

Laboratory Exercise #3

Logic Minimization with Karnaugh Maps

ECEN 248: Introduction to Digital Design
Department of Electrical and Computer Engineering
Texas A&M University



1 Introduction

The purpose of this week's lab assignment is to introduce a real-world application of digital electronics while demonstrating the use of Karnaugh-Maps for logic minimization. In the laboratory, you will bread-board a simple circuit which calculates the profits of a small farm. You will also feed the output of your circuit into a seven-segment display for ease of viewing.

2 Background

2.1 Logic Minimization

Logic minimization refers to the reduction of terms within a Boolean expression. Among the various methods used to minimize Boolean expressions, the three most prominent ones are Quine-McCluskey (Q-M) algorithm, Karnaugh Maps (K-Maps), and Espresso heuristic logic minimization.

1. The Quine-McClusky algorithm is an 'exact' approach to logic minimization, which means that it returns the solution with the fewest product terms, but requires a significant amount of computational power.
2. Karnaugh Maps provide a tabular method for performing the Q-M algorithm; however, this method is only practical for logic expressions with 5 or less boolean variables (i.e. inputs).
3. Espresso heuristic methods are fast compared to 'exact' solutions and achieve results that are almost as good.

In this lab, we will use Karnaugh Maps to reduce our Boolean expressions.

2.2 Karnaugh Maps

The Karnaugh Map is a visual tool used to simplify Boolean algebra expressions. Starting with a grid representation of output values from a truth table, logic minimization is performed by grouping minterms (or maxterms in the case of a product of sum expression) into powers-of-two. The minimized expression can then be generated from these groups using the axiom laws of Boolean algebra. Please consult the course text book if this is unclear!

2.3 Binary Coded Decimal

Binary Coded Decimal (BCD) is an encoding method in which each decimal digit is represented by a 4-bit binary sequence. BCD can simplify the process of displaying decimal digits on a seven segment display as you will see in a moment.

2.4 Seven Segment Display

The seven-segment display contains seven individual Light Emitting Diode (LED) segments, arranged in the shape of the decimal digit **8** as shown in Figure 1. In such a display, each segment can be individually enabled, allowing the display of decimal digits 0 through 9 depending on which segments are lit up at any point in time. Figure 2 demonstrates the display of decimal digits 0 through 9 on a seven-segment display.

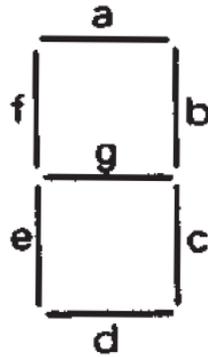


Figure 1: Seven-Segment Display

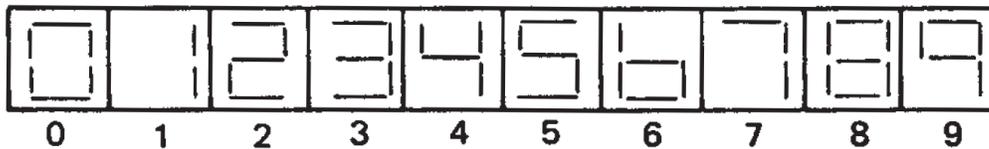


Figure 2: Digits 0-9 on the Seven Segment Display

2.5 BCD Seven-Segment Display Decoder

A BCD seven-segment display decoder (Figure 3) takes as input a BCD digit and decodes it into seven outputs which individually control the seven segments of a seven-segment display. In Figure 3, **A** is the least significant bit of the BCD input, while **D** is the most significant bit of the BCD input. The output bits of the decoder (**a**, **b**, **c**, **d**, **e**, **f**, **g**) correspond to the individual segments of the seven-segment display shown in Figure 1.

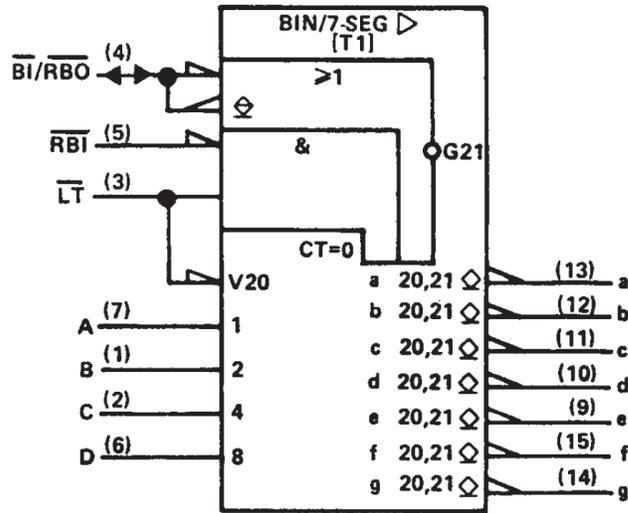


Figure 3: Seven Segment Display Decoder/Driver

Note: For more information on the seven-segment display, consult the datasheet for the following parts: SN74LS47 (the decoder/driver) and LDS-A514RI (the display).

