CMPSCI 145
FALL 2009
REVIEW SHEET FOR MIDTERM EXAM
Professor William T. Verts

This is a reminder that the exam will be in-class on Friday, November 6TH. The TA will hand out both an in-class portion worth 70% of the midterm score, and a take-home portion worth 30% of the midterm score. The take-home is due in class on Monday, November 9TH.

The exam is open-book, open notes. Anything on paper is OK to bring in. We will cover from the beginning of the semester up through integers and floating point (all binary topics) and symbolic arithmetic, but we will not cover representations of equations or any later topic. Although I do not think you will need a calculator, simple calculators are allowed. No graphing calculators, no calculators that have built-in base conversion functions, and no cell phone calculators, please. No other electronics permitted (cell phones, computers, PDAs, iPods, etc.). I will try very hard to make the exam short enough to do in the 50-minute class time, but when you get the exam please go through it once to identify the easy problems and get them out of the way quickly. If you run short on time, I do not want you to have a bunch of easy problems left unanswered.

I know that there is a lot of material here, and obviously I cannot ask detailed questions about each topic (not unless you want a ten-hour exam; yeah, me neither), but here are the high spots. I'll try to ask representative questions from each topic group and still bring it in under 50 minutes expected completion time. Feel free to email me with questions over the next few days; if there are topics I feel to be of benefit to the entire group I'll write them up and send them out in an email broadcast.

General Questions

What is the difference between an analog device and a digital device? What general characteristics of each type lead you to decide whether a particular device is analog or digital? How can you perform computations with gears, sticks, fluids, marbles, etc., and what kinds of computations can you perform? How does a change in representation (such as changing a linear scale to a logarithmic scale) alter the types of calculations that can be performed? How can you use analog functional elements (e.g., resistor voltage dividers, amplifiers, ENIAC devices, etc.) to solve some computation? What advantages do analog devices have over digital devices? What advantages do digital devices have over analog devices?

How are digital functional elements (gates) similar to and different from analog functional elements? What are the functional differences between AND, OR, NAND, NOR, XOR, and NOT gates? How can you use digital functional elements to create a circuit to solve some computation? How do punched card devices store their information? What is a bit?

How do you use 9's and 10's complement arithmetic to perform subtraction using a machine capable only of addition? How does BCD and XS3 tie in here? How do you use 1's and 2's complement binary arithmetic to perform subtraction using addition? How is 1's complement similar to 9's complement, and how is 2's complement similar to 10's complement?

How do you convert between decimal (base 10) and any of the binary forms we covered? What are the problems with each of the signed binary forms (Sign and Magnitude, 1's Complement, 2's Complement) and where on the number wheel do those problems occur? How is shifting a number left or right in binary equivalent to multiplication or division by two? What did you learn from the Z99 project that leads to an understanding of the differences between unsigned and signed representations?

What are the advantages and disadvantages of the BCD and XS3 forms? What kinds of problems do those representations attempt to solve? How do you perform an addition in BCD or XS3? What advantages does the Gray Code binary representation have over regular binary? When can Gray code be used to its advantage?

How do you convert a decimal (base 10) number with a fraction into rational, true binary, binary scientific notation, and then into quarter, half, single or double precision? How is binary scientific notation similar to and different from traditional decimal scientific notation? How do you convert a quarter-precision, half-precision, single-precision or double-precision floating-point number into decimal? What is a Denormal and why do we use them? How do we represent Infinity? What happens when we add 1 to Infinity? What is a NaN and how are the two types of NaN used? Why do we drop the leading 1 bit in the mantissa, except for Denormals? What are the common binary patterns for zero, the smallest non-zero Denormal, the largest Denormal, the smallest normalized number, and the largest normalized number for any of the listed precisions? Why are exponents biased? What are the biases for different widths of exponents? Can we have positive and negative zero in floating point? What happens when we attempt to store a number which is irrational or a repeating rational in binary (such as the binary value of one-tenth) into a floating-point number format? What did you learn from the Z99 project that leads to an understanding of fixed-point formats?

How do we use symbolic computation to avoid many of the problems with floating-point arithmetic? How is a rule such as "Replace (X + 0) with X" or "Replace (X / X) with 1" or "Replace Cos(π / 2) with 0" applied? How do these rules help us with the limitations of floating-point? How would you resolve the conflict between the two rules "Replace (X ^ 0) with 1" and "Replace (0 ^ X) with 0" when X=0? How do you develop new rules to include?

Appropriate Chapters

Here are the chapters from the book that you should study for the exam:

```
Chapter 1 – Introduction
Chapter 2 - Analog Mechanical Techniques
Chapter 3 - Digital Mechanical Techniques
Chapter 4 - Electrical and Electronic Devices
Chapter 5 - Integer Representations
Chapter 6 - Real Number Representations
Chapter 7 – Symbolic Computation
```