THE UNIVERSITY OF ALABAMA IN HUNTSVILLE
Department of Electrical and Computer Engineering

CPE 322 Digital Hardware Design Fundamentals

Spring Semester 2015: Homework Assignment #2
Basic Sequential Digital Design Methodology Review --Design of a Two's Complement Negation Element

Purpose

The purpose of this homework is to review traditional sequential Finite State Machine, FSM, design methodologies that were introduced in the EE 202, Digital Logic Design class and reviewed in Chapter 1 of the course text. The homework also illustrates how such designs might be created using ROM type lookup tables and multiplexers to implement the combinational portion of the logic. Implementing combinational logic in this way is a common method employed by CAD tools when they synthesize designs for FPGA implementation.

Assignment

This assignment requires that students develop multiple Mealy and Moore Finite State Machine implementations of a functional unit that performs two's complement negation of a number that is of arbitrary length.

Background

This homework assignment centers around the design of a sequential Boolean network whose function is to produce the two's complement negation of the number that is presented on its input. This number is to be entered serially, one bit at a time, and the results are to be produced in a similar serial manner one bit at a time as shown in Figure 1 below. It is assumed that this number is always preceded by an active high reset signal which is applied on clock cycle before the next number is sent.The **Reset** signal should be connected to the synchronous preset or clear inputs of the flip-flops that maintain the state information and should <u>not</u> be considered as an input when deriving state machine representations. Also it is assumed that the control logic for the **Reset** signal has already been developed and is not part of this design problem.:

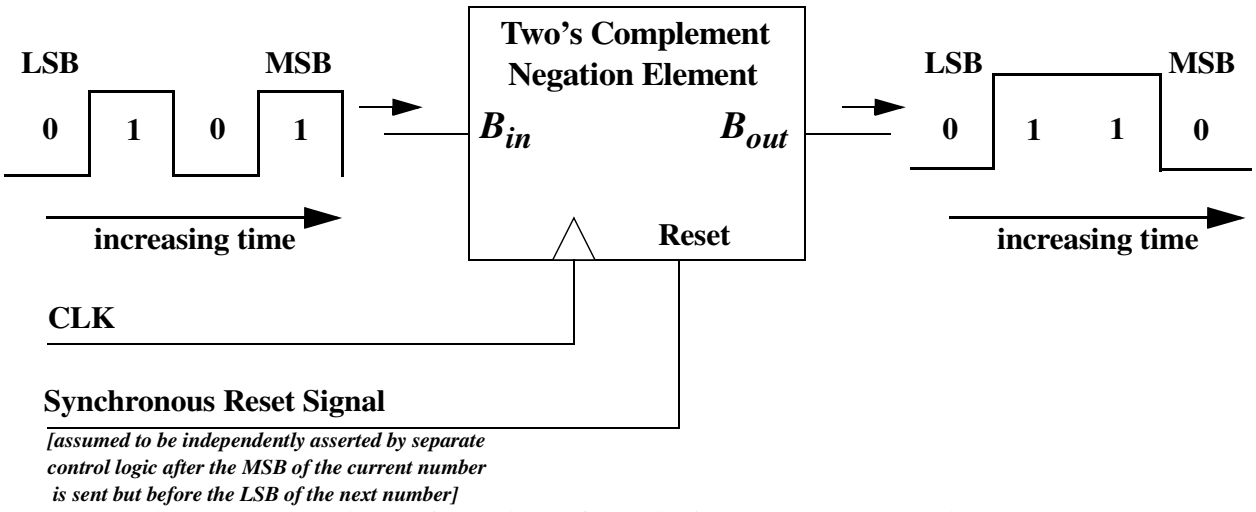

Figure 1: Functional Overview of Two's Complement Negation Element

Assignment Part 1

Develop a Mealy State Graph (SG) for the Two's Complement Design shown in Figure 1. Assume that the active high Reset pulse will place the FSM in its initial state. The design should contain a minimum number of states to implement the desired function. Identify the initial state and fully label all states in the diagram. Also label the input(s) and output(s) of the design. Using this state graph develop an input sequence which will transverse all arcs and nodes of the SG.

Assignment Part 2

Implement the SG design in Part 1 using the minimum set of discrete logic gates (AND, OR, and NOT) and D-flip-flops for the state assignment that you have chosen. Show all step in this process. Evaluate the correctness of your design by neatly comparing the output of your final implementation with that which is predicted by the SG for the sequence that you developed in Part 1.

Assignment Part 3

Replace the combinational logic potion (AND, OR and NOT gates) of Part 2 with a ROM that is of minimal size to implement the SG of Part 1 with a minimal number of D-flip-flops. Clearly identify the Address and Data buses of the ROM element and specify its complete ROM Table.

Assignment Part 4

Replace the combinational logic potion (AND, OR and NOT gates) of Part 2 with one or more 4-to-1 multiplexers that will implement the SG of Part 1 with a minimal number of D-flip-flops. Clearly identify the inputs and outputs of the multiplexers. Assume positive logic with a Logic 1 representing VCC and a Logic 0 representing GND. You may also use inverters (NOT gates) as needed.

Assignment Part 5

Develop a Moore State Graph, SG, for the Two's Complement Design shown in Figure 1. Assume that the active high reset will place the FSM in its initial state. The design should contain a minimum number of states to implement the desired function. Identify the initial state and fully label all states in the diagram. Also label the input(s) and output(s) of the design. Using this state graph develop an input sequence which will transverse all arcs and nodes of the SG.

Assignment Part 6

Implement the SG design in Part 5 using the minimum set of discrete logic gates (AND, OR, and NOT) and D-flip-flops for the state assignment that you have chosen. Show all step in this process. Evaluate the correctness of your design by neatly comparing the output of your final implementation with that which is predicted by the SG for the sequence that you developed in Part 5.

Assignment Part 7

Implement the SG design in Part 5 using again the minimum set of discrete logic gates (AND, OR, and NOT) but this time implement your state assignments using one-hot encoding. In this encoding scheme each state has a corresponding D-flip-flop which is at a Logic 1 whenever the design is in that state and is a Logic 0 otherwise. Assume in this case that the D-flip flops have both presets and clears which will be used to make sure that the state is initialized with only the flip-flop associated with the initial state being a Logic 1 and the remaining flip-fops being a Logic 0. Show all step in this process. Evaluate the correctness of your design by neatly comparing the output of your final implementation with that which is predicted by the SG for the sequence that you developed in Part 5.

Homework Assignment Turn in Procedure

The Due date for this homework is Friday 11:59 PM February 18, 2015. This is to be an electronic submission -- no hardcopy needs to be turned in to the instructor. You are to upload to the course UAH Canvas dropbox a copy of the homework assignment on or before its due date. Acceptable file formats include pdf, doc, and docx. Homework will be graded based on correctness, neatness, and completeness.