08-Karnaugh Maps

Text: Unit 5

ECEGR/ISSC 201
Digital Operations and Computations
Winter 2011

Overview

- Minimum Switching Forms
- Two and Three Variable K-Maps
- Four Variable K-Maps
- Prime Implicants

Minimizing Switching Functions

- Cost of implementing switching function is dependent on the number of AND, OR and NOT gates (and inputs) used
- Previously discussed methods
  - Difficult to apply a systematic way of minimizing switching functions
  - Difficult to determine when to stop simplfiying
- Karnaugh map (K-map) is a useful tool for switching function minimization

---

K-Maps

- Similar in concept to a truth table
  - Specified for each combination of inputs
  - Example

B \ A	0	1
0	A = 0 B = 0	A = 1 B = 0
1	A = 0 B = 1	A = 1 B = 1

K-Maps

- Example
  - Use K-map to determine an expression for F

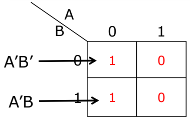
A	B	F
0	0	1
0	1	1
1	0	0
1	1	0

B \ A	0	1
0	1	0
1	1	0

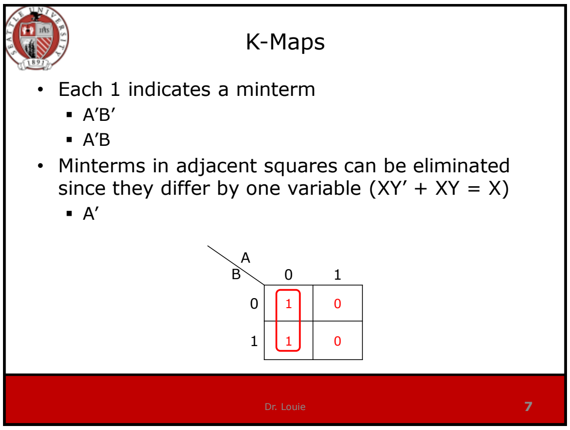
K-Maps

- Example
  - Use K-map to determine an expression for F

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



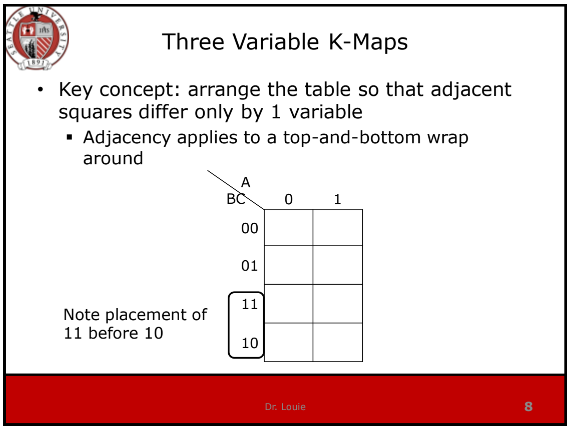
F = A’B’ + A’B = A’



K-Maps

- Each 1 indicates a minterm
  - A'B'
  - A'B
- Minterms in adjacent squares can be eliminated since they differ by one variable (XY' + XY = X)
  - A'

	A	
B	0	1
0	1	0
1	1	0

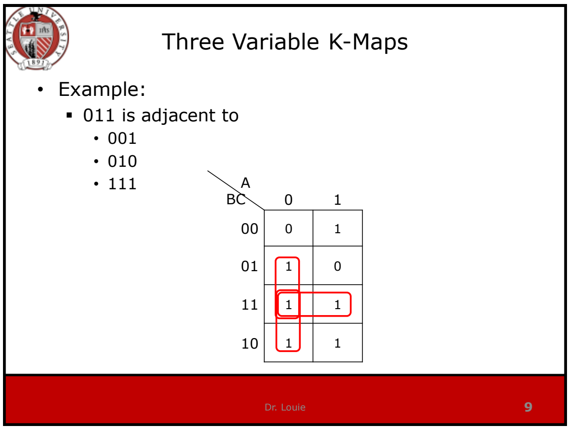


Three Variable K-Maps

- Key concept: arrange the table so that adjacent squares differ only by 1 variable
  - Adjacency applies to a top-and-bottom wrap around

