Chapter 3
Special Section
Focus on Karnaugh Maps
3A.1 Introduction

• Simplification of Boolean functions leads to simpler (and usually faster) digital circuits.
• Simplifying Boolean functions using identities is time-consuming and error-prone.
• This special section presents an easy, systematic method for reducing Boolean expressions.
3A.1 Introduction

• In 1953, Maurice Karnaugh was a telecommunications engineer at Bell Labs.

• While exploring the new field of digital logic and its application to the design of telephone circuits, he invented a graphical way of visualizing and then simplifying Boolean expressions.

• This graphical representation, now known as a Karnaugh map, or Kmap, is named in his honor.
3A.2 Description of Kmaps and Terminology

- A Kmap is a matrix consisting of rows and columns that represent the output values of a Boolean function.

- The output values placed in each cell are derived from the minterms of a Boolean function.

- A minterm is a product term that contains all of the function’s variables exactly once, either complemented or not complemented.
3A.2 Description of Kmaps and Terminology

• For example, the minterms for a function having the inputs $x$ and $y$ are $x'y$, $x'y$, $xy'$, and $xy$.

• Consider the Boolean function, $F(x, y) = xy + xy'$

• Its minterms are:

<table>
<thead>
<tr>
<th>Minterm</th>
<th>$X$</th>
<th>$Y$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x'y'$</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$x'y$</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>$xy'$</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>$xy$</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>
Similarly, a function having three inputs, has the minterms that are shown in this diagram.

<table>
<thead>
<tr>
<th>Minterm</th>
<th>X</th>
<th>Y</th>
<th>Z</th>
</tr>
</thead>
<tbody>
<tr>
<td>X'Y'Z'</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>X'Y'Z</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>X'YZ'</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>X'YZ</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>X'Y'Z</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>XX'Z'</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>X'YZ</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>XYZ</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>
3A.2 Description of Kmaps and Terminology

• A Kmap has a cell for each minterm.

• This means that it has a cell for each line for the truth table of a function.

• The truth table for the function $F(x, y) = xy$ is shown at the right along with its corresponding Kmap.

<table>
<thead>
<tr>
<th>$X$</th>
<th>$Y$</th>
<th>$XY$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

$$F(x, y) = xy$$
3A.2 Description of Kmaps and Terminology

• As another example, we give the truth table and KMap for the function, \( F(x,y) = x + y \) at the right.

• This function is equivalent to the OR of all of the minterms that have a value of 1. Thus:

\[
F(x, y) = x + y = x' y + xy' + xy
\]
3A.3 Kmap Simplification for Two Variables

- Of course, the minterm function that we derived from our Kmap was not in simplest terms.
  - That’s what we started with in this example.
- We can, however, reduce our complicated expression to its simplest terms by finding adjacent 1s in the Kmap that can be collected into groups that are powers of two.
- In our example, we have two such groups.
  - Can you find them?
3A.3 Kmap Simplification for Two Variables

- The best way of selecting two groups of 1s form our simple Kmap is shown below.
- We see that both groups are powers of two and that the groups overlap.
- The next slide gives guidance for selecting Kmap groups.
3A.3 Kmap Simplification for Two Variables

The rules of Kmap simplification are:

• Groupings can contain only 1s; no 0s.
• Groups can be formed only at right angles; diagonal groups are not allowed.
• The number of 1s in a group must be a power of 2 – even if it contains a single 1.
• The groups must be made as large as possible.
• Groups can overlap and wrap around the sides of the Kmap.
3A.3 Kmap Simplification for Three Variables

- A Kmap for three variables is constructed as shown in the diagram below.
- We have placed each minterm in the cell that will hold its value.
  - Notice that the values for the $yz$ combination at the top of the matrix form a pattern that is not a normal binary sequence.

```
<table>
<thead>
<tr>
<th></th>
<th>00</th>
<th>01</th>
<th>11</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>$x'y'z'$</td>
<td>$x'y'z$</td>
<td>$xyz$</td>
<td>$x'y'z'$</td>
</tr>
<tr>
<td>1</td>
<td>$x'y'z'$</td>
<td>$xy'z$</td>
<td>$xyz$</td>
<td>$xyz'$</td>
</tr>
</tbody>
</table>
```
3A.3 Kmap Simplification for Three Variables

- Thus, the first row of the Kmap contains all minterms where $x$ has a value of zero.
- The first column contains all minterms where $y$ and $z$ both have a value of zero.
3A.3 Kmap Simplification for Three Variables

- Consider the function:
  \[ F(X, Y, Z) = X'Y'Z + X'YZ + XY'Z + XYZ \]
- Its Kmap is given below.
  - What is the largest group of 1s that is a power of 2?
3A.3 Kmap Simplification for Three Variables

- This grouping tells us that changes in the variables $x$ and $y$ have no influence upon the value of the function: They are irrelevant.

- This means that the function,

$$F(X, Y, Z) = X'Y'Z + X'YZ + XY'Z + XYZ$$

reduces to $F(x) = z$.

