Lecture 26: Review for the Exam

• What you need to bring
• What I will provide (if needed)
• Review of Topics

Exam

• Closed book
• Closed notes (except for “cheat sheet”)
• NO CALCULATORS

You Need to Bring

• An 8 ½ by 11” sheet of paper with notes.
  – both sides
  – hand written

What I will Provide

• IF NEEDED
  – ASCII table giving Hex codes
  – The Microprogram portion of the microprogram handout
How Should I Study?

- Work out the sample problems from the web (I’ll go over some today).
- Look at your homework assignments. If you didn’t get something right, figure out why!
- Make sure you understand the more recent material (microprogramming!) that you haven’t had much chance to practice with yet.
- Go through your lecture notes. You’ll have to do this to prepare your cheat sheet.
- Read the books: Irvine, Tannenbaum (on reserve), AoA! You’ll also need information from here for your cheat sheet.

Homework and Exams

- There may be questions about the homework assignments on the exam.
- If you chose to not do a homework assignment you are still responsible for knowing how to do it!
- Review the homework, especially if you did not do one of them or if you did one using the wrong method.
  - the exam may ask about ascii to binary conversion
  - the exam may ask about using the stack for reverse polish notation
  - etc.

Test Taking Hints

- **SHOW YOUR WORK.** This is needed if you want any partial credit for an answer.
- Skim the entire exam before starting to figure out which problems you should do first (start with the easy ones!)
- Read the problems carefully. Make sure you understand what you are expected to do.
- **If you don’t understand a question, ask the instructor or a TA**
- Keep track of time
- Be aware of problem point values!

Review of Topics

- Disclaimer: Everything we have covered in class is fair game for showing up on the exam unless I tell you otherwise.
- This review will touch on some points that people have had trouble with so far.
Review of Topics

• The exam will focus on the material covered since the midterm but this does not mean you won’t need to know the earlier material!!
  – you need to know ALL the covered assembly instructions to understand and write code on the exam.
  – this code will probably use some of the addressing modes.
  – you may need to do some hex arithmetic and hex-binary conversions.
  – you’ll need to know binary to create truth tables.

Stack

• Push
• Pop
• Relationship between SS, SP, BP
• What happens to SS, SP, BP when pushing and popping.

Subroutines and Parameter Passing

• How to set up a subroutine.
• Call and Ret
• Stack Frames
• Accessing parameters passed on the stack
• Cleaning up parameters
• Recursion

Linking and Loading/ C and C++ interface

• How the assembly, linking and loading process works.
• What you have to do in the assembly language to write procedures that can be linked as separate files (public, extern...)
• C/Assembly interface (16 bit registers)
• C++/Assembly interface (32 bit registers)
MUL/DIV

- Know how the signed and unsigned multiply and divide instructions work.
  - which registers can be used as input.
  - how/where the output is stored.

Boolean Algebra

- Know the boolean functions:
  - AND, OR, XOR, NOT, NAND, NOR
- Know how to draw a truth table.
- Know how to use a truth table to prove an equality.
- Know how to use the truth table to create the sum-of-products.

Boolean Instructions

- Know what they are!
- Know which ones don’t exist (NAND and NOR are not valid instructions in the Intel instruction set)
- Know how they are used:
  - AND for masking,
  - OR for setting some bits, leaving others unchanged,
  - etc…

Equivalent Circuits

- Know how to draw the basic logic gates (AND, OR, NOT, XOR, NAND, NOR).
- Know how to draw and use Karnaugh maps to simplify a circuit.
- Be able to draw your circuit.
Combinational Circuits

- Know all the combinational circuits described in class and on the homework
  - What they do
  - Potential uses
- Be able to draw a block diagram or a truth table if asked.

Sequential Circuits

- Know and understand all the sequential circuits described in class (SR, JK, D)
- Know the difference between a flip-flop and a latch.
- Be able to draw characteristic tables, excitation tables, and timing diagrams for those or other circuits (for circuits other than the SR, JK and D a diagram would be given).
- Understand how memory works and what the different types are.

Microprogramming

- Fetch-Execute Cycle
- Instruction decoding (types, advantages/disadvantages)
- Vertical vs Horizontal Microinstructions
- Control unit – what it does, basic components
- ALU – what it does, basic components
- Microprogramming vs. Hardwired control

Microprogramming, cont.

- How the microprogram interprets machine code.
- How to encode a MAL instruction into microcode (binary).
- How to decode the microcode into MAL instructions.