

# 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.

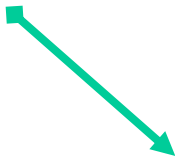
# The Karnaugh MAP

---

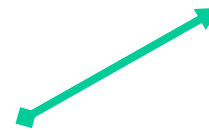
- K-MAP is ideally suited for four or less variables, becoming cumbersome for five or more variables.
  - Each square represents a Minterm
  - *Map is arranged such that two neighbors differ in only one variable (e.g.  $ABC + ABC'$ )*
  - *Two terms must be “adjacent” in the map*
  - A K-map of  $n$  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

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



	A	0	1
0	00	10	
1	01	11	



	A	0	1
0	m <sub>0</sub>	m <sub>2</sub>	
1	m <sub>1</sub>	m <sub>3</sub>	

# K-Map with 3 variables

---

		<b>AB</b>			
		<b>A'B'</b>	<b>A'B</b>	<b>AB</b>	<b>AB'</b>
<b>C</b>	<b>C</b>	<b>A'B'C'</b>	<b>A'BC'</b>	<b>ABC'</b>	<b>AB'C'</b>
	<b>C'</b>	<b>A'B'C</b>	<b>A'BC</b>	<b>ABC</b>	<b>AB'C</b>



		<b>AB</b>			
		00	01	11	10
<b>C</b>	0	0	2	6	4
	1	1	3	7	5

# Kmap With 4 variables

CD		AB		A	
		00	01	11	10
C	00	0	4	12	8
	01	1	5	13	9
	11	3	7	15	11
	10	2	6	14	10

The diagram shows a 4x4 Karnaugh map for 4 variables. The columns are labeled with AB (00, 01, 11, 10) and the rows with CD (00, 01, 11, 10). The cells contain the decimal values 0 through 15. Brackets indicate groupings: 'A' groups the top two columns (11, 10), 'B' groups the bottom two columns (11, 10), 'C' groups the left two rows (00, 01), and 'D' groups the right two rows (11, 10).

