

What's new in this lab:

| | |
|------|--|
| VHDL | Concatenation, named association, positional association, If_then_elsif_else, case_when, enumeration type. |
| EDA | State machine viewer. |
| FPGA | PS/2 port. |
| RTL | Sequential circuits, Finite state machines, handshaking, Shifter. |

Exercise #1

- Download lab3_1.qar from ftp://ele.uri.edu/outgoing/jcl/306/lab3_1.qar.
- Restore, and compile the project, and then program the DE2 board.
- Hook up the keyboard to the DE2's PS/2 port. The sequential circuit, lines 40-55, was designed to detect when the key is released. When a key is released, its "scan_code" is added to the 7-segment displays while the old display contents are shifted to the left. *Line 52 is how the shift is accomplished by using the concatenation operator, "&".
- Note the differences between the port maps between "keyboard" (lines 21-29) and "sevenseg" (lines 31-28). The component "keyboard" uses *named associations*, while "sevenseg" uses *positional associations*.
- The *Boolean* signal "status" is used to decide which state the circuit is in. According to the keyboard's behavior, when a key is released, the scan_code is followed by x"F0".
- Describe the behavior of this sequential circuit using either a state diagram or a flowchart. This will be included in the report.

Exercise #2

- Download lab3_2.qar from ftp://ele.uri.edu/outgoing/jcl/306/lab3_2.qar.
- Restore, and compile the project, and then program the DE2 board.
- This circuit uses signals "ready" and "ack" to establish handshaking with the keyboard interface (keyboard.vhd). Note that the circuit in lab3_1 hardwired "ack" to logic 1; or disabled the handshaking. The sequential circuit goes through six states to make distinction between "make code" and "break code". The scan_code is accepted when the key is pressed. Note that, in this project, the signal "status" is a local integer variable representing the states.
- Use RTL viewer to see the circuit details. The circuit should occupy two pages. Print the RTL view out for the report. Also try "Tools → Netlist Viewers → State Machine Viewer". You should receive a message saying "This Design has no State Machine"!
- Describe the behavior of this sequential circuit using a flowchart, to be included in the report.

Exercise #3

- Download lab3_3.qar from ftp://ele.uri.edu/outgoing/jcl/306/lab3_3.qar.
- Restore, and compile the project, and then program the DE2 board.
- This circuit behaves exactly the same as that in "lab3_2". However, the sequential circuit is now described using the "CASE_WHEN" sequential statement. The state variable "SV" is defined as type "state" (line 19), while "state" is defined as an enumerated type with all the state names defined (line 18). With this method, the six states of this circuit need not be assigned with a number, but a name, the compiler will do the state assignments.

- Use RTL viewer to see the circuit details. The circuit should occupy just one page. Print the RTL view out for the report.
- The sequential circuit is now recognized as a “state machine” and will show up as a yellow block. Double click on the yellow will bring out the “State Machine Viewer.” You could achieve the same by selecting the “state machine viewer” command. Print out the state diagram and add to it the conditions on all edges. All the conditions are listed at the bottom. By clicking on an edge in the top state diagram, the corresponding condition is highlighted.

Assignment (lab3_4)

- Create an electronic lock with a 4-digit (either decimal or hexadecimal digits) code. The status of the lock is displayed on the 7-seg displays as “close” or “open”. The code is entered via the PS/2 keyboard. Whenever the “enter” key is pressed, the four digits entered previously will be examined against the “predefined secret code” and decide on the lock’s status. This means you can continue to enter numbers (0-9 or 0-F), but only the last four digits before the enter key will count. You should work out the algorithm or flowchart first before process to design your FSM. Reset the circuit will make the lock goes back to “close”.

Lab 3 Report:

Title page: Lab title, section number, your name, student ID and email address.

The body of your report should include a one paragraph introduction. Descriptions of your experiences from the exercise parts which associated with the items required below. The following required items should be embedded with the text explanation not aggregated at the end of your report. Approaches and thinking behind your solution to the assignment. Lesson learned from this lab.

From the Exercise #1:

1. The state diagram or flowchart of the sequential circuit in lab3_1.

From the Exercise #2:

2. Print out of the RTL views of lab3_2.
3. The flowchart of the sequential circuit in lab3_2.

From the Exercise #3:

4. Print out of the RTL views of lab3_3.
5. Print out of the State Machine Viewer with transition conditions added.

From the Assignment part:

6. Flowchart or state diagram of electronic lock.
7. VHDL listing.
8. The “predefined secret code” in your design is obviously hardwired. Come up with at least two possible way to change this secret code by the user.

Also, send the followings to TA:

1. Lab3_4.qar (assignment part)