Advanced Learning

Tech Talks

Where you go in life depends on what you know. These tech-talks are designed to get you down the road. The point here is not to be the expert in the field but be savvy about technology in general so you can think about technical issues and be able to deal with the modern world.

You could go out into the world and start making things, bust through walls, and grab what comes along and force it together into some product. That's what junkyards are full of. Wouldn't it be better to be aware of what tools and materials are available and craft a product or idea.

Let's begin with some math. How do we count and why is this important to the high-tech world? Counting is a given, too simple. Nope. You were never taught in

The office of the future is where you can gain knowledge the quickest and build your personal brand elementary school how counting actually works, because it is really a rather complex concept. We use the decimal system by counting our fingers up to ten, and then adding a zero for larger counts. We are sophisticated individuals now, so let's get stuck in the facts of the matter.

Counting involves a digit times a power of a base which are all added up towards infinity. Sure. What's that mean? Consider the following:

COUNT = digit X 10^n... + digit X 10^3 + digit X 10^2 + digit X 10^1 + digit X 10^0

Where the digit is 0 to 9. By the way, 10^0 is equal to 1. So our first column is 0 to 9, the next column to the left is the count of tens, then 100s, then thousands. You know the drill, but you should be aware that this is a power series. In elementary school, that was too hard a concept to grab onto. Okay, why is this such a big deal now for us clever clogs? Because electronic computers came onto the scene.

It took civilization a thousand years of progress to even grasp the idea that zero was a number at all in any counting system. Who cares about zero? Well, it's a state. You have to start with a null baseline. In electronics, a row of switches can all be off. That's a state the machine can be in. What's the simplest system of counting - binary. Two states. There is the baseline state - zero. You only need one more state and that's a count of one. Adding one more one to the last count progresses the count onward. The math is the same except the base is now two and not ten.

COUNT = digit X 2^n... + digit X 2^3 + digit X 2^2 + digit X 2^1 + digit X 2^0

This is very inefficient because we have to carry on with a massive number of terms to add up to much of a count, but it works. Why would you ever use such a silly system when decimal works great for us humans? The concept of the computer came along with the idea of doing calculations... counting stuff. All the other things that it became was the result of "abstraction" of what these two states could mean. More on that later. Electronic circuits can easily be made to switch on and off or have two states. Ten state switching is crazy complicated and not a practical path. It's way to expensive and fault sensitive. By combining the idea of binary counting and electronic switching we can make a machine that counts! Well, not quite. We have to know about control logic first. But lucky for us, that is also a two-state system.

Digital Logic

The Greeks first and then a bunch of German engineers threw out on the table the philosophical idea of a premise and tests that would result in a truth, and then computing devices came along. You know, "If it walks like a duck and quacks like a duck, it's probably a duck." If you jump a few centuries later after electronics was available to play with, we have, "If I wire up two switches to a light bulb and they are in series, then I have to close both before the light comes on. If I wire the switches in parallel and close either switch the light comes on, or indeed both switches, the light still comes on. This circuitry was quickly labelled an AND gate and an OR gate. Gate, meaning it gated the signal from an a couple of inputs to get a resulting output. This was probably first discovered by a poor electrician that wired a house wrong but later got the lights on. How could this be relevant to modern computers? If he wasn't dead, you could ask Alan Turning and he would have been glad to spell it out for you. I'll do my best. Keep reading.

We need a few more ingredients before we can really make this two-state computing machine do much of anything, but we're getting closer. There is another way to mess with two inputs and that is develop a circuit to produce an output only when one of the inputs is on at a time. If both are off or both on at the same time, the output remains off. This is called the exclusive OR gate or XOR.

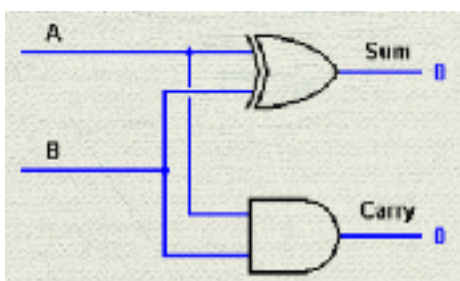
We can now add numbers with just these logic gates. At this point, we have three savvy concepts under our belts. Computers are state machines that once you flip a set of switches, it changes state and stops there. Binary math only needs two states to represent all real numbers. And we can easily build electronic circuits that produce two state functional logic. Now we need to wire this logic into an adding machine. Here's how:

$$\begin{array}{rcr} \text{Binary} & & \text{Decimal} \\ 1011 & = & 11 \\ +\,0011 & = & 3 \\ \hline 1110 & = & 14 \end{array}$$

First we have to agree that 11 + 3 = 14 and that 14 is 8+4+2+0 in binary, and then the binary numbers will make sense as well.

