# Lab 7: Karnaugh Maps

Submission timestamps will be checked and enforced strictly by the CourseWeb; **late submissions will not be accepted**. Check the due date of this lab on the CourseWeb. Remember that, per the course syllabus, if you are not marked by your recitation instructor as having attended a recitation, your score will be cut in half.

In this lab, you will develop a circuit that displays a short binary-coded decimal (BCD) number with seven-segment display.

Computing Minimized Boolean Expressions

For this assignment, we will use the "7-Segment Display" component which can be found under the "Input/Output" library. The seven-segment display unit takes eight inputs, each of which maps directly to a specific segment in the display, as shown below. Take care **not** to use the similar-looking "Hex Digit Display", which only takes two inputs.
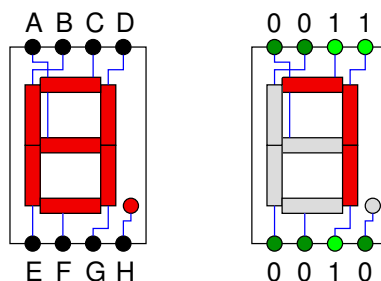
Observe, using the truth table provided in Table 1, how a four-bit BCD input can be used to map to the eight output signals necessary to display a digit with the appropriate appearance. For example, the digit 7, encoded as 0111 which result in segment C, D, and G being on, while segment A, B, E, F, and H remain off as shown above.

Your task is to create a Karnaugh map (or "K-map") for each of the output A through H **from the values presented in the truth table in Table 1**. A K-map is an alternative representation of a truth table which allows us to minimize expressions in Boolean algebra, which will in turn allow us to simplify the circuits we build to represent those Boolean expressions.

To help you get started, let's walk through building the K-map for segment A. We start by arranging our input – the four bits b_3, b_2, b_1, and b_0 – along the sides of the grid, ordered in **Gray code**. We then fill in the desired output values for segment A in the appropriate square of the K-map grid.

Next, we find grounds of adjacent 1s in the K-map to obtain the minterms. Remember that value minterm groupings must be rectangular, must have an area that is a power of two, and should be as large as possible without containing 0s. Keep in mind that groupings can wrap around any edges of the K-map grid. Multiple groupins may also overlap, if necessary, in order to make each one larger. **For segment A, the resulting K-map with one possible set of optimal groupings is shown in Table 2.** Finally, we use these groupings to write a minimized expression for
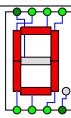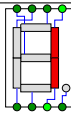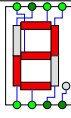
$$A = b_2 b_1' + b_3 + b_2' b_1 + b_1 b_0'$$
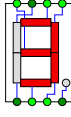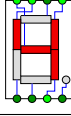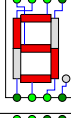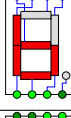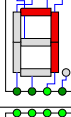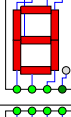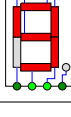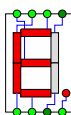
Input Digit					Seven-Segment Display								
Decimal	BCD Encoding				Appearance	Output Signals							
	b_3	b_2	b_1	b_0		A	B	C	D	E	F	G	H
0	0	0	0	0		0	1	1	1	1	1	1	0
1	0	0	0	1		0	0	0	1	0	0	1	0
2	0	0	1	0		1	0	1	1	1	1	0	0
3	0	0	1	1		1	0	1	1	0	1	1	0
4	0	1	0	0		1	1	0	1	0	0	1	0
5	0	1	0	1		1	1	1	0	0	1	1	0
6	0	1	1	0		1	1	0	0	1	1	1	0
7	0	1	1	1		0	0	1	1	0	0	1	0
8	1	0	0	0		1	1	1	1	1	1	1	0
9	1	0	0	1		1	1	1	1	0	1	1	0
Error	1	0	1	0		1	1	1	0	1	1	0	1
	1	0	1	1									
	1	1	0	0									
	1	1	0	1									
	1	1	1	0									
	1	1	1	1									

Table 1. Truth table for a seven-segment display. Invalid BCD input results in the error output, which activates segment **H**, the decimal point.

