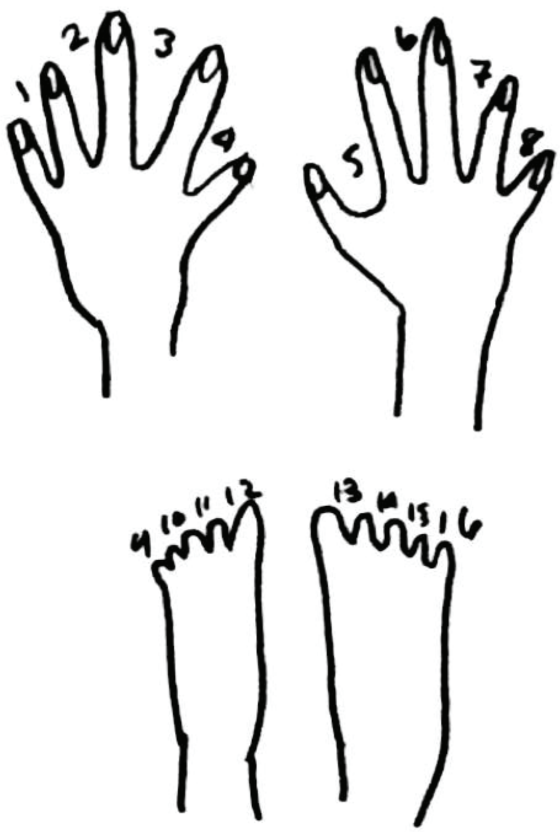
chapter 2:
binary numbers

So, lots of ancient counting systems revolved around 10 and multiples of 10. That is, they either had special symbols for 10, 100, 1000, 10000, etc. (for example, Egypt) or, they started repeating symbols after the first 10. It makes sense that 10 is significant in many counting systems, since we have 10 fingers which are very useful for counting, and fingers were the inspiration for most counting systems.

excerpt from BubbleSort Zines

But counting systems can have any number as their important/special number. There were some ancient counting systems that used 16 as a significant number, and these counting systems repeated their symbols after the first 16.

Why was 16 special to them? Instead of counting things on fingers, this culture counted the spaces between their fingers and toes!



excerpt from BubbleSort Zines

How would that even work to have a counting system based on 16? How would we count?

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

that's the same so far, but what's next?

We'd want new symbols to represent 10, 11, 12, 13, 14, and 15 but with a single symbol instead of two symbols. We'll use letters to represent them, but we could use anything else, really.

0, 1, 2, 3, 4, 5, 6, 7, 8, 9, a, b, c, d, e, f...

and then what?

Now it is finally time to repeat!

so

0, 1, 2, 3, 4, 5, 6, 7, 8, 9, a, b, c, d, e, f, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 1a, 1b, 1c, 1d, 1e, 1f, 20...

So! A number system can have any number as its special and important number. We call this special number of a counting system its base. The Arabic counting system that we use now has base 10.

Computers use base 2. If base 10 means that you count from 0 to 9 and then start repeating those ten numbers, base 2 means you count from 0 to 1 before repeating just those two numbers. You might be thinking— I see all kinds of numbers when I use a computer, not just 1's and 0's. The computer shows us base 10 numbers like we are used to, but internally, when it is thinking to itself, it uses all 1's and 0's!

Can we really represent all numbers with just 1's and 0's?

It's just like any other base: when counting in base 10, we start repeating digits after 9:

0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11...

In base 10, We start repeating digits with the front digit representing how many tens we have: 30 is three tens, 53 is five tens and an additional three.

We do the same for base 2 except we start repeating digits much sooner:

0, 1, 10, 11, 100, 101, 110, 111
0, 1, 2, 3, 4, 5, 6, 7

This is reminiscent of the tally marks from the last chapter. When you have so few symbols, you have to repeat them frequently for large numbers and this takes up lots of space. It also takes longer to read.

Why would computers even use binary numbers, with all these drawbacks? It seems like a step backwards.

Computers didn't always use binary numbers. Earlier computers used base 10 numbers just like we count with.

Old computers were mechanical or pneumatic (operating on air pressure) but even when computers had already become electronic, they represented base 10 numbers with voltages (amount of electrical charge).

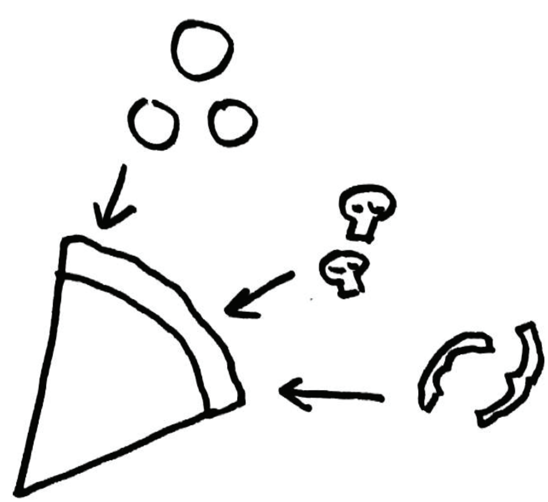
So, if you were adding the numbers 1 and 5 together with this type of computer, you might have 1 input wire with a voltage of 1 millivolt (millivolt is to volt as millimeter is to meter) and another with voltage of 5 millivolts.

But electronic parts can be kind of flaky. Instead of 1 millivolt, that wire might have 1.1 millivolts, and instead of 5, the other wire might have 5.1 millivolts. And the more unreliable numbers you add together, the more error you have as it accumulates.

It's much more reliable to check if a wire has any voltage at all. Even if the voltage were .1 mV or .5 mV different from what we expected, it doesn't matter. Everything that has voltage greater than 0, we consider having value 1. Everything that has 0 voltage, we consider having value 0.

Here is a similar situation. You and your friend are at the fair. It's lunch time and you're in line for pizza, and she's in line for lemonades, to save time. You realize that you haven't asked her what kind of toppings she wants. It's loud and crowded around both of you. You finally manage to get her attention. You try to yell out "What kind of toppings do you want?" She can't hear your or tell what you're saying. She mouths "What? Say it again?" You try again, and she mouths something you can't understand. You get an idea. You point at the poster of toppings.

You point at the big sign of toppings. You point at pep-peroni. You indicate to her that she can nod or shake her head for yes or no. She nods at the pepperoni. You point at the mushroom. She shakes her head. You point at the peppers. She nods.



excerpt from BubbleSort Zines

This situation is similar to the previous state of computers in that both are error prone. In error prone situations, it's more reliable to have fewer and simpler options for questions and answers (or inputs and outputs). And this is why modern computers only use 1's and 0's!