Dept. of Computer Science and Engineering
CSE3201 – Digital Logic Design
Lab 8
Design and Implementation of a Synchronous Control Circuitry for a Digital Lock

In this lab you will use state diagrams in order to design, implement, and test a control circuit to control the operation of a digital lock.

Introduction

A very simple and basic digital lock control circuit is a circuit that receives an input sequence (code, password…) and if that is correct, it issues a control signal to open the lock. It also locks the lock and one must provide the correct sequence to open it again.
A simple added feature is an alarm signal if one enters the wrong code.

Input:
The input to the lock is through the KEY[ ] switches. For ease of reference consider the following names

KEY[0] = A
KEY[1] = B
KEY[2] = C
KEY[3] = D

Output:
Two outputs, one is the lock/unlock signal (if 1, it is locked, if 0, it is unlocked). The other is the ALARM signal, to indicate that someone has entered the wrong code.

Specifications

- When D is pushed in any state, except when the ALARM signal is activated, the lock is closed (locked).
- To open the lock, you have to push the following sequence “ABA”.
- If you entered a wrong sequence, the lock is not opened and the ALARM signal goes off.
- Once the ALARM signal is set, you have to enter “AA” followed by “D” to reset the ALARM signal (but the lock remains close).
- If you start entering the wrong code, in order to stop and try again, you have to push D, and then you can start over entering the correct sequence. Note

that this is possible only if you find out that you made a mistake before entering 3 signals/keys. If you entered 3 signals, that will activate the ALARM and you have to enter “AAD” to start over.

- The LOCK/OPEN signal is shown on LEDR[0]. 1 (off) means locked, 0 (ON) means open.
- The ALARM signal is shown on LEDR[1].

In order to test your design, here is a simple test case (you have to do more than that to be sure it works). The cases show the input as a sequence, and the output. If there is no output it means no change, the previous output stays as is.

Input sequence	Output	
D	LOCK = 1	ALARM = OFF
ABA	LOCK = 0	
D	LOCK = 1	
AAA		ALARM = ON
ABA		
AAD		ALARM=OFF
D		
ABD		
ABA	LOCK=0	

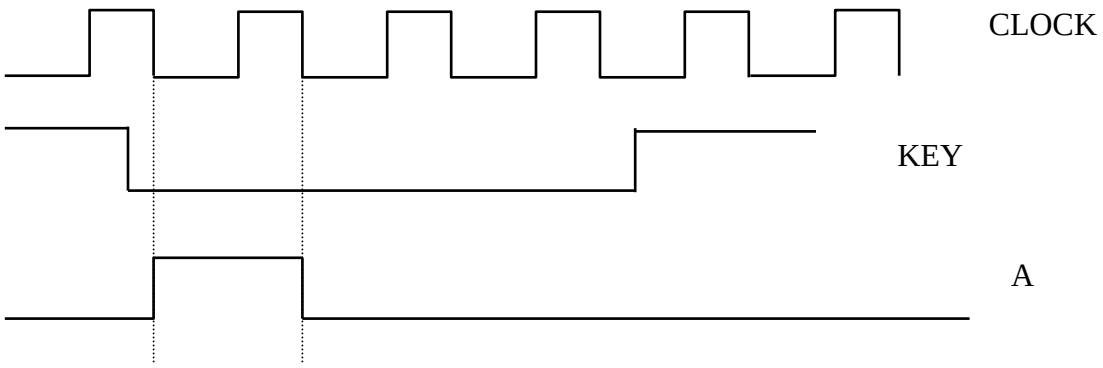
IMPORTANT

This is a synchronous circuit, which means the inputs will be read at the beginning of each cycle.

When you push on of the keys, it goes to 0 for few thousands cycles (if you use the 50MHz clock, and you push A for 1 msec, it will go to zero for 50,000 cycles). That means the circuit will read 50,000 consecutive A signals.

To deal with his, you have to design a one-shot circuits.

The one-shot circuit produces one pulse for a transition from High-To-Low on the input. Thus we create only one pulse when one of the keys is pushed even if it is pushed for a very long time.



As you can see from the above diagram, when one of the keys is pushed (to ground), one pulse is generated at the trailing edge of the clock for a one clock cycle. For every input, you have to use a one-shot with the output of the one-shot going to the finite state machine.
Of course we are assuming that the finite state machine reads the input at the positive edge of the clock.

Another point to consider here is usually when an asynchronous signal (pushing the key) is interfaced with a synchronous circuit, we need double synchronization. That is to say the Key signal will go through two D type flip-flops before it will be considered the input to the one-shot circuit.

Pre-Lab Work

- Draw the State diagram representing your design.
- Indicate what each state represents.
- If you made any assumptions (that does not violate the specifications), you have to clearly state them.
- Write the Verilog code for the circuit.
- Draw your circuit as a block diagram.

Lab report

See the guidelines for the lab report on the Lab section of the course web page. In your report you have to justify the design decisions you made in your design.