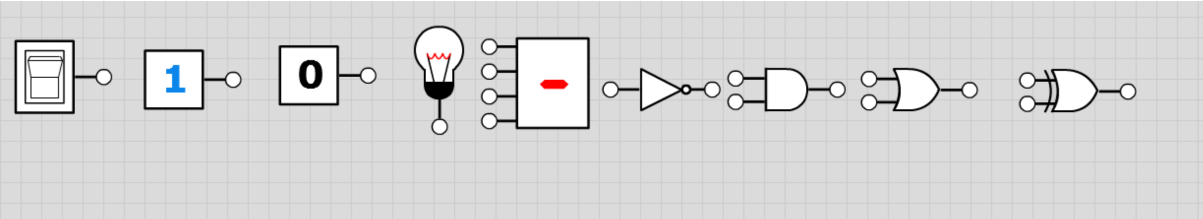
Calculator Project: by James Topalian

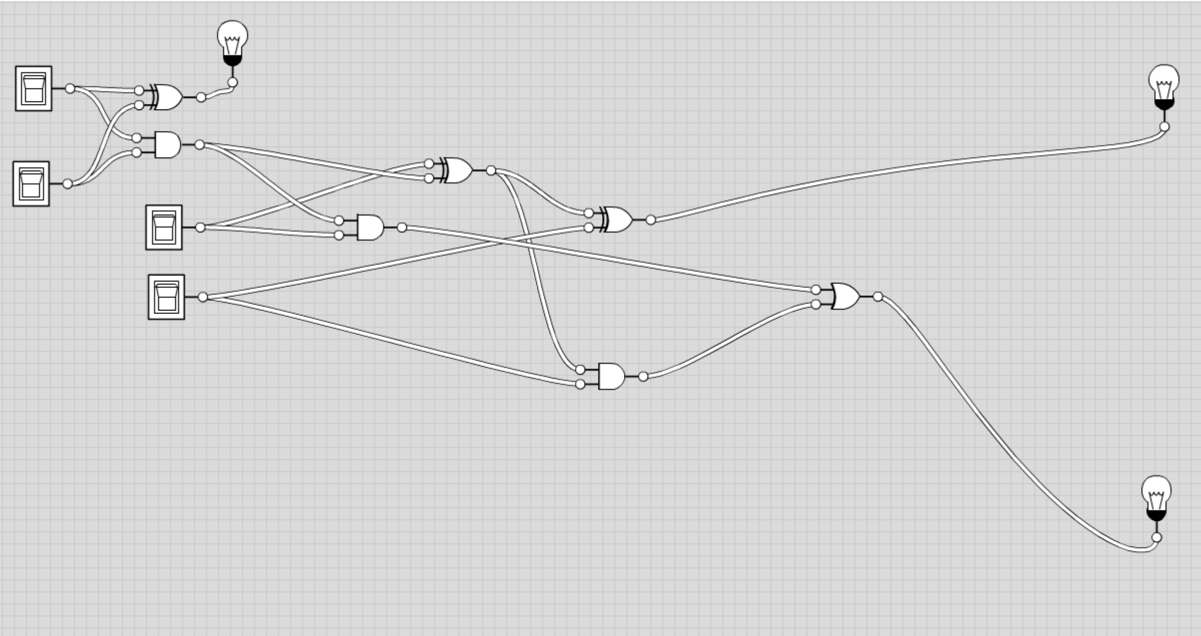
Introduction: For this project the goal was to research and improve our understanding of logic functions (specifically ADDERS), to a degree where we could construct a binary calculator using logic circuits. The idea being that we could input an 8-bit binary number and then add or subtract a second 8-bit binary number and get a result produced automatically in the form of an 8-bit binary number. So the output would be 8 objects that can show two forms to represent 1 and 0 in binary. Light bulbs would likely serve this function. The inputs would be 8 buttons to represent the first number, each with an on or off state (for 1 and 0) and another set of 8 for the second number as well as a button to toggle between an adding and subtracting function. The program we used to create these is called “logicly”.

