CS 211: Computer Architecture
Digital Logic

Topics:

- Converting truth tables to expressions
- Karnaugh maps

Converting Truth Table to Boolean Expression

sensor inputs

A	B	C	Output
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

Given a circuit, isolate the rows in which the output of the circuit should be true

Converting Truth Table to Boolean Expression

sensor inputs

A	B	C	Output
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

sensor inputs

A	B	C	Output	
0	0	0	0	
0	0	1	0	
0	1	0	0	
0	1	1	1	$\overline{A}BC = 1$
1	0	0	0	
1	0	1	1	$A\overline{B}C = 1$
1	1	0	1	$AB\overline{C} = 1$
1	1	1	1	$ABC = 1$

Given a circuit, isolate that rows in which the output of the circuit should be true

A product term that contains exactly one instance of every variable is called a minterm

Converting Truth Table to Boolean Expression

sensor inputs

A	B	C	Output	
0	0	0	0	
0	0	1	0	
0	1	0	0	
0	1	1	1	$\overline{A}BC = 1$
1	0	0	0	
1	0	1	1	$A\overline{B}C = 1$
1	1	0	1	$AB\overline{C} = 1$
1	1	1	1	$ABC = 1$

$$\text{Output} = \overline{A}BC + A\overline{B}C + AB\overline{C} + ABC$$

Given the expressions for each row, build a larger Boolean expression for the entire table.

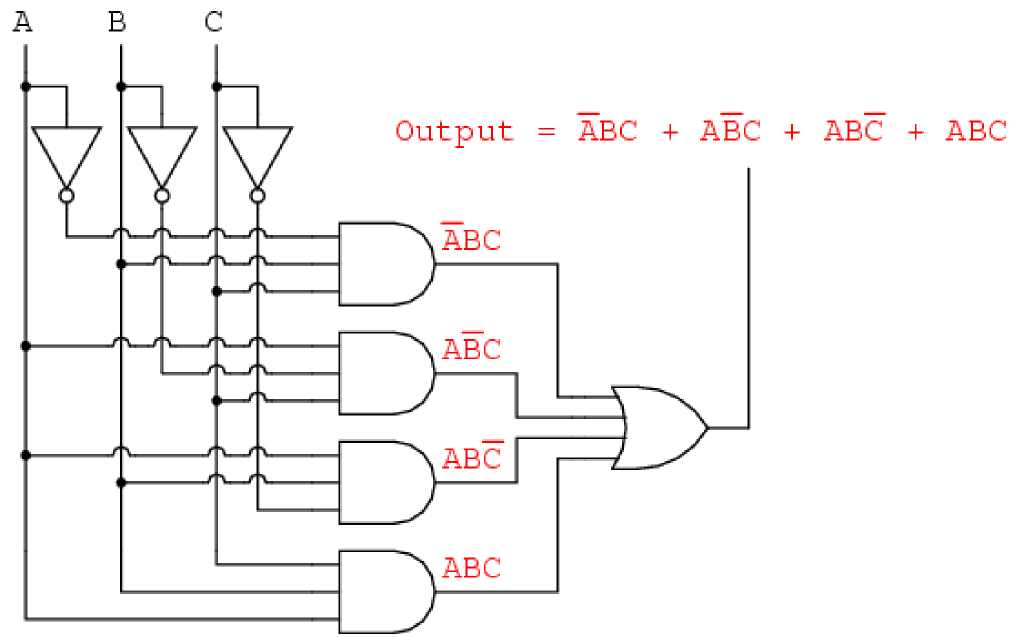
- This is a sum-of-products (SOP) form.

Converting Truth Table to Boolean Expression

sensor inputs

A	B	C	Output	
0	0	0	0	
0	0	1	0	
0	1	0	0	
0	1	1	1	$\overline{A}BC = 1$
1	0	0	0	
1	0	1	1	$A\overline{B}C = 1$
1	1	0	1	$AB\overline{C} = 1$
1	1	1	1	$ABC = 1$

$\text{Output} = \overline{A}BC + A\overline{B}C + AB\overline{C} + ABC$



Finally build the circuit.

- Problem: SOP forms are often not minimal.
- Solution: Make it minimal. We'll go over two ways.