	A	
BC	0	1
00		
01		
11		
10		

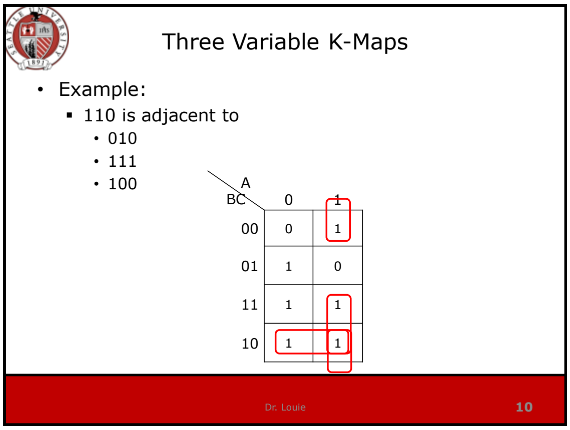
Note placement of 11 before 10



Three Variable K-Maps

- Example:
  - 011 is adjacent to
    - 001
    - 010
    - 111

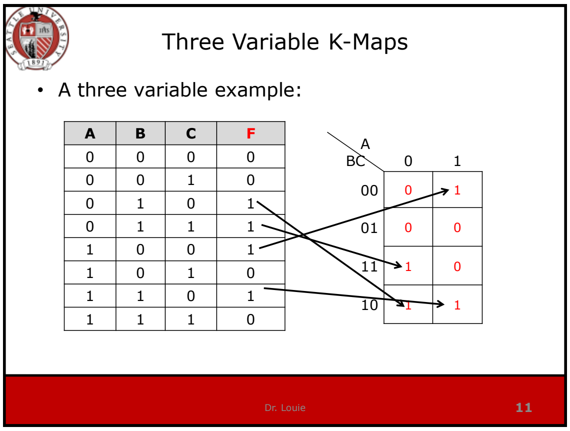
	A	
BC	0	1
00	0	1
01	1	0
11	1	1
10	1	1



Three Variable K-Maps

- Example:
  - 110 is adjacent to
    - 010
    - 111
    - 100

	A	
BC	0	1
00	0	1
01	1	0
11	1	1
10	1	1

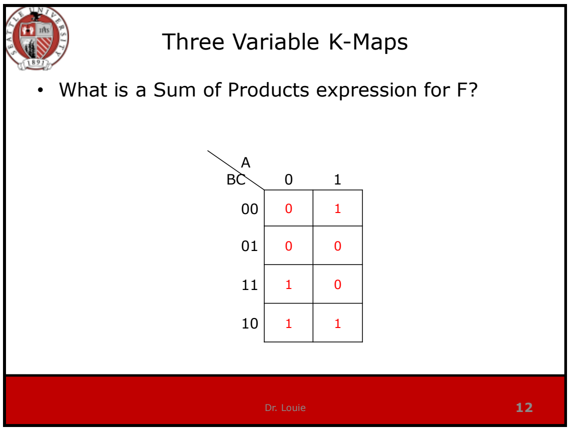


Three Variable K-Maps

- A three variable example:

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

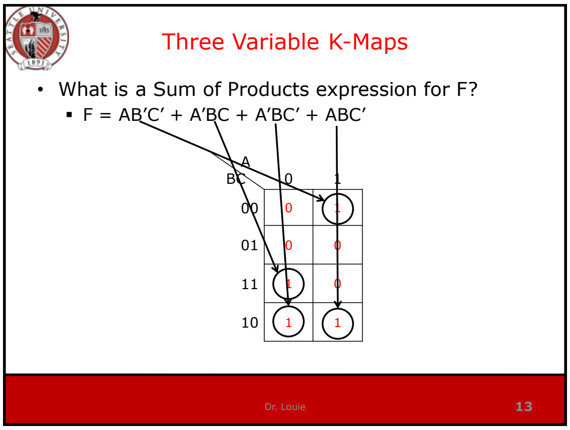
	A	
BC	0	1
00	0	1
01	0	0
11	1	0
10	1	1



Three Variable K-Maps

- What is a Sum of Products expression for F?

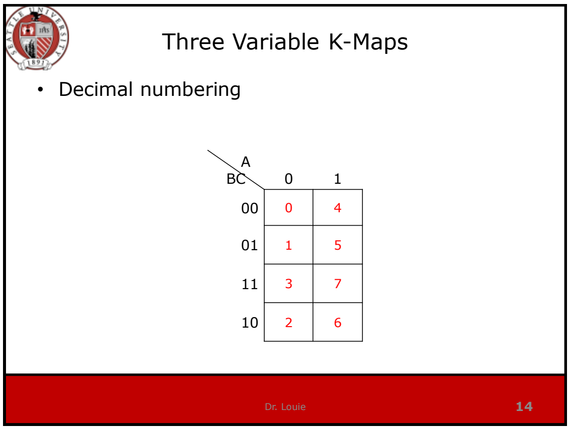
	A	
BC	0	1
00	0	1
01	0	0
11	1	0
10	1	1



Three Variable K-Maps

- What is a Sum of Products expression for F?
  - F = AB'C' + A'BC + A'BC' + ABC'