# Assigning 1's and 0's in Kmap

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

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

		AB			
		00	01	11	10
CD	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

$$x + x = 1$$

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

	A	0	1
B	0	00	10
	1	01	11

→

$$F = AB' + AB$$

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

	A	A
B		1
B		1

$$F = A$$

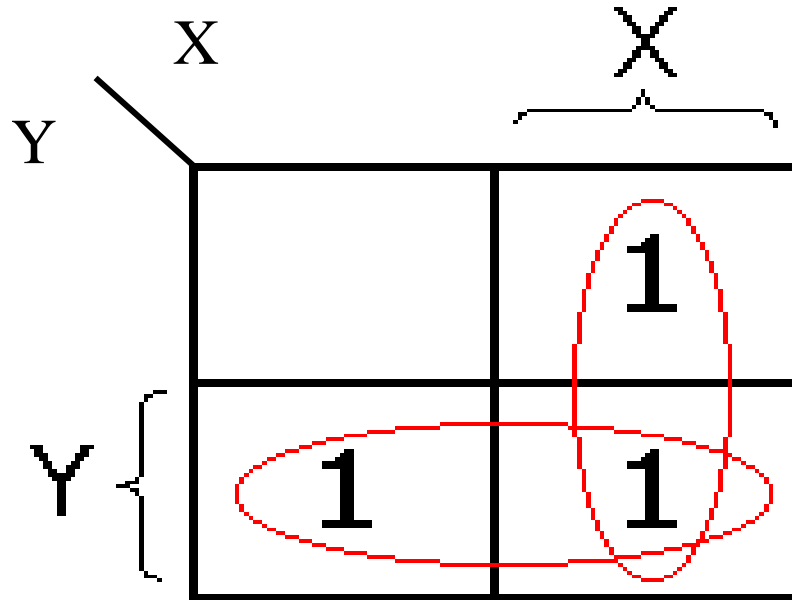
	A	A
B	1	
B	1	

$$F = A'$$

# 2 variable Karnaugh Map

---

$$F = X'Y + XY + XY'$$



$$F = X + Y$$

# 3 Variable Karnaugh Map

**AB**

**C**

$A'B'C'$	$A'BC'$	$ABC'$	$AB'C'$
