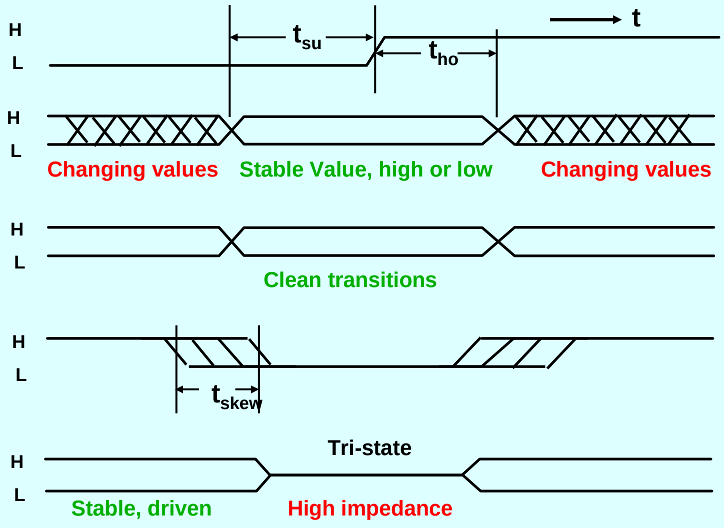
Timing Diagrams

- Graphical representation of circuit behavior over time
  - ➔ illustrates the logic behavior of signals in a digital circuit as a function of time
- May be used as a device specification
  - ➔ illustrates device performance
- May be used as a module or system specification
  - ➔ identifies a requirement for system performance
- May be used as a tool in system analysis

2

Timing Diagram Notation



Timing Terminology

Caution: terminology may vary slightly between vendors.

Set-up time , t_{su}: the minimum length of time that a signal must be valid at a circuit input before a second triggering signal arrives at a second input.

Usually a clock

Delay time , t_{co}: the length of time that a circuit requires for its output(s) to begin to change in response to a triggering signal arriving at an input. (also called propagation delay)

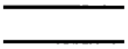
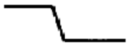
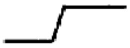
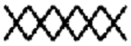
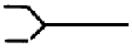
Hold time , t_{ho}: the minimum length of time that a signal must be kept valid at a circuit input after a triggering signal has been received at a second input.

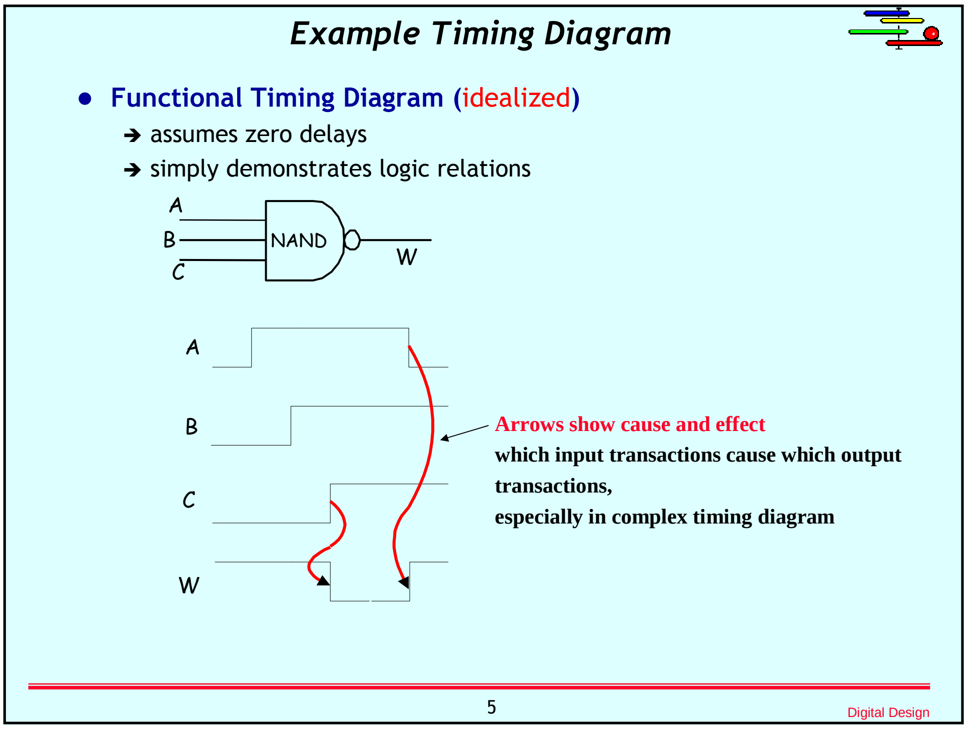
Timing skew , t_{skew}: the maximum range of times over which a particular signal transition can occur.

-- Due to variations in driver output impedance

-- Problems in clock distribution

Nomenclature for Timing Diagrams

Symbol	Input	Output
	The input must be valid	The output will be valid
	If the input were to fall	Then the output will fall
	If the input were to rise	Then the output will rise
	Don't care, it will work regardless	Don't know, the output value is indeterminate
	Nonsense	High impedence, tristate, HiZ, Not driven, floating



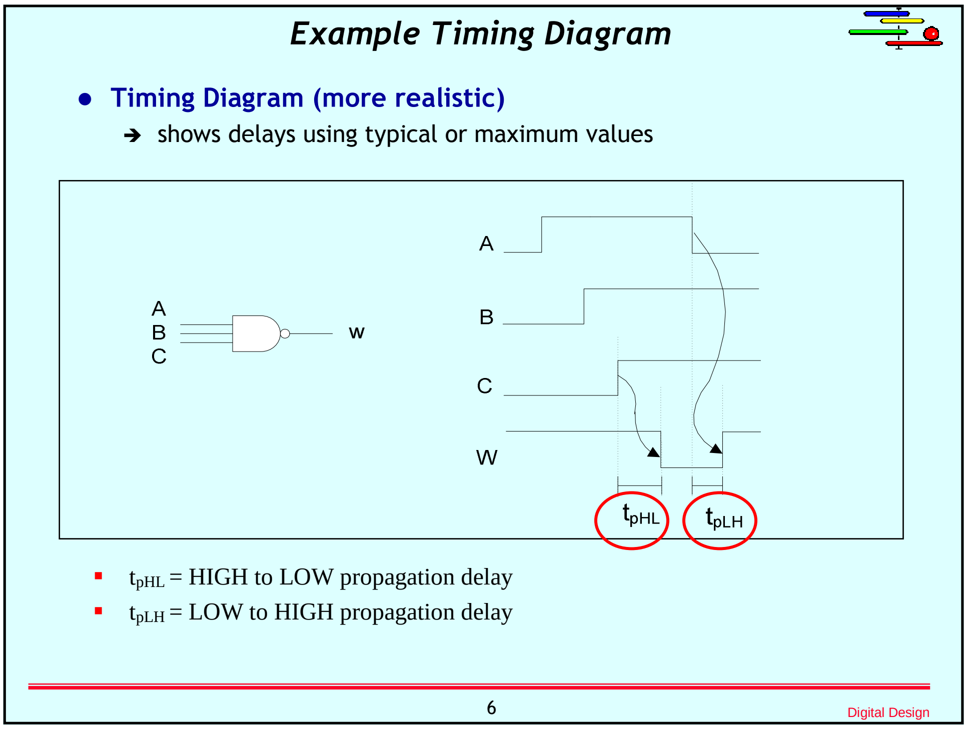
Example Timing Diagram

- Functional Timing Diagram (idealized)
  - ➔ assumes zero delays
  - ➔ simply demonstrates logic relations

