I had messed up before and didn’t allow a circuit to be NANDed with itself. I fixed this. I also made it so that the costPerOp and approvedCircuits were in a separate file that the Circuit file looked at. This sped things up and saved space because each circuit no longer had to save these values individually. I also changed the switch/case to if/else if so it would work on Bryan’s computer.

I successfully got the results to write to a Json file in the correct format instead of writing to an arbitrary format. It makes a new file for each cost it does and includes the date in the file name which decreases the chance of overwriting a file.

I updated the read me file to reflect the changes made. I sent the new files along with the half sheet explaining Jing’s project that Bryan asked for yesterday to Bryan.

Now I am going to look into the SRA toolkit that Jing mentioned earlier.

During the night I added a bit of code that prints the percent completion of a level approximately every 5 minutes. Initially I had it print randomly but I changed it to being every 5 minutes.

It seems like the rate at which it builds circuits slows down. It should print 1 milisecond after 300 seconds after the last print if circuits are built at a rate of 1 circuit per millisecond. 15 minutes into 7.0 cost circuits of @ only, the program is building circuits at approximately 1 circuit per 1-1.5 seconds. Which means the increased usage of memory considerably slows things down. Perhaps there is a way to not store the objects in current memory to speed things up. Possibly writing new circuits to a file instead of saving them as an ArrayList in the HashMap sortedByCost.