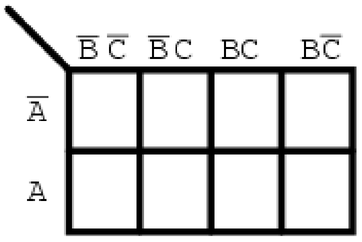
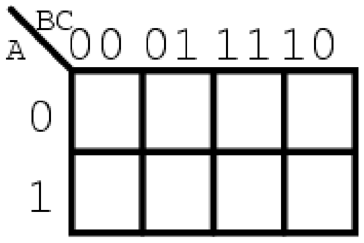
SOME OF PRODUCTS(SOP)

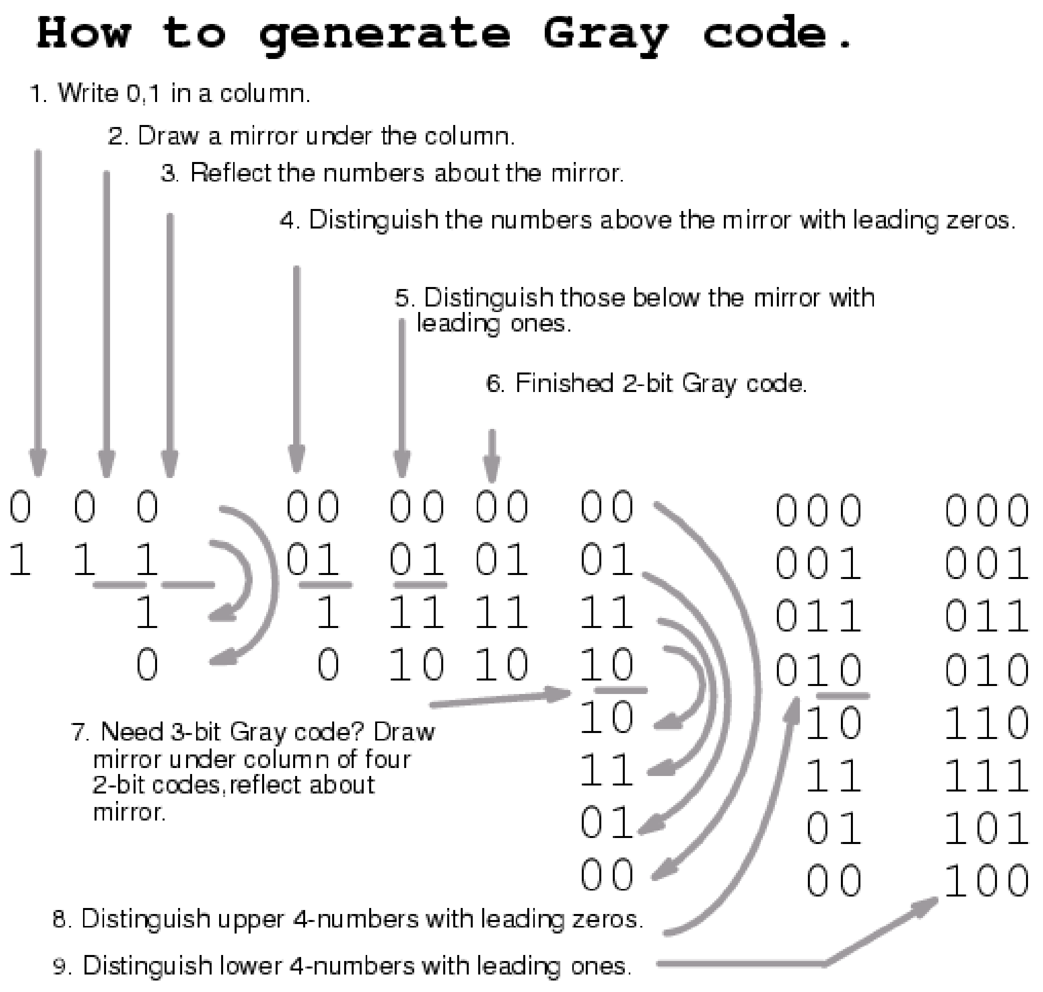
The logic simplification examples that we have done so could have been performed with Boolean algebra about as quickly. Real world logic simplification problems call for larger Karnaugh maps so that we may do serious work. We will work some contrived examples in this section, leaving most of the real world applications for the Combinatorial Logic chapter. By contrived, we mean examples which illustrate techniques. This approach will develop the tools we need to transition to the more complex applications in the Combinatorial Logic chapter.

We show our previously developed Karnaugh map. We will use the form on the right.



