CS 402 HW-2

**1.5 [4] <§1.6> Consider three diff erent 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.**

|  |  |  |  |
| --- | --- | --- | --- |
|  | P1 (GHz) | P2 (GHz) | P3 (GHz) |
| f  (clock rate) | 3 | 2.5 | 4.0 |
| CPI | 1.5 | 1.0 | 2.2 |

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

A) Instructions per second = f / CPI

P1 → 3 x 109 / 1.5  = 2 x 109

P2 → 2.5 x 109 / 1.0  = 2.5 x 109

P3 → 4 x 109 / 2.2  = 1.8 x 109

P2 has the highest performance

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

cycles and the number of instructions.

A) Given CPUtime = 10s

Instructions

IC = (CPUtime x f) / CPI      IC: Instruction count

P1 → (10 x 3 x 109 ) / 1.5  =  2 x 1010 Instructions

P2 → (10 x 2.5 x 109 ) / 1  =  2.5 x 1010 Instructions

P3 → (10 x 4 x 109 ) / 2.2  =  1.82 x 1010 Instructions

Cycles

Clock cycles = CPUtime x f

P1 → (10 x 3 x 109 )   = 3 x 1010  cycles

P2 → (10 x 2.5 x 109 ) = 2.5 x 1010 cycles

P3 → (10 x 4 x 109 )  = 4 x 1010 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?

A)   fnew =  (Told x CPInew x fold) / (CPIold x Tnew) T: CPUtime

Let us consider the CPU time (execution time ) to be **y** and the new execution time is **0.7y** (because we are trying to reduce the execution time by 30%)

Increase of 20% in the CPI =>  CPInew = 1.2 x 1.5

P1 →   (y x  (1.2 x 1.5) x 3 x 109 ) / ( 1.5 x 0.7y )  =  5.14 GHz

P2 →   (y x  (1.2 x 1) x 2.5 x 109 ) / ( 1 x 0.7y )  =     4.29 GHz

P3 →   (y x  (1.2 x 2.2) x 4 x 109 ) / ( 2.2 x 0.7y )  =  6.86 GHz

**1.6 [20] <§1.6> Consider two different implementations 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 with a clock rate of 2.5 GHz and CPIs of 1, 2, 3,**

**and 3, and P2 with a clock rate of 3 GHz and CPIs of 2, 2, 2, and 2.**

**Given a program with a dynamic instruction count of 1.0E6 instructions divided**

**into classes as follows: 10% class A, 20% class B, 50% class C, and 20% class D,**

**which implementation is faster?**

A)

CPUtime = ∑ (IC x CPI) / f         IC: Instruction count, f: clock rate

a. What is the global CPI for each implementation?

b. Find the clock cycles required in both cases.

**P1**

**CPUtime**= ( ICA x CPIA + ICB x CPIB + ICC x CPIC  + ICD x CPID ) / f

= (105 x 26) / (2.5 x 109)

= 10.4 x 10-4 s

**Global CPI** =  (CPUtime x f) / IC

= (10.4 x 10-4 x 2.5 x 109) / 106

= 2.6

**Clock Cycles** = ∑ (IC x CPI) = 105 x 26 = 2.6 x 106

**P2**

**CPUtime**= ( ICA x CPIA + ICB x CPIB + ICC x CPIC  + ICD x CPID ) / f

= (105 x 20) / (3 x 109)

= 6.67 x 10-4 s

**Global CPI** = (CPUtime x f) / IC

= (6.67 x 10-4  x 3 x 109) / 106

= 2

**Clock Cycles** = ∑ (IC x CPI) = 105 x 20 = 2x106

**P2 implementation is faster (CPUtime of P1 > CPUtime of P2  ).**

**1.7 [15] <§1.6> 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)  Given,

|  |  |  |
| --- | --- | --- |
|  | Compiler A | Compiler B |
| Execution time ( CPU time) T | 1.1s | 1.5s |
| Instruction count (IC) | 109 | 1.2 x 109 |
| Frequency (f) | 109 | 109 |

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

time of 1 ns.   f  = 1 / clock cycle = 1 / 10-9 = 109

CPIA = (f x TA) / ICA

= 109  x 1.1 / 109

= 1.1

CPIB = (f x TB) / ICB

= 109  x 1.5 / 1.2 x 109

= 1.25

b. Assume the compiled programs run on two diff erent 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?