$A'B'C$	$A'BC$	$ABC$	$AB'C$

**AB**

**C**

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

**B**

$$F = X'YZ' + XYZ + X'YZ$$

$$F = XY'Z' + XYZ'$$

**YZ**

**X**

	00	01	11	10
0			1	1
1			1	

$$F = X'Y + YZ$$

**YZ**

**X**

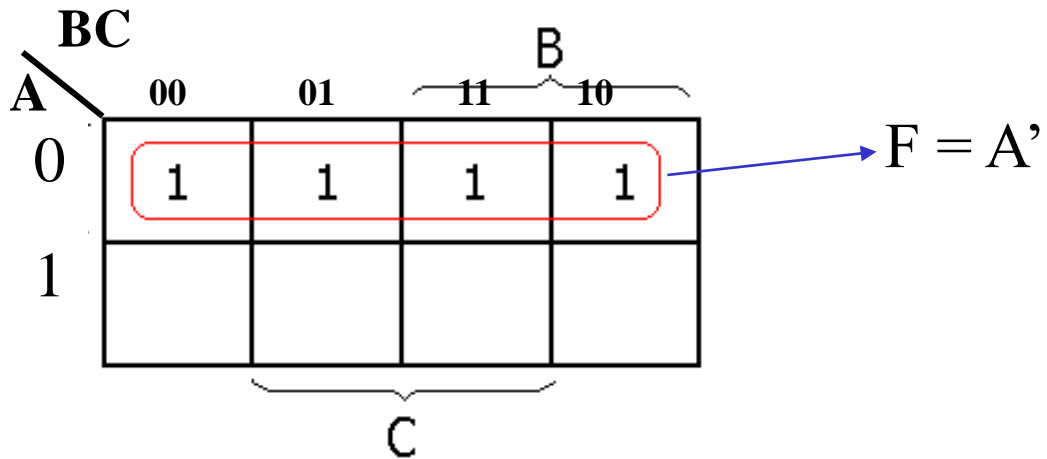
	00	01	11	10
0				
1	1			1

$$F = XZ'$$

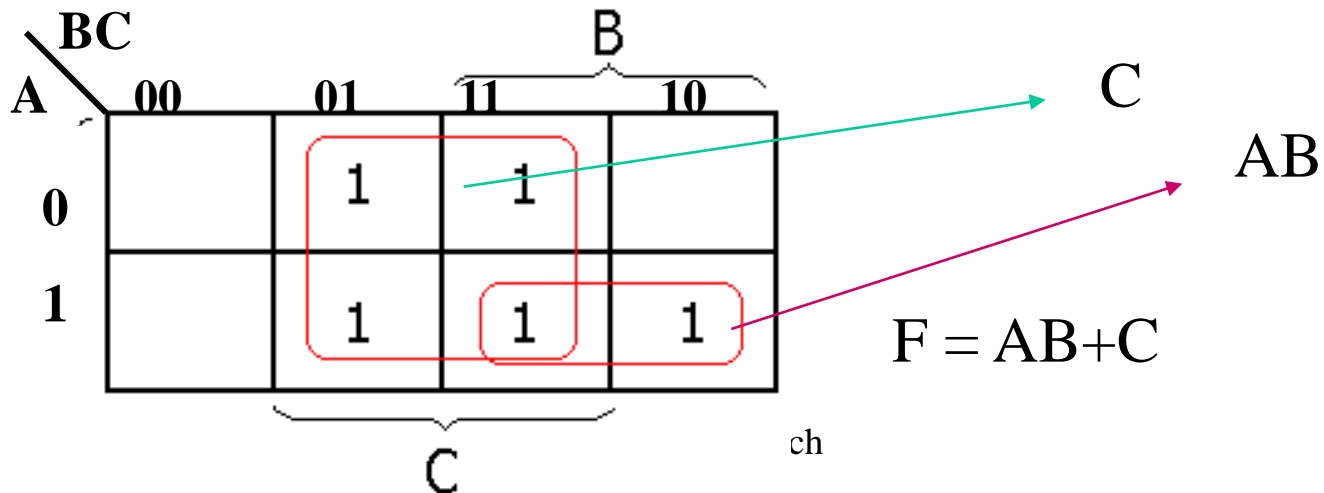
*Wrapping around edges*

# 3 variable Karnaugh Map

$$F(A,B,C) = A'BC' + A'B'C' + A'BC + A'B'C$$

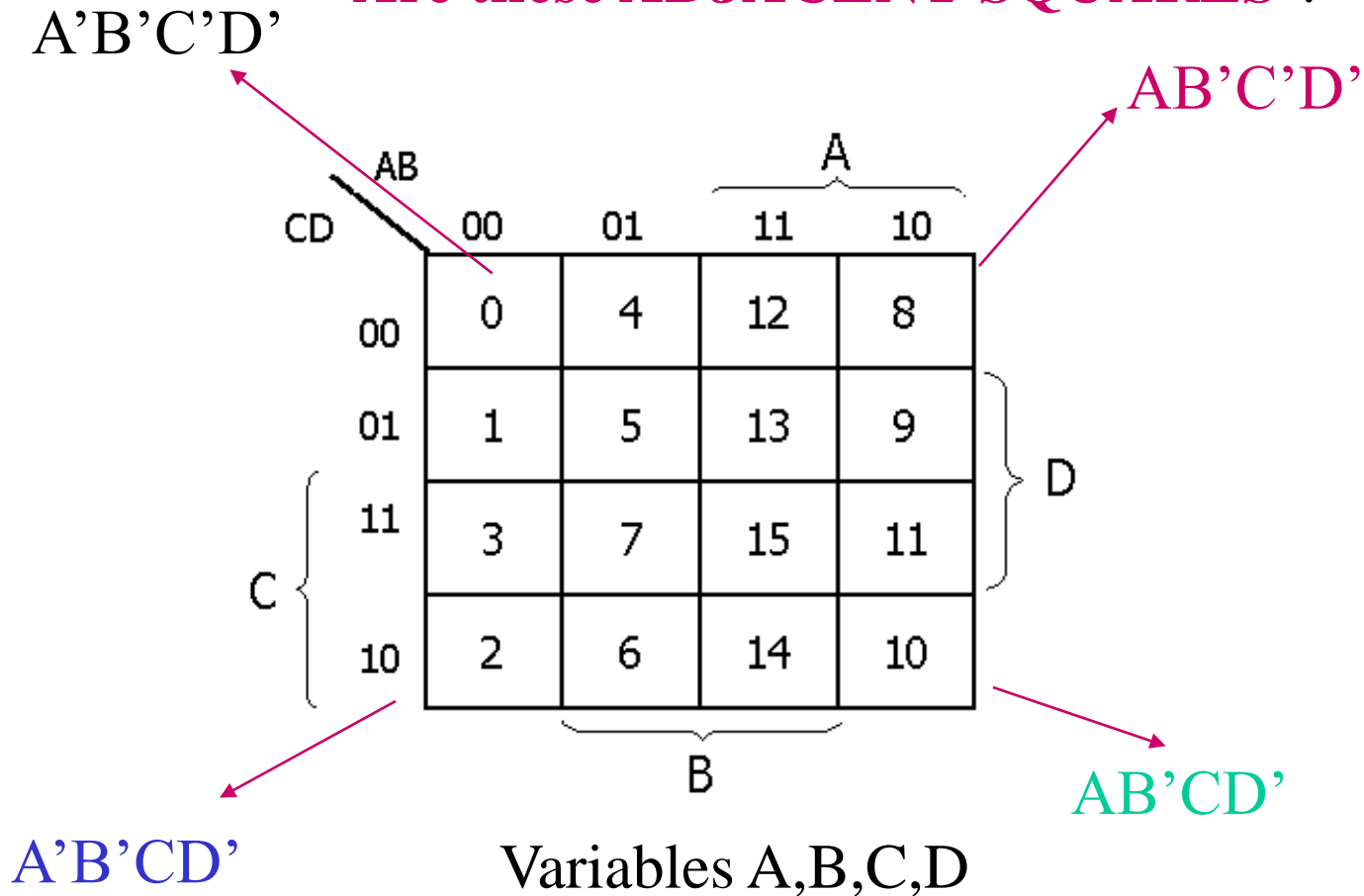


$$F(A,B,C) = A'BC + A'B'C + AB'C + ABC + ABC'$$



# 4 variables K-MAP

Are these ADJACENT SQUARES ?



