## 5.2 Two and Three Variable Karnaugh Maps

A <u>Karnaugh Map (K-Map)</u> 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 <u>systematic</u>.

<u>Ex:</u> Consider the following three input truth table.

| A | B | C | X | Y | Z |
|---|---|---|---|---|---|
| 0 | 0 | 0 | 1 | 1 | 1 |
| 0 | 0 | 1 | 1 | 0 | 0 |
| 0 | 1 | 0 | 0 | 1 | 0 |
| 0 | 1 | 1 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 | 1 |
| 1 | 0 | 1 | 0 | 0 | 0 |
| 1 | 1 | 0 | 0 | 0 | 0 |
| 1 | 1 | 1 | 0 | 0 | 0 |

$$X = A'B'C' + A'B'C = A'B'(C'+C) = A'B'$$

$$Y = A'B'C' + A'BC' = A'C'(B'+B) = A'C'$$

$$Z = A'B'C' + AB'C' = \underbrace{(A'+A)}B'C' = B'C'$$

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

### <u>Two Variable Karnaugh Map</u>

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

<u>Ex:</u>

| A | B | Z |
|---|---|---|
| 0 | 0 | 1 |
| 0 | 1 | 0 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

→

| | | |
|---|---|---|
| A'B' 00 | 1 | |
| A'B 01 | 0 | |
| AB' 10 | 1 | |
| AB 11 | 0 | |

→

| B \ A | 0 | 1 |
|---|---|---|
| 0 | 1 (A'B') | 1 (AB') |
| 1 | 0 | 0 |

$$Z = AB' + A'B' = B'$$

Alternatively,

| A \ B | 0 | 1 |
|---|---|---|
| 0 | 1 | 0 |
| 1 | 1 | 0 |

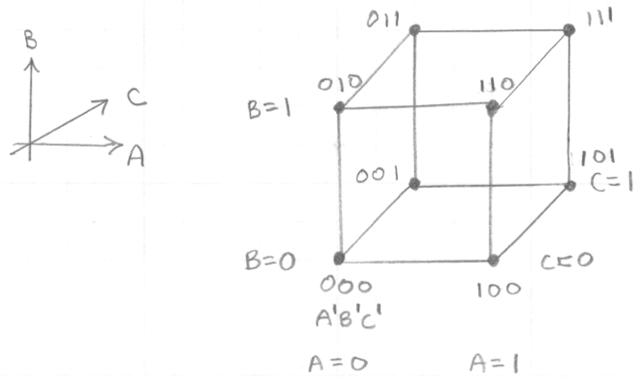
$$Z = B'$$

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:

| AC \ B | 0 | 1 |
|---|---|---|
| 00 | 000 $m_0$ | 010 $m_2$ |
| 01 | 001 $m_1$ | 011 $m_3$ |
| 11 | 101 $m_5$ | 111 $m_7$ |
| 10 | 100 $m_4$ | 110 $m_6$ |

Note: AC binary values are <u>not</u> in binary order → only 1 bit changes between each entry (Grey Code).

Note: Edges are implicitly connected (wrapped around). For example, 000 connects to 100, as in the 3D figure. This helps group larger numbers of 1's when simplifying.

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

<u>Ex:</u> $Z(A,B,C) = \sum m(0,2,3,7)$

| AB \ C | 0 | 1 |
|---|---|---|
| 00 | 1 | |
| 01 | 1 | 1 |
| 11 | | 1 |
| 10 | | |

Group 1

Consensus term (redundant)

Group 2

Group 1: $A'B'C' + A'BC' = A'C'$

Group 2: $A'BC + ABC = BC$

$\Big\} \; Z = A'C' + BC$

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

| AB \ C | 0 | 1 |
|---|---|---|
| 00 | 1 | |
| 01 | | 1 |
| 11 | 1 | 1 |
| 10 | 1 | 1 |

A'B'C' BC A

$Z = A + BC + B'A'C'$

| AB \ C | 0 | 1 |
|---|---|---|
| 00 | 1 | 1 |
| 01 | | |
| 11 | | |
| 10 | 1 | 1 |

B'

$Z = B'$

| AB \ C | 0 | 1 |
|---|---|---|
| 00 | | 1 |
| 01 | | |
| 11 | | |
| 10 | | 1 |

B'C

$Z = B'C$

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:

D=0

| AB \ C | 0 | 1 |
|---|---|---|
| 00 | | |
| 01 | | |
| 11 | | |
| 10 | | |

+

D=1

| AB \ C | 0 | 1 |
|---|---|---|
| 00 | | |
| 01 | | |
| 11 | | |
| 10 | | |

=

| AB \ CD | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | | | | |
| 01 | | | | |
| 11 | | | | |
| 10 | | | | |

← Grey code on top as well

Ex:

| AB \ CD | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | | | 1 | 1 |
| 01 | | 1 | 1 | 1 |
| 11 | 1 | 1 | 1 | 1 |
| 10 | | | | |

