S.2 Two and Three Variable Karnaugh Maps

A Karnaugh Map (K-Map) is a graphical representation of a Boolean expression which is convenient for simplifying expressions (and analyzing hazards).

Karnaugh maps help simplify expressions by removing redundant terms and literals. Its main advantage is that it’s systematic.

Ex: Consider the following three input truth table.

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>X</th>
<th>Y</th>
<th>Z</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

\[ X = A'B'C' + A'BC = A'B' (C' + C) = A'B' \]
\[ Y = A'B'C' + A'BC' = A'C' (B' + B) = A'C' \]
\[ Z = A'B'C' + AB'C = (A' + A) BC' = B'C' \]

K-map helps find terms like these and eliminates them.

Two Variable Karnaugh Map

Think of it as a 2D truth table...

Ex: \[ \begin{array}{ccc|c}
A & B & Z \\
0 & 0 & 1 \\
0 & 1 & 0 \\
1 & 0 & 1 \\
1 & 1 & 0 \\
\end{array} \]

\[ \begin{array}{cc|cc}
A'B' & AB' & 00 & 01 \\
1 & 0 \\
1 & 0 \\
\end{array} \]

\[ \begin{array}{cc}
A'B' & AB' \\
0 & 1 \\
1 & 0 \\
\end{array} \]

\[ Z = AB' + A'B' = B' \]

Alternatively,

\[ Z = B' \]

Grouping adjacent ones in the K-map leads to simplified expressions.

We want to cover as many 1's as possible to yield the simplest terms (fewest literals).
Three Variable Karnaugh Map

Think of the combinations if inputs in 3D "Boolean Space":

Now, unfold the cube to create the 3 variable K-map:

Each entry corresponds to a minterm \( ABC, A'B'C', \) etc.

Ex: \( Z(A, B, C) = \Sigma m(0, 2, 3, 7) \)

Group 1: \( A'B'C' + ABC = A'C' \)

Group 2: \( A'BC + ABC = BC \)

Drop the literal which changes value in the group, e.g. \( A \) in Group 2.

Note that we can fill in the K-map from the truth table, a minterm expansion, or directly from a Boolean expression.
Ex: Some examples of groupings

\[ Z = A + BC + B'A'C' \]

\[ Z = B' \]

\[ Z = B'C \]

Goal: cover more area per group to get simpler terms.

5.3 Four Variable Karnaugh Maps

Think of it as two 3-variable K-maps stitched together:

Ex:

\[ A'BC'D + ABC' + A'C + BC \]

\[ AB + B'C'D + A'C + BC \]

Larger groups, fewer literals
Ex: Four corners

Ex: Include "Don't Cares" to create larger groups.

Without "don't care": $A'Bc' + BC'D$

With "don't care": $BC'$

Deriving Maxterms (Product of Sums)

Maxterms $M(\cdot)$ correspond to input combinations which force the output to 0 in the truth table.

Ex: $AB \cdot CD \quad 00 \quad 01 \quad 11 \quad 10$

SOP: $B' + A'C$

POS: Find the minterms for $Z'$ (i.e., when $Z=0$) and convert using DeMorgan's Laws:

$Z' = BC' + AB$

$Z = (Z')' = (BC' + AB)' = (BC')' \cdot (AB)'$

$= (B' + C) \cdot (A' + B')$
5.4 Minimum Expressions Using Essential Prime Implicants

**Def:** Implicant of a function $F$ is any single 1 or combinable group of 1's in the K-map of $F$.

**Def:** Prime Implicant of a function $F$ is any implicant that cannot be combined with another implicant to eliminate a literal.

A minimum sum-of-products solution must consist of prime implicants. In general, not all prime implicants are in the minimum SOP solution.

**Def:** Essential Prime Implicant is a prime implicant which covers a minterm that is covered by no other prime implicant.

Essential P.I.'s can also be found by:

1) Choose a minterm and look at all adjacent 1's and X's

2) If they are all covered by a single prime implicant, that P.I. is essential.

**Ex:**

*Diagram showing a K-map with minterms labeled and prime implicants identified.*

Implicants: $A'B'$, $B'D'$, $BCD$, $ACD$

Prime Implicants: $A'B'$, $B'D'$, $BCD$, $ACD$

Essential Prime Implicants:

Check $m_2 \Rightarrow B'D'$ essential

Check $m_7 \Rightarrow BCD$ essential

Check $m_9 \Rightarrow AB'$ essential

Check $m_{15} \Rightarrow ACD$ not essential
5.5 Five Variable Karnaugh Maps

Think of it as two 4-variable K-maps laid on top of each other:

Ex:

Checks:

- \( AB'C'D' \): 5 variable K-map, 4 literals \( \Rightarrow \) group of 2
  
  Are all minterms in this group (10000, 10001) circled? Yes \( \checkmark \)

- \( AB'C'D'E' \): 5 literals \( \Rightarrow \) group of 1
  
- \( ACD' \): 3 literals \( \Rightarrow \) group of 4

\[ A'B'C'D' \rightarrow 1 \]