

## Karnaugh mapping

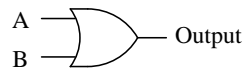
This worksheet and all related files are licensed under the Creative Commons Attribution License, version 1.0. To view a copy of this license, visit <http://creativecommons.org/licenses/by/1.0/>, or send a letter to Creative Commons, 559 Nathan Abbott Way, Stanford, California 94305, USA. The terms and conditions of this license allow for free copying, distribution, and/or modification of all licensed works by the general public.

---

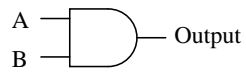
Resources and methods for learning about these subjects (list a few here, in preparation for your research):

# Question 1

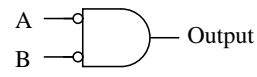
Identify each of these logic gates by name, and complete their respective truth tables:



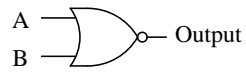
A	B	Output
0	0	
0	1	
1	0	
1	1	



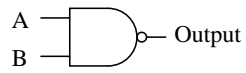
A	B	Output
0	0	
0	1	
1	0	
1	1	



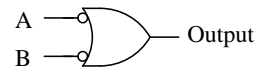
A	B	Output
0	0	
0	1	
1	0	
1	1	



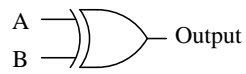
A	B	Output
0	0	
0	1	
1	0	
1	1	



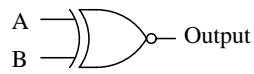
A	B	Output
0	0	
0	1	
1	0	
1	1	



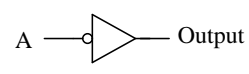
A	B	Output
0	0	
0	1	
1	0	
1	1	



A	B	Output
0	0	
0	1	
1	0	
1	1	

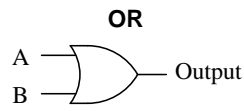


A	B	Output
0	0	
0	1	
1	0	
1	1	

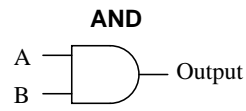


A	Output
0	
1	

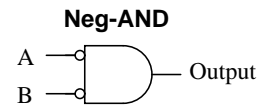
file 01249



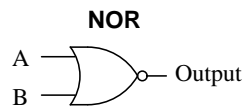
A	B	Output
0	0	0
0	1	1
1	0	1
1	1	1



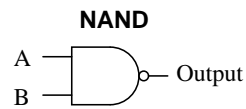
A	B	Output
0	0	0
0	1	0
1	0	0
1	1	1



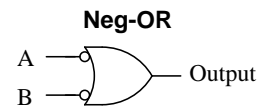
A	B	Output
0	0	1
0	1	0
1	0	0
1	1	0



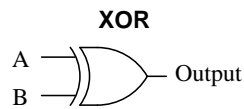
A	B	Output
0	0	1
0	1	0
1	0	0
1	1	0



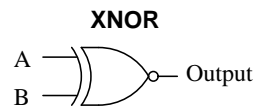
A	B	Output
0	0	1
0	1	1
1	0	1
1	1	0



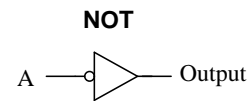
A	B	Output
0	0	1
0	1	1
1	0	1
1	1	0



A	B	Output
0	0	0
0	1	1
1	0	1
1	1	0



A	B	Output
0	0	1
0	1	0
1	0	0
1	1	1



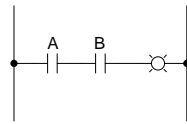
A	Output
0	1
1	0

## Notes 1

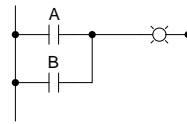
In order to familiarize students with the standard logic gate types, I like to given them practice with identification and truth tables each day. Students need to be able to recognize these logic gate types at a glance, or else they will have difficulty analyzing circuits that use them.