A'C A'BC'D ABC' BC

$A'BC'D + ABC' + A'C + BC$

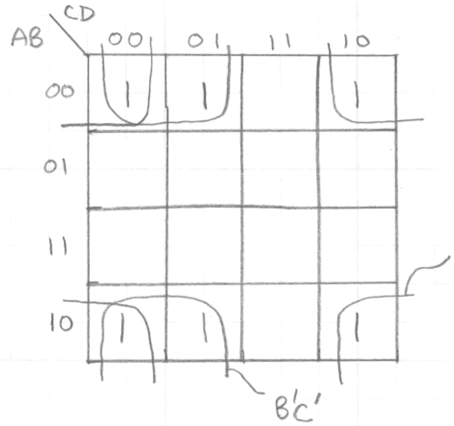
| AB \ CD | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | | | 1 | 1 |
| 01 | | 1 | 1 | 1 |
| 11 | 1 | 1 | 1 | 1 |
| 10 | | | | |

A'C BC'D AB BC

$AB + BC'D + A'C + BC$

↑ Larger groups, fewer literals

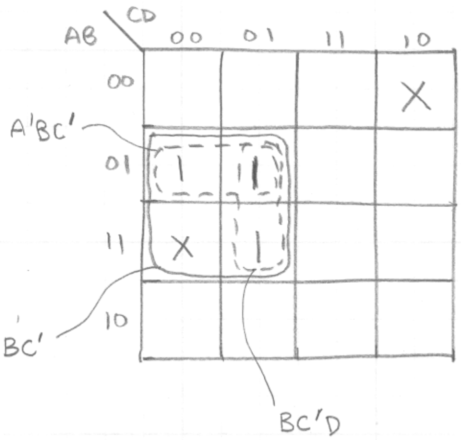
Ex: Four corners




Four corners = $B'D'$

$B'C' + B'D'$

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



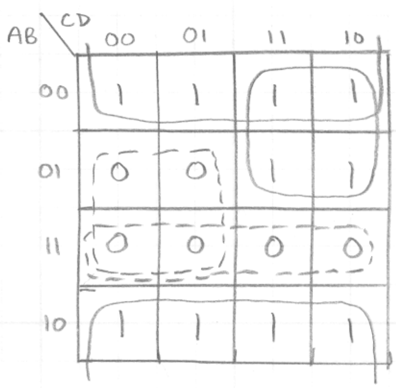

Without "don't care": $A'BC' + BC'D$

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

## Deriving Maxterms (Product of Sums)

Maxterms M( ) correspond to input combinations which force the output to Ø in the truth table.

Ex:




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.

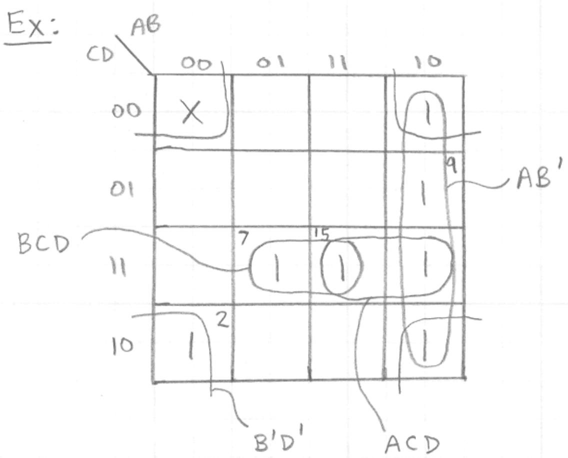
Def: Essential Prime Implicant is a prime implicant which covers a minterm that is covered by no other prime implicant.

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:




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

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

Essential Prime Implicants :

Check $m_2 \Rightarrow B'D'$ essential

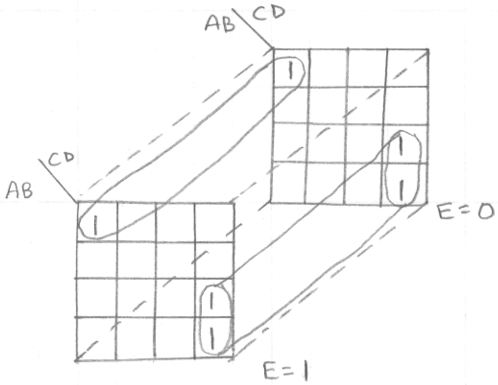
Check $m_7 \Rightarrow BCD$ essential

Check $m_9 \Rightarrow AB'$ essential

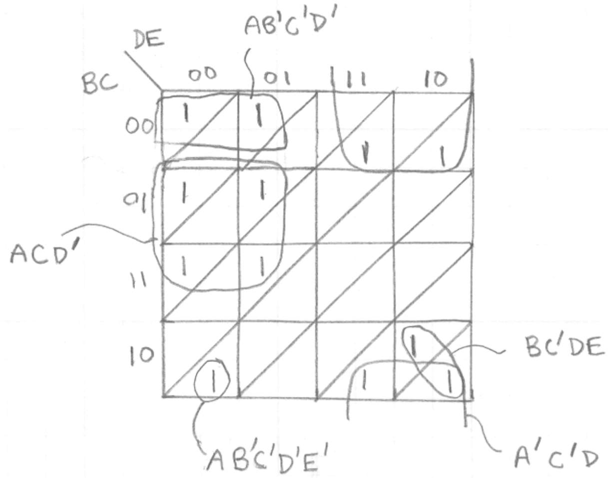
Check $m_{15} \Rightarrow ACD$ not essential

## 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 ✓

$AB'C'D'E'$ : 5 literals $\Rightarrow$ group of 1 ✓

$ACD'$ : 3 literals $\Rightarrow$ group of 4 ✓