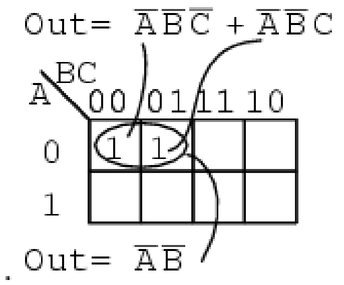


Note the sequence of numbers across the top of the map. It is not in binary sequence which would be **00, 01, 10, 11**. It is **00, 01, 11 10**, which is Gray code sequence. Gray code sequence only changes one binary bit as we go from one number to the next in the sequence, unlike binary. That means that adjacent cells will only vary by one bit, or Boolean variable. This is what we need to organize the outputs of a logic function so that we may view commonality. Moreover, the column and row headings must be in Gray code order, or the map will not work as a Karnaugh map. Cells sharing common Boolean variables would no longer be adjacent, nor show visual patterns. Adjacent cells vary by only one bit because a Gray code sequence varies by only one bit. If we sketch our own Karnaugh maps, we need to generate Gray code for any size map that we may use. This is how we generate Gray code of any size.

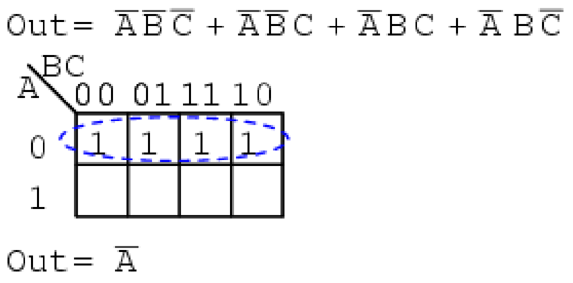

How to generate Gray code.
1. Write 0,1 in a column.
2. Draw a mirror under the column.
3. Reflect the numbers about the mirror.
4. Distinguish the numbers above the mirror with leading zeros.
5. Distinguish those below the mirror with leading ones.
6. Finished 2-bit Gray code.
7. Need 3-bit Gray code? Draw mirror under column of four 2-bit codes,reflect about mirror.
8. Distinguish upper 4-numbers with leading zeros.
9. Distinguish lower 4-numbers with leading ones.
0 1
0 1
0 1 1 0
00 01 1 0
00 01 11 10
00 01 11 10
00 01 11 10 10 11 01 00
000 001 011 010 10 11 01 00
000 001 011 010 110 111 101 100

Note that the Gray code sequence, above right, only varies by one bit as we go down the list, or bottom to top up the list. This property of Gray code is often useful in digital electronics in general. In particular, it is applicable to Karnaugh maps.

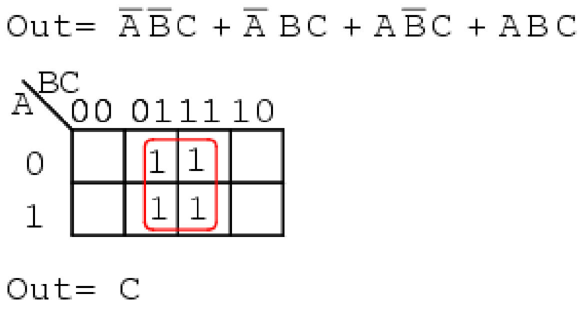
Let us move on to some examples of simplification with 3-variable Karnaugh maps. We show how to map the product terms of the unsimplified logic to the K-map. We illustrate how to identify groups of adjacent cells which leads to a Sum-of-Products simplification of the digital logic.


Out= $\overline{A}\,\overline{B}\,\overline{C} + \overline{A}\,\overline{B}\,C$
BC
A
00 01 11 10
0
1
1 1
Out= $\overline{A}\,\overline{B}$

Above we, place the 1's in the K-map for each of the product terms, identify a group of two, then write a p-term (product term) for the sole group as our simplified result.

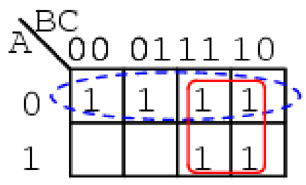

Out= $\overline{A}\,\overline{B}\,\overline{C} + \overline{A}\,\overline{B}\,C + \overline{A}\,B\,C + \overline{A}\,B\,\overline{C}$
BC
A
00 01 11 10
0
1
1 1 1 1
Out= $\overline{A}$

Mapping the four product terms above yields a group of four covered by Boolean **A'**


Out= $\overline{A}\,\overline{B}\,C + \overline{A}\,B\,C + A\,\overline{B}\,C + A\,B\,C$
BC
A
00 01 11 10
0
1
1 1
1 1
Out= C

Mapping the four p-terms yields a group of four, which is covered by one variable **C**.

