CMPE 212
Principles of Digital Design

Lecture 10

Reduction of Combinational Logic

February 24, 2016

www.csee.umbc.edu/~younis/CMPE212/CMPE212.htm

Lecture's Overview

- ***Previous Lecture:***
  - ➔ Logic Synthesis
    (Definition, Procedure)
  - ➔ Gate Type Conversion
    (Role of DeMorgan Theorem, Bubble matching)
  - ➔ Popular Cell Library
    (AND-OR, AND-OR-INVERT, OR-AND, OR_AND_INVERT)
  - ➔ Technology Mapping
    (Splitting forest to trees, Fan-in constraints, Factoring)
- ***This Lecture***
  - ➔ Simplification of Switching Functions using K-map

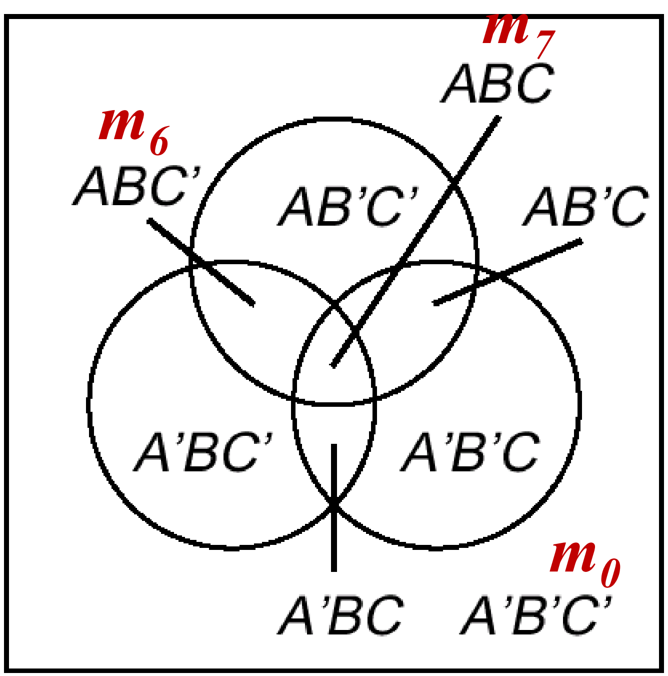
Design Optimization

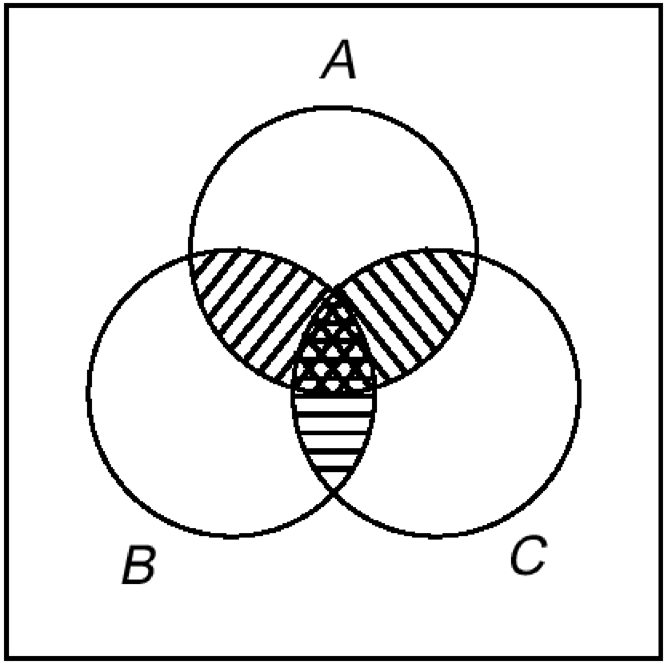
- Quality of combinational circuit design is measured using following metrics:
  - *Gate counts*: fewer gates require smaller area and cost less
  - *Propagation delay*: time for the output to become available after applying input. This time depends on transistor-level gate implementation
  - *Gate fan-in*: large gate fan-in can lead to increased gate counts and propagation delay (by using multi-level of gates)
  - *Gate fan-out*: large gate fan-out may mandate logic replication
- In many cases the canonical sum-of-products or product-of-sums forms are not minimal in terms of their number and size
- Since a smaller Boolean equation translates to a lower gate input count in the target circuit, reduction of the equation is an important consideration when circuit complexity is an issue
- Three methods for reducing Boolean equations are considered:
  - Algebraic reduction
  - Karnaugh map (K-map) reduction
  - Tabular reduction (Quine-McCluskey)

Karnaugh Maps

- Karnaugh maps (K-map) is a graphical technique to visualize minterms along with variables that are common to them
- Variables common to multiple minterms are candidates for elimination
- The basis of K-maps is the Venn diagram representation for visualizing concepts in set theory
- Each distinct region in the "Universe" represents a minterm
- *Example:* Majority Function

F = BC + AC + AB



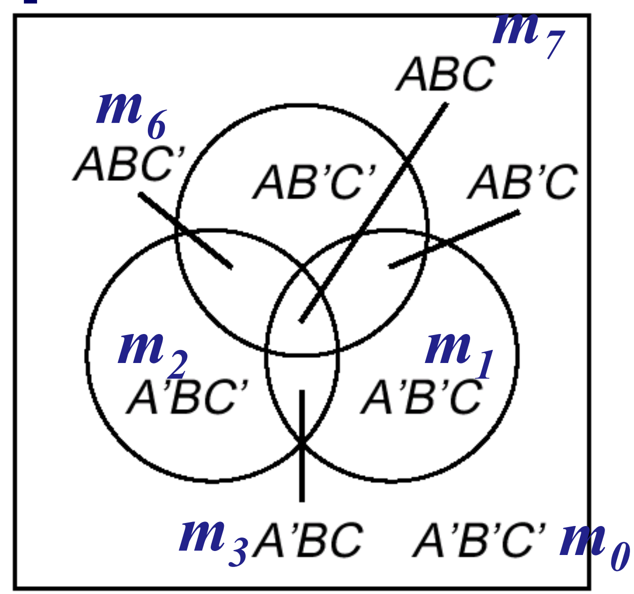


Forming K-Map

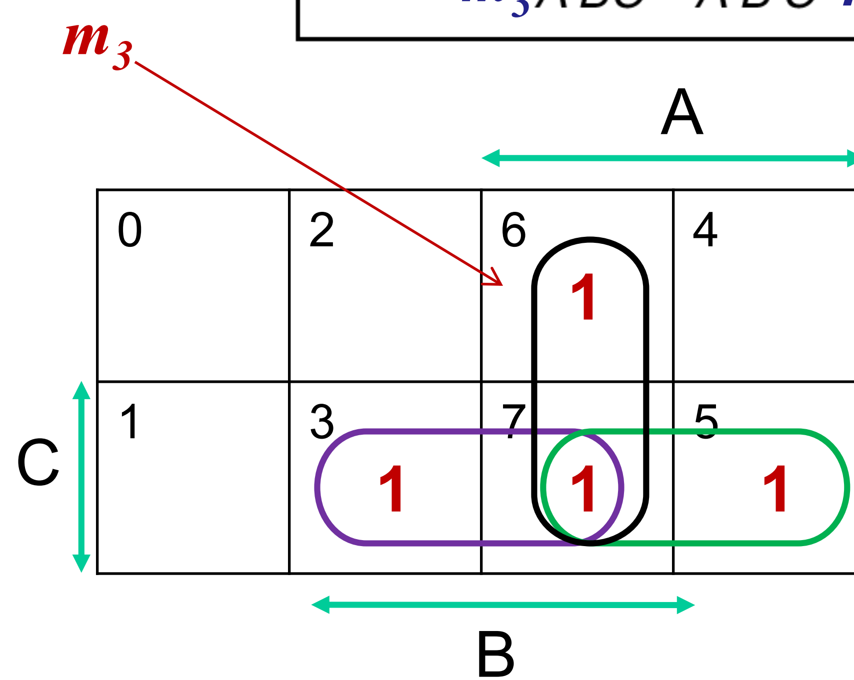
Minterm Index	A	B	C	F
0	0	0	0	0
1	0	0	1	0
2	0	1	0	0
3	0	1	1	1
4	1	0	0	0
5	1	0	1	1
6	1	1	0	1
7	1	1	1	1

Minterm adjacency matches the K-map

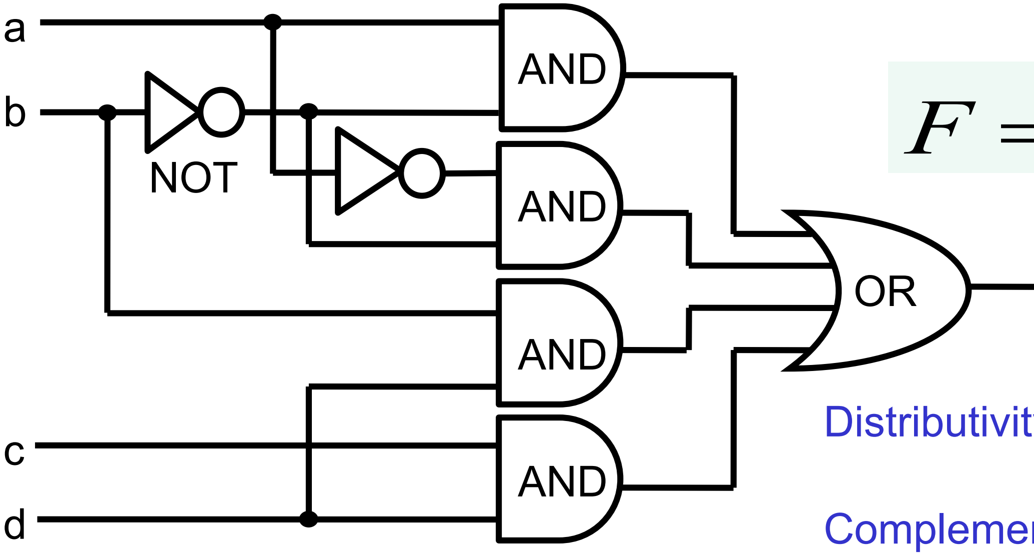
F = BC + AB + AC



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

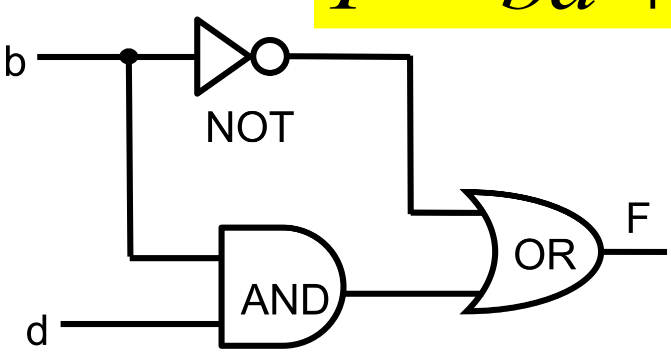


Understanding Minimization



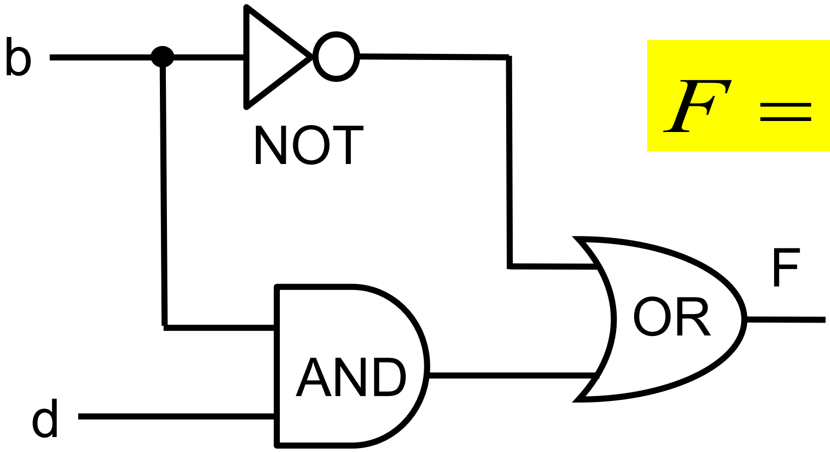
$$F = a\bar{b} + \bar{a}\bar{b} + bd + cd$$

$$F = bd + \bar{b}$$

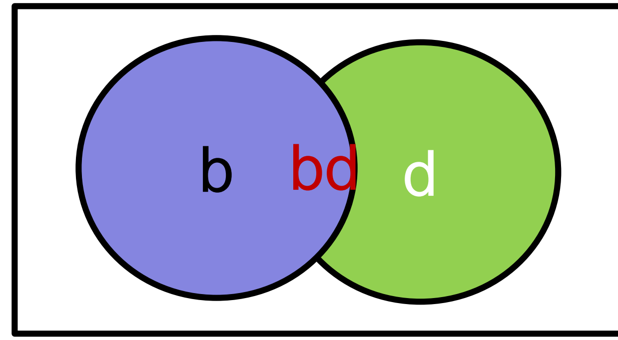


Distributivity	$F = a\bar{b} + \bar{a}\bar{b} + bd + cd$
Complementation	$= \bar{b}(a + \bar{a}) + bd + cd$
Identity	$= \bar{b}1 + bd + cd$
Complementation	$= \bar{b}(c + \bar{c}) + bd + cd$
Distribitivity	$= bd + \bar{b}c + cd + \bar{b}\bar{c}$
Consensus theorem	$= bd + \bar{b}c + \bar{b}\bar{c}$
Distributivity	$= bd + \bar{b}(c + \bar{c})$
Complement, identity	$= bd + \bar{b}$

