Study Guide for Final Exam
CSCI 2340 Assembly

Exam Rules

- This is a closed book exam.
- No reference materials.

Material to Study

From Exam 1 (Through Lecture 7)

- Memory vs registers
- Where are data and program instructions stored
- Size of memory addresses in MIPS
- Size of word
- Number of bits in a byte
- Which is faster to access - register or memory?
- How to convert C statements to MIPS
- How to convert MIPS statements to C
- How MIPS add, addi, sub, sll, srl, and li work
- How to load values into registers (lw)
- How to store values into memory (sw)]

From Exam 2 (Lectures 8-16)

- Branch and jump instructions
    - Conditional branches: beq, bne, bgt,blt
    - Unconditional branch and jump: b, j
    - Jump and Link and Jump Return: jal, jr
- How to read/write/interpret a conditional (if) statement in MIPS
- How to read/write/interpret a for loop in MIPS
- How to read/write/interpret a while loop in MIPS
- How to write a procedure and return from it
- Be able to determine the output of a MIPS program.

New Material (Lectures 17-23)

- Static arrays – be able to translate C/pseudo code to MIPS code (examples in slides: **g = h + A[8]; A[10] = h + A[8]; B [3] = A [i - j];)**
- Byte instructions (lb, sb)
- Byte arrays vs integer array
    - Know how to traverse arrays
    - Know how to load values from arrays
    - Know how to store values back into arrays
- Dynamic memory allocation – arrays and structures
    - Describe what dynamic memory allocation is
    - Understand how to access fields in a structure in MIPS (that were allocated dynamically)

- Binary fractions and floating point – IEEE 754 format for single precision floating point
    - Mantissa (23 bits)
    - Exponent (biased integer) (8 bits)
    - Sign bit (1 bit)
- If I have a value in a register in the CPU and need to do floating point, what are the steps to prepare the value for floating point operations?
- Floating point instructions in FPU/Coprocessor1
    - lwc1,swc1, mul.s, div.s, sub.s, add.s
    - mtc1, mfc1
- Floating point conditions and branching
    - What is the condition bit and how is it used?
- How to call procedures in MIPS (jal), how to pass floating point arguments ($f12-$f18) and return values ($f0-$f1), also how to return (jr). **Don't forget label to start the procedure.**
- Double precision floating point vs single precision.
- Exceptions and traps
    - Name the registers used in coprocessor 0
    - Instructions: mtc0, mfc0
    - Handled in kernel mode
- Jump tables in MIPS can be used to implement switch statements from HLL.

Ideas for studying

- Look over lectures and make sure you understand material – especially in areas called out in this study guide
- Can you translate from pseudo code/C to MIPS and back? If presented with a MIPS program with control structures in it, can you describe what the program is doing (algorithm) and what the output is?
- Try writing procedures that accept floating point parameters and do floating point arithmetic. Make sure you can call them.
- Try translating floating point code into pseudo code/C and vice versa.