

2020 COMP3222 Lab 7 Requirements, Marking Guide and Additional Notes

There are 6 marks available for completing Lab 7. You are required to complete and submit solutions for Parts I, II and IV. You are encouraged to complete Part III as well, but we will not mark your solution for this part. We will award up to 1 mark for each correct solution to Parts I, II and IV, as specified in the Lab 7 exercise sheet. In addition, we will award up to 1 mark for each of your paper designs for Parts I & IV, as specified below, and award up to 1 mark overall for your coding style.

To assist with your completion of the lab, project archives containing VHDL file stubs and simulation waveforms for Parts I, II and IV have been provided in the Lab 7 section of the course website.

Note: This lab relies on the use of up to 10 LEDs. The DE0 and DE1 boards differ in the LEDs they provide – the DE0 only has 10 green LEDs, while the DE1 board has 10 red LEDs and 8 green LEDs. You will therefore need to connect to the green LEDs if you are using the DE0 board, or alternatively, connect to the red LEDs if you are targeting the DE1 board. These differences are pointed out in the Lab 7 exercise sheet and the code templates provided with the project archives.

Part I

- Use only logic expressions (see L06/S47) to implement your finite state machine (FSM). Derive the state assigned table and the logic expressions of the state variables for Steps 2. and 6. on paper and submit a neat copy as detailed below.
- A common bug is for the output z to be off by one clock cycle (either too late or too early). We will check that your functional simulation is correct. At the very least, your simulation should include the input and output displayed in Figure 1.

Part II

- Note that the design for Step 2. includes a synchronous reset input.
- For the “state machine processing” setting in Step 7. to work, you must comment out the following lines:

```
attribute syn_encoding : string;  
attribute syn_encoding of State_type : type is ".....";
```

Part IV

- You have been provided with skeleton code that implements and connects the components depicted in Figure 5. Your task for this Part is limited to
 1. Designing the FSM on paper i.e. producing a state diagram, and
 2. Implementing your state diagram within the skeleton code.
- With reference to Figure 5, if you are targeting the DE1 board, the design will display the length counter shift register contents on LEDR(7:4) and the Morse code shift register contents on LEDR(3:0) for the code selected using SW(2:0) while you hold down KEY(1). If you are targeting the DE0 board instead, the design will display the length counter shift register contents on LEDG(7:4) and the Morse code shift register contents on LEDG(3:0). You will need to familiarize yourself with the code that has been provided in order to implement the next state logic and output logic processes.
- You may design either a Moore or Mealy FSM. Your FSM should be minimal. Take care (i) to carefully specify each state (choose a meaningful name for each one), (ii) to determine the conditions for each state transition and to label the edges in your diagram with the names and values of the signals that cause each transition to be followed, and (iii) to specify the outputs that are asserted either upon each transition (Mealy machine) or in each state (Moore machine). The usual conventions for drawing a Moore or Mealy type state diagram as discussed in L06 are to be followed. Your sketch should be neat or it will not be marked.
- Follow the conventional approach of specifying the next state logic as a process.
- Due to the complexity of the design and the desire to keep related logic together, it is advisable to also use a combinational process to implement the output logic. The conventional approach for doing so is to assign all outputs a default value as soon as the process is entered. These default values are then over-written on a case by case basis depending upon the current state (and input condition, if designing a Mealy machine).

Coding style

Apart from correctness, the guiding principle for code style in COMP3222 is understandability and maintainability. Up to ONE mark will be awarded for your coding style on all three assessable parts of the lab. Points to be taken into account include:

- Adherence to the lab specifications
- Alignment between your paper designs and your code
- Appropriate decomposition of a design into sub-components
- Correct VHDL use, including appropriate use of behavioural statements and component instantiation
- Use of meaningful names for entities, architectures, signals and labels
- Proper indentation

Submission

1. Create a *Project* → *Archive* for Parts I, II and IV of Lab 7 – use meaningful names to distinguish each project archive. Include your surname, student number, lab and part number in the archive name e.g. Diessel-3002283-L07P4.qar
2. Create a PDF or JPEG file of the paper design for Parts I & IV. Use the same naming convention as above, but append -design.pdf or -design.jpg
e.g. Diessel-3002283-L07P4-design.pdf
3. Create a zip file including the above 5 files and submit it to...