2019 COMP3222 Lab 11 Notes and Additional Questions

There are 5 marks in total available for this lab.

Part 1 (1 mark)
This part essentially asks you to implement the bit counter (popcount operation) discussed in lectures (see L09/S44-S51). Note, however, that in contrast to the lecture implementation, the circuit you are to implement needs to load the input word *A* in state S1 while *s* is not asserted.

Part 2 (3 marks – 1 mark for your design, 1 mark for implementing the circuit, 1 mark for your simulation)
Before you start coding this part of the lab, you are required to neatly and professionally:

- **List** the pseudocode for your binary search algorithm,
- **Sketch** the ASM chart for your design,
- **Sketch** the data path for your design and label all signals including all internal wires,
- **Annotate** your ASM chart with the names of the signals that are to be asserted/tested during each state or state transition, and
- **Sketch** a timing diagram of your design assuming that the input value is found at the second address that the search explores. The diagram should include the name of the current state and the address and data of the RAM component. Remember that it takes **2 clock cycles** to read from memory using the default settings.

Implement your design but DO NOT map it to the board – that is, have it marked off after you are happy with your simulation (see below). In other words, perform steps 1, 2, 3 and 5 of the lab instructions for Part 2 BUT NOT STEP 4 and finish this part by completing the simulation.

- HINT: Your binary search will require a division by 2, but there is **no division operator** in VHDL! Use a logical right shift to perform the same arithmetic.
- Include a 'DONE' signal in your design that behaves in the same way as in Part 1.
- Prepare a **simulation** that sequentially searches for the values 0 to 64. The odd values from 1 to 63 should be stored in RAM.
    - Use the 'clock' signal setup to configure the 'clock' and 'run' signals
    - Use the 'counter' signal setup to configure the input

Note that a project archive containing skeleton VHDL code, including the interface of the design, a preconfigured and pre-initialized 1-port RAM module, and a simulation waveform file using the provided interface is available for your use. The simulation waveform assumes your design uses 6 clock cycles or less to explore each search address.

Further note for advanced students (not marked): while we are not marking off the circuit operating on the board, you may like to challenge yourself to minimize the number of clock cycles needed to explore each search address. It is possible to compute a search address and to ascertain the value stored at that address every two clock cycles. Getting to one clock cycle would be brilliant! Check the impact on Fmax as you reduce the number of clock cycles used.

Coding style (1 mark)

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 both parts of the lab. Points to be taken into account include:

- Proper indentation
- Use of meaningful names for entities, architectures, signals and labels
- Alignment between your ASM chart for part 2 and your code
- Correct VHDL use
- Appropriate decomposition of a design into sub-components
- Adherence to the lab specifications