UMSL

CMP SCI 2700 – Homework 1

Complete Chapter 2 End of Chapter **Problems**

- 2.2
- 2.5

Complete Chapter 9 End of Chapter **Problems**

- 9.5
- 9.7
- 9.11
- 9.13

Complete Chapter 10 End of Chapter **Problems**

- 10.11
- 10.15

Review Questions

2.1 List and briefly define some of the techniques used in contemporary processors to increase speed.

2.2 Explain the concept of performance balance.

2.3 Explain the differences among multicore systems, MICs, and GPGPUs.

2.4 Briefly characterize Amdahl's law.

2.5 Briefly characterize Little's law.

2.6 Define MIPS and FLOPS.

2.7 List and define three methods for calculating a mean value of a set of data values.

2.8 List the desirable characteristics of a benchmark program.

2.9 What are the SPEC benchmarks?

2.10 What are the differences among base metric, peak metric, speed metric, and rate metric?

Problems

2.1 A benchmark program is run on a 40 MHz processor. The executed program consists of 100,000 instruction executions, with the following instruction mix and clock cycle count:

Instruction Type	Instruction Count	Cycles per Instruction
Integer arithmetic	45,000	1
Data transfer	32,000	2
Floating point	15,000	2
Control transfer	8000	2

Determine the effective *CPI*, MIPS rate, and execution time for this program.

2.2 Consider two different machines, with two different instruction sets, both of which have a clock rate of 200 MHz. The following measurements are recorded on the two machines running a given set of benchmark programs:

Instruction Type	Instruction Count (millions)	Cycles per Instruction
Machine A		
Arithmetic and logic	8	1
Load and store	4	3
Branch	2	4
Others	4	3
Machine A		
Arithmetic and logic	10	1
Load and store	8	2
Branch	2	4
Others	4	3

a. Determine the effective *CPI*, MIPS rate, and execution time for each machine.
b. Comment on the results.

2.3 Early examples of CISC and RISC design are the VAX 11/780 and the IBM RS/6000, respectively. Using a typical benchmark program, the following machine characteristics result:

Processor	Clock Frequency (MHz)	Performance (MIPS)	CPU Time (secs)
VAX 11/780	5	1	12 x
IBM RS/6000	25	18	x

The final column shows that the VAX required 12 times longer than the IBM measured in CPU time.

a. What is the relative size of the instruction count of the machine code for this benchmark program running on the two machines?

b. What are the *CPI* values for the two machines?

2.4 Four benchmark programs are executed on three computers with the following results:

	Computer A	Computer B	Computer C
Program 1	1	10	20
Program 2	1000	100	20
Program 3	500	1000	50
Program 4	100	800	100

The table shows the execution time in seconds, with 100,000,000 instructions executed in each of the four programs. Calculate the MIPS values for each computer for each program. Then calculate the arithmetic and harmonic means assuming equal weights for the four programs, and rank the computers based on arithmetic mean and harmonic mean.

2.5 The following table, based on data reported in the literature [HEAT84], shows the execution times, in seconds, for five different benchmark programs on three machines.

Benchmark	Processor		
	R	M	Z
E	417	244	134
F	83	70	70
H	66	153	135
I	39,449	35,527	66,000
K	772	368	369

a. Compute the speed metric for each processor for each benchmark, normalized to machine R. That is, the ratio values for R are all 1.0. Other ratios are calculated using Equation (2.5) with R treated as the reference system. Then compute the arithmetic mean value for each system using Equation (2.3). This is the approach taken in [HEAT84].

b. Repeat part (a) using M as the reference machine. This calculation was not tried in [HEAT84].

c. Which machine is the slowest based on each of the preceding two calculations?

d. Repeat the calculations of parts (a) and (b) using the geometric mean, defined in Equation (2.6). Which machine is the slowest based on the two calculations?

Hexadecimal notation is not only used for representing integers but also used as a concise notation for representing any sequence of binary digits, whether they represent text, numbers, or some other type of data. The reasons for using hexadecimal notation are as follows:

1. It is more compact than binary notation.
2. In most computers, binary data occupy some multiple of 4 bits, and hence some multiple of a single hexadecimal digit.
3. It is extremely easy to convert between binary and hexadecimal notation.

As an example of the last point, consider the binary string 110111100001. This is equivalent to

$$\begin{array}{cccl} 1101 & 1110 & 0001 & = DE1_{16} \\ D & E & 1 & \end{array}$$

This process is performed so naturally that an experienced programmer can mentally convert visual representations of binary data to their hexadecimal equivalent without written effort.

9.6 KEY TERMS AND PROBLEMS

Key Terms

base	hexadecimal	nibble
binary	integer	positional number system
decimal	least significant digit	radix
fraction	most significant digit	radix point

Problems

9.1 Count from 1 to 20_{10} in the following bases:
a. 8 **b.** 6 **c.** 5 **d.** 3

9.2 Order the numbers $(1.1)_2, (1.4)_{10}$, and $(1.5)_{16}$ from smallest to largest.

9.3 Perform the indicated base conversions:
a. 54_8 to base 5 **b.** 312_4 to base 7 **c.** 520_6 to base 7 **d.** 12212_3 to base 9

9.4 What generalizations can you draw about converting a number from one base to a power of that base; e.g., from base 3 to base 9 (3^2) or from base 2 to base 4 (2^2) or base 8 (2^3)?

