**Reverse Engineering of IoT Systems**

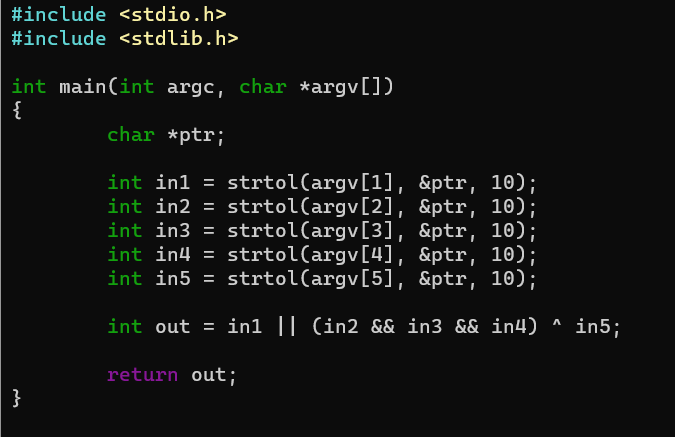
**Lab 2**

**Completed by Coleton Sanheim**

**Submitted to Lubos Kuzma**

**2022-02-09**

Part 1:



This is the code I wrote for part 1, it take 5 inputs from the command line and converts them to their integer values (0 or 1) and then puts them through some simple Boolean logic and returns the output. A picture containing text

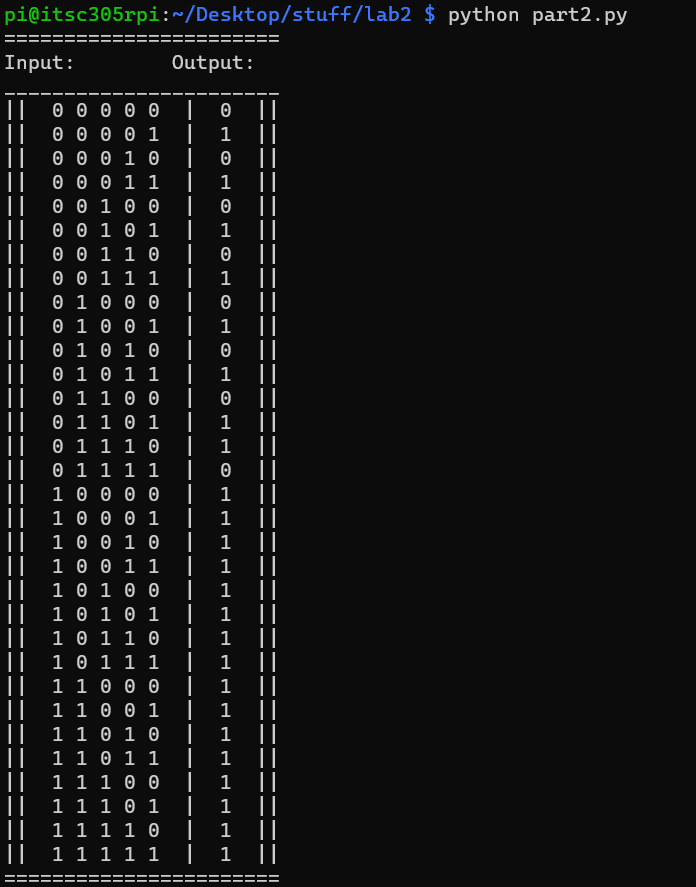
Description automatically generated

As requested this is the assembly code for the program above. I used the debugger gdb to disassemble it.

