1. *Study the two alternative methods of working on the question after completing the second version.*

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

Code A:
```
Timer Resolution = 1 nanoseconds
 Calibrartion time = 0 seconds and 542 nanoseconds
 The measured code took 0 seconds and  4294967142 nano seconds to run
```
Code B:
```
Timer Resolution = 1 nanoseconds
 Calibrartion time = 0 seconds and 517 nanoseconds
 The measured code took 0 seconds and  4294967141 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.*

My Code:
```
Timer Resolution = 1 nanoseconds
 Calibrartion time = 0 seconds and 493 nanoseconds
 The measured code took 0 seconds and  329 nano seconds to run
```

Overall, my code runs the fastest, clocking in at about 450 - 600 nanoseconds, and almost always staying under 575 ns.
Second fastest would be code b, which clocks in around 500 - 750 nanoseconds.
Last is code a, which in about 540 to 850 nanoseconds.

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

I believe the reason why code a is slowest, is that it uses the most comparisons and has multiple switch and if statements to run through. Whereas code b uses only one switch statement, therefore it one has one statement read while being timed. I think the reason why my code is the fastest, is because if statements are faster than switch statements, and that's all I use.

5. *Identify ways to improve your code (or the solution code, in case your code performs better).*
A: Apparently my code runs very fast considering it beat out the other two programs. I can't really think of a way to beat that.

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

NO = Normally open
SPST = Single pole, single throw

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.*

A: 7474

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.*

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

When the clock is high, and the input in 1000, then actuator Door 1 will activate.
When the clock is high, and the input in 0100, then actuator Door 2 will activate.
When the clock is high, and the input in 0010, then actuator Door 3 will activate.
When the clock is high, and the input in 0001, then actuator Door 4 will activate.
When the clock is high, and the input in 1100, then actuator A/C will activate.
When the clock is high, and the input in 0110, then actuator Engine will activate.
When the clock is high, and the input in 0011, then actuator Wipers will activate.