Algebraic Simplification of Logic Circuit

- Use equivalence relations to reduce the number of operands and operators

- $x \lor (x \land y) = x$
- $x \land (x \lor y) = x$
- $x \lor (\neg x \land y) = x \lor y$
- $x \land (\neg x \lor y) = x \land y$
- $(x \lor y) \land (\neg x \lor z) \land (y \lor z) = (x \lor y) \land (\neg x \lor z)$

Properties of Boolean Algebra

- Associative Law
  - $(x \lor y) \lor z = x \lor (y \lor z)$
  - $(x \land y) \land z = x \land (y \land z)$
- Commutative Law
  - $x \lor y = y \lor x$
  - $x \land y = y \land x$
- Distributive Law
  - $x \lor (y \land z) = (x \lor y) \land (x \lor z)$
  - $x \land (y \lor z) = (x \land y) \lor (x \land z)$

De Morgan’s law

$$\neg(x \vee y) = \neg x \wedge \neg y$$

$$\neg(x \wedge y) = \neg x \vee \neg y$$

Atomic Propositions		Left-hand sides		Right-hand side		
x	y	x⋁y	¬(x⋁y)	¬x	¬y	¬x⋀¬y
0	0					
0	1					
1	0					
1	1					

Identity and Complement

- Identity elements
  - $x \vee 1 = 1$
  - $x \wedge 0 = 0$

- Complements
  - $x \vee x = x$
  - $x \wedge \neg x = 0$

Exercise

1: $(x \vee y) \vee z = x \vee (y \vee z)$	7: $x \vee x = x$
2: $x \vee y = y \vee x$	8: $x \vee \neg x = 1$
3: $x \wedge (y \vee z) = (x \wedge y) \vee (x \wedge z)$	9: $x \vee (x \wedge y) = x$
4: $\neg(x \vee y) = \neg x \wedge \neg y$	10: $x \wedge (x \vee y) = x$
5: $x \vee 1 = 1$	11: $x \vee (\neg x \wedge y) = x \vee y$
6: $x \wedge 1 = x$	12: $x \wedge (\neg x \vee y) = x \wedge y$

$f(x, y, z) = x'yz + xyz' + xyz = (\neg x \wedge y \wedge z) \vee (x \wedge y \wedge \neg z) \vee (x \wedge y \wedge z)$

$x'yz + xyz' + xyz = (x + x')yz + xyz'$ – (3)

$= yz + xyz'$ –(8), (6)

$= yz + xyz + xyz'$ –(9)

$= yz + xy(z + z')$ –(3)

$= yz + xy$ –(8),(6)

The Karnaugh Map: Practical Minimization Method

Minimization through algebraic transformations

The Karnaugh map

Intuition

A	*B*	*C*	*F*
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

A’BC + A’BC’ = A’B

A	*C*	*B*	*F*
0	0	0	0
~~0~~	~~0~~	~~1~~	~~1~~
0	1	0	0
~~0~~	~~1~~	~~1~~	~~1~~
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

ACB’ + ACB = AC

Karnaugh-Map (K-Map): Overview

K-map is a graphical method to find SOPs for a given truth table

Good point: help a human find a minimum SOP

- Highlight changes to apply the adjacency rule to minterms to reduce them into a smaller SOP

Limitation: only applicable for a truth table with 2 to 6 input variables, no guarantee to indicate minimal SOP.

K-Map: Example

A	B	C	F
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

B

C \ AB	$A'B'$ 00	$A'B$ 01	AB 11	AB' 10
C' **0**		1	1	
C **1**		1	1	

1

F = B

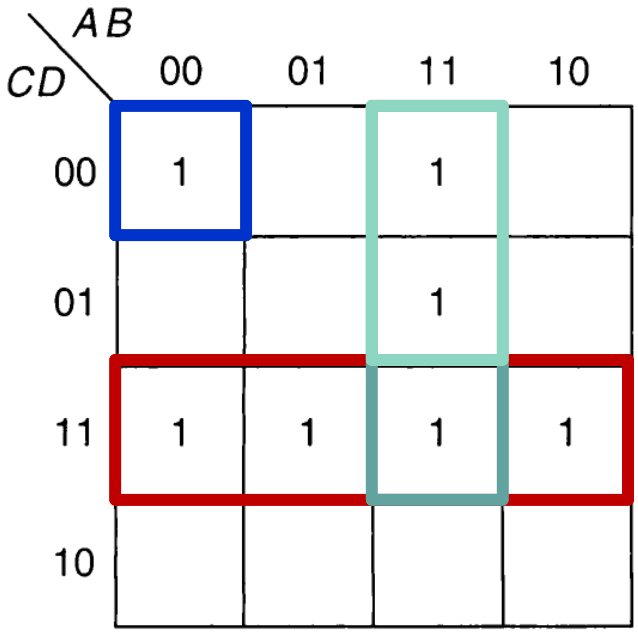
Translation (1/2)

An implicant is a rectangular of 1, 2, …, 2^n squares in K-map where all containing values are 1.

An implicant corresponds to a product term whose value is 1 only if the output value is 1.

- An implicant is the sum of all its covering minterms (squares)

The function of an output variable is the sum of the product terms for all implicants



A’B’C’D’ +
ABC’ +
CD

Translation (2/2)

A **minimum SOP** can be found by obtaining **a minimum set of implicants**

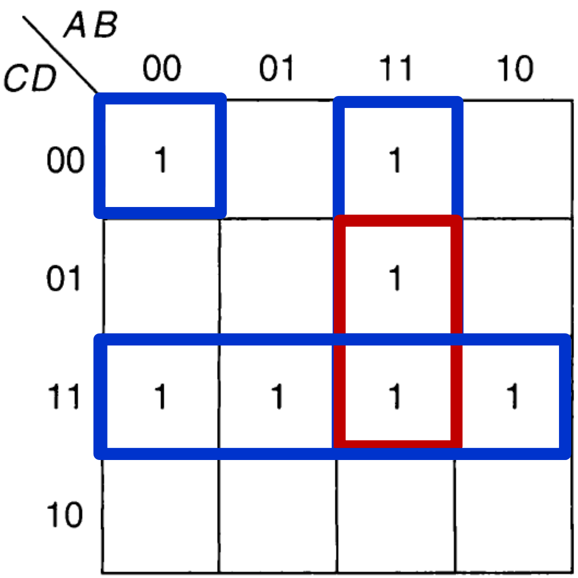
- less implicants, less terms
- wider implicants, less literals

A **prime implicant** is an implicant that is not fully contained in any other implicant

- If any literal is removed from the product term of an implicant, the product term is not fully filled with 1.

An **essential prime implicant** is a prime implicant that includes at least one 1 not included in any other prime implicants

- must appears in a minimum SOP



F = A'B'C'D' + CD + ABC'

Another Example

Minimize **a’b’c’ + a’bc’ + a’bc + ab’c’**

= a’c’b’ + a’c’b + a’bc’+ a’bc + ab’c’

= a’c’ + a’b + ab’c’

= a’b + c’(a’+ab’)

= a’b + c’(a’+b’)

= a’b + c’a’ + c’b’

= ba’ + b’c’ + a’c’

= ba’ + b’c’