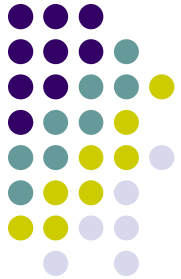
LOGIC GATES



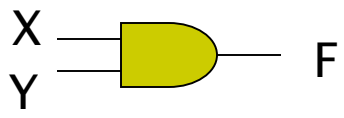
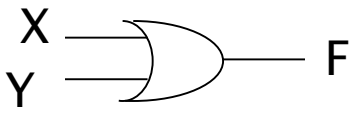
BASIC GATES

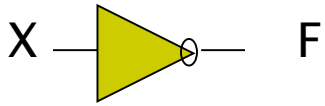
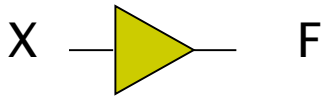
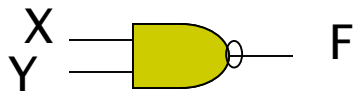
1.NOT

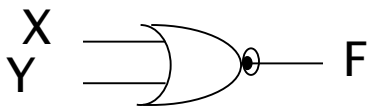
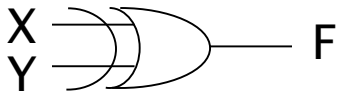
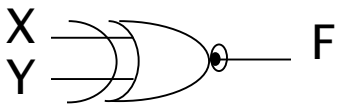
2.AND

3.OR

DIGITAL LOGIC GATES

NAME	GRAPHIC SYMBOL	ALGEBRIC FUNCTION	TRUTH TABLE
AND		F=XY	X Y F 0 0 0 0 1 0 1 0 0 1 1 1
OR		F=X+Y	X Y F 0 0 0 0 1 1 1 0 1 1 1 1

NAME	GRAPHIC SYMBOL	ALGEBRIC FUNCTION	TRUTH TABLE
Inverter		F=X’	X F 0 1 1 0
Buffer		F=X	X F 0 0 1 1
NAND		F=(XY)’	X Y F 0 0 1 0 1 1 1 0 1 1 1 0

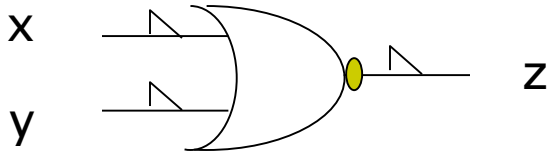
NAME	GRAPHIC SYMBOL	ALGEBRIC FUNCTION	TRUTH TABLE
NOR		F=(X+Y)’	X Y F 0 0 1 0 1 0 1 0 0 1 1 0
Exclusive-OR (XOR)		F=XY’+X’Y = X ⊕ Y	X Y F 0 0 0 0 1 1 1 0 1 1 1 0
Exclusive-NOR or Equivalence		F=XY+X’Y’ =X⊙ Y	X Y F 0 0 1 0 1 0 1 0 0 1 1 1

