Lecture 7 – Gate-Level Minimization – Part I

Rami Abielmona
University of Ottawa

February 03, 2015

ITI 1100 B

Digital Systems I

Presentation Outline

- Karnaugh Maps
- 2-,3-, 4-variable K-Maps
- Don’t Care Outputs
- K-Map Examples
- Key terms

The Karnaugh MAP

- An alternate approach to representing Boolean functions
- used to minimize Boolean functions
- Easy conversion from truth table to K-map
- Easy to obtain minimized SOP function.
- Simple steps used to perform minimization

→ **Much faster and more efficient than previous minimization techniques with Boolean algebra.**

The Karnaugh MAP

- K-MAP is ideally suited for four or less variables, becoming cumbersome for five or more variables.

→Each square represents a Minterm

→*Map is arranged such that two neighbors differ in only one variable* *(e.g. ABC +ABC')*

→ *Two terms must be "adjacent" in the map*

→ A K-map of n variables will have 2^n squares

→ For a Boolean expression, product terms are denoted by 1's, while sum terms are denoted by 0's – or left blank (represented by Minterms in the map)

K-Map with Two variables

B \ A	A'	A
B'	A'B'	AB'
B	A'B	AB

B \ A	0	1
0	00	10
1	01	11

B \ A	0	1
0	m_0	m_2
1	m_1	m_3

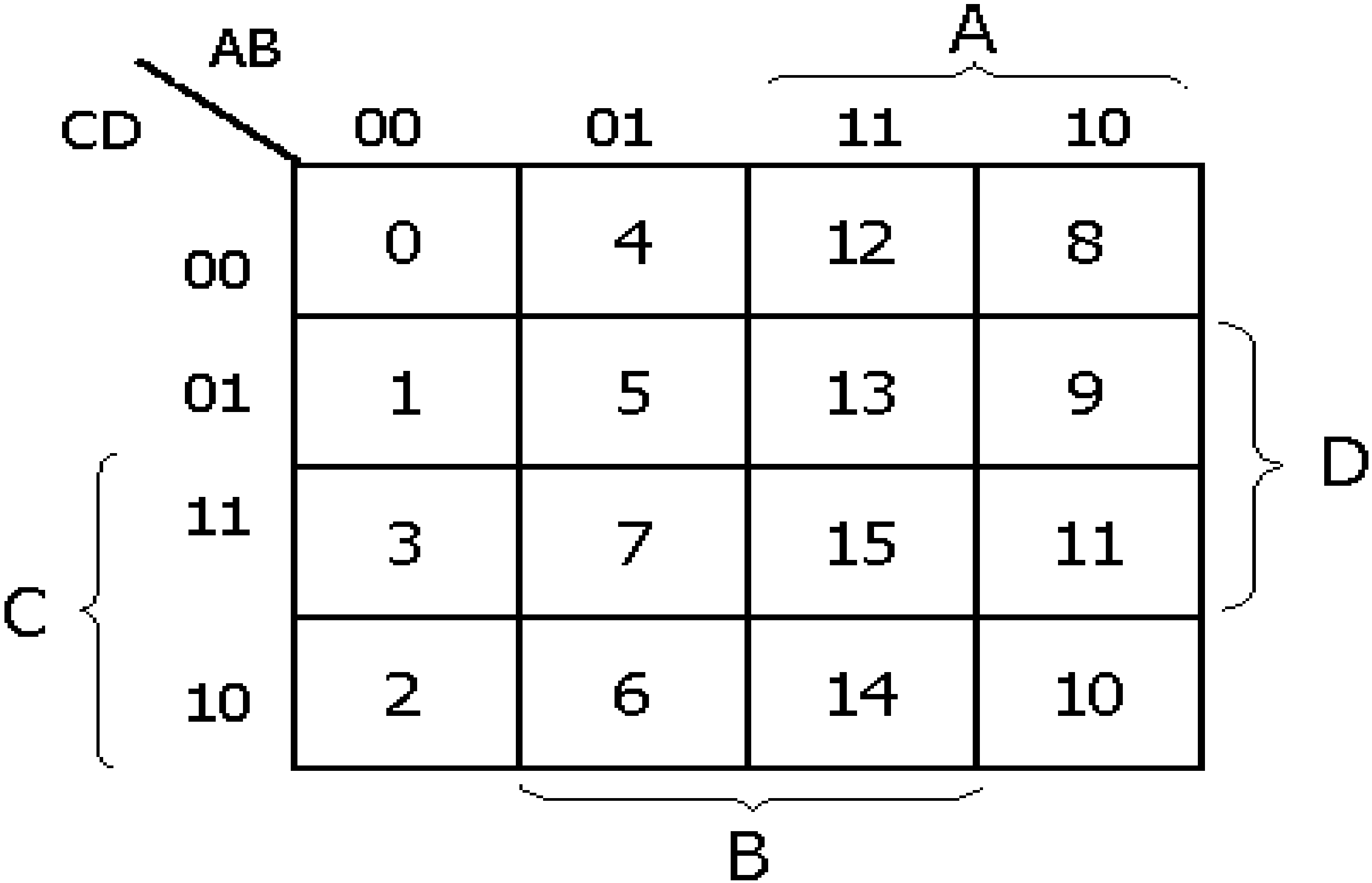
K-Map with 3 variables

C \ AB	A'B'	A'B	AB	AB'
