Project 2

Due February 7, 2018 at 11:59 PM

For this project you will be working with a partner. Note: all members of the group are not guaranteed equal credit. You will be using Logisim for this project. The project will be broken up into three subcircuits. Submit a README.txt, and interactive grading signup with your circuit file compressed together in a tgz (tar gzip) file. Only one member of the group needs to submit the tgz file. Your README file must have your name, SID number, partner's name, partner's SID number, a brief description of what circuits work/don't work, and a list of sources you used for designing of your circuit (you do not need to list the book or lecture notes it is assumed these have been used). You may use any of the built in components of Logisim, except for those in the Arithmetic group. All class projects will be run through MOSS like software to determine if students have excessively collaborated. Excessive collaboration, or failure to list external sources will result in the matter being referred to Student Judicial Affairs.

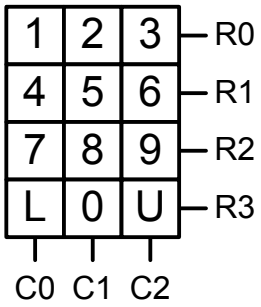

Figure 1. Control Panel

1. The control for the digital lock is a 12-button panel with 7 outputs R0..R3 and C0..C2. Figure 1**Error! Reference source not found.** shows the layout of the control panel. When a button is pressed the column line C0, C1, or C2 will output a 1, and the row line R0, R1, R2, or R3 will output a 1. In order to simplify subsequent subcircuits you will be converting the outputs of the control panel into an output button code and a valid bit. If only a single button is pressed the valid bit V will be 1, and the button code bits B3..B0 will be the value of the button 0 = 0000, 1 = 0001, …, L=1010, and U=1011. If multiple buttons are pressed or no button is pressed V will be 0.

```
Inputs:     R0 R1 R2 R3 C0 C1 C2
Outputs:    B3 B2 B1 B0 V
Circuit:    Panel
```

2. The input from the control panel must be buffered so that the subsequent logic can distinguish between the multiple button presses of the same button and holding a single button. When a valid button press is detected (V=1), the buffer will output the button code for one clock cycle and then return to the no code 1111. The input must go back to invalid button press (V=0) for one clock cycle in between button presses. The second button press

can be ignored of a two-button press sequence if the valid bit V does not return to zero first. In practice this will never happen since the clock CLK is much faster than a human can press.

```
Inputs:     B3 B2 B1 B0 V CLK
Output:     C3 C2 C1 C0
Circuit:    InputBuffer
```

3. Once the input from the control panel has been buffered, the last step is to design the lock circuit. The lock circuit controls a small motor that will spin to lock or unlock the lock. In order to lock the lock the outputs to the motor must be set so L=1 and U=0 for 4 clock cycles. In order to unlock the lock the outputs to the motor must be set so L=0 and U=1 for 4 clock cycles. After the lock or unlock sequence, the motor must be put into idle with the output L=0 and U=0. During the locking or unlocking of the lock, codes input from the input buffer can be ignored. The lock initializes in the unlocked state with the code 000_{10}.

   The lock can be locked (from the unlocked state) by receiving the L code (1010) twice in a row or by the L code followed by a three digit unlock combination. The double L code will lock the lock keeping the previous three digit unlock combination, the L code followed by the three digit combination will update the unlock combination. Input of the L code during a three digit lock combination will start the sequence over again ignoring the one or two digits that were input. Input of the U code (1011) during a three digit lock combination will return the lock to the unlocked state cancelling the lock process.

   The lock can be unlocked (from the locked state) by receiving the U code followed by a three digit unlock combination that matches the previously stored unlock code. If combination does not match, the lock returns to the locked state. Input of the L code during the three digit unlock combination will return the lock to the locked state with the previously input digits being ignored. Input of the U code during the three digit unlock combination will start the process over, where the lock will be waiting for the three digit unlock combination.

```
Inputs:     C3 C2 C1 C0 CLK
Output:     L U
Circuit:    Lock
```