DEMONSTRATION OF POSITIVE AND NEGATIVE LOGIC

X	y	z
1	1	0
1	0	1
0	1	1
0	0	1

Truth table for negative logic

L=1 H=0

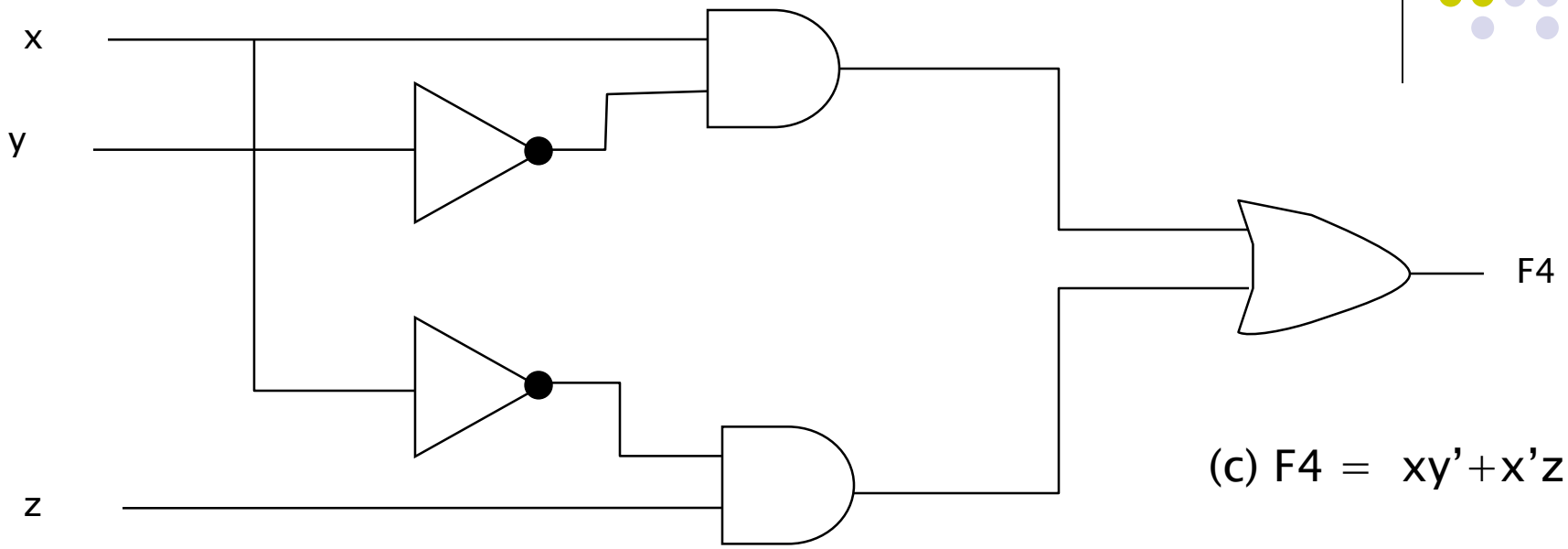


Graphic symbol for negative logic NOR gate

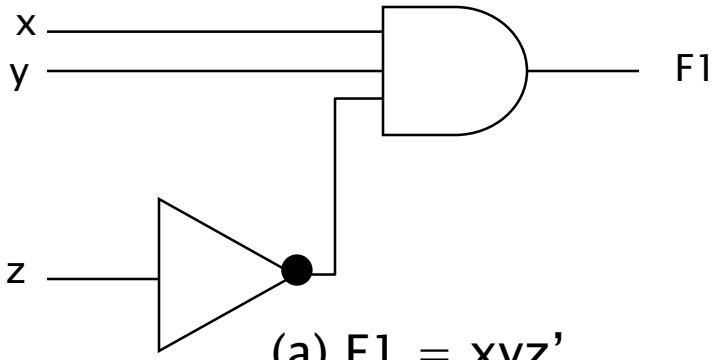
Same gate can function

+ive logic NAND or -ive logic NOR

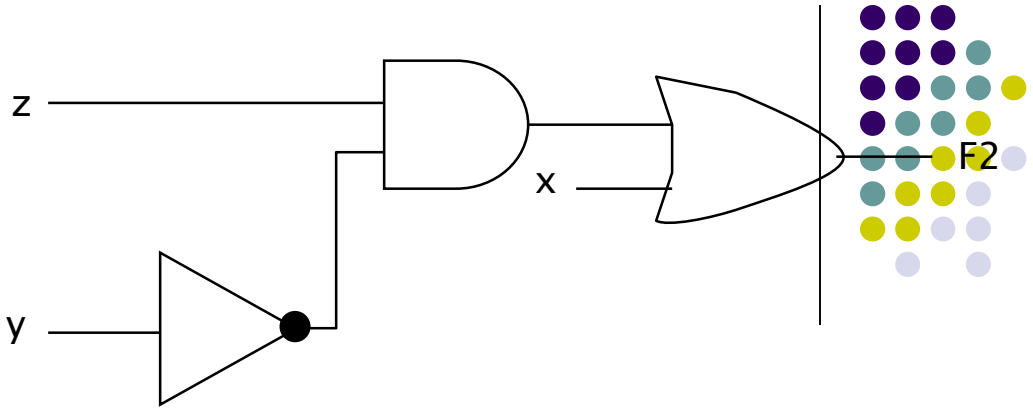
+ive logic NOR or -ive logic NAND



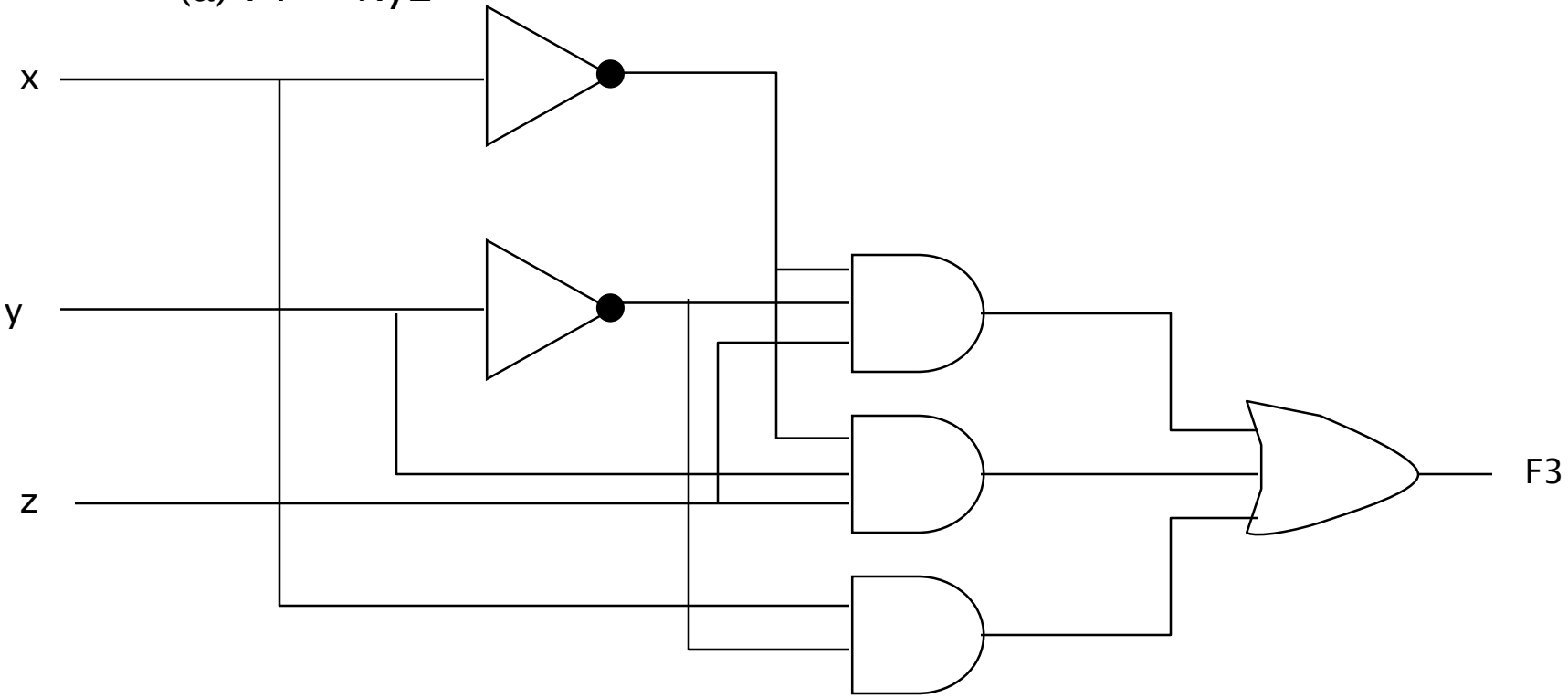
Implementation of Boolean Function with GATES



(a) F1 = xyz'



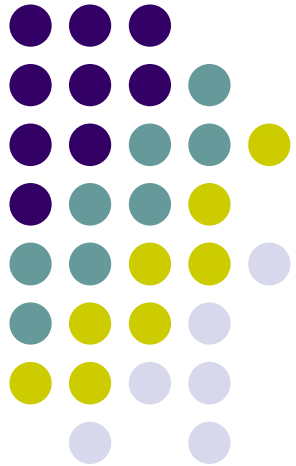
(b) F2 = x+y'z



(c) F3 = x'y'z+x'yz+xy'

Minimization of Switching Functions

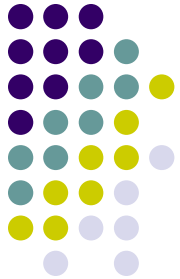
Unit-II



Topics:

- Karnaugh Map Method
- Prime Implicants
- Don’t Care Combinations
- Minimal SOP and POS forms
- Quine Mc Cluskey Tabular Method
- Prime Implicant Chart
- Simplification Rules

Simplifying Switching Functions



- **SOP and POS expressions**

  ==> 2-level circuits

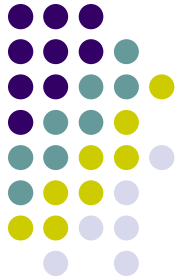
  Minimum SOP/POS expression:

  - **Minimize the number of literals**
  - **Minimum number of terms**

  How?

