Lab #3: Finite State Machines

1 Introduction

In this lab you will learn how to describe finite state machines (FSMs) in VHDL. Specifically, you will design a special kind of a 4-bit counter using an FSM. This counter is not a regular up-counter. In addition to being bi-directional, i.e., it counts up and down, it goes through the following sequence of numbers:

$$1 \longleftrightarrow 2 \longleftrightarrow 4 \longleftrightarrow 8 \longleftrightarrow 3 \longleftrightarrow 6 \longleftrightarrow 12 \longleftrightarrow 11 \longleftrightarrow 5 \longleftrightarrow 10 \longleftrightarrow 7 \longleftrightarrow 14 \longleftrightarrow 15 \longleftrightarrow 13 \longleftrightarrow 9 \,(\longleftrightarrow 1 \longleftrightarrow 2 \longleftrightarrow 4 \ldots)$$

Note that the right-arrow (i.e., $\rightarrow$) denotes the upward direction whereas the left-arrow (i.e., $\leftarrow$) denotes the downward direction of the counter. See Section 8.7.5 of the textbook for a similar counter.

Pushbuttons, slide switches and 7-segment LEDs will be used during this lab to control the counters when running on the Altera DE1-SoC board.

2 Learning Outcomes

After completing this lab you should know how to:

- Design an FSM for a counter
- Perform functional simulation of the FSM using ModelSim
- Test the FSM on the Altera board

3 Finite State Machine

Similar to the previous lab, the FSM circuit works when the enable signal is high. The direction signal denotes the counting direction. Note that the counter should be initialized according to their direction when the asynchronous active-low reset signal is asserted. Use the leftmost value (1) in the counting sequence for the upward direction and the rightmost value (9) for the downward direction.

First, draw the state diagram for the counter. Then using the state diagram describe the counter in VHDL. Use the following entity declaration to describe a Moore-style FSM implementing the above counter:

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.NUMERIC_STD.ALL;
entity gNN_FSM is
    Port (enable     : in  std_logic;
          direction  : in  std_logic;
          reset      : in  std_logic;
          clk        : in  std_logic;
          count      : out std_logic_vector(3 downto 0));
end gNN_FSM;
```

See Section 8.4 of the textbook for a discussion on how to design FSMs using VHDL.

Once you have your circuit described in VHDL, you should simulate it. Write a testbench code and perform a functional simulation for your VHDL description of the FSM.

4 Multi-Mode Counter

In this part, you will test your FSM circuit on the DE1-SoC FPGA board using the clock divider and 7-segment decoder circuits from the previous labs. You will also use pushbuttons, slide switches to control the functionality of the FSM. Pressing pushbuttons PB0, PB1 and PB2 starts/resumes, stops/pauses and resets the counter, respectively. When these buttons are released, the circuit has to remain at the new state denoted by their corresponding function. For example, when PB1 is pushed and then released, the FSM circuit pauses the count until told otherwise by pushing other pushbuttons. Therefore, you need a memory to hold the state imported by the pushbuttons. Note that the output of a pushbutton is high when the button is not being pushed, and is low when the button is being pushed. The counter direction is entered from one of the slide switches on the board. The first two 7-segment displays (i.e., HEX1-0) are used to show the count in decimal format. Use the clock divider circuit from the previous lab to show the current count on the 7-segment displays at every 1 second.

To test the FSM circuit on the FPGA board, you will need to create instances of your gNN_clock_divider and gNN_7_segment_decoder to display the current value of the counter after each 1 second. Describe the system testing the FSM circuit on the FPGA board, which is hereafter referred to as multi-mode counter, in VHDL using the following entity declaration:

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.NUMERIC_STD.ALL;
entity gNN_multi_mode_counter is
    Port (start      : in  std_logic;
          stop       : in  std_logic;
          direction  : in  std_logic;
          reset      : in  std_logic;
          clk        : in  std_logic;
          HEX0       : out std_logic_vector (6 downto 0);
          HEX1       : out std_logic_vector (6 downto 0));
end gNN_multi_mode_counter;
```

Compile the multi-mode counter circuit in the Quartus software. Once you have compiled the multi-mode counter circuit, it is time to map it on the Altera DE1-SoC board. Perform the pin assignment for HEX displays, slide switches and pushbuttons according to the DE1 user's manual. Make sure that you connect the clock signal of your design to the 50 MHz clock frequency. Program the board and demonstrate your work to the TA. You should be able stop, start and reset your counter using the pushbuttons. Also, you should be able to change the counting direction using the slide switch on the board.

5 Deliverables and Grading

5.1 Demo

Once completed, you will demo your project to the TA. You will be expected to:

- fully explain how the HDL code works,
- perform functional simulation using ModelSim, and
- demonstrate that the FSM circuit is functioning properly using the pushbuttons, slide switches and 7-segment LEDs on the DE1-SoC board.

5.2 Written report

You are also required to submit a written report and your code on myCourses. Your report must include:

- The state diagram of your FSM.
- A description of the FSM circuit.
- A discussion of how the FSM circuit was tested, showing representative simulation plots. How do you know that these circuits work correctly?
- A description of the multi-mode counter circuit.
- A discussion of how the multi-mode counter circuit was tested on the FPGA board.
- A summary of the FPGA resource utilization (from the Compilation Report's Flow Summary) and the RTL schematic diagram for the multi-mode counter circuit. Clearly specify which part of your code maps to which part of the schematic diagram.

Finally, when you prepare your report have in mind the following:

- The title page must include the lab number, name and student ID of the students, as well as the group number.
- All figures and tables must be clearly visible.
- The report should be submitted in PDF format.
- It should document every design choice clearly.
- The grader should not have to struggle to understand your design. That is,
  - Everything should be organized for the grader to easily reproduce your results by running your code through the tools.
  - The code should be well-documented and easy to read.

Grading Sheet

Group Number:
Name 1:
Name 2:

Task	Grade	/Total	TA Signature
VHDL code for the FSM circuit		/40	
Creating testbench code for the FSM circuit		/5	
Functional simulation of the FSM circuit		/5	
VHDL code for the multi-mode counter circuit		/10	
Testing the multi-mode counter circuit on the DE1-SoC board		/40	
Total		/100	