Understanding Minimization

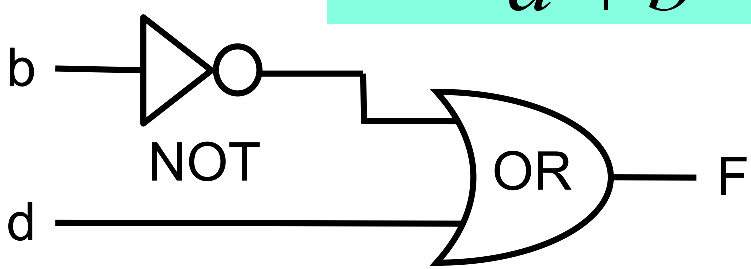


$$F = bd + \bar{b}$$



Absorption theorem

$$F = bd + \bar{b}$$
$$= d + \bar{b}$$



Logic minimization means:

- Minimize gates, i.e., # products in SOP or # sums in POS
- Minimize literals (reduce gate inputs)

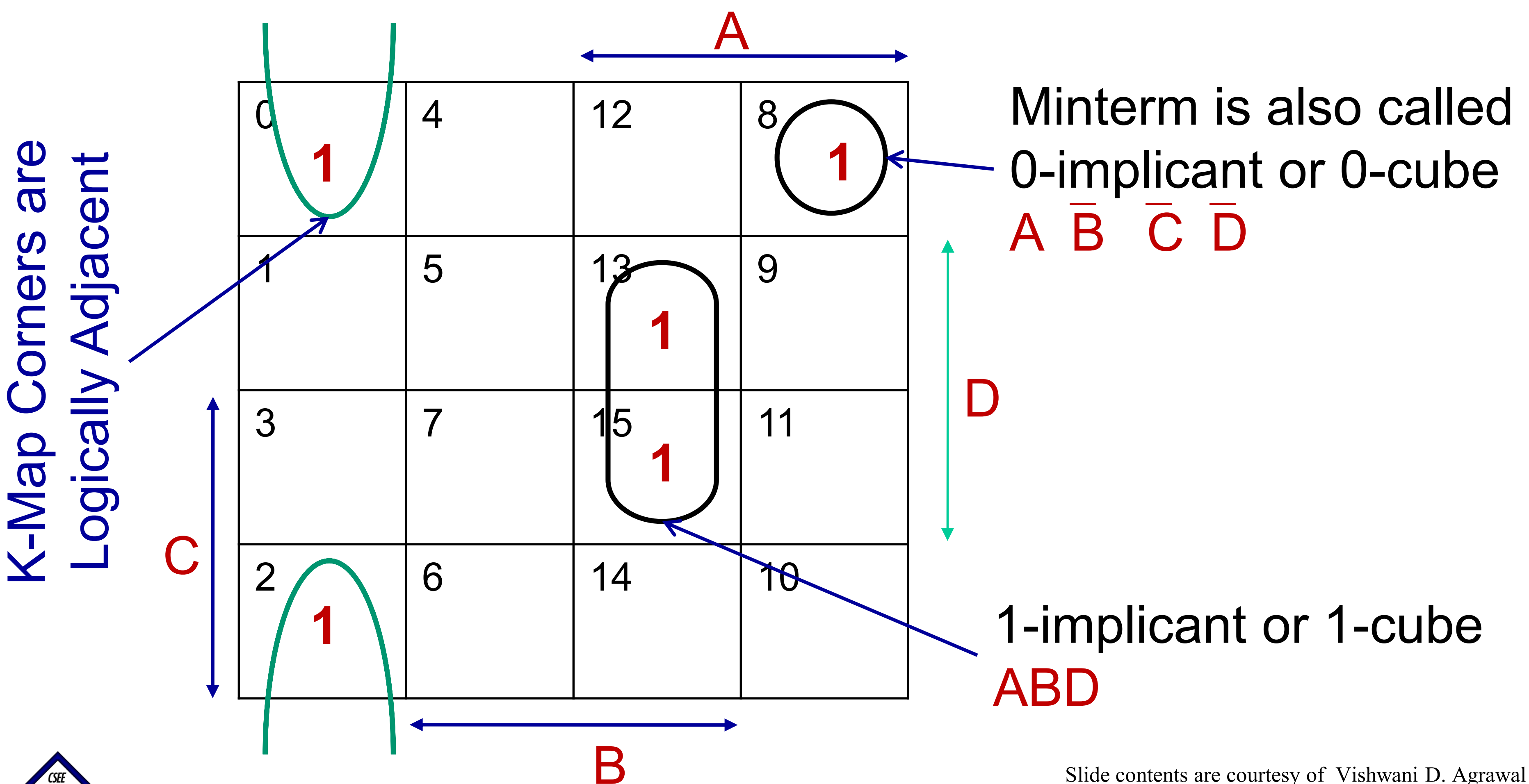
Product, Implicant or Cube

- Any set of literals ANDed together
- Minterm is a special case where all variables are present (largest product) and often called 0-implicant or 0-cube

Slide contents are courtesy of Vishwani D. Agrawal

Cubes (Implicants) of 4 Variables

- A 1-implicant (1-cube) is a product with one variable eliminated:
  - Obtained by combining two adjacent 0-cubes
  - $ABCD + ABC\bar{D} = ABC(D + \bar{D}) = ABC$