  - **Algebraically: I.e. using the axioms and theorems of Boolean algebra.**
  - **Karnaugh Map**
  - **McCluskey Method**

Simplifying Switching Functions: K-Map



Simplifying Theorem:

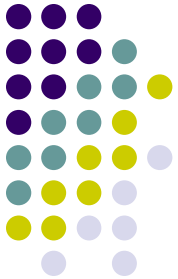
XY + X'Y = Y

<u>Definition:</u> Logical Adjacency

- **Two terms are logically adjacent if they differ in only one literal: the literal is complemented in one term and non-complemented in the other.**

- **Two Logically adjacent terms can be combined into one term consisting of only the common literals**

Karnaugh-Map (K-Map)



2-dimensional representation of a truth table.

Logically adjacent terms are physically adjacent in the map.

<u>2-Variable Functions:</u> **F(X,Y) = XY + X'Y**

	X Y	F(X,Y)
m0	0 0	
m1	0 1	
m2	1 0	
m3	1 1	

2-Variable K-Map

Y \ X	0	1
0	00	10
1	01	11

Y \ X	0	1
0	X'Y'	XY'
1	X'Y	XY

Y \ X	0	1
0	m0	m2
1	m1	m3

Note:

- **Logically adjacent cells are physically adjacent in the k-map**
- **Each cells has two adjacent cells**

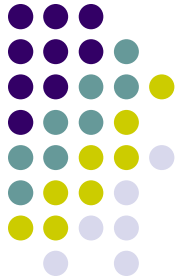
Function Minimization Using K-Maps

1. Each square (minterm) in a k-map of 2 variables has 2 logically adjacent squares, each square in a 3-variable k-map has 3 adjacent squares, etc.

2. Combine only the minterms for which the function is 1.

3. When combining terms on a k-map, group adjacent squares in groups of powers of 2 (I.e. 2, 4, 8, etc.). Grouping two squares eliminates one variables, grouping 4 squares eliminates 2 variables, etc.
   - Can't combine a group of 3 minterms

Function Minimization Using K-Maps



4. Group as many squares together as possible; the larger the group is the fewer the number of literals in the resulting product term

5. Select as few groups as possible to cover all the minterms of the functions. A minterm is covered if it is included in at least one group. Each minterm may be covered as many times as it is needed; however, it must be covered at least once.

6. In combining squares on the map, always begin with those squares for which there are the fewest number of adjacent squares (the “loneliest" squares on the map).

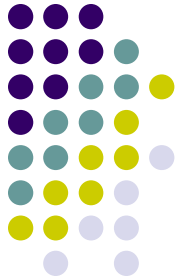
Definitions

- **Implicant: a product term that could be used to cover one or more minterms**

- **Prime Implicant: A product term obtained by combining the maximum number of adjacent squares in the map.**

- **Essential Prime Implicant: A prime implicant that covers at least one minterm that is not covered by any other prime implicant.**
  - **All essential prime implicants must be included in the final minimal expression.**

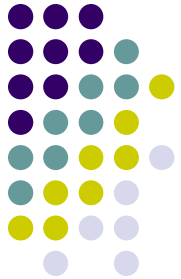
Definitions (Cont.)



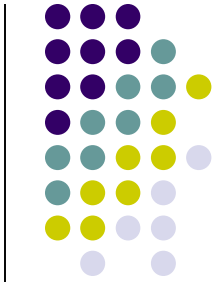
- **Cover of function:** **is a set of prime implicants for which each minterm of the function is covered by at least one prime implicant.**

