# Lab 5: Sequential Logic & State Machines

### Due Date: Monday 4/2/2018 11:59PM

**If you're unsure about a result, be sure to explain your thinking. In general, you should show your work**. A well-reasoned but wrong answer (for example with a minor error) will receive more points than a wrong answer with no context at all.

## Problems (*100 points total*)

1. Draw the state digram for a recognizer which takes in input alphabet $\Sigma = \{0, 1\}$ and accepts all strings which contain the substring 0101. Draw the state transition table for the transition function $\delta$ as well.

2. You're going to design a state machine for a traffic light. There are three LED arrays (green, yellow, red) controlled by a timer circuit and a state machine. The timer circuit has one input (RESET) and one output (FIRED). The timer is hardwired to countdown for some pre-determined amount of time (e.g., 40 seconds). When the timer expires, the timer circuit will assert its FIRED signal. It will stay in this state until its RESET signal is asserted, and when the RESET is then deasserted, it will start counting down again. Your state machine should have FIRED as an input, and RESET and the 3 LED enable lines as outputs. Your state machine will perform a transition on every clock cycle, so take care not to transition to a new state until the timer indicates that you should.

3. Draw out the design of a device which stores a number (say, a 4-bit number here), and when its write enable (WE) input is one, it accepts an input bit on its right-hand side and moves every other bit to the left. The left-most bit in this case will be discarded. **Hint:** you'll need to use flip-flops.

4. Consider the input alphabet $\Sigma = \{a, b, c, d, 0, 1, 2, \text{``''}\}$. The last element represents a single whitespace character. Draw a state diagram for an FSM that recognizes the following regular expression[1]

```
b[^bc]\d+
```

[1] Consider going through this linked tutorial on regular expressions. These are *incredibly* useful in menial programming and scripting tasks. `http://regexone.com`. All modern scripting languages support them for text processing/searching and often-used command line utilities like grep, awk, sed, and find understand them as well.

5. Repeat with this regex:

```
a[b-d].*\dc
```

# Hand-in Instructions

**Make sure to put your name on your submission. Submissions without names will be given zero points! For code, this means put a comment at the top of your C file with your name on it.**

**Physical** : If you're submitting a written copy, hand it to one of the TAs or to the instructor. You can also leave it in the instructor's mailbox in the CS department office, but make sure to get it time stamped when you do (see the "Submitting Work" section of the syllabus).

**Digital** : If you would like to submit an electronic copy, note that I will only accept PDF files (no Word docs please). Again, see the "Submitting Work" section of the syllabus. Please do not take a poorly lit picture of your assignment. Your grade will suffer commensurately with our inability to read your work. Once you have a PDF, you should submit it on `fourier`. You should name your file `yourid-lab5.pdf` where `yourid` is the thing in front of the @hawk.iit.edu in your e-mail address.

You can first get your PDF (for example, for me it might be called `kh123-lab5.pdf`) onto `fourier` like so:

```
[me@mylocalmachine]$ scp kh123-lab5.pdf kh123@fourier.cs.iit.edu:
```

Then you can login to `fourier` via ssh and submit it:

```
[kh123@fourier]$ cp kh123-lab5.pdf /home/khale/HANDIN/lab5
```