LAB 6

Brent Johnson

Computer Science

Texas State University

Department of Computer Science

Bj1107@txstate.edu

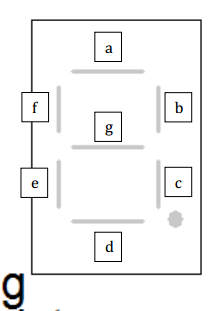
*Abstract* –  This lab was a rather monotonous one. Combinational logic is used to design and implement (digitally) a logic function that can be used as a LED light display that outputs 0 – 9 and a – f. In order to ensure that each of these 16 outputs can be fully represented, seven individual lights are used with an equation for each. In order to represent an output, a combination of each light represents an output. This lab was important practice in optimizing functions and combining them to represent combinatorial functions. It is important to know and understand how to optimize combinatorial functions because it allows for much more complexity when creating functions.

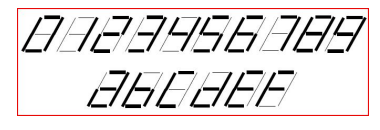
# Introduction

The theme of today’s lab is both patience and thoroughness. It is important to be patient when creating combinatorial functions because the slightest error will delay progress for the entire lab. Moreover, combinatorial logic and combinatorial functions are important because they allow for optimization. For this lab, I optimized several functions and represented them in both written logic and also digitally using DSCH. These are important skills because logic functions and electronics are increasing in complexity every day. In order to understand the logic that powers our devices, it is imperative to understand the processes of optimization and combination.

# Experimental Method

 For the first portion of this lab, I made a truth table for the seven segment component. The following picture shows both the outline of unlit seven segment component and also which segments should be active to represent each output.





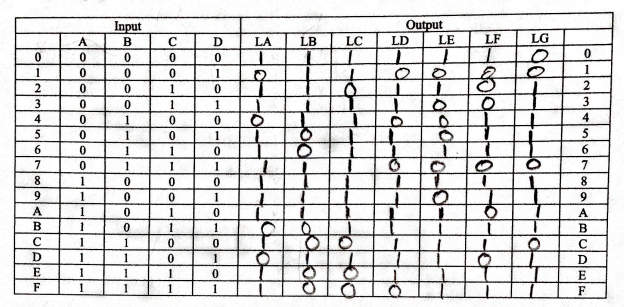
I used these desired outputs to fill in a truth table that corresponds with which segments should be illuminated in order to represent each output. In other words, for the output of 0, every segment should be illuminated other than then middle one (segment g).

Next I used Kmaps in order to optimize the expressions from the truth table. Each segment got its own Kmap and was thereby optimized. I used these Kmaps to come up with a combined logical Boolean expressions that represents the entire function.

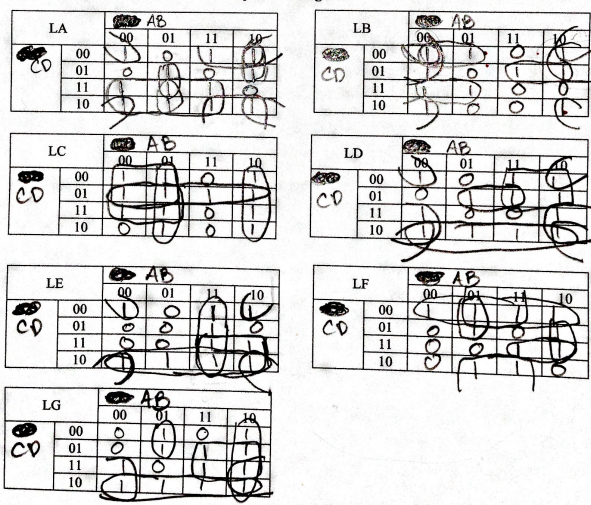
Finally, I implemented the combined function in DSCH. Using a hex editor input and seven digit display, I created one diagram that will output the letters and numbers listed above.

# Results

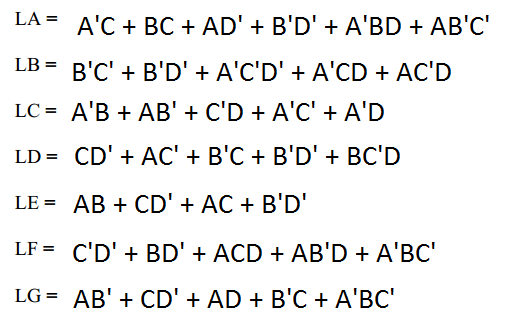
The first step was translating the desired output of the seven segment display into a truth table that represents those outputs. Here is a copy of the truth table I came up with:



Next was the chore of transcribing each individual segment function into a Kmap in order to optimize them. Here are the Kmaps for the segments:



These Kmaps led to these optimized equations:



# Conclusion

If you had any difficulty in the lab and your results seem off, this is where you say why things seemed off and mention if you could have done anything to prevent it.  Explain what you learned in lab.