## Question 2

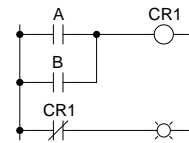
Identify each of these relay logic functions by name (AND, OR, NOR, etc.) and complete their respective truth tables:



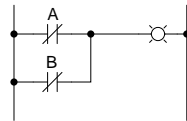
A	B	Output
0	0	
0	1	
1	0	
1	1	



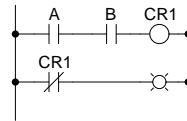
A	B	Output
0	0	
0	1	
1	0	
1	1	



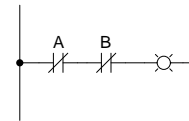
A	B	Output
0	0	
0	1	
1	0	
1	1	



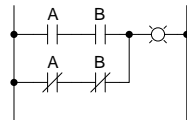
A	B	Output
0	0	
0	1	
1	0	
1	1	



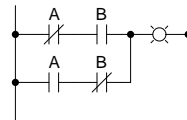
A	B	Output
0	0	
0	1	
1	0	
1	1	



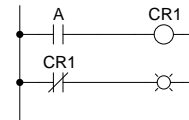
A	B	Output
0	0	
0	1	
1	0	
1	1	



A	B	Output
0	0	
0	1	
1	0	
1	1	

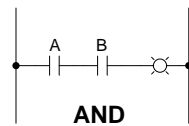


A	B	Output
0	0	
0	1	
1	0	
1	1	



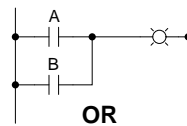
A	Output
0	
1	

file 01335



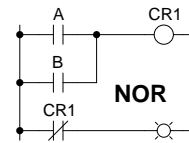
**AND**

A	B	Output
0	0	0
0	1	0
1	0	0
1	1	1



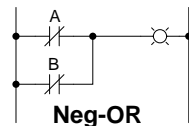
**OR**

A	B	Output
0	0	0
0	1	1
1	0	1
1	1	1



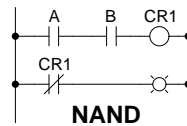
**NOR**

A	B	Output
0	0	1
0	1	0
1	0	0
1	1	0



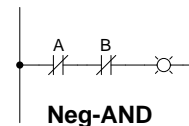
**Neg-OR**

A	B	Output
0	0	1
0	1	1
1	0	1
1	1	0



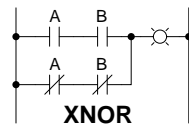
**NAND**

A	B	Output
0	0	1
0	1	1
1	0	1
1	1	0



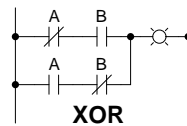
**Neg-AND**

A	B	Output
0	0	1
0	1	0
1	0	0
1	1	0



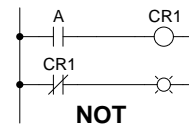
**XNOR**

A	B	Output
0	0	1
0	1	0
1	0	0
1	1	1



**XOR**

A	B	Output
0	0	0
0	1	1
1	0	1
1	1	0



**NOT**

A	Output
0	1
1	0

## Notes 2

In order to familiarize students with standard switch contact configurations, I like to given them practice with identification and truth tables each day. Students need to be able to recognize these ladder logic sub-circuits at a glance, or else they will have difficulty analyzing more complex relay circuits that use them.

---

### Question 3

A *Karnaugh map* is nothing more than a special form of truth table, useful for reducing logic functions into minimal Boolean expressions.

Here is a truth table for a specific three-input logic circuit:

A	B	C	Out
0	0	0	1
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	0

Complete the following Karnaugh map, according to the values found in the above truth table:

AB \ C	C	
	0	1
00		
01		
11		
10		

[file 02834](#)

---

### Answer 3

AB \ C	C	
	0	1
00	1	1
01	0	1
11	0	0
10	0	1

---

### Notes 3

Showing students that Karnaugh maps are really nothing more than truth tables in disguise helps them to more readily learn this powerful new tool.

---

#### Question 4

A *Karnaugh map* is nothing more than a special form of truth table, useful for reducing logic functions into minimal Boolean expressions.