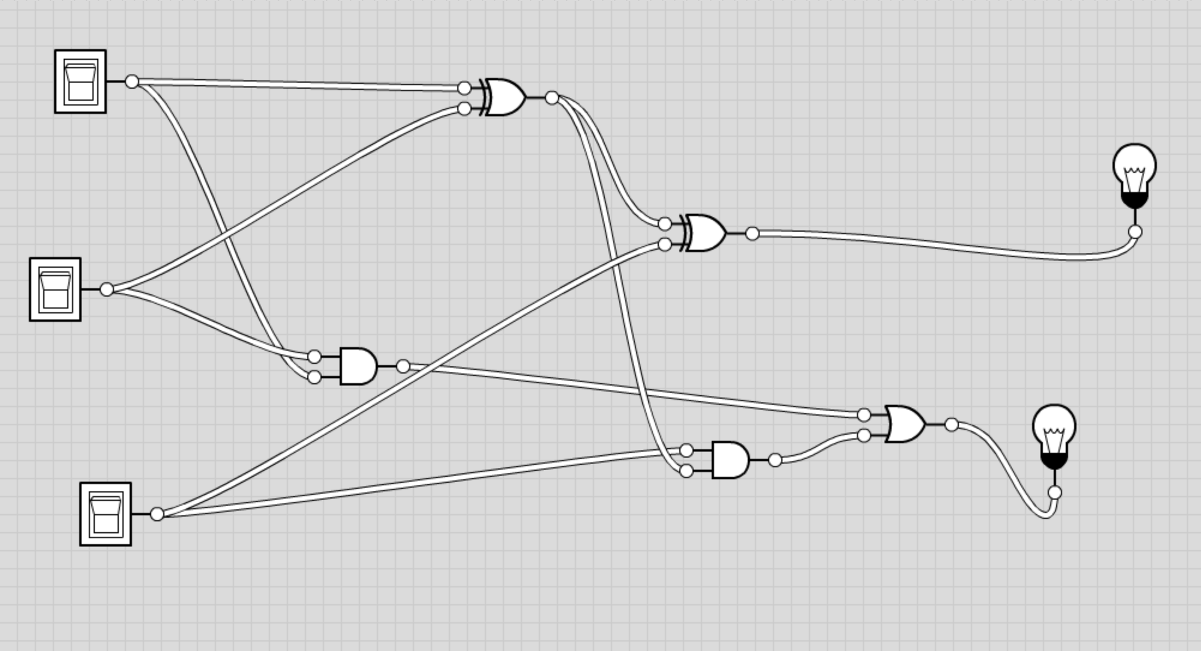
Explanation: Firstly I shall try to explain the function of the symbols used in my logic circuits.