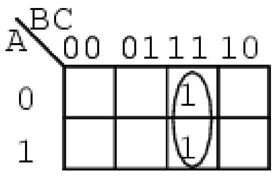
$$Out = \overline{A}\,\overline{B}\,\overline{C} + \overline{A}\,\overline{B}\,C + \overline{A}\,B\,C + \overline{A}\,B\,\overline{C} + ABC + A\,B\,\overline{C}$$



$$Out = \overline{A} + B$$

After mapping the six p-terms above, identify the upper group of four, pick up the lower two cells as a group of four by sharing the two with two more from the other group. Covering these two with a group of four gives a simpler result. Since there are two groups, there will be two p-terms in the Sum-of-Products result **A'+B**

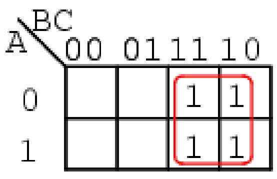
$$Out = \overline{A}\,B\,C + A\,B\,C$$



$$Out = B\,C$$

The two product terms above form one group of two and simplifies to **BC**

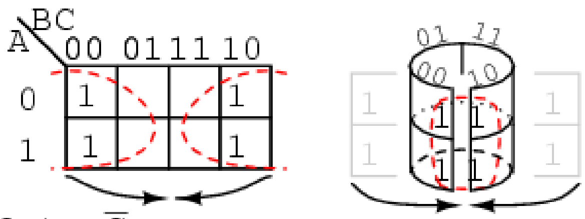
$$Out = \overline{A}\,B\,C + \overline{A}\,B\,\overline{C} + ABC + AB\,\overline{C}$$



$$Out = B$$

Mapping the four p-terms yields a single group of four, which is **B**

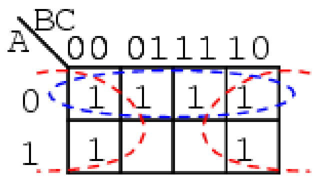
$$Out = \overline{A}\,\overline{B}\,\overline{C} + A\,\overline{B}\,\overline{C} + \overline{A}\,B\,\overline{C} + A\,B\,\overline{C}$$



$$Out = \overline{C}$$

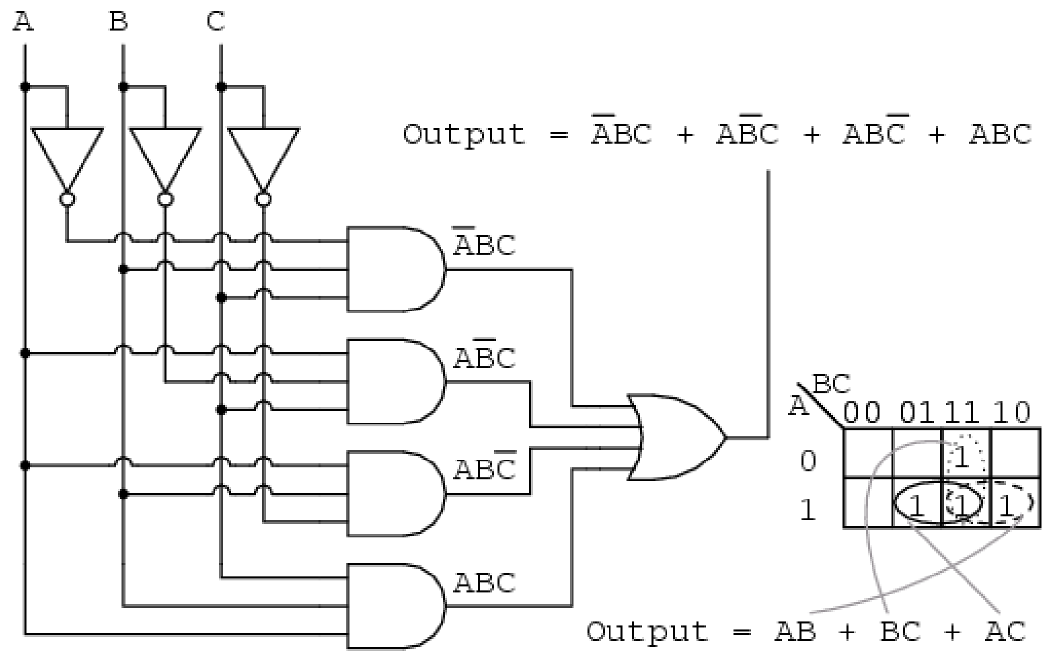
Mapping the four p-terms above yields a group of four. Visualize the group of four by rolling up the ends of the map to form a cylinder, then the cells are adjacent. We normally mark the group of four as above left. Out of the variables A, B, C, there is a common variable: C'. C' is a 0 over all four cells. Final result is **C'**.

