ELEC3342: Digital System Design
First Semester, 2022–23

Lab 5

`Objective: Implement FSMs in VHDL`

1 Getting the files

Download the files for this lab from

`http://www.eee.hku.hk/~elec3342/fa22/handout/elec3342lab5.tar.gz`

2 Simple FSM Lock

In this lab, you will design a simple electronic lock by implementing it as a finite state machine (FSM).

1. There are 3 input buttons that are labeled 1, 2 and 3 respectively.
2. A `locked` signal is asserted when the locking mechanism is engaged.
3. Starting from LOCKED state, if the user input the correct *secret* sequence of buttons with 4 numbers, the lock will be unlocked.
4. Once the lock is unlocked, it remains unlocked until it is LOCKED again by the `reset` signal.

Your FSM has the following input and output signals:

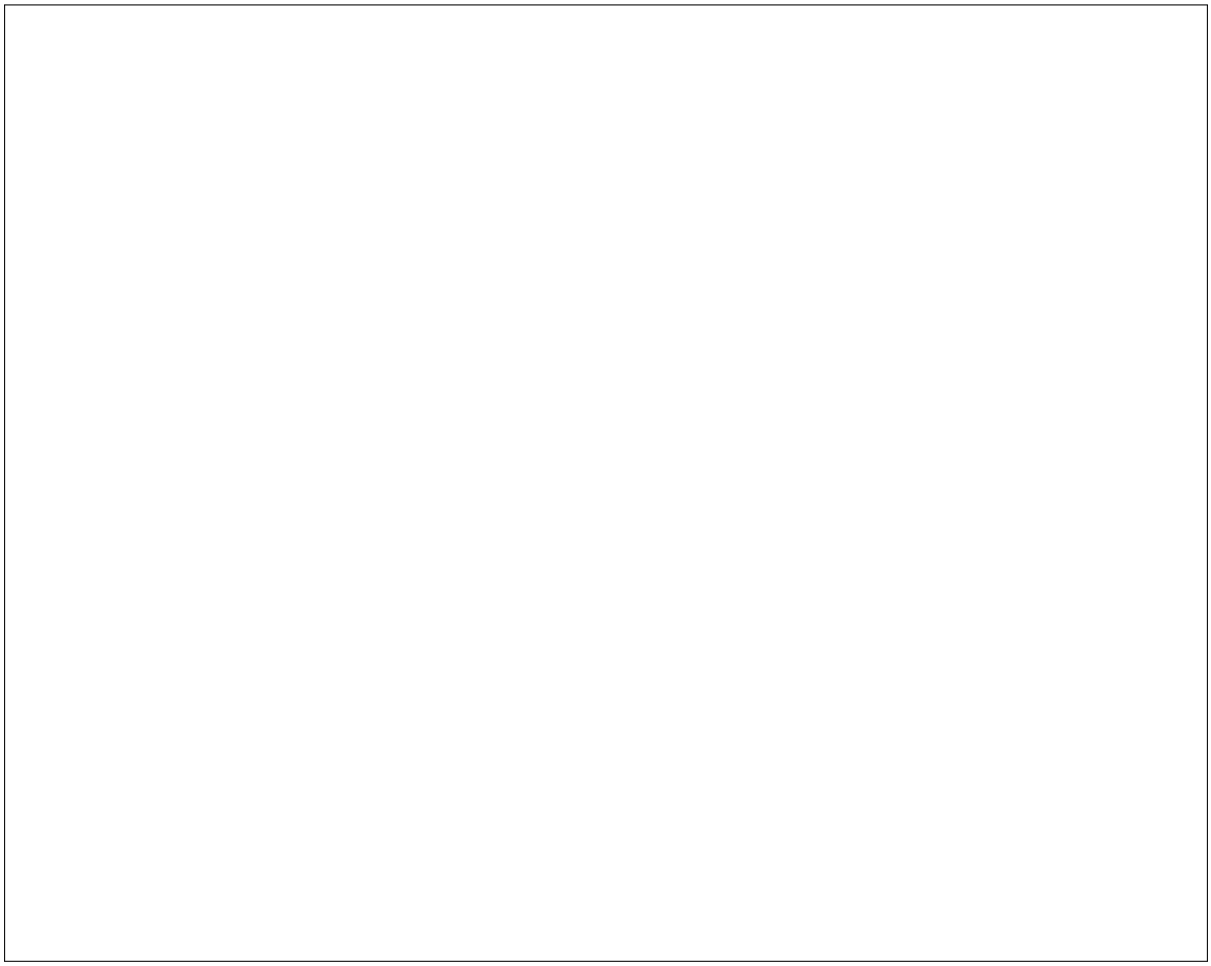
Direction	Name	Description
Input	`btn1`	'1' if button 1 is pressed, '0' otherwise
	`btn2`	'1' if button 2 is pressed, '0' otherwise
	`btn3`	'1' if button 3 is pressed, '0' otherwise
	`engage`	'1' to engage (close) the lock when it is UNLOCKED. No effect otherwise.
Output	`locked`	'1' if the lock is locked, '0' otherwise
	`error`	'1' if an incorrect sequence was input. Held at '1' until lock is reset by pressing both `btn1` and `btn3` together, '0' otherwise

2.1 State Transition Diagram Your FSM has 6 states with the following meanings:

State Name	Description
OPEN	Reset state. Your FSM is reset to this state being opened.
ENGAGED	Lock is engaged (closed), waiting for the first button.
OK1	The first correct button has been pressed
OK2	The second correct button has been pressed
OK3	The third correct button has been pressed
ERROR	An incorrect button is pressed. The FSM will wait in this state until the user press BOTH btn1 and btn3 together.

In the following space, complete the state transition diagram of your electronic lock. Make sure your state machine is a complete state machine — all the necessary transition arcs must be included. For this part, your secret unlock sequence is:

$1 \rightarrow 3 \rightarrow 2 \rightarrow 1$



2.2 Checkoff 1

Answer the following questions:

- Which state does your FSM begins with?
- If your FSM is expecting the button 2 but button 1 is pressed, which state will it end up at?
- If your FSM is expecting the button 2 but BOTH button 1 and 2 are pressed *at the same time*, is your FSM going to accept it as correct button pressed?

Submit your FSM diagram and your answer to the above for this Checkoff.

3 Hardware Implementation

In this part, you will implement your electronic lock designed with VHDL. A template FSM in VHDL together with a testbench has been created for you. To implement your FSM, do the following:

1. Create a new project in Vivado.
2. Add the file `rtl/lock.vhd` into the project as *design source*.
3. Open the file `lock.vhd` and you will see we already put 3 processes there as we talked in the lecture. Now implement your FSM by filling the processes.
4. Add the file `tb/lock_tb.vhd` into the project as *simulation source*.
5. Run the simulation and check the waveform.

3.1 Checkoff 2

Answer the following questions:

- Starting from system power up, which state is your FSM at?
- When your FSM is in the OK1 state, and all 3 input buttons are pressed concurrently, what will happen to your FSM?
- Trick question: with your hardware implementation, is there a way you can unlock the FSM if you have forgotten the secret code?

Submit your answer above and your `lock.vhd` file for this Checkoff.

4 Submission

Submit your answers and files from Checkoff 1 and 2 on Moodle.

```
<your-zip-file>
├─ lock.vhd
└─ answers to the checkoff questions
```