Laboratory 3, Total Points: 30

Due Date: February 21, 2019 (Thursday), 11:59 PM CST

Textbook Sections: Digital Design and Computer Architecture: 3.2, 4.4, and 5.4

In this lab, students will implement a few sequential building blocks of digital systems. Students may refer the above-mentioned textbook sections and Quartus tutorial to complete the lab exercises.

Exercises

1. [7 pts] A JK flip-flop receives a clock and two inputs, J and K. On the rising edge of the clock, it updates the output, Q. If J and K are both 0, Q retains its old value. If only J is 1, Q becomes 1. If only K is 1, Q becomes 0. If both J and K are 1, Q becomes the opposite of its present state. The toggle (T) flip-flop has a clock, one input, and one output, Q. On each rising edge of the clock, Q toggles to the complement of its previous value. Implement a VHDL module for JK flip-flop and use this module to implement a T flip-flop. Submit VHDL code, simulation results, and synthesized circuits for JK and T flip-flops.

2. [8 pts] A shift register has a clock, a serial input S_{in}, a serial output S_{out}, and N parallel outputs $Q_{N-1:0}$, as shown in Figure 1. On each rising edge of the clock, a new bit is shifted in from S_{in} and all the subsequent contents are shifted forward. The last bit in the shift register is available at S_{out}. Shift registers can be viewed as serial-to-parallel converters. The input is provided serially (one bit at a time) at S_{in}. After N cycles, the past N inputs are available in parallel at Q. Implement a generic N-bit shift register and use it for 4 and 8-bit shift register. Submit VHDL code, simulation results, and synthesized circuits.

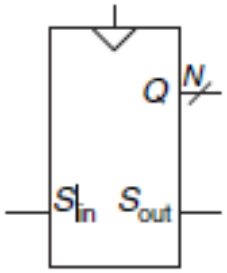

Figure 1: Shift Register

3. [5 points] An N-bit Johnson counter consists of an N-bit shift register with a reset signal. The output of the shift register (S_{out}) is inverted and fed back to the input (S_{in}). When the counter is reset, all of the bits are cleared to 0. Implement a generic N-bit Johnson counter and use it for 3 and 5-bit Johnson counter. Submit VHDL code, simulation results, and synthesized circuits.

4. [10 points] Gray codes have a useful property in that consecutive numbers differ in only a single bit position. Table 1 lists a 3-bit Gray code representing the numbers 0 to 7. Design a UP/DOWN 3-bit modulo 8 Gray code counter with no inputs and three outputs. When reset, the output should be 000. If UP=1, the output should advance to the next Gray code on each rising clock edge. After reaching 100, it should repeat with 000. If UP = 0, the counter retreats to the previous number. Use **if** and **case** structs in VHDL to implement the exercise.

Number	Gray code		
0	0	0	0
1	0	0	1
2	0	1	1
3	0	1	0
4	1	1	0
5	1	1	1
6	1	0	1
7	1	0	0

Table 1: 3-bit Gray Code