First approach: algebraic

$$\bar{A}BC + A\bar{B}C + AB\bar{C} + ABC$$

$$\downarrow$$

$$BC(\bar{A} + A) + A\bar{B}C + AB\bar{C}$$

$$\downarrow$$

$$BC(1) + A\bar{B}C + AB\bar{C}$$

$$\downarrow$$

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

$$\downarrow$$

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

$$\downarrow$$

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

$$\downarrow$$

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

$$\downarrow$$

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

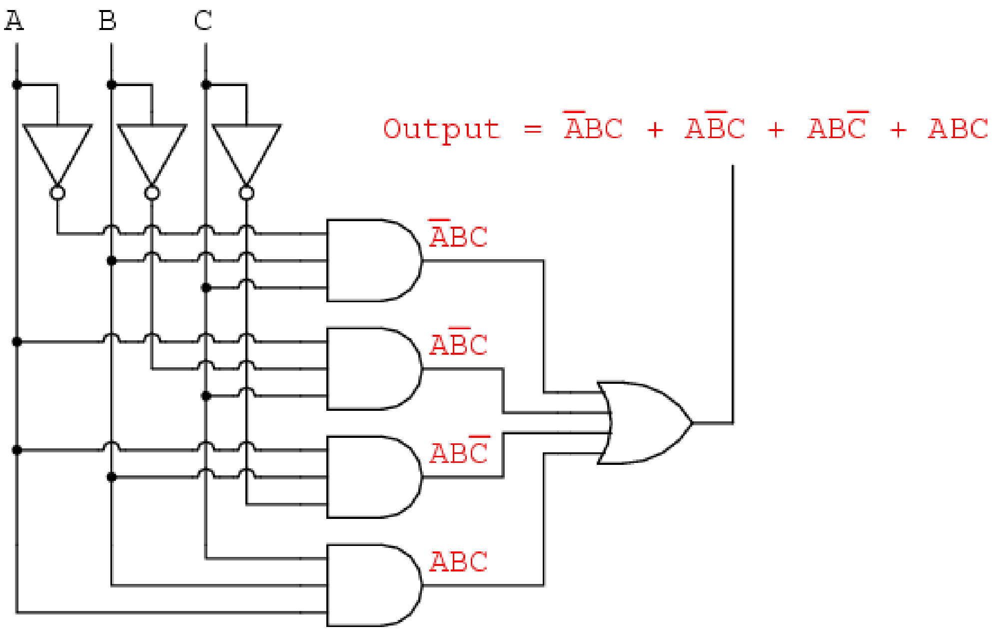
$$\downarrow$$

$$BC + A(B + C)$$

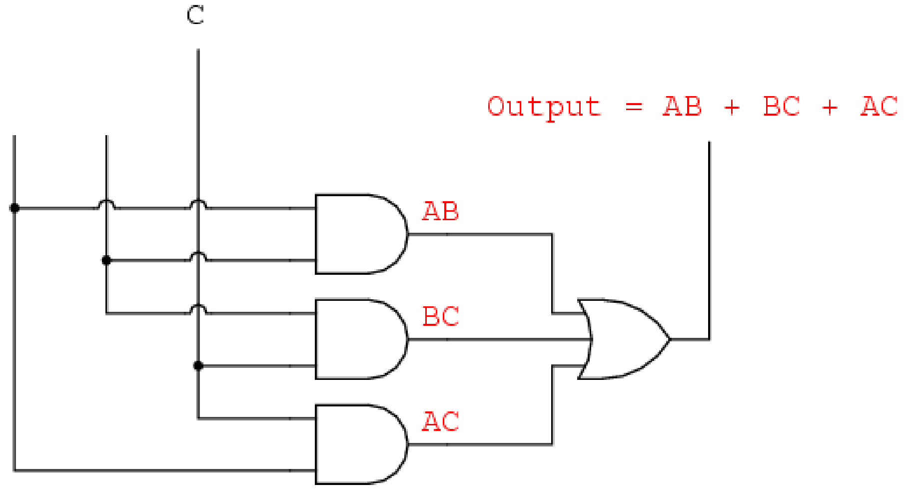
$$\downarrow$$

$$BC + AB + AC$$

The Result



A
B
C
Output = $\overline{A}BC + A\overline{B}C + AB\overline{C} + ABC$
$\overline{A}BC$
$A\overline{B}C$
$AB\overline{C}$
ABC



