Digital Design Lab 3

Tara Moses

Section L001

October 5, 2015

tsmoses@uark.edu

**Abstract**

The purpose of this lab was to construct multiple circuits and document their operation using IC chips. A function was constructed using three different circuits: a circuit using standard sum-of-products form, one using only NAND gates, and one using only NOR gates. The circuits were drawn in Quartus II before being wired on the trainer board.

**Introduction**

Quartus II is software that allows the user to create logic circuits and connect them to hardware. It works by letting the user drag and drop basic logic gates such as NOR, AND, OR, etc. The software offers various forms of these logic gates, each having a different number of inputs. After the logic circuit has been made, the user can rename each input (e.g. “switch 1”, “button 2”) and output (e.g. “LED”) to make the circuit easier to understand. In this lab, Quartus II was only used to make the diagram of each circuit; it was not used to run the circuits.

This lab was designed to familiarize the student with taking a function and wiring a circuit to represent it using various IC chips. In this lab, a truth table was created for the given function. Then a number of circuits were drawn up to represent the function. After each function was drawn in Quartus, it was wired and tested on the trainer board. The board was expected to light up green when the truth table indicated a 0 and red when the truth table indicated a 1.

**Design and Implementation**

In this lab, a truth table was created for the function

. In order to create the truth table, inputs were listed and for the 0th, 3rd, and 7th combinations of inputs, the output of F was a 1. For all other input combinations, F returned a 0. Because of this, the circuit to build was represented in sum-of-products form as

. This equation was simplified using Boolean algebra as

. The circuit was drawn in Quartus as shown in Figure 1 below.

Figure 1: circuit diagram of F using inverters, OR gates, and AND gates

After the figure was drawn, it was wired on the trainer board using a 7411 IC chip for the 3-input AND gates, 7408 for 2-input AND gates, 7432 for 2-input OR gates, and 7404 for the inverters. It was tested successfully; the trainer board LED was green for inputs 000, 111, and 011.

For the second part of the lab, the circuit was represented using only NAND gates. In order to do this, each inverter, OR, and AND gate was replaced with a NAND gate. The inverters were replaced by 2-input NAND gates, with the input wired as both inputs into the gate. The circuit was drawn in Quartus, and the schematic was shown in Figure 2 below.

Figure 2: circuit diagram of F using NAND gates only

After the schematic was created, the IC chips for the previous wiring were all replaced with 2- and 3-input NAND gates, represented by 7400 and 7410 IC chips, respectively. The circuit was tested and was successful.

For the third part of the lab, two circuits for F’ were created: one using standard product-of-sums form, and one using only NOR gates. The truth table for F was used, except for F’ the inputs of F were flipped: 0s were turned into 1s and vice versa. The Boolean equation representing F’ was

. This circuit was drawn in Quartus, and the schematic is shown in Figure 3 below.

Figure 3: circuit diagram of F’ using inverters, OR gates and AND gates

After the circuit was tested, it was changed to only feature NOR gates. To do this, each gate was replaced with NOR gates. Inverters were replaced with 2-input NOR gates, and each input was fed into the NOR gate as both of its inputs. A schematic of this is shown in Figure 4 below.

Figure 4: circuit diagram of F’ using only NOR gates

This circuit was successful as well.

**Results**

The LED only lit up green when the inputs were 000, 011, and 111 for the F circuits. For the F’ circuits, the LED lit up red on these inputs, indicating that the logic circuits were correctly built. The lab took about an hour and a half to complete.

**Conclusion**

The lab was designed to give students a hands-on experience with building logic circuits to represent functions. Using Quantus and a trainer board, logic circuits were created to model a function and its inverse. In the future, these skills could be used to model more intricate functions with more inputs.