CMPE 160 Laboratory Exercise 04

Combinational Logic Circuit Design Using Boolean Algrebra Simplification

Christopher Larson
Performed: 7 February 2018
Submitted: 14 February 2018

Lab Section: L3
Instructor: Mr. Dominguez
TA: TA Andrew Ramsey
TA Matthew Miller
TA Madeline Mooney

Lecture Section: 01
Professor: Professor Beato

By submitting this report, you attest that you neither have given nor have received any assistance (including writing, collecting data, plotting figures, tables or graphs, or using previous student reports as a reference), and you further acknowledge that giving or receiving such assistance will result in a failing grade for this course.

Your Signature: ______________________________

Abstract

The objective of this exercise is to simplify Boolean algebraic expressions to a simplified expression and implement the expressions using a combinational logic circuit. Combination logic circuits provide control signals to device or perform mathematical operations. The logic gate designed in the exercise read a a single two-bit binary number, N= (N1,N0), and produced a single four-bit output, F=(W,X,Y,Z), which depends on the selection control signal, "C". If C was "0" then the output value would be the square of N, (F = N^2), and if C was "1" then the output value would be fives times N, (F= 5N). The control signal, two-bit input and the four-bit output were recorded in a table and used to create the Boolean expressions used for the circuit. The physical cicuit performed the correct function that was theorized on the truth table.

Design Methodology

The exercise used a two-bit binary number, N=(N1,N0) and a control signal to determine the output of the four-bit binary number shown in Table 1. The control signal determined whether the output was found using F= N^2 when C was "0" and F = 5N when C was "1".

Table 1: Truth table for mathematical operations N^2 and 5N

C	N1	N0	W	X	Y	Z
0	0	0	0	0	0	0
0	0	1	0	0	0	1
0	1	0	0	1	0	0
0	1	0	1	0	0	1
1	0	0	0	0	0	0
1	0	1	0	1	0	1
1	1	0	1	0	1	0
1	1	1	1	1	1	1

The outputs were then used to create sum of products Boolean expression, where each product term included all of the input variables. Using Boolean algebra properties the Boolean expressions were simplified so they could be implemented with the minimal parts from the lab kit. The simplification are shown below with the Boolean algebraic properties stated next to the step that they apply to.

$$W(C,N1,N0) = \overline{C}N1N0 + CN1\overline{N0} + CN1N0 \tag{1}$$

$$(Distributivity)\overline{C}N1 + N0 + CN1(\overline{N}) + N0) \tag{2}$$

$$(Complements)\overline{C}N1 + N0 + CN1 \tag{3}$$

$$(RedundantLiteralRule)CN1 + N1N0 \tag{4}$$

$$X(C,N1,N0) = \overline{C}N1\overline{N0} + C\overline{N1}N0 + CN1N0 \tag{5}$$

$$(Distributivity)\overline{C}N1N0 + CN0(N1 + \overline{N1}) \tag{6}$$

$$(Complements)\overline{C}N1N0 + CN0 \tag{7}$$

$$Y(C, N1, N0) = CN1\overline{N0} + CN1N0 \tag{8}$$

$$(Distributivity)CN1(\overline{N0} + N0) \tag{9}$$

$$(Complements)CN1 \tag{10}$$

$$Z(C, N1, N0) = \overline{C}\,\overline{N1}N0 + \overline{C}N1N0 + C\overline{N1}N0 + CN1N0 \tag{11}$$

$$(Distributivity)N1N0(\overline{C} + C) + \overline{C}\,\overline{N1}N0 + C\overline{N1}N0 \tag{12}$$

$$(Complements)N1N0 + \overline{C}\,\overline{N1}N0 + C\overline{N1}N0 \tag{13}$$

$$(Distributivity)N1N0 + \overline{N1}N0(\overline{C} + C) \tag{14}$$

$$(Complements)N1N0 + \overline{N1}N0 \tag{15}$$

$$(Distributivity)N0(\overline{N1} + N1) \tag{16}$$

$$(Complements)N0 \tag{17}$$

Using these simplified expressions, a circuit was designed using basic gates. These are limited to the gates that are in the lab kits, the gates used were Inverters, dual-input AND, triple-input NAND, and dual-input OR gates, as shown below in Figure 1.