C
Output = AB + BC + AC
AB
BC
AC

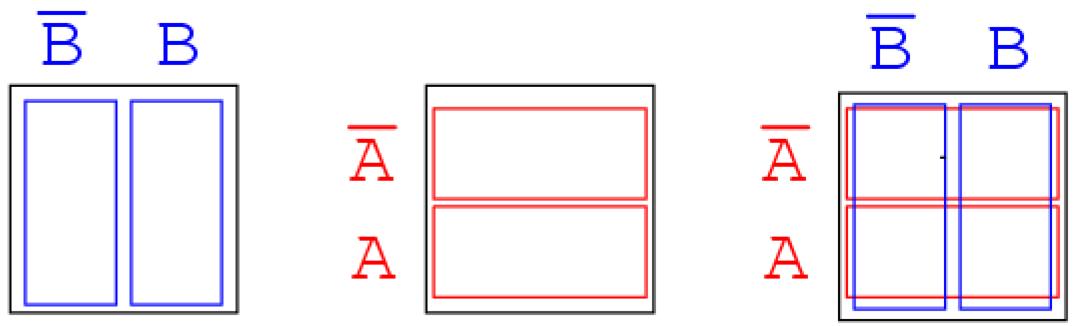
Karnaugh Maps or K-Maps

K-maps are a graphical technique to view minterms and how they relate.

The “map” is a diagram made up of squares, with each square representing a single minterm.

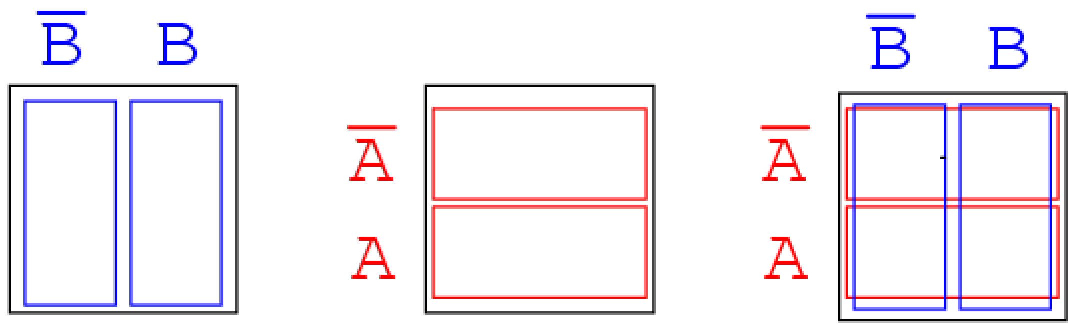
Minterms resulting in a “1” are marked as “1”, all others are marked “0”