Arrows show cause and effect

which input transactions cause which output transactions,

especially in complex timing diagram

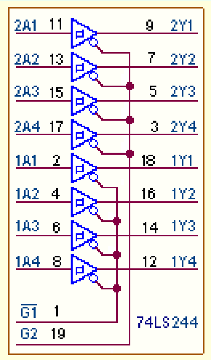


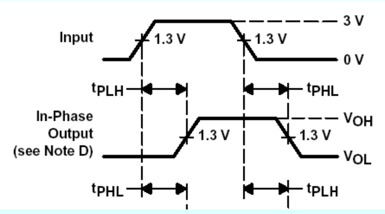
Example Timing Diagram

- Timing Diagram (more realistic)
  - ➔ shows delays using typical or maximum values

- t_{pHL} = HIGH to LOW propagation delay
- t_{pLH} = LOW to HIGH propagation delay

74LS244 Buffer 1

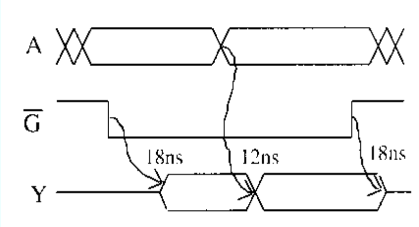




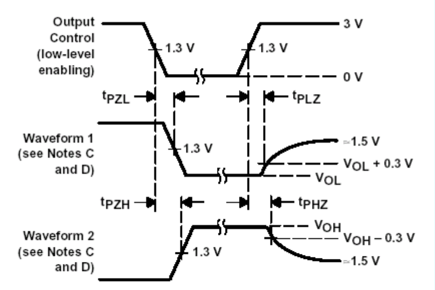
PARAMETER	TEST CONDITIONS	'LS240			'LS241, 'LS244			UNIT
		MIN	TYP	MAX	MIN	TYP	MAX	
t_{PLH}	R_L = 667 Ω, C_L = 45 pF		9	14		12	18	ns
t_{PHL}			12	18		12	18	

- Propagation delay = 12nS (enabled) - typical value
  - ➔ Analysis (design) should use MAXIMUM values for worst case.

74LS244 Buffer 2

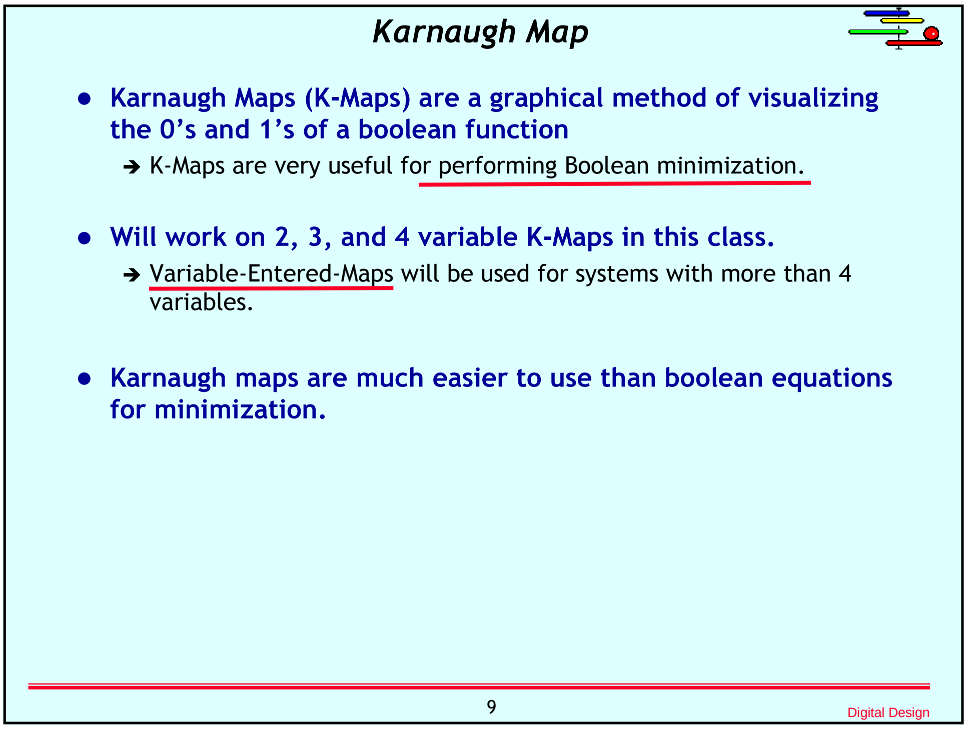


- tri-state buffer (see later slide)
- Propagation delay = 12nS (enabled)