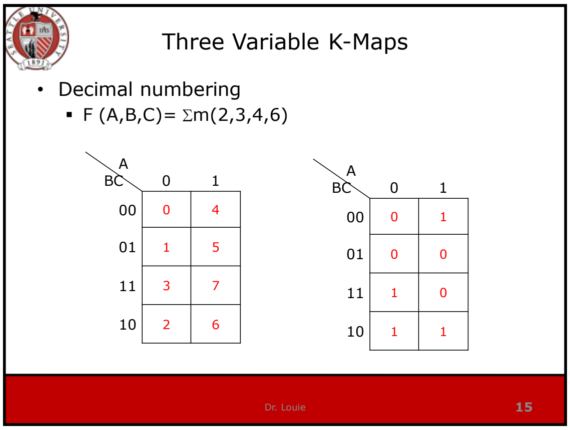
A BC	0	1
00	0	1
01	0	0
11	1	0
10	1	1



Three Variable K-Maps

- Decimal numbering

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

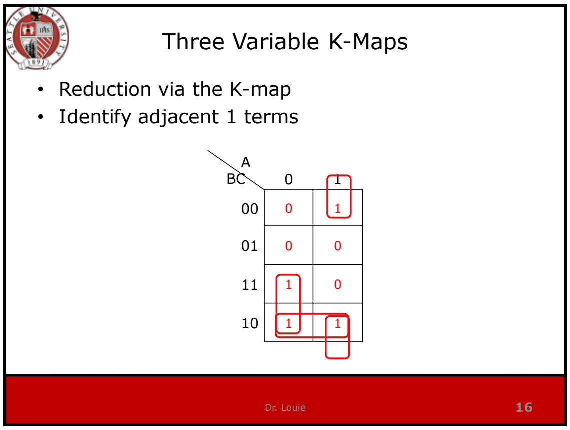


Three Variable K-Maps

- Decimal numbering
  - F (A,B,C)= Σm(2,3,4,6)

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

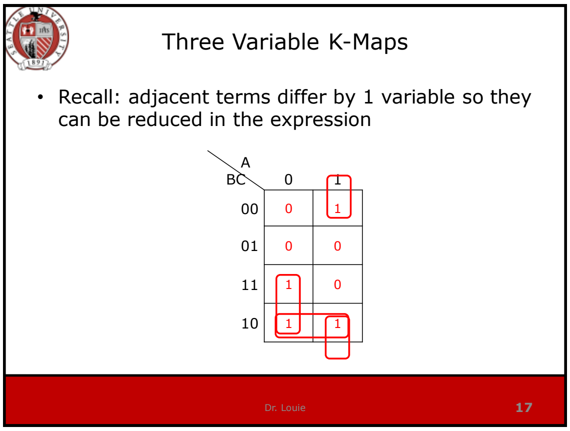
A BC	0	1
00	0	1
01	0	0
11	1	0
10	1	1



Three Variable K-Maps

- Reduction via the K-map
- Identify adjacent 1 terms

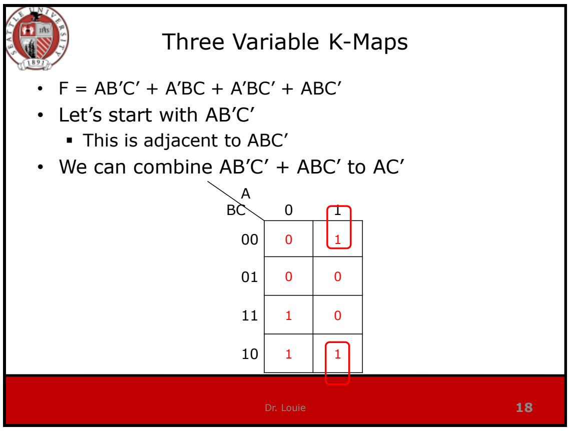
A BC	0	1
00	0	1
01	0	0
11	1	0
10	1	1



Three Variable K-Maps

- Recall: adjacent terms differ by 1 variable so they can be reduced in the expression

A BC	0	1
00	0	1
01	0	0
11	1	0
10	1	1



Three Variable K-Maps

- F = AB'C' + A'BC + A'BC' + ABC'
- Let's start with AB'C'
  - This is adjacent to ABC'
- We can combine AB'C' + ABC' to AC'

A BC	0	1
00	0	1
01	0	0
11	1	0
10	1	1

Three Variable K-Maps

- F = AC′ + A′BC + A′BC′
- Next consider A′BC′
- Adjacent to:
  - ABC′ and A′BC
- We cannot group it with ABC′ since we already grouped ABC′ with AB′C′
- So group A′BC′ with A′BC to be A′B
- F = AC′ + A′B
- A more methodical way of doing this is presented later

BC \ A	0	1
00	0	1
01	0	0
11	1	0
10	1	1

Three Variable K-Maps

- Creating a K-Map directly from a Boolean equation
- For a single variable (F =A′)
- Place a 1 in the four squares with A = 0

BC \ A	0	1
00	1	
01	1	
11	1	
10	1	

Three Variable K-Maps

- Consider F = A′ + B′C
- Place a 1 in the two squares with B = 0 and C = 1

redundant

BC \ A	0	1
00	1	
01	1	1
11	1	
10	1	

Three Variable K-Maps

- Consider F = A′ + B′C + ABC′
- Place a 1 in the single square with A = B =1 and C = 0

