Digital Logic Group 69

ECSE 222

Anthony Porporino: 260863300

Tyler Watson: 260867260

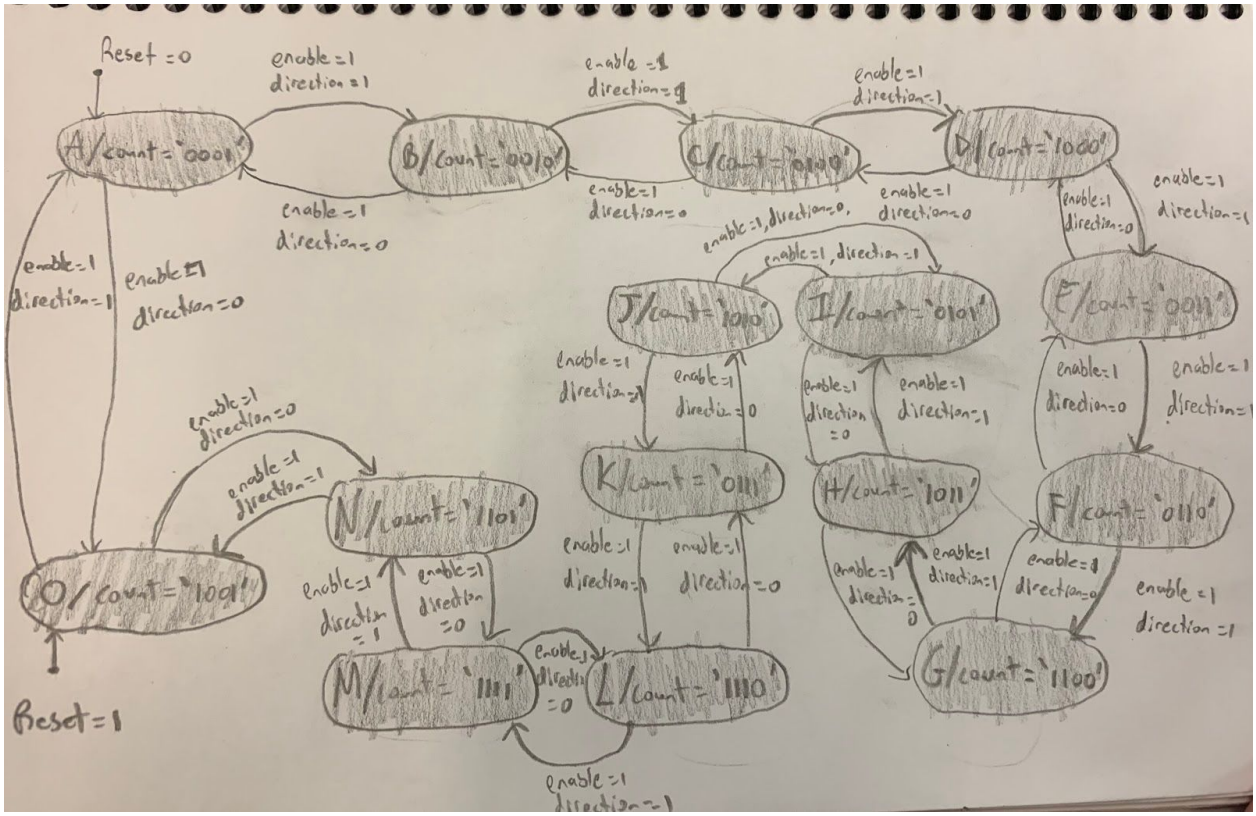
Lab Assignment 3: Written Report

Prof. Ioannis Psaromiligkos

McGill Faculty of Engineering

April 12, 2019

State Diagram:



Description of the FSM circuit:

A finite state machine is used to design a 4 bit counter that counts through the decimal digits 1-2-4-8-3-6-12-11-5-10-7-14-15-13-9. The finite state machine takes in as input 4 std_logic variables; enable, direction, reset and clock. The output is an std_logic vector (3 downto 1) that is the binary representation of the current state of the count. If the enable is 1, the output will switch to the next state on every rising edge of the clock and will go in the direction depending on the direction variable (If direction is 0 it counts down and 1 counts up). Reset is an asynchronous active low signal so when it is 0, the state is initialized to the state with 1 or 9 depending on the value of direction.

A discussion of how the FSM circuit was tested, showing representative simulation plots.

The FSM was tested using a testbench writer (see attached code) that set enable and direction to 1. Then using a wait time changed reset from 0 to 1. Using ModelSim, the outputs were then examined as a wave shown in figure 1.

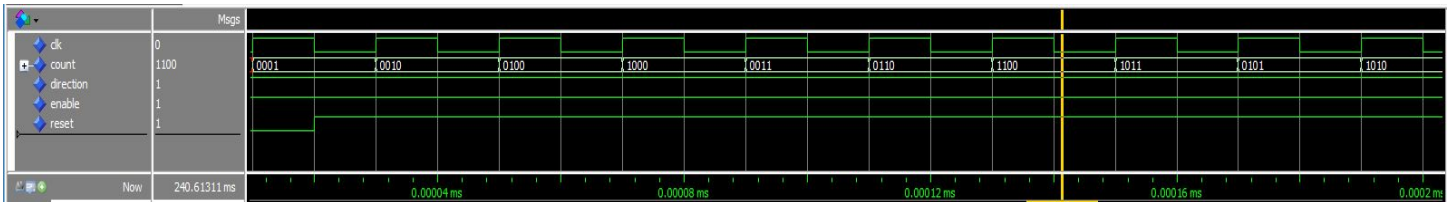


Figure 1: FSM Simulation Wave

How do you know that these circuits work correctly?

