CSCE 312 Lab manual

Lab-3 - Sequential logic design

Instructor: Dr. Ki HwanYum

Prepared by

Dr. Rabi Mahapatra.
Suneil Mohan & Amitava Biswas

Fall 2016

Department of Computer Science & Engineering
Texas A&M University

Chapter 4: Sequential logic design

In this chapter, we focus on the design of sequential digital circuits for real-life applications. Sequential circuits allow us to capture the notion of time, so that it is possible to store and track different states across time. Thus sequential circuits can manifest higher level of machine intelligence and perform more complicated tasks than combinational circuits.

1. Learning duration: 2 weeks. **Required Tools:** Logisim 2.5.x

2. Objective:

To learn -

Primary topics

1. How to design sequential digital circuits using logic gates and standard components.
2. How to document the design and its operations for others to understand.
3. What are the basic design considerations for sequential circuits.
4. How to standardize designs to generate reusable circuit design patterns to benefit from standardization and reusability (concept of macro).
5. How to apply standard design patterns to organize basic IO and peripheral circuitry.

Secondary topics

6. Using the schematic design tool (“Logisim”) to create reusable libraries of design patterns.
7. Familiarity with all the basic sequential circuit blocks.
8. How to interpret data sheets for industrial components as apart of the design activity.

3. Instructions: Use Logisim for your designs. Logisim should be available on the lab machines since you used it for the previous lab. If not, you can download a copy from the previously mentioned website.

4. Useful resources:

1. Chapter 1, 2, 3 of Frank Vahid’s “Digital Design” or the first few chapters of any digital circuit/design/architecture book by Morris Mano (available in TAMU library).

5. Exercises to do

5.1 Problem 1: Download the Lab-3 extra files from the lab web page. When you extract the zip archive, you will see **two possible solutions** to problem4 from Lab-1. The second solution is incomplete (does not implement all 5 requirements). There is also an **Excel sheet** that can help you understand how the solution code was created and a timing framework that you will use.

Activities to do-

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

5.2 Problem 2: You performed well in you last design assignment at Ford Motor Company, so you still retain your job there. Today your manager asked you to design the switch which starts and stops the car's air conditioning system. The requirement is given as –

"….When the a/c button is pushed it should start the a/c, pushing this button for the second time should stop the a/c. The button also incorporates a built-in green LED which glows when the a/c is operating, and remains off otherwise. This switch signals the car's main control logic to turn on the relay which switches on the a/c compressor… "

You asked one of your senior colleagues to help you get started. He told you that this kind of button functionality is known as a "**toggle push button**". He also told you that as Ford is facing competition to cut costs, it would be better to use a "normally off (push-to-make) illuminated push button (SPST, NO)" to implement this toggle push button (which is cheaper and more reliable), rather than using a costly and less reliable mechanical toggle button. This kind of push button with built-in LED (for illumination) is available in the market. The "SPST NO illuminated push button switch" which you finally choose has four terminals – two for the switch, and other two are for the LED which has a register in series (to limit the current through the LED), you can apply 3 to 5 V to the LED terminal to illuminate the LED.

Hint: Different types of switches are described in the below link –
http://www.kpsec.freeuk.com/components/switch.htm

Activities to do-

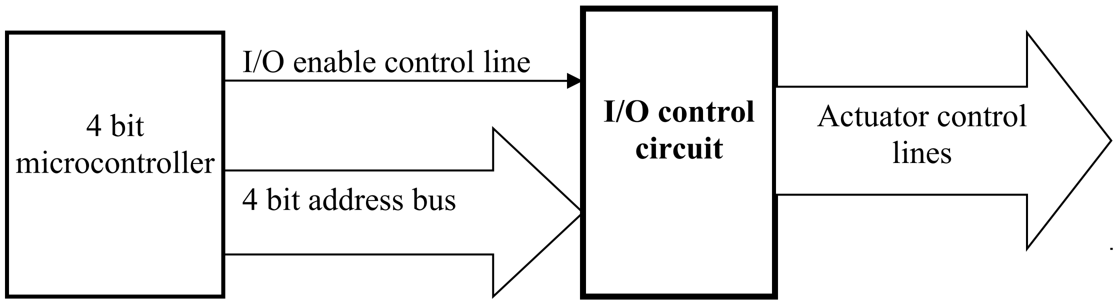
1. What is the meaning of "SPST" and "NO" in the context of electro-mechanical switches?
2. Design a small digital circuit (with Logisim) using basic logic gates, so that when this circuit is used with the chosen switch, it will make the circuit function as a toggle switch.