BC \ A	0	1
00	1	
01	1	1
11	1	
10	1	1

Three Variable K-Maps

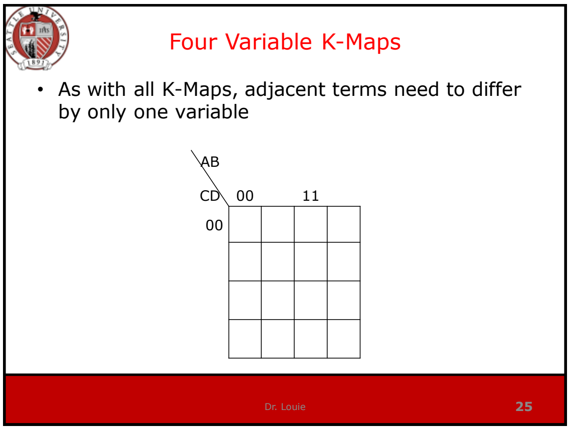
- Place 0s in the remaining squares
- F′ can be found by replacing the 1s with 0s

BC \ A	0	1
00	1	0
01	1	1
11	1	0
10	1	1

Four Variable K-Maps

- As with all K-Maps, adjacent terms need to differ by only one variable
- What should the other rows/columns be?

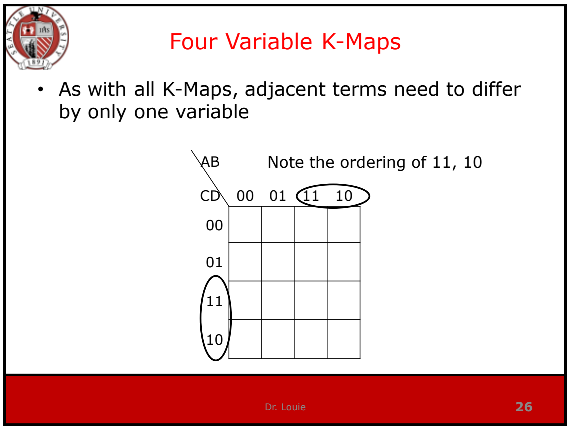
CD \ AB	00			
00				



Four Variable K-Maps

- As with all K-Maps, adjacent terms need to differ by only one variable

AB CD	00		11	
00				

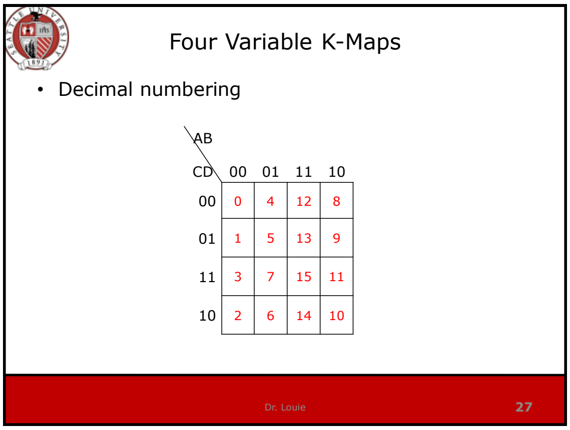


Four Variable K-Maps

- As with all K-Maps, adjacent terms need to differ by only one variable

Note the ordering of 11, 10

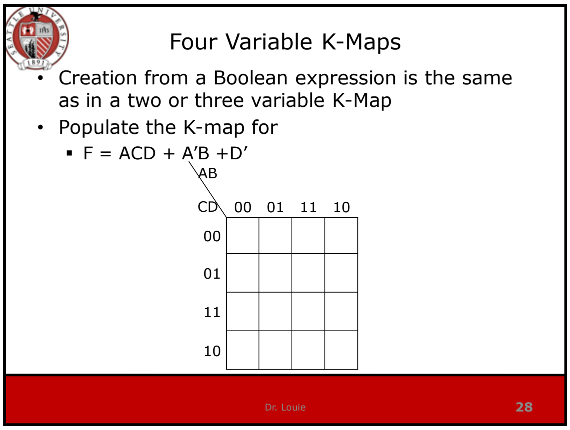
AB CD	00	01	11	10
00				
01				
11				
10				



Four Variable K-Maps

- Decimal numbering

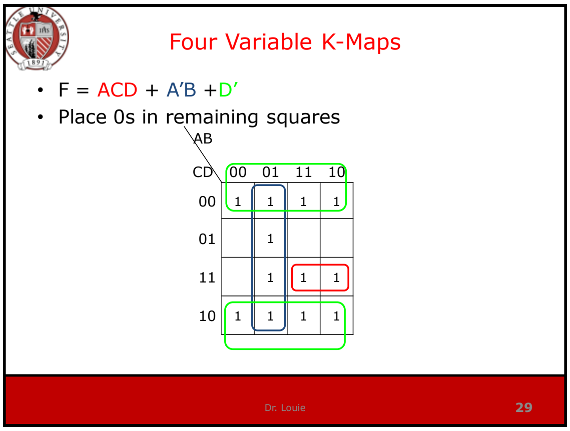
AB CD	00	01	11	10
00	0	4	12	8
01	1	5	13	9
11	3	7	15	11
10	2	6	14	10



Four Variable K-Maps

- Creation from a Boolean expression is the same as in a two or three variable K-Map
- Populate the K-map for
  - F = ACD + A'B +D'