By looking at the simulation wave from ModelSim it is easily seen that the outputs for the inputs in the testbench correctly corresponded to the expected output values.

A description of the multi-mode counter circuit:

The multi-mode counter circuit takes in as input a stop, start, reset, clock, and direction std_logic variable. The output of the circuit is 2 std_logic_vectors (6 downto 1) that correspond to the seven segment decoder result of the left and right digit of the count. First a component of the clock divider, finite state machine, seven segment decoder and digits circuit was instantiated. Two std_logic_vector (3 downto 1) signals were then initialized to represent the decimal number corresponding to the left and right digits and 4 std_logic variables were initialized to represent count-out, counter state, start state and enable_0. The enable_0 signal is mapped to the output of the clock divider with inputs, start, reset and clock. A process block then determines the start_state variable based on inputs start stop and reset. This start_state variable is then passed as input to the finite state machine along with direction, reset and clock. The output that corresponds to the binary representation of the current state is mapped to the count-out signal.

The digits component will receive input count-out and return two std_logic_vectors (3 downto 1) that correspond to the left and right digit of the number of the current state. Lastly these two numbers are inputted into the seven segment decoder and it returns the output of the multi-mode circuit, 2 std_logic_vectors ready to be mapped to the FPGA board.

A description of how the multi-mode counter circuit was tested on the FPGA board:

We first wrote a testbench file (see attached code) that had a clock process and stim_process that forced direction = 1, start = 1, stop = 1, and reset = 0. Using a wait time the values were then changed, start = 0, stop = 1, reset = 1. Using ModelSim, the outputs were then examined as a wave shown in figure 2.

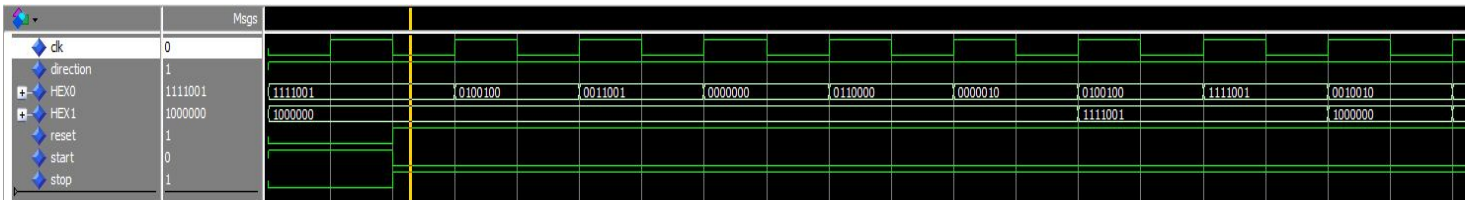


Figure 2: Multi-Mode circuit simulation wave

After seeing the correct output values for the inputs we then mapped signals to pins on the FPGA board and saw the values appear on the board itself.

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:

Flow Summary	
Flow Status	Successful - Thu Apr 11 09:52:58 2019
Quartus II 64-Bit Version	13.1.0 Build 162 10/23/2013 SJ Full Version
Revision Name	g69_lab3
Top-level Entity Name	g69_multi_mode_counter
Family	Cyclone V
Device	5CSEMA5F31C6
Timing Models	Preliminary
Logic utilization (in ALMs)	28 / 32,070 (< 1 %)
Total registers	23
Total pins	19 / 457 (4 %)
Total virtual pins	0
Total block memory bits	0 / 4,065,280 (0 %)
Total DSP Blocks	0 / 87 (0 %)
Total HSSI RX PCSs	0
Total HSSI PMA RX Deserializers	0
Total HSSI TX PCSs	0
Total HSSI TX Channels	0
Total PLLs	0 / 6 (0 %)
Total DLLs	0 / 4 (0 %)

Figure 3: Compilation Report's Flow Summary:

The RTL schematic for the multi-mode circuit is shown below in figure 4. It is straightforward and each box represent a different vhdl file. All files are attached to the submission and have the same name as in the figure.

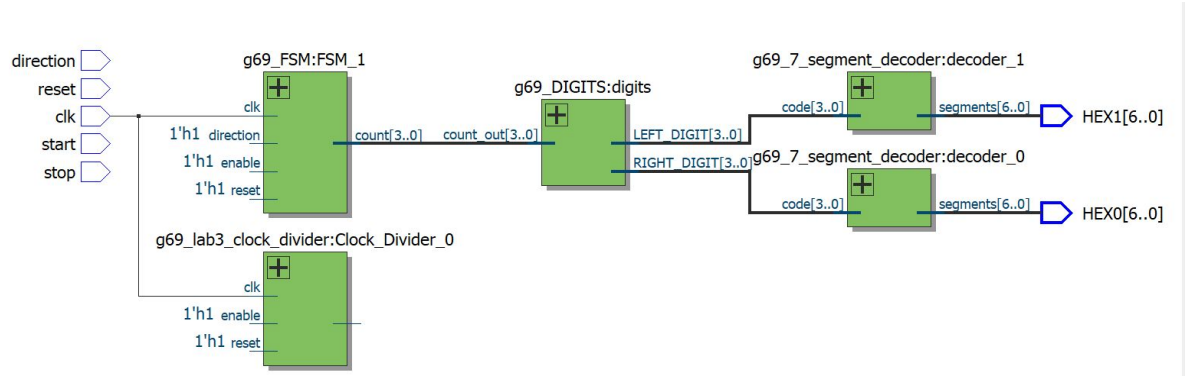


Figure 4: RTL schematic: