ECE272 Lab 3
Spring 2017

Combinational Logic (Seven Segment Driver)

Isak Foshay

April 30th, 2018

1. Introduction

This lab aims to introduce students to K-maps and the seven segment display board. In the lab students will solder the seven segment display board, create K-maps for each of the seven segments. Being able to utilize the display board for future projects opens up a plethora of possibilities due to the ability to easily and effectively display output data to the programmer and ultimately user.

2. Design

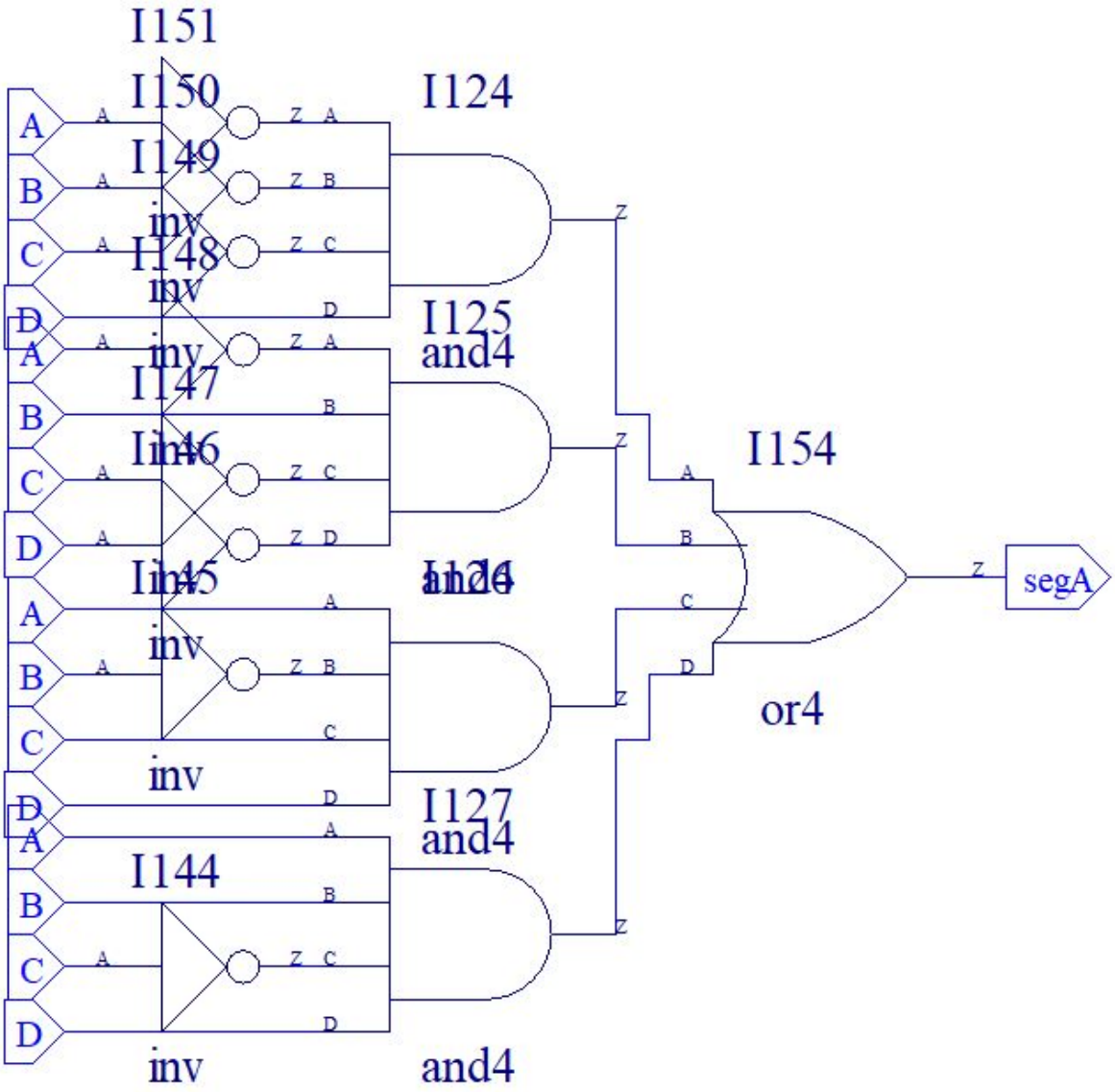

Figure 1: Schematic for segment A

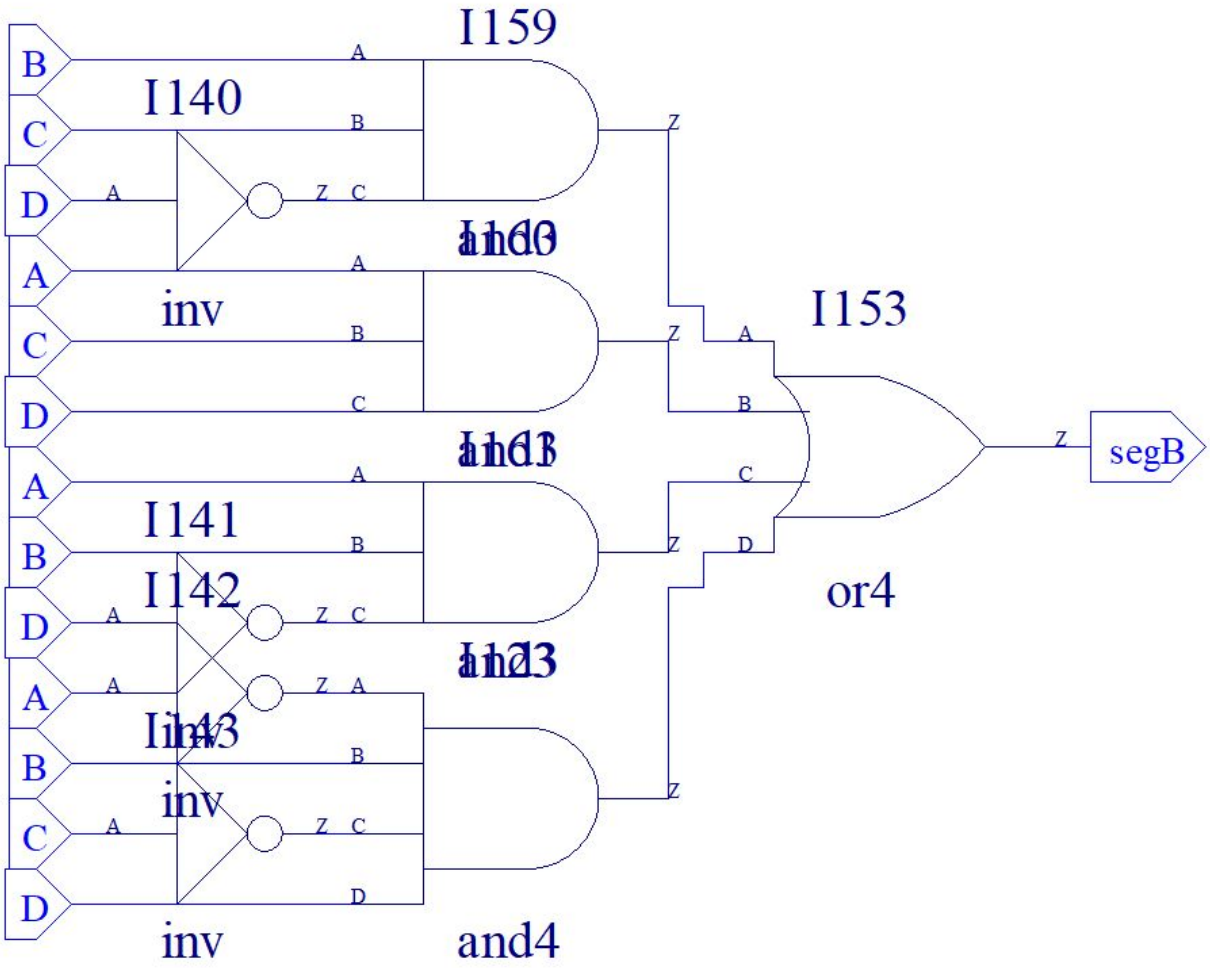

Figure 2: Schematic for segment B

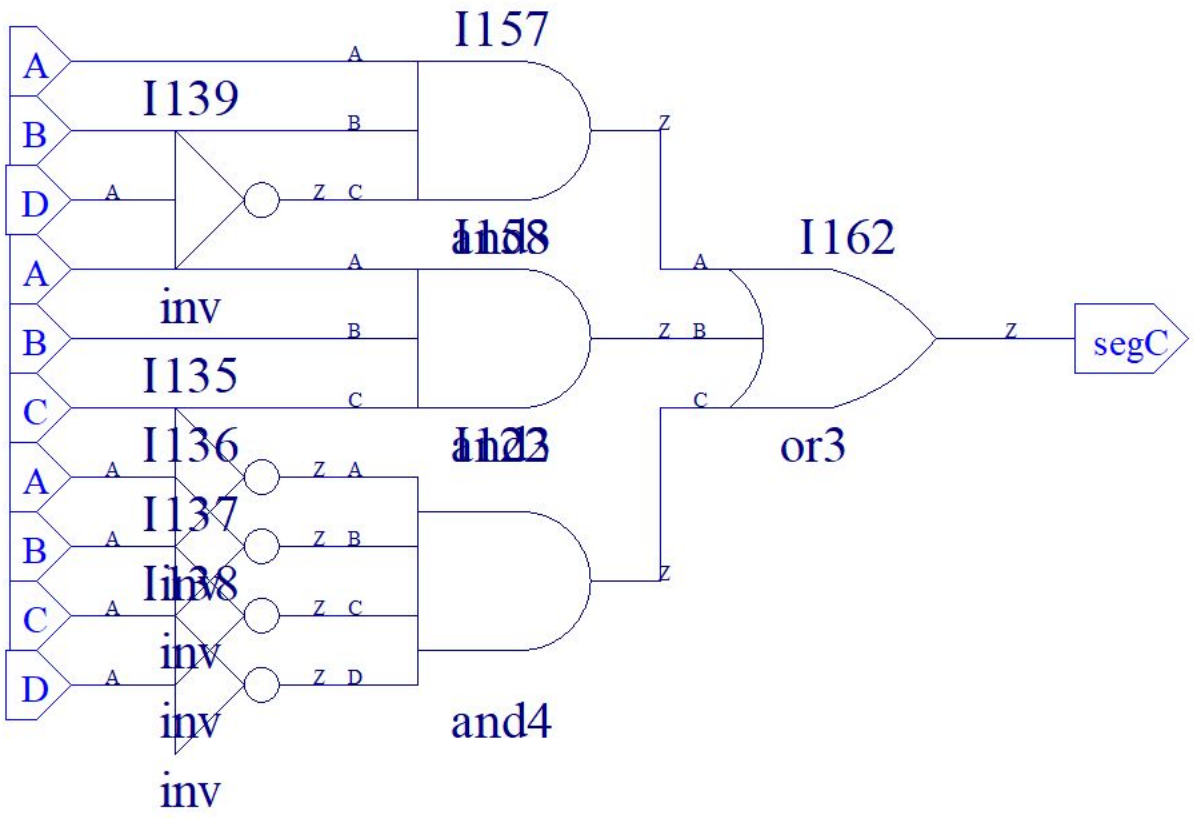

Figure 3: Schematic for segment C

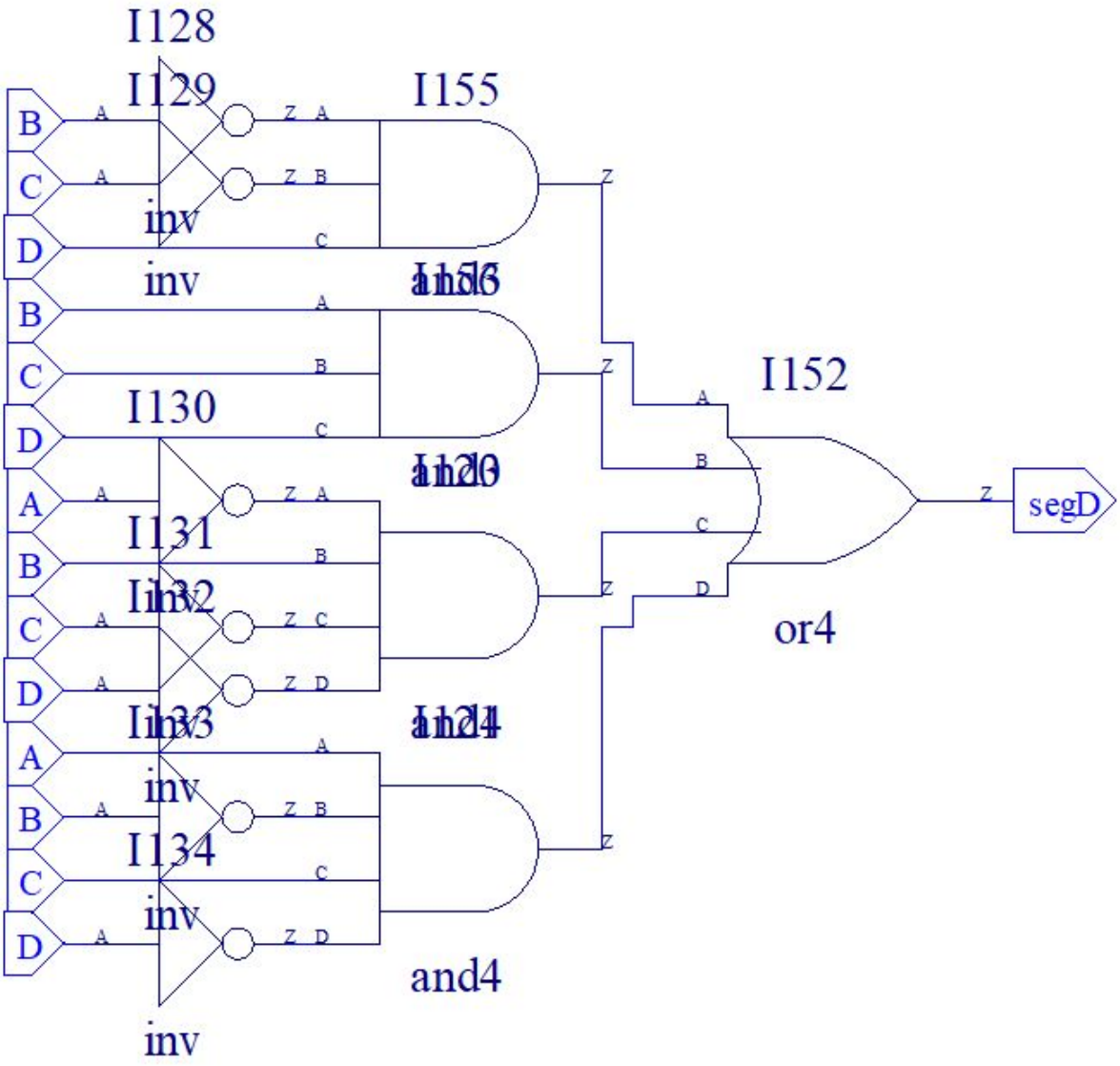

Figure 4: Schematic for segment D

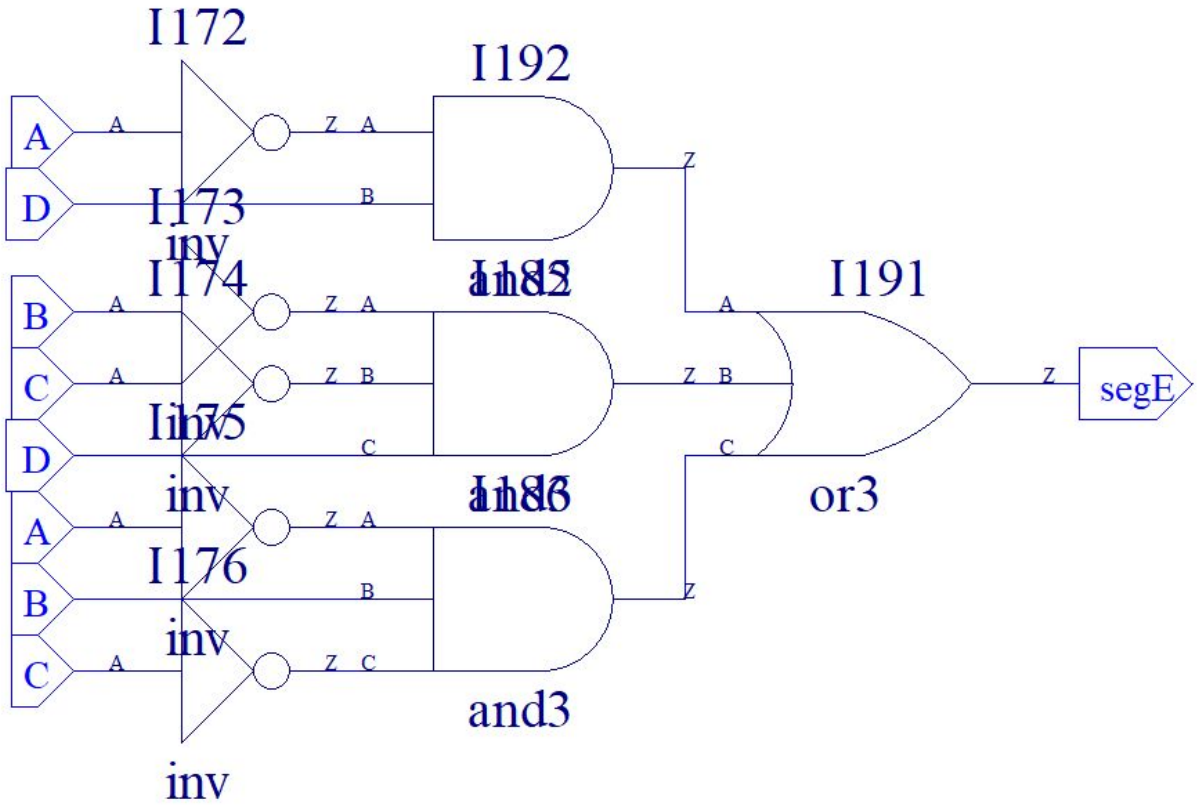

Figure 5: Schematic for segment E

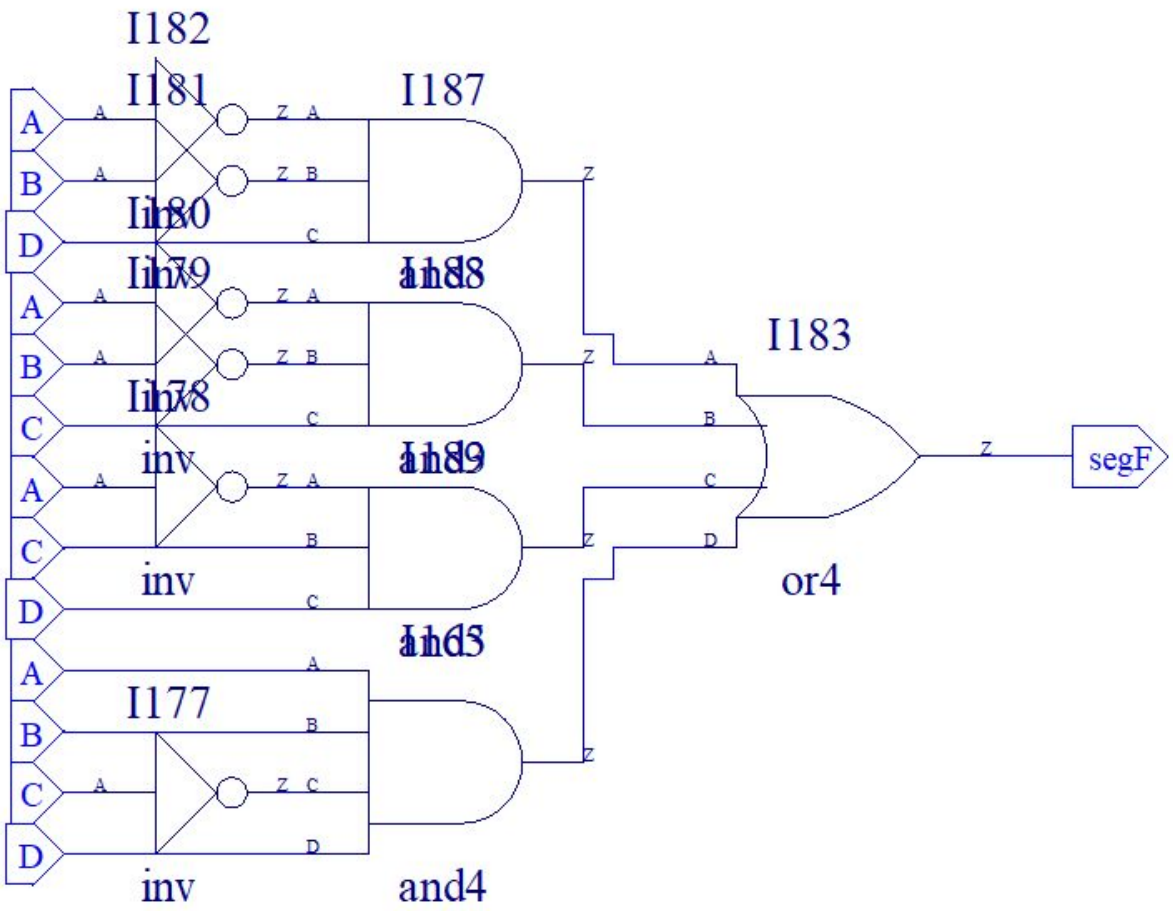

Figure 6: Schematic for segment F

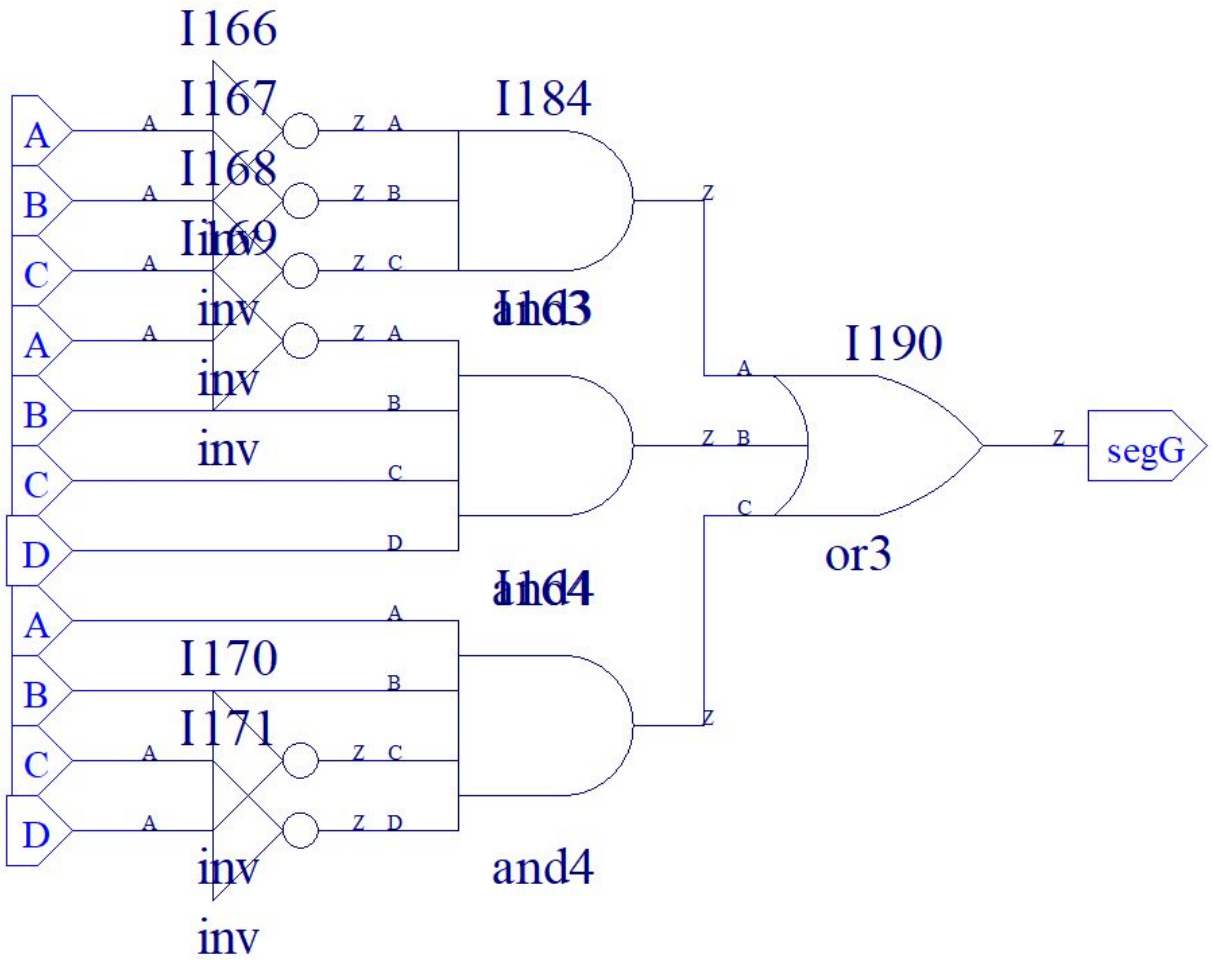

Figure 7: Schematic for segment G

	FPGA PIN	PULLMODE
INPUT A	F5	UP
INPUT B	E3	UP
INPUT C	B1	UP
INPUT D	C1	UP
OUTPUT SEG_A	G14	DOWN
OUTPUT SEG_B	B16	DOWN
OUTPUT SEG_C	D14	DOWN
OUTPUT SEG_D	F14	DOWN
OUTPUT SEG_E	D16	DOWN
OUTPUT SEG_F	C15	DOWN
OUTPUT SEG_G	E16	DOWN

Table 1: Chosen Pins and Pull Modes

I chose pins A3, A4, and A5 for my input pins as shown in the figure above. I set these to pullmode up because the button board buttons are active low. Meaning that they are sending a 1 when not pushed down and a 0 when pushed down. If I had set my pins to pullmode down I would not see a difference when the buttons were pressed or not.

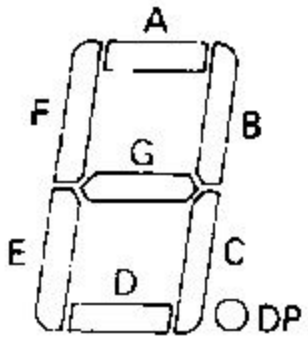

Figure 9: The seven segments and which places they correspond to.

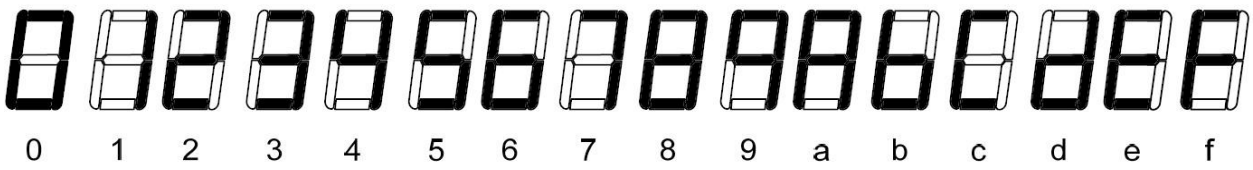

Figure 9: Hex numbers on seven segment display

INPUT (hex)	INPUT (4-bit bin)	SEG_A	SEG_B	SEG_C	SEG_D	SEG_E	SEG_F	SEG_G
0	0000	0	0	0	0	0	0	1
1	0001	1	0	0	1	1	1	1
2	0010	0	0	1	0	0	1	0
3	0011	0	0	0	0	1	1	0
4	0100	1	0	0	1	1	0	0
5	0101	0	1	0	0	1	0	0
6	0110	0	1	0	0	0	0	0
7	0111	0	0	0	1	1	1	1
8	1000	0	0	0	0	0	0	0
9	1001	0	0	0	1	1	0	0
A	1010	0	0	0	1	0	0	0
B	1011	1	1	0	0	0	0	0
C	1100	0	1	1	0	0	0	1
D	1101	1	0	0	0	0	1	0
E	1110	0	1	1	0	0	0	0
F	1111	0	1	1	1	0	0	0