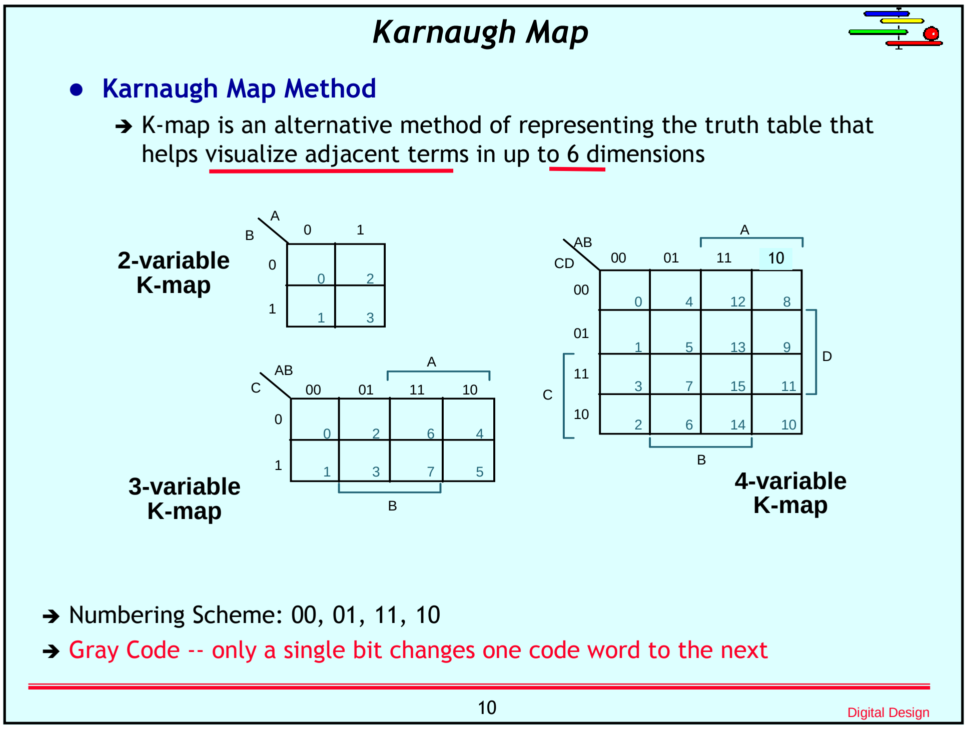
VOLTAGE WAVEFORMS
ENABLE AND DISABLE TIMES, 3-STATE OUTPUTS

PARAMETER	TEST CONDITIONS	'LS240			'LS241, 'LS244			UNIT
		MIN	TYP	MAX	MIN	TYP	MAX	
t_{PLH}	R_L = 667 Ω, C_L = 45 pF		9	14		12	18	ns
t_{PHL}			12	18		12	18	
t_{PZL}	R_L = 667 Ω, C_L = 45 pF		20	30		20	30	ns
t_{PZH}			15	23		15	23	
t_{PLZ}	R_L = 667 Ω, C_L = 5 pF		10	20		10	20	ns
t_{PHZ}			15	25		15	25	



Karnaugh Map

- Karnaugh Maps (K-Maps) are a graphical method of visualizing the 0's and 1's of a boolean function
  - ➔ K-Maps are very useful for performing Boolean minimization.
- Will work on 2, 3, and 4 variable K-Maps in this class.
  - ➔ Variable-Entered-Maps will be used for systems with more than 4 variables.
- Karnaugh maps are much easier to use than boolean equations for minimization.



Karnaugh Map

- Karnaugh Map Method
  - ➔ K-map is an alternative method of representing the truth table that helps visualize adjacent terms in up to 6 dimensions

2-variable K-map

B \ A	0	1
0	0	2
1	1	3

3-variable K-map

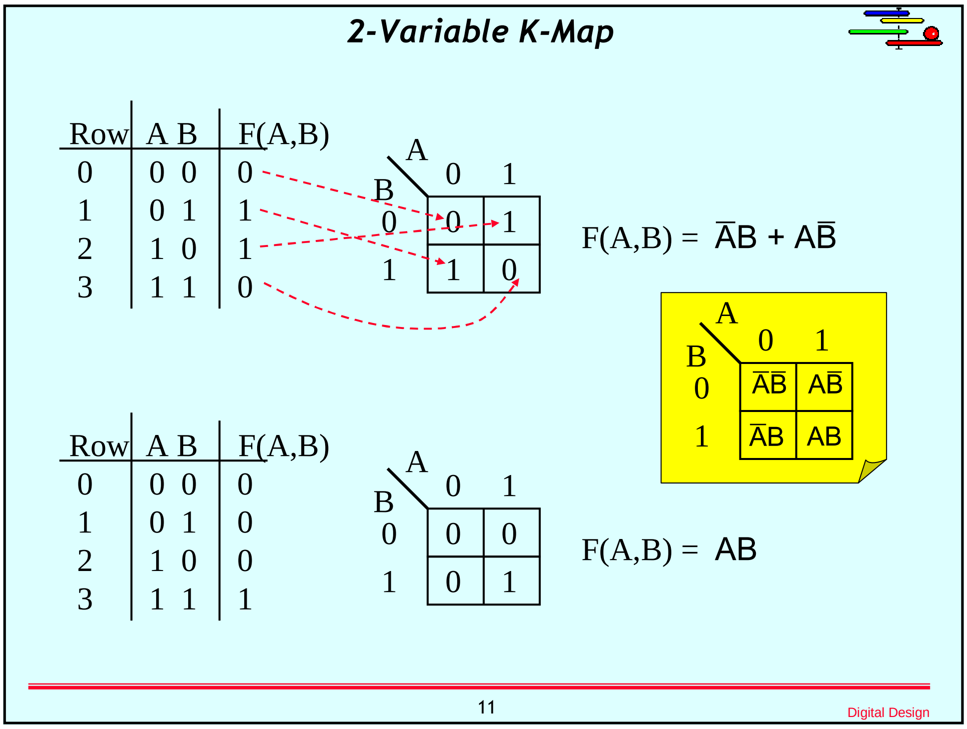
			A	A
C \ AB	00	01	11	10
0	0	2	6	4
1	1	3	7	5
		B	B	

