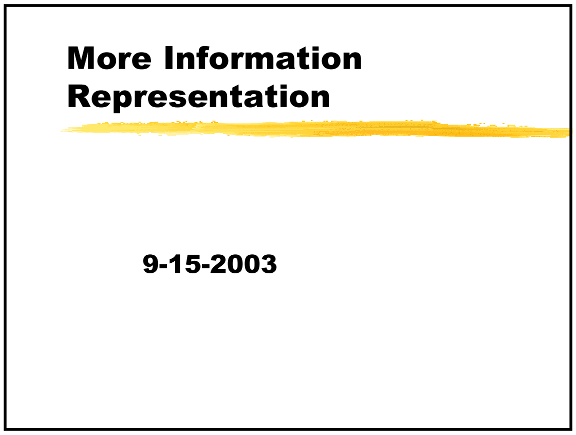
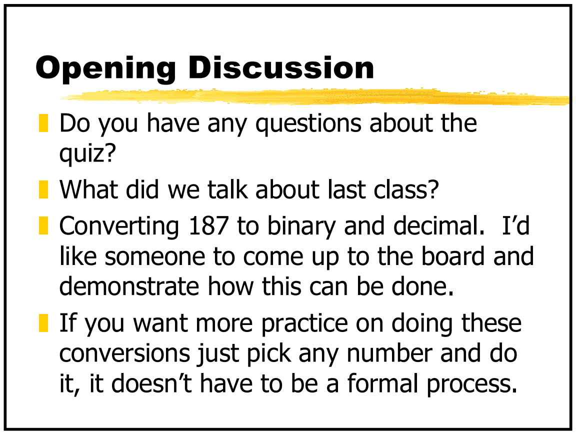


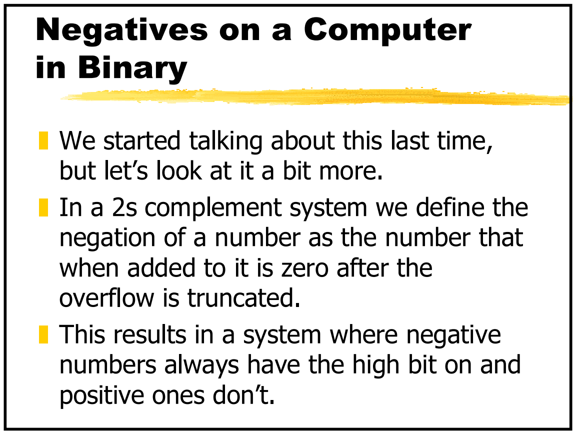
More Information Representation

9-15-2003



Opening Discussion

- Do you have any questions about the quiz?
- What did we talk about last class?
- Converting 187 to binary and decimal. I'd like someone to come up to the board and demonstrate how this can be done.
- If you want more practice on doing these conversions just pick any number and do it, it doesn't have to be a formal process.



Negatives on a Computer in Binary

- We started talking about this last time, but let's look at it a bit more.
- In a 2s complement system we define the negation of a number as the number that when added to it is zero after the overflow is truncated.
- This results in a system where negative numbers always have the high bit on and positive ones don't.

Other Integer Operations

- In addition to addition there are a few operations that we like to do with numbers, and in particular binary format numbers. We are going to look at three groups of these.
  - Multiplication
  - Bit Shifting
  - Bitwise Logic

Multiplication - The Slow Way

- By definition, multiplication is repeated addition so we could just define it that way in the computer.
- This is less than ideal because that is a very slow way of doing things when the numbers we are multiplying get large.
- For fun though we should look at some special cases. Maybe multiplying things by 2 or 4.

Bit Shifting

- As it turns out, multiplying by a power of 2 is the same as "shifting" the bits of the number up by whatever power of two we are multiplying by. Same as *10 in decimal.
- In C there are operators that do bit shifting. The "<<" operator shirts the first argument a number of positions equal to the second argument.
- Downshifting (>>) is integer division by powers of 1.

Faster Multiplication

- With knowledge of bit shifting and the properties of multiplication, we can do faster multiplication. Same as long multiplication in decimal.
- Consider that n*5=n*4+n*1=(n<<2)+n.
- By using a binary representation we can multiply by any number and the number of times we have to add is one less than then number of 1 bits in the representation of that number.

Bitwise Logic

- We will talk about Boolean logic later at which time we will revisit this, but for now we can say that there are other operations you can do with numbers in binary representations.
- We can apply the logical operations of or, and, xor, and not to the bits of binary numbers to aid in certain calculations.

Character Representation

- C has a data type called char. As it turns out, a char is a byte on most machines. The char type is used in C to represent both a number and a character.

Floating Point Numbers

- Your book calls these float numbers which matches C-syntax, but isn't quite correct. They are called this because the decimal point is allowed to "float" around.
- Floating point numbers are represented in memory in a manner much like scientific notation. They have 3 parts, a sign bit, a mantissa, and an exponent.
- Of course, the exponent is a power of two.

Minute Essay

- Next class we will start looking at programming in C and we will explore some of the concepts that we have talked about in the last two classes in programs. What questions do you have about the topics of the last two days?
- Please reads chapters 1 and 2 of the text for Wednesday.