EE334

Binary Numbers and Arithmetic

Most of you know that computers and many modern gadgets (calculators, cell phones, MP3 players, cameras, etc) use a thing called “binary”. But what is binary?

1. Binary is…

2. In electronic devices, binary is really…

Even though devices that use binary use only two values, we can combine those two values (make groups) that we then give meaning to. In other words, **the usefulness of binary comes from assigning meaning to the binary values or groups of binary digits.** The meaning is whatever we want it to be. In fact, we can create entire languages from binary just like we use our alphabet to form words for English. Thankfully, there are some commonly understood interpretations for binary digits and groupings of digits. For example, binary where one digit is a “0” (zero) and the other digit a “1” (one) is probably the most common form of binary digits because of its mathematical usefulness (as we’ll see later). Let’s expand your mind about assigning meanings to binary digits this concept with some problems.

3. Below are two rows. One row starts with 0 and the other with a 1. Write down pairs of meanings for the binary digits. One has already been done for you.

Digit	Possible Meanings							
0	Off							
1	On							

4. Now we’ll consider pairs of binary digits. Fill in the table below with possible interpretations of the pairs. Again, a first one has been done for you.

Digits	Possible Meanings							
00	0							
01	1							
10	2							
11	3							

As we saw on the previous page, groups of binary digits can be used to represent almost anything. One of the more common uses of binary digits is for representing numbers (counting). When representing numbers, we'll use the term **binary numbers**. Not only can we count in binary, but we can also do math (next lesson). Let's explore binary numbers now. We'll start by looking at the numbers we're familiar with first: **the decimal or base-10 system**.

5. How many number symbols do we have in our base-10 number system? List them.

We give meaning to base-10 numbers by associating a power of 10 to each number's position. You probably remember a scale similar to the one below from your elementary school years.

$$\ldots\ 10^4\ 10^3\ 10^2\ 10^1\ 10^0\ \ 10^{-1}\ 10^{-2}\ 10^{-3}\ \ldots$$

Numbers like 45,027.139 are readily understood by use because we're so familiar with the base-10 number system. We don't usually think about the meaning of the number itself. However, recall that each digit in a number has a value that is the digit multiplied by a power of 10. For example, the digit "4" in 45,027.139 is worth 4 times 10^4. The digit "3" is worth 3 times 10^{-2}. The value of the whole number comes from multiplying each digit by its place value (power of 10) and then adding each digit's value together:

$$4x10^4 + 5x10^3 + 0x10^2 + 2x10^1 + 7x10^0 + 1x10^{-1} + 3x10^{-2} + 9x10^{-3}$$

Now here's the leap: binary numbers work just the same as base-10 numbers except that binary numbers are base-2.

6. Below each entry in the base-2 scale, write the corresponding base-10 value.

$$\ldots \quad 2^4 \quad 2^3 \quad 2^2 \quad 2^1 \quad 2^0 \quad 2^{-1} \quad 2^{-2} \quad 2^{-3} \quad \ldots$$

7. Write the base-10 version of the following binary numbers.

    a. 10

    b. 1010

    c. 0010

    d. 1110.1

    e. 111.0010

    f. 10011101

8. Convert the following base-10 numbers into binary numbers.

   a. 0

   b. 2

   c. 8

   d. 10

   e. 15

   f. 47

   g. 100

9. Use the remainder method to convert the numbers from problems 8f and 8g into base-10 (decimal) numbers.

   a. 47

   b. 100

<u>Adding Binary Numbers</u>

Adding two binary numbers together works exactly like adding two decimal numbers, except that **you carry to the next higher place when sum of two binary digits is greater than 2**.

1. Add the following binary numbers.

0 <u>+0</u>	0 <u>+1</u>	1 <u>+0</u>	1 <u>+1</u>
01 <u>+10</u>	10 <u>+1</u>	11 <u>+10</u>	11 <u>+11</u>
1001 <u>+1011</u>		101100 <u>+110111</u>	

2. A binary operation (like addition) that results in a number that has more digits than either of its operands is said to have ___________________. Using pencil and paper, we can handle overflows by adding more digits. However, for computers the number of digits available for representing numbers is limited and overflows can be a problem. To see this, look at the last sum in problem #1. What would the answer have been if the space for the answer was limited to six digits?

Negative Binary Numbers

So far all the binary numbers we've seen have been non-negative (zero or greater). It is possible to have negative binary numbers. The decimal system uses a special symbol called the minus sign, –, to show that a number is less than zero. There is no special symbol that indicates that a binary number is negative. Instead, we use the digits themselves to show that a binary number is negative.

When using signed binary numbers, we use the left-most **bit** (digit) of a binary number to indicate whether or not a number is negative or positive. If the left-most bit is a "1", the number is negative. Otherwise, the number is positive.

3. Indicate whether or not the signed binary numbers below are negative or positive.

   11111111　　　01011110　　　10000000　　　00000011

4. We can sign extend signed binary numbers to as many digits as necessary (for doing math ops for example) by simply replicating the left-most bit as many times as is needed. Sign extend the following binary numbers to a length of 8 bits.

   1111　　　1110　　　011　　　000011

5. Is 1111 = 11111111? Why?

CAUTION: Like everything else in "Binary Land", the notion of signed and unsigned binary numbers is a concept – one that can get you into trouble if you're not careful. For example, if I give you two binary numbers, say 1101 and 1101, would you say that they have the same value? They do have the same value if I tell you that both numbers are unsigned (non-negative). They also have the same value if I tell you that they are signed numbers. They are not the same, however, if the first 1101 is unsigned and the second one is signed.

6. How do you know if a binary number is signed or unsigned?

Changing the Sign of a Binary Number

There are at least two ways to invert the sign of a binary number. One method, called "1's complement", is rather laborious. We will not deal with it. The other method, "2's complement" is quick and easy. Let's look at the 2's complement method.

To change the sign of a binary number using 2's complement, do the following.

a. Starting at the right side of the number and going left, copy down each digit until you've copied down the first "1" digit.
b. Invert each remaining digit of the original number.
c. Sign extend the number as necessary to achieve the desired length.

7. Invert the sign of each of the following signed binary numbers. Sign extend each number to 10 digits. Next to each binary number, write its value in decimal.

   100 0001 01011 10001

8. Quick! What's the value of 110 in decimal (base-10)?

> WARNING: Negative binary numbers don't always look like their positive counter parts. For example, -5 and 5 don't look alike in binary: 11011 ⇔ 00101. However, -8 and 8 do except for the sign bit (left-most bit): 11000 ⇔ 01000.

Subtraction

Subtraction of one binary number from another suffers from the same problem as subtracting two decimal numbers: borrowing takes a lot of thought. In computers, thought equals time. Since we want our computers to be fast, we want their operations to be simple. It turns out the easiest way for both humans and computers to do binary subtraction of the form X – Y = Z is to do X + (-Y) = Z. Translation, **invert the sign of the number being subtracted then add**. Doing a sign inversion followed by addition eliminates the need to borrow and is actually faster than trying to do subtraction (2 fast operations can take less time than one long one).

9. Solve the following problems using binary math. Check your work by converting each binary number to its decimal equivalent and then do the math in decimal. All the binary numbers are signed.

0111	0001	1011	1011
-0100	-1110	-1101	-0011

10. Did you notice anything weird or wrong about the right-most difference in problem #9? Hint: Look at problem #2. Is your answer correct if limited to four bits? Is your answer correct if five bits are used?