Here is a truth table for a specific four-input logic circuit:

A	B	C	D	Out
0	0	0	0	0
0	0	0	1	1
0	0	1	0	0
0	0	1	1	0
0	1	0	0	0
0	1	0	1	1
0	1	1	0	0
0	1	1	1	0
1	0	0	0	0
1	0	0	1	1
1	0	1	0	1
1	0	1	1	1
1	1	0	0	0
1	1	0	1	1
1	1	1	0	0
1	1	1	1	1

Complete the following Karnaugh map, according to the values found in the above truth table:

		CD			
		00	01	11	10
AB	00				
	01				
	11				
	10				

file 01310

---

Answer 4

		CD			
		00	01	11	10
AB	00	0	1	0	0
	01	0	1	0	0
	11	0	1	1	0
	10	0	1	1	1

Showing students that Karnaugh maps are really nothing more than truth tables in disguise helps them to more readily learn this powerful new tool.



Question 5

Here is a truth table for a four-input logic circuit:

A	B	C	D	Out
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	0
0	1	0	1	1
0	1	1	0	0
0	1	1	1	1
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	0
1	1	0	0	0
1	1	0	1	1
1	1	1	0	0
1	1	1	1	1

If we translate this truth table into a Karnaugh map, we obtain the following result:

		CD			
		00	01	11	10
AB	00	0	0	0	0
	01	0	1	1	0
	11	0	1	1	0
	10	0	0	0	0

Note how the only 1's in the map are clustered together in a group of four:

		CD			
		00	01	11	10
AB	00	0	0	0	0
	01	0	1	1	0
	11	0	1	1	0
	10	0	0	0	0

If you look at the input variables (A, B, C, and D), you should notice that only two of them actually change within this cluster of four 1's. The other two variables hold the same value for each of these conditions where the output is a "1". Identify which variables change, and which stay the same, for this cluster.

file 01311

---

Answer 5

For this cluster of four 1's, variables A and C are the only two inputs that change. Variables B and D remain the same ( $B = 1$  and  $D = 1$ ) for each of the four "high" outputs.

---

Notes 5

This question introduces students to the Karnaugh reduction principle of detecting *contradictory variables* in a grouped set.

# Question 6

Here is a truth table for a four-input logic circuit:

A	B	C	D	Out
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	0
0	1	0	1	0
0	1	1	0	0
0	1	1	1	0
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	0
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1

If we translate this truth table into a Karnaugh map, we obtain the following result:

		CD			
		00	01	11	10
AB	00	0	0	0	0
	01	0	0	0	0
	11	1	1	1	1
	10	0	0	0	0

Note how the only 1's in the map all exist on the same row:

		CD			
		00	01	11	10
AB	00	0	0	0	0
	01	0	0	0	0
	11	1	1	1	1
	10	0	0	0	0

If you look at the input variables (A, B, C, and D), you should notice that only two of them are constant for each of the "1" conditions on the Karnaugh map. Identify these variables, and remember them.

Now, write an SOP (Sum-of-Products) expression for the truth table, and use Boolean algebra to reduce that raw expression to its simplest form. What do you notice about the simplified SOP expression, in relation to the common variables noted on the Karnaugh map?

Answer 6

For this cluster of four 1's, variables A and B are the only two inputs that remain constant for the four "1" conditions shown in the Karnaugh map. The simplified Boolean expression for the truth table is  $AB$ . See a pattern here?

---

Notes 6

This question strongly suggests to students that the Karnaugh map is a graphical method of achieving a reduced-form SOP expression for a truth table. Once students realize Karnaugh mapping holds the key to escaping arduous Boolean algebra simplifications, their interest will be piqued!

# Question 7

One of the essential characteristics of Karnaugh maps is that the input variable sequences are always arranged in Gray code sequence. That is, you never see a Karnaugh map with the input combinations arranged in binary order:

Proper form

		CD			
		00	01	11	10
AB	00				
	01				
	11				
	10				

Improper form

		CD			
		00	01	10	11
AB	00				
	01				
	10				
	11				

The reason for this is apparent when we consider the use of Karnaugh maps to detect common variables in output sets. For instance, here we have a Karnaugh map with a cluster of four 1's at the center:

AB \ CD		CD			
		00	01	11	10
AB	00	0	0	0	0
	01	0	1	1	0
	11	0	1	1	0
	10	0	0	0	0

Arranged in this order, it is apparent that two of the input variables have the same values for each of the four "high" output conditions. Re-draw this Karnaugh map with the input variables sequenced in binary order, and comment on what happens. Can you still tell which input variables remain the same for all four output conditions?

