ECE 85L Digital Logic Design Laboratory

Fresno State, Lyles College of Engineering
Electrical and Computer Engineering Department
Spring 2015

Laboratory 6 – Circuit Minimization and Implementation with NAND Gates

1. OBJECTIVES

- Understanding design procedure of a Combinational Circuit
- Develop a Boolean Truth Table from a verbal problem description.
- Utilize the Karnaugh Map technique to minimize Boolean functions.
- Understand the role of "Don't Care" terms to facilitate minimization
- Implement Boolean functions using a Universal Gate type.

2. DISCUSSION

This experiment demonstrates the relationship between a Boolean function and the corresponding logic diagram. The Boolean functions are simplified by using the K-map method. Follow the design methodology discussed in the previous lab to design a **BCD to Gray Code** converter using Karnaugh maps to minimize the design and a single universal gate type (NAND) to implement the circuit.

A "code" is a function or a mapping of one set of entities (usually numbers or letters) into another set of entities (again usually numbers or letters). One example of a code is ASCII (the American Standard Code for Information Interchange) which maps English alphanumeric characters (numbers, upper and lower case letters and punctuation symbols) into 7 bit binary numbers.

Another frequently used code is BCD. BCD is an abbreviation for Binary Coded Decimal; each (single digit) decimal number (0-9) is represented by the corresponding 4 bit binary number. The binary numbers 1010-1111 are not used. The table below illustrates how one would count in BCD from 0 to 15. Note that two decimal digits (and hence 8 BCD bits) are required.

Decimal	BCD
00	0000 0000
01	0000 0001
02	0000 0010
03	0000 0011
04	0000 0100
05	0000 0101
06	0000 0110
07	0000 0111
08	0000 1000
09	0000 1001
10	0001 0000
11	0001 0001
12	0001 0010
13	0001 0011
14	0001 0100
15	0001 0101

A third relatively infrequently used code is the Gray code (named after its developer, Elisha Gray, on or about 1878). This code, originally devised for telegraph usage, has been used more recently in electron beam coding tubes and parallel encoders (for analog to digital conversion). The output data of many physical systems produce quantities that are continuous. These data must be converted into digital form before they are applied to a digital system. Continuous or analog information is converted into digital form by means of an analog-to-digital converter. The gray code is used to represent the digital data when it is converted from analog data as shown in the table below. The advantage of the Gray code over the straight binary number sequence is that only one bit in the code group changes when going from one number to the next. For more information on Gray code refer to your text book. This code (which is often called the reflected-binary code) has two interesting properties:

1. In adjacent code positions only a single binary digit changes state.

2. The code is symmetric (reflected) about a line which bisects the entire code set.

Decimal Value	Gray Code Equivalent
0	0000
1	0001
2	0011
3	0010
4	0110
5	0111
6	0101
7	0100

8	1100
9	1101
10	1111
11	1110
12	1010
13	1011
14	1001
15	1000

3. PRELAB

1. Derive the Truth Table for a 4-bit (single decimal digit) **BCD to Gray code** converter. Note that the BCD code words corresponding to the decimal numbers from 10 to 15 correspond to "Don't Care" outputs.

2. Using Karnaugh maps, derive the (minimized) Boolean equations that describe the 4 bit BCD to Gray code converter. Be sure that the final expression is in SOP form. Discuss your design equations with your lab partner. Obtain a consensus, and verify the equations with your Lab Instructor, if necessary.

3. Verify the operation of your minimized equations by implementing the design on Multisim using only NAND gates. This process will help in building and debugging the circuit. Print out your schematic captured design and include it in your laboratory notebook.

4. LAB ASSIGNMENT

4.1 Lab Assignment – Part I

1. Implement your design on the Breadboard. Try to use as few chips as possible. Number each gate as to which IC package it belongs, and number each pin to each gate in sequence. This process will help in building and debugging the circuit.

2. Build and debug your circuit. Verify its correct operation with a measured Truth Table (measure all input and output combinations). Does your measured Truth Table differ from that which you derived in Prelab? If so how? Use four logic Bit Output LED's as the outputs from the converter. Demonstrate your working circuit to the Lab Instructor.

4.2 Lab Assignment – Part II

Although you may be persuaded to the contrary, the K-map minimization technique does not necessarily minimize the number of chips (or even the number of gates). To illustrate this point you are to re-design the entire BCD to Gray Code converter using only XOR gates (note that the XOR function is not part of K-map minimization). Once you have completed your design, describe how you would implement this design using 7486 XOR chip(s), and compare the XOR chip count to your NAND gate design (you need not implement the XOR design).

4.3 Extra Credit

Implement your XOR design in the lab with 7486 XOR chips, and demonstrate a working circuit to your Lab Instructor.

5. REVIEW QUESTIONS

1. Explain how a NAND gate can be used to universally implement the 7 basic logic gates (AND, OR, NOT, NAND, NOR, XOR, XNOR).

2. Explain how K-Maps only result in reduced SOP or POS expressions.

3. Explain how a NOR gate can also be used to universally implement the 7 basic logic functions, and how a NOR gate design naturally lends itself to a POS implementation.