Abstract:

Computers, for the most part, do four things. They read from their memory, take inputs, do calculations based on the memory and inputs, and store data to their memory. When you tell your computer to display an image, what it is doing is it is processing the input to display an image, reads its memory on how to display an image, reads its memory on the image, and calculates, information like: where to put the image, and what color each pixel needs to be. Almost, if not, everything a computer does involves calculations. That's why it is called a computer, It computes things. My project was to learn about how computers do addition and make a combination of physical and digital representations of how computers do addition. I made a single digit binary adder called a half-adder circuit using manly transistors and other electronic components, a 4 digit binary adder called a 4-bit adder using an integrated circuit, and another 4-bit adder in Minecraft. Although, I was not able to accomplish my original goal of designing, then making a 4- bit adder out of transistors, I still was able to able to accomplish my learning goals. I learned how computers do calculations, I improved in my ability to design electronic circuits, and I increased my understanding of electronics and computers.

Introduction:

You may or may not know that computers run on binary. 1’s and 0’s.(and hexadecimal in fancy computers, but that’s not important for this project) effectively in a computer, a 1 is represented by power being on and a 0 is power being off. When I learned about this I thought, well that's good and all, but how do you do math with just 1’s and 0’s.

A quick lesson on binary:
Binary is something called a base two system. This is because there are only two values; 1 and 0. The system we normally use is a base ten system because we have ten values; 1, 2, 3, 4, 5, 6, 7, 8, 9, and 0. In base ten, when you get to 9, you then put a 1 in the tens column and a 0 in the ones column. Then you repeat until 19 and then you put a 2 in the tens column and a 0 in the ones collom, so on and so forth. In binary, there are only 2 values so after 1, you then put a 1 in the colon to the left and a 0 in the ones column to get 10. Then you go to 11. Again, you only have 1’s and 0’s so you go to 100. Then 101, 110, 111, 1000 and so on and so forth. Addition in binary is just like in base 10. You start by adding the values in the farthest most right column and if there is carryover you add that to the column to the left and repeat.

Now, to do this in a computer you need to make a half adder, which adds two binary numbers and puts out the value for that coloumn, and the value to be carried over to the next column. Now, you may be wondering how to make a half adder. You make a half adder out of logic gates. Logic gates are things that take 2 inputs and give out one output. Different logic gates give different outputs based on the inputs. Here are some logic gates and tables with there outputs:

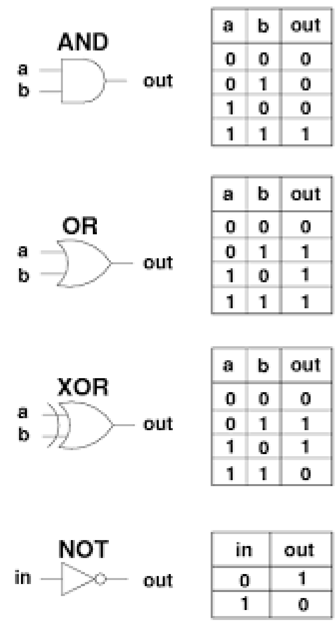

Image from:
https://www.google.com/search?q=and+or+xor+not+truth+tables&hl=en&source=lnms&tbm=isch&sa=X&ved=0ahUKEwjHiIy8jKTaAhVxkuAKHdwODDkQ_AUICigB&biw=1269&bih=659#imgrc=lOVMehhNqXdopM:

The half adders value is an XOR gate because 1+0=1 and 0+0=0 but 1+1=10 so the carryover is an AND gate because that is the only time it would be a two digit number.

A	B	value	carry
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

You can make logic gates out of an electronic component called transistors. You can use half-adders to add larger numbers. A 2-bit adder is something that adds 2 digit binary numbers. A 3-bit adder adds 3 digit binary numbers and a 4-bit adder adds 4 digit binary numbers.

Methouds:

First, a redstone[1] circuit was created in Minecraft[2], that followed the logic of an AND logic gate. Next, a second redstone circuit was created in Minecraft, that followed the logic of an XOR gate. Then, two more compact redstone circuits were created. One with the same function as the first circuit, and the other with the same function as the second circuit. Then, two inputs were made using levers. One lever was connected to the corresponding inputs of the two compact redstone circuits. The other lever was connected to the other corresponding inputs. This replicated the function of a half adder.

Copies of this redstone circuit were constructed and connected together as seen in diagram A, to form a redstone circuit that functions as a 2-bit adder. Then, a redstone circuit that functions as a 3-bit adder was created using the same technique, as shown in diagram B. Then, a third redstone circuit that functions as a 4-bit adder was created, using the same technique, as seen in diagram C. These redstone circuits were then tested with multiple binary addition problems, to ensure they worked.