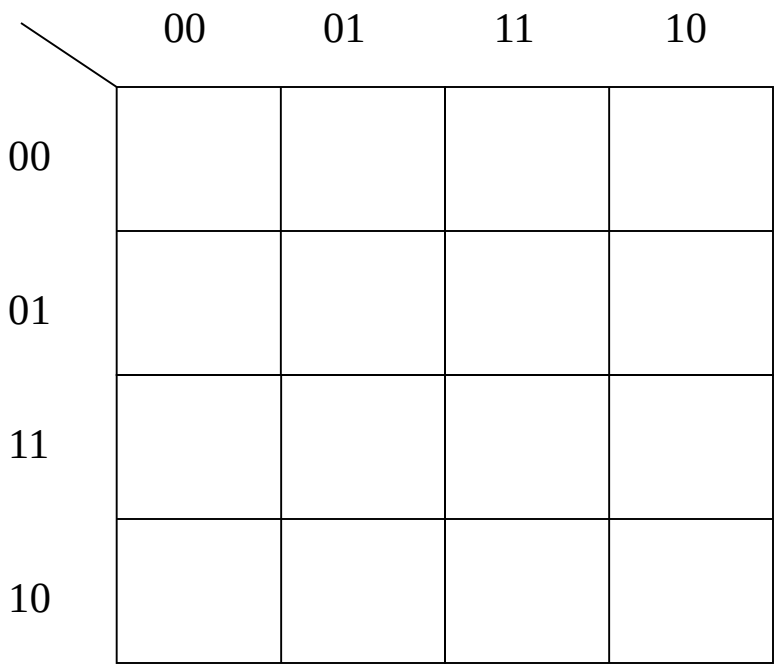
  - **All essential prime implicants must be included in the cover of a function.**

Algorithm for Deriving the Minimal SOP



- **1. Circle all prime implicants on the k-map**
- **2. Identify and select all essential prime implicants**
- **3. Select a minimum subset of the remaining prime implicants to cover those minterms not covered by the essential prime implicants.**





Four Variable K-map

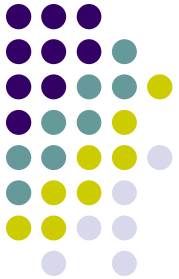
ABC / DE	000	001	011	010	100	101	111	110
00	m0	m4	m12	m8	m16	m20	m28	m24
01	m1	m5	m13	m9	m17	m21	m29	m25
11	m3	m7	m15	m11	m19	m23	m31	m27
10	m2	m6	m14	m10	m18	m22	m30	m26

Five Variable K-map

More Examples

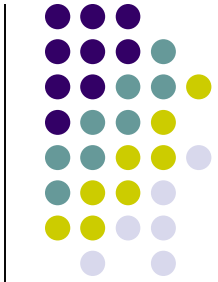
Find min. SOP and POS expression for each of the following functions

1. F(A,B,C,D) = Σ m(2,3,5,7,10,11,13,14,15)
2. G(W,X,Y,Z)= Π M(1,3,4,5,7)
3. H(A,B,C,D) = **Σm(1,3,4,7,11)+d(5,12,13,14,15)**
4. **F2(A,B,C,D)= Σ m(1,2,7,12,15)+d(5,9,10,11,13)**
5. **F3(A,B,C,D,E) = Σ m(0,1,2,4,5,6,13,15,16,18,22,24,26,29)**



Tabulation Method

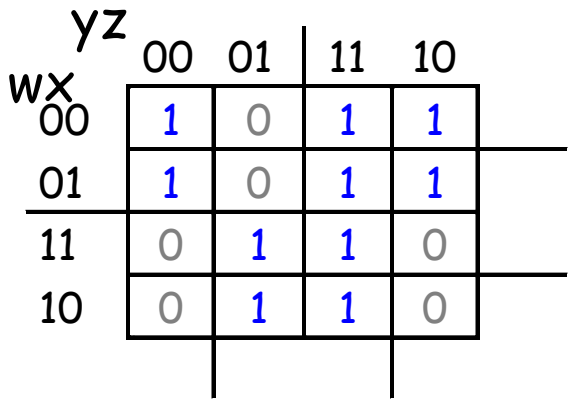
Tabulation Method



- Map method is a trial-and-error procedure
- Tabulation method performs thorough search
- It starts with SOM and consists of 2 steps:
  - PIs generation
    - group minterms by number of 1s
    - compare minterms & find pairs that differ in 1 variable
    - generate subcubes
    - repeat the above 3 steps to generate subcubes until no more subcubes can be generated
  - Minimal cover generation
    - find EPIs through a selection table
    - find minimal cover through the POS of PIs

Example: simplify w'y'z' + wz + xyz + w'y

- K-map representation:



- PIs generation:
  - 0-subcubes

Group ID	Subcube minterms	Subcube expression w	x	y	z	Subcube covered
G_0	(0)	0	0	0	0	yes
G_1	(2)	0	0	1	0	yes
	(4)	0	1	0	0	yes
G_2	(3)	0	0	1	1	yes
	(6)	0	1	1	0	yes
	(9)	1	0	0	1	yes
G_3	(7)	0	1	1	1	yes
	(11)	1	0	1	1	yes
	(13)	1	1	0	1	yes
G_4	(15)	1	1	1	1	yes

Example: simplify w'y'z' + wz + xyz + w'y (cont.)