# 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.

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

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

# 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:  $F(A,B,C,D) = \Sigma m(1,3,7,11,15)$  with  $\Sigma d(0,2,5)$

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

$$F = AD + A'B'$$

- 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

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*

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

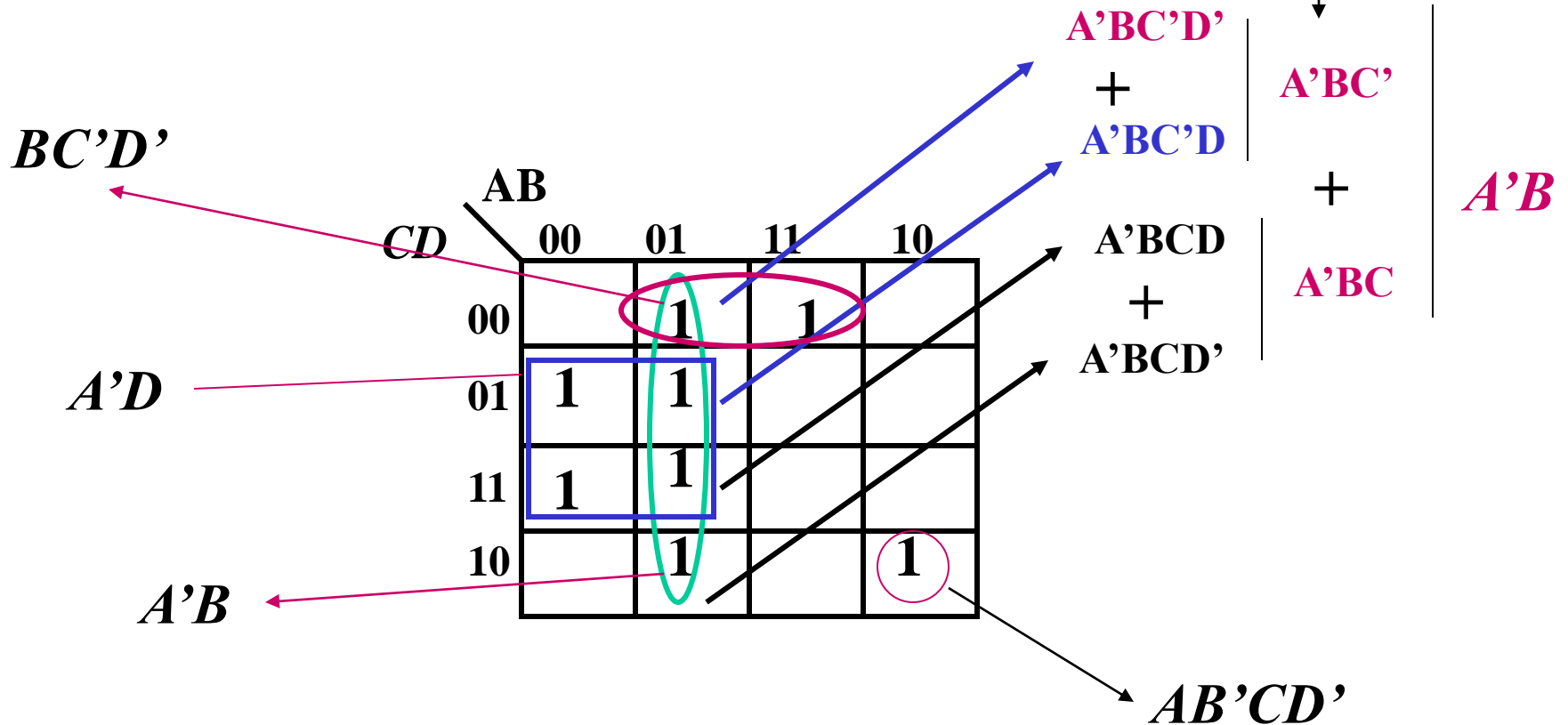
*a) Building K-Map for F*

CD \ AB		A			
		00	01	11	10
C	00	0	4 <b>1</b>	12 <b>1</b>	8
	01	<b>1</b> 1	5 <b>1</b>	13	9
	11	<b>1</b> 3	7 <b>1</b>	15	11
	10	2	6 <b>1</b>	14	<b>1</b> 10
		B			