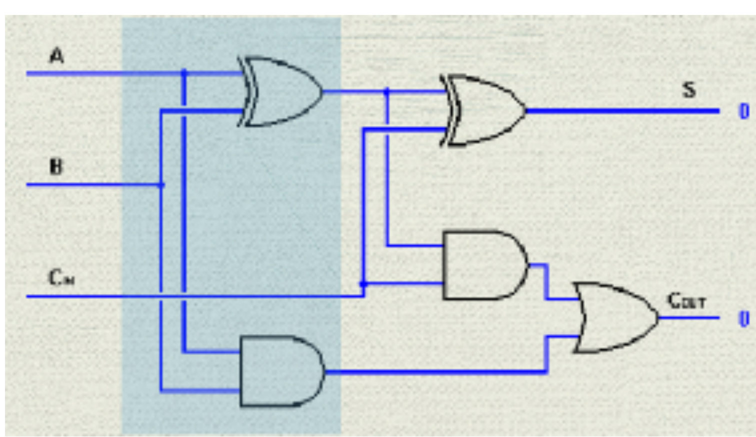
Now we have to make some rules. In binary, 0+0=0, 1+0=1, 0+1=1 and 1+1=0 carry a 1. The carry, of course, goes in the next column to the left. So for the first column: 1+1=0 and gets a carry. In the second column, carry+1=0 and another carry, then the remaining 0+1=1. Now we have 0010, but we have a carry in the third column so that drops down and we have 0110. The fourth column falls through with a 1, to give the answer 1110, which is 14 in decimal. Yeah! Notice each bit in both operands is really just comparing two inputs at a time and that the result matches what an exclusive OR gate does.

What if both inputs are 1's and a carry is generated? We can handle that with the AND gate. Let's add that.



That's great, but we have three other A-B bits to compare. Here it's best to visualize there are four of these circuits needed. With a bit pair in each circuit and the outputs will give us a four bit binary result. We have to cover one last condition.

If we have a carry coming in from subsequent columns, then we need to apply it to this circuit by adding it in. Here's how:



So the least significant bit only needs the first circuit and each subsequent bit comparisons require the second circuit. Now what do we do if the last two bits produce a carry? That literally means we need an addition column to accommodate it. At some point, our resulting addition could exceed our circuit's ability to accommodate it. This is handled by the computer as an "overflow" condition and is reported in a status register where a programmer can be alerted to the issue. But we have to build the rest of the computer to know that happened. This does present the next big thing. We should realize there must be a lot of control circuitry needed to just manage adding two numbers. Yes there is. For now, just hold that thought.

If we use the output bits to turn on lights or nowadays LED indicators and place the LEDs in a row so each tells us if that power of two is on or off, we could read the answer to our math problem. Or we could develop a hairy circuit that could display the whole business on a screen. What we should realize at this point is Mr Turing and his colleagues knew this could actually be done, given enough technical equipment and invention of some special parts.

So what else is needed to really get something workable? We need to hold binary data somewhere, move said data around, build our adder, and then store the answer somewhere or display it at least. And each of these steps has to be a controlled state. Each action results in that step being completed before the next step can be started or the whole thing jumbles and the results would be arbitrary.

We started this discussion with counting things and moved on to adding things. A couple of simple ideas, but if you apply these concepts to electronic circuit possibilities, which is what happened over time, the vision turned into hardware that could actually do this. By keeping everything basic, the simplest approach of only using two states, we came up with logic that could handle the situation and we should see now this can be done. It's a start. What came after Turning writing a paper about how this process might work, was truly amazing. Every techie in the world started working on this and from 1940 to 1950 it all became a reality. Yes, we went from relays to vacuum tubes, to transistors, to integrated circuit to get to where we are today, but it is all based on very simple ideas and a huge amount of imagination. That's why these pioneers are called visionaries.