$b_3 b_2$ \\ $b_1 b_0$	00	01	11	10
00	0	0	1	1
01	1	1	0	1
11	1	1	1	1
10	1	1	1	1

$$b_2 b_1' + b_3 + b_2' b_1 + b_1 b_0'$$

Table 2. Karnaugh map for segment A of the seven-segment display described in Figure 2.
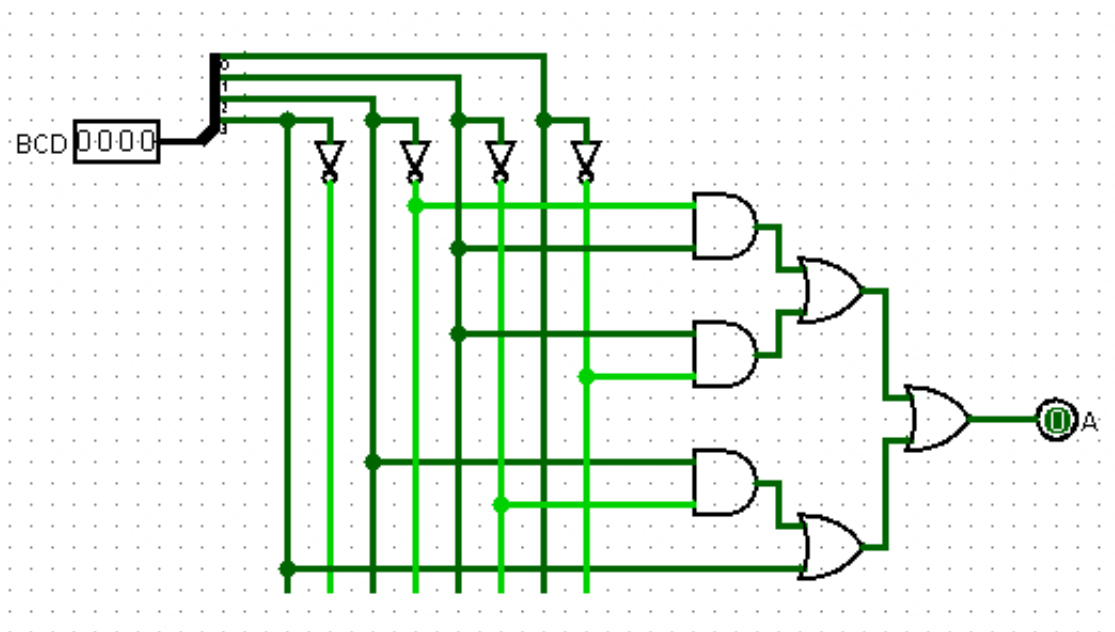

Figure 2. Circuit for segment A, based on the minimized formula $b_2 b_1' + b_3 + b_2' b_1 + b_1 b_0'$

Compute the Karnaugh maps and minimized expressions for segments A through H. If you write this out (neatly) by hand, scan it and convert to PDF as `lab07.pdf`. Alternatively, you may submit your K-maps and expressions as a Word document as `lab07.docx`.

Building a Two-Digit BCD Display

Now that we have the minimal expressions for output A thorough H, let's use them to build a sub-circuit which takes a BCD-encoded digit as a four-bit input and outputs the eight one-bit signals necessary to produce the appropriate digit on a seven-segment display.

Download the file `lab07.circ` from the CourseWeb under this lab. In the file, there are two circuits, "main" and "BCD to Seven Segment". Double click on the "BCD to Seven Segment"

Lab 7: Karnaugh Maps

circuit and you will find a partially build circuit for the segment A as we discussed earlier.