4-variable K-map

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

➔ Numbering Scheme: 00, 01, 11, 10

➔ Gray Code -- only a single bit changes one code word to the next



2-Variable K-Map

Row	A B	F(A,B)
0	0 0	0
1	0 1	1
2	1 0	1
3	1 1	0

B \ A	0	1
0	0	1
1	1	0

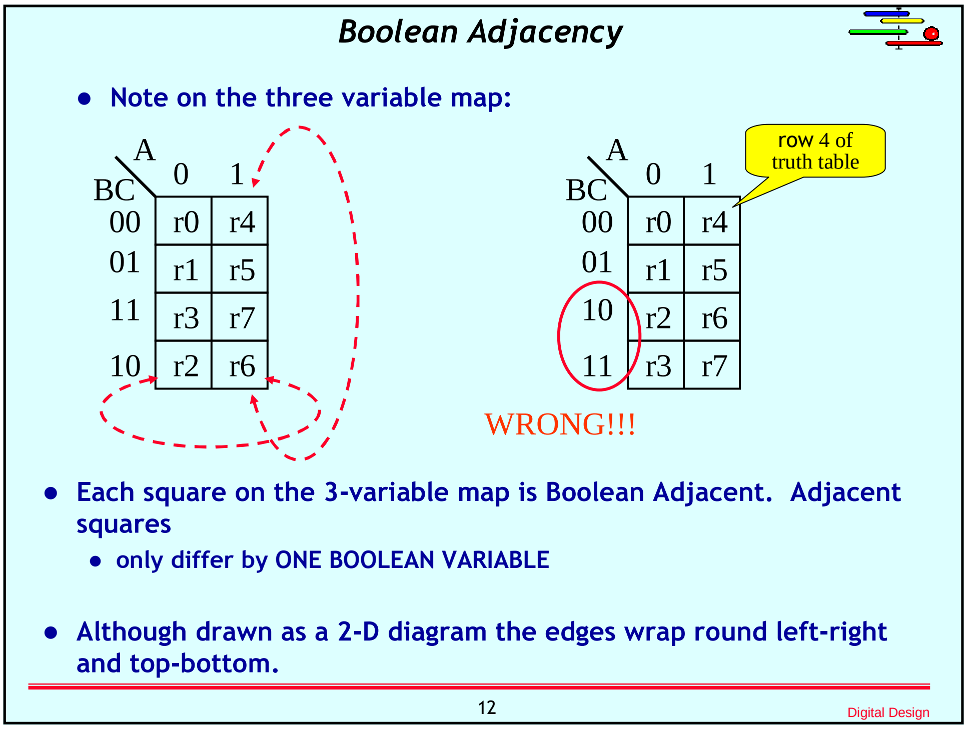
F(A,B) = $\bar{A}B + A\bar{B}$

B \ A	0	1
0	$\bar{A}\bar{B}$	$A\bar{B}$
1	$\bar{A}B$	AB

Row	A B	F(A,B)
0	0 0	0
1	0 1	0
2	1 0	0
3	1 1	1

B \ A	0	1
0	0	0
1	0	1

F(A,B) = AB



Boolean Adjacency

- Note on the three variable map:

BC \ A	0	1
00	r0	r4
01	r1	r5
11	r3	r7
10	r2	r6

BC \ A	0	1
00	r0	r4
01	r1	r5
10	r2	r6
11	r3	r7

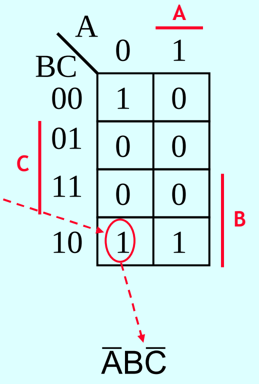
row 4 of truth table

WRONG!!!

- Each square on the 3-variable map is Boolean Adjacent. Adjacent squares
  - only differ by ONE BOOLEAN VARIABLE
- Although drawn as a 2-D diagram the edges wrap round left-right and top-bottom.

3-Variable K-Map

Row	A B C	F(A,B,C)
0	0 0 0	1
1	0 0 1	0
2	0 1 0	1
3	0 1 1	0
4	1 0 0	0
5	1 0 1	0
6	1 1 0	1
7	1 1 1	0



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

$F(A,B,C) = \Sigma\, m(0,2,6)$

sum **of minterms**

Plotting 4-Variable Functions

Row	A B C D	F(A,B,C,D)
0	0 0 0 0	?
1	0 0 0 1	?
2	0 0 1 0	?
3	0 0 1 1	?
4	0 1 0 0	?
5	0 1 0 1	?
6	0 1 1 0	?
7	0 1 1 1	?
8	1 0 0 0	?
9	1 0 0 1	?
10	1 0 1 0	?
11	1 0 1 1	?
12	1 1 0 0	?
13	1 1 0 1	?
14	1 1 1 0	?
15	1 1 1 1	?

CD \ AB	00	01	11	10
00	?	?	?	?
01	?	?	?	?
11	?	?	?	?
10	?	?	?	?

CD \ AB	00	01	11	10
00	r0	r4	r12	r8
01	r1	r5	r13	r9
11	r3	r7	r15	r11
10	r2	r6	r14	r10

Plotting 4-Variable Functions

Row	A B C D	F(A,B,C,D)
0	0 0 0 0	0
1	0 0 0 1	0
2	0 0 1 0	1
3	0 0 1 1	1
4	0 1 0 0	0
5	0 1 0 1	0
6	0 1 1 0	1
7	0 1 1 1	0
8	1 0 0 0	0
9	1 0 0 1	0
10	1 0 1 0	1
11	1 0 1 1	0
12	1 1 0 0	0
13	1 1 0 1	0
14	1 1 1 0	0
15	1 1 1 1	1

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

$\bar{A}\bar{B}CD$

F =Σ m(2,3,6,10,15)

Minimization via K-Maps

Row	**A B C**	**F(A,B,C)**
0	**0 0 0**	**0**
1	**0 0 1**	**0**
2	**0 1 0**	**1**
3	**0 1 1**	**0**
4	**1 0 0**	**0**
5	**1 0 1**	**0**
6	**1 1 0**	**1**
7	**1 1 1**	**0**

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

F(A,B,C) = Σ m(2,6)

$= \bar{A}B\bar{C} + AB\bar{C}$

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

$= B\bar{C}$

Boolean adjacency can be used to minimize functions!

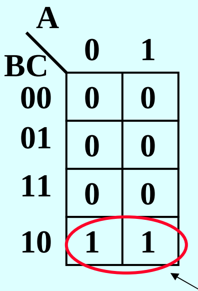
Simplification using K-Maps

- **Grouping blocks of '1'**
  - ➔ A group must consist of 16,8,4,2 or 1 cells
  - ➔ Each cell must be horizontally and/or vertically adjacent to cells in the other group
  - ➔ Always include the largest number of '1's in a group
  - ➔ Each '1' in the map should be included in a group
  - ➔ Groups can overlap
  - ➔ Map edge cells connect in a loop to cells at the opposite edge

- **Naming groups**
  - ➔ The product description for a group will include ALL variables that are CONSTANT over the group
  - ➔ For example, for a 4-variable map
    - ➢ An 8-cell group is described by a 1-variable product term
    - ➢ An 4-cell group is described by a 2-variable product term
    - ➢ An 2-cell group is described by a 3-variable product term
    - ➢ An 1-cell group is described by a 4-variable product term