3 Pre-lab

3.1 Design

In this design, you will implement a simple digital circuit to compute total profits made by an animal farm. Although the owner, John, has a lot of experience running the farm, he does not have the time to compute the profits and would like to have a device which can compute the profits for him based on the kinds of animals that he is raising. John has to follow the set of guidelines below for setting up the farm.

1. He can raise cows, sheep, hogs, and chickens on his farm but cannot raise all at the same time.
2. The farm does not have enough space to raise more than 2 different types of animals. Therefore, his farm never has more than 2 types of animals at any particular time.
3. Each type of the animal earns a certain level of profit for the farm. Cows earn him a profit of **4** units, sheep earn him a profit of **2** units, hogs earn a profit of **3** units and chickens being small livestock earn him a profit of **1** unit.
4. The total profit of the farm is the sum of profits made by each type of animal that is raised with the exceptions provided below.

5. Hogs and chickens have very different food, so John needs to spend extra for the transportation of food for each separately. Therefore, if he plans to raise hogs and chickens together then his profits go down by **1** unit.
6. Similarly, if he raises cows and chickens his profits go down by **1** unit because of extra transportation costs. Additionally, cows have a tendency to kill some chickens on the farm. This further reduces his profits by **1** unit.
7. However, cows and sheep do very well if raised together. They have many similarities especially with respect to the kind of food they eat. Also due to the huge demand for sheep and cows, the profit they make goes up by **1** unit if cows and sheep are reared together.

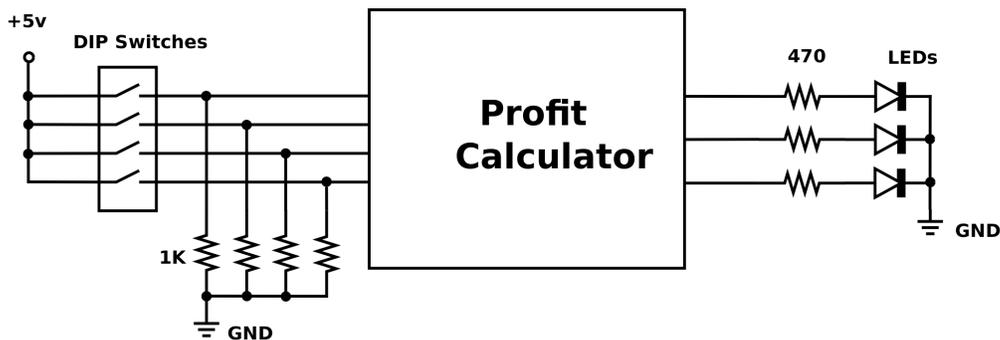


Figure 4: Profit Calculator Block Diagram

Your objective is to design a digital circuit which takes in the types of animals on the farm and outputs a binary number representing the corresponding profit. The four input bits are I (Chicken), S (Sheep), C (Cow), and H (Hog), and the output bits are, P2, P1, P0, such that P2 is the most significant bit. Begin by creating a truth table containing all of the input combinations. Because there are four input bits, your table should contain $2^4 = 16$ rows. Then fill in the output columns of your truth table based on the guidelines provided above. For outputs that are not possible due to restrictions on the farm, insert 'X' to represent "don't cares." Once your truth table is complete, use Karnaugh Maps as discussed in class to simplify the boolean algebra expressions for each of the output bits. To conclude the design, draw a schematic of the final digital circuit. Be sure to label all inputs and outputs. In lab, we will use Light Emitting Diodes (LEDs) to display bits P2 through P0 and dip switches to provide input to our circuit. Please see Figure 4. Be sure to include these components in your schematic.

3.2 Pre-lab Deliverables

Please include the following items in your pre-lab write-up.

1. Truth table for profit calculator.
2. Karnaugh Maps for each of the output bits, P2, P1, P0.
3. Minimized boolean algebra expression in terms of input bits (I, S, C, and H) for each output bit.
4. Gate level schematic of your profit calculator which includes DIP switches and LEDs.

4 Lab Procedure

For the lab procedure below, consult the datasheets of each component for the pin-outs and electrical/timing characteristics of these circuits. Remember that the LED cathode is marked by a flat spot on the round plastic package and should be connected to ground.

4.1 Experiment 1

Bread-board and test your Profit Calculator using the gates in the Integrated Circuits (ICs) provided. Use the DIP switches as shown in Figure 4 for input to your digital circuit and the LEDs for output. Be sure to follow the procedures outlined in the *Policies and Procedures* document to ensure a neat and tidy circuit. Do not forget to provide +5V and a common ground to all of the IC packages. For each allowable input combination, the corresponding profit in binary should be displayed on the LEDs.

Note: Demonstrate your progress to the TA when complete.