2 Variable K-Map

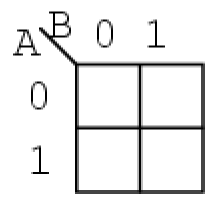


9

2 Variable K-Map

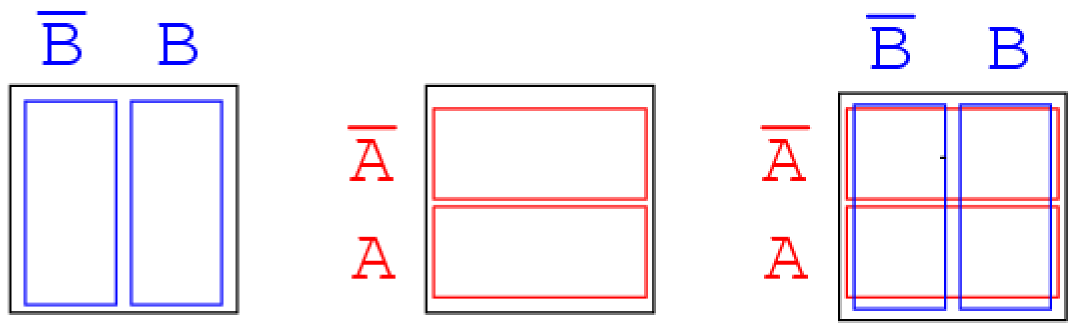


A	B	Output
0	0	0
0	1	1
1	0	0
1	1	1



10

2 Variable K-Map

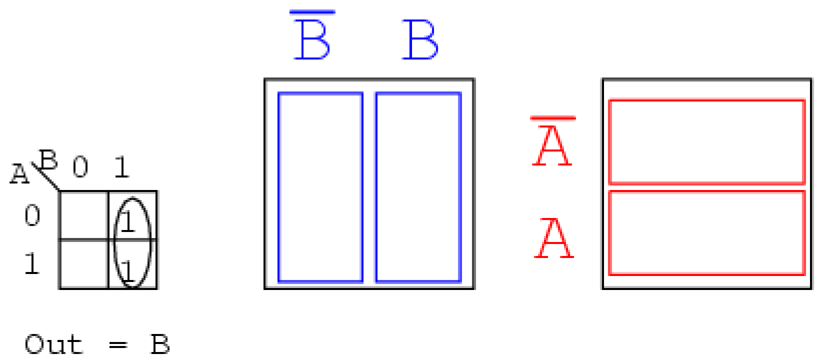


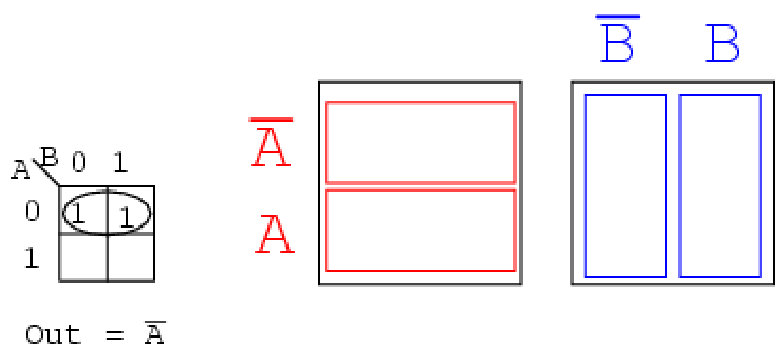
A	B	Output
0	0	0
0	1	1
1	0	0
1	1	1

A \ B	0	1
0	0	1
1	0	1

11

Finding Commonality

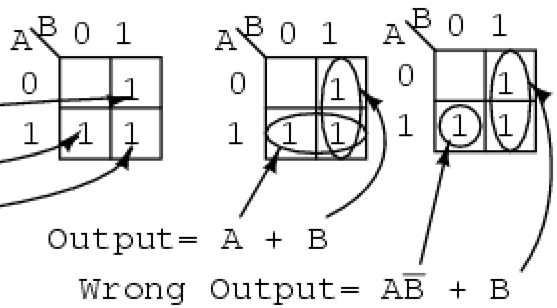




12

Finding the “best” solution

A	B	Output
0	0	0
0	1	1
1	0	1
1	1	1

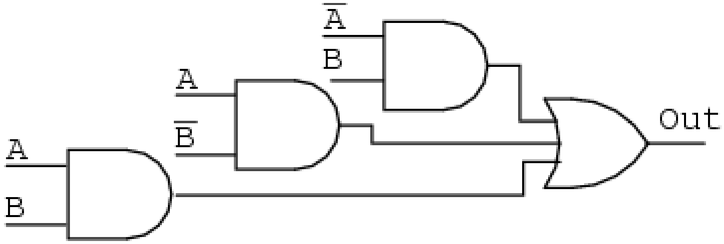


Grouping become simplified products.

Both are “correct”. “A+B” is preferred.

