EC605: Computer Engineering Fundamentals Fall 2016
Lab 3: Finite State Machines

Sequential Logic

Goals

- Introduction to sequential logic Verilog.
- Introduction to state machine design in Verilog.

Overview

In this lab you will design a counter, and a state machine that will light a sequence of LEDs in response to the counter wrap-around (i.e. reaching zero).

Tasks
Task 1: 8-bit Counter

1. Implement an 8-bit bi-directional counter with the following inputs and outputs:
   Inputs: clk, rst, direction, [7:0] maximum;
   Output: [7:0] count.

   - The rst input resets the counter to zero.
   - The direction input determines if the counter counts up (direction = 1) or down (direction = 0).
   - The maximum input determines the maximum value that the counter may reach.
   - As an up counter, every positive clock edge, the counter increments by one, until reaching the upper limit, at which point it wraps around to zero.
   - As a down counter, every positive clock edge, the counter decrements by one, until reaching zero, at which point it wraps around to the upper limit.
   - Note that the output of the counter should also be stored in a register.

2. Implement a testbench and simulate your design.

Task 2: Clock Divider

1. We would like to utilize the clock provided in the Xilinx FPGA board to control our circuit. However, the frequency of this clock is too fast for the eye to detect. A clock divider takes an input clock of a given frequency and generates an output clock of a different frequency. We have used a clock divider block in Lab 1. Implement a clock divider that receives clk and rst as inputs, and outputs a clock with a frequency of 1Hz. The input clock frequency is 100MHz.

2. Implement a testbench and simulate your design.

Task 3: Wrap-around LEDs

1. A state machine receives, in addition to the clock and reset inputs, two push buttons (btn0 and btn1), and the value of an 8-bit counter as inputs, and generates 4 LED outputs (or a single 4-bit LED output), LED3, 2, 1, and 0.

   The state machine operates as follows:

   - On reset, the state machine moves to state 0, the initial state.
   - From the initial state, if btn0 is pressed, the state machine waits for a counter to wrap-around (reach zero), and turns on LED 0. For the next wrap-around it turns on LED 1 only, then LED 2, and finally LED 3, then returns to the initial state.
   - From the initial state, if btn1 is pressed, the state machine waits for a counter to wrap-around (reach zero), and turns on LED 3. For the next wrap-around it turns on LED 2 only, then LED 1, and finally LED 0, then returns to the initial state.

   Draw a state diagram of your state machine.

2. In a new project, implement the above state machine. Your project should include the counter and clock divider from the previous tasks.

   The clock input of the state machine is the 1 Hz clock, which is the output of the clock divider. In addition, the state machine receives two push buttons and the value of your counter as input, and generates 4 LED outputs (or a single 4-bit LED output).

   Implement the state machine in Verilog. Refer to lecture notes for a Verilog state machine implementation example.

3. The Top module of the project should contain the following inputs and outputs which will be connected to the board:

   Inputs:
   clk - 100 Mhz input clock from board
   rst - push button which resets state machine elements
   [1:0] push button - two push buttons which when pressed control the direction
   [7:0] switches - The binary value of these switches control the maximum of the counter

   Outputs:
   [3:0] LEDs - Four LEDs used for State Machine

Deliverables

- Submit your Verilog code and testbench for each task. Also submit a pdf with a waveform and a description of the tests done for each task, and your state diagram.
- Sign-up to demo your design on the Xilinx FPGA board to a TA. Come prepared to answer questions on your design.