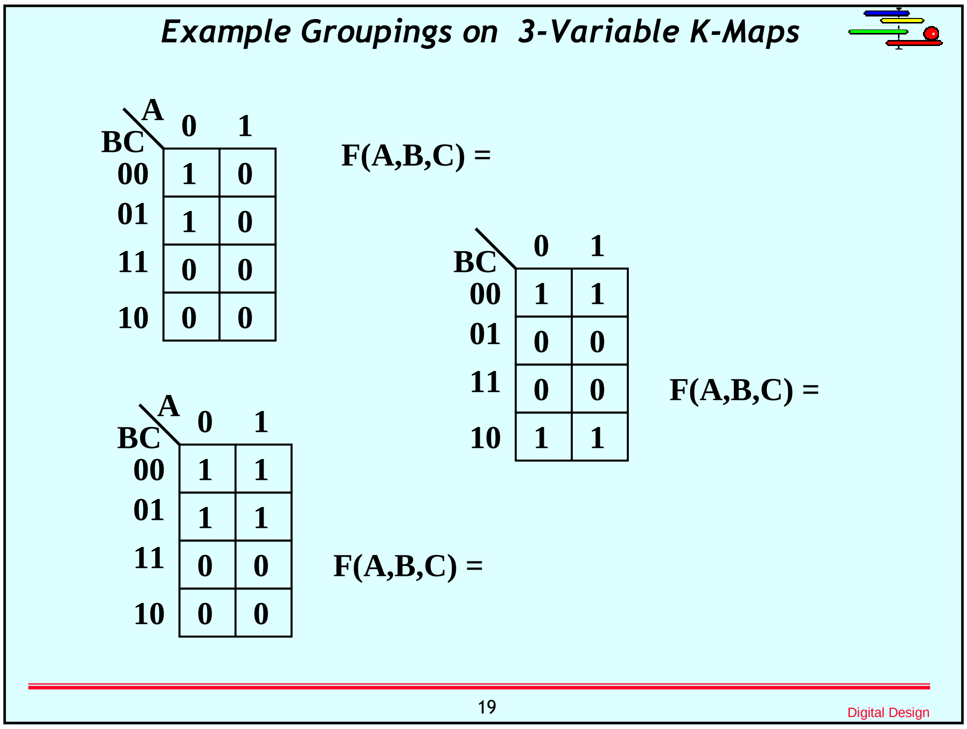
Groupings on K-Maps

- **Grouping can be read DIRECTLY as "BC" by looking at what is COMMON within the circled group.**



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

$F(A,B,C) = B\overline{C}$



Example Groupings on 3-Variable K-Maps

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

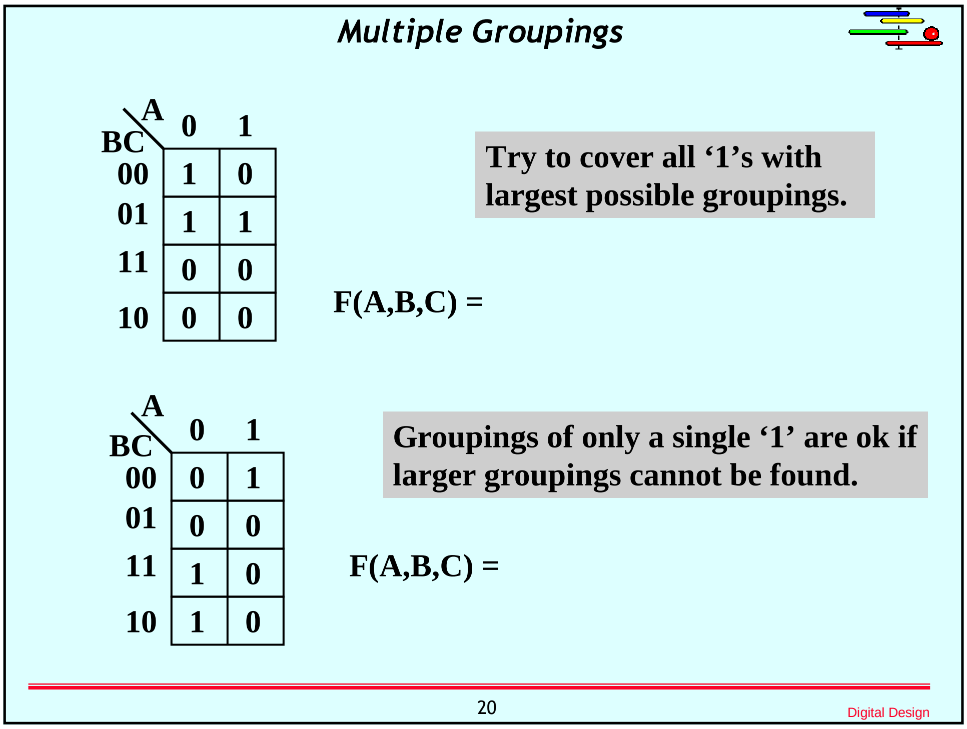
F(A,B,C) =

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

F(A,B,C) =

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

F(A,B,C) =



Multiple Groupings

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

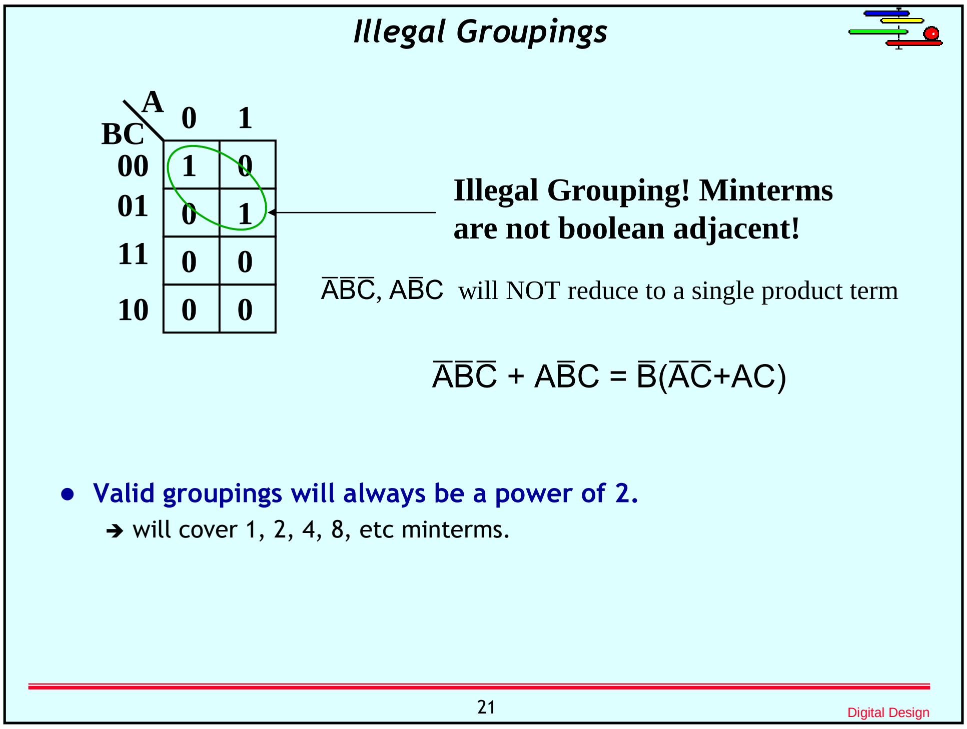
Try to cover all ‘1’s with largest possible groupings.