$$Out = \overline{A}\,\overline{B}\,\overline{C} + \overline{A}\,\overline{B}\,C + \overline{A}\,B\,C + \overline{A}\,B\,\overline{C} + A\,\overline{B}\,\overline{C} + A\,B\,\overline{C}$$

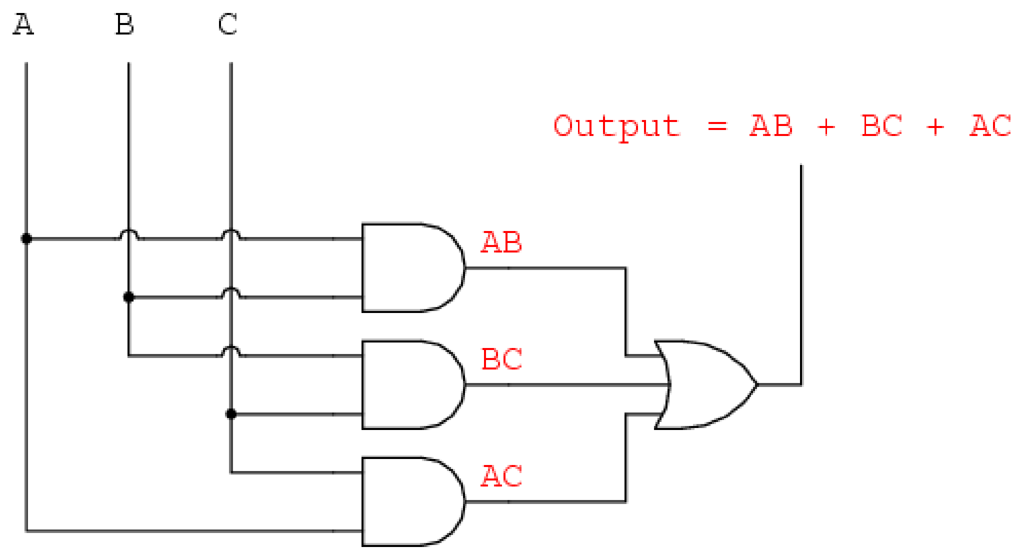


$$Out = \overline{A} + \overline{C}$$

The six cells above from the unsimplified equation can be organized into two groups of four. These two groups should give us two p-terms in our simplified result of **A' + C'**. Below, we revisit the Toxic Waste Incinerator from the Boolean algebra chapter. See Boolean algebra chapter for details on this example. We will simplify the logic using a Karnaugh map.



The Boolean equation for the output has four product terms. Map four 1's corresponding to the p-terms. Forming groups of cells, we have three groups of two. There will be three p-terms in the simplified result, one for each group. See "Toxic Waste Incinerator", Boolean algebra chapter for a gate diagram of the result, which is reproduced below.


A
B
C
Output = AB + BC + AC
AB
BC
AC

Below we repeat the Boolean algebra simplification of Toxic waste incinerator for comparison.

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

$$\Big\downarrow \quad \text{Factoring } \mathbf{BC} \text{ out of 1st and 4th terms}$$

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

$$\Big\downarrow \quad \text{Applying identity } \mathbf{A + \overline{A} = 1}$$

$$BC(1) + A\overline{B}C + AB\overline{C}$$

$$\Big\downarrow \quad \text{Applying identity } \mathbf{1A = A}$$

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

$$\Big\downarrow \quad \text{Factoring } \mathbf{B} \text{ out of 1st and 3rd terms}$$

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

$$\Big\downarrow \quad \text{Applying rule } \mathbf{A + \overline{A}B = A + B} \text{ to the } C + A\overline{C} \text{ term}$$

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

$$\Big\downarrow \quad \text{Distributing terms}$$

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

$$\Big\downarrow \quad \text{Factoring } \mathbf{A} \text{ out of 2nd and 3rd terms}$$

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

$$\Big\downarrow \quad \text{Applying rule } \mathbf{A + \overline{A}B = A + B} \text{ to the } B + \overline{B}C \text{ term}$$

$$BC + A(B + C)$$

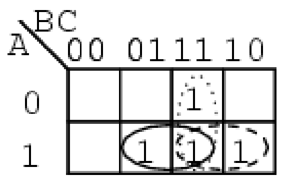
$$\Big\downarrow \quad \text{Distributing terms}$$

$$BC + AB + AC$$

or Simplified result

$$AB + BC + AC$$

Below we repeat the Toxic waste incinerator Karnaugh map solution for comparison to the above Boolean algebra simplification. This case illustrates why the Karnaugh map is widely used for logic simplification.



Output = AB + BC + AC

The Karnaugh map method looks easier than the previous page of boolean algebra.