COE 202: Digital Logic Design
Combinational Logic
Part 3

Dr. Ahmad Almulhem
Email: ahmadsrm AT kfupm
Phone: 860-7554
Office: 22-324
Objectives

- Karnaugh Maps (K-Maps)
- Learn to minimize a function using K-Maps
  - 2-Variables
  - 3-Variables
  - 4-Variables
- Don’t care conditions
- Important Definitions
- 5-Variables K-Maps
Introduction

• Algebraic methods for minimization is limited:
  • No formal steps (id 10 first, then id 4, etc?), need experience.
  • No guarantee that a minimum is reached
  • Easy to make mistakes
• Karnaugh maps (k-maps) is an alternative convenient way for minimization:
  • A graphical technique
  • Introduced by Maurice Karnaugh in 1953
• K-maps for up to 4 variables are straightforward to build
• Building higher order K-maps (5 or 6 variable) are a bit more cumbersome
• Simplified expression produced by K-maps are in SOP or POS forms
Gray Codes (review)

• Only one bit changes with each number increment

• Build using recursive reflection

• To translate a binary value into the corresponding Gray code, each bit is inverted if the next higher bit of the input value is set to one.

Ahmad Almulhem, KFUPM 2009

src: wikipedia.org
Truth Table Adjacencies

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>F</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

These minterms are adjacent in a gray code sense – they differ by only one bit.

We can apply $XY + XY' = X$.

$$F = A'\overline{B} + AB = A'(B' + B) = A' \text{ (1)}$$

$$= A'$$

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>F</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</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>

Same idea:

$$F = A'B + AB = B$$

Keep common literal only!

Ahmad Almulhem, KFUPM 2009
### K-Map

<table>
<thead>
<tr>
<th></th>
<th></th>
<th>F</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</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>

A different way to draw a truth table!

Take advantage of adjacency

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

Keep common literal only!
Minimization with K-maps

1. Draw a K-map
2. Combine maximum number of 1’s following rules:
   1. Only adjacent squares can be combined
   2. All 1’s must be covered
   3. Covering rectangles must be of size 1, 2, 4, 8, … \(2^n\)
3. Check if all covering are really needed
4. Read off the SOP expression
2-variable K-map

Given a function with 2 variables: \( F(X,Y) \), the total number of minterms are equal to 4:

\[ m_0, m_1, m_2, m_3 \]

The size of the k-map is always equal to the total number of minterms.

Each entry of the k-map corresponds to one minterm for the function:

Row 0 represents: \( X'Y' \), \( X'Y \)
Row 1 represents: \( XY' \), \( XY \)

Ahmad Almulhem, KFUPM 2009
Example 1

For a given function $F(X, Y)$ with the following truth table, minimize it using k-maps

<table>
<thead>
<tr>
<th>X</th>
<th>Y</th>
<th>F</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>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

The final reduced expression is given by the common literals from the combination:

Therefore, since for the combination, $Y$ has different values $(0, 1)$, and $X$ has a fixed value of 1,

The reduced function is: $F(X, Y) = X$
Example 2

Q. Simplify the function $F(X,Y) = \sum m(1,2,3)$

Sol. This function has 2 variables, and three 1-squares (three minterms where function is 1)

$$F = m_1 + m_2 + m_3$$

Note: The 1-squares can be combined more than once

Minimized expression: $F = X + Y$
2 variable K-Maps (Adjacency)

In an $n$-variable $k$-map, each square is adjacent to exactly $n$ other squares.

Q: What if you have 1 in all squares?
### 3-variable K-maps

For 3-variable functions, the k-maps are larger and look different.

Total number of minterms that need to be accommodated in the k-map = 8

<table>
<thead>
<tr>
<th>BC</th>
<th>00</th>
<th>01</th>
<th>11</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>m₀</td>
<td>m₁</td>
<td>m₃</td>
<td>m₂</td>
</tr>
<tr>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>A</td>
<td>m₄</td>
<td>m₅</td>
<td>m₇</td>
<td>m₆</td>
</tr>
<tr>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

To maintain adjacency neighbors don’t have more than 1 different bit

Ahmad Almulhem, KFUPM 2009
3-variable K-maps

Note: You can only combine a power of 2 adjacent 1-squares. For e.g. 2, 4, 8, 16 squares. You cannot combine 3, 7 or 5 squares.

Minterms \( m_0 \), \( m_2 \), \( m_4 \), \( m_6 \) can be combined as \( m_0 \) and \( m_2 \) are adjacent to each other, \( m_4 \) and \( m_6 \) are adjacent to each other.

\( m_0 \) and \( m_4 \) are also adjacent to each other, \( m_2 \) and \( m_6 \) are also adjacent to each other.

Ahmad Almulhem, KFUPM 2009
Example 1

Simplify $F = \sum m(1, 3, 4, 6)$ using K-map
Example 1

Simplify \( F = \sum m(1, 3, 4, 6) \) using K-map

\[ F = A'C + AC' \]
Example 2

Simplify \( F = \sum m(0,1, 2, 4, 6) \) using K-map
Example 2

Simplify \( F = \sum m(0,1, 2, 4, 6) \) using K-map

\[ F = A' B' + C' \]
3 variable K-Maps (Adjacency)

A 3-variable map has 12 possible groups of 2 minterms

They become product terms with 2 literals

Ahmad Almulhem, KFUPM 2009
A 3-variable map has 6 possible groups of 4 minterms

They become product terms with 1 literals
4-variable K-maps

A 4-variable function will consist of 16 minterms and therefore a size 16 k-map is needed.

Each square is adjacent to 4 other squares.
A square by itself will represent a minterm with 4 literals.
Combining 2 squares will generate a 3-literal output.
Combining 4 squares will generate a 2-literal output.
Combining 8 squares will generate a 1-literal output.

Ahmad Almulhem, KFUPM 2009
# 4-variable K-maps (Adjacency)

Right column and left column are adjacent; can be combined

Top row and bottom column are adjacent; can be combined

Many possible $2, 4, 8$ groupings

<table>
<thead>
<tr>
<th>AB</th>
<th>CD</th>
<th>00</th>
<th>01</th>
<th>11</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td></td>
<td>$m_0$</td>
<td>$m_1$</td>
<td>$m_3$</td>
<td>$m_2$</td>
</tr>
<tr>
<td>01</td>
<td></td>
<td>$m_4$</td>
<td>$m_5$</td>
<td>$m_7$</td>
<td>$m_6$</td>
</tr>
<tr>
<td>11</td>
<td></td>
<td>$m_{12}$</td>
<td>$m_{13}$</td>
<td>$m_{15}$</td>
<td>$m_{14}$</td>
</tr>
<tr>
<td>10</td>
<td></td>
<td>$m_8$</td>
<td>$m_9$</td>
<td>$m_{11}$</td>
<td>$m_{10}$</td>
</tr>
</tbody>
</table>

Note: You can only combine a power of 2 adjacent 1-squares. For e.g. 2, 4, 8, 16 squares. You cannot combine 3, 7 or 5 squares.
Example

Minimize the function $F(A,B,C,D) = \sum m(1, 3, 5, 6, 7, 8, 9, 11, 14, 15)$

$$F = CD + A'D + BC + AB'C'$$
Example

\[ F(A, B, C, D) = \Sigma m(0, 1, 2, 5, 8, 9, 10) \]
Example

\[ F(A,B,C,D) = \Sigma m(0,1,2,5,8,9,10) \]

Solution:
\[ F = B' D' + B' C' + A' C' D \]

Ahmad Almulhem, KFUPM 2009
Example (POS)

\[ F(A,B,C,D) = \Sigma m(0,1,2,5,8,9,10) \]

Write \( F \) in the simplified product of sums (POS)

Two methods?
You already know one!
Example (POS)

F(A,B,C,D) = \Sigma m(0,1,2,5,8,9,10)

Write \( F \) in the simplified product of sums (POS)

