Final Exam Study guide

CS 350: Computer Organization and Assembler Language Programming

Final Exam is 8:00 - 10:00 am, Tue, May 7, 2019

You can bring one sheet (both sides) of to the exam notes. The last two pages of this handout include a MIPS instruction/register reference. I'll include a copy of them with the exam (but feel free to add any information you want to your own notes). Be sure to include labs in your studies.

Midterm Exam Material (25% of exam, ± 5%)
• Be sure your review includes
  • Great ideas in computer architecture
  • C code to manipulate array elements to/from MIPS code
  • C code with labels to/from MIPS code
  • C code with decisions and loops to/from MIPS code
  • Properties of sign-magnitude and 2's complement
  • Structure of the runtime stack

Final Exam Material (roughly 75% of exam, ± 5%)
• C Pointers, Arrays, Strings — see activity questions for lectures 9, 10

• Translation, Linking
  • What are object modules?
  • What do object modules contain?
  • How does linking create executable files?

• Arithmetic (see Ch. 3 and especially Lab 6)
  • Hexadecimal numbers can represent bitstrings.
  • Be able to translate floating-point numbers between formats
  • E.g., 35 \frac{17}{32}, 35.0.53125, 100011.10001, 1.0001110001 \times 2^4, and IEEE format
  • For integers, what is overflow?
  • For floating-point numbers, how are they represented? What is overflow? Underflow? (see pp. 197-198)
  • Be able to show how addition of floating-point numbers can cause loss of precision (see p. 203-204)

• Processor Design (see Ch. 4 and esp. activity questions for Lecture 14)

For the following sections, see the activity questions for the lectures
• Structures and Unions in C (see Lab 7 and activity questions for Lecture 15).
• Pipelining (Lecture 16)
• Memory (Thu 5/2: Questions have been added to Lectures 17 and 18)
• Parallel Processors (Fri 5/3: Questions for Lectures 19 and 20 have been added to Lecture 20.)
• Switches, Transistors (Lecture 21)
• Logic Gates (Lecture 22)
• Combinatorial Circuits (Lecture 23)
• Storage elements and Memory (Lecture 24)
  • Make sure to be prepared to trace what happens to an R-S latch under various conditions

• Sequential Logic, clocks (skip finite state machines)
• What can go wrong if we don’t use a clock?
• With a master-slave flip-flop
  • When the clock is zero, data from the combinatorial circuit is read into the right-hand latch but
    doesn't affect the left-hand latch (and therefore doesn't change the output to the
    combinatorial circuit).
  • When the clock is one, the new value is written into the left-hand latch (and goes out to the
    combinatorial circuit), but any change in data from the combinatorial circuit to the right-hand
    latch is ignored.

C programming — be able to read / write code to
• Read/write integers in decimal and hex formats.
• Create/use masks to do bitwise operations on numbers.: Retrieve, zero out, and set fields of
  numbers; shift left and right.
  • Note left shift corresponds to multiplication by powers of 2.  On unsigned and nonnegative
    numbers, right shift corresponds to division by powers of 2.
• Pointers, Arrays, Structures
  • Draw a memory diagram describing the situation after evaluating some given C code.
  • Given a memory diagram, write code that declares variables and sets values for them to match
    the diagram.
• In addition to the relevant lectures, don't forget to study the code for Labs 5, 6, 7.