Midterm Review

Format

- 10 multiple choice questions – 8 points each
- Typically have small code segment and you have to specify what the value of a register is after execution
- You can write a sentence to the side describing why you feel your answer is right and you may get partial credit depending on your answer
- Another good idea is to write comments next to each line of code. This will help you get the answer right and help me see your train of thought

Format

- 2 short answer – 10 points each
- 1 of these will be a longer equivalent to the multiple choice questions. That is, you’ll have a full program to parse through and you’ll need to say what the final output is
- 1 of these will be writing a short code segment based off a small passage of text. Typically the problem involves bit manipulation, so know your logic instructions well. You’ll have a code line limit (typically between 5 and 10 lines) and in going over, you’ll be deducted two points per extra line. Psudo-instructions will be expanded as indicated a few slides down
- Again, by writing a sentence or two describing your intent can potentially net you partial credit if you don’t get the question right

Others

- There will be a handout given (1st page excerpt from the book) to be used for reference on the exam
- Calculators will not be allowed (and should not be needed)
- The exam will be 75 mins in length. I can give you a few minutes over that if necessary, but there is another class after ours and I can’t hold them up.
- Try not to spend too much time on any one question (a good estimate is 4 and a half minutes for each multiple choice and 15 for each short answer) and to move on if you start to get stuck
- If it appears that the majority of the class doesn’t finish, I will scale the exam accordingly. However, it’s still in your best interest to try and answer all the questions

What to Know: Numbers

- Binary->decimal
- Decimal -> binary

- Unsigned numbers
- Signed numbers
- Floating point numbers

What to Know: Basic MIPS

- How to write instructions
- How to do arithmetic / logic operations
- How memory is represented and how to load/store values from/to memory
- How branching works and how to implement if statements and loops

What to Know: More MIPS

- What instructions are pseudo-instructions and how many physical instructions they map to
  - blt/ble/bgt/bge ->slt, beq / slt, bne
  - li/la -> ori / lui / ori, lui (depending on value passed in)
- How to implement functions
- How to use the stack
- What the basics of interrupt handlers are
  - How to return to main
  - What registers you can use
  - How to use other registers

What will NOT be on the exam

- Interrupt programming
- Memory mapping
- Syscalls
- Recursion
- Floating point programming
- Sorting
- Encoding
- Characters / strings