		CD			
		00	01	10	11
AB	00				
	01				
	10				
	11				

file 01312

AB \ CD	CD			
	00	01	10	11
00	0	0	0	0
01	0	1	0	1
10	0	0	0	0
11	0	1	0	1

Looking at this, we can still tell that  $B = 1$  and  $D = 1$  for all four "high" output conditions, but this is *not* apparent by proximity as it was before.

---

#### Notes 7

You could simply tell your students that the input variables must be sequenced according to Gray code in order for Karnaugh mapping to work as a simplification tool, but this wouldn't explain to students *why* it needs to be such. This question shows students the purpose of Gray code sequencing in Karnaugh maps, by showing them the alternative (binary sequencing), and allowing them to see how the task of seeking noncontradictory variables is complicated by it.

# Question 8

Examine this truth table and corresponding Karnaugh map:

A	B	C	D	Out
0	0	0	0	1
0	0	0	1	0
0	0	1	0	1
0	0	1	1	0
0	1	0	0	0
0	1	0	1	0
0	1	1	0	0
0	1	1	1	0
1	0	0	0	1
1	0	0	1	0
1	0	1	0	1
1	0	1	1	0
1	1	0	0	0
1	1	0	1	0
1	1	1	0	0
1	1	1	1	0

CD \ AB	00	01	11	10
00	1	0	0	1
01	0	0	0	0
11	0	0	0	0
10	1	0	0	1

Though it may not be obvious from first appearances, the four "high" conditions in the Karnaugh map actually belong to the same group. To make this more apparent, I will draw a new (oversized) Karnaugh map template, with the Gray code sequences repeated twice along each axis:

CD \ AB	00	01	11	10	00	01	11	10
00								
01								
11								
10								
00								
01								
11								
10								

Fill in this map with the 0 and 1 values from the truth table, and then see if a grouping of four "high" conditions becomes apparent.

[file 01342](#)

AB \ CD								
	00	01	11	10	00	01	11	10
00	1	0	0	1	1	0	0	1
01	0	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0
10	1	0	0	1	1	0	0	1
00	1	0	0	1	1	0	0	1
01	0	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0
10	1	0	0	1	1	0	0	1

Follow-up question: what does this problem tell us about grouping? In other words, how can we identify groups of "high" states without having to make oversized Karnaugh maps?

---

### Notes 8

The concept of bit groups extending past the boundaries of a Karnaugh map tends to confuse students. In fact, it is about the only thing that tends to confuse students about Karnaugh maps! Simply telling them to group past the borders of the map doesn't really teach them *why* the technique is valid. Here, they should see with little difficulty why the technique works.

And, if for some reason they just can't visualize bit groups past the boundaries of a Karnaugh map, they know they can just draw an oversized map and it will become obvious!



# Question 9

A student is asked to use Karnaugh mapping to generate a minimal SOP expression for the following truth table:

A	B	C	Output
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

Following the truth table shown, the student plots this Karnaugh map:

AB \ C	0	1
	0	1
00	0	0
01	0	1
11	0	1
10	0	1

"This is easy," says the student to himself. "All the '1' conditions fall within the same group!" The student then highlights a triplet of 1's as a single group:

AB \ C	0	1
	0	1
00	0	0
01	0	1
11	0	1
10	0	1

Looking at this cluster of 1's, the student identifies  $C$  as remaining constant (1) for all three conditions in the group. Therefore, the student concludes, the minimal expression for this truth table must simply be  $C$ .