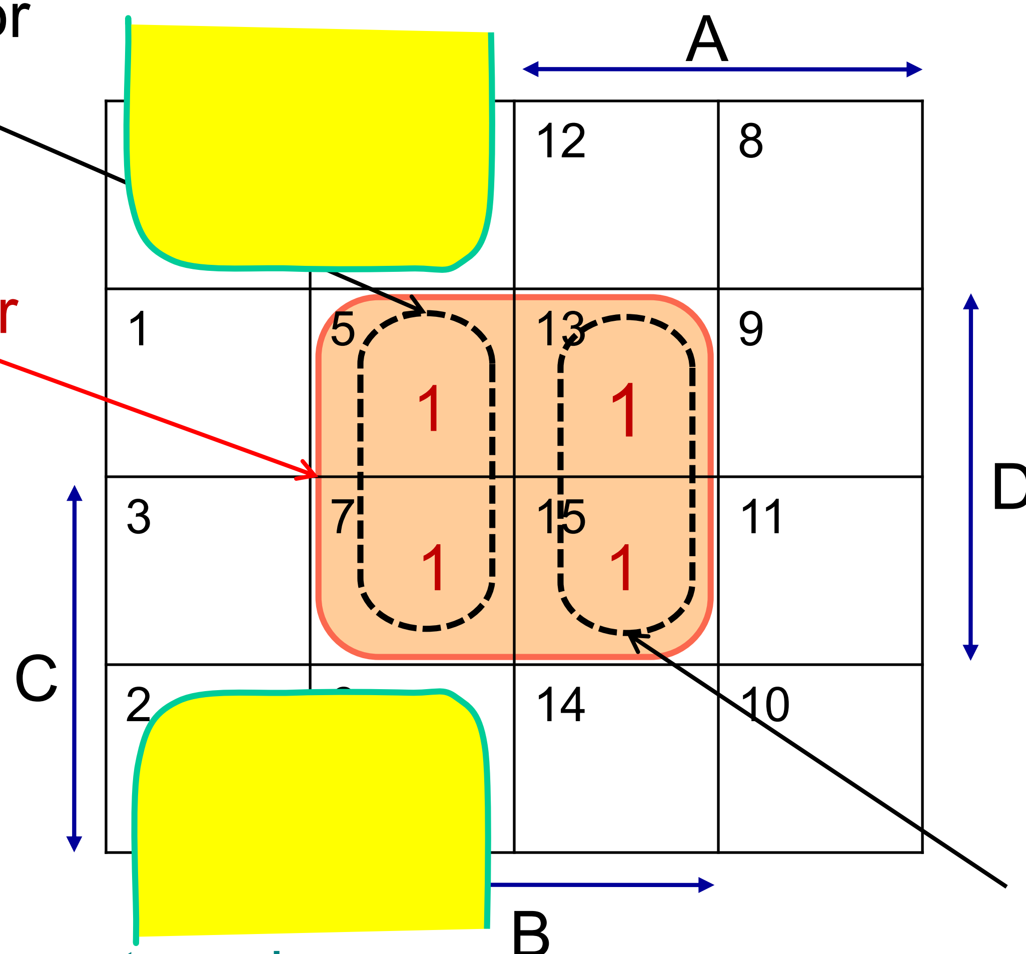


Slide contents are courtesy of Vishwani D. Agrawal

Growing Cubes, Reducing Products

1-implicant or
1-cube
$\bar{A}B\bar{C}D$

2-implicant or
2-cube
BD

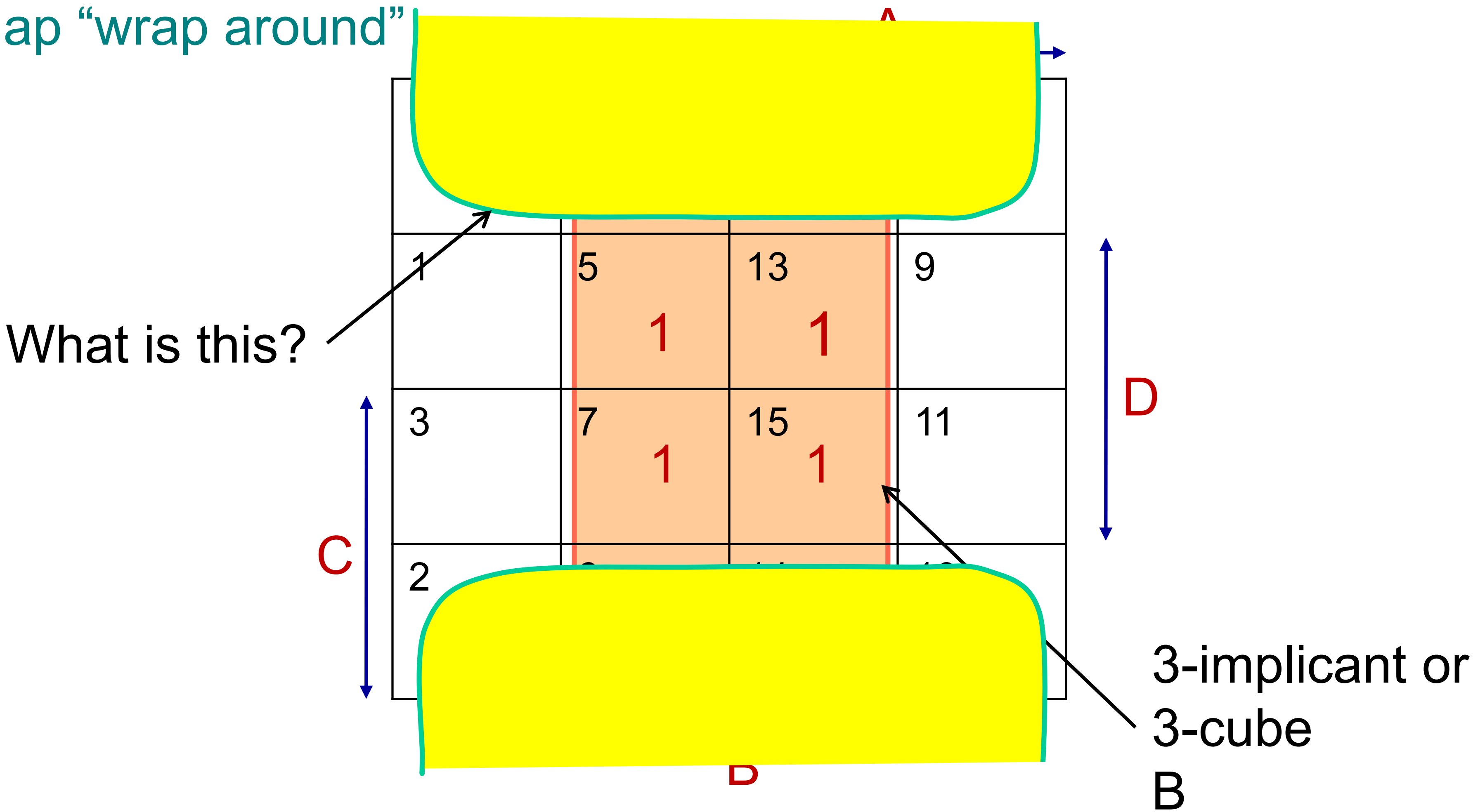


1-implicant or
1-cube
ABD

Cells on the outer edge
of the map "wrap around"

Largest Cubes or Smallest Products

Cells on the outer edge of the map "wrap around"



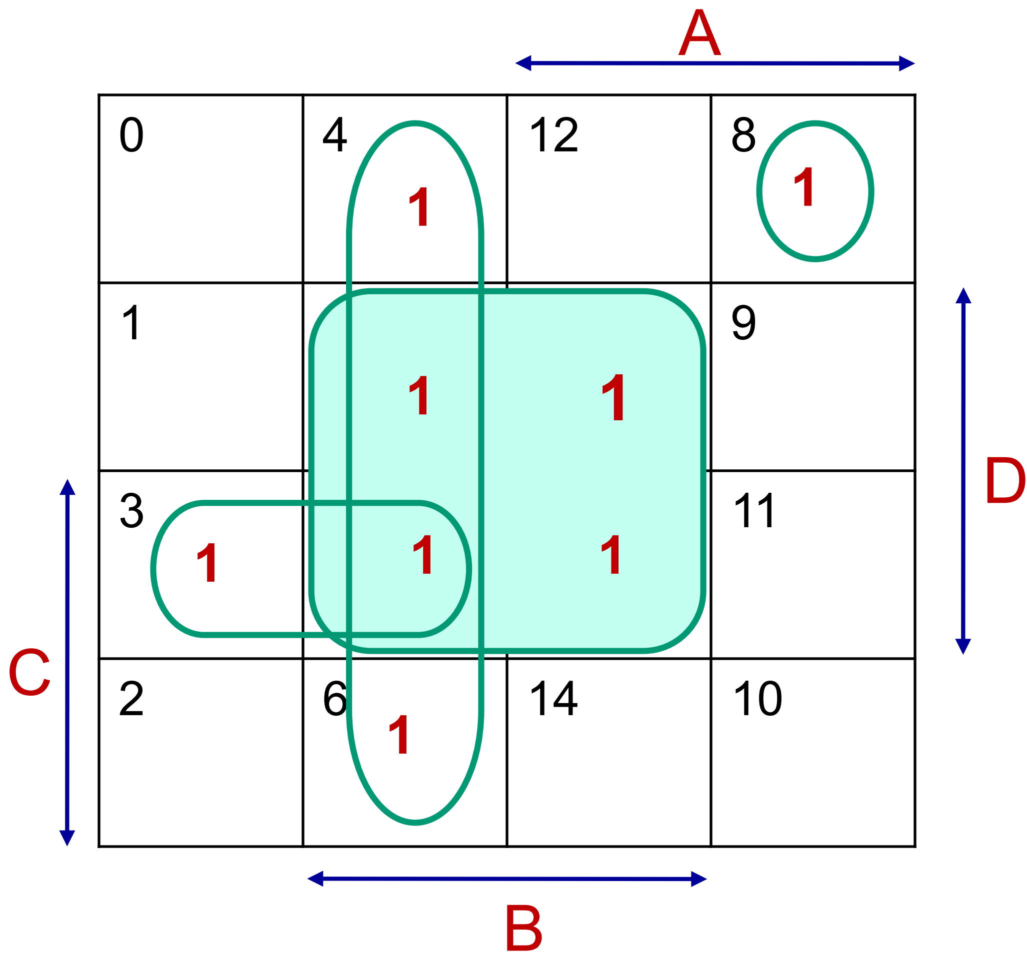
Implicants of a Function

- Minterms, products, cubes that imply the function

Implication and Covering

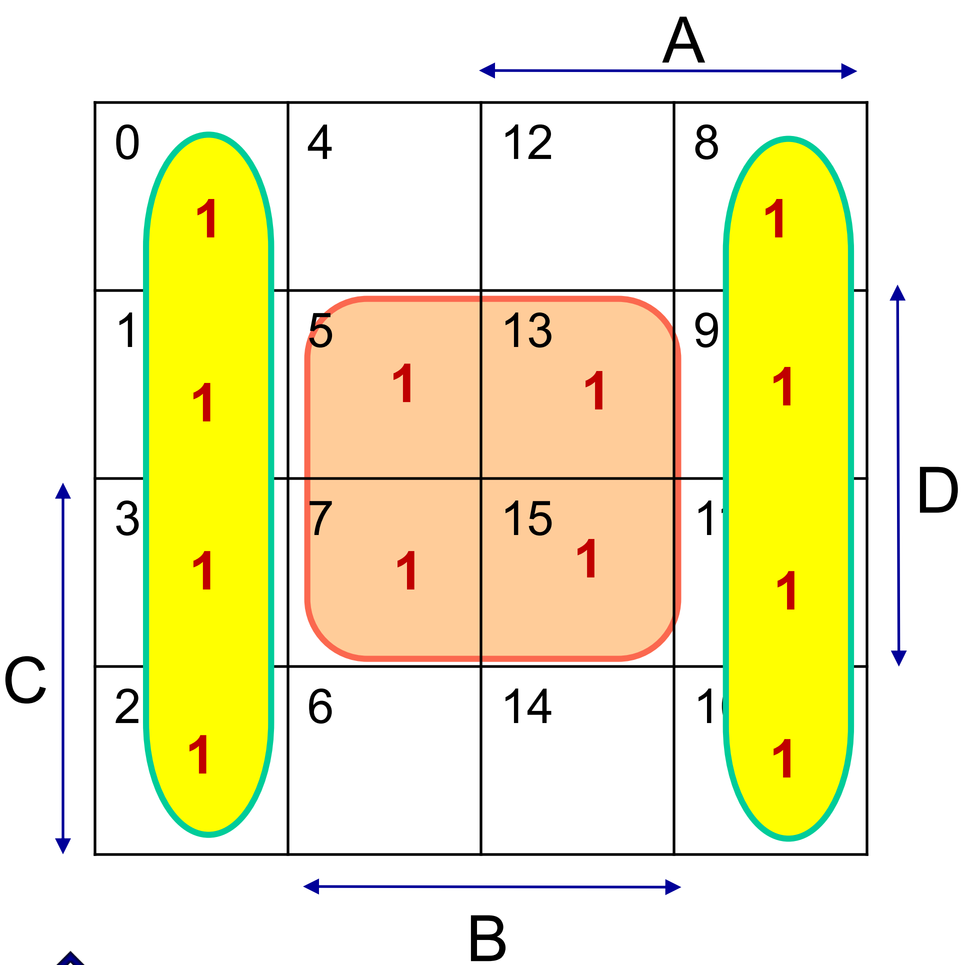
- A larger cube covers a smaller cube if all minterms of the smaller cube are included in the larger cube.
- A smaller cube implies (or subsumes) a larger cube if all minterms of the smaller cube are included in the larger cube.

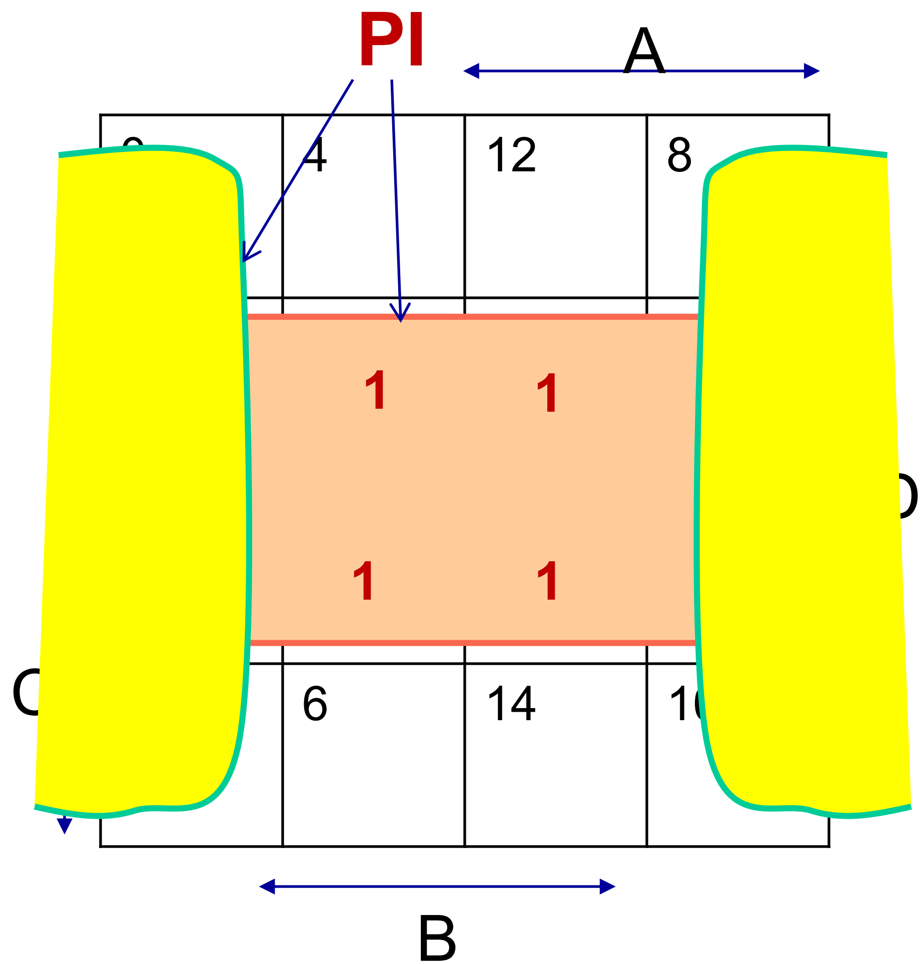
$$F = \overline{A}B + BD + \overline{A}CD + A\overline{B}\,\overline{C}\,\overline{D}$$



Prime Implicant (PI)

- A cube or implicant of a function that cannot grow larger by expanding into other cubes