C	**A'B'C'**	**A'BC'**	**ABC'**	**AB'C'**
C'	**A'B'C**	**A'BC**	**ABC**	**AB'C**

↓

C \ AB	00	01	11	10
0	0	2	6	4
1	1	3	7	5

Kmap With 4 variables



Assigning 1's and 0's in Kmap

- Assign the value of the outputs to the corresponding Minterms in the K-map

F (A,B,C,D)= A'B'C'D'+A'BC'D'+ AB'C'D' + A'BC'D+ ABC'D + ABCD'+AB'CD'

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

→Consider the squares with 1's to simplify SOP

→ Consider the squares with 0's to simplify POS

Karnaugh Maps - grouping squares

- **Groups of squares are formed in considering the following rules:**

-Every square containing 1 must be considered at least once

- A square containing 1 can be included in as many groups as desired

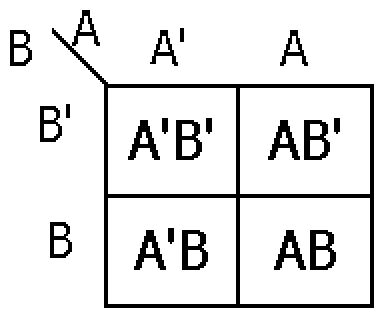
- A group must be as large as possible (i.e. large number of squares)

- The number of squares in a group must be equal to 2^n, i.e. 2,4,8,.

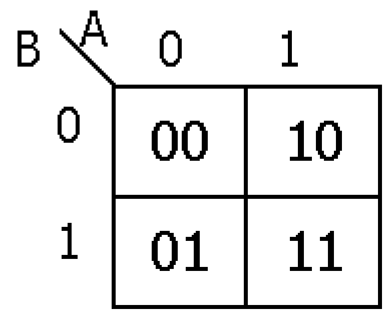
→ the simplified logic expression obtained from a K-map is not always unique. Groupings can be made in different ways.

2 variable Karnaugh Map

$x + x = 1$

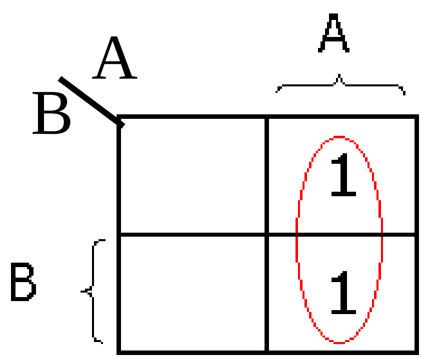


→

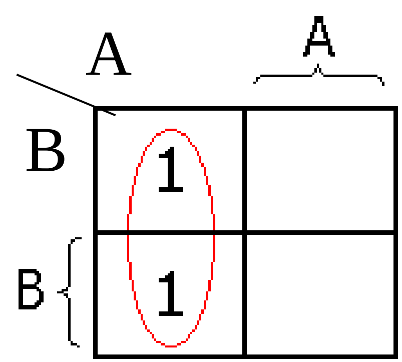


F=AB’+AB

F=A’B’+A’B



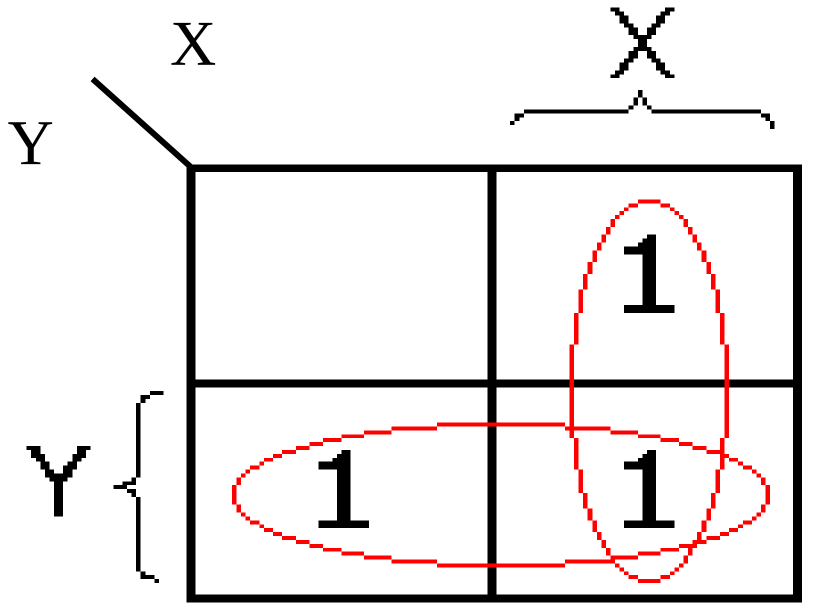
F = A



F = A’