Part 2:

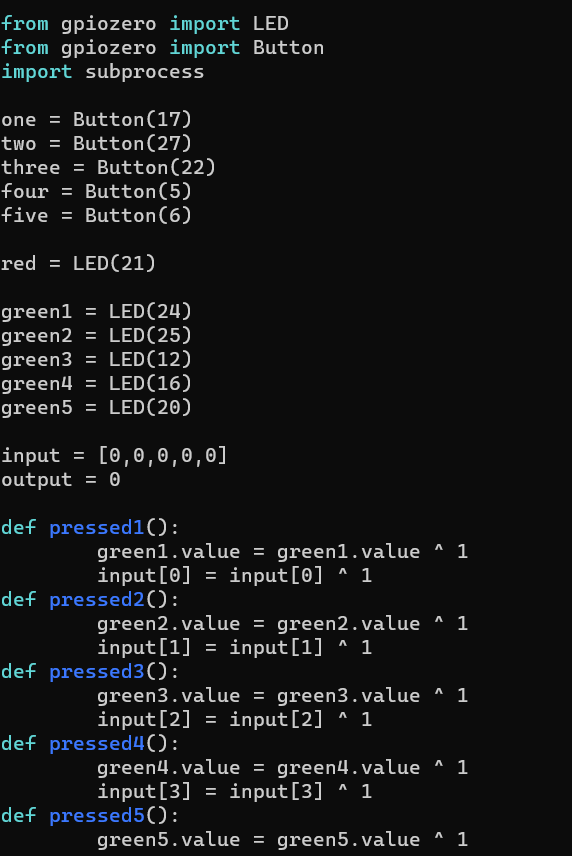


This is a python script that uses the itertools library to create a list of every permutation of 0 and 1 with a length of 5. I then use the subprocess library to run these created digits through the program created in part 1, I then get the outputs from that program and printed them out to the screen in a table. This will run through every possible input of 0’s and 1’s to create every possible output.



This is the truth table generated by the program being ran above.

Part 3:

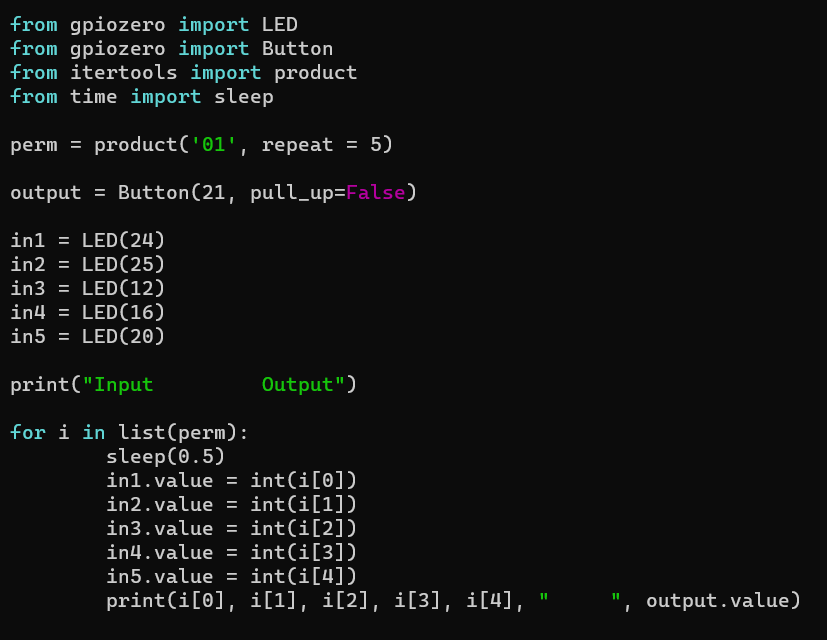




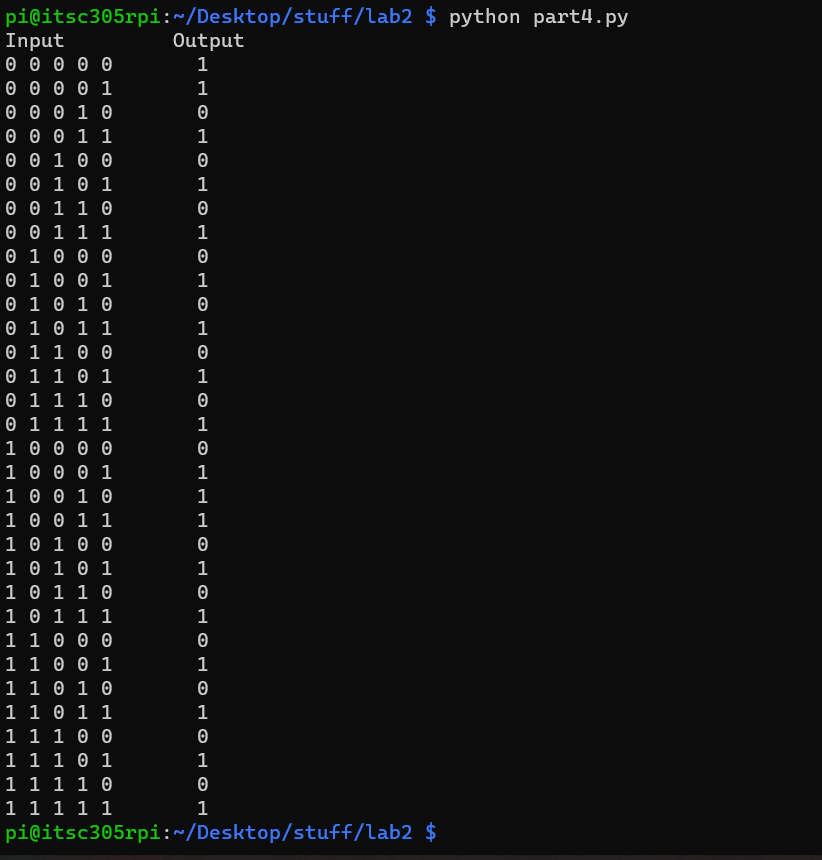
The code was too long to fit in a single screenshot so there is two here. I did something a little different then what was asked when making this code. I added 5 LED’s to the circuit, one for each button and make each button act as a toggle switch, where each LED represents whether a button is “on” or “off” where on represents a 1 and off represents a 0. I expected to have to declare the array as global, but apparently it works without. The output will update in real time and output in the form a red LED. After some testing this works as expected. I feel like there was a better way to accomplish this however, instead of making 5 different functions, one for each button, I would try to make one function that can handle it, but I was unable to figure this out.

Youtube link: <https://www.youtube.com/watch?v=D4I1a-ddTRY&list=PLnRsOe1-tLI1NaS9RGdMRn1eGAGkkBFum&index=3>

Part 4:

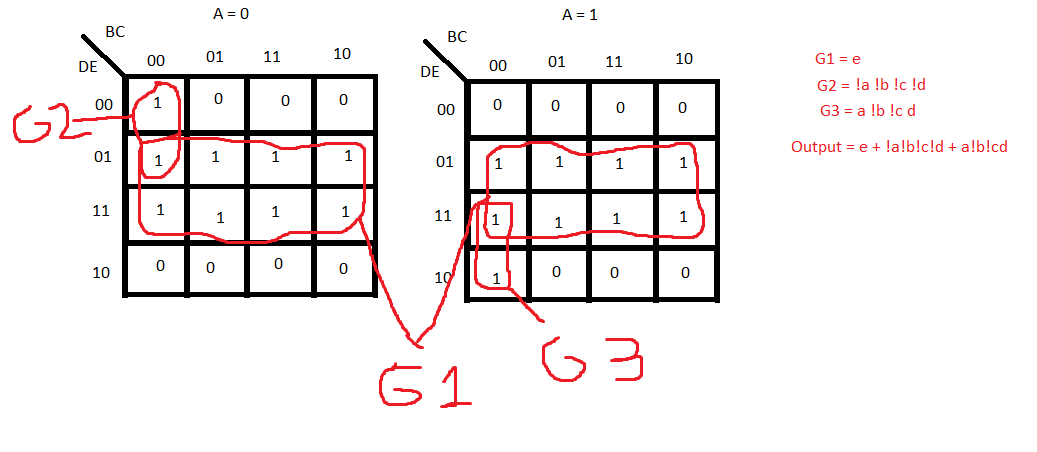


This is the code used to iterate through each possible input of the inductor provided black box. The inputs are in the form of LED objects and the output is in the form of a Button object, however there is no buttons or LED’s connected but is instead connected to the black box.