F(A,B,C) =

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

Groupings of only a single ‘1’ are ok if larger groupings cannot be found.

F(A,B,C) =



Illegal Groupings

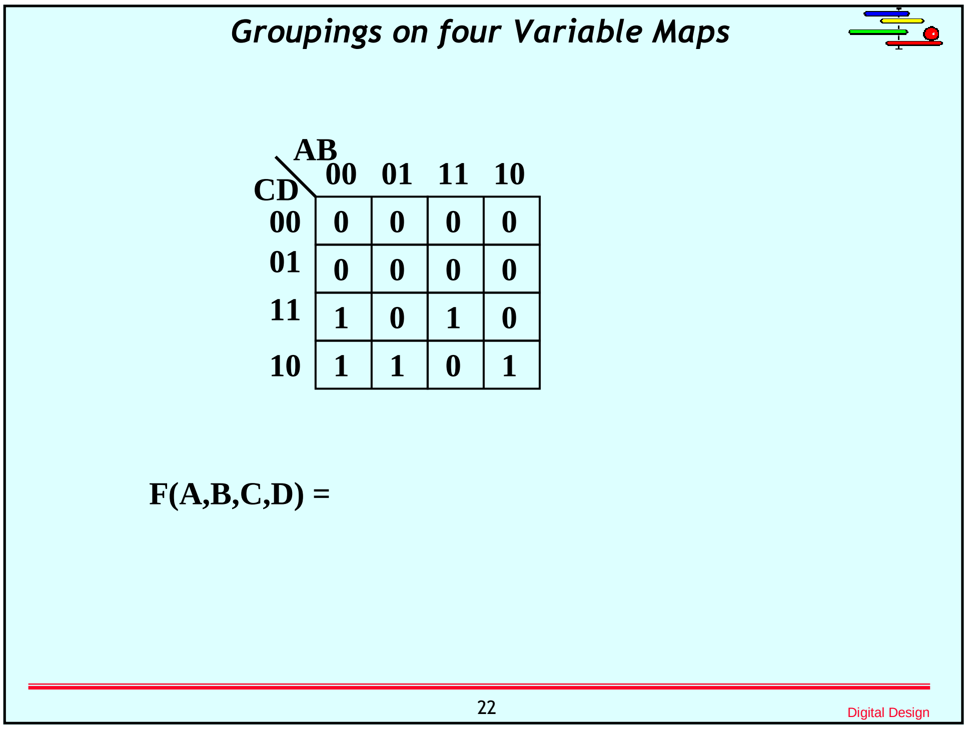
BC \ A	0	1
00	1	0
01	0	1
11	0	0
10	0	0

Illegal Grouping! Minterms are not boolean adjacent!

$\overline{A}\overline{B}\overline{C}$, $A\overline{B}C$ will NOT reduce to a single product term

$$\overline{A}\overline{B}\overline{C} + A\overline{B}C = \overline{B}(\overline{A}\overline{C}+AC)$$

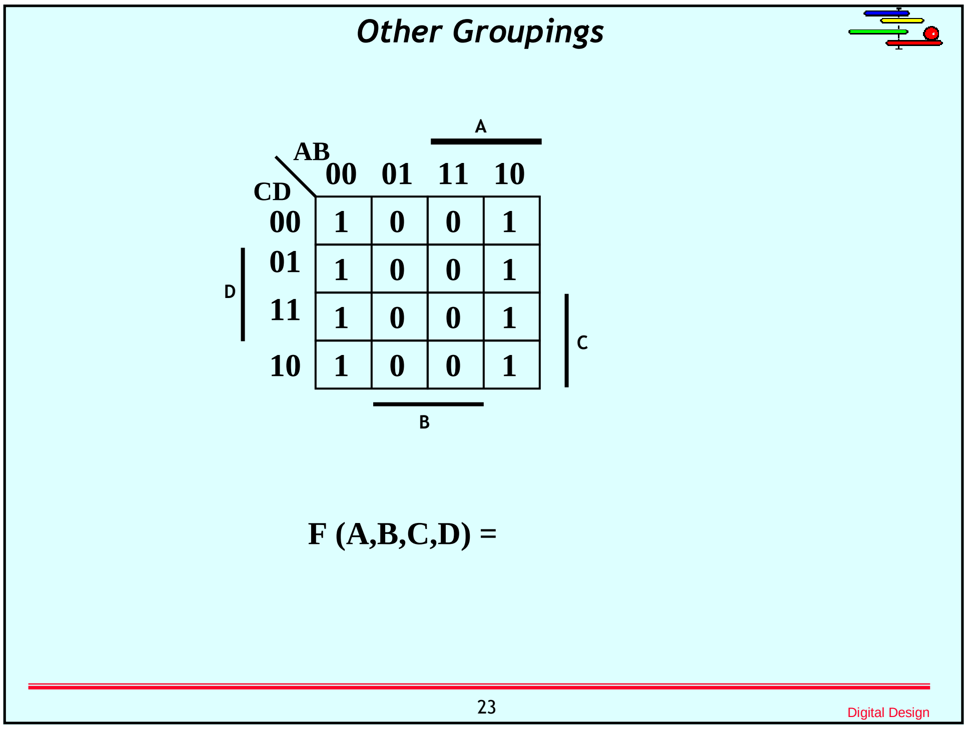
- Valid groupings will always be a power of 2.
  - ➔ will cover 1, 2, 4, 8, etc minterms.