A half adder was constructed in real life, using 6 2N3904 NPN transistors, 2 10KΩ resistors, 2 1KΩ resistors, 2 LEDs, and a ~3V battery case as seen in diagram D. Then, a 74LS283 4-bit adder, was wired up to 8 switches, 5 LED's, 8 1MΩ resistors, 5 220Ω resistors, and a 5V power supply.(see diagram E) When the 4-bit adder circuit was tested, the LED readout was inverted from the expected readout. The switches were then relabeled to have the offside be 1, and the onside be 0.

Results:

In a computer, transistors are hooked together in specific ways to do addition. First, the transistors are connected together to form logic gates, specifically AND, and XOR logic gates.

(show diagram for transistors making AND and XOR logic gates)

Then these logic gates can be connected to form half adders.

(show images for half adders in real life and in Minecraft and modify them to show where the AND and XOR logic gates are)

Then these half adders are connected to make adders of all sizes.

(show adder diagrams and Minecraft adders)

Currently, many tech companies are mass producing transistors that are 14 nanometers wide, or 70 silicon atoms wide. Scientists have even been able to make a working transistor that is less than 7 nanometers wide with a 1-nanometer wide gate, using carbon nanotubes and molybdenum disulfide.

~~Due to the nature of my project, my results are the circuits I made, the Minecraft world I made, and the diagrams for the varying circuits.~~

~~(add diagrams A through E here)~~

Discussion:

When I started this project, I originally planned to learn about logic gates and transistors, and then use what I had learned to make my own adders in real life and in Minecraft. I had hoped by doing this I would learn about how computers worked, specifically how they do math, as well as have the opportunity to practice designing and building circuits. Part way through my project, I realized that my goal of making my own adders was unreasonable and decided to compromise.

Instead of making my own adders in Minecraft, I would figure out how to make the AND and XOR logic gates on my own, but for the adders use more compact versions of the logic gates I found online. I would still use the diagram for the adders I made myself, but for the sake of time and simplicity, I would use the compacted versions of the logic gates. This allowed me to still get the experience of figuring out how to make the adders out of logic gates, and how to make the logic gates, but without the hassle of using bulky and large logic gates.

Due to warnings from Moe and Gosha about how finicky the transistors where when used in large amounts, resource constraints, and the fact that I had limited experience with complex circuit design I decided to change my plans for the real-life adders. I instead, made my own diagram on how to make AND, and XOR logic gates out of transistors, and even a diagram that used the gates I designed to make a half adder, but not build my design. I instead found a much simpler half-adder design that used the same logic gates, but made them slightly differently, and wired it slightly differently. This way I would still get experience with making complex circuits, but not have to spend excessive amounts of time building and troubleshooting. Instead of making a 4-bit adder, I would use Minecraft to show how to use half-adders to make a 4-bit adder, and just use a 4-bit adder Integrated circuit.

If I were to do this project again, I would try to have a clear schedule for when I would work on different aspects of my project. I think if I did this, I would have been able to accomplish my original goals much better. If I had better scheduled my project I would not have had the resources constraints because I would have had a clear time to order resources, instead of getting them at the last minute. I also would have had more time to research and design simpler and more efficient circuits.

Bibliography:

"Logic Not Gate Tutorial with Logic Not Gate Truth tables" *Basic Electronic Tutorials.* Aspencore, ~september 2014. Web 19 March 2018.
https://www.electronics-tutorials.ws/logic/logic_4.html

Blackburn, Wayne. "Transistor Level Half Adder" *Simplestcomputer. Web. 2:42pm 25 March 2018.*
https://simplestcomputer.wordpress.com/route-map/stage-2-arithmetic-handling-numbers/half-adder/transistor-level-half-adder/

Gartenberg, Chaim. "The world's smallest transistor is 1nm long, physics be damned" *Theverge.* Theverge, 3:34pm 6 October 2016. Web. 9:02pm 8 April 2018.
https://www.theverge.com/circuitbreaker/2016/10/6/13187820/one-nanometer-transistor-berkeley-lab-moores-law

Hazari, Arnab. "Electronics are about to reach their limit in processing power—but there's a solution" *Quartz.* 5 January 2017. Web 9:00pm 8 April 2018.
https://qz.com/1228103/flying-cameras-are-watching-over-our-oil-and-gas-facilities/

Special thanks to:
Jonathan Hunt
Moe Frumkin
Gosha Lubashev
Michael Hirsch