

Computer Architecture and Logic HW#1 (Due: 4/2 (Fri) 11:59 PM)

#1. 1.6

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.

a)

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?

b)

what is the global CPI for each implementation?

c) Find the clock cycles required in both cases.

#2. 1.8

1.8 The Pentium 4 Prescott processor, released in 2004, had a clock rate of 3.6 GHz and voltage of 1.25 V. Assume that, on average, it consumed 10 W of static power and 90 W of dynamic power.

The Core i5 Ivy Bridge, released in 2012, had a clock rate of 3.4 GHz and voltage of 0.9 V. Assume that, on average, it consumed 30 W of static power and 40 W of dynamic power.

1.8.1 [5] <\$1.7> For each processor find the average capacitive loads.

1.8.2 [5] <\$1.7> Find the percentage of the total dissipated power comprised by static power and the ratio of static power to dynamic power for each technology.

1.8.3 [15] <\$1.7> If the total dissipated power is to be reduced by 10%, how much should the voltage be reduced to maintain the same leakage current? Note: power is defined as the product of voltage and current.

#3. 1.11

The result of the SPEC CPU2006 bzip2 benchmark running on an AMD Barcelona has an instruction count of 2.389×10^{12} , 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.

1.11.2) Find the SPECratio.

1.11.4) 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)

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.

#4. 1.12

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.

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 different 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

$$\text{MFLOPS} = \text{No. FP operations} / (\text{execution time} \times 1\text{E}6)$$

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.

#5. 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 executed L/S instructions, and 40 s spent executing branch instructions.

1.13.1)

By how much is the total time reduced if the time for FP operations is reduced by 20%?

1.13.2)

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

1.13.3)

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

#6. 2.4

2.4 [5] <§§2.2, 2.3> For the MIPS assembly instructions below, what is the corresponding C statement? 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.

```
sll    $t0, $s0, 2      # $t0 = f * 4
add    $t0, $s6, $t0    # $t0 = &A[f]
sll    $t1, $s1, 2      # $t1 = g * 4
add    $t1, $s7, $t1    # $t1 = &B[g]
lw     $s0, 0($t0)      # f = A[f]
addi   $t2, $t0, 4
lw     $t0, 0($t2)
add    $t0, $t0, $s0
sw     $t0, 0($t1)
```

#7. 2.15

2.15 [5] <§§2.2, 2.5> Provide the type and hexadecimal representation of following instruction: `sw $t1, 32($t2)`

#8. 2.24

Suppose the program counter (PC) is set to 0x2000 0000. Is it possible to use the jump (j) MIPS assembly instruction to set the PC to the address as 0x4000 0000?

Is it possible to use the branch-on-equal (beq) MIPS assembly instruction to set the PC to this same address?

#9. 2.25

The following instruction is not included in the MIPS instruction set:

`rpt $t2, loop // if(R[rs] > 0) R[rs] = R[rs] - 1, PC = PC + 4 + BranchAddr`

2.25.1)

If this instruction were to be implemented in the MIPS instruction set, what is the most appropriate instruction format?

2.25.2)

What is the shortest sequence of MIPS instructions that performs the same operation?