Cora English | 127007634

CSCE 312-201 Lab 3

Problem 1:

1. Study the two alternative methods of working on the question after completing the second version.
    - The two methods consist of separately implementing the bitmasks for each output or implementing a combined bitmask for all possible inputs.

2. Run both solutions using the timing framework provided and note down how long each method takes to process.
    - Solution 1 took anywhere from 200-1500 nanoseconds to run while solution 2 took anywhere from 800-3000 nanoseconds 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 from Lab1 took the same amount of time to run on average as solution code 1. The actual solution code is very similar to what I wrote for lab 1 which makes sense as to why the runtimes were similar.

4. Write an analysis of why different versions of the solution code might be faster. Also, provide an analysis of why your code is slower or faster than the solution code(s).
    - My code has the same runtime as solution 1 and faster than solution 2 because the computation complexity in each program is very low compared to that of solution 2. There are over 100 cases that solution 2 needs to check which makes it less efficient than my code and solution 1.

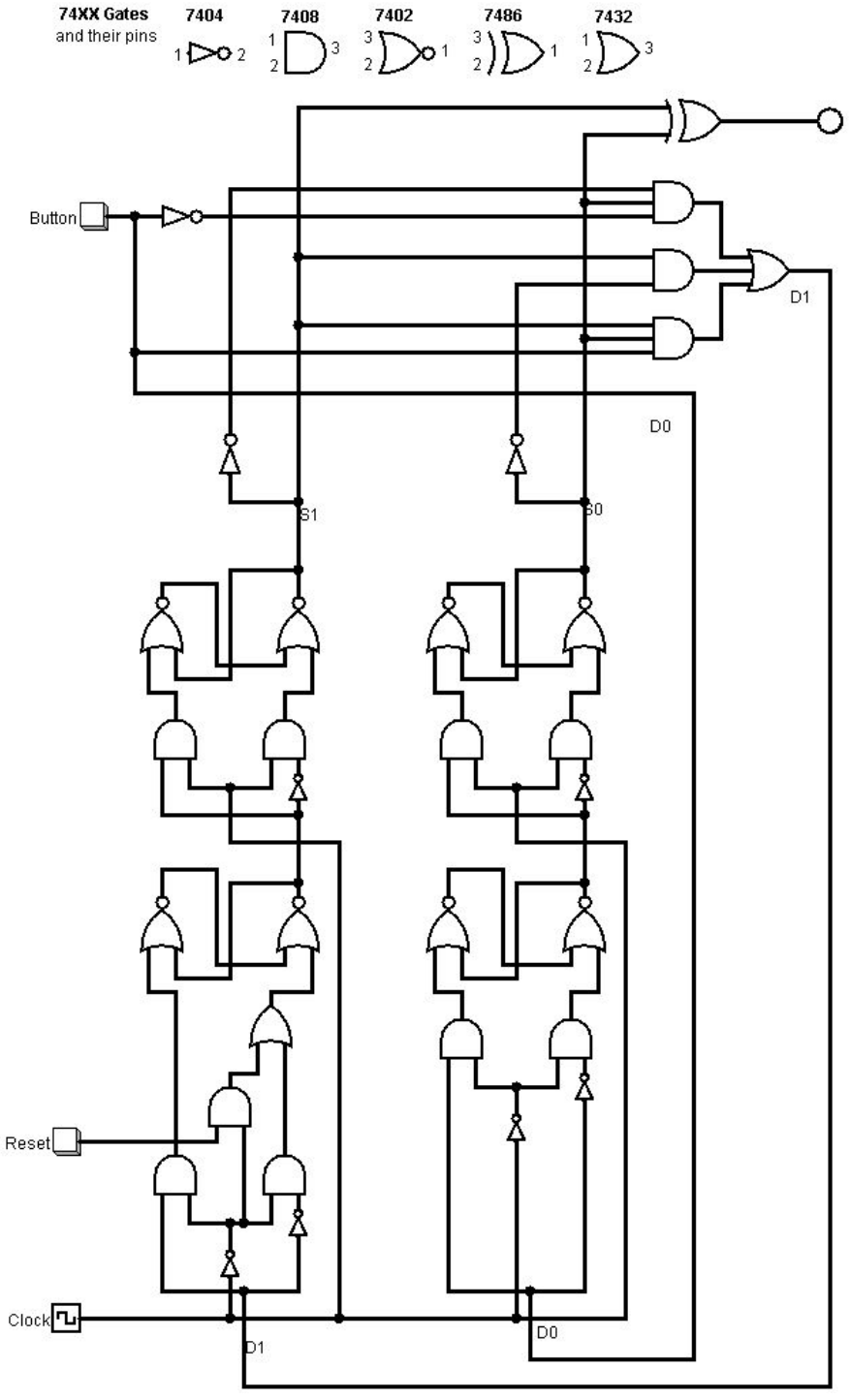
5. Identify ways to improve your code (or the solution code, in case your code performs better).
    - I could have condensed my code to make the logic a little bit simpler for each of the requirements. In solution 2, a way to condense it would be to not do it that way and instead implement it as done in solution 1.