9.5 Convert the following binary numbers to their decimal equivalents:
a. 001100 **b.** 000011 **c.** 011100 **d.** 111100 **e.** 101010

9.6 Convert the following binary numbers to their decimal equivalents:
a. 11100.011 **b.** 110011.10011 **c.** 1010101010.1

9.7 Convert the following decimal numbers to their binary equivalents:
a. 64 **b.** 100 **c.** 111 **d.** 145 **e.** 255

9.8 Convert the following decimal numbers to their binary equivalents:
a. 34.75 **b.** 25.25 **c.** 27.1875

9.9 Prove that every real number with a terminating binary representation (finite number of digits to the right of the binary point) also has a terminating decimal representation (finite number of digits to the right of the decimal point).

9.10 Express the following octal numbers (number with radix 8) in hexadecimal notation:

a. 12 **b.** 5655 **c.** 2550276 **d.** 76545336 **e.** 3726755

9.11 Convert the following hexadecimal numbers to their decimal equivalents:

a. C **b.** 9F **c.** D52 **d.** 67E **e.** ABCD

9.12 Convert the following hexadecimal numbers to their decimal equivalents:

a. F.4 **b.** D3.E **c.** 1111.1 **d.** 888.8 **e.** EBA.C

9.13 Convert the following decimal numbers to their hexadecimal equivalents:

a. 16 **b.** 80 **c.** 2560 **d.** 3000 **e.** 62,500

9.14 Convert the following decimal numbers to their hexadecimal equivalents:

a. 204.125 **b.** 255.875 **c.** 631.25 **d.** 10000.00390625

9.15 Convert the following hexadecimal numbers to their binary equivalents:

a. E **b.** 1C **c.** A64 **d.** 1F.C **e.** 239.4

9.16 Convert the following binary numbers to their hexadecimal equivalents:

a. 1001.1111 **b.** 110101.011001 **c.** 10100111.111011

Input	x_{n-1}	0	0	0	0	1	1	1	1
	y_{n-1}	0	0	1	1	0	0	1	1
	c_{n-2}	0	1	0	1	0	1	0	1
Output	z_{n-1}								
	v								

10.10 Assume numbers are represented in 8-bit twos complement representation. Show the calculation of the following:

a. $6 + 13$ **b.** $-6 + 13$ **c.** $6 - 13$ **d.** $-6 - 13$

10.11 Find the following differences using twos complement arithmetic:

a. $\begin{array}{r} 111000 \\ -110011 \end{array}$ **b.** $\begin{array}{r} 11001100 \\ -101110 \end{array}$ **c.** $\begin{array}{r} 111100001111 \\ -110011110011 \end{array}$ **d.** $\begin{array}{r} 11000011 \\ -11101000 \end{array}$

10.12 Is the following a valid alternative definition of overflow in twos complement arithmetic?

If the exclusive-OR of the carry bits into and out of the leftmost column is 1, then there is an overflow condition. Otherwise, there is not.

10.13 Compare Figures 10.9 and 10.12. Why is the C bit not used in the latter?

10.14 Given $x = 0101$ and $y = 1010$ in twos complement notation (i.e., $x = 5, y = -6$), compute the product $p = x \times y$ with Booth's algorithm.

10.15 Use the Booth algorithm to multiply 23 (multiplicand) by 29 (multiplier), where each number is represented using 6 bits.

10.16 Prove that the multiplication of two n-digit numbers in base B gives a product of no more than $2n$ digits.

10.17 Verify the validity of the unsigned binary division algorithm of Figure 10.16 by showing the steps involved in calculating the division depicted in Figure 10.15. Use a presentation similar to that of Figure 10.17.

10.18 The twos complement integer division algorithm described in Section 10.3 is known as the restoring method because the value in the A register must be restored following unsuccessful subtraction. A slightly more complex approach, known as nonrestoring, avoids the unnecessary subtraction and addition. Propose an algorithm for this latter approach.

10.19 Under computer integer arithmetic, the quotient J/K of two integers J and K is less than or equal to the usual quotient. True or false?

10.20 Divide -145 by 13 in binary twos complement notation, using 12-bit words. Use the algorithm described in Section 10.3.

10.21 **a.** Consider a fixed-point representation using decimal digits, in which the implied radix point can be in any position (to the right of the least significant digit, to the right of the most significant digit, and so on). How many decimal digits are needed to represent the approximations of both Planck's constant (6.63×10^{-27}) and Avogadro's number (6.02×10^{23})? The implied radix point must be in the same position for both numbers.

b. Now consider a decimal floating-point format with the exponent stored in a biased representation with a bias of 50. A normalized representation is assumed. How many decimal digits are needed to represent these constants in this floating-point format?

10.22 Assume that the exponent e is constrained to lie in the range $0 \leq e \leq X$, with a bias of q, that the base is b, and that the significand is p digits in length.

a. What are the largest and smallest positive values that can be written?

b. What are the largest and smallest positive values that can be written as normalized floating-point numbers?