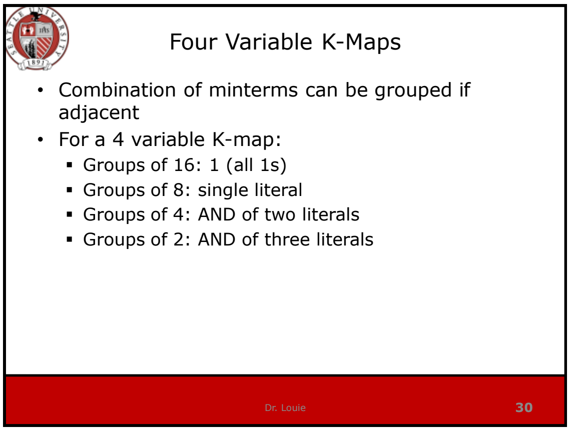
AB CD	00	01	11	10
00				
01				
11				
10				



Four Variable K-Maps

- F = ACD + A'B +D'
- Place 0s in remaining squares

AB CD	00	01	11	10
00	1	1	1	1
01		1		
11		1	1	1
10	1	1	1	1



Four Variable K-Maps

- Combination of minterms can be grouped if adjacent
- For a 4 variable K-map:
  - Groups of 16: 1 (all 1s)
  - Groups of 8: single literal
  - Groups of 4: AND of two literals
  - Groups of 2: AND of three literals

Four Variable K-Maps

- Write the reduced Boolean Expression for:

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

Four Variable K-Maps

- Write the reduced Boolean Expression for:
- F = AB’CD’ +

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

Four Variable K-Maps

- Write the reduced Boolean Expression for:
- F = AB’CD’ + A’B’D

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

Four Variable K-Maps

- Write the reduced Boolean Expression for:
- F = AB’CD’ + A’B’D + BC’

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

Example

- Write the reduced Boolean Expression for:
- F =

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

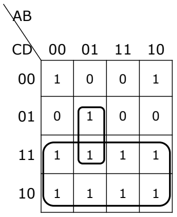
Example

- Write the reduced Boolean Expression for:
- F = C

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

Example

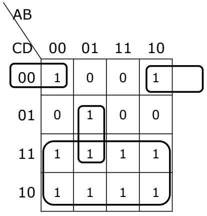
- Write the reduced Boolean Expression for:
- F = C + A’BD



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

Example

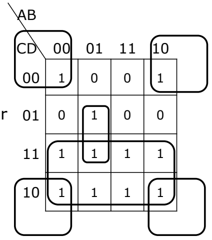
- Write the reduced Boolean Expression for:
- F = C + A’BD + B’C’D’



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

Example

- Write the reduced Boolean Expression for:
- F = C + A’BD + B’D’



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

We can reduce further if we use the corners

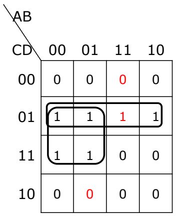
K-Maps with Don’t Cares

- X: Don’t Care terms
- Select value for X in order to simplify the expression

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

K-Maps with Don’t Cares

- F = A’D + C’D



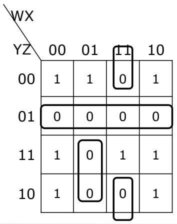
AB CD	00	01	11	10
00	0	0	0	0
01	1	1	1	1
11	1	1	0	0
10	0	0	0	0

PoS Using K-Maps

- Procedure for finding PoS from a K-Map
  - Find F’
  - Complement the SoP of F’ to find F in PoS form

PoS Using K-Maps

- Find F′ by grouping 0s of the K-Map
  - F′ = Y′Z + WXZ′ + W′XY



YZ \ WX	00	01	11	10
00	1	1	0	1
01	0	0	0	0
11	1	0	1	1
10	1	0	0	1

PoS Using K-Maps

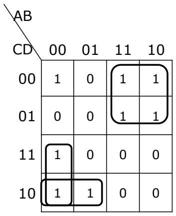
- Complement the SoP of F′ to find F in PoS form
  - F′ = Y′Z + WXZ′ + W′XY
  - (F′)′ = (Y′Z + WXZ′ + W′XY)′
    - Use DeMorgan’s Law
  - F = (Y + Z′)(W′ + X′ + Z)(W + X′ + Y′)

Prime Implicants

- Implicant: any single 1 or group of 1s that can be combined together
- Prime Implicant: a product term that cannot be combined with another term to eliminate a variable
- An SoP expression cannot be minimal if it contains a term that is not a prime implicant
- Need to find all prime implicants to make a minimal SoP expression

Prime Implicants

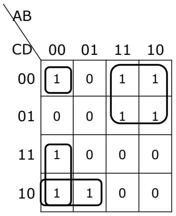
- Prime Implicants: AC′, A′B′C, A′CD′



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

Prime Implicants

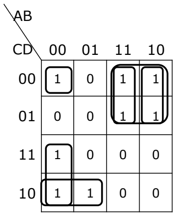
- A′B′C′D′ is not a prime implicant because it can be grouped with A′B′CD′ or AB′C′D′
  - but A′B′D′ and B′C′D′ are prime implicants