Groupings on four Variable Maps

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

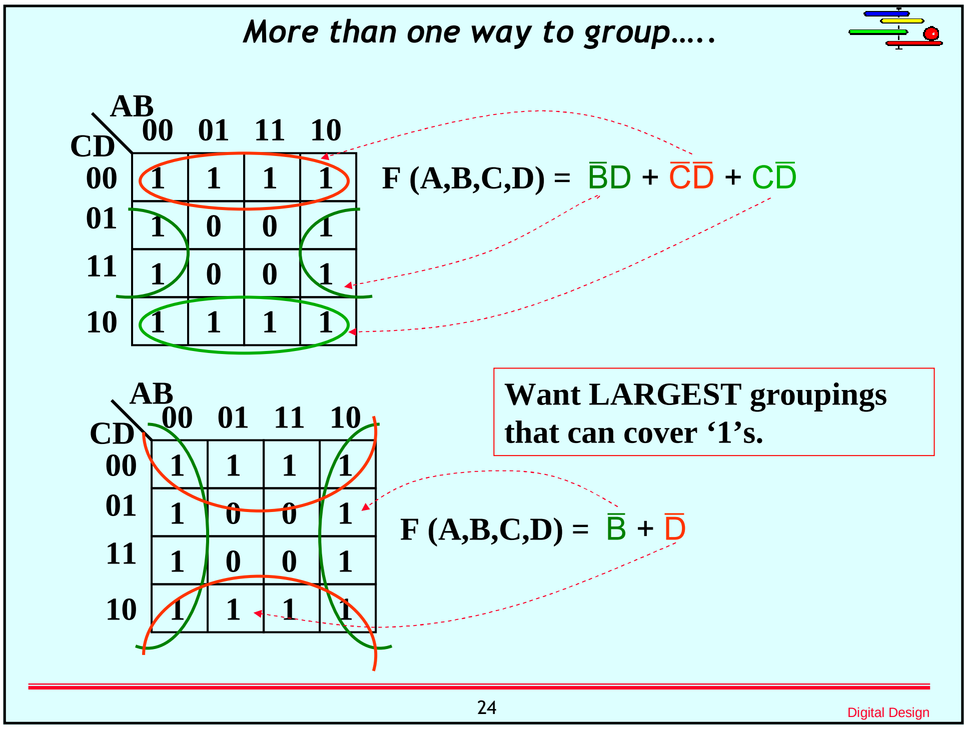
F(A,B,C,D) =



Other Groupings

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

F (A,B,C,D) =



More than one way to group…..

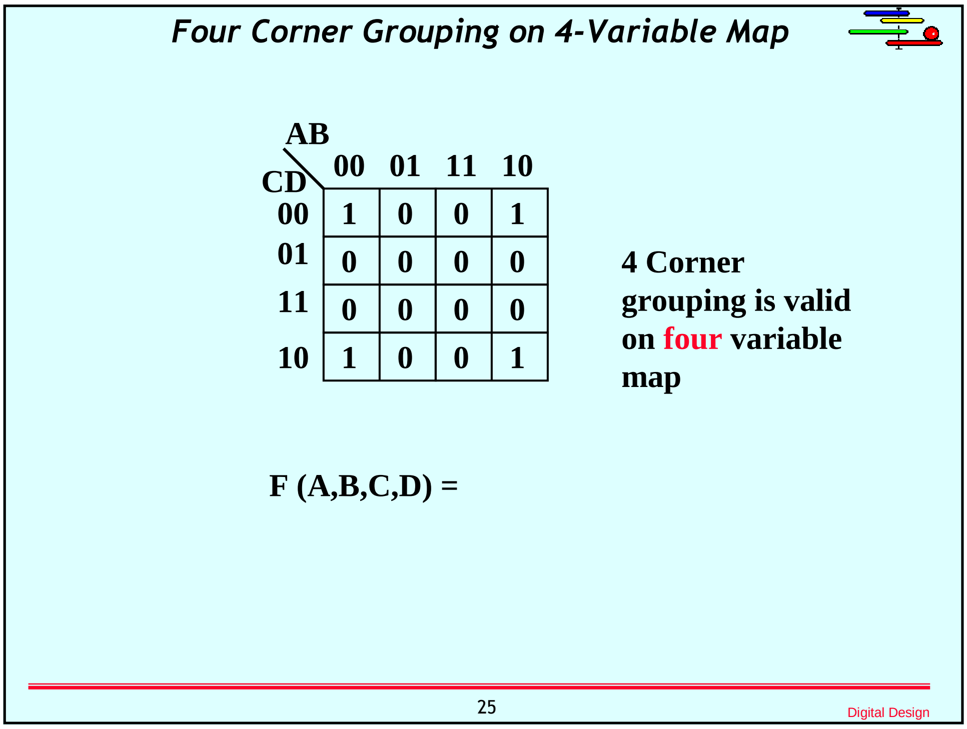
CD \ AB	00	01	11	10
00	1	1	1	1
01	1	0	0	1
11	1	0	0	1
10	1	1	1	1

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

Want LARGEST groupings that can cover ‘1’s.

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

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

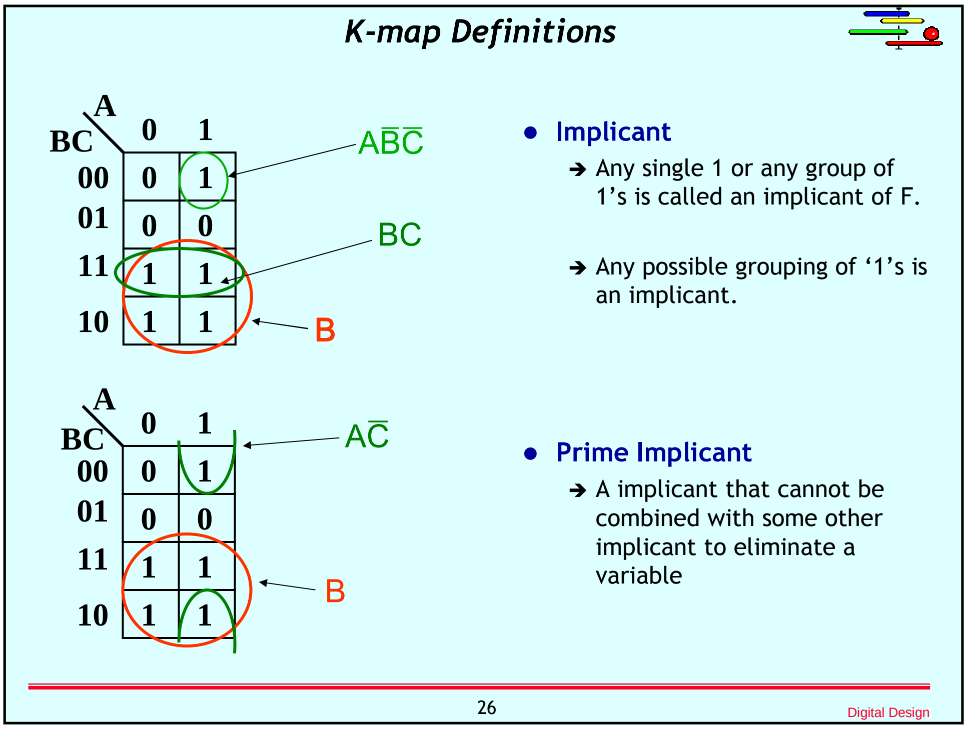


Four Corner Grouping on 4-Variable Map

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

4 Corner grouping is valid on four variable map

F (A,B,C,D) =



K-map Definitions

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

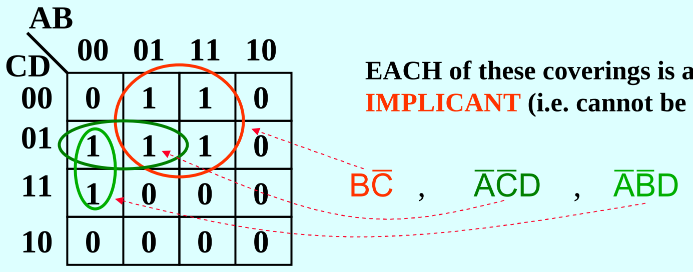
BC \ A	0	1
00	0	1
01	0	0
11	1	1
10	1	1

- **Implicant**
  - ➔ Any single 1 or any group of 1's is called an implicant of F.
  - ➔ Any possible grouping of '1's is an implicant.
- **Prime Implicant**
  - ➔ A implicant that cannot be combined with some other implicant to eliminate a variable

