**1.5**

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

CPU Time1 = (IC \* CPI)/CR = (IC \* 1.5)/3\*10^9 = 5\*10^-10(IC)

Performace1 = 1/(CPU time) = 1/ (5\*10^-10(IC))=2\*10^9(IC)

CPU Time2 = (IC \* CPI)/CR = (IC \*1)/2.5\*10^9 = 4\*10^-10(IC)

Performace2 = 1/(4\*10^-10(IC))= 2.5\*10^9(IC)

CPU Time3 = (IC \* CPI)/CR = (IC \* 2.2)/ 4\*10^9 = 5.5\*10^-10

Performace3 = 1/ 5.5\*10^-10 = 1.8\*10^9(IC)

**CPU P2 has the highest performance 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.

P1

Cpu time = (IC \* CPI)/CR

10 = (CC)/3\*10^9

**(3\*10^10)=CC**

CC = (IC \* CPI) =

10\*(3\*10^9) = (IC \* 1.5)

**IC = 10\*(3\*10^9)/1.5 = 2x10^10**

P2

Cpu time = (IC \* CPI)/CR

10 = (CC)/2.5\*10^9

**CC = (2.5\*10^10)**

IC = CC/cpi

IC = (2.5\*10^10)/1

**IC = (2.5\*10^10)**

P3

Cpu time = (IC \* CPI)/CR

10 = (CC)/4\*10^9

**CC = 4 \* 10^10**

IC = CC/cpi

IC= (4\*10^10)/2.2

IC = 1.8\*10^10

1.6

a. What is the global CPI for each implementation?

P1=(1\*10%)+(2\*20%)+(3\*50%)+(3\*20%)

=.1+.4+1.5+.6

=2.6

P2 = (2\*10%)+(2\*20%)+(2\*50%)+(2\*20%)

= .2+.4+1+.4

=2

b. Find the clock cycles required in both cases.

1.7

a. Find the average CPI for each program given that the processor has a clock cycle

time of 1 ns.

Compiler a:

Cpu time = IC \* CPI \* CT

1.1=(1 \* 10^9)(CPI)(1x10^-9)

1.1 = CPI

Compiler b:

Cpu time = IC \* CPI \* CT

1.5 = (1.2\*10^9)(CPI)(1x10^-9)

1.5=1.2(CPI)

1.25 = CPI

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 the processor running

compiler B’s code?

c. A new compiler is developed that uses only 6.0E8 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?

Cpu time = IC \* CPI \* CT

=(6\*10^8)\*(1.1)\*(1\*10^-9)

= .66 s

Speedup = execution time2/execution time1

Compiler a = 1.1/.66 = 1.66 speedup

Compiler b = 1.5/.66 = 2.27 speedup

1.12.1 One usual fallacy is to consider the computer with the

largest clock rate as having the largest performance. Check if this is true for P1 and

P2.

P1

Cpu time = (IC \* CPI)/CR

=(5\*10^9)\*.9/(4\*10^9)

= 1.125

P2

Cpu time = (IC \*CPI)/CR

= (1 \*10^9)\*.75/(3\*10^9)

=.25

P2 completes the program a lot faster even though its clock rate is 1GHz slower.

1.12.2 Another fallacy is to consider that the processor executing

the largest number of instructions will need a larger CPU time. Considering that

processor P1 is executing a sequence of 1.0E9 instructions and that the CPI of

processors P1 and P2 do not change, determine the number of instructions that P2

can execute in the same time that P1 needs to execute 1.0E9 instructions. `

P1

Cpu time = (IC \* CPI)/CR

= (1 \* 10^9)\*.9/(4 \* 10^9)

=.225

P2

Cpu time = (IC \* CPI)/CR

.225 = (IC \* .75)/(3\*10^9)

.225 \* (3 \* 10^9) = IC \* .75

(.225\*(3\*10^9)/.75 = IC

IC = (9\*10^8)

1.12.3 A common fallacy is to use MIPS (millions of

instructions per second) to compare the performance of two diff erent processors,

and consider that the processor with the largest MIPS has the largest performance.

Check if this is true for P1 and P2.

MIPS = CR/ (CPI\*10^6)

P1

= (4\*10^9)/(.9\*10^6)

= 4.4\*10^3

P2

=(3\*10^9)/(.75\*10^6)

=4\*10^3

P1 has a higher mips rating but it got outperformed in 1.12.1

**1.13.1** By how much is the total time reduced if the time for FP

operations is reduced by 20%

time after improvement = time affected/

The total time is only reduced by 14 seconds since FP only accounts for 70 seconds of the execution time and there is only a 20% reduction on those 70 seconds.

**1.13.2** [5] <§1.10> By how much is the time for INT operations reduced if the

total time is reduced by 20%?

55/250 =.22

The expected time reduction for integers is about 2.42 seconds since integers operations only account for 22% of the total time.

**1.13.3** Can the total time can be reduced by 20% by reducing only

the time for branch instructions?

250 – 250\*.2 =200

The total time cannot be reduced by 20 percent by only changing branch instruction time. This is because 20 percent of 250 is 200 seconds, since branch instructions only account for 40s there is no way to reduce the time by 50s by only affecting the 40s portion.