- 1-subcubes

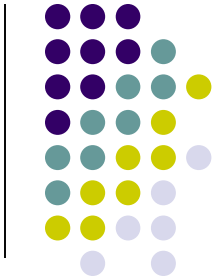
Group ID	Subcube minterms	Subcube expression				Subcube covered
		w	x	y	z	
G_0	(0)	0	0	0	0	yes
G_1	(2)	0	0	1	0	yes
	(4)	0	1	0	0	yes
G_2	(3)	0	0	1	1	yes
	(6)	0	1	1	0	yes
	(9)	1	0	0	1	yes
G_3	(7)	0	1	1	1	yes
	(11)	1	0	1	1	yes
	(13)	1	1	0	1	yes
G_4	(15)	1	1	1	1	yes

Group ID	Subcube minterms	Subcube expression				Subcube covered
		w	x	y	z	
G_0	(0,2)	0	0	-	0	yes
	(0,4)	0	-	0	0	yes
G_1	(2,3)	0	0	1	-	yes
	(2,6)	0	-	1	0	yes
	(4,6)	0	1	-	0	yes
G_2	(3,7)	0	-	1	1	yes
	(3,11)	-	0	1	1	yes
	(6,7)	0	1	1	-	yes
	(9,11)	1	0	-	1	yes
	(9,13)	1	-	0	1	yes
G_3	(7,15)	-	1	1	1	yes
	(11,15)	1	-	1	1	yes
	(13,15)	1	1	-	1	yes

- 2-subcubes

Group ID	Subcube minterms	Subcube expression				Subcube covered
		w	x	y	z	
G_0	(0,2,4,6)	0	-	-	0	no
G_1	(2,3,6,7)	0	-	1	-	no
G_2	(3,7,11,15)	-	-	1	1	no
	(9,11,13,15)	1	-	-	1	no

Example: simplify w'y'z' + wz + xyz + w'y (cont.)



- Minimal cover generation:
  - EPIs selection

Prime implicant name	Prime implicant expression	Implicant minterms	Function minterms									
			0	2	3	4	6	7	9	11	13	15
P_1	$w'z'$	(0,2,4,6)	⊗	×		⊗	×					
P_2	$w'y$	(2,3,6,7)		×	×		×	×				
P_3	yz	(3,7,11,15)			×			×		×		×
P_4	wz	(9,11,13,15)							⊗	×	⊗	×
	EPI covered minterms:		0	2		4	6		9	11	13	15
	Not covered minterms:				3			7				

- PI list: w'z', w'y, yz, wz
  - EPI list: w'z', wz
  - POS: (P2 + P3)(P2 + P3) = P2 + P3
  - Minimal cover expressions:
    - F1 = w'z' + wz + w'y
    - F2 = w'z' + wz + yz

Another example

- K-map representation:

wx \ yz	00	01	11	10
00	0	0	0	1
01	0	0	1	1
11	0	1	1	0
10	1	1	0	0

- PIs generation:
  - 0-subcubes, 1-subcubes

Group ID	Subcube minterms	Subcube expression w	x	y	z	Subcube covered by 1-subcube
G_1	(2)	0	0	1	0	yes
	(8)	1	0	0	0	yes
G_2	(6)	0	1	1	0	yes
	(9)	1	0	0	1	yes
G_3	(7)	0	1	1	1	yes
	(13)	1	1	0	1	yes
G_4	(15)	1	1	1	1	yes

Group ID	Subcube minterms	Subcube expression w	x	y	z	Subcube covered by 2-subcube
G_1	(2,6)	0	-	1	0	no
	(8,9)	-	0	0	1	no
G_2	(6,7)	0	1	1	-	no
	(9,13)	1	0	-	1	no
G_3	(7,15)	-	1	1	1	no
	(13,15)	1	-	1	1	no

Another example (cont.)