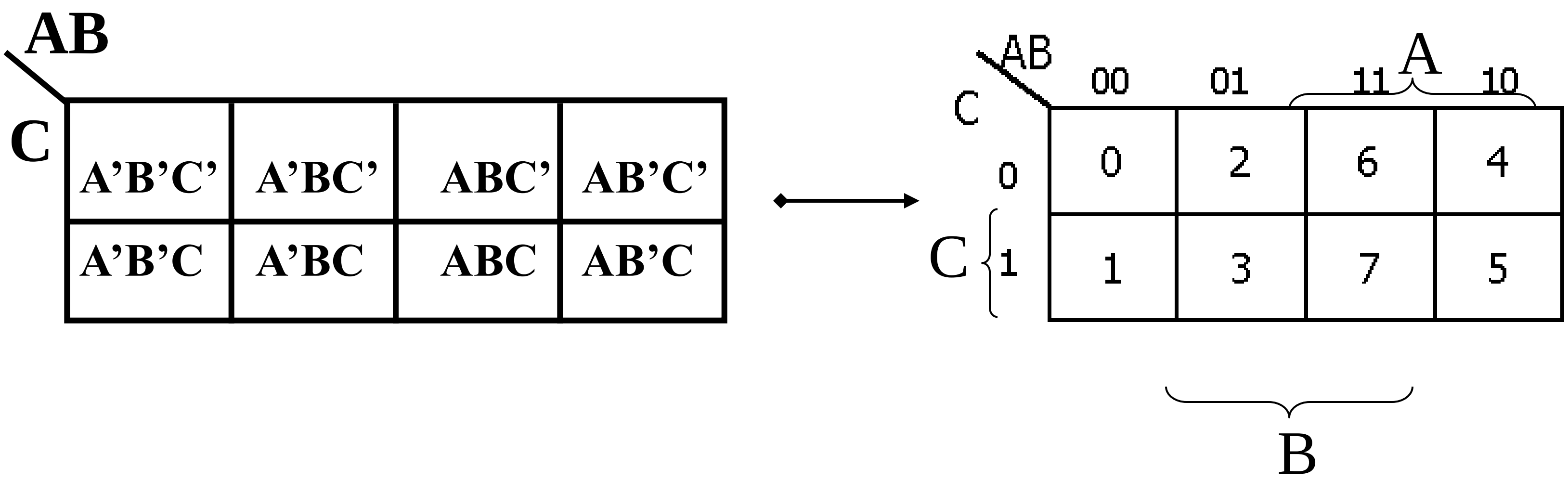
2 variable Karnaugh Map

F =X'Y+XY+XY'