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

Prime Implicants

- ABC′ is not a prime implicant because it can be combined with AC′ to eliminate AB′C′



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

Prime Implicants

- A single 1 can only be a prime implicant if there are no adjacent 1s
- Two adjacent 1s can only be a prime implicant if there are not part of a group of four 1s
- Four adjacent 1s can only be a prime implicant if they are not part of a group of eight 1s
- etc

Prime Implicants

- Goal: find the minimum number of prime implicants that cover all 1s on a K-map

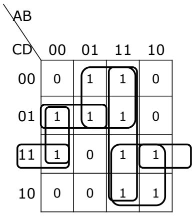
Prime Implicants

- Find all the prime implicants

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

Prime Implicants

- Find all the prime implicants



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

Prime Implicants

- Six prime implicants: BC′, AB, A′B′D, AC, A′C′D, B′CD
- F = BC′ + AB + A′B′D + AC + A′C′D + B′CD is valid, but not minimal
- How can we cover all 1s using the least number of prime implicants?

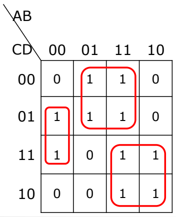
Prime Implicants

- How can we cover all 1s using the least number of prime implicants?

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

Prime Implicants

- F = A’B’D + BC’ + AC



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

Prime Implicants

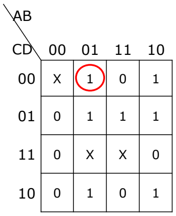
- Need to develop a systematic method of selecting the minimal amount of prime implicants
- *Essential Prime Implicants*: a minterm that is covered by only one prime implicant
  - It can only be covered by one loop

Prime Implicants

1. Choose a 1 that has not been covered
2. Find all adjacent 1s and Xs
3. Are the chosen 1 and its adjacent 1s and Xs covered by a single term?
   - If yes, the term is essential so loop it, else do not loop it
4. All uncovered 1s checked?
   - If no, go to step 1, else
5. Find minimum set of prime implicants that cover the remaining 1s on the K-map
6. End

Prime Implicants

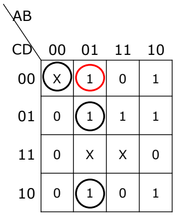
- “Choose a 1 that has not been covered”
- Since we just started, select any 1
- Better to be methodical, so select A’BC’D’



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

Prime Implicants

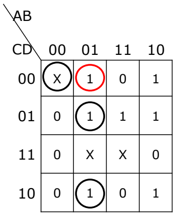
- “Find all adjacent 1s and Xs”



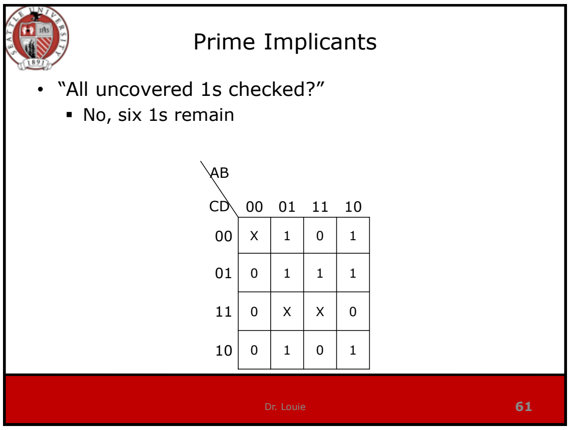
CD \ AB	00	01	11	10
00	X	1	0	1
01	0	1	1	1
11	0	X	X	0
10	0	1	0	1

Prime Implicants

- “Are the chosen 1 and its adjacent 1s and Xs covered by a single term?”
  - No, A’C’ covers a 0 and A’B does not cover A’B’C’D’
  - Not essential



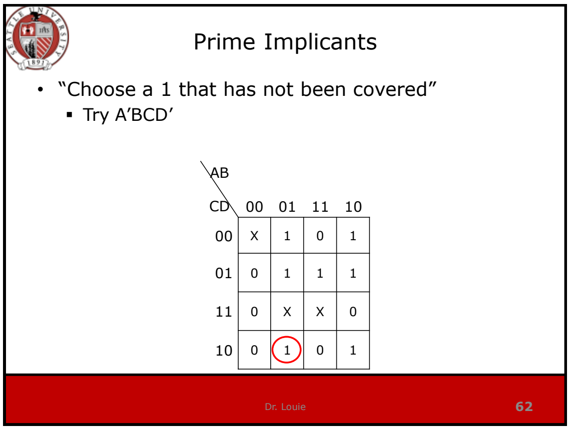
CD \ AB	00	01	11	10
00	X	1	0	1
01	0	1	1	1
11	0	X	X	0
10	0	1	0	1