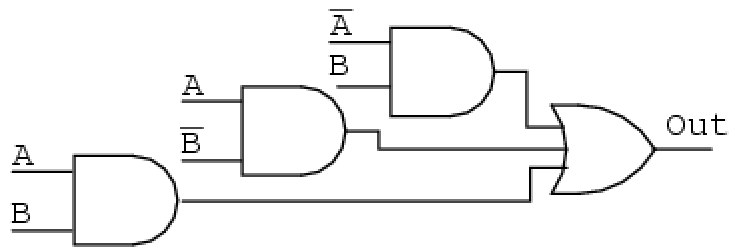
13

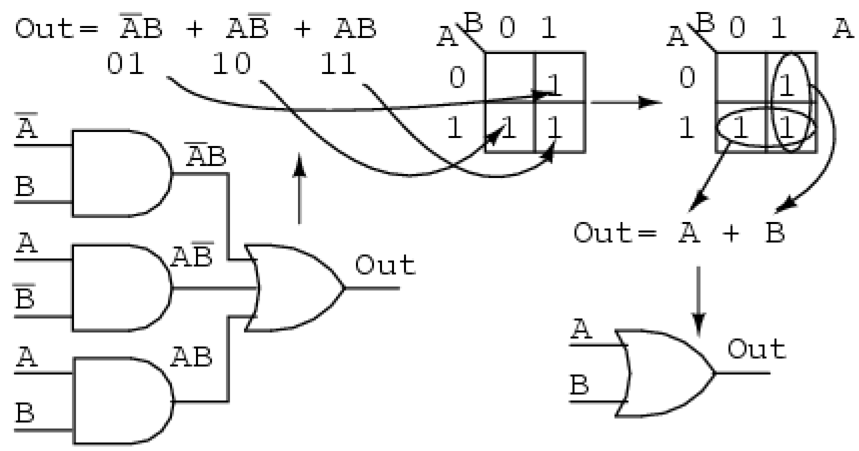
Simplify Example



14

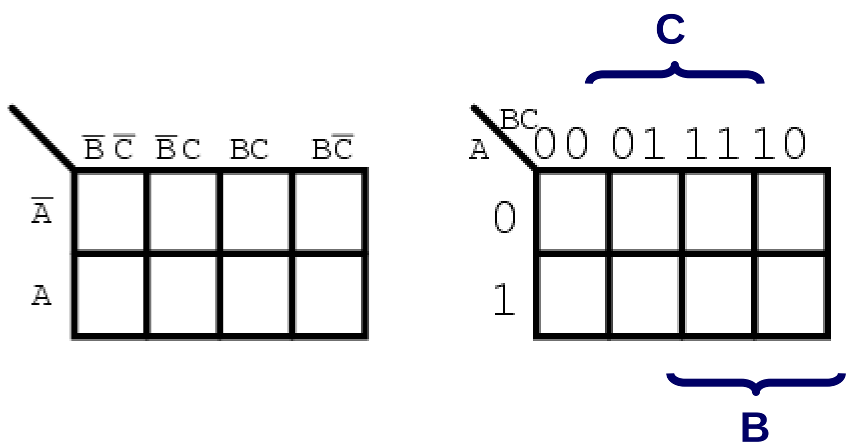
Simplify Example





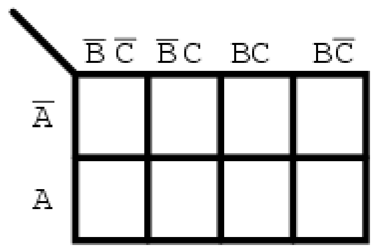
15

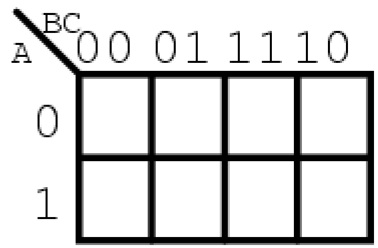
3 Variable K-Maps

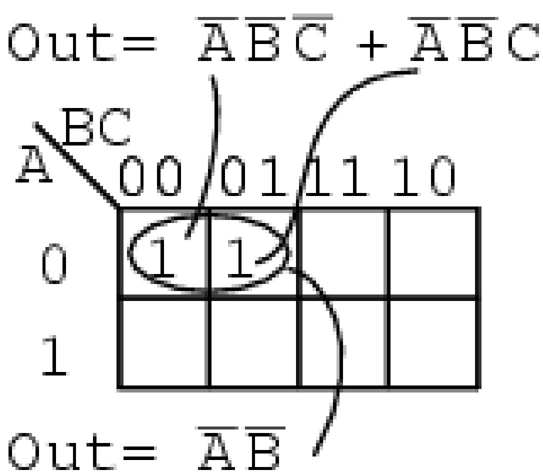


- Note in higher maps, several variables occupy a given axis
- The sequence of 1s and 0s follow a Gray Code Sequence.