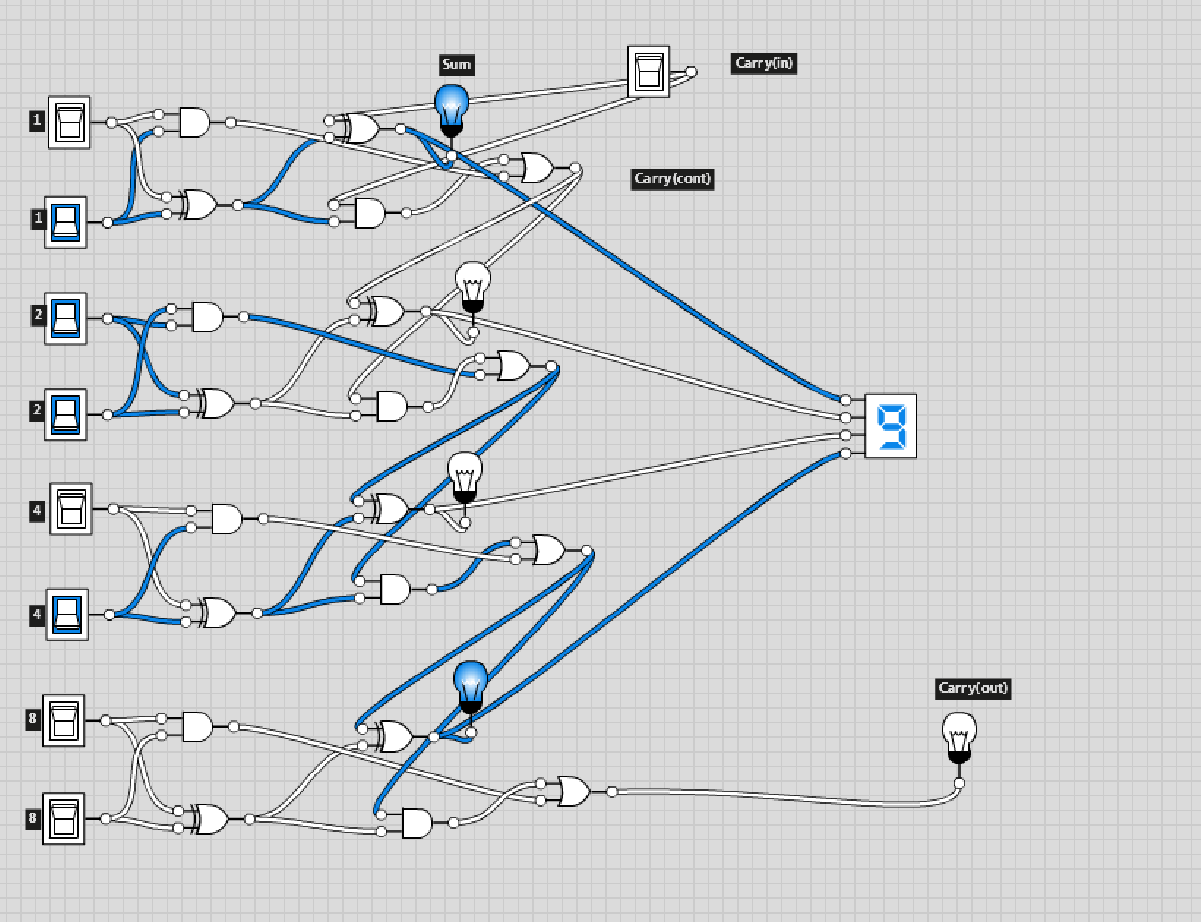
The first three on the left are the inputs, first is the button that has two states on and off (which as previously explained can represent a 1 or a 0), then we have the 1-constant which is always in an “on state” and finally the 0-constant which is always in an “off state”. Following them we have the 2 outputs the light bulb which can clearly show an on or off state, and the hexadecimal display which links up to 4 inputs giving them 2-base values and translating into a hexadecimal state. Next are the 4 commonly used functions in my circuits, which link the outputs into the outputs in such a way as to produce the desired results. In order from left to right we have the “not gate” which turns an off connection on or vice versa. Next we have the “and gate” which requires all inputs attached to it to be in an on state in order to produce an on output connection. Then we have the “or gate“which will produce an on state output so long as one of the inputs is in an on state. Finally the “xor gate” which will produce an on output only if one input is on and the other is off.

Adders: We started off by building basic adder constructs. Such as linking two 2-bit adders together. As well as building a 3-bit adder