Table 2: Seven-segment decoder truth table

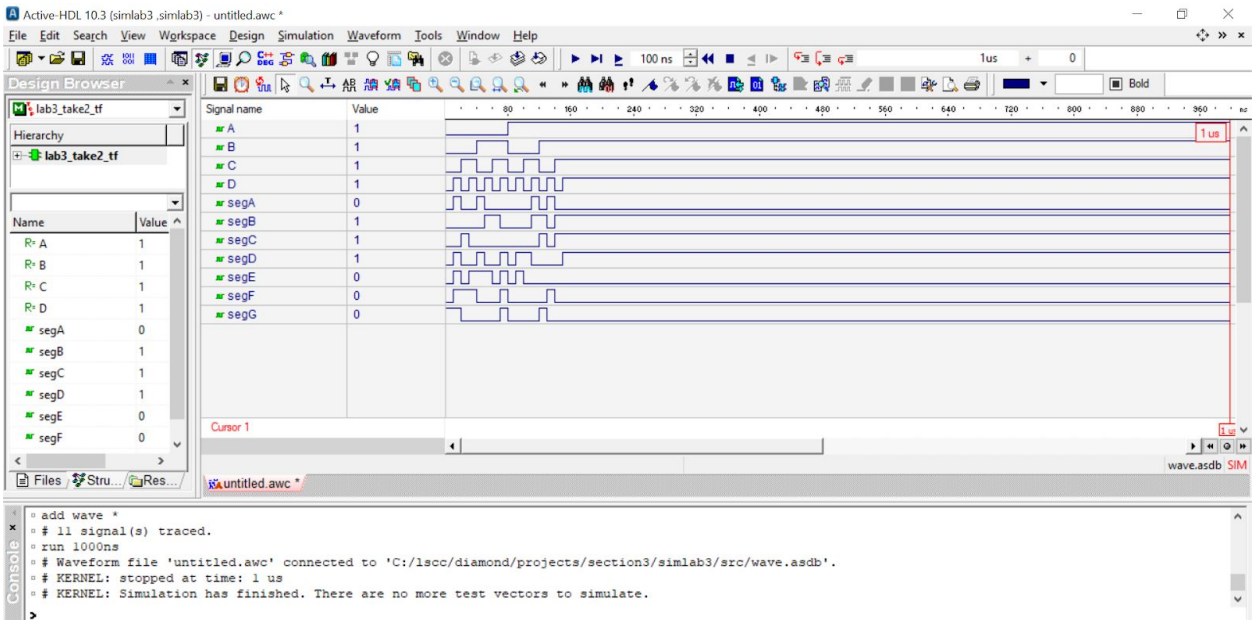

Figure 10: lab 3 simulation

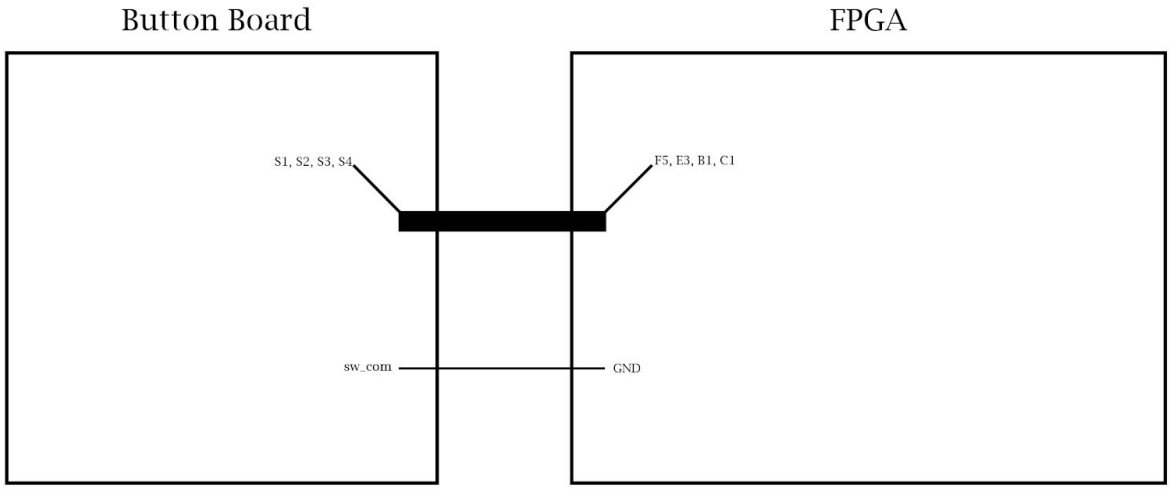

Figure 11a: Block Diagram for Hardware

This diagram shows what buttons the design uses as well as what pin needs to be grounded from the button board. This shows that we will be using the first 4 buttons and what pins they will be connected to.

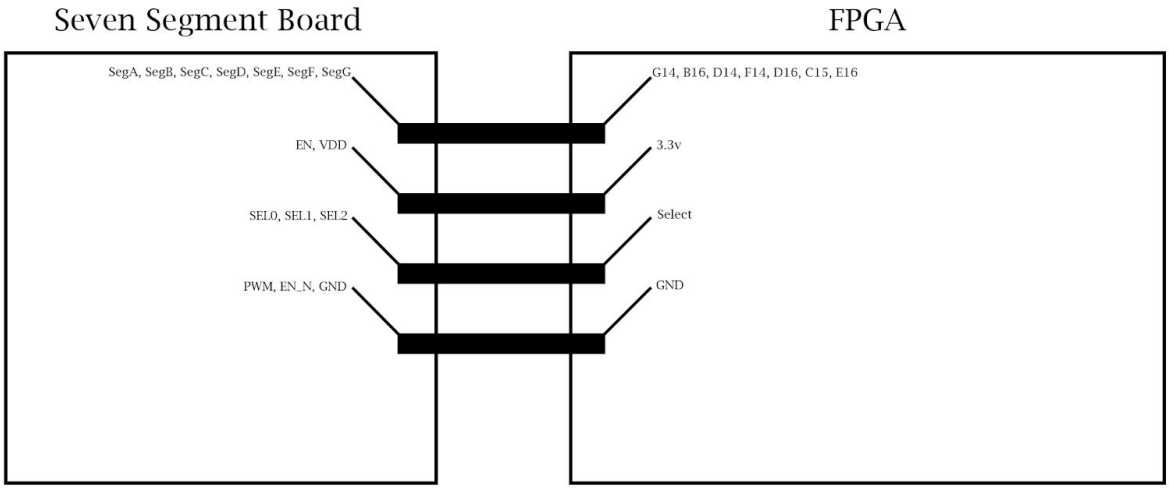

Figure 11b: Block Diagram for Hardware

These diagrams show what needs to be connected to ground, 3.3v and other controlling wires. More interestingly, it shows that the select pins are not inherently set and it is up to the user to select what digit they would like to test.

3. Kmaps

K-maps are useful for efficiently creating the most efficient logic for a desired outcome. Doing K-maps by hand like we do in lab is usually not the way to go now that computers are able to quickly process huge k-maps with way more than 4 variables that we do by hand in this lab. But for the purpose of learning, doing the k-maps by hand is good practice, and gives some appreciation for the computer programs that can pump out k-maps like it's nothing.