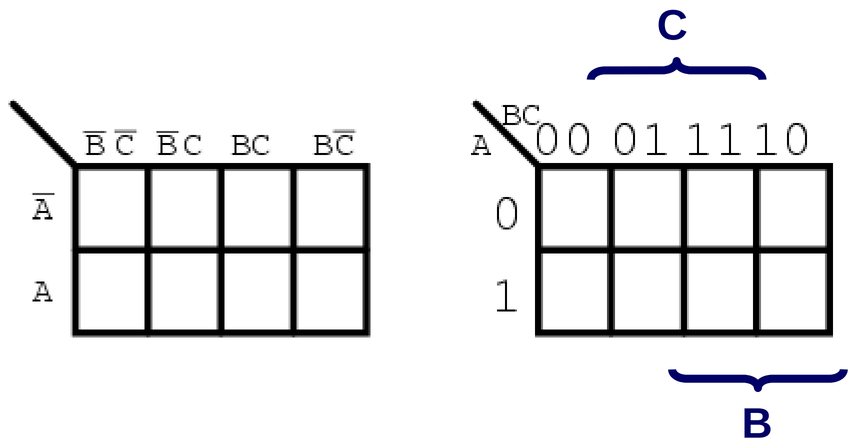
3 Variable K-Maps



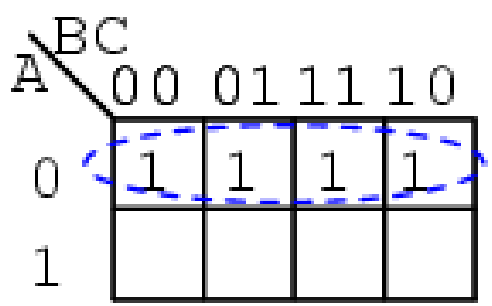




3 Variable K-Maps

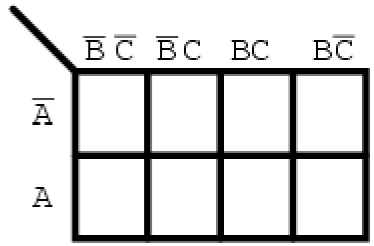


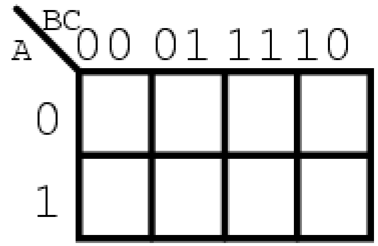
$$Out = \bar{A}\bar{B}\bar{C} + \bar{A}\bar{B}C + \bar{A}BC + \bar{A}B\bar{C}$$



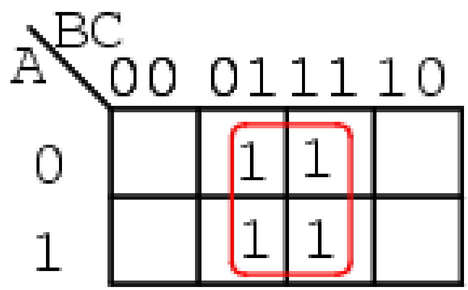
$$Out = \bar{A}$$

3 Variable K-Maps





$$Out = \bar{A}\bar{B}C + \bar{A}BC + A\bar{B}C + ABC$$



$$Out = C$$

3 Variable K-Maps

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

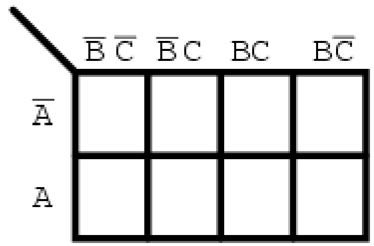
A \ BC	00	01	11	10
0				
1				

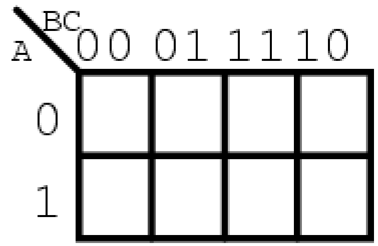
$$Out = \overline{A}\,\overline{B}\,\overline{C} + \overline{A}\,\overline{B}\,C + \overline{A}\,B\,C + \overline{A}\,B\,\overline{C} + ABC + A\,B\,\overline{C}$$

