

## PRELAB!

Read the Mini-Project lab document and complete as much of the answer sheet as you can before lab.

### 1.0 Objective

The objective of this lab is to gain experience with the digital circuit design process. You will be given a non-functional circuit and you will be asked to redesign it so that it meets its intended specification, which is to detect 4-bit prime numbers.

### 2.0 Setup

Begin by creating a new folder `/home/username/CPRE281/mProj`. You will be saving your work for this lab in this directory. Information about the DE2 board, such as hardware device pin assignments, can be found in the user's manual provided with the Lab05 files. An excerpt from the user's manual containing only the pin assignment information is also available with the Lab 03 files.

### 3.0 Overview

In this lab, you will be working with two functions,  $B(W, X, Y, Z)$  and  $P(W, X, Y, Z)$ .

The first function,  $B$ , is given to you as a NAND circuit diagram. It is *supposed* to implement a 4-bit prime number detector, but for some input cases it computes the wrong value. In other words, it outputs that a number is prime when it is not and vice versa.

The second function,  $P$ , is the correct 4-bit prime number detector, for which you will have to write down its complete truth table. Then you will have to implement  $P$  in two different ways: first from scratch and then by adding additional logic to correct the  $B$  function for those cases in which it is incorrect.

### 4.0 Uncle Bob's Circuit: NANDs to ANDs/ORs/NOTs

Uncle Bob calls you one evening and exclaims that he has created a fantastical digital logic circuit. He has mailed you a package which you received earlier in the day. You open the package to find a logic circuit as well as a schematic detailing its design. Uncle Bob asks you to guess the circuit's purpose. The circuit is seen in **Figure 1**.

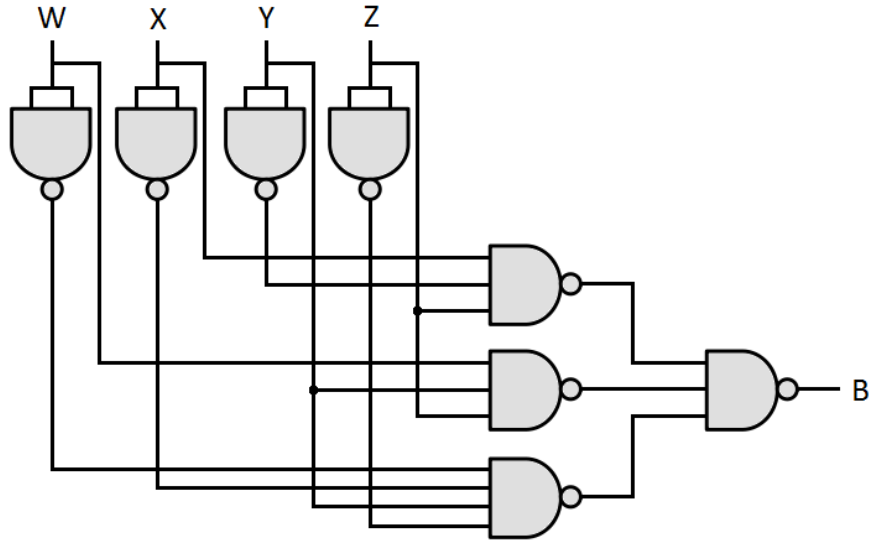


Figure 1 – Uncle Bob's Circuit

At first you are not sure what to make of the circuit. The schematic is implemented only with NAND gates, making it difficult to analyze. Convert Uncle Bob's circuit diagram, which uses only NANDs, to an equivalent circuit which uses only ANDs, ORs, and NOTs. Draw your circuit on the answer sheet.

### 5.0 Implementing From Canonical SOP

If you did section 4.0 correctly, your schematic should be a canonical SOP circuit. On the answer sheet write down the canonical SOP expression for this circuit using shorthand notation (i.e. sigma).

Next, implement Uncle Bob's circuit in Verilog as an SOP expression (i.e. a behavioral circuit description). Program the FPGA board and demonstrate the results to a TA, assigning pins as necessary. Write or attach the completed Verilog function on the answer sheet.

### 6.0 Prime time: Debugging Function B

After observing the seemingly nonsensical output of the circuit, you give up and ask Uncle Bob what it was supposed to do. He is a bit disappointed and tells you that the circuit determines if a 4-bit number is prime. That is, the circuit outputs 0 if the 4-bit number formed by the inputs is not prime and 1 if it is prime.

You check the output behavior of Uncle Bob's circuit, and it seems that it gives the wrong value in some cases. You are determined to fix it. Write out the truth table for Uncle Bob's function (**B**) side-by-side with the truth table of the correct 4-bit prime detector function (**P**) that you will soon design. (Note that both 0 and 1 are not prime.) With the functions side-by-side, you should see several discrepancies between these two functions.

Lastly, fill out the K-map of **P**, and use it to find a simplified SOP expression for the function.

### 7.0 Reimplement Function P

Implement the correct 4-bit prime number detector (**P**) from scratch using the truth table and simplified SOP expression. *You may use either Verilog or schematic design.* Include this implementation with your answer sheet, then demonstrate the results to the TAs.

### 8.0 Add Corrective Logic To Fix the Circuit for B

The circuit you simulated on the board is nice, but you can't give it to Uncle Bob; he doesn't have an FPGA of his own. It would also hurt Uncle Bob's feelings if you didn't somehow incorporate his design into your final circuit. So, you decide to create an additional design that uses Uncle Bob's circuit.

*Design and implement a circuit that uses Uncle Bob's circuit but fixes his mistakes. Use a 4-to-1 multiplexer in your design to select between the output of Uncle Bob's circuit and the correct values. Draw the circuit on the answer sheet and demonstrate the results.*

Create a symbol file for Uncle Bob's function (**B**) that you implemented in Verilog previously. (See Section 4.0 of Lab 5 for instructions on how to create and use symbol files.)

### 9.0 Complete

You present Uncle Bob with the corrected prime detector circuit. He is impressed by your efforts and says he'll put in a good word for you at his company.

You are done with this lab. Ensure that all lab files are closed, exit Quartus II, log off the computer, and hand in your answer sheet. **Don't forget to write down your name, student ID, and your lab section number.**