However, a second student decides to use Boolean algebra on this problem instead of Karnaugh mapping. Beginning with the original truth table and generating a Sum-of-Products (SOP) expression for it, the simplification goes as follows:

$$\overline{A}BC + A\overline{B}C + ABC$$

$$BC(\overline{A} + A) + A\overline{B}C$$

$$BC + A\overline{B}C$$

$$C(B + A\overline{B})$$

$$C(B + A)$$

$$AC + BC$$

Obviously, the answer given by the second student's Boolean reduction ( $AC + BC$ ) does not match the answer given by the first student's Karnaugh map analysis ( $C$ ).

Perplexed by the disagreement between these two methods, and failing to see a mistake in the Boolean algebra used by the second student, the first student decides to check his Karnaugh mapping again. Upon reflection, it becomes apparent that if the answer really were  $C$ , the Karnaugh map would look different. Instead of having three cells with 1's in them, there would be four cells with 1's in them (the output of the function being "1" any time  $C = 1$ ):

AB \ C	0	1
00	0	1
01	0	1
11	0	1
10	0	1

Somewhere, there must have been a mistake made in the first student's grouping of 1's in the Karnaugh map, because the map shown above is the only one proper for an answer of  $C$ , and it is not the same as the real map for the given truth table. Explain where the mistake was made, and what the proper grouping of 1's should be.

[file 02836](#)

---

Answer 9

Proper grouping of 1's in the Karnaugh map:

AB \ C	0	1
00	0	0
01	0	1
11	0	1
10	0	1

BC

AC

---

Notes 9

The purpose of this question is to illustrate how it is incorrect to identify clusters of arbitrary size in a Karnaugh map. A cluster of three, as seen in this scenario, leads to an incorrect conclusion. Of course, one could easily quote a textbook as to the proper numbers and patterns of 1's to identify in a Karnaugh map, but it is so much more informative (in my opinion) to illustrate by example. Posing a dilemma such as this makes students *think* about why the answer is wrong, rather than asking them to remember seemingly arbitrary rules.

---

Question 10

State the rules for properly identifying common groups in a Karnaugh map.  
file 02837

---

Answer 10

Any good introductory digital textbook will give the rules you need to do Karnaugh mapping. I leave you to research these rules for yourself!

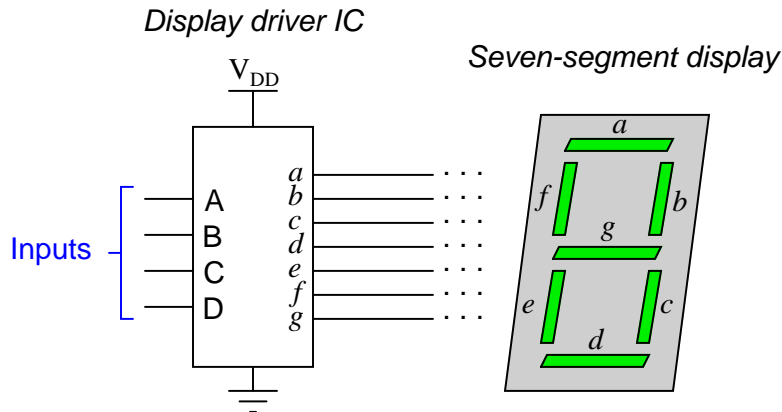
---

Notes 10

The answer speaks for itself here – let your students research these rules, and ask them exactly where they found them (including the page numbers in their textbook(s)!).

# Question 11

A *seven segment decoder* is a digital circuit designed to drive a very common type of digital display device: a set of LED (or LCD) segments that render numerals 0 through 9 at the command of a four-bit code:



The behavior of the display driver IC may be represented by a truth table with seven outputs: one for each segment of the seven-segment display (*a* through *g*). In the following table, a "1" output represents an active display segment, while a "0" output represents an inactive segment:

D	C	B	A	a	b	c	d	e	f	g	Display
0	0	0	0	1	1	1	1	1	1	0	"0"
0	0	0	1	0	1	1	0	0	0	0	"1"
0	0	1	0	1	1	0	1	1	0	1	"2"
0	0	1	1	1	1	1	1	0	0	1	"3"
0	1	0	0	0	1	1	0	0	1	1	"4"
0	1	0	1	1	0	1	1	0	1	1	"5"
0	1	1	0	1	0	1	1	1	1	1	"6"
0	1	1	1	1	1	1	0	0	0	0	"7"
1	0	0	0	1	1	1	1	1	1	1	"8"
1	0	0	1	1	1	1	1	0	1	1	"9"