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

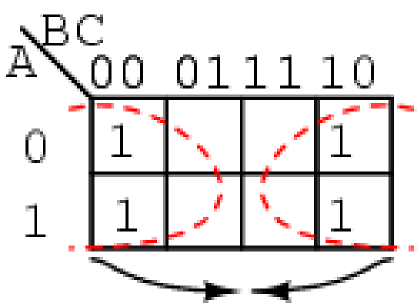
$$Out = \overline{A} + B$$

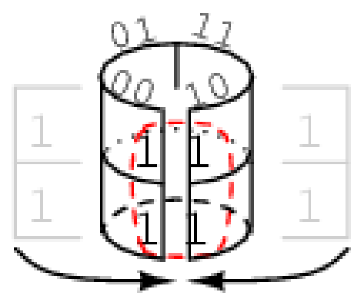
3 Variable K-Maps





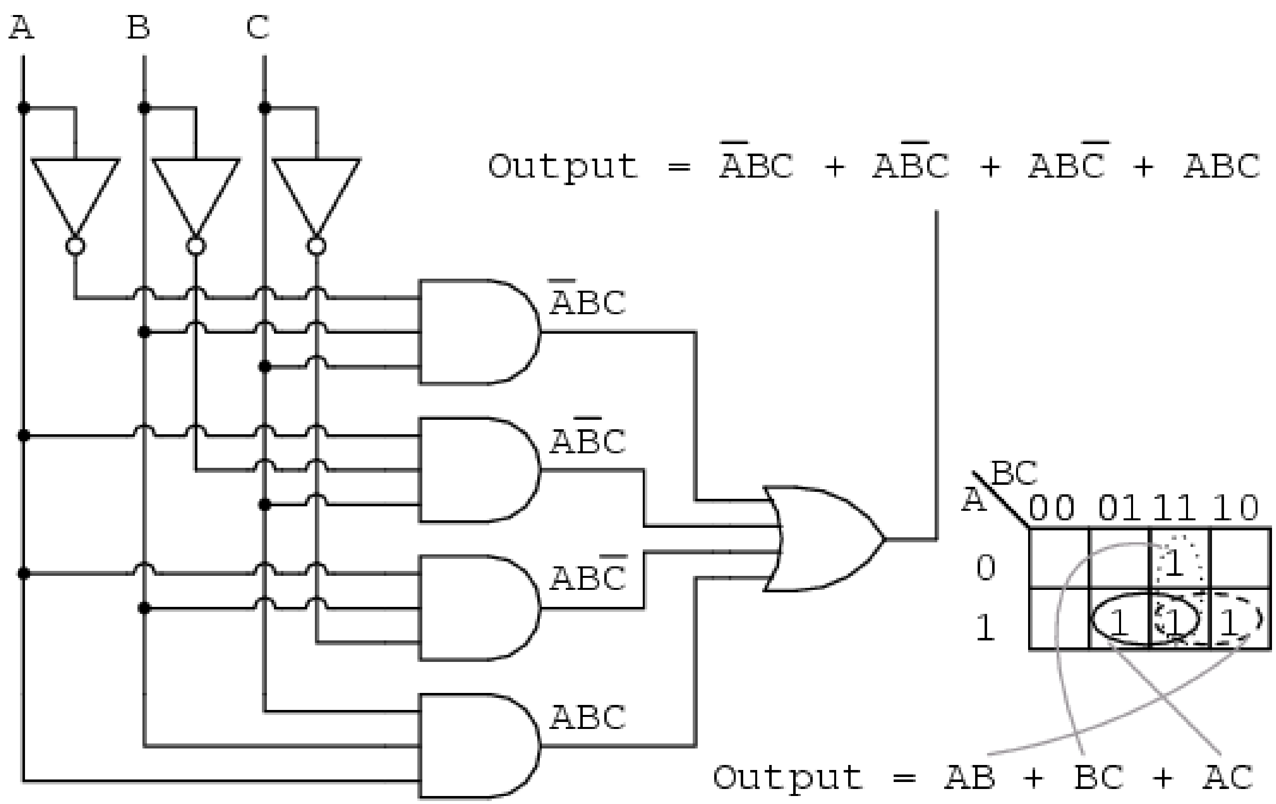
$$Out = \bar{A}\bar{B}\bar{C} + A\bar{B}\bar{C