf stat -e instructions -e cycles ./whetstone_O2 200000

Loops: 200000, Iterations: 1, Duration: 3 sec.
C Converted Double Precision Whetstones: 6666.7 MIPS

 Performance counter stats for './whetstone_O2 200000':

     6,054,132,512      instructions              #    0.99  insns per cycle
     6,140,402,667      cycles

       2.492565900 seconds time elapsed

[11:07:18] epaulz@newton:~/spring18/3300/hw2 [24] perf stat -e instructions -e cycles ./whetstone_O3 200000

Loops: 200000, Iterations: 1, Duration: 3 sec.
C Converted Double Precision Whetstones: 6666.7 MIPS

 Performance counter stats for './whetstone_O3 200000':

     6,024,060,884      instructions              #    0.99  insns per cycle
     6,092,805,113      cycles

       2.478046043 seconds time elapsed

[11:07:39] epaulz@newton:~/spring18/3300/hw2 [25] perf stat -e instructions -e cycles ./whetstone_O3_funroll-loops 200000

Loops: 200000, Iterations: 1, Duration: 3 sec.
C Converted Double Precision Whetstones: 6666.7 MIPS

 Performance counter stats for './whetstone_O3_funroll-loops 200000':

     5,013,689,208      instructions              #    0.91  insns per cycle
     5,480,624,673      cycles

       2.260385855 seconds time elapsed

[11:08:11] epaulz@newton:~/spring18/3300/hw2 [26]
```