Minimum Sum-Of-Products (SOP)

- The minimum SOP expression consists of some (but not necessarily all) of the prime implicants of a function.
- If a SOP expression contains a term which is NOT a prime implicant, then it CANNOT be minimum.

Prime Implicants



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	0

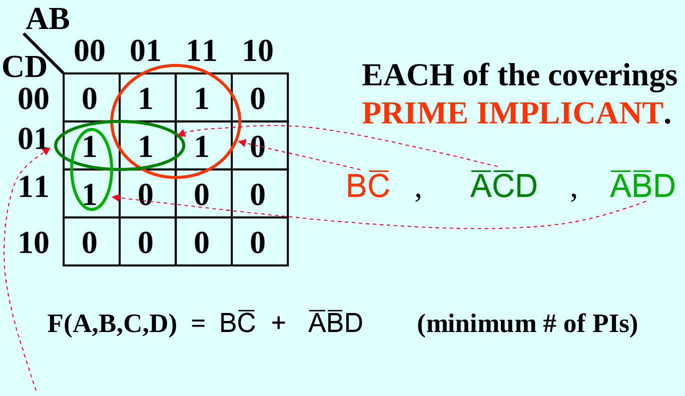
$B\overline{C}$, $\overline{A}\,\overline{C}D$, $\overline{A}\,\overline{B}D$

EACH of these coverings is a PRIME IMPLICANT (i.e. cannot be reduced)

Minimum SOP will have some or all of these prime implicants.
The included prime implicants must cover all of the ONEs.

F(A,B,C,D)	$= B\overline{C} + \overline{A}\,\overline{B}D$	**(minimum set of PIs)**
	$= B\overline{C} + \overline{A}\,\overline{B}D + \overline{A}\,\overline{C}D$	**(valid set of PIs, but not minimum)**
	$\neq \overline{A}\,\overline{B}D + \overline{A}\,\overline{C}D$	**(both PI's, but all '1's not included!)**

Non-Essential vs. Essential Prime Implicants



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	0

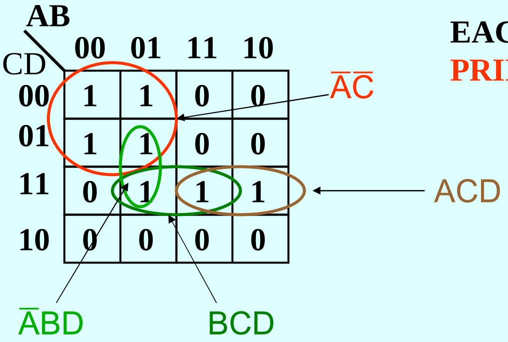
EACH of the coverings is a **PRIME IMPLICANT**.

$B\bar{C}$, $\bar{A}\bar{C}D$, $\bar{A}\bar{B}D$

F(A,B,C,D) = $B\bar{C}$ + $\bar{A}\bar{B}D$ (minimum # of PIs)

- NON-ESSENTIAL prime implicant
  - ➔ Prime Implicant $\bar{A}\bar{C}D$ is non-essential because its ‘1’s are covered by other PIs . A PI is ESSENTIAL if it covers a MINTERM that cannot be covered by any other PI.

An example with more than one solution



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

$\bar{A}\bar{C}$

ACD

$\bar{A}BD$

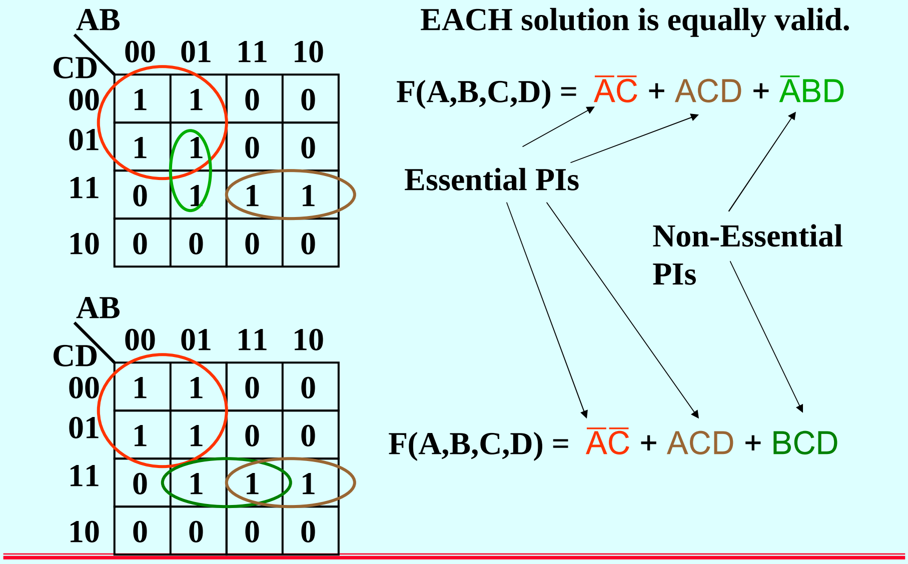
BCD

EACH of the coverings is a **PRIME IMPLICANT**.

Recall that a covering is a Prime Implicant if it cannot be combined with another covering to eliminate a variable.

Two Solutions

EACH solution is equally valid.



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

$F(A,B,C,D) = \overline{A}\,\overline{C} + ACD + \overline{A}BD$

Essential PIs

Non-Essential PIs

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

$F(A,B,C,D) = \overline{A}\,\overline{C} + ACD + BCD$

Minimal Solution

- A minimal SOP will consist of prime implicants.
- A minimal SOP equation will have all of the essential prime implicants on the map. By definition, these cover a minterm that may not be covered by some other prime implicant.
- The minimal SOP equation may or may not include non-essential prime implicants. It will include non-essential prime implicants if there are ‘1’s remaining that have not been covered by an essential prime implicant.