Prime Implicants

- “All uncovered 1s checked?”
  - No, six 1s remain

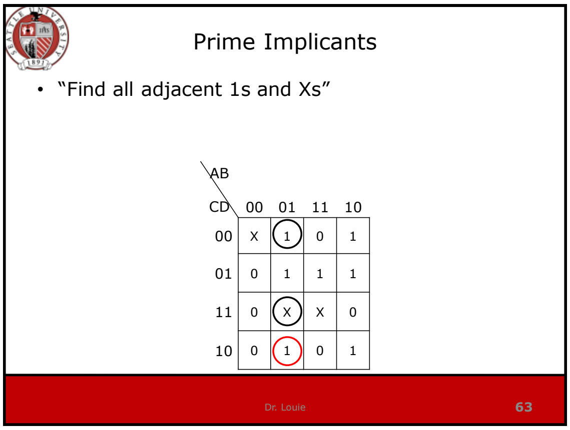
AB CD	00	01	11	10
00	X	1	0	1
01	0	1	1	1
11	0	X	X	0
10	0	1	0	1



Prime Implicants

- “Choose a 1 that has not been covered”
  - Try A’BCD’

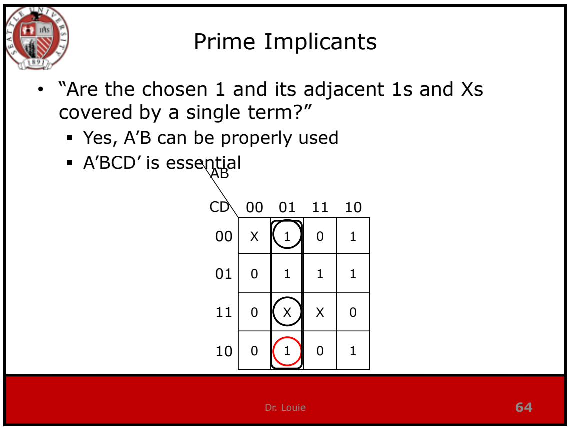
AB CD	00	01	11	10
00	X	1	0	1
01	0	1	1	1
11	0	X	X	0
10	0	1	0	1



Prime Implicants

- “Find all adjacent 1s and Xs”

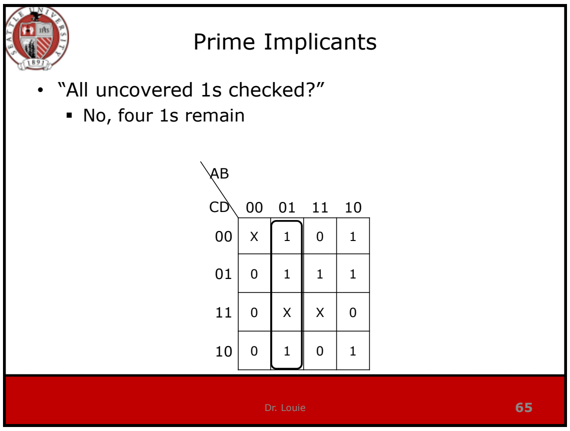
AB CD	00	01	11	10
00	X	1	0	1
01	0	1	1	1
11	0	X	X	0
10	0	1	0	1



Prime Implicants

- “Are the chosen 1 and its adjacent 1s and Xs covered by a single term?”
  - Yes, A’B can be properly used
  - A’BCD’ is essential

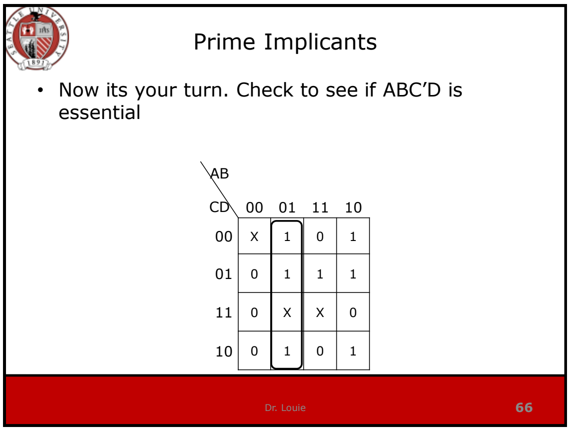
AB CD	00	01	11	10
00	X	1	0	1
01	0	1	1	1
11	0	X	X	0
10	0	1	0	1



Prime Implicants

- “All uncovered 1s checked?”
  - No, four 1s remain

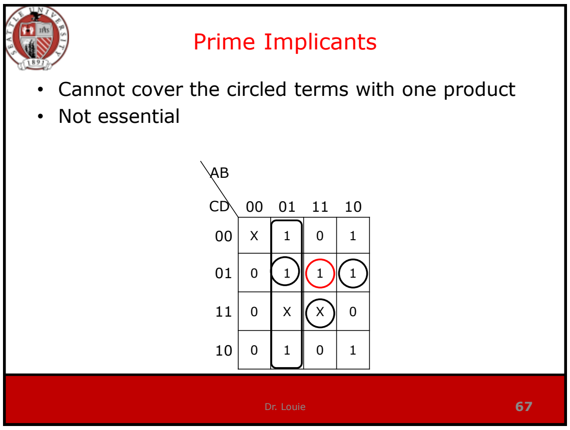
AB CD	00	01	11	10
00	X	1	0	1
01	0	1	1	1
11	0	X	X	0
10	0	1	0	1