A real-life example such as this provides an excellent showcase for techniques such as Karnaugh mapping. Let's take output *a* for example, showing it without all the other outputs included in the truth table:

D	C	B	A	a
0	0	0	0	1
0	0	0	1	0
0	0	1	0	1
0	0	1	1	1
0	1	0	0	0
0	1	0	1	1
0	1	1	0	1
0	1	1	1	1
1	0	0	0	1
1	0	0	1	1

Plotting a Karnaugh map for output *a*, we get this result:

BA		00	01	11	10
DC	00	1	0	1	1
	01	0	1	1	1
	11				
	10	1	1		

Identify adjacent groups of 1's in this Karnaugh map, and generate a minimal SOP expression from those groupings.

Note that six of the cells are blank because the truth table does not list all the possible input combinations with four variables (A, B, C, and D). With these large gaps in the Karnaugh map, it is difficult to form large groupings of 1's, and thus the resulting "minimal" SOP expression has several terms.

However, if we do not care about output  $a$ 's state in the six non-specified truth table rows, we can fill in the remaining cells of the Karnaugh map with "don't care" symbols (usually the letter  $X$ ) and use those cells as "wildcards" in determining groupings:

BA		00	01	11	10
DC	00	1	0	1	1
	01	0	1	1	1
	11	X	X	X	X
	10	1	1	X	X

With this new Karnaugh map, identify adjacent groups of 1's, and generate a minimal SOP expression from those groupings.

[file 02838](#)

Karnaugh map groupings with strict "1" groups:

$$\overline{D}B + \overline{D}CA + D\overline{C}\overline{B} + \overline{C}\overline{B}\overline{A}$$

BA \ DC		00	01	11	10
		00	01	11	10
00	00	1	0	1	1
01	01	0	1	1	1
11	11				
10	10	1	1		

Karnaugh map groupings with "don't care" wildcards:

$$D + B + CA + \overline{C}\overline{A}$$

BA \ DC		00	01	11	10
		00	01	11	10
00	00	1	0	1	1
01	01	0	1	1	1
11	11	X	X	X	X
10	10	1	1	X	X

Follow-up question: this question and answer merely focused on the *a* output for the BCD-to-7-segment decoder circuit. Imagine if we were to approach all seven outputs of the decoder circuit in these two fashions, first developing SOP expressions using strict groupings of "1" outputs, and then using "don't care" wildcards. Which of these two approaches do you suppose would yield the simplest gate circuitry overall? What impact would the two different solutions have on the decoder circuit's behavior for the six unspecified input combinations 1010, 1011, 1100, 1101, 1110, and 1111?

#### Notes 11

One of the points of this question is for students to realize that bigger groups are better, in that they yield simpler SOP terms. Also, students should realize that the ability to use "don't care" states as "wildcard" placeholders in the Karnaugh map cells increases the chances of creating bigger groups.

Truth be known, I chose a pretty bad example to try to make an SOP expression from, since there are only two non-zero output conditions out of ten! Formulating a POS expression would have been easier, but that's a subject for another question!

---

### Question 12

When designing a circuit to emulate a truth table such as this where nearly all the input conditions result in "1" output states, it is easier to use Product-of-Sums (POS) expressions rather than Sum-of-Products (SOP) expressions:

A	B	C	Output
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	0

Is it possible to use a Karnaugh map to generate the appropriate POS expression for this truth table, or are Karnaugh maps limited to SOP expressions only? Explain your answer, and how you were able to obtain it.

[file 02839](#)

---

### Answer 12

Yes, you can use Karnaugh maps to generate POS expressions, not just SOP expressions!

