Because my Text files seem to have a problem opening, I decided to create a lab report document with the information that would be on them.

**Problem 1:**

***2. Run both solutions using the timing framework provided and note down how long each method takes to process.***

*Solution 1:*

**Timer Resolution = 1 nanosecond**

**Calibration time = 0 seconds and 542 nanoseconds**

**The measured code took 0 seconds and 4296712 nano seconds to run**

*Solution 2:*

**Timer Resolution = 1 nanosecond**

**Calibration time = 0 seconds and 517 nanoseconds**

**The measured code took 0 seconds and 4294967142 nano seconds to run**

***3. Now, using the same timing framework, run the code version that you created for Lab 1 (Problem 4). Report the difference between the performance between your code and the two versions of the solution codes***

**Timer Resolution = 1 nanosecond**

**Calibration time = 0 seconds and 547 nanoseconds**

**The measured code took 0 seconds and 4297712 nano seconds to run**

Overall, my code takes the longest to run and the longest to calibrate. I do believe that it has something to do with all of the if else statements that my code needs to run because when I first competed this code, I did a lot of if else statements and was not quite sure on bit masking. Solution two has the fastest speed followed by solution one (which may be contributed to my own knowledge on bit masking I am still working on faster solutions) and following in last place is my own code.

***4. Write an analysis on why different versions of the solution code might be faster. Also provide an analysis on why your code is slower or faster than the solution code(s).***

Overall, I do believe that a switch statement runs faster than an if/else especially when the if else has various different sections to revolve around. In the case of a switch a code can rapidly check the conditions while following through only on those met while an if else has to check each condition at what appears to be a slower pace.

***5. Identify ways to improve your code (or the solution code in case your code performs better).***

The best way to improve my code is make a flawless and fast transitions through switch statements, it appears the more conditions added the faster the code.

**Problem 2:**

***1. What is the meaning of “SPST” and “NO” in the context of electro-mechanical switches?***

SPST in the context of an electro-mechanical switch is a single pole, single throw switch. This type of switch is much like an on or off switch in the case that it has only one input and one output. A NO switch is a Normally open switch when the switch is off it is compressed.

***2. Design a small digital circuit (with Logisim) using basic logic gates, so that when this circuit is used with the chosen switch, it will make the circuit function as a toggle switch. Use the button component in Logisim for this question. (Input/Output > Button) The way the button switch operates in Logisim, as long as you keep the button pressed an LED or other output signal will stay ON. When you release the button, the LED or output signal goes OFF. Design a toggle switch such that when you press the button once, your LED turns ON and then remains ON (even after the button is released) until the button is pushed again.***

***\*Seems to have problems when logism is reopened. \****

***Letter

Description automatically generated with medium confidence***

This logism circuit has been designed at redesigned several times, I have not been able to completely work out the kinks but I am still going keep coming back to it to fix any and all kinks.

***3. Using the text-insert feature in Logisim mention in the circuit the chip number(s) (from the 74XX family) that you would use for the circuit and identify the pin numbers that would be used by your circuit.***

7474

**Problem 3:**

***1. Design and verify the required circuit to implement the I/O control sub-system. Note that the given specifications do not specify a way to turn OFF a device once it has been turned ON. Making appropriate assumptions, include a control sequence that can also allow you to turn OFF a device.***

***A close - up of a graph

Description automatically generated with low confidenceCalendar, whiteboard

Description automatically generated***

***2. Create timing diagrams showing the change of state of the various signal and control lines as implemented by your circuit. Show the clock and the set of pattern waveforms on A0-A3 and the control lines based on the following activities. Clock cycle 1: A3-A0: value 0010 Clock cycle 3: A3-A0: value 0001 Clock cycle 5: A3-A0: value 0011 I/O goes to state-1, one clock cycle after A3-A0 gets a new value and remains at a high state for one cycle. At Clock cycle 0; All lines are at 0 state***.

Door 1: 1 0 0 0

Door 2: 0 1 0 0

Door 3: 0 0 1 0

Door 4: 0 0 0 1

A/C: 1 1 0 0

Engine: 0 1 1 0

Wipers: 0 0 1 1

For the piece to activate the clock must be on high and input has to match the input above. Otherwise they do nothing.