You could verify this reduction with identities or a truth table.
3A.3 Kmap Simplification for Three Variables

- Now for a more complicated Kmap. Consider the function:

$$F(X, Y, Z) = X'Y'Z' + X'YZ + X'YZ' + XY'Z' + XYZ'$$

- Its Kmap is shown below. There are (only) two groupings of 1s.
  - Can you find them?

![Kmap Diagram]

<table>
<thead>
<tr>
<th>X</th>
<th>YZ</th>
<th>00</th>
<th>01</th>
<th>11</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td></td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td></td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>
3A.3 Kmap Simplification for Three Variables

- In this Kmap, we see an example of a group that wraps around the sides of a Kmap.
- This group tells us that the values of $x$ and $y$ are not relevant to the term of the function that is encompassed by the group.
  - What does this tell us about this term of the function?

What about the green group in the top row?
3A.3 Kmap Simplification for Three Variables

• The green group in the top row tells us that only the value of $x$ is significant in that group.
• We see that it is complemented in that row, so the other term of the reduced function is $x'$. 
• Our reduced function is $F(X,Y,Z) = x' + z'$

Recall that we had six minterms in our original function!
3A.3 Kmap Simplification for Four Variables

- Our model can be extended to accommodate the 16 minterms that are produced by a four-input function.
- This is the format for a 16-minterm Kmap:

<table>
<thead>
<tr>
<th></th>
<th>00</th>
<th>01</th>
<th>11</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>wx</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>00</td>
<td>(w'x'y'z')</td>
<td>(w'x'y'z)</td>
<td>(w'xyz)</td>
<td>(w'x'y'z')</td>
</tr>
<tr>
<td>01</td>
<td>(w'x'yz')</td>
<td>(w'xy'z)</td>
<td>(w'xyz)</td>
<td>(w'xy'z')</td>
</tr>
<tr>
<td>11</td>
<td>(wx'y'z')</td>
<td>(wx'y'z)</td>
<td>(wxyz)</td>
<td>(wx'y'z')</td>
</tr>
<tr>
<td>10</td>
<td>(wx'yz')</td>
<td>(wx'y'z)</td>
<td>(wxyz)</td>
<td>(wxyz)</td>
</tr>
</tbody>
</table>
3A.3 Kmap Simplification for Four Variables

- We have populated the Kmap shown below with the nonzero minterms from the function:

\[ F(W, X, Y, Z) = W'X'Y'Z' + W'X'Y'Z + W'X'YZ' + W'XYZ' + WX'Y'Z' + WX'Y'Z + WX'YZ' \]

- Can you identify (only) three groups in this Kmap?

Recall that groups can overlap.
3A.3 Kmap Simplification for Four Variables

• Our three groups consist of:
  – A purple group entirely within the Kmap at the right.
  – A pink group that wraps the top and bottom.
  – A green group that spans the corners.

• Thus we have three terms in our final function:

\[
F(W, X, Y, Z) = W'Y' + X'Z' \\
+ W'YZ'
\]
It is possible to have a choice as to how to pick groups within a Kmap, while keeping the groups as large as possible.

The (different) functions that result from the groupings below are logically equivalent.
3A.6 Don’t Care Conditions

- Real circuits don’t always need to have an output defined for every possible input.
  - For example, some calculator displays consist of 7-segment LEDs. These LEDs can display $2^7 - 1$ patterns, but only ten of them are useful.
- If a circuit is designed so that a particular set of inputs can never happen, we call this set of inputs a *don’t care* condition.
- They are very helpful to us in Kmap circuit simplification.
3A.6 Don’t Care Conditions

- In a Kmap, a don’t care condition is identified by an X in the cell of the minterm(s) for the don’t care inputs, as shown below.
- In performing the simplification, we are free to include or ignore the X’s when creating our groups.
3A.6 Don’t Care Conditions

- In one grouping in the Kmap below, we have the function:

\[ F(W, X, Y, Z) = W'Y' + YZ \]
3A.6 Don’t Care Conditions

• A different grouping gives us the function:

\[ F(W, X, Y, Z) = W'Z + YZ \]
3A.6 Don’t Care Conditions

- The truth table of: \( F(W, X, Y, Z) = W' Y' + YZ \)
- differs from the truth table of: \( F(W, X, Y, Z) = W' Z + YZ \)
- However, the values for which they differ, are the inputs for which we have don’t care conditions.
3A Conclusion

- Kmaps provide an easy graphical method of simplifying Boolean expressions.
- A Kmap is a matrix consisting of the outputs of the minterms of a Boolean function.
- In this section, we have discussed 2- 3- and 4-input Kmaps. This method can be extended to any number of inputs through the use of multiple tables.
3A Conclusion

Recapping the rules of Kmap simplification:

- Groupings can contain only 1s; no 0s.
- Groups can be formed only at right angles; diagonal groups are not allowed.
- The number of 1s in a group must be a power of 2 – even if it contains a single 1.
- The groups must be made as large as possible.
- Groups can overlap and wrap around the sides of the Kmap.
- Use don’t care conditions when you can.