CS 33:
Introduction to Computer Organization

Week 3

Today's Agenda

- Midterm
- Note: The midterm questions will not be posted online. The common errors will be, however.

Question 1: Common Errors

- "+1" instead of "+x" in some questions (half credit)
- The correct answer but used the opposite shift operator (i.e. used << instead of >>, or vice versa). Shifting left multiplies by powers of 2. Shifting right divides by powers of 2 (sort of, see Q9 for more).
- Shift by powers of two instead of the exponent (i.e. "x<<32" instead of "x<<5", or "x<<8" instead of "x<<3")

Question 2: Common Errors

- Many people were off by just one power of two. (i.e. 2^-15 vs. 2^-14)
- Some people wrote the binary interpretation. This garnered half credit if the binary was correct.
- Last part of prob 2, some did not realize this was NaN. Some thought it was a normalized number, some thought it was infinity.
- Many people forgot about the "assumed 1" that we must place back in front of the mantissa.

Question 3: Common Errors

- Many people thought this one had something to do with endianness.
- Endianness is only an issue when examining how a multi-byte variable is stored in memory.
- When you are dealing with a variable in C/C++, the way that the value is stored in memory IS subject to endianness, but when you operate on the variable itself, it formatted in the correct order.

Question 4: Common Errors

- Some students correctly determined when the destination did not have the granularity to represent the right value, but rounded the wrong way. We asked you to round up.
- For example:
  - -2^13 is converted to the negative non-infinity number with the greatest magnitude, not negative infinity.
  - 2^-17 is converted to positive non-zero number with the least magnitude, not zero.

Question 5: Common Errors

- A signed value that is extended will be sign extended, regardless of whether the destination type is signed. Similarly, an unsigned value that is extended will always be zero extended.

  IE. UC: 1101 0011 => SS: 0000 0000 1101 0011

  SC: 1001 0001 => US: 1111 1111 1001 0001

- However, despite the fact that 1111 1111 1001 0001 was a result of sign extension, because the destination type is an unsigned short, the value is 65425, not -111.
- The C comparisons are 32 bits, which means that both operands will be extended before comparison.

Question 7: Common Errors

- Operator Precedence Order:

  ~, >>/<<, &, ^, |

- Not knowing how to set up a truth table easily (counting up in binary). Because of the order, some people missed a row.

Question 8: Common Errors:

- An immediate ($0x22) has no address.
- %al is the least significant bit of %rax. If %rax is 0x00000133, %al is 0x33
- Incorrect hex addition.
- Memory to memory is an illegal operation.

Question 9: Common Errors

- (a << 4) + (a << 2) + (a << 1) is a(2^4 + 2^2 + 2^1), not a*7.
- Matched a/4 with a >> 2. This is not accurate for negative values.
- Shifting right by 2 divides by 2^2, but it always rounds down because the two least significant bits are truncated.
- Integer division should round to zero, which means that dividing positive numbers rounds down.
  - 13 / 4 = 3.25, approximate down to 3.
- However, dividing negative values rounds up
  - -13 / 4 = - 3.25, approximate up to -3.
- Because shifting right always rounds down, the negative result for – 13 / 4 is -4, the wrong answer. So if negative, add 3 to offset this.

Question 10: Common Errors

- %rip does not stand for index pointer, it stands for instruction pointer.
- %rip, %rsp, and %rbp are registers for 64-bit machines while %eip, %esp, and %ebp are registers for a 32-bit machine. If push applies to %esp, it decrements by 4 but if it applies to %rsp, it ought to decrement by 8 (this did not usually incur a penalty).

End of

The Third Week

-Seven Weeks Remain-