EE 3120: Digital Circuits Laboratory
LAB 4: SEQUENTIAL LOGIC DESIGN USING VERILOG

Objectives

The objectives of this laboratory assignment are:

- To understand and use different options available in Verilog HDL to describe sequential logic circuits (finite state machine);
- To design and implement counters;
- To design and implement practical finite state machines.

Laboratory Instructions

- Create the Verilog source file(s) for your designs before coming to the lab. You can use the Xilinx HDL editor or any text editor to create your files;
- Use the Xilinx software to create a new project and then add your Verilog (.v) file(s) to that new project;
- Perform functional simulation of your design and have it checked by the lab instructor or your TA before proceeding with the implementation;
- In case you modify your original Verilog source file, remember to copy it back to your floppy disk
- If the circuit works as expected, implement it using the prototyping board assigned to you;
- Use switches and LEDs available on the breadboard to test and demonstrate your circuit to the lab instructor or your TA;
- Before you leave the lab please remove any files or directories you created on your lab PC and leave your workplace at least as clean and tidy as you found it.

Pre-lab Work

Complete your design and its Verilog implementation and bring your Verilog program on a floppy disk. You can use any text editor to create an Verilog file. If you have any problems with Verilog syntax and other pre-lab related issues, please resolve them with the instructor or your TA before coming to the lab. Your TA may not be able to help you with these issues during the lab session.

Laboratory Report Instructions (due at the start of next lab)

- Your report should be typed and prepared as per the guidelines given on the EE 3120 web page.
- For each design of this lab, submit the following:
    - Detailed state diagram for each sequential logic circuit.
    - Documented listing of your Verilog source file(s) with appropriate pin assignments for the top level files in the same or separate file(s).
    - Simulation waveforms: Label the waveforms to indicate proper operation of your circuit(s).

DESIGN PROBLEMS

Using the Xilinx CAD tools, design, test and demonstrate logic circuit(s) which implement the following sequential logic circuits. Your circuits should be as small as possible.

1. Oven Controller:

Assume that an oven has an electronic thermometer attached to circuitry with two outputs: one (H) is asserted when the temperature is above h, and the other (L) is asserted when the temperature is below l. Design a circuit that will turn the oven on when the temperature falls below l; turn the oven off when the temperature rises above h; and when the temperature is between l and h, leave the oven in the condition it was in before l or h was reached. Design this controller and demonstrate its working on the prototyping board assigned to you.

2. Stack Controller:

A stack controller is a digital circuit in a computer that implements last-in first-out (LIFO) storage for temporary data. It operates analogously to a stack of plates where you can put a new plate on the top of the stack or you can take a plate off the top of the stack. For this circuit, three operations can be performed:

- PUSH adds an element to the top of the stack.
- POP removes the top element from the stack.
- EXCHNG swaps the top two elements of the stack.

When using a stack, 2 types of errors can occur:

- Underflow occurs when a POP operation is attempted on an empty stack or when an EXCHNG is attempted on a stack with fewer than 2 elements on it.
- Overflow occurs when a PUSH operation is attempted on a stack which already contains its maximum number of elements.

Design a circuit which performs error checking on a stack which can contain a maximum of 5 elements. The input signals are PUSH, POP and EXCHNG. You may assume that at most one of the inputs will be asserted in any clock cycle. The outputs of your circuit should be OVF and UNF which indicate overflow and underflow, respectively.
Draw the Mealy state diagram for the memory controller and implement it on the prototyping board using Verilog HDL.