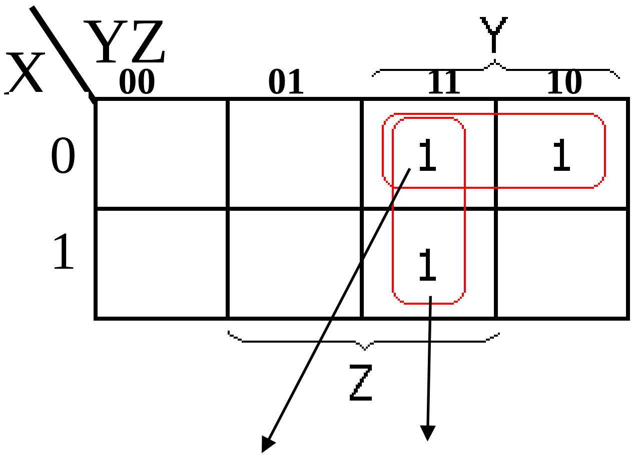


F = X + Y

3 Variable Karnaugh Map

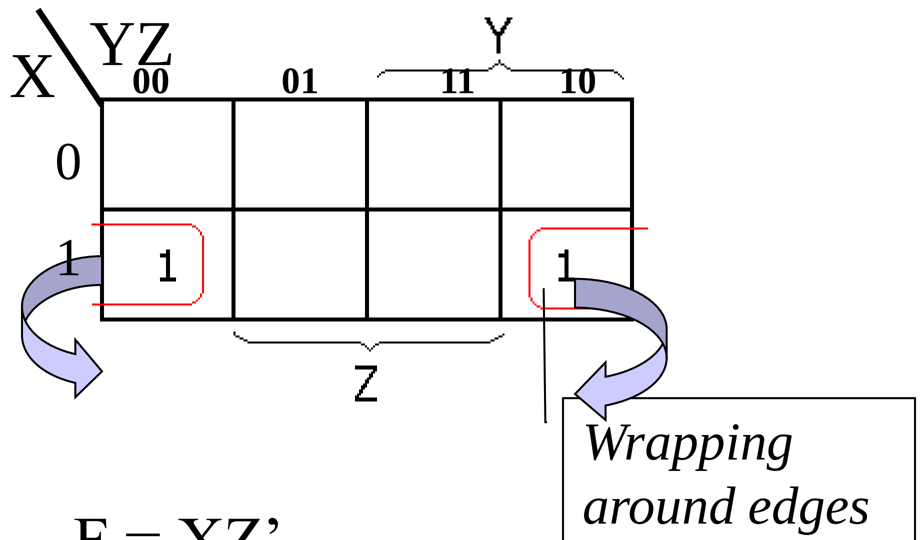


F = X’YZ’+XYZ+X’YZ



F = X’Y+YZ

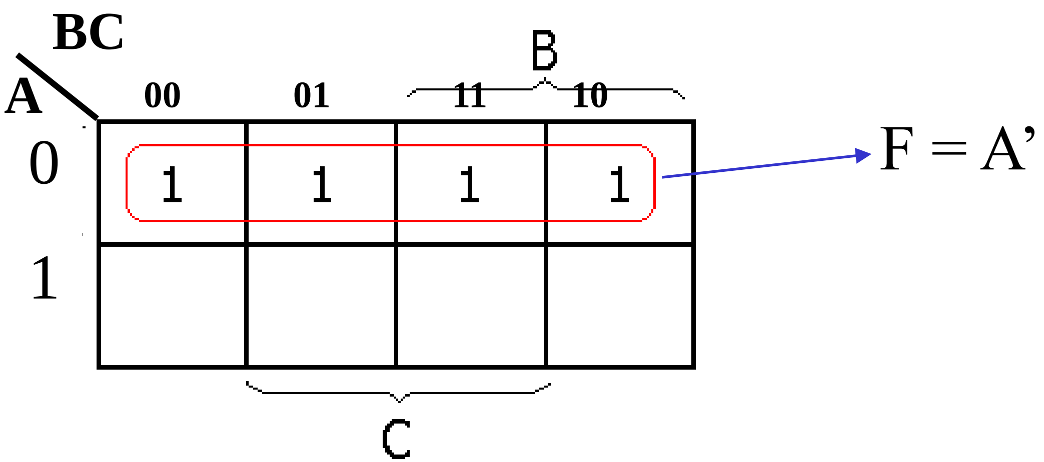
F = XY’Z’+XYZ’



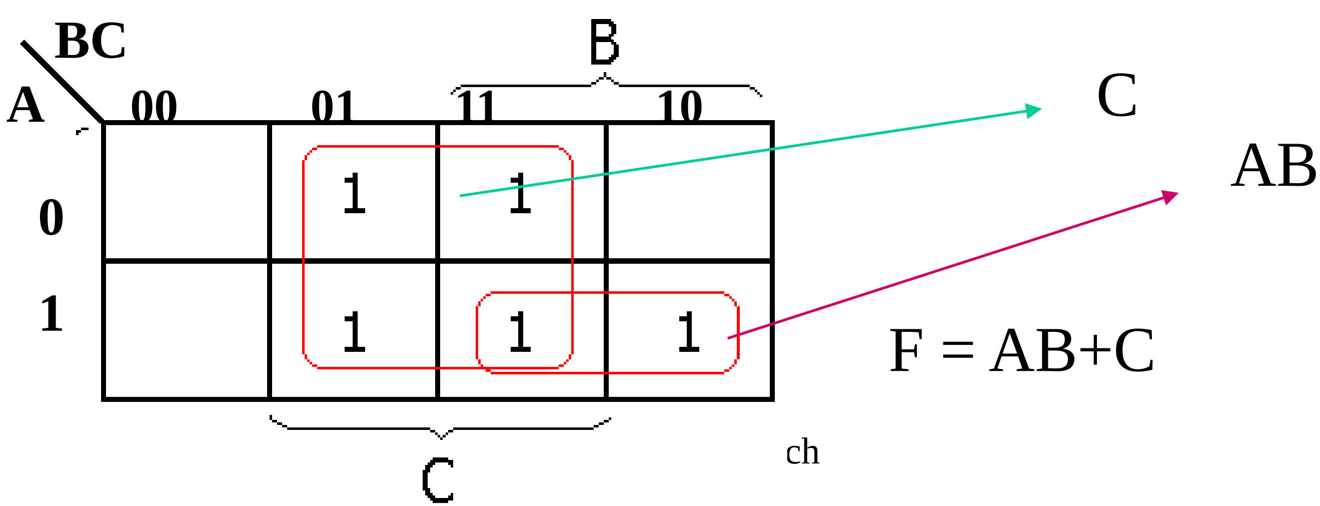
F = XZ’