The K-Map shows the function F(A,B,C,D) with minterms 1, 3, 4, 5, 6, 7, 10, and 12 marked with a pink '1'. The map is organized with AB as columns and CD as rows. Brackets indicate groupings for variables A, B, and C. The minterms are: 1 (0011), 3 (0111), 4 (0101), 5 (0110), 6 (0110), 7 (0101), 10 (1011), and 12 (1101).

**b) Grouping of squares**

$$A'BC'(D+D') = A'BC'$$



**c) Write the Simplified Expression**

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

a) Building K-map from the truth table

<i>A</i>	<i>B</i>	<i>C</i>	<i>D</i>	<i>F</i>
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

		<i>AB</i>			
		00	01	11	10
<i>CD</i>	00	1	1	x	1
	01	0	1	x	0
	11	0	0	x	x
	10	0	0	x	x

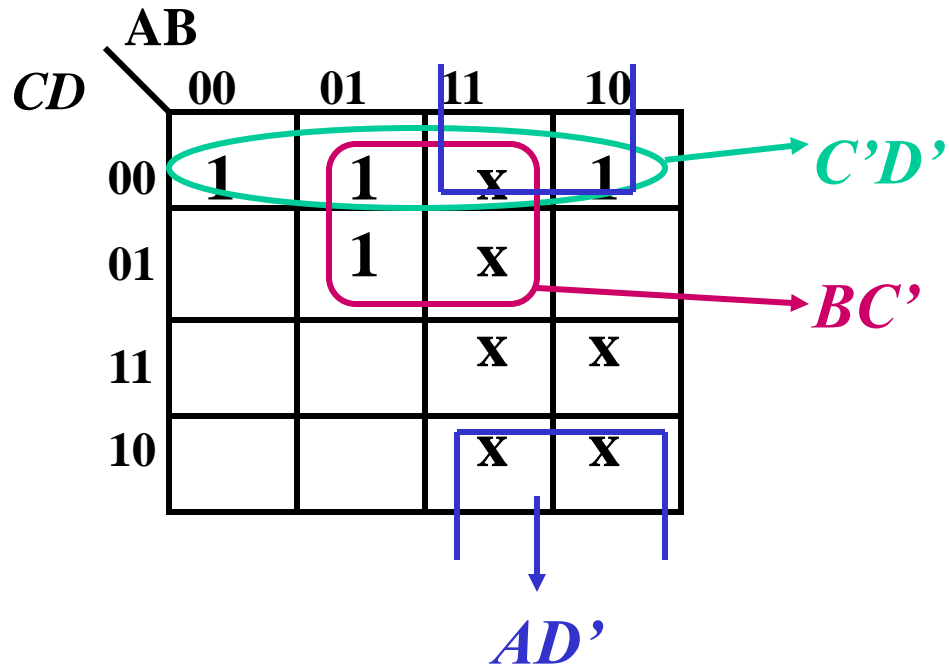
**A** | **B** | **C** | **D** | **F**

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**

		AB			
		00	01	11	10
CD	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