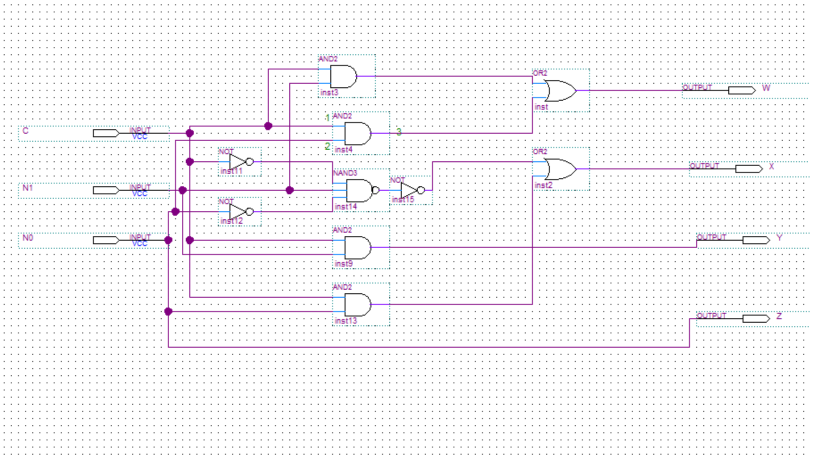

Figure 1: The schematic for the circuit built in Exercise Four

Using the circuit represented in Figure 1, the input and outputs were tested using four LEDs to test each of the outputs(W, X, Y, Z) and three switches for the inputs(C, N1, N0). A lit LED indicated that the gate was open and an unlit LED indicated that the gate was closed.

Results and Analysis

In order to verify the correctness of Table 1, a simulation was created to generate a waveform that would verify the correctness of the schematic.

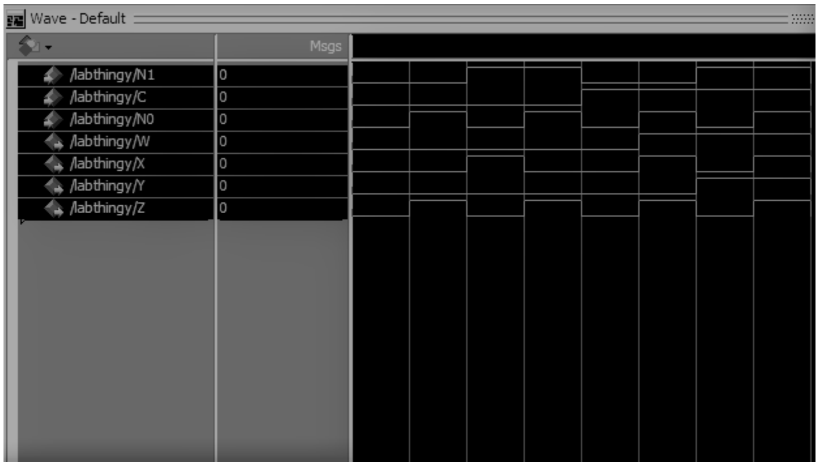

Figure 2: The simulated waveform of the circuit built in Exercise Four

The waveform in Figure 2 matched the truth table in Figure 1 exactly and shows that the circuit design was correct. The test results from the breadboarded circuit are shown in Table 2.

Table 2: Table of results for the LED test

IN1	IN2	IN3	LED1	LED2	LED3	LED4
0	0	0	OFF	OFF	OFF	OFF
0	0	1	OFF	OFF	OFF	ON
0	1	0	OFF	ON	OFF	OFF
0	1	1	ON	OFF	OFF	ON
1	0	0	OFF	OFF	OFF	OFF
1	0	1	OFF	ON	OFF	ON
1	1	0	ON	OFF	ON	OFF
1	1	0	ON	OFF	ON	OFF

The results shown in Table 2 match the truth table shown in Table 1, where IN1 is input 1 and so on, and L1 is LED1 and so on. This shows that the circuit was constructed correctly.

Conclusion

The use of Boolean algebraic properties is helpful in translating a truth table or a larger expression to a simplified and minimal expression that would be useful in a circuit. It is important to design and simulate the simplified expressions to ensure that the Boolean algebra was done correctly and still relevant. Once the expressions are simulated it is much simpler to create a functional circuit, and with the help of LEDs and switch can prove the expected results to be true. The exercise proved successful as the recorded results match the expected values.

Questions

1. The equivalent gate count or EGC of the circuit is equal to the total number of dual-input gates required to implement the expression and in this case the answer is eight gates.
2.The chip count for basic gates of the circuit is the total number of chips required to implement the expression and in this case the answer is four chips.