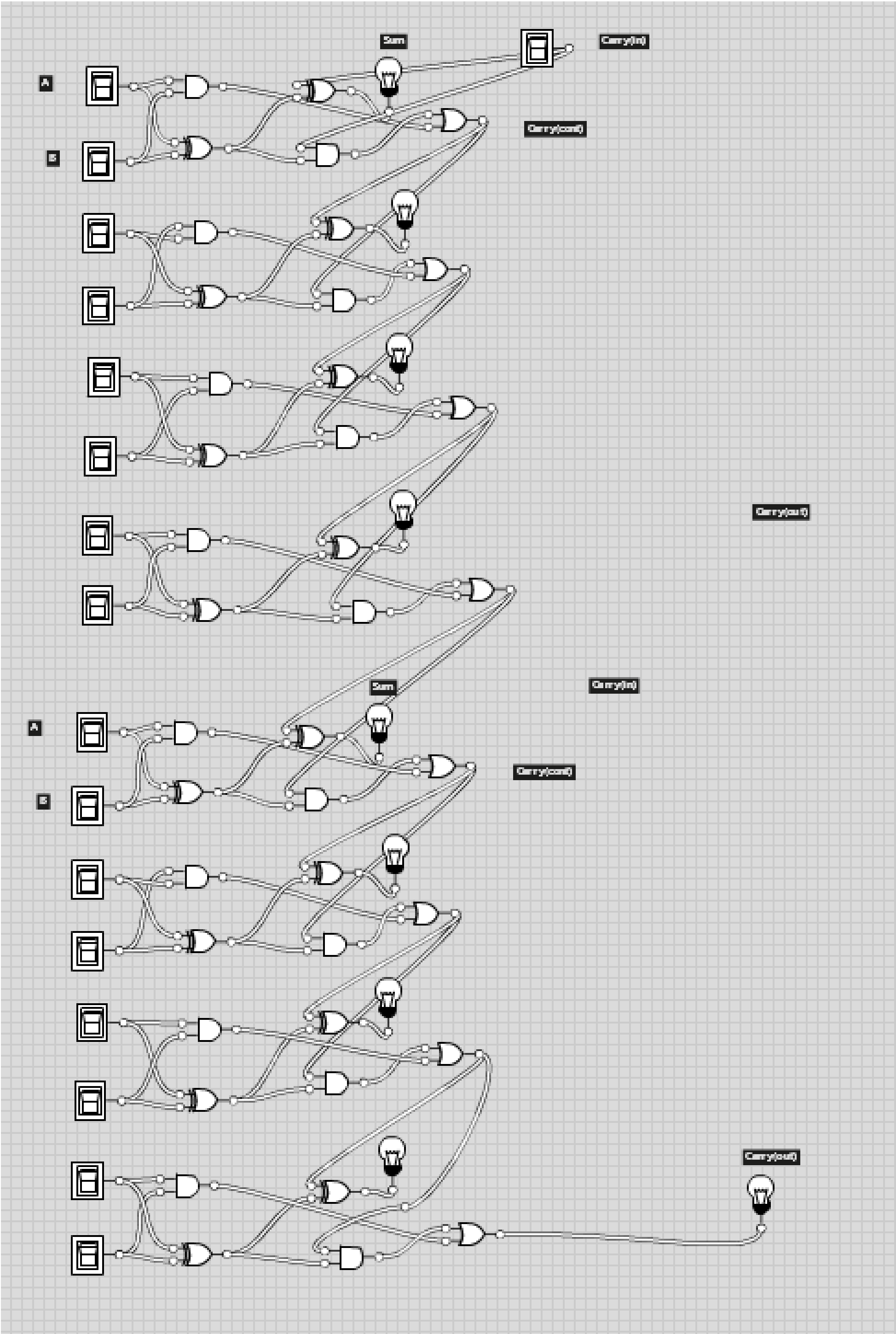


Then we came to understand how by assigning values to the inputs and having the circuit carry over values we could assemble a logic function capable of adding 2 bits and realizing that they are equal in value to a higher bit. Which led to my next adder construction.



