INTRODUCTION TO DIGITAL SYSTEMS

SYSC2310 (Fall 2020)
Dr. Mostafa Taha

Lab 5: Sequential Logic Circuits

Objectives

- Design a binary counter and a binary sequence detector

Relation to course outcomes

The work in this lab is related to the two following course outcomes:

- Able to design, implement and analyze sequential logic circuits with flip-flops.

Preparation for the lab

1. Under your 'sysc2310lab/lab5' folder, create a folder 'Ex1', 'Ex2' and 'Ex3' to save your work during this lab.

Exercise 1: Design a Binary Counter

Design, construct and test a 3-bit binary counter that goes through the following sequence of binary states: 0, 1, 2, 3, 4, 5, 6, then back to 0.

Note that the binary state '7' is not used. The counter must be self-starting; that is, if the circuit starts on the invalid state '7', the counter must move to a valid state on the next pulse and continue to work correctly. The design should resemble a FSM without inputs or outputs, using D flip-flops. The counter is controlled only by the clock.

(Pre-lab work) Derive the state table and draw the state diagram of the system.

- In Logisim, please use D flip-flops from the memory library. For help with the D flip-flop, please hover over the input pins to read its function, and refer to (http://www.cburch.com/logisim/docs/2.7/en/html/libs/mem/index.html).
- Please use the clock input from the wiring library. For help with the clock input, please refer to (http://www.cburch.com/logisim/docs/2.7/en/html/libs/wiring/clock.html).
- To check operation of the counter, use a 7-seg displays.
- To check operation of the counter during the invalid states, please use the preset/clear inputs of the D flip-flop to initialize the counter to the invalid state of '7'. Then, proceed with the clock to ensure correct transition to a valid state.

Exercise 2: Design a Binary Sequence Detector

Design, construct and test a binary sequence detector that can be used to search for a specific sequence of binary values within a long stream of bits. Search for the binary sequence "1101".

The sequence detector should be "Overlapping". For instance, the following input sequence will generate the corresponding output:

Input :01101101001

Output :00001001000

Note how one bit can be common between two detected sequences. In other words, after a successful detection of ‘1101’ if the next input is ‘1’, the system should goes to the state ‘11’, not the reset state.

The design should resemble a Moore FSM, using D flip-flops.

(Pre-lab work) Derive the state table and draw the state diagram of the system.

- In Logisim, the circuit has only two inputs: data input x, and the input clock. Use an input clock from the wiring library, and D flip-flops from the memory library. The flip-flops are +ve edge triggered. You should change the input x to the selected new value before changing the clock from low to high (the +ve edge).

Exercise 3 (after-lab): Design using a JK flip-flop

Solve the same problem description of Exercise 2, using JK flip-flops.

- Derive the state table and draw the state diagram of the system.
- Compare the number of gates used in the design using D flip-flops (Ex.2), versus that using JK flip-flops (Ex.3).

This exercise is used for practice on designing sequential circuits using JK flip-flops. Students are asked to solve this exercise on their own, since the topic will be covered after the first two lab sessions. You can ask your TAs for their help during their office hours.

Demonstration:

- In order to get full credit, complete all exercises and be prepared to answer questions asked by the TA during final demonstration.
- Before you leave the lab, even if you could not complete the lab, make sure you sign-out with the TA.
- If you could not complete the lab during the scheduled lab time, finish up the lab on your own time. In this case, the demonstration points will be reduced in accordance to the percentage of work completed.
- If the student is absent or does not show the TA the work completed, 0 points are attributed for this part.