- Minimal cover generation:
  - EPIs selection

Prime implicant name	Prime implicant expression	Implicant minterms	Function minterms 2	6	7	8	9	13	15
P_1	$w'yz'$	(2,6)	⊗	×					
P_2	$x'y'z$	(8,9)				⊗	×		
P_3	$w'xy$	(6,7)		×	×				
P_4	$wx'z$	(9,13)					×	×	
P_5	xyz	(7,15)			×				×
P_6	wyz	(13,15)						×	×
	EPI covered minterms:		2	6		8	9		
	Not covered minterms:				7			13	15

  - PI list: w'yz', x'y'z, w'xy, wx'z, xyz, wyz
  - EPI list: w'yz', x'y'z
  - POS: (P3 + P5)(P4 + P6)(P5 + P6) =
    (P3 + P5)(P4P5 + P5P6 + P4P6 + P6) =
    P3P4P5 + **P4P5 + P3P6 + P5P6**

- Minimal cover expressions:
  - F1 = w'yz' + x'y'z + wx'z + xyz
  - F2 = w'yz' + x'y'z + w'xy + wyz
  - F3 = w'yz' + x'y'z + xyz + wyz

EXAMPLE

$f^{on} = \{m_0, m_1, m_2, m_3, m_5, m_8, m_{10}, m_{11}, m_{13}, m_{15}\} = \sum (0, 1, 2, 3, 5, 8, 10, 11, 13, 15)$

Minterm	Cube				
0	0	0	0	0	✓
1	0	0	0	1	✓
2	0	0	1	0	✓
8	1	0	0	0	✓
3	0	0	1	1	✓
5	0	1	0	1	✓
10	1	0	1	0	✓
11	1	0	1	1	✓
13	1	1	0	1	✓
15	1	1	1	1	✓

Minterm	Cube				
0,1	0	0	0	-	✓
0,2	0	0	-	0	✓
0,8	-	0	0	0	✓
1,3	0	0	-	1	✓
1,5	0	-	0	1	PI=D
2,3	0	0	1	-	✓
2,10	-	0	1	0	✓
8,10	1	0	-	0	✓
3,11	-	0	1	1	✓
5,13	-	1	0	1	PI=E
10,11	1	0	1	-	✓
11,15	1	-	1	1	PI=F
13,15	1	1	-	1	PI=G

Minterm	Cube				
0,1,2,3	0	0	-	-	PI=A
0,8,2,10	-	0	-	0	PI=C
2,3,10,11	-	0	1	-	PI=B

$f^{on} = \{A,B,C,D,E,F,G\} = \{00\text{--}, \text{-}01\text{-}, \text{-}0\text{-}0, 0\text{-}01, \text{-}101, 1\text{-}11, 11\text{-}1\}$

Construct Cover Table

- PIs Along Vertical Axis (in order of # of literals)
- Minterms Along Horizontal Axis

	0	1	2	3	5	8	10	11	13	15
A	x	x	x	x						
B			x	x			x	x		
C	x		x			x	x			
D		x			x					
E					x				x	
F								x		x
G									x	x

Finding the Minimum Cover

- Extract All Essential Prime Implicants, EPI
- EPIs are the PI for which a Single x Appears in a Column

	0	1	2	3	5	8	10	11	13	15
A	x	x	x	x						
B			x	x			x	x		
C	x		x			x	x			
D		x			x					
E					x				x	
F								x		x
G									x	x

- C is an EPI .
- Row C and Columns 0, 2, 8, and 10 can be Eliminated Giving Reduced Cover Table
- Examine Reduced Table for New EPIs

Reduced Table

Distinguished Column

	0	1	2	3	5	8	10	11	13	15
A	x	x	x	x						
B			x	x			x	x		
C	x		x			x	x			
D		x			x					
E					x				x	
F								x		x
G									x	x

Essential row

	1	3	5	11	13	15
A	x	x				
B		x		x		
D	x		x			
E			x		x	
F				x		x
G					x	x

- The Row of an EPI is an *Essential row*
- The Column of the Single x in the *Essential Row* is a *Distinguished Column*

The Reduced Cover Table

- Initially, Columns 0, 2, 8 and 10 Removed

	1	3	5	11	13	15
A	x	x				
B		x		x		
D	x		x			
E			x		x	
F				x		x
G					x	x

- No EPIs are Present
- No Row Dominance Exists
- No Column Dominance Exists
- This is *Cyclic Cover* Table
- Must Solve Exactly OR Use a Heuristic