3 variable Karnaugh Map

F(A,B,C) = A’BC’+A’B’C’+A’BC+A’B’C

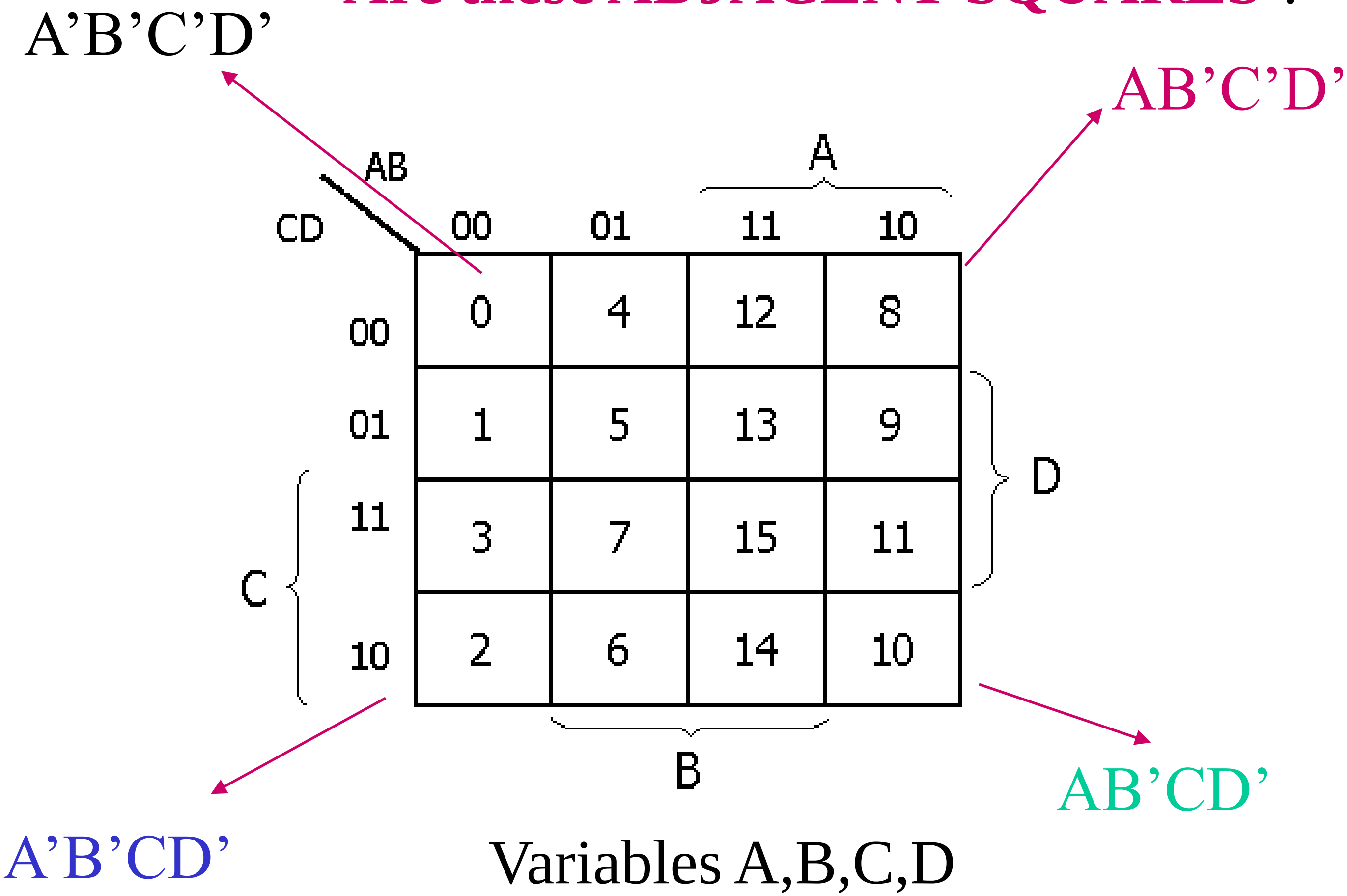


F(A,B,C) = A’BC+A’B’C+AB’C+ABC+ABC’



4 variables K-MAP

Are these ADJACENT SQUARES ?



Variables A,B,C,D

Function with “don’t care” Outputs

• Example

A purely binary number is converted into a 5-4-2-1 BCD number recall that BCD is often used to represent numbers in computers. The truth table is as below.