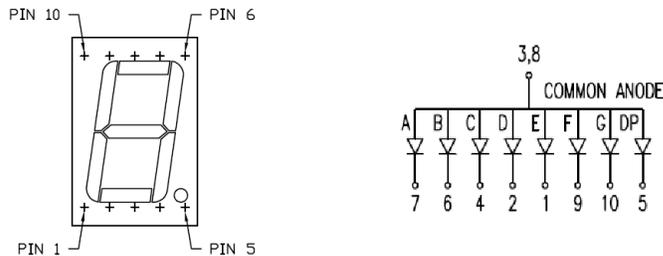


Figure 5: Seven-Segment Display Pinout

4.2 Experiment 2

For experiment 2, we would like to display the output of our profit calculator on a seven-segment display (Figure 5). To do so, we must utilize a BCD to seven-segment decoder/driver, part number SN74LS47,

discussed in the Background section. Figure 6 depicts the pinout diagram for the 16-pin DIP version of the decoder used in lab. Remove the LEDs from your circuit created in Experiment 1 and feed the output bits of your profit calculator into the input of the seven-segment display decoder. Remember that your profit calculator outputs 3 bits of data while the display driver requires 4. Thus, you will need to determine how to handle this. Pins 3, 4, and 5 of the decoder/driver provide functionality that we will not use in this lab. Since they have internal pull-up resistors, they can be left floating. Locate V_{CC} and GND in Figure 6 and connect +5 volts to V_{CC} and Ground to GND .

**SN5446A, SN5447A, SN54LS47, SN5448,
SN54LS48 . . . J PACKAGE
SN7446A, SN7447A,
SN7448 . . . N PACKAGE
SN74LS47, SN74LS48 . . . D OR N PACKAGE
(TOP VIEW)**

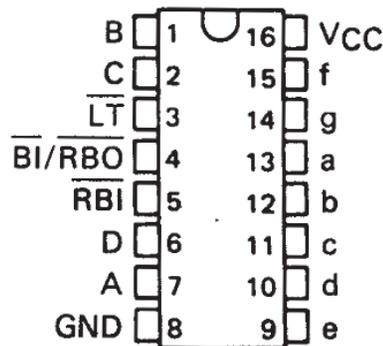


Figure 6: Seven-Segment Decoder Pinout

As seen in Figure 5, the seven-segment display used in this lab is of a common anode configuration, meaning the positive terminals of the individual LED segments are connected together. Connect the output signals of the decoder to the seven-segment display provided in your lab kit with resistors as shown in Figure 7.

Important Note! Be sure to wire up the 470 ohm resistors shown in the provided circuit. **If you do not include a resistor for every LED segment, you will destroy the 7-segment display!** Please consult the LDS-A514RI datasheet for more details. Once you have found your circuit to be operating properly, demonstrate your progress to the TA.

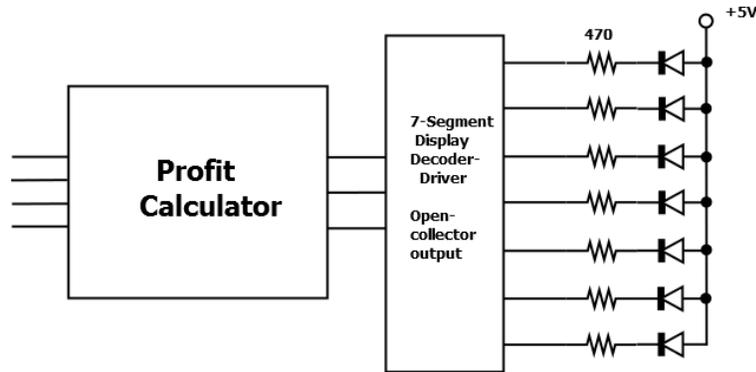


Figure 7: Open-Collector Driver to Display Circuit

5 Post-lab Deliverables

Please include the following items in your post-lab write-up in addition to the deliverables mentioned in the *Policies and Procedures* document.

1. Make a table showing the digital inputs and outputs of your circuit observed during lab. For input combinations not allowed by the aforementioned guidelines, provide the output values observed rather than 'X'.
2. Provide a comparison between the table in item 1 and the truth table you created in the pre-lab. What values does your circuit output for the "don't cares" and why?
3. Derive the Product of Sum (POS) expression for a 3-input (A, B, C) XNOR Gate. Assume that output of the gate is called 'Y'.
4. Figure 7 labels the driver output as having an open-collector output. Explain what this means. Consult the datasheet for more information.
5. (**Honors**) Use Karnaugh Maps to derive the minimized Boolean expressions for each output of the BCD to seven segment decoder. Your input should be the driver inputs (**D, C, B, A**) with **A** being the least significant bit and **D** being the most significant bit (as shown in Figure 3). Similarly, the outputs should be the driver outputs (**a, b, c, d, e, f, g, h**). Start by creating a truth table based on Figure 1 and Figure 2.
Hint: Remember that a 4-bit BCD number cannot take on a value greater than 9.
6. Include **ALL** appropriate diagrams in your write-up, even if they are provided in this lab manual.

6 Important Student Feedback

The last part of lab requests your feedback. We are continually trying to improve the laboratory exercises to enhance your learning experience, and we are unable to do so without your feedback. Please include the following post-lab deliverables in your lab write-up.

Note: If you have any other comments regarding the lab that you wish to bring to your instructor's attention, please feel free to include them as well.

1. What did you like most about the lab assignment and why? What did you like least about it and why?
2. Were there any section of the lab manual that were unclear? If so, what was unclear? Do you have any suggestions for improving the clarity?
3. What suggestions do you have to improve the overall lab assignment?