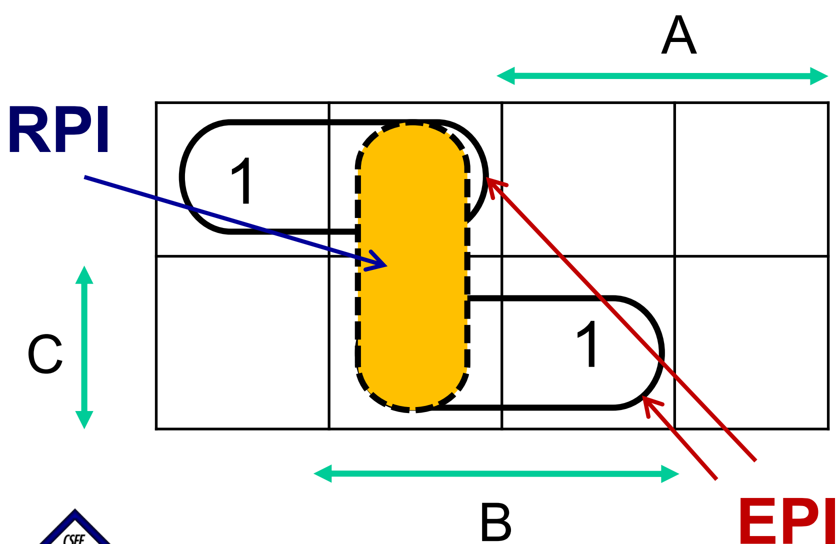


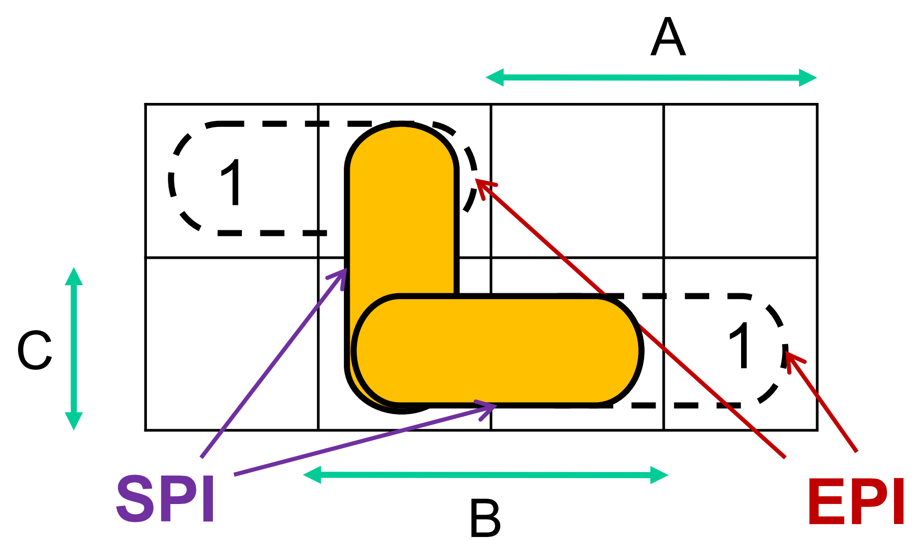


Slide contents are courtesy of Vishwani D. Agrawal

Type of Prime Implicants

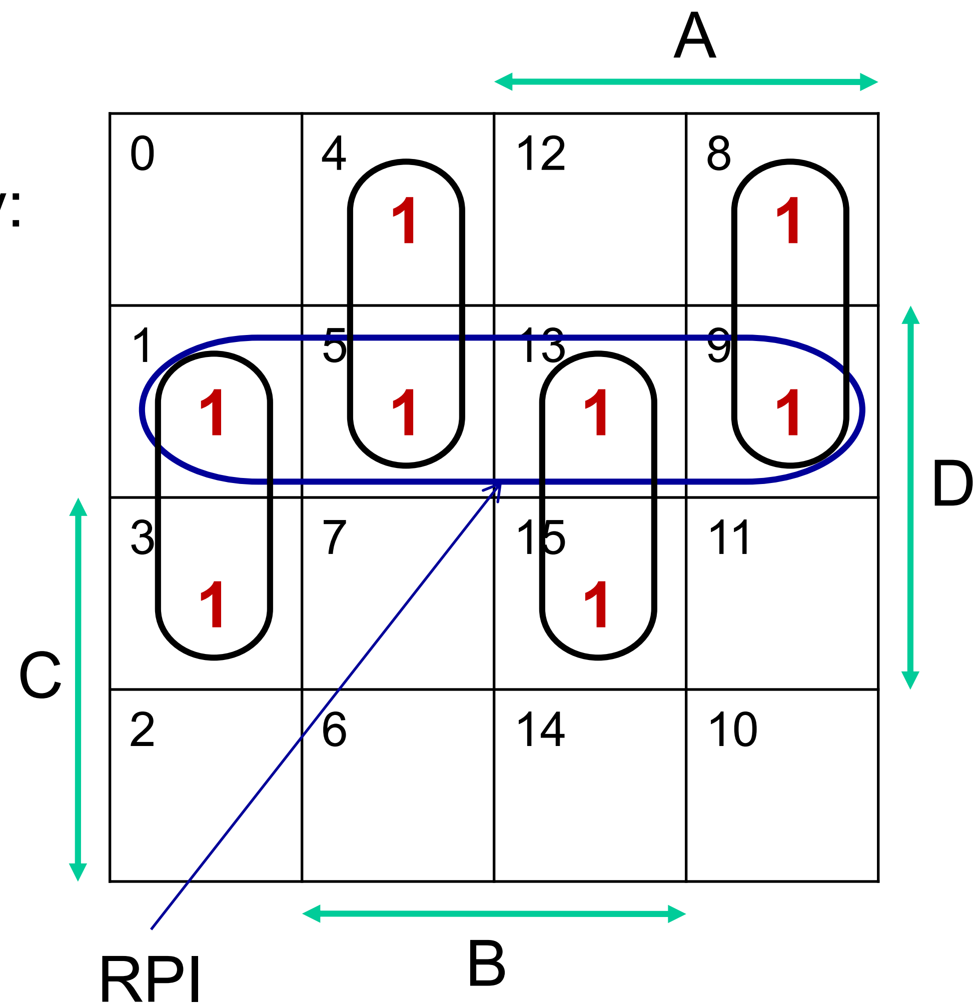
- If among the minterms subsuming a prime implicant (PI), there is at least one minterm that is covered by this and only this PI, then the PI is called an *essential* prime implicant (EPI)
- If each minterm subsuming a prime implicant (PI) is also covered by other essential prime implicants, then that PI is called a *redundant* prime implicant (RPI)
- A prime implicant (PI) that is neither EPI nor RPI is called a *selective* prime implicant (SPI); SPIs occur in pairs.





Minimum Sum of Products (MSOP)

- Identify all prime implicants (by letting implicants grow)
- Construct MSOP with PIs only:
  - Cover all minterms
  - Use only essential prime implicants (EPI)
  - Use no redundant prime implicant (RPI)
  - Use cheapest selective prime implicants (SPI), e.g. choose EPI in ascending order, starting from 0-implicant, then 1-implicant, 2-implicant, . . .



Example: $F=\sum m(1,3,4,5,8,9,13,15)$

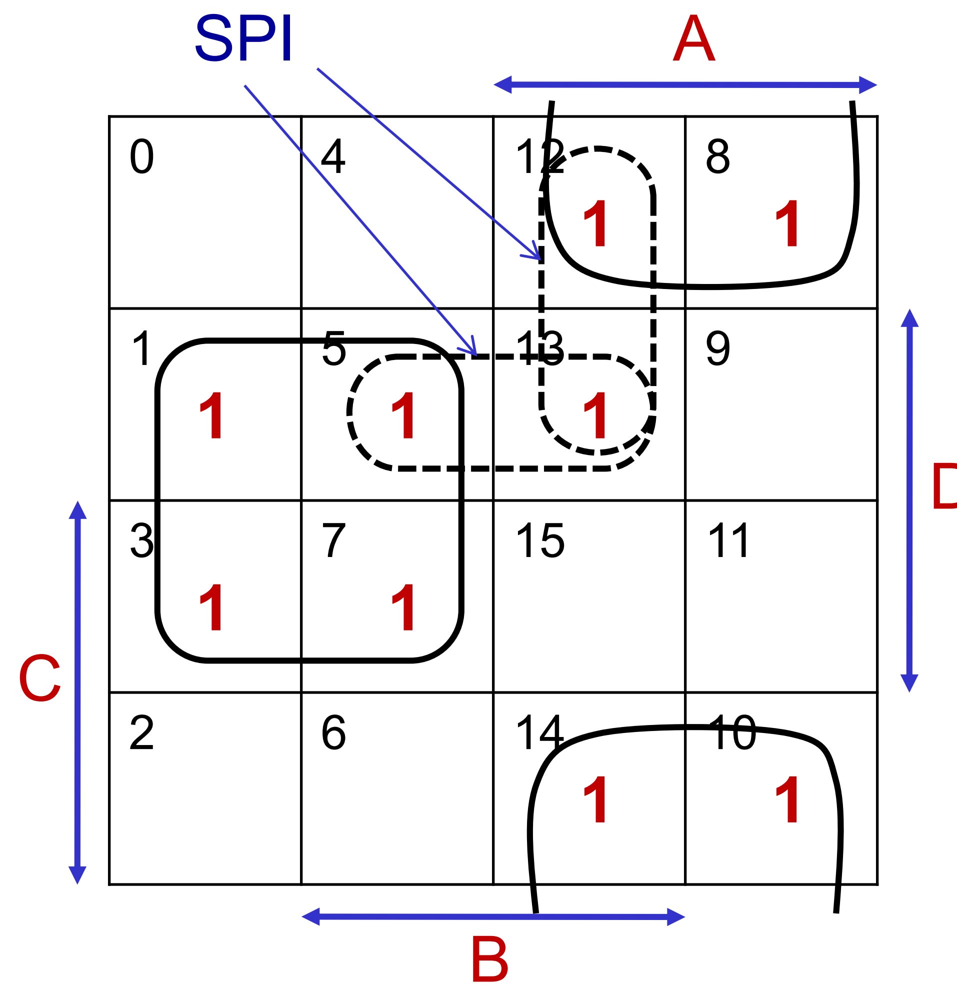
$$F = \bar{A}\,\bar{B}\,D + \bar{A}\,B\,\bar{C} + A\,B\,D + A\,\bar{B}\,\bar{C}$$

Example

$F=\sum m(1,3,5,7,8,10,12,13,14)$

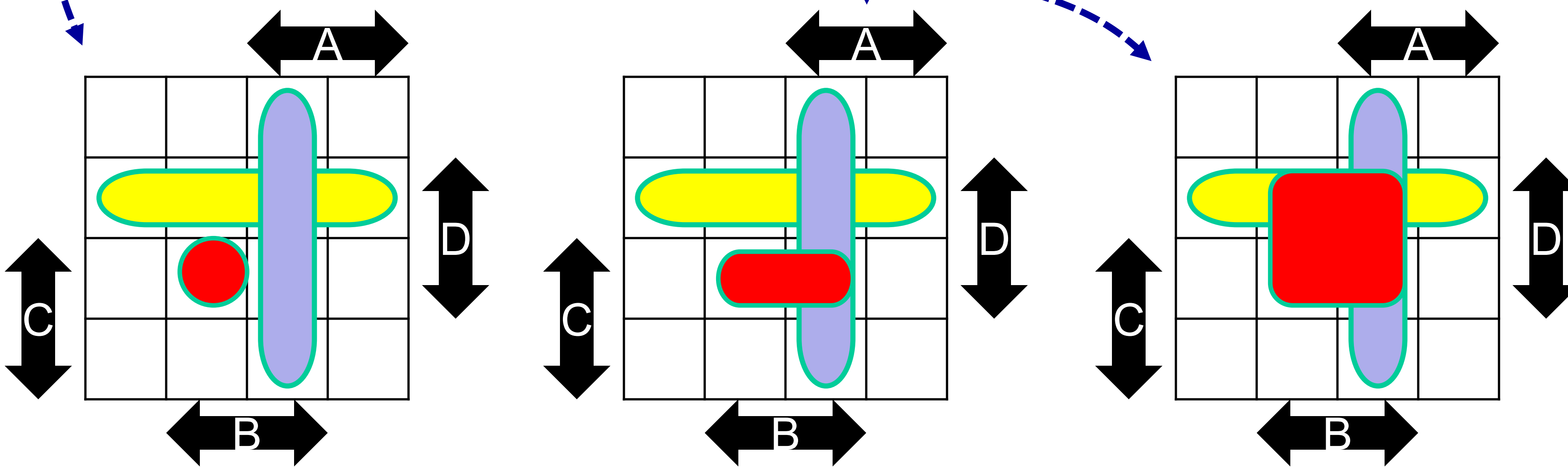
MSOP:

$F = \overline{A}D + A\overline{D} + AB\overline{C}$



Growing Implicants to PI

- $F = AB + \overline{C}D + \overline{A}BCD$ *initial implicants*
  $= AB + \overline{A}BCD + BCD + \overline{C}D$ *consensus th.*
  $= AB + BCD + \overline{C}D$ *absorption th.*
  $= AB + BCD + \overline{C}D + BD$ *consensus th.*
  $= AB + \overline{C}D + BD$ *absorption th.*



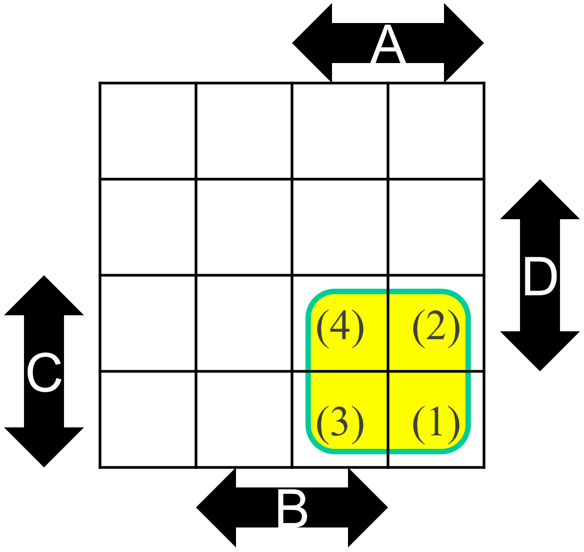
Minterms Covered by a Product

- A product from which k variables have been eliminated, covers 2^k minterms.
- Example: For four variables, A, B, C, D

Product AC covers $2^2 = 4$ minterms:

1) $A\ \bar{B}\ C\ \bar{D}$
2) $A\ \bar{B}\ C\ D$
3) $A\ B\ C\ \bar{D}$
4) $A\ B\ C\ D$

Obtained by inserting the eliminated variables in all possible ways.



Slide contents are courtesy of Vishwani D. Agrawal

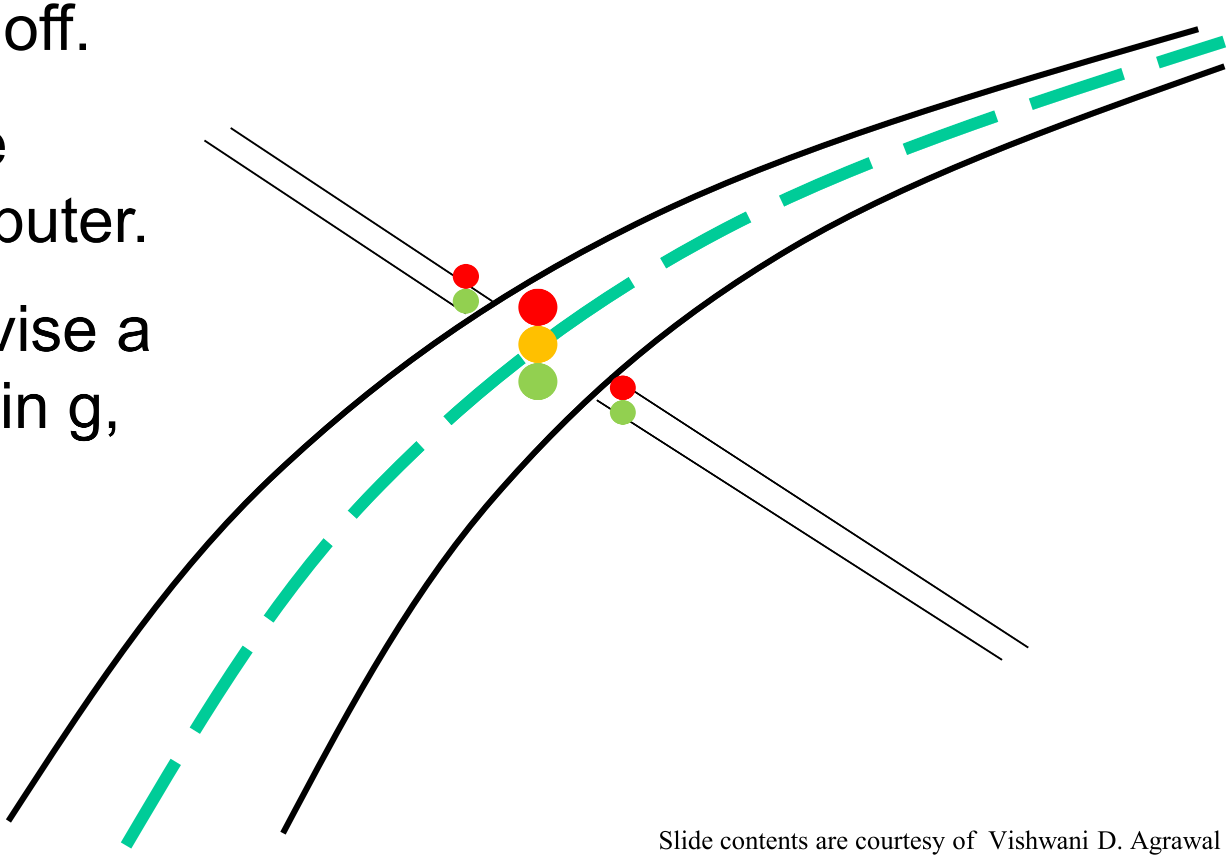
Truth Table with Don't Cares

- Don't cares entries represent combinations that are impossible to happen
- Example: X=1 indicates an elevator at top, and Y=1 indicates elevator at ground ➔ X and Y can never be one at the same time
- Don't care entries can be assumed any value and can thus leverage the simplification process
- There can be more than one minimal grouping, as a result of don't cares

A	B	C	D	F
0	0	0	0	d
0	0	0	1	1
0	0	1	0	0
0	0	1	1	1
0	1	0	0	0
0	1	0	1	1
0	1	1	0	1
0	1	1	1	1
1	0	0	0	0
1	0	0	1	0
1	0	1	0	1
1	0	1	1	d
1	1	0	0	0
1	1	0	1	1
1	1	1	0	0
1	1	1	1	d

Example: Traffic Signals

- Consider two roads crossing:
  - Highway with traffic signal, red (R), yellow (Y) or green (G).
  - Rural road with red (r) or green (g) signal.
- Here R, Y, G, r and g are Boolean variables; 1 implies light is on, 0 means light is off.
- Highway signals are controlled by a computer.
- We only need to devise a digital circuit to obtain g, because $r = \overline{g}$.

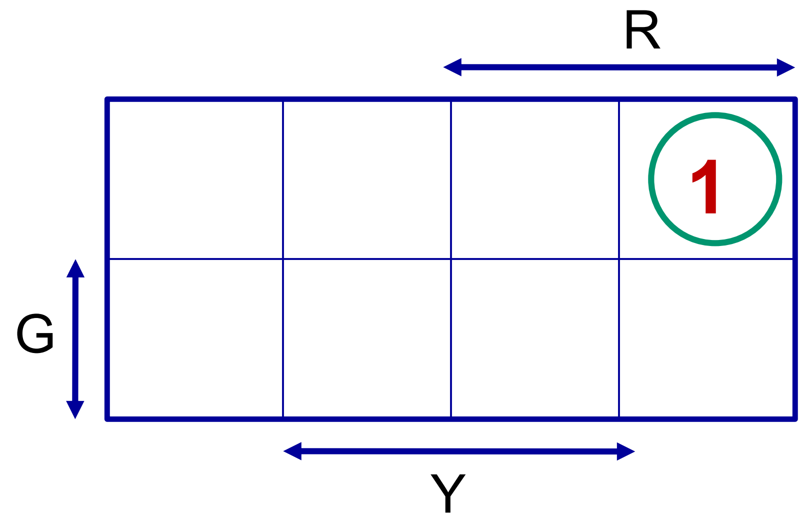


Slide contents are courtesy of Vishwani D. Agrawal

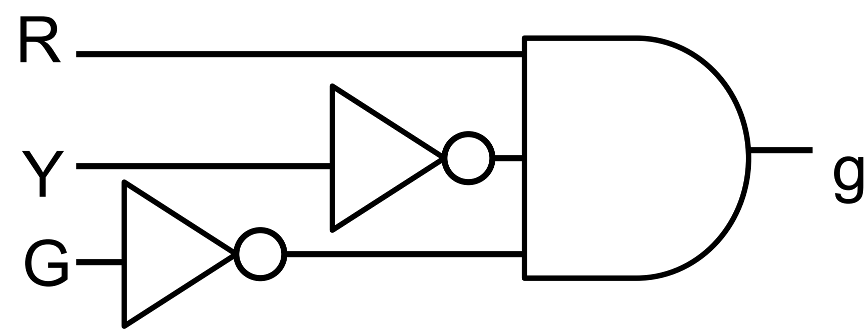
Completely Specified Function

Truth Table

[redacted]				
0	0	0	0	0
[redacted]				
2	0	1	0	0
[redacted]				
4	1	0	0	1
[redacted]				
6	1	1	0	0
[redacted]				

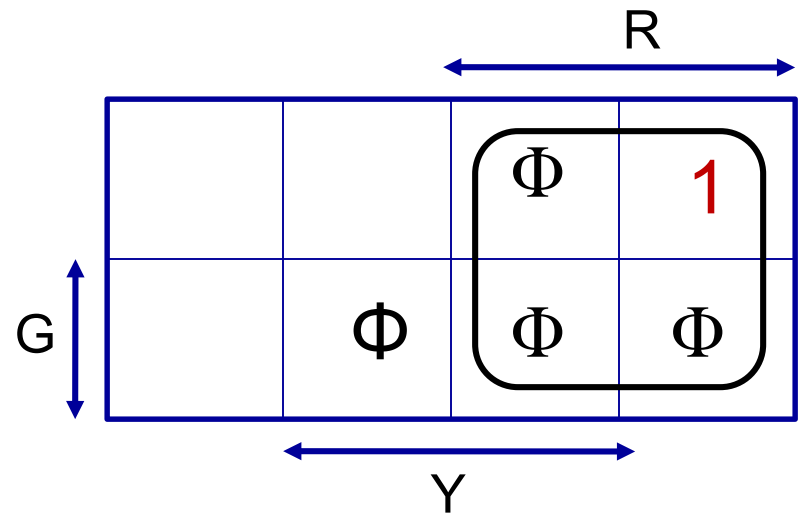


$$g = R\,\overline{Y}\,\overline{G}$$



Incompletely Specified Function

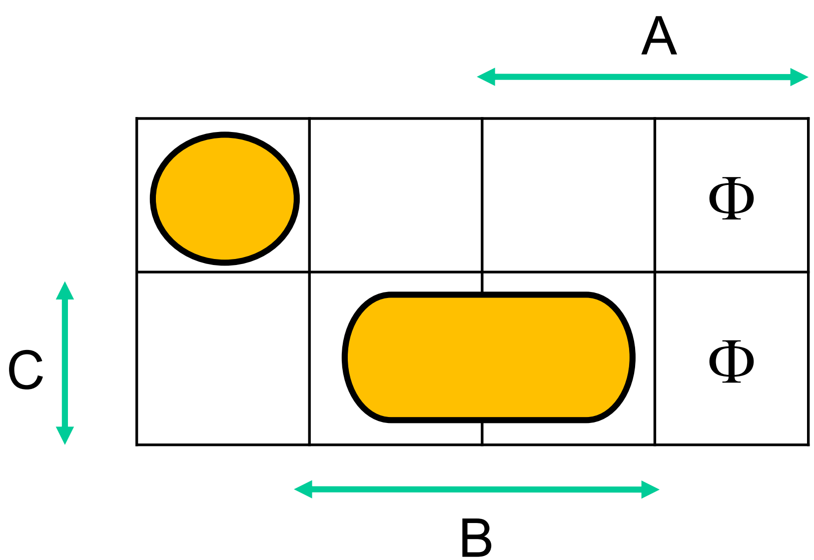
Truth Table				
[redacted]				
0	0	0	0	0
[redacted]				
2	0	1	0	0
[redacted]				
4	1	0	0	1
[redacted]				
6	1	1	0	Φ
[redacted]				