Complete the "BCD to Seven Segment" circuit by constructing the circuits for segments B through H based on your work from Part 1 (Karnaugh Maps). You can simply extend the main lines from your input segnals downward, and branch off the necessary AND gates and OR gates. **Outputs with their appropriate labels have been placed in the circuit for you. Be sure to sue them**.

Once your "BCD to Seven Segment" circuit is completed, return to your main circuit by **double-click** "main" in the left-hand pane. To **instantiate** the "BCD to Seven Segment" subcircuit here, **single-click** on the "BCD to Seven Segment" entry, then click in the main circuit to place it somewhere. It will appear as a small rectangle. If you did everything correctly, you should see nine pins for this component: one for a four-bit input, and eight for the one-bit output.

Now, let's build a two-digit BCD display. Place **two** of your "BCD to Seven Segment" subcircuits in the main circuit, alongside **two** seven-segment display units, and an eight-bit input pin. Use a splitter to split the eight-bit input into two four-bit signals – one for the "upper" digit and one for the "lower" digit. Send one of these signals to each of your "BCD to Seven Segment" subcircuits, then connect the corresponding outputs to the seven segment displays.
The end result should somewhat resemble the circuit shown below. Note that, depending on the layout of your "BCD to Seven Segment" subcircuit, the pins may be laid out differently on the instantiated component, so your wires may connect in a different arrangement from what is shown below. As always, we can mouse over these pins to see that labels we created earlier, to help us determine what's what and connect our wires accordingly.
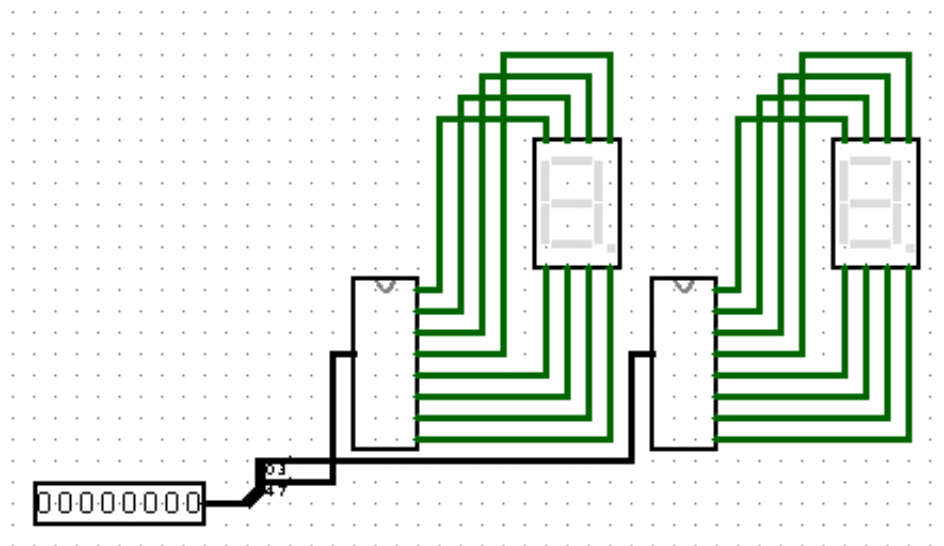
Lastly, **poke your input bits** to make sure that everything is working as expected. Try encoding various decimal numbers into BCD, such as 12, 47, 86, and ensure that the appearance of each digit is as you expect. Try to make each BCD-encoded digit invalid and ensure that the error display works properly.

If you encounter any unexpected results which stem from the logic of your subcircuit, simply double-click on the subcircuit name in the left-hand pane, and modify the subcircuit accordingly. When

---

you return to the main circuit, each instance will take on the updated behavior.

Submission

Zip the file `lab07.circ` together with either `lab07.pdf` or `lab07.docx` to a new file named `lab07.zip`. Then submit the file `lab07.zip` via CourseWeb before the due date stated on the CourseWeb.