Prime Implicants

- Now its your turn. Check to see if ABC’D is essential

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

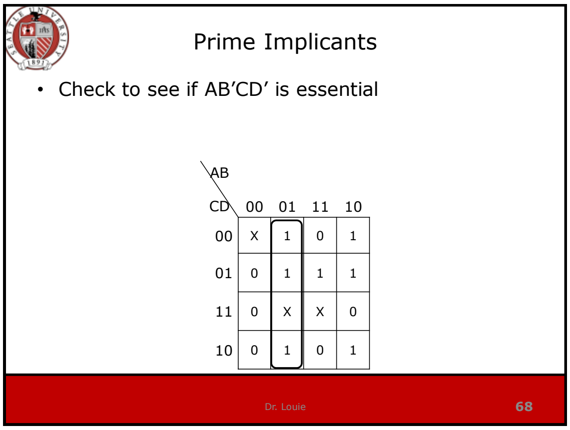


Prime Implicants

- Cannot cover the circled terms with one product
- Not essential

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

Dr. Louie 67

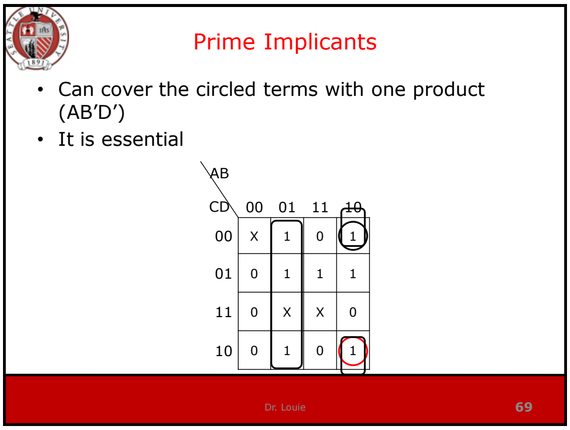


Prime Implicants

- Check to see if AB’CD’ is essential

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

Dr. Louie 68

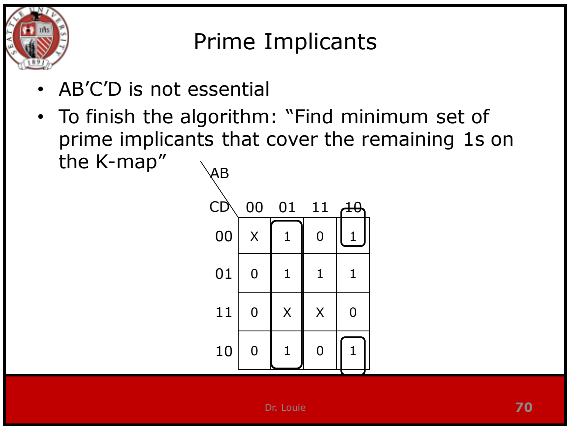


Prime Implicants

- Can cover the circled terms with one product (AB’D’)
- It is essential

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

Dr. Louie 69

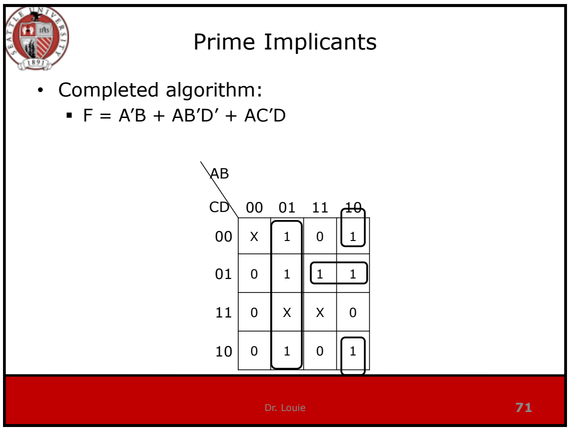


Prime Implicants

- AB’C’D is not essential
- To finish the algorithm: “Find minimum set of prime implicants that cover the remaining 1s on the K-map”

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

Dr. Louie 70

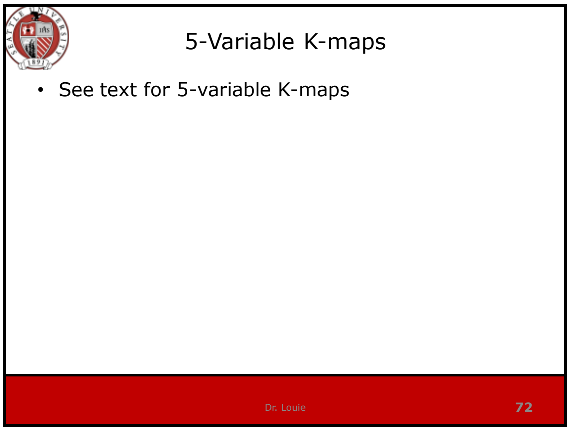


Prime Implicants

- Completed algorithm:
  - F = A’B + AB’D’ + AC’D

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

Dr. Louie 71



5-Variable K-maps

- See text for 5-variable K-maps

Dr. Louie 72