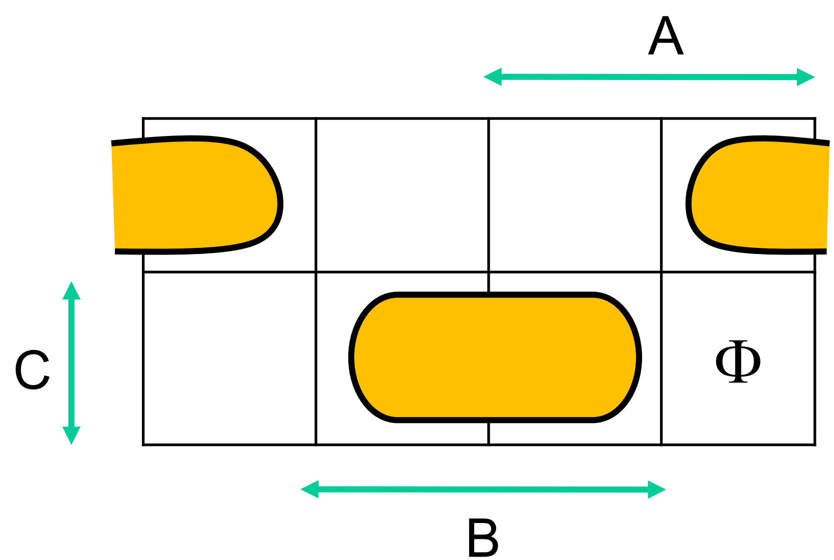
$$g = R$$

Functions with Don’t Care Minterms

- $F(A,B,C) = \sum m(0,3,7) + d(4,5)$
- Include don’t care minterms when beneficial



$F = B\,C + \bar{A}\,\bar{B}\,\bar{C}$

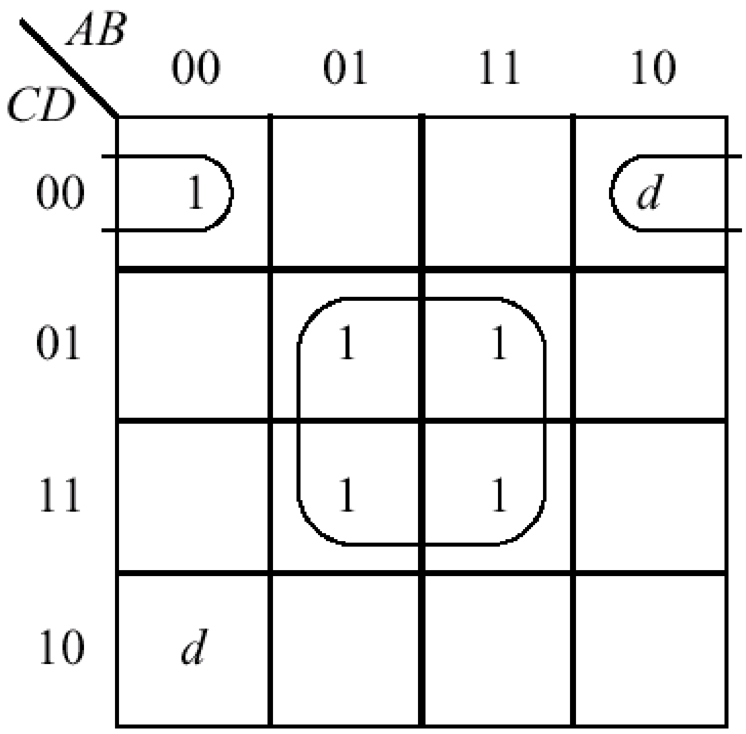


$F = B\,C + \bar{B}\,\bar{C}$

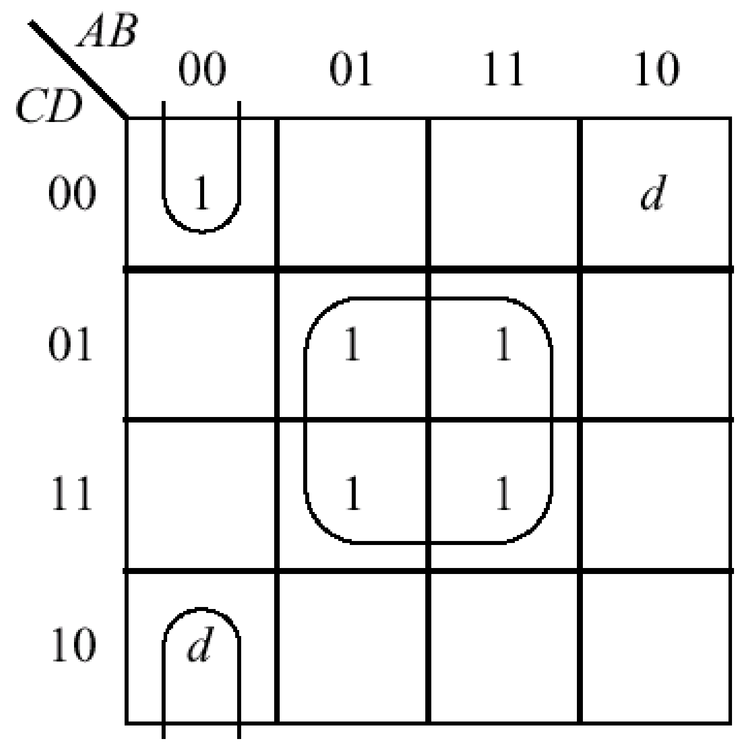
Slide contents are courtesy of Vishwani D. Agrawal

K-Maps and Don't Cares

- Don't care entries can be assumed any value and can thus leverage the simplification process
- There can be more than one minimal grouping, as a result of don't cares



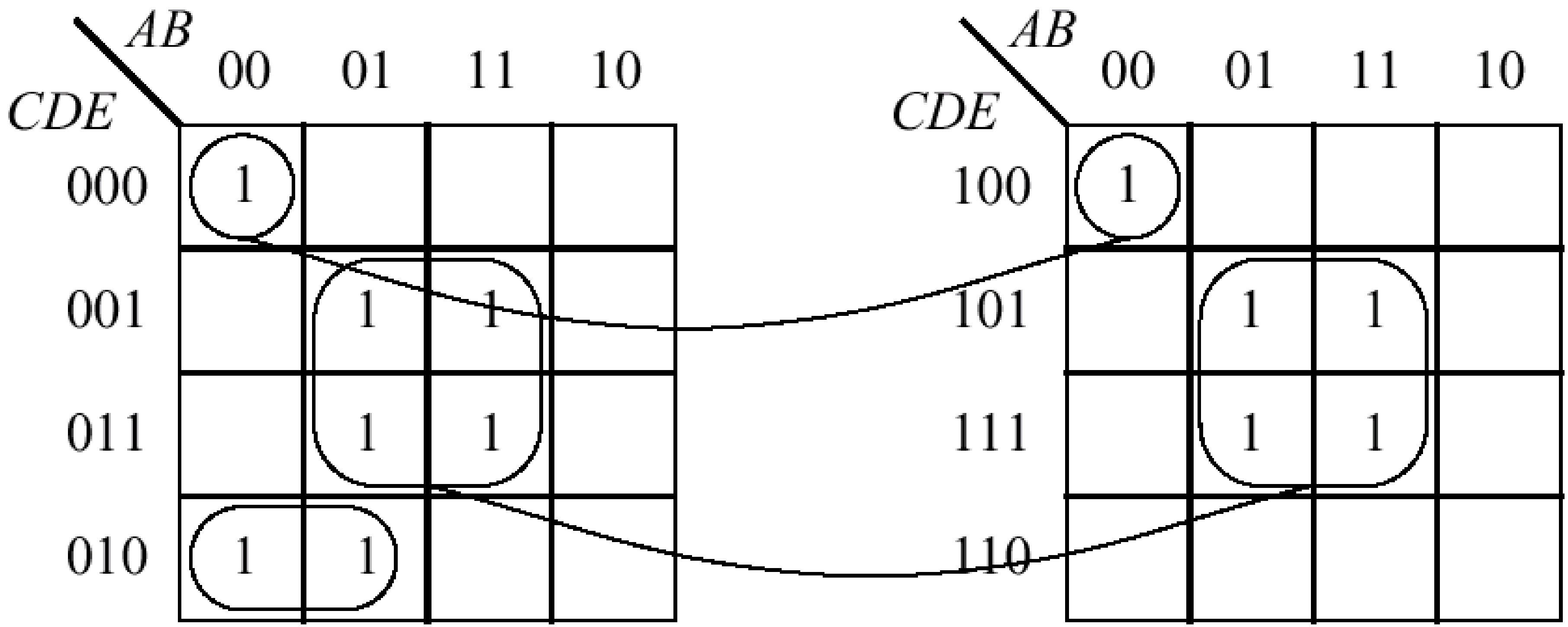
$$F = \overline{B}\,\overline{C}\,\overline{D} + B\,D$$



$$F = \overline{A}\,\overline{B}\,\overline{D} + B\,D$$

Five-Variable K-Map

- Visualize two 4-variable K-maps stacked one on top of the other; groupings are made in three dimensional cubes.

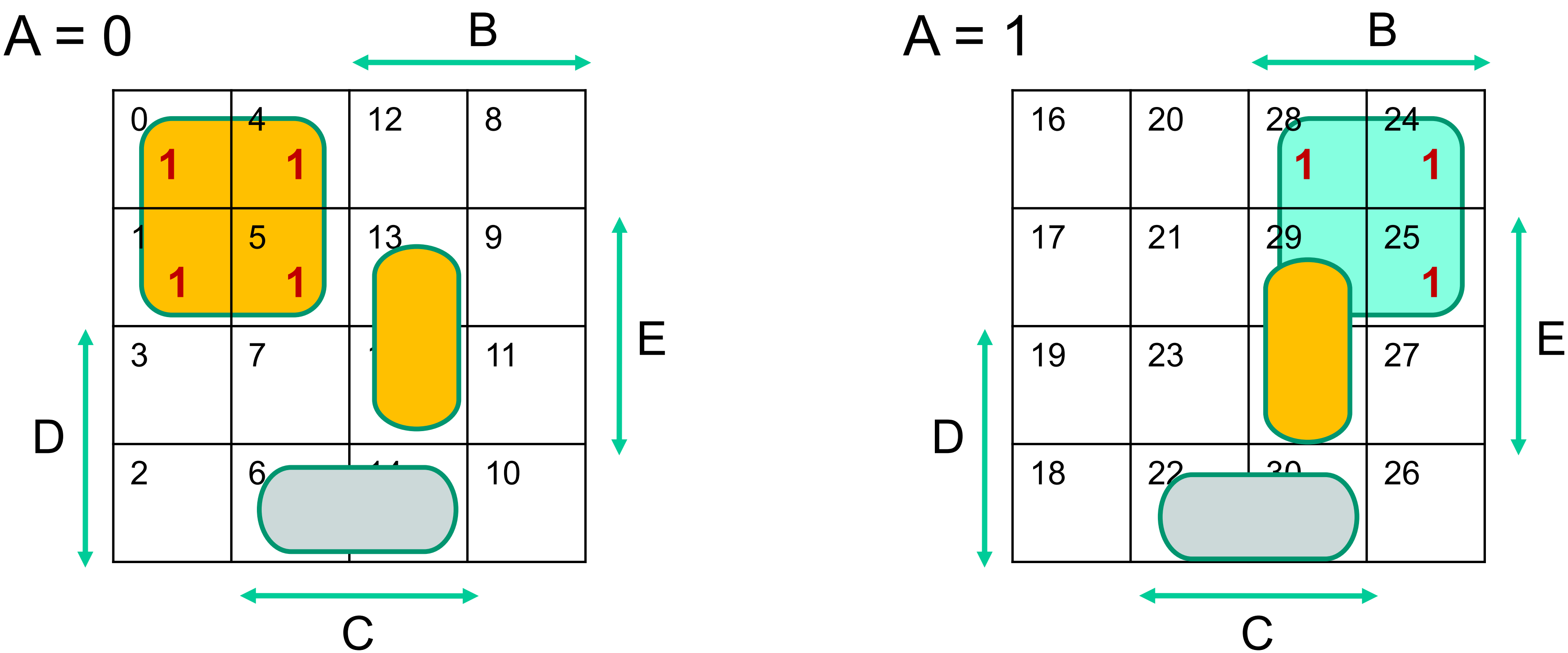


$$F = \overline{A}\,\overline{C}\,D\,\overline{E} + \overline{A}\,\overline{B}\,\overline{D}\,\overline{E} + B\,E$$