A	B	C	D	W	X	Y	Z
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1
0	0	1	0	0	0	1	0
0	0	1	1	0	0	1	1
0	1	0	0	0	1	0	0
0	1	0	1	1	0	0	0
0	1	1	0	1	0	0	1
0	1	1	1	1	0	1	0
1	0	0	0	1	0	1	1
1	0	0	1	1	1	0	0
1	0	1	0				
1	0	1	1				
1	1	0	0				
1	1	0	1				
1	1	1	0				
1	1	1	1				

Function with "don't care" Outputs

- Example

A purely binary number is converted into a 5-4-2-1 BCD number recall that BCD is often used to represent numbers in computers. The truth table is as below.

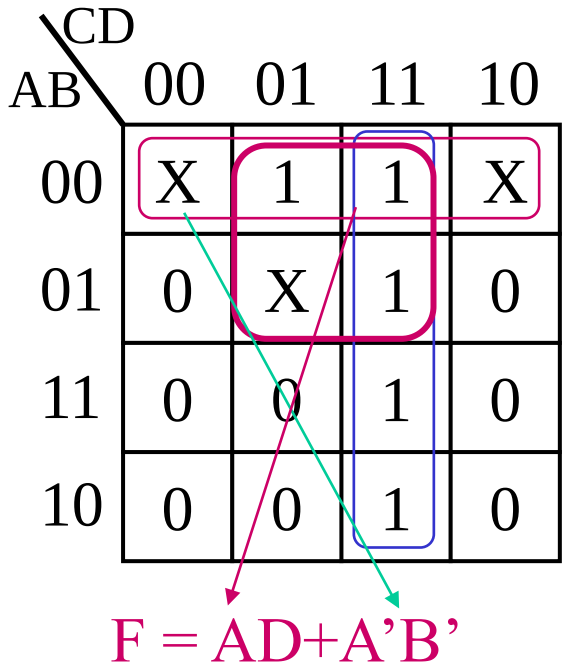
A	B	C	D	W	X	Y	Z
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1
0	0	1	0	0	0	1	0
0	0	1	1	0	0	1	1
0	1	0	0	0	1	0	0
0	1	0	1	1	0	0	0
0	1	1	0	1	0	0	1
0	1	1	1	1	0	1	0
1	0	0	0	1	0	1	1
1	0	0	1	1	1	0	0
1	0	1	0				
1	0	1	1				
1	1	0	0				
1	1	0	1				
1	1	1	0				
1	1	1	1				

Don't care terms (rows 1010 through 1111)

$\Sigma d(10,11,12,13,14,15)$ are don't care outputs for W, X, Y,Z

K-map with Don’t Care outputs

- Don’t care outputs can be either 0 or 1.
- This can be used to help simplify logic functions.
- Example: F(A,B,C,D)= Σ m(1,3,7,11,15 with Σ d(0,2,5)



- X denotes a “don’t care” term.
- X are used as 1’s or 0’s to increase the number of squares during the grouping

Solution to the 5-4-2-1 BCD example

A	B	C	D	W	X	Y	Z
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1
0	0	1	0	0	0	1	0
0	0	1	1	0	0	1	1
0	1	0	0	0	1	0	0
0	1	0	1	1	0	0	0
0	1	1	0	1	0	0	1
0	1	1	1	1	0	1	0
1	0	0	0	1	0	1	1
1	0	0	1	1	1	0	0
1	0	1	0				
1	0	1	1				
1	1	0	0				
1	1	0	1				
1	1	1	0				
1	1	1	1				

Don’t care terms (rows 1010 through 1111)

Using K-maps for the 4 variable we obtain:

W=A+BD+BC

X=BC’D’+AD

Y=CD+B’C+AD’