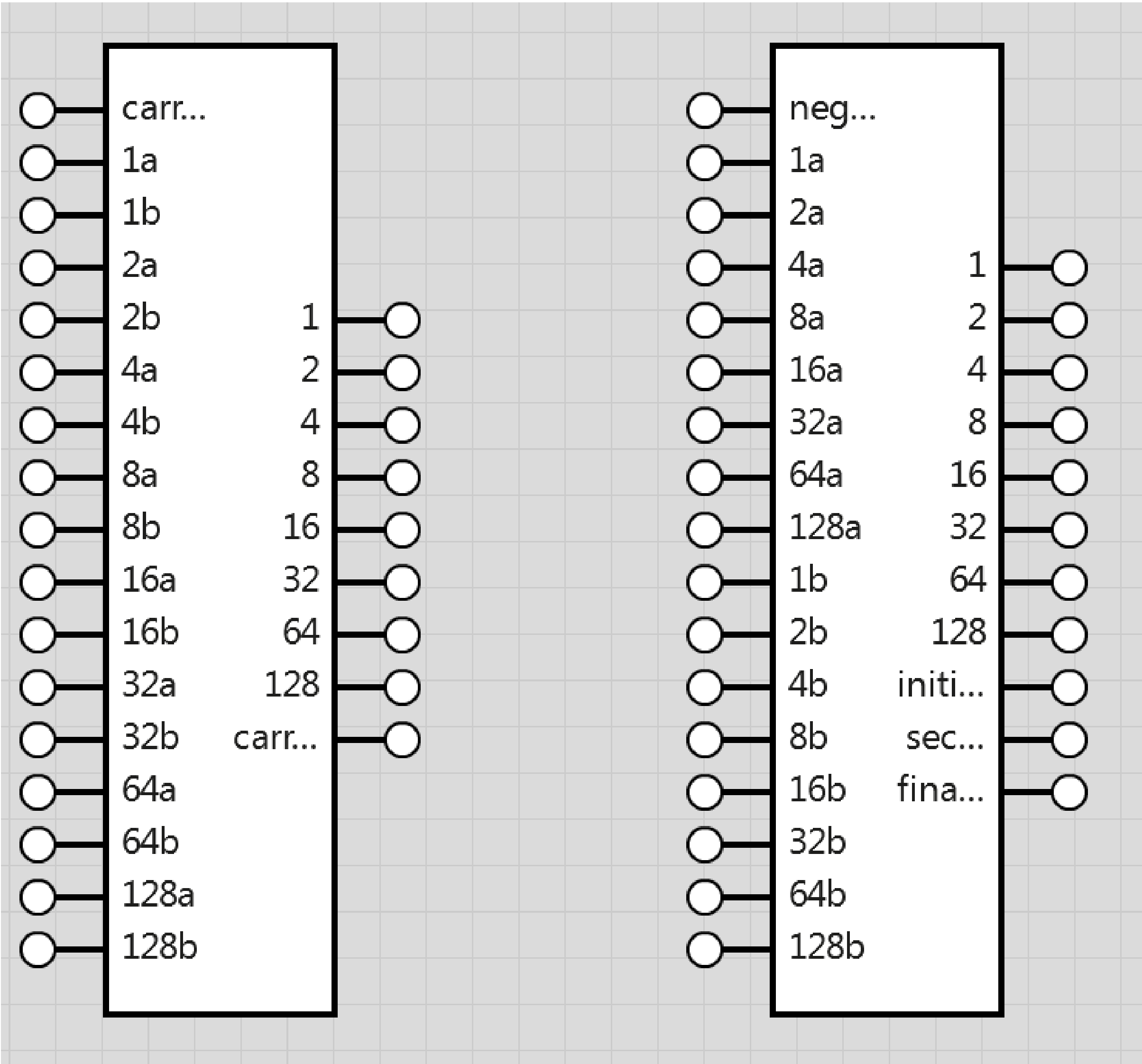
As you can see I have assigned values in accordance with a binary(base 2) system, to the inputs. The lightbulbs possess a correspondant value to the inputs alongside them(with the exception of the carry out, which would possess a value of 15), so the highest is valued at 1 the next at 2 then 4 and then 8. As is visible when inputs being at 4+2+2+1 the lightbulbs show 8 and 1 giving the answer of 9, This is more clearly shown on the hexadecimal displayed I tagged into the system.

The next step was to create a circuit capable of adding 2 8-bit numbers together.