* Slide is courtesy of M. Murdocca and V. Heuring

Minimizing Five-Variable Function

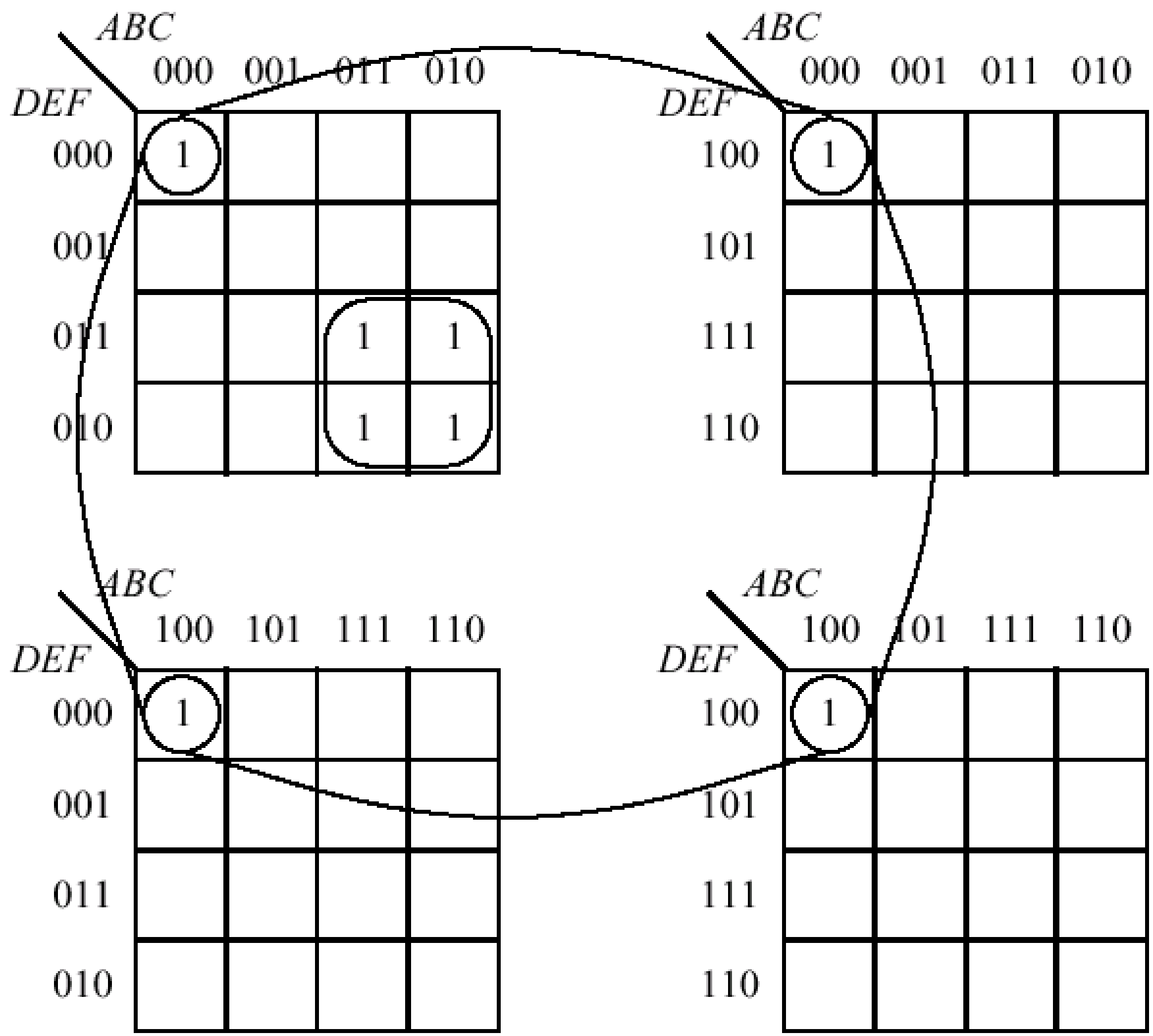
$$F(A,B,C,D,E) = \sum m(0,1,4,5,6,13,14,15,22,24,25,28,29,30,31)$$



$$F = \bar{A}\,\bar{B}\,\bar{D} + A\,B\,\bar{D} + B\,C\,E + C\,D\,\bar{E}$$

Six-Variable K-Map

- Visualize four 4-variable K-maps stacked one on top of the other; groupings are made in three dimensional cubes.



$$G = \overline{B}\,\overline{C}\,\overline{E}\,\overline{F} + \overline{A}\,B\,\overline{D}\,E$$

* Slide is courtesy of M. Murdocca and V. Heuring

Map-Entered Variables

- Allowing variables to be entered in the K-map simplifies the representation of some functions (less map size)

C \ AB	00	01	11	10
0	D			
1		1	1	

$$F = B\,C + \overline{A}\,\overline{B}\,\overline{C}\,D$$

- The map-entered variable D is treated as a 1 for the purpose of grouping, which in this case results in a 1-group since there are no adjacent 1's to the D cell

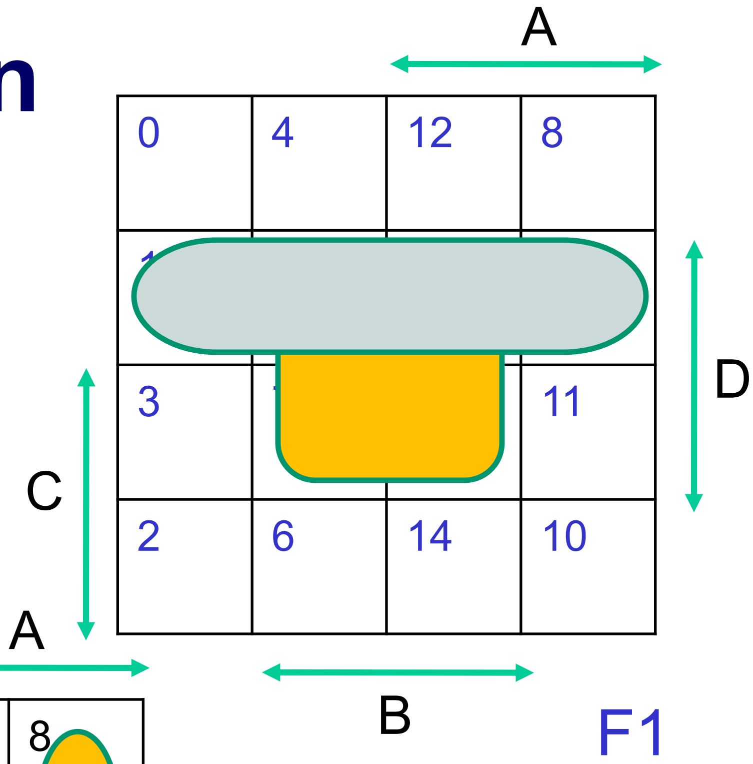
- Notice that the variable D appears in the final expression

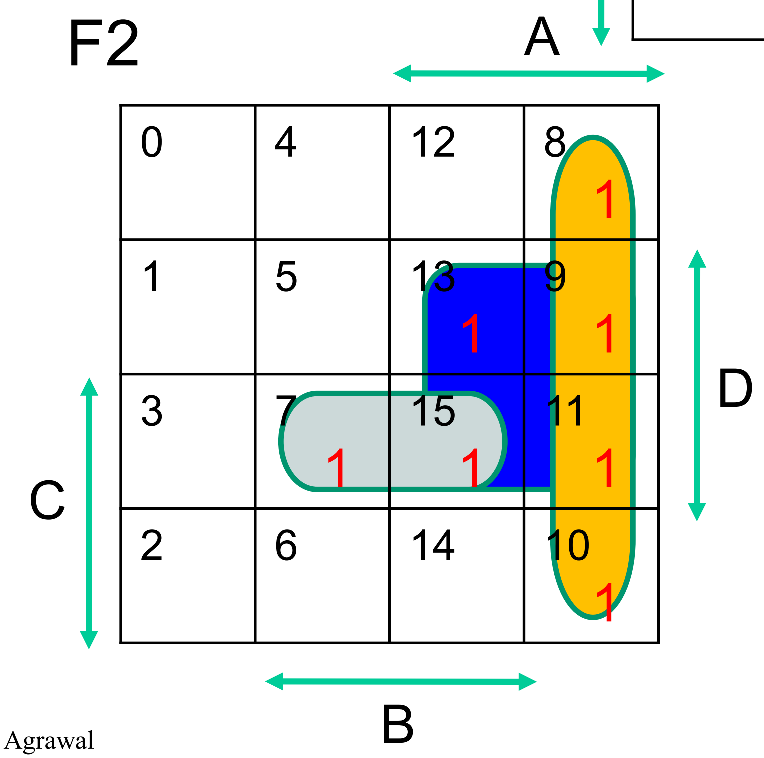
* Slide is courtesy of M. Murdocca and V. Heuring

Multiple-Output Function

Inputs				Outputs	
A	B	C	D	F1	F2
0	0	0	0	0	0
0	0	0	1	1	0
0	0	1	0	0	0
0	0	1	1	0	0
0	1	0	0	0	0
0	1	0	1	1	0
0	1	1	0	0	0
0	1	1	1	1	1
1	0	0	0	0	1
1	0	0	1	1	1
1	0	1	0	0	1
1	0	1	1	0	1
1	1	0	0	0	0
1	1	0	1	1	1
1	1	1	0	0	0
1	1	1	1	1	1