Problem 2:

1. What is the meaning of “SPST” and “NO” in the context of electro-mechanical switches?
    - An SPST switch stands for single-pole, single-throw. It has one input and one output and will either be closed or disconnected. It makes a good on-off switch.
    - A “normal” state for a momentary switch occurs when it is not actuated. This state can be an open or short circuit. When it is open it is said to be normally-open (NO) - when you close the circuit you actuate the switch.

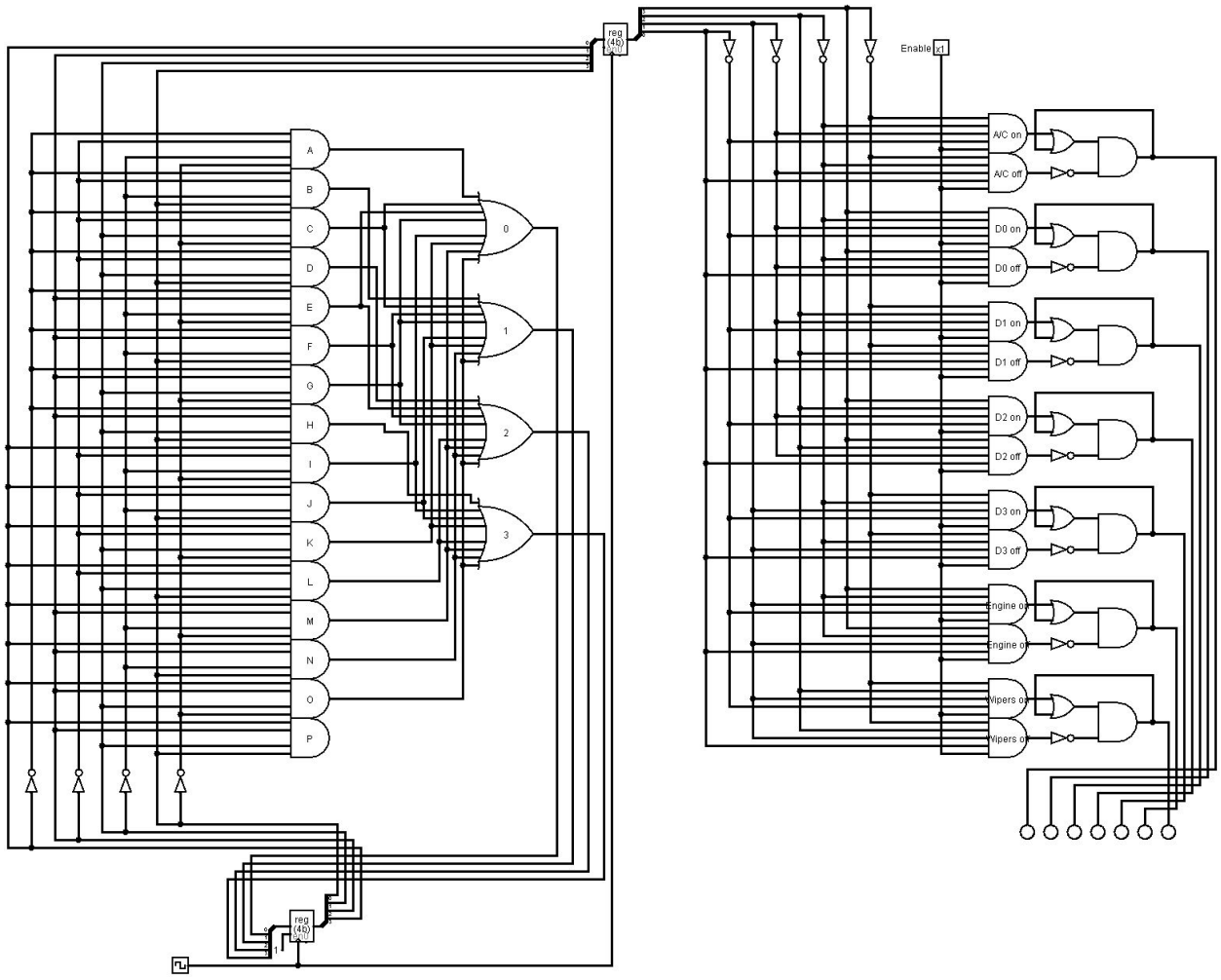
2. Design a small digital circuit (with Logisim) using basic logic gates, so that when this circuit is used with the chosen switch, it will make the circuit function as a toggle switch. When you press the button once, your LED turns ON and then remains ON (even after the button is released) until the button is pushed again.

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.



Problem 3:

1. Design and verify the required circuit to implement the I/O control sub-system.



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.