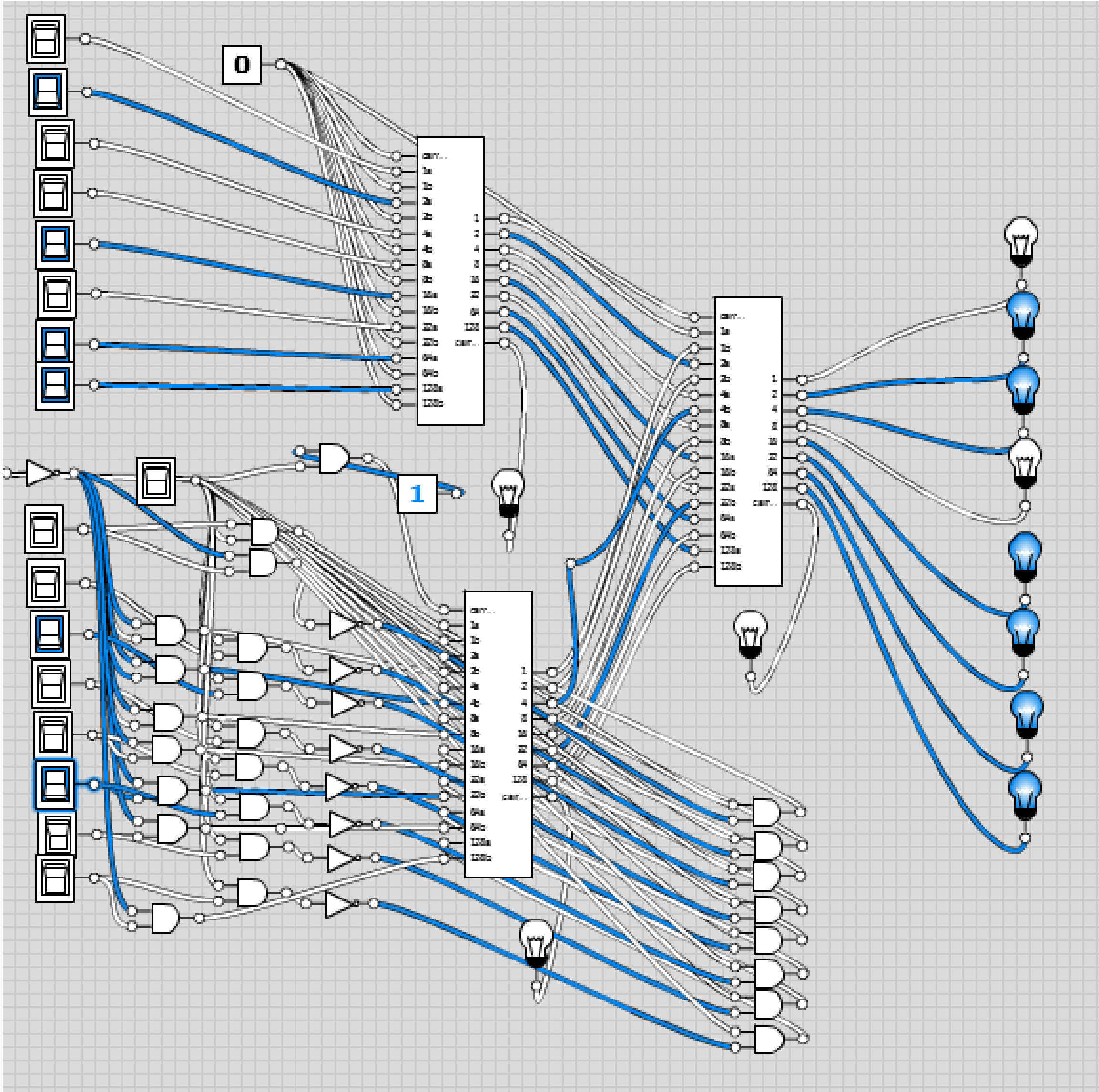
From the previous build this was rather straightforward.

At this stage we created integrated circuits in order to tidy up and simplify the work from here on out.