If TA= TB

fB/ fA = (ICB X CPIB) / (ICA X CPIA)

= (1.25 x 1.2 )/ 1.1

= 1.36

The clock speed of B is 36 % faster than A.

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?

Given,

IC = 6 X 108

CPI= 1.1

T = (IC x CPI) / f

Tnew (execution time) = (6 X 108 X 1.1) / 109 =  0.66s

Speedup = TA / Tnew  = 1.1/ 0.66 = 1.67

Speedup = TB / Tnew  = 1.5/ 0.66 = 2.27

The new compiler  is 67% faster than compiler A

The new compiler  is 127% faster than the compiler B.

**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 [5] <§1.7> 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.**

A)

Given,

f = 2 x 109 Hz

|  |  |  |  |
| --- | --- | --- | --- |
|  | Arithmetic (A) | Load / Store (LS) | Branch (B) |
| CPI | 1 | 12 | 5 |
| IC | 2.56E9 | 1.28E9 | 2.56E8 |
|  |  |  |  |

 Total execution time (TT )  = TA + TLS + TB

When p = 1

TT    = ((ICA x CPIA) + (ICLS x CPILS) + (ICB x CPIB)) /   f

TT  = 9.6s

When p=2

TT    =  (((ICA x CPIA) + (ICLS x CPILS)) / (0.7x 2)) + (ICB x CPIB)) /   f

TT    = 7.04s                        Speedup = 9.6 / 7.04 = 1.36

When p=4

TT    =  (((ICA x CPIA) + (ICLS x CPILS)) / (0.7x 4)) + (ICB x CPIB)) /   f

TT    = 3.84s Speedup = 9.6 / 3.84 = 2.5

When p=8

TT    =  (((ICA x CPIA) + (ICLS x CPILS)) / (0.7x 8)) + (ICB x CPIB)) /   f

TT    = 2.24s Speedup = 9.6 / 2.24 = 4.29

**1.9.2 [10] <§§1.6, 1.8> If the CPI of the arithmetic instructions was doubled,**

**what would the impact be on the execution time of the program on 1, 2, 4, or 8**

**processors?**

Given,

f = 2 x 109 Hz

|  |  |  |  |
| --- | --- | --- | --- |
|  | Arithmetic (A) | Load / Store (LS) | Branch (B) |
| CPI | 1 | 12 | 5 |
| IC | 2.56E9 x 2 | 1.28E9 | 2.56E8 |
|  |  |  |  |

 Total execution time (TT )  = TA + TLS + TB

When p = 1

TT    = ((ICA x CPIA) + (ICLS x CPILS) + (ICB x CPIB)) /   f

TT  = 10.88s

When p=2

TT    =  (((ICA x CPIA) + (ICLS x CPILS)) / (0.7x 2)) + (ICB x CPIB)) /   f

TT    = 7.954s                        Speedup = 9.6 / 7.954 = 1.2

When p=4

TT    =  (((ICA x CPIA) + (ICLS x CPILS)) / (0.7x 4)) + (ICB x CPIB)) /   f

TT    = 4.29s Speedup = 9.6 / 4.29 = 2.23

When p=8

TT    =  (((ICA x CPIA) + (ICLS x CPILS)) / (0.7x 8)) + (ICB x CPIB)) /   f

TT    = 2.46s Speedup = 9.6 / 2.46= 3.9

**1.9.3 [10] <§§1.6, 1.8> 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 values?**

A)  Performance of four processors  (when p=4) = 3.84s

CPInew : New CPI of load/ store instructions

3.84 =  ((ICA x CPIA) + (ICLS x CPInew) + (ICB x CPIB)) /   f

3.84 = ((2.56E9 x 1) + (1.28E9 x CPInew) + (5 x 2.56E8)) / 2E9

CPInew  = 3

**1.11 Th e 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 [5] <§§1.6, 1.9> Find the CPI if the clock cycle time is 0.333 ns.**

A)

Given,

IC = 2.389E12

CPUtime = 750s

Clock cycle time (TCLK ) = 0.333 x 10-9 s

CPUtime = IC x CPI x TCLK

750 = 2.389E12 x CPI x 333 x 10-12

CPI = 0.942

**1.11.2 [5] <§1.9> Find the SPECratio.**

A)