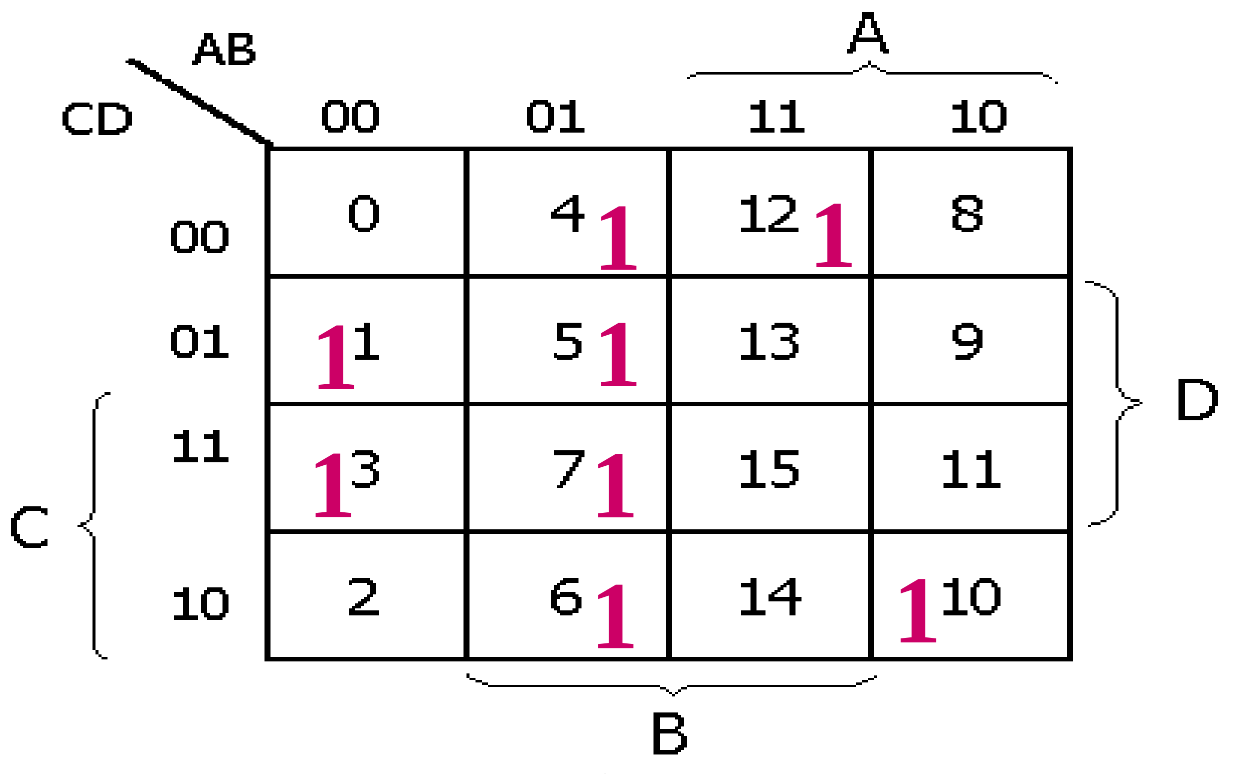
Z=AD’+A’B’D+BCD’

K-Maps- Examples

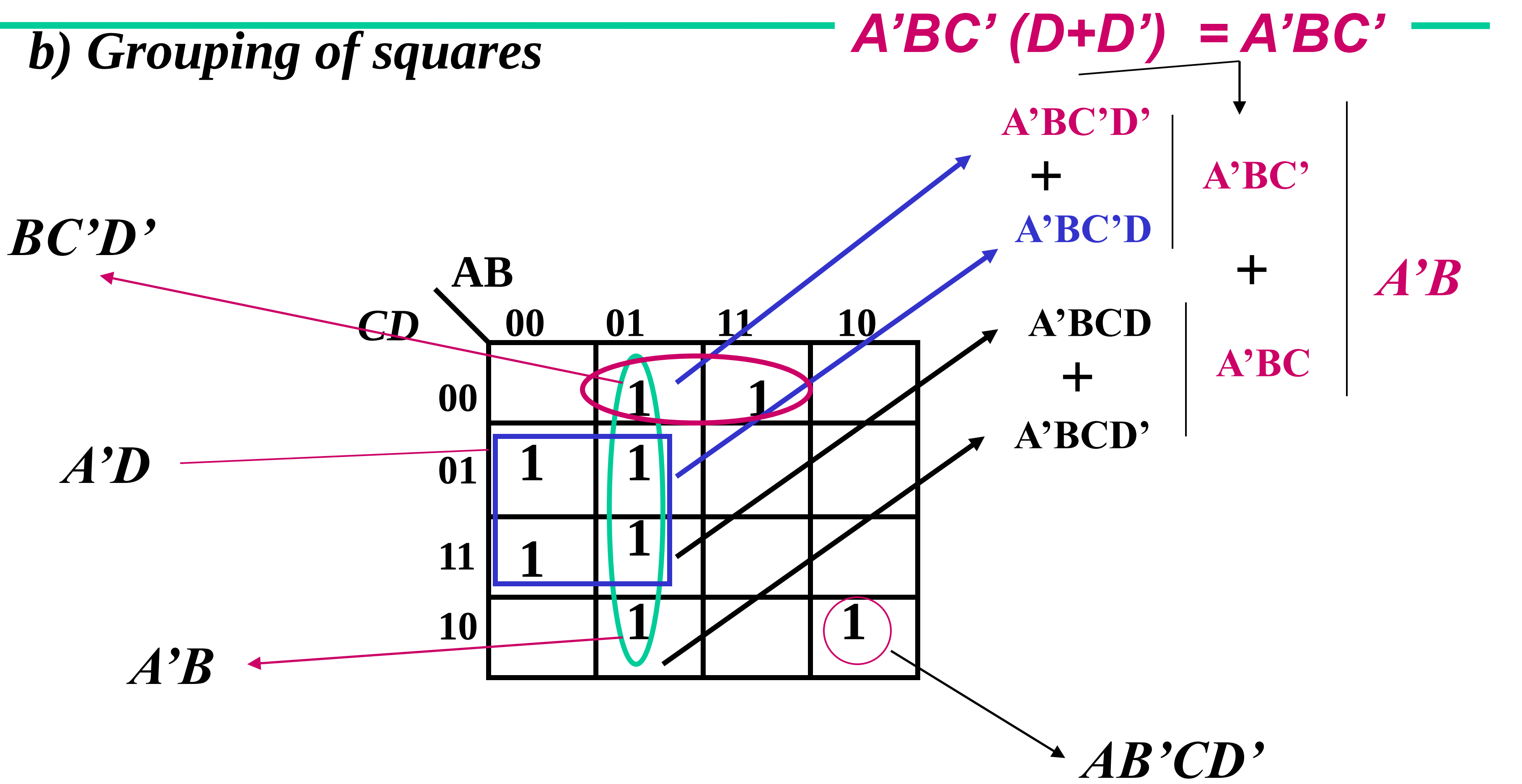
1- simplify the following expression using K-Maps