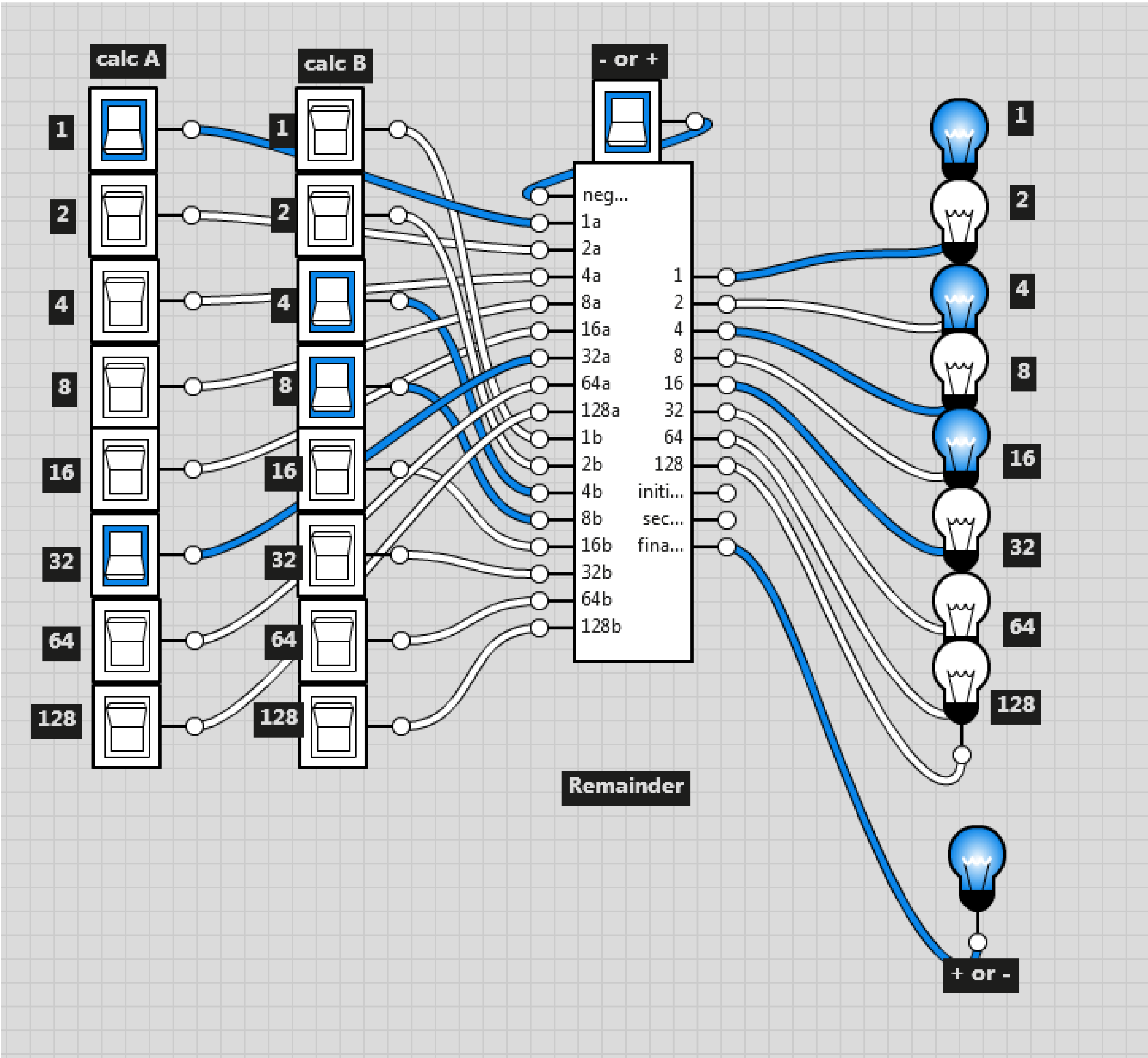
The one on the left being the 8bit adder , and on the right a compiler that takes inputs which have already run through two adders and puts them together in order to add or subtract them.

The next major step was to build the subtractor for which, as in binary we had to create a complimentary sequence to the number being subtracted and then carry 1 into said complimentary sequence, the result of which can then be put with another number to have it subtracted from it.



As can be seen here using the aforementioned integrated circuits the higher inputs produce a positive number and the lower input circuit produces a complimentary sequence to the number you put in to subtract and finally the two are compiled with the 2nd type of integrated circuit I produced to come out with a result.

Then for the final stage of our main project we had to create a system that offered a choice between adding and subtracting.



So here you can see my final product, calc A represents your starting number and calc B the number you add or subtract from it. The" – or +" switch being turned on makes it a subtraction equation and the "+ or –"light bulb being on shows that the result is positive.

Conclusion: This has been an interesting and educational project, which has allowed us to develop our logical thinking skills. We have learned ways in which binary can be applied to circuitry to produce a variety of specific results. This type of logical thinking can be applied to many things; a clear and notable example is the game minecraft. In minecraft you can create circuitry to perform the same logic functions and can then use them to interact with the game objects in advanced ways. These logic gates are also implemented in engineering and mechanics. The project has given us an understanding of how inputs are converted to outputs and in the future I hope to develop these skills further. To become capable of creating more advanced circuits to perform specific tasks and to include more user friendly features such as a denary display.

References: -Constructed on downloaded program logicly-
The teachings of Thomas Pasquier
Program (logicly) associated help page
http://www.allaboutcircuits.com/pdf/DIGI.pdf