Name ____________
Date _____________

Introduction to Binary Numbers

Overview

Most computers use a series of parallel data lines, called computer buses, to send data between the CPU and adapter cards, including the network. The majority of the earliest generations PCs used an 8-bit bus, which means they used 8 lines for data in parallel and could move 8 bits worth of data in a single bus transfer. At the basis of the 8-bit line of data is the binary number system. Today we'll talk about number systems, how binary systems work, and what binary numbers represent in base-10 numbers.

Materials

- ☒ This worksheet/handout
- ☒ Calculator (if you choose to use one)
- ☒ Pencil

Introduction

The number system we are most familiar with is the base-10 number system. This numbering system is also referred to as Arabic numerals or the Arabic numbering system. The "base-10" means that there are 10 significant digits used in representing numbers in this method of calculation. So, in other words, 10 is the "base" for this numbering system. For example, we count from 0 to 9, then a digit is created in the tens place so we know that we have looped over the maximum value for the ones place. It would be very difficult to get any meaning if our method of counting would go like this:

1, 2, 3, 4, 5, 6, 7, 8, 9, 0, 1, 2, 3, 4, 5, 6, . . .

because 11 would be represented by a 1, which immediately causes confusion no matter what type of calculation we are working with. This is why we use a place holder in the tens spot. This lets anyone know that we have been through the one values one time and the ones digit now represents that we have that many MORE ones towards our next advancement in the tens place. For example, 24 would mean that we have cycled through the ones digits twice and have 4 more ones towards another step in the tens place.

You may have also worked with Roman numerals previously. This lets symbols represent numbers in a different way. It works more like a base-5 number system. In the Roman numeral system, I means "1", V means "5", X means "10", L means "50", C means "100", D means "500", and M means "1000". We can combine values and use their order to make relevant numbers in this system. For example, 11 would be represented in Roman numerals as XI. But on the other hand, 34 would be represented as XXXIV, since 4 is one number before 5.

Lesson

Binary numbers are similar to Arabic numbers, except binary numbers use a base of two. Just like in Arabic numerals, where the base is 10, but 10 is not reached until another digit is added. Hence, in base-2 numbers, we will carry to the next units place every time a two is encountered. Think of it like carrying a one when adding numbers. The way we look at place values for base-10 looks like this:

10,000,000	**1,000,000**	**100,000**	**10,000**	**1,000**	**100**	**10**	**1**

So each digit of our number holds a special value. That is, a number in the 10's place means there are that many tens in the value of the number. Base-2 numbers use this same premise, but rather than go by powers of 10, they go by powers of 2. So the place values would look something like this:

1024	**512**	**256**	**128**	**64**	**32**	**16**	**8**	**4**	**2**	**1**

So, by this chart, the base-2 number "100" would actually represent 4 in a base-10 system. This is because there are zero 2's, zero 1', but one 4. Look at a few other examples:

Convert the base-2 number 1011 to base-10.

<u>Solution:</u> A 1 in the 1's spot means we have one 1. 1
A 1 in the 2's spot means we have one 2. 2
A 0 in the 4's spot means we have zero 4's. 0
A 1 in the 8's spot means we have one 8. <u>+8</u>
11

The base-2 number 1011 is the same as 11 in base-10.

Convert the base-10 number 43 to a base-2 numeral.

<u>Solution:</u> Working out in this manner deals a lot with using remainders. The closest we can get to 43 as a single power of 2 is 32. So our base-2 number will start out looking like 100000 to hold a value of 32 to start out with. But, since 43-32=11, we still have 11 more until we

get our value of 43. Putting a 1 in the 16's place would go over 43, so we must leave a zero in that place. We can add one 8 to our base-2 number, which would give us 40 thus far. So our base-2 number now looks like 101000, which is equivalent to 40. To get three more, we can add a 1 in the 2's place and add a 1 in the 1's place to finish out our 43. *Thus, we can display the base-10 number 43 in base-2 as 101011.*

Try some examples on your own.

a. Convert the base-2 number 1011011 to base-10.

b. Convert the base-10 number 432 to base 2.

That concludes the information on base-2 numbering systems.

Question and Answer Time

Are there any questions you have at this time?

How can we add numbers in base-2? Does it involve carrying?

Exercises

Add these binary (base-2) numbers. It may be necessary to carry.

1. 10100+11
2. 11111+101

Convert these base-10 number to binary.

3. 17,376
4. 576
5. 98

Convert these base-2 numbers to base-10 numbers.

6. 110110010
7. 1111111
8. 1001001001

Critical Thinking: If we converted the base-2 number 11001100110011 to base-10, how can we easily tell if it is an odd or even number?