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

• To design a logic circuit, we start with the truth table.
• Then we find the boolean function from the truth table:

\[ F(x, y, z) = x\overline{z} + \overline{x}y\overline{z} + x\overline{y}z + xy\overline{z} + xyz \]

• Finally we simplify it using Boolean identities:
\[ F(x, y, z) = x\overline{z} + y \]

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 of cells 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 \textit{minterm} is a product term that contains all of the function’s variables exactly once, either complemented or not complemented.

For example, the 4 minterms for a function having 2 inputs \(x\) and \(y\) are: \(\overline{x}\overline{y}, \overline{x}y, x\overline{y},\) and \(xy\)

Consider the Boolean function, \(F(x, y) = xy + \overline{x}\overline{y}\)

- Its minterms are:

<table>
<thead>
<tr>
<th>Minterm</th>
<th>(X)</th>
<th>(Y)</th>
<th>(F)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(\overline{x}\overline{y})</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>(\overline{x}y)</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>(x\overline{y})</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>(xy)</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

Similarly, a function having three inputs, has 8 minterms that are shown in this diagram.

A Kmap has a cell for each minterm.
- This means that it has a cell for each row 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.
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

\[
F(x, y) = x + y = \bar{x}y + x\bar{y} + xy
\]

(Why?):

\[
\begin{array}{cc|c}
 x & y & x + y \\
 0 & 0 & 0 \\
 0 & 1 & 1 \\
 1 & 0 & 1 \\
 1 & 1 & 1 \\
\end{array}
\]

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?)
- The best way of selecting two groups of 1s from our simple Kmap is shown in the next slide.

Proof:

- Red group: \( x \bar{y} + x y = x(y + y) = x \)
- Green group: \( x' y + x y = (x' + x) y = y \)

\[
F(x, y) = x'y + xy + x'y = x + y
\]

3A.3 Kmap Simplification for Two Variables

- There are two groups of 1s. Both groups are powers of two and they overlap.
- The next slide gives detailed guidance for selecting Kmap groups.
- Once the groups are found, we write down the simplified term for each group. A variable in a minterm can be removed if it changes value inside the group:
- Red group: \( x \)
  - Green group: \( y \)

\[
\begin{array}{cc|c}
 x & y & 0 \\
 0 & 0 & 1 \\
 1 & 1 & 1 \\
\end{array}
\]

\[
\begin{array}{cc|c}
 x & y & 0 \\
 0 & 0 & 1 \\
 1 & 1 & 1 \\
\end{array}
\]
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 (=2^0).
- The groups must be made as large as possible.
- Groups can overlap and wrap around the sides of the Kmap.

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.

Consider the function:

\[
F(x, y, z) = \overline{x}y\bar{z} + xy\bar{z} + x\overline{y}z + xyz
\]

Its Kmap is given below.

What is the largest group of 1s that is a power of 2?
3A.4 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, 
  \[
  F(x, y, z) = \overline{x}y\overline{z} + \overline{x}yz + x\overline{y}z + xyz
  \]
  reduces to \( F(x) = z \).

You could verify this reduction with identities or a truth table.

3A.4 Kmap Simplification for Three Variables

- Now for a more complicated Kmap. Consider the function:
  \[
  F(x, y, z) = \overline{x}y\overline{z} + \overline{x}yz + x\overline{y}z + x\overline{y}z + xyz
  \]
- Its Kmap is shown below. There are (only) two groupings of 1s.
  - Can you find them?

3A.4 Kmap Simplification for Three Variables

- In this Kmap, we see an example of a group that wraps around the sides of a Kmap (envision the map as being drawn on a cylinder).
- 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?

3A.4 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 \( \overline{x} \).
- Our reduced function is: 
  \[
  F(x, y, z) = \overline{x} + \overline{z}
  \]

Recall that we had six minterms in our original function!
3A.5 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>WX</th>
<th>YZ</th>
<th>00</th>
<th>01</th>
<th>11</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>WXZ</td>
<td>WXYZ</td>
<td>WXYZ</td>
<td>WXYZ</td>
<td></td>
</tr>
<tr>
<td>01</td>
<td>WXYZ</td>
<td>WXYZ</td>
<td>WXYZ</td>
<td>WXYZ</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>WXYZ</td>
<td>WXYZ</td>
<td>WXYZ</td>
<td>WXYZ</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>WXYZ</td>
<td>WXYZ</td>
<td>WXYZ</td>
<td>WXYZ</td>
<td></td>
</tr>
</tbody>
</table>
```

3A.5 Kmap Simplification for Four Variables

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

\[
F(W, X, Y, Z) = \overline{WYZ} + \overline{WXYZ} + \overline{WYZ} + \overline{WXYZ} + \overline{WXYZ} + \overline{WXYZ}
\]

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

Recall that groups can overlap.

3A.5 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:

\[
F(W, X, Y, Z) = \overline{XY} + \overline{WXYZ} + \overline{WXYZ}
\]

3A.5 Kmap Simplification for Four Variables

• 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.

```
<table>
<thead>
<tr>
<th>WX</th>
<th>YZ</th>
<th>00</th>
<th>01</th>
<th>11</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>01</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
```
3A.6 Don’t Care Conditions

• Real circuits don’t always need to have an output defined for every possible input.
  – See the next slide for a 7-segment LED example. (Only 10 out of the 16 input combinations are defined.)
• 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:
  \[ F(W, X, Y, Z) = \overline{WX} + YZ \]
3A.6 Don’t Care Conditions

- A different grouping gives us the function:
  \[ F(W, X, Y, Z) = \overline{W}Z + YZ \]

The truth table of:

\[
\begin{array}{cccc}
  W & X & Y & Z \\
  00 & 01 & 11 & 10 \\
  \times & 1 & 1 & \times \\
  \times & 1 & 1 & \times \\
\end{array}
\]

is different from the truth table of:

\[
\begin{array}{cccc}
  W & X & Y & Z \\
  00 & 01 & 11 & 10 \\
  \times & 1 & 1 & \times \\
  \times & 1 & 1 & \times \\
\end{array}
\]

- 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.

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 (=2^0).
- 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.