Method 2:
Follow same rule as before but for the ZEROs

\[ F' = AB + CD + BD' \]

Therefore,
\[ F'' = F = (A' + B')(C' + D')(B' + D) \]

Ahmad Almulhem, KFUPM 2009
Don’t Cares

- In some cases, the output of the function (1 or 0) is not specified for certain input combinations either because
  - The input combination never occurs (Example BCD codes), or
  - We don’t care about the output of this particular combination
- Such functions are called incompletely specified functions
- Unspecified minterms for these functions are called don’t cares
- While minimizing a k-map with don’t care minterms, their values can be selected to be either 1 or 0 depending on what is needed for achieving a minimized output.
Example

\[ F = \sum m(1, 3, 7) + \sum d(0, 5) \]

Circle the x’s that help get bigger groups of 1’s (or 0’s if POS).

Don’t circle the x’s that don’t help.
Example

$F = \sum m(1, 3, 7) + \sum d(0, 5)$

Circle the x’s that help get bigger groups of 1’s (or 0’s if POS).

Don’t circle the x’s that don’t help.

$F = C$
Example 2

\[ F(A, B, C, D) = \sum m(1, 3, 7, 11, 15) + \sum d(0, 2, 5) \]

- Two possible solutions!
- Both acceptable.
- All 1’s covered
Definitions

• An **implicant** is a product term of a function
  • Any group of 1’s in a K-Map

• A **prime implicant** is a product term obtained by combining the maximum possible number of adjacent 1’s in a k-map
  • Biggest groups of 1’s
  • Not all prime implicants are needed!

• If a minterm is covered by exactly one prime implicant then this prime implicant is called an **essential prime implicant**
Example

Consider \( F(X,Y,Z) = \Sigma m(1,3,4,5,6) \)

List all implicants, prime implicants and essential prime implicants

Solution:

Implicants: \( XY'Z' \), \( XZ' \), \( XY' \), \( XY'Z \), \( X'Y'Z \), \( Y'Z \), …

P.Is: \( XY' \), \( XZ' \), \( Y'Z \), \( X'Z \)

EPIs: \( X'Z \), \( XZ' \)

The simplest expression is NOT unique!
Finding minimum SOP

1. Find each essential prime implicant and include it in the solution.

2. If any minterms are not yet covered, find minimum number of prime implicants to cover them (minimize overlap).
Example 2

Simplify $F(A, B, C, D) = \sum m(0, 1, 2, 4, 5, 10, 11, 13, 15)$

Note:

- Only $A'C'$ is E.P.I
- For the remaining minterms:
  - Choose 1 and 2 (minimize overlap)
  - For $m_2$, choose either $A'B'D'$ or $B'CD'$

$F = A'C' + ABD + AB'C + A'B'D'$

Src: Mano’s Textbook
5-variable K-maps

<table>
<thead>
<tr>
<th>DE</th>
<th>A=0</th>
<th>A=1</th>
</tr>
</thead>
<tbody>
<tr>
<td>BC</td>
<td>00</td>
<td>01</td>
</tr>
<tr>
<td>00</td>
<td>(m_0)</td>
<td>(m_1)</td>
</tr>
<tr>
<td>01</td>
<td>(m_4)</td>
<td>(m_5)</td>
</tr>
<tr>
<td>11</td>
<td>(m_{12})</td>
<td>(m_{13})</td>
</tr>
<tr>
<td>10</td>
<td>(m_8)</td>
<td>(m_9)</td>
</tr>
</tbody>
</table>

- 32 minterms require 32 squares in the k-map
- Minterms 0-15 belong to the squares with variable A=0, and minterms 16-32 belong to the squares with variable A=1
- Each square in A' is also adjacent to a square in A (one is above the other)
- Minterm 4 is adjacent to 20, and minterm 15 is to 31

Ahmad Almulhem, KFUPM 2009
Conclusion

• A K-Map is simply a folded truth table, where physical adjacency implies logical adjacency

• K-Maps are most commonly used hand method for logic minimization.