$$F(A,B,C,D) = \Sigma\, m\,(1, 3, 4, 5, 6, 7, 10, 12)$$

a) Building K-Map for F



b) Grouping of squares



c) Write the Simplified Expression

F(A,B,C,D) = A'B + A'D + BC'D' + AB'CD'

A	*B*	*C*	*D*	*F*
0	0	0	0	1
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	1
0	1	0	1	1
0	1	1	0	0
0	1	1	1	0
1	0	0	0	1
1	0	0	1	0
1	0	1	0	x
1	0	1	1	x
1	1	0	0	x
1	1	0	1	x
1	1	1	0	x
1	1	1	1	x

a) Building K-map from the truth table

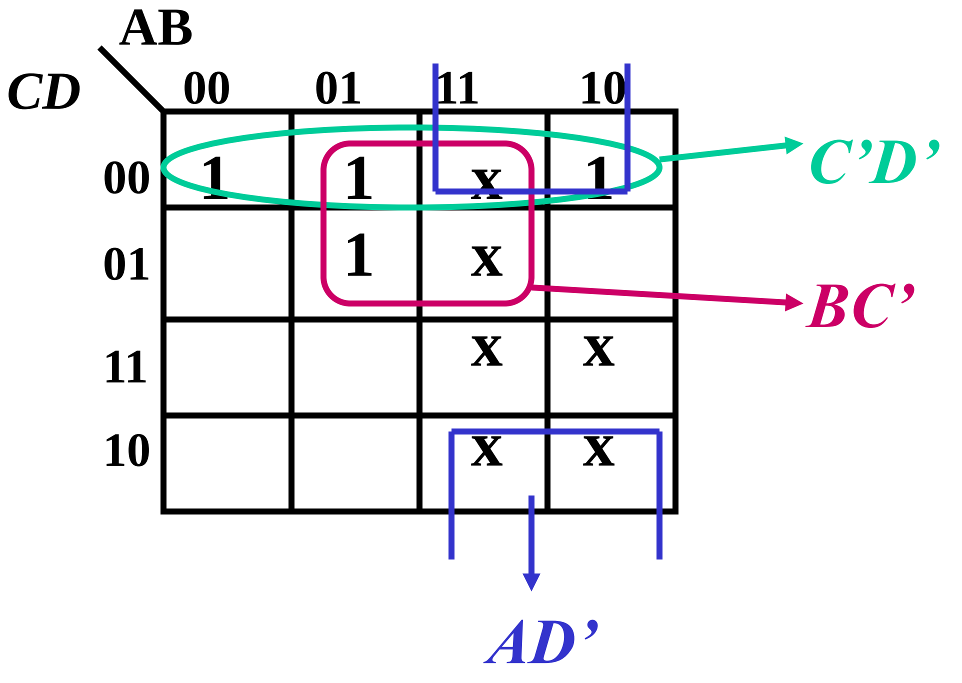
CD \ AB	00	01	11	10
00	**1**	**1**	**x**	**1**
01	**0**	**1**	**x**	**0**
11	**0**	**0**	**x**	**x**
10	**0**	**0**	**x**	**x**

A	*B*	*C*	*D*	*F*
0	0	0	0	1
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	1
0	1	0	1	1
0	1	1	0	0
0	1	1	1	0
1	0	0	0	1
1	0	0	1	0
1	0	1	0	x
1	0	1	1	x
1	1	0	0	x
1	1	0	1	x
1	1	1	0	x
1	1	1	1	x

a) Building K-map from the truth table

CD \ AB	00	01	11	10
00	1	1	x	1
01	0	1	x	0
11	0	0	x	x
10	0	0	x	x

b) Obtain Sum of Products for F



$$F = BC' + C'D'$$

Key Terms and Review Points

- Adjacent Cells
- Don’t Care Outputs
- Edge Wrapping
- Karnaugh Map
- K-Map Minimization for 2-, 3- and 4-variable Functions
- Minimization
- Optimization of Digital Circuits

- References: Dr. Karmouch ITI 1100 Slides