   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.
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.

5.3 Problem 3: Your manager at Ford has asked you to design a circuit to implement **"addressable I/O" for a 4 bit microprocessor**/ microcontroller. This will be used to activate 7 actuators (the a/c compressor, 4 individual door locks, the engine starter and the windshield wipers). This circuit will allow a CPU with a 4 bit address bus to activate actuators by placing/writing suitable addresses on the address bus. He has provided the following text requirement to get you started –

"…The a/c compressor is turned on by putting a binary equivalent of 0 and enabling the I/O enable line. An address value "2" gets the 2nd door locked (Door 2)….."

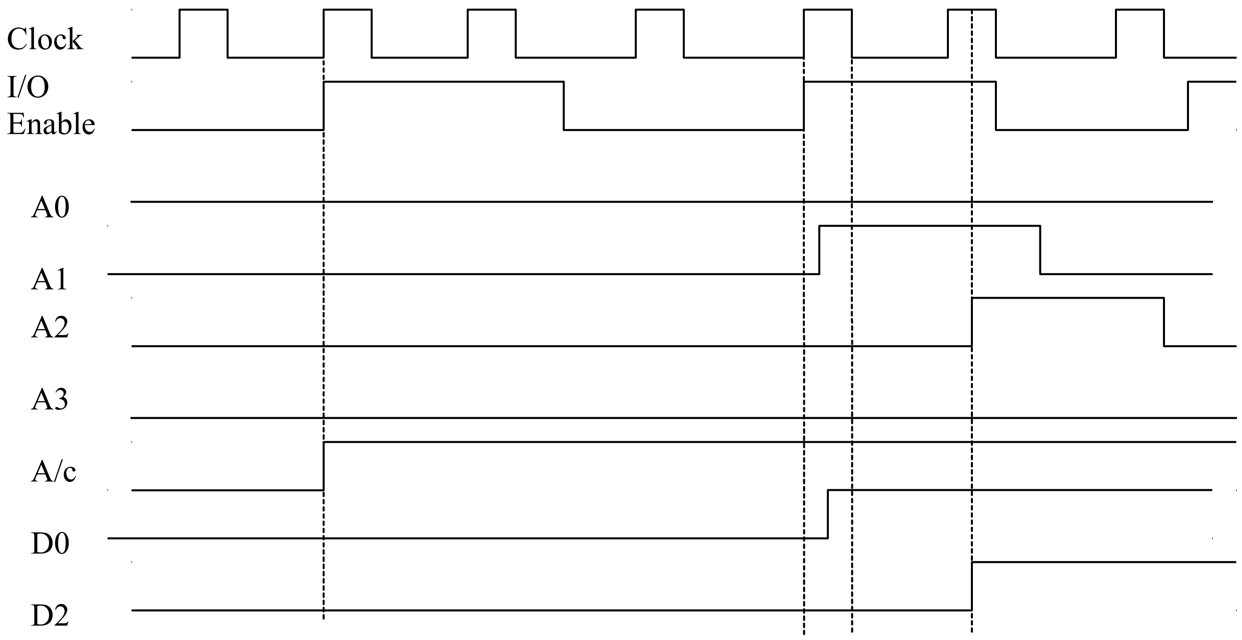
He has also sketched out the following system level diagram for you. He expects you to **detail out the I/O control block and generate the required gate level circuitry** for it. For the moment, you may ignore the internals of the microprocessor. This will be discussed later.



He has also given you an incomplete **truth table** for the output actuator operation –

Address bus lines				**I/O Enable**	**Actuator control lines**							**Comment**
A3	**A2**	**A1**	**A0**		**A/c**	**D0**	**D1**	**D2**	**D3**	**Engine starter**	**Wipers**	
X	X	X	X	0	0	0	0	0	0	0	0	Nothing happens
0	0	0	0	0	0	0	0	0	0	0	0	Nothing happens
0	0	0	0	1	1	0	0	0	0	0	0	A/c get turned on
0	0	0	1	1	0	1	0	0	0	0	0	Door 1 gets locked
0	0	1	0	1	0	0	1	0	0	0	0	Door 2 gets locked
0	0	1	1	1	0	0	0	1	0	0	0	Door 3 gets locked

To help you with the design, another one of your colleagues has provided you the following **timing diagram** –



Tips: To understand the big picture of the final system design which you is in your manager's mind, look at http://www.math.luc.edu/~jdg/w3teaching/comp_260/f02/JPG/p142_04-01.JPG

Activities to do-

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.