Need five products



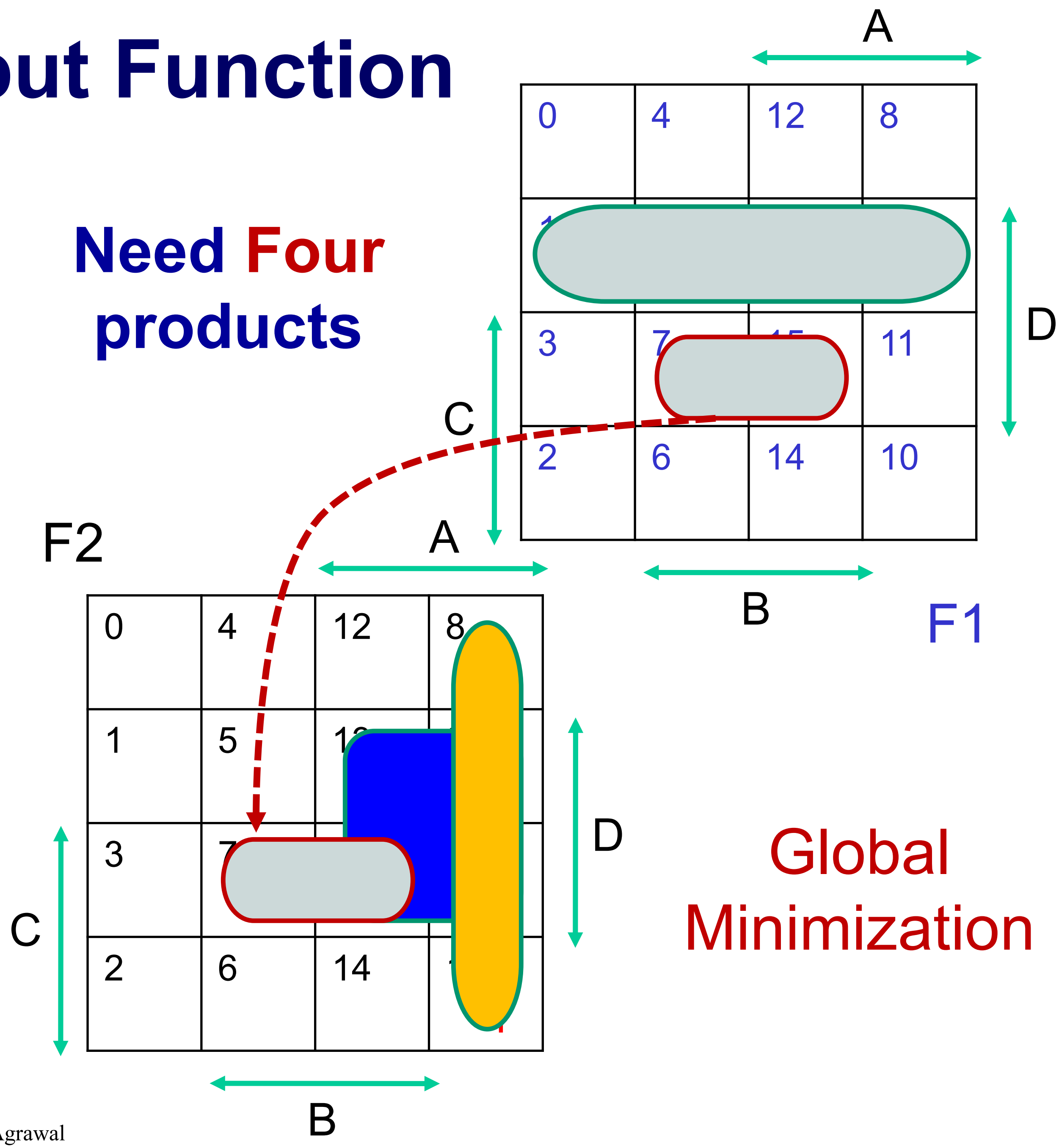


Individual Output Minimization

Multiple-Output Function

Inputs				Outputs	
A	B	C	D	F1	F2
0	0	0	0	0	0
0	0	0	1	1	0
0	0	1	0	0	0
0	0	1	1	0	0
0	1	0	0	0	0
0	1	0	1	1	0
0	1	1	0	0	0
0	1	1	1	1	1
1	0	0	0	0	1
1	0	0	1	1	1
1	0	1	0	0	1
1	0	1	1	0	1
1	1	0	0	0	0
1	1	0	1	1	1
1	1	1	0	0	0
1	1	1	1	1	1

Need Four products

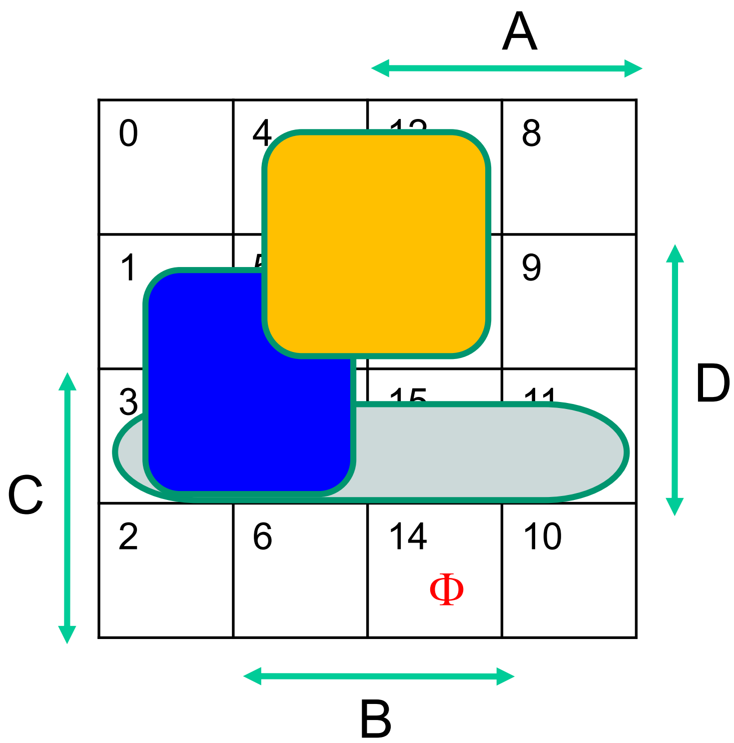


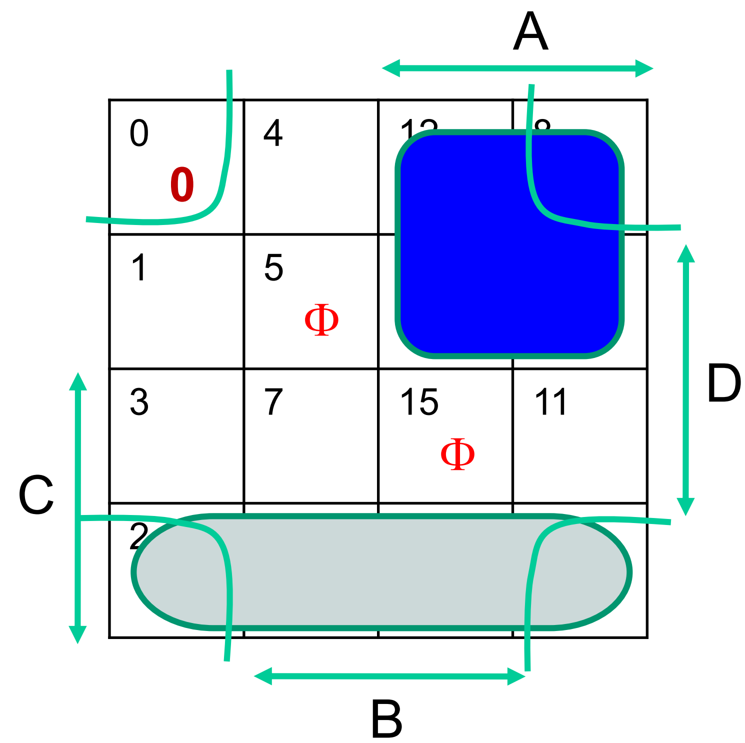
Global Minimization

Minimized SOP and POS

- $F(A,B,C,D) \quad = \quad \sum m(1,3,4,7,11) + d(5,12,13,14,15)$

  $\quad = \quad \Pi\, M(0,2,6,8,9,10)\;\; D(5,12,13,14,15)$





$F \quad = \quad B\,\overline{C} + \overline{A}D + C\,D$

$\overline{F} \quad = \quad \overline{B}\,\overline{D} + C\,\overline{D} + A\,\overline{C}$

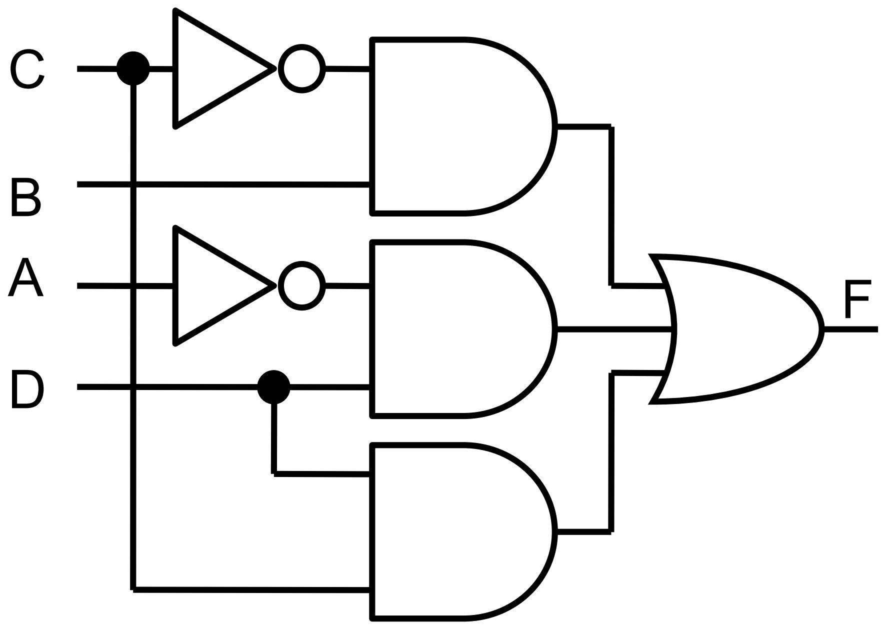
$F \quad = \quad (B + D)(\overline{C} + D)(\overline{A} + C)$

SOP and POS Circuits

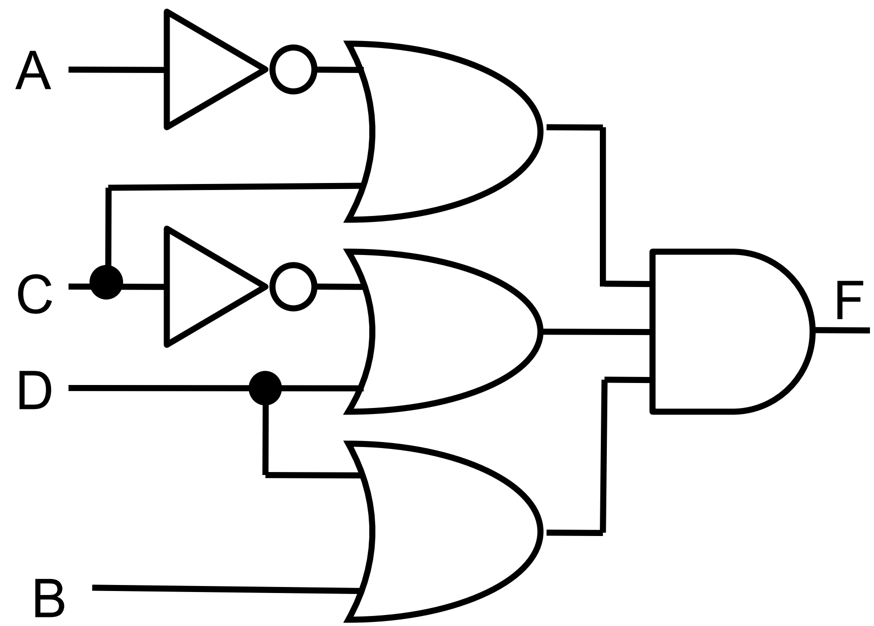
- $F(A,B,C,D) = \sum m(1,3,4,7,11)+d(5,12,13,14,15)$

  $= \prod M(0,2,6,8,9,10)\ D(5,12,13,14,15)$

$F = B\bar{C} + \bar{A}D + CD$



$F = (B + D)(\bar{C} + D)(\bar{A} + C)$



Are two circuits functionally Identical?

Conclusion

- ***Summary***
  - ➔ Reduction of combinational logic
    (Types, Goals, Methodology, Fundamental Concept)
  - ➔ The Karnaugh map method
    (Basic concept, Implicants, prime implicants)
  - ➔ Extended K-map procedure
    (Minterm covering, Multi-output optimization, map-entered variable)
  - ➔ Optimization of incompletely specified functions
  - ➔ Scalability limitation of the K-map method

- ***Next Lecture***
  - ➔ The Quine-McCluskey algorithm
  - ➔ Other circuit performance considerations

Reading assignment: Sections 3.1-3.7 in the textbook