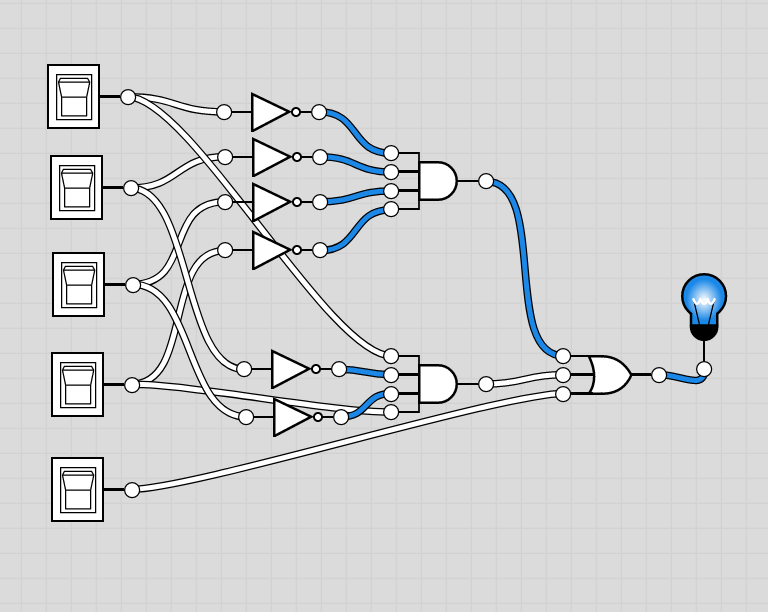


Here is the output from the above program.

Part 5:



Here is the Karnaugh Maps I created for the truth table generated in part 5. In it there is the Boolean expression that was calculated. Also included below is a simple circuit I designed that uses this Boolean expression.



Critical Reflection:

In this lab we explored the behaviour of black box programs and how to analyze their outputs to determine the Boolean representation of the logic happening inside the program. I learned how to create wrapper programs and how to evaluate output. This seems like a great way to help figure out the behaviour of IoT devices.

In the first part we created a .C program that takes 5 inputs from the command line and runs it through some Boolean logic, this is a pretty simple program and something that I have done before in previous classes.

In part 2 I created a python wrapper to interact with the executable generated from part 1, then print the outputs in a truth table. This was accomplished fairly easy, but I had to do a lot of research on two python libraries; subprocess and itertools. Subprocess I learned was such an easy way to call another program, before I used CTYPES and that was very complicated, I did run into a problem with using the .run method, but learned that the .call method was the better one to use in this specific situation, as it only returns the return value instead of a bunch of other information. I ran into some problems using itertools as I could not figure out how to get it to iterate through all the inputs properly, but after some research and testing, the .product method worked.

In part 3 I adapted the program from part 2 to instead of iterating through each possible input, to take input from a series of buttons and display the output in the form of an LED. This was fun to set up, and no real challenges came up, however I did something different than what the lab asked for, I added 5 LED’s and set them to turn off and on in relation to the buttons, so if the button is pressed it turns on and if the button is pressed again it turns off. In this way the buttons act more like toggle switches. The way I have my program set up is kind of obtuse and I would like to figure out a way to do with only 1 or 2 functions instead of 5.

In part 4 I hooked up my pi to the instructors pi and tested a black box program running there, this took a little time to wrap my head around as I had to set the inputs and outputs to an LED and buttons even though there were no components. However I now understand that it registers those functions as input and output values, rather than the actual component and we can use that in this situation to give output and take input even without the components themselves.

In part 5 I created a Karnaugh map based on the output recorded in part 4, I learned how k maps work and how to extract a Boolean expression from them, this seems useful to evaluate behaviour of black box programs.