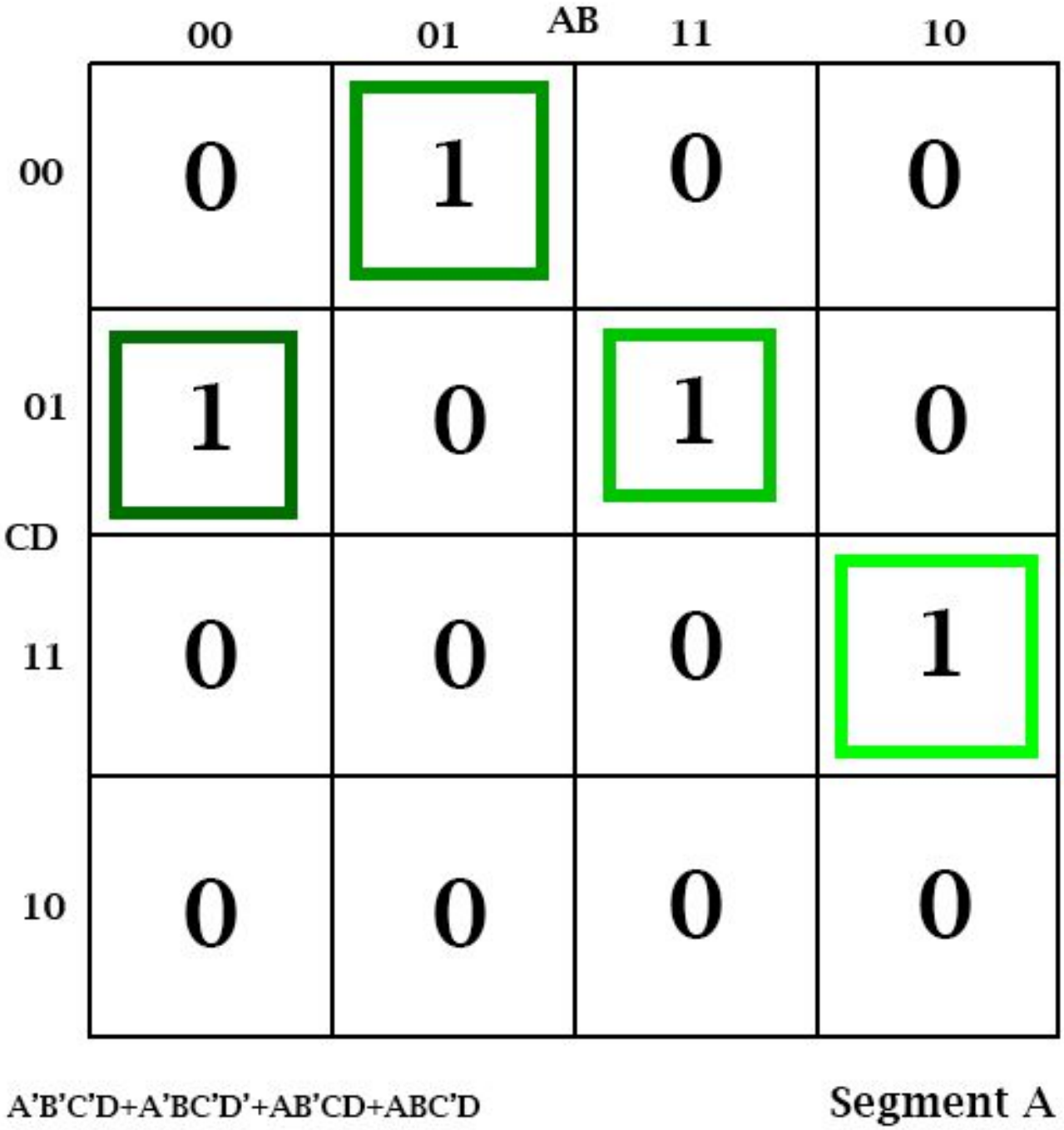

Figure 12: K-map for Segment-A

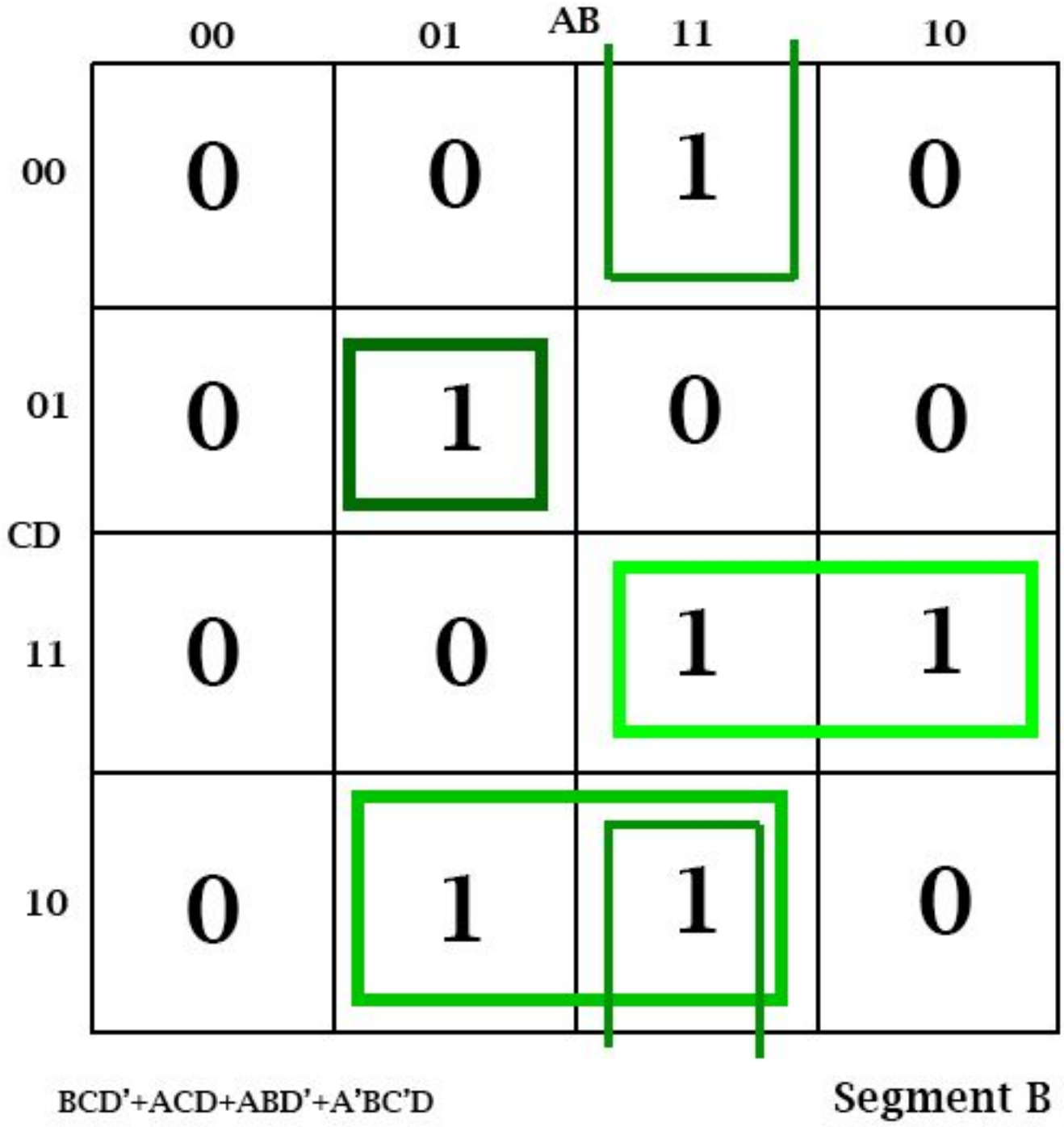

Figure 13: K-map for Segment-B

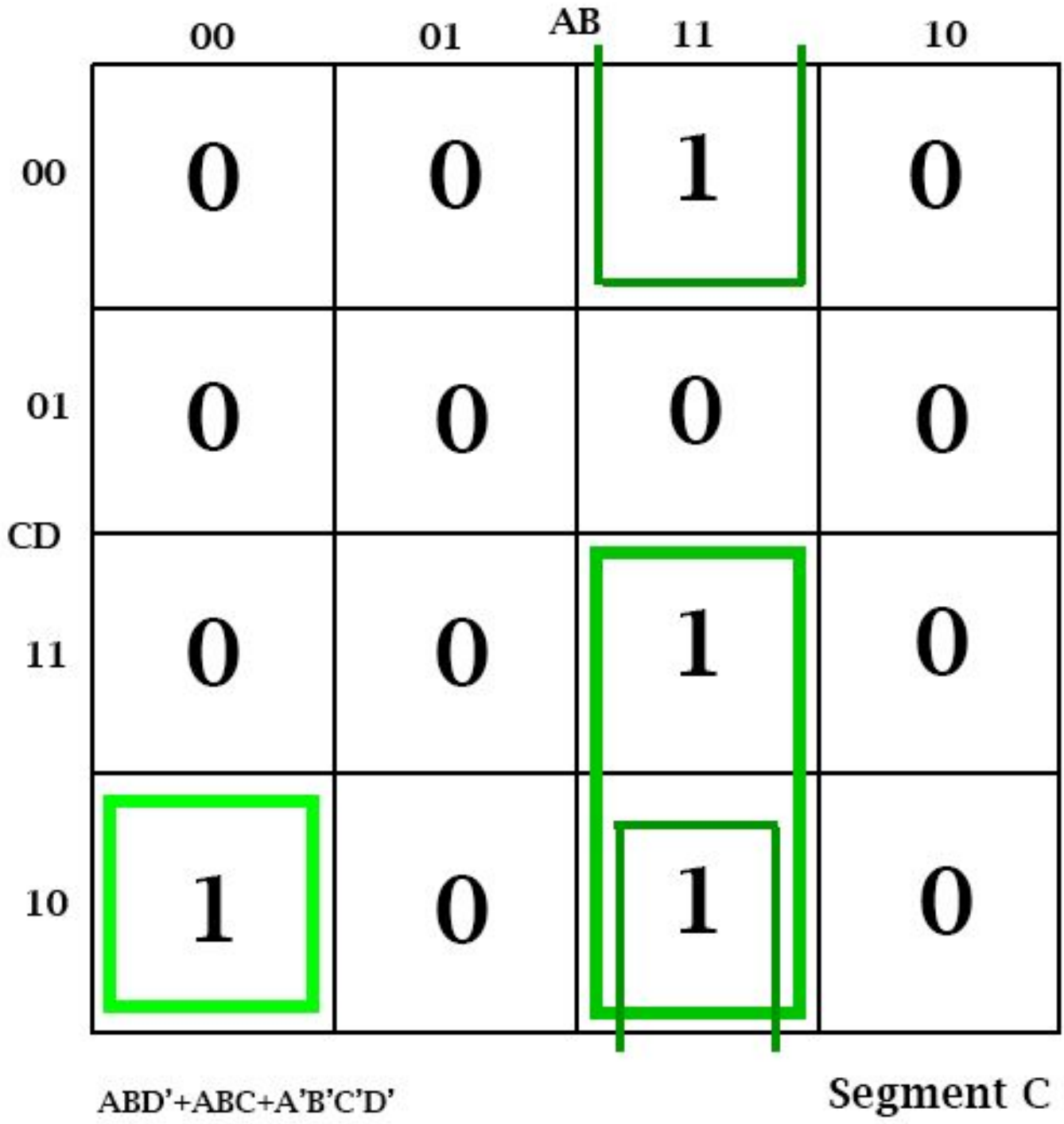

Figure 14: K-map for Segment-C

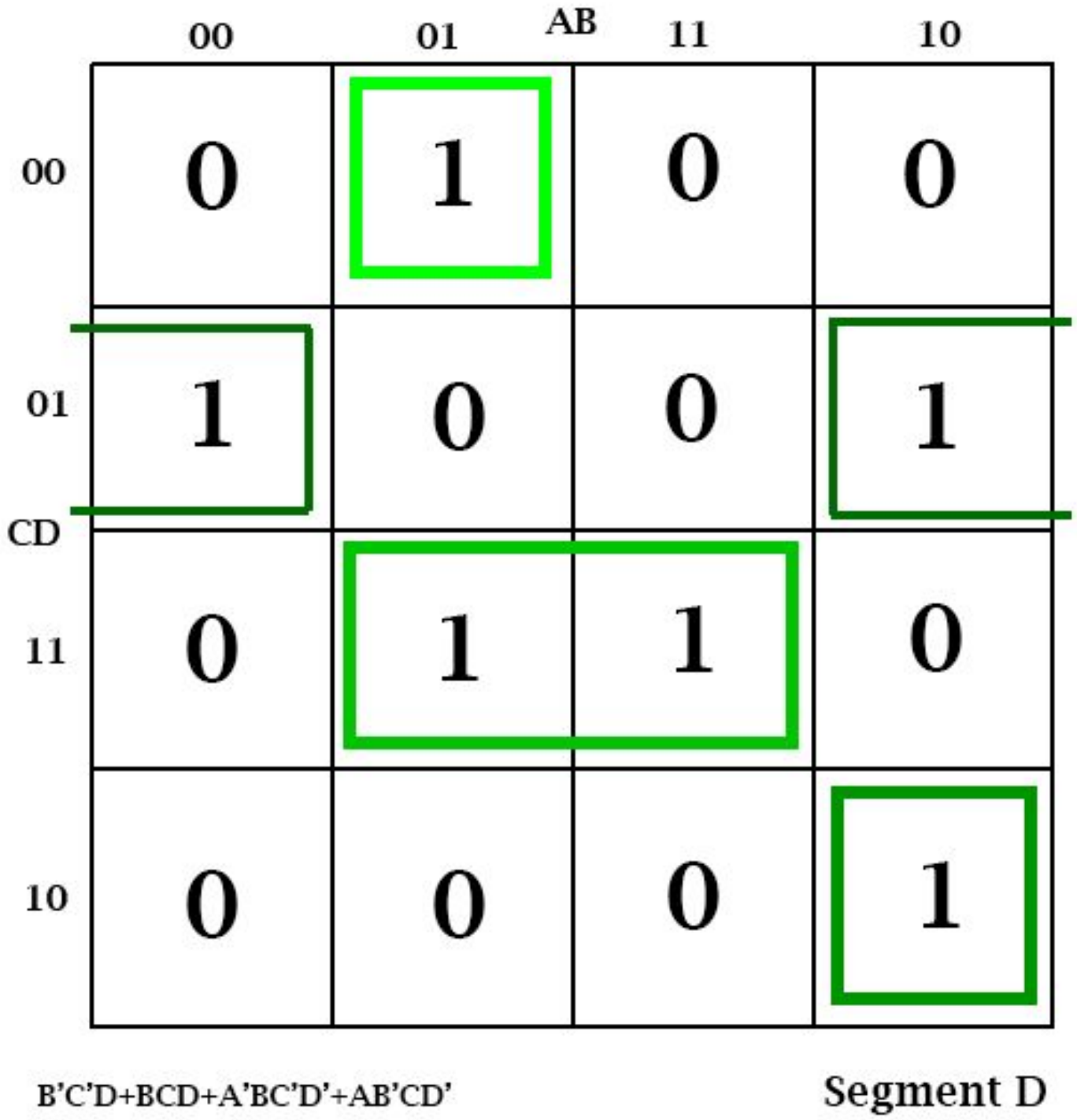

Figure 15: K-map for Segment-D

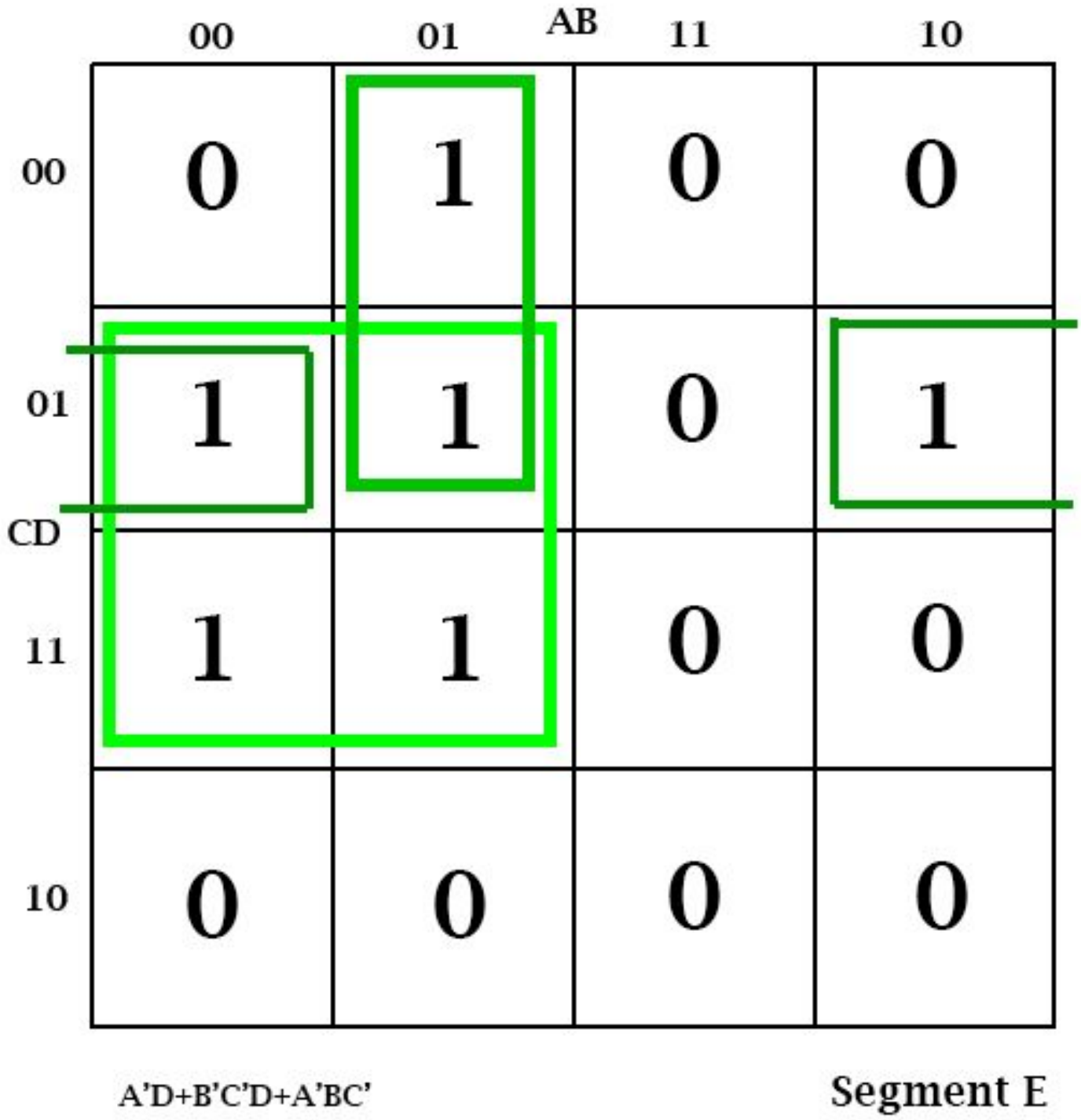

Figure 16: K-map for Segment-E

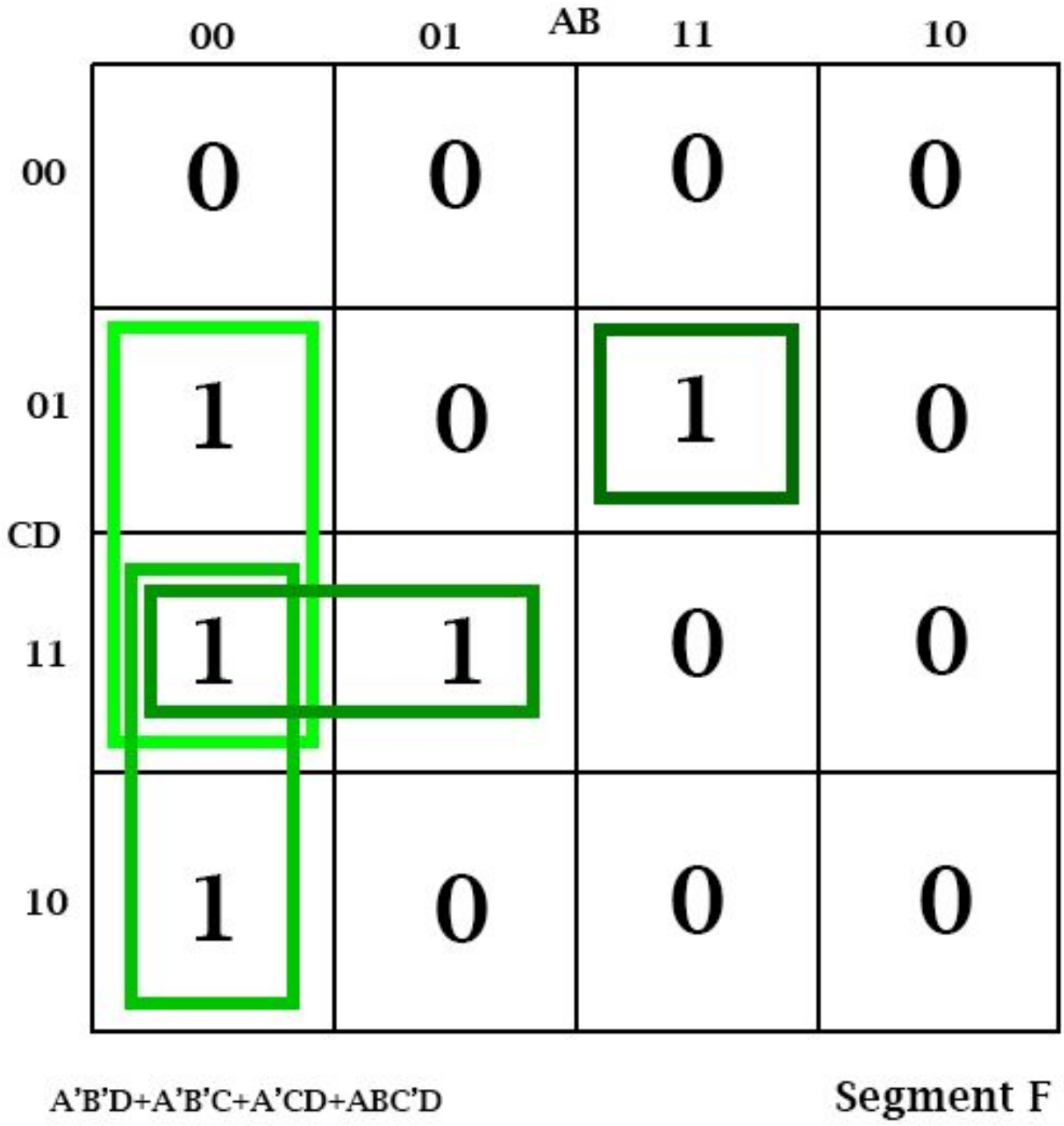

Figure 17: K-map for Segment-F

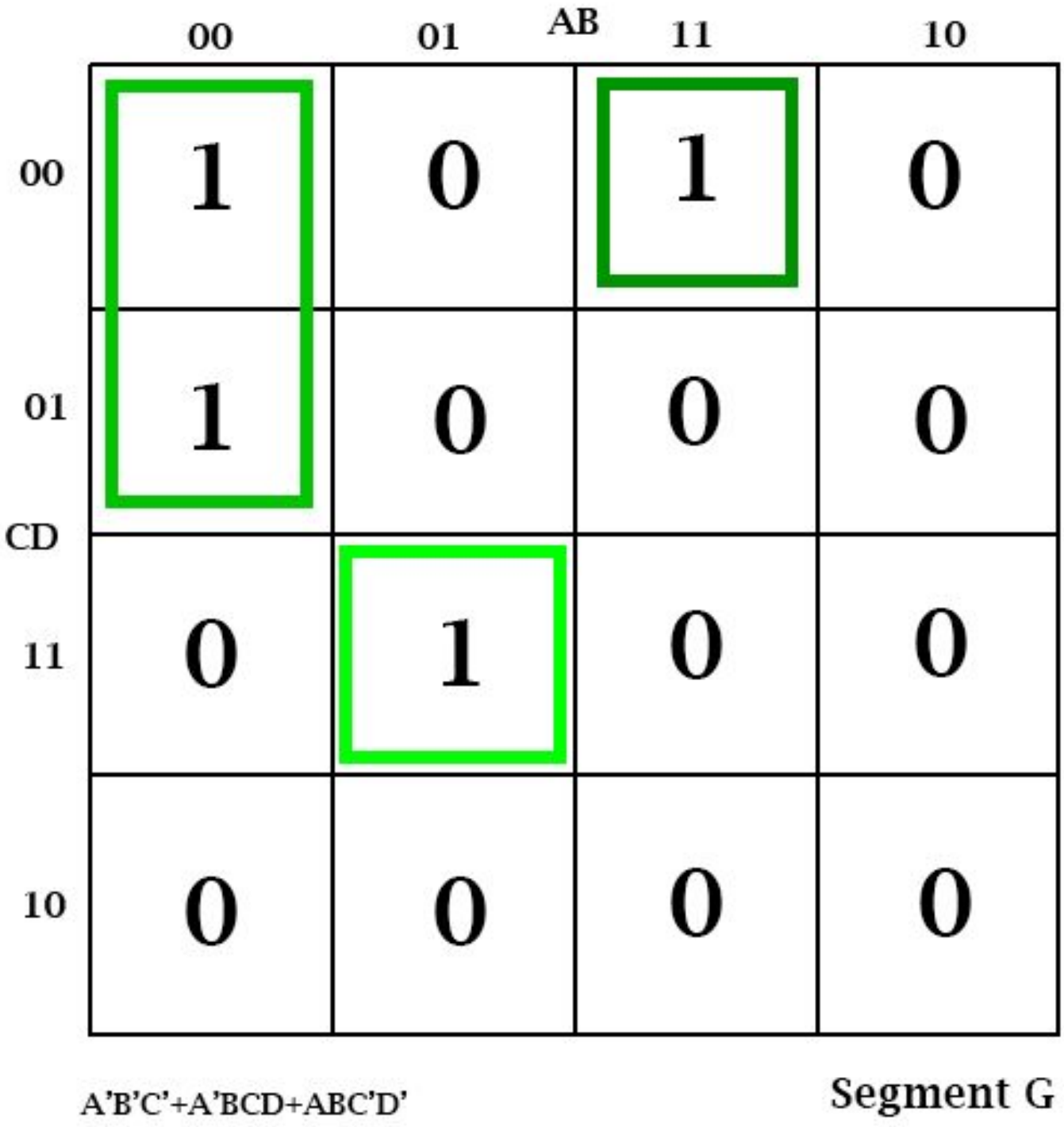

Figure 18: K-map for Segment-G

4. Results

Results were as expected and matched my table from the beginning of the lab. The seven-segment display worked properly on each of the digit selections as well.

5. Experiment Notes

This lab took a long time to complete. I was rusty on K-maps so I repeatedly messed things up. I also had the pullmodes reversed accidentally for a while and that took a long time for me to notice as well. Using the simulation definitely helped because I didn't have to program and reprogram my FPGA over and over again in order to debug my stuff. One other hiccup I ran into was my soldering was bad. So I ended up redoing a lot of my pins solder and just soldered every pin location so I didn't have to worry about a pin being soldered or not.

Study Questions:

1. **When is a simulation necessary? Was it useful for this section?**
   A simulation is necessary for when something will take a long time to program, something is only able to be programmed once, or if seeing a table of outputs vs inputs would be helpful to see. It was useful for this section because I could see when things didn't line up with my table that I had made at the start of the lab of what the inputs vs outputs should be.

Appendix

No code this time.