**CMPE 160 Laboratory Exercise Five**

**Combination Logic Circuit Design**

**Using Karnaugh Map Simplification**

Ari Sanders

Performed: February 26th, 2015

Submitted: March 5th, 2015

Lab Section: L4

Instructor: Biru Cui

TA: Terry

Justin

Lecture Section: 03

Professor: Richard Cliver

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 purpose of this exercise was to use Karnaugh mapping to reach minimal SOP and POS functions in order to build the simplest circuit. First, a K-map was composed using the min-terms given. Then, one SOP and one POS expression were each derived from the K-map. Finally, the circuit with the least equivalent gate count was implemented on the breadboard. In this case, it was the circuit represented by the SOP form.

**Design Methodology:**

The exercise began with the generation of a K-map from the function F, which was defined as the function with min-terms on lines 0, 2, 4, 6, 7, 9, 13, 14, and 15.

Table 1: Karnaugh Map for Function

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| CD\AB | 00 | 01 | 11 | 10 |
| 00 | 1 | 1 | 0 | 0 |
| 01 | 0 | 0 | 1 | 1 |
| 11 | 0 | 1 | 1 | 0 |
| 10 | 1 | 1 | 1 | 0 |

From the groupings formed by the ones came a minimal sum-of-products expression as follows:

CB + A’D’ + AC’D

From the groupings formed by the zeros came a minimal product-of-sums expression as follows:

(A’+ C + D)(A’ + B + C’)(A + C + D’)(A + B + D’)

The POS expression would require 2 OR chips, 1 AND chip, and 1 INVERTER chip for a total of 4 chips. The SOP expression, however, would require only 1 AND chip, 1 OR chip, and 1 INVERTER, for a total of 3 chips. This wins out on EGC, so it was the circuit that was built.

This is where it stops.

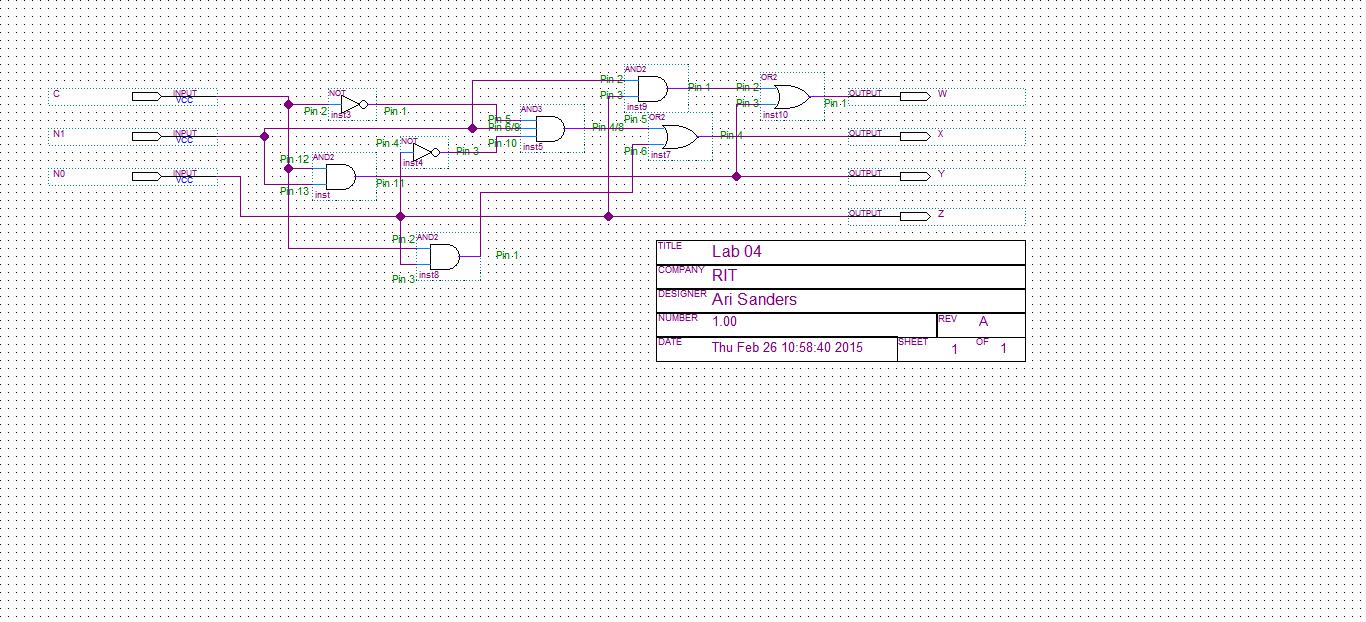


Figure 1: Circuit Diagram

Another note about Figure 1: the virtual design features a three-input AND gate, cutting the total number of gates by one. In the physical implementation, this was replaced with two two-input AND gates, which explains the odd pin numbering on that gate in the figure.