SPEC ratio = Reference time / execution time = 9650 /750 = 12.87

**1.11.3 [5] <§§1.6, 1.9> Find the increase in CPU time if the number of instructions**

**of the benchmark is increased by 10% without affecting the CPI.**

A)

Given,

ICnew = 1.1 x2.389E12

CPUtime = ICnew x CPI x TCLK

= 1.1 x2.389E12 x 0.942 x 0.333 x 10-9

= 824.33s

**1.11.4 [5] <§§1.6, 1.9> Find the increase in CPU time if the number of instructions**

**of the benchmark is increased by 10% and the CPI is increased by 5%.**

A)

Given,

ICnew = 1.1 x2.389E12

CPInew = 1.05 x 0.942

CPUtime = ICnew x CPInew x TCLK

= 1.1 x2.389E12 x 1.05x 0.942 x 0.333 x 10-9

= 865.55s

**1.11.5 [5] <§§1.6, 1.9> Find the change in the SPECratio for this change.**

A)

SPEC ratio = Reference time / execution time  = 9650 / 865.55 = 11.14

**1.11.6 [10] <§1.6> Suppose that we are developing a new version of the AMD**

**Barcelona processor with a 4 GHz clock rate. We have added some additional**

**instructions to the instruction set in such a way that the number of instructions**

**has been reduced by 15%. The execution time is reduced to 700 s and the new**

**SPECratio is 13.7. Find the new CPI.**

A)

 Given,

 f = 4E9 Hz

ICnew = 0.75 x 2.389E12

CPUtime = 700s

SPECratio = 13.7

CPUtime = IC x CPInew / f

700 = (0.85 x  2.389E12 x CPInew )/ 4E9

CPInew =   1.378

**1.11.7 [10] <§1.6> This CPI value is larger than obtained in 1.11.1 as the clock**

**rate was increased from 3 GHz to 4 GHz. Determine whether the increase in the**

**CPI is similar to that of the clock rate. If they are dissimilar, why?**

A)  CPI increase is similar to that of clock rate, since instructions count is reduced

**1.11.8 [5] <§1.6> By how much has the CPU time been reduced?**

A)  Reduction % = ((CPUtime old  - CPUtime new)/ CPUtime old) x 100

    = (750-700 / 750 ) x 100

    = 6.67 %

**1.11.9 [10] <§1.6> For a second benchmark, libquantum, assume an execution**

**time of 960 ns, CPI of 1.61, and clock rate of 3 GHz. If the execution time is**

**reduced by an additional 10% without affecting to the CPI and with a clock rate of**

**4 GHz, determine the number of instructions.**

A)

IC = CPUtime x f / CPI

= 960 x 10-9 x 4E9 / 1.61

= 2385

ICnew = (CPUtime new  x fnew )/ ( CPI )

ICnew = (0.9x 960 x 10-9  x 4E9) /(1.61)

ICnew = 2147

**1.11.10 [10] <§1.6> Determine the clock rate required to give a further 10%**

**reduction in CPU time while maintaining the number of instructions and with the**

**CPI unchanged.**

A)   f = IC  x CPI / CPUtime

CPInew = 1.61

CPUtime new = (0.9 x 960 x 10-9)s

IC = 2147

f  = 2147  x 1.61 / 777.6x 10-9

   = 4.45E9 Hz

**1.11.11 [10] <§1.6> Determine the clock rate if the CPI is reduced by 15% and**

**the CPU time by 20% while the number of instructions is unchanged.**

A)

CPInew = 0.85 x 1.61

CPUtime new = 0.8 x 777.6 x 10-9 s

IC = 2147

f  = 2147  x 1.368 / 0.8 x 777.6x 10-9

   = 4.72E9 Hz

**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 5.0E9 instructions. P2 has a clock rate of 3 GHz, an average CPI of**

**0.75, and requires the execution of 1.0E9 instructions.**

**1.12.1 [5] <§§1.6, 1.10> 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.**

A)

**1.12.2 [10] <§§1.6, 1.10> 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.**

**1.12.3 [10] <§§1.6, 1.10> 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.**

**1.12.4 [10] <§1.10> Another common performance figure is MFLOPS (millions**

**of floating-point operations per second), defined as**

**MFLOPS = No. FP operations / (execution time × 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.**