Computer Architecture and Logic HW 1

Due : 4/12(Sun) 23:59

Problem #1 [15points]

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

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?

a. $CPI = \frac{\text{CPU Time}}{\text{Instruction Count} \times \text{Clock Cycle Time}}$

CPI of Compiler A : $\frac{1.1s}{1.0E9 \times 1ns} = 1.1$

CPI of Compiler B : $\frac{1.5s}{1.2E9 \times 1ns} = 1.25$

b. CPU Time = Instruction Count x CPI x Clock Cycle Time

$\frac{\text{Clock Cycle Time of A}}{\text{Clock Cycle Time of B}} = \frac{1.2E9 \times 1.25}{1.0E9 \times 1.1} = 1.36...$ ∴ About 1.36 times faster

c. $\text{CPU Time} = \frac{\text{Instructon Count} \times \text{CPI}}{\text{Clock Rate}}$

$\frac{\text{CPU time of A}}{\text{CPU Time of new}} = \frac{1.0E9 \times 1.1}{6.0E8 \times 1.1} = 1.66...$ ∴ About 1.67 times faster than A

$\frac{\text{CPU time of B}}{\text{CPU Time of new}} = \frac{1.2E9 \times 1.25}{6.0E8 \times 1.1} = 2.27...$ ∴ About 2.27 times faster than B

Problem #2 [25points]

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.

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?

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?

1.9.1. CPU Time = $\frac{\text{Instructon Count x CPI}}{\text{Clock Frequency}}$

1 processor : $\frac{2.56E9 \times 1 + 1.28E9 \times 12 + 2.56E8 \times 5}{2GHz}$ = 9.6s

2 processors : $\frac{(2.56E9 \times 1 + 1.28E9 \times 12)/(0.7 \times 2) + 2.56E8 \times 5}{2GHz}$ = 7.04s

→ 1.36 times speed up

4 processors : $\frac{(2.56E9 \times 1 + 1.28E9 \times 12)/(0.7 \times 4) + 2.56E8 \times 5}{2GHz}$ = 3.84s

→ 2.5 times speed up

8 processors : $\frac{(2.56E9 \times 1 + 1.28E9 \times 12)/(0.7 \times 8) + 2.56E8 \times 5}{2GHz}$ = 2.24s

→ 4.29 times speed up

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.

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?

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?

1.9.2. CPU Time = $\frac{\text{Instructon Count x CPI}}{\text{Clock Frequency}}$

1 processor : $\frac{2.56E9 \times 2 + 1.28E9 \times 12 + 2.56E8 \times 5}{2GHz} = 10.88s$

→ 1.13 times slow down

2 processors : $\frac{(2.56E9 \times 2 + 1.28E9 \times 12)/(0.7 \times 2) + 2.56E8 \times 5}{2GHz} = 7.95s$

→ 1.13 times slow down

4 processors : $\frac{(2.56E9 \times 2 + 1.28E9 \times 12)/(0.7 \times 4) + 2.56E8 \times 5}{2GHz} = 4.30s$

→ 1.12 times slow down

8 processors : $\frac{(2.56E9 \times 2 + 1.28E9 \times 12)/(0.7 \times 8) + 2.56E8 \times 5}{2GHz} = 2.47s$

→ 1.10 times slow down

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.

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?

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?

1.9.3. $$\text{CPU Time} = \frac{\text{Instructon Count x CPI}}{\text{Clock Frequency}}$$

$$\text{4 processors} : \frac{(2.56E9 \times 1 + 1.28E9 \times 12)/(0.7 \times 4) + 2.56E8 \times 5}{2GHz} = 3.84s$$

$$\text{1 processor} : \frac{2.56E9 \times 1 + 1.28E9 \times CPI + 2.56E8 \times 5}{2GHz} = 3.84s \rightarrow CPI = 3$$

$\therefore$ should be reduced by 25%

Problem #3 [40points]

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

1.11.2 [5] <§1.9> Find the SPECratio.

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%.

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.

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?

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

1.11.1.

$$\text{CPI} = \frac{\text{CPU Time}}{\text{Instruction Count} \times \text{Clock Cycle Time}}$$

$$= \frac{750\text{s}}{2.389\text{E}12 \times 0.333\text{ns}} = 0.94$$

1.11.2. SPECratio = Reference Time / Execution Time

= 9650s / 750s = 12.87

1.11.4. CPU Time = Instructon Count x CPI x Clock Cycle Time

= (2.389E12 x 1.1) x 0.94 x 1.05 x 0.333ns = 863.71s

CPU Time new / CPU Time origin = 863.71s / 750s = 1.15

∴ About 15% increased

1.11.6.

$$\text{CPI} = \frac{\text{CPU Time}}{\text{Instruction Count} \times \text{Clock Cycle Time}}$$

$$= \frac{700\text{s}}{(2.389\text{E}12 \times 0.85) \times 0.25\text{ns}} = 1.38$$

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.

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?

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

1.11.7. Clock Rate : 4GHz / 3GHz = 1.3 times increased
CPI : 1.38 / 0.94 = 1.47 times increased

Increase in the CPI is dissimilar to that of the Clock Rate.
This is because the reduced Instruction Count also affected the CPI's increase.

1.11.8. $$\text{CPU Time} = \frac{\text{Instructon Count} \times \text{CPI}}{\text{Clock Frequency}}$$

$$\text{CPU Time new} = \frac{(2.389\text{E}12 \times 0.85) \times 1.38}{4\text{GHz}} = 700.57$$

CPU Time origin / CPU Time new = 1.07 ∴ About 7% reduced

Problem #4 [20points]

2.3 [5] <§§2.2, 2.3> For the following C statement, what is the corresponding MIPS assembly code? Assume that the variables `f`, `g`, `h`, `i`, and `j` are assigned to registers `$s0`, `$s1`, `$s2`, `$s3`, and `$s4`, respectively. Assume that the base address of the arrays `A` and `B` are in registers `$s6` and `$s7`, respectively.

```
B[8] = A[i-j];
```

```
sub  $t0, $s3, $s4
sll  $t0, $t0, 2
add  $t0, $t0, $s6
lw   $t1, 0($t0)
sw   $t1, 32($t7)
```

2.10 [5] <§§2.2, 2.3> Translate the following MIPS code to C. Assume that the variables `f`, `g`, `h`, `i`, and `j` are assigned to registers `$s0`, `$s1`, `$s2`, `$s3`, and `$s4`, respectively. Assume that the base address of the arrays `A` and `B` are in registers `$s6` and `$s7`, respectively.

```
addi $t0, $s6, 4      # $t0 = &A[1]
add  $t1, $s6, $0     # $t1 = &A[0]
sw   $t1, 0($t0)      # A[1] = $t1 = &A[0]
lw   $t0, 0($t0)      # $t0 = A[1] = &A[0]
add  $s0, $t1, $t0    # f = &A[0] + &A[0] = 2*(&A[0])
```

∴ f = 2*(&A[0]);

2.14 [5] <§§2.2, 2.5> Provide the type and assembly language instruction for the following binary value: $0000\ 0010\ 0001\ 0000\ 1000\ 0000\ 0010\ 0000_{two}$

ALU	$s0	$s0	$s0	0	add

∴ R-format, add $s0, $s0, $s0

2.17 [5] <§2.5> Provide the type, assembly language instruction, and binary representation of instruction described by the following MIPS fields:

```
op=0x23, rs=1, rt=2, const=0x4
```

∴ I-format, lw $v0, 4($at)

opcode	rs	rt	constant
100011	00001	00010	0000 0000 0000 0100