**Results and Analysis:**

The result of this exercise was a minimal circuit utilizing only what it needed in terms of gates and chips. Post-lab analysis shows (as mentioned in several other places throughout this report) that the implemented circuit was actually not the most minimal. The most minimal would have used DeMorgan’s law to convert one of the terms from an N1N0 to (N1 + N0)’. While this uses one more gate, it fills in empty spots on chips instead of adding another one, resulting in one fewer IC than the implemented version had. However, despite this small breach, overall the circuit was minimal, and in fact used the least number of gates. The resultant circuit can be seen in Figure 1.

**Questions:**

There are various metrics to evaluate the efficiency of an implementation of a Boolean expression. For each of the metrics below, give it’s value a) for direct implementation of the expressions from the Prelab Part 1 with AND, OR, and inverter gates and b) for your simplified expressions as implemented in lab.

1. Equivalent gate count (ECG) --- the total number of 2-input gates required to implement the expression.
   1. (C’N1N0 + CN1N0’ + CN1N0)[six AND, two OR, two INV] plus

(C’N1N0’ + CN1’N0 + CN1N0)[six AND, two OR, three INV] plus

(CN1N0’ + CN1N0)[four AND, one OR, one INV] plus

(C’N1’N0 + C’N1N0 + CN1’N0 + CN1N0)[eight AND, three OR, four IN] is

6 + 6 + 4 + 8 = 24 AND,

2 + 2 + 1 + 3 = 8 OR, and

2 + 3 + 1 + 4 = 10 INV,

for a grand total of 24 + 8 + 10 = 42 gates.

However: there are only eight possible min-terms, of which six were used here. If terms are re-used, this cuts the number of AND gates down to twelve.

New total: 12 + 8 + 10 = 30 gates.

Furthermore, there are only six possible inputs to an AND gate in a three-input circuit. This means that by re-using ANDs across terms, the number can be reduced further, but it will not be calculated further here.

Final answer: 30

* 1. (N1N0 + CN1)[two AND, one OR, zero INV] plus

(C’N1N0’ + CN0)[three AND, one OR, two INV] plus

(CN1)[one AND, zero OR, zero INV] plus

(N0)[zero AND, zero OR, zero INV] is

2 + 3 + 1 + 0 = 6 AND,

1 + 1 + 0 + 0 = 2 OR,

0 + 2 + 0 + 0 = 2 INV,

for a grand total of 6 + 2 + 2 = 10 gates.

With re-use of gates, only CN1 can be re-used, bringing the total down to 5 ANDs.

New total: 5 + 2 + 2 = 9 gates.

Final answer: 9

1. Chip count from basic gates (i.e. ANDs, ORs, and inverters) --- the total number of ICs (chips) required to implement the expression
   1. Based on the previous answer of 12 AND, 8 OR, and 10 INV, assuming the chips hold four gates each, that would require 3 AND, 2 OR, and 3 INV, for a total of 8 ICs.

Final answer: 8

* 1. Based on the previous answer of 5 AND, 2 OR, and 2 INV, assuming the chips hold four gates each, that would require 2 AND, 1 OR, and 1 INV, for a total of 4 ICs.

The best implementation would trade the fifth AND for a third OR and third INV, changing the totals to one of each chip. Unfortunately, this design was not used.

Final answer: 4

**Conclusion:**

The exercise began with forming expressions from a truth table and simplifying them. This was simple. It then moved on to forming this into a circuit. This was also simple. The difficult part was creating the physical circuit. This involved knowing the proper pinout, as well as how to troubleshoot and find where the voltage is not what it should be to determine where to fix it. This shouldn’t be much of a problem in the future, since it was one here.

As can be clearly seen by the various information within this report, simplifying results in circuits that are both easier to understand, implement, and troubleshoot, as well as use less chips and gates for a neater, cheaper, and much more beautiful circuit.

Simplifying is good, troubleshooting is important to do properly. Exercise was a success.