ECE 380: PRELAB #10: Finite State Machine

Flip-flops are often used to design finite state machines (FSMs). Example usages can be categorized into sequence generators or sequence detectors. One common application of counters that also incorporates elements of sequence detection and commonly appears in the various embedded systems is that of a "watchdog" timer. Essentially a watchdog timer is an independent, programmable, free running counter that can generate a reset signal if the counter reaches a preset value. Most microcontrollers provide one or more built-in watchdog timer functions with varying capabilities. For example, the Freescale 68HC9S12DP256B microprocessor, used in ECE-383, incorporates a system-wide watchdog timer that their documentation denotes as the COP (for **C**omputer **O**perating **P**roperly).

Prelab tasks

The requirements for this lab consist of completing designs, printing any necessary circuit diagrams, performing Quartus simulations, and completing the laboratory report. Please include functional simulations of the VHDL designs in the lab report.

Before your lab session, you need to complete the following tasks:

1. **Design A:** Using behavioral VHDL, design a **Moore-type** finite state machine that detects input test vector that contains the sequence of '100'. If the sequence '100' is detected, the output Z should go high. The input is to be named W, the output is to be named Z, a Clock input is to be used and an active low reset signal (Resetn) should asynchronously reset the machine.

   a) Draw the Moore-type model state diagram for the FSM.
   b) Write the VHDL code to implement the FSM.

2. **Design B:** Using behavioral VHDL, design a **Mealy-type** finite state machine that detects input test vector that contains the sequence of '100'. If the sequence '100' is detected, the output Z should go high. The input is to be named W, the output is to be named Z, a Clock input is to be used and an active low reset signal (Resetn) should asynchronously reset the machine.

   a) Draw the Mealy-type state diagram for the FSM.
   b) Write the VHDL code to implement the FSM.

You need to complete VHDL codes and functional simulations in Quartus, before your lab sessions.

Pre-Lab (30 pts)	**Score**	**TA initial**
30 pts Designs		

Report (70 pts)	
10 pts Introduction	
10 pts Procedures	
20 pts Results	
30 pts Conclusions	
Lab Grade (100 pts)	