---

### Notes 12

I am more interested in seeing students' approach to this problem than acknowledgment of the answer (that Karnaugh maps may be used to generate SOP and POS expressions alike). Setting up a Karnaugh map to see if a POS expression may be obtained for this truth table is not difficult to do, but many students are so unfamiliar/uncomfortable with "experimenting" in this manner than they tend to freeze when presented with a problem like this. Without specific instructions on what to do, the obvious steps of "try it and see" elude them.

It is your charge as their instructor to encourage an experimental mindset among your students. Do not simply tell them how to go about "discovering" the answer on their own, for if you do you will rob them of an authentic discovery experience.

---

### Question 13

Use a Karnaugh map to generate a simple Boolean expression for this truth table, and draw a relay logic circuit equivalent to that expression:

A	B	C	Output
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	0

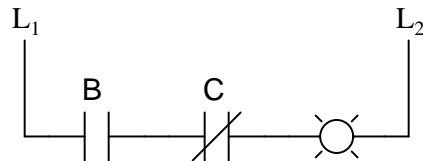
file 02840

---

### Answer 13

Simple expression and relay circuit:

$$B\overline{C}$$



---

### Notes 13

One of the things you may want to have your students share in front of the class is their Karnaugh maps, and how they grouped common output states to arrive at Boolean expression terms. I have found that an overhead (acetate) or computer-projected image of a blank Karnaugh map on a whiteboard serves well to present Karnaugh maps on. This way, cell entries may be easily erased and re-drawn without having to re-draw the map (grid lines) itself.

Ask your students to compare using a Karnaugh map versus using standard SOP/Boolean simplifications to arrive at the simplest expression for this truth table. Which technique would they prefer to use, and why?



---

### Question 14

Use a Karnaugh map to generate a simple Boolean expression for this truth table, and draw a gate circuit equivalent to that expression:

A	B	C	D	Output
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	0
0	1	0	1	0
0	1	1	0	1
0	1	1	1	0
1	0	0	0	0
1	0	0	1	0
1	0	1	0	1
1	0	1	1	1
1	1	0	0	0
1	1	0	1	0
1	1	1	0	1
1	1	1	1	1

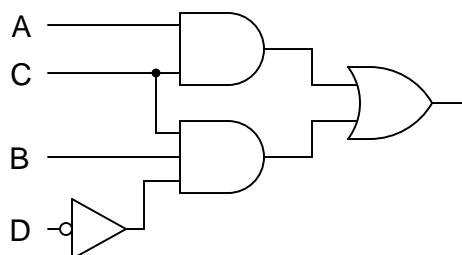
[file 02841](#)

---

### Answer 14

Simple expression and gate circuit:

$$AC + BCD$$



Challenge question: use Boolean algebra techniques to simplify the table's raw SOP expression into minimal form without the use of a Karnaugh map.

---

### Notes 14

One of the things you may want to have your students share in front of the class is their Karnaugh maps, and how they grouped common output states to arrive at Boolean expression terms. I have found that an overhead (acetate) or computer-projected image of a blank Karnaugh map on a whiteboard serves well to present Karnaugh maps on. This way, cell entries may be easily erased and re-drawn without having to re-draw the map (grid lines) itself.

---

### Question 15

Use a Karnaugh map to generate a simple Boolean expression for this truth table, and draw a gate circuit equivalent to that expression:

A	B	C	D	Output
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	0
0	1	0	1	0
0	1	1	0	0
0	1	1	1	0
1	0	0	0	1
1	0	0	1	0
1	0	1	0	1
1	0	1	1	1
1	1	0	0	0
1	1	0	1	0
1	1	1	0	1
1	1	1	1	1

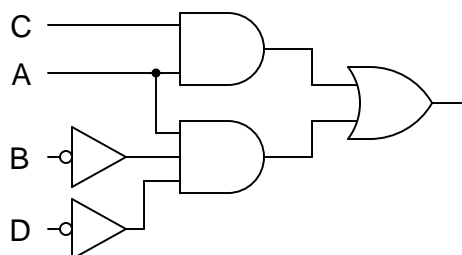
[file 02842](#)

---

### Answer 15

Simple expression and gate circuit:

$$AC + A\overline{B}\overline{D}$$



Challenge question: use Boolean algebra techniques to simplify the table's raw SOP expression into minimal form without the use of a Karnaugh map.

---

### Notes 15

One of the things you may want to have your students share in front of the class is their Karnaugh maps, and how they grouped common output states to arrive at Boolean expression terms. I have found that an overhead (acetate) or computer-projected image of a blank Karnaugh map on a whiteboard serves well to present Karnaugh maps on. This way, cell entries may be easily erased and re-drawn without having to re-draw the map (grid lines) itself.

This is one of those situations where an important group "wraps around" the edge of the Karnaugh map, and thus is likely to be overlooked by students.

---

**Question 16**

Use a Karnaugh map to generate a simple Boolean expression for this truth table, and draw a relay circuit equivalent to that expression:

A	B	C	D	Output
0	0	0	0	1
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	1
0	1	0	1	0
0	1	1	0	1
0	1	1	1	0
1	0	0	0	1
1	0	0	1	0
1	0	1	0	0
1	0	1	1	0
1	1	0	0	1
1	1	0	1	0
1	1	1	0	1
1	1	1	1	0

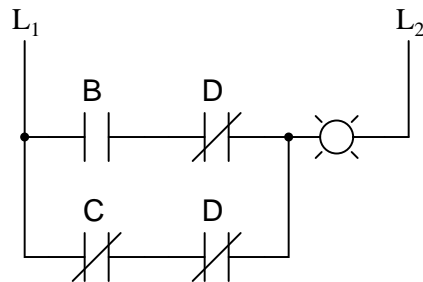
file 02843

---

**Answer 16**

Simple expression and relay circuit:

$$B\overline{D} + \overline{C}\overline{D}$$



Follow-up question: although the relay circuit shown above does satisfy the minimal SOP Boolean expression, there is a way to make it simpler yet. Hint: done properly, you may eliminate one of the contacts in the circuit!

Challenge question: use Boolean algebra techniques to simplify the table's raw SOP expression into minimal form without the use of a Karnaugh map.

One of the things you may want to have your students share in front of the class is their Karnaugh maps, and how they grouped common output states to arrive at Boolean expression terms. I have found that an overhead (acetate) or computer-projected image of a blank Karnaugh map on a whiteboard serves well to present Karnaugh maps on. This way, cell entries may be easily erased and re-drawn without having to re-draw the map (grid lines) itself.

This is one of those situations where an important group "wraps around" the edge of the Karnaugh map, and thus is likely to be overlooked by students.

---

### Question 17

Use a Karnaugh map to generate a simple Boolean expression for this truth table, and draw a relay circuit equivalent to that expression:

A	B	C	D	Output
0	0	0	0	1
0	0	0	1	0
0	0	1	0	1
0	0	1	1	0
0	1	0	0	1
0	1	0	1	1
0	1	1	0	1
0	1	1	1	1
1	0	0	0	1
1	0	0	1	0
1	0	1	0	1
1	0	1	1	0
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1

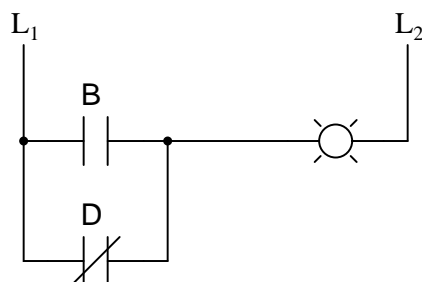
file 02844

---

### Answer 17

Simple expression and relay circuit:

$$B + \overline{D}$$



---

### Notes 17

Given the preponderance of 1's in this truth table, it is rational to try developing a POS expression rather than an SOP expression. However, your students may find that the elegance of Karnaugh mapping makes it easy enough to do it both ways! This is one question where you definitely want to have your students explain their methods of solution in front of the class, and you definitely want them to see how Karnaugh maps could be used both ways (SOP and POS).

I have found that an overhead (acetate) or computer-projected image of a blank Karnaugh map on a whiteboard serves well to present Karnaugh maps on. This way, cell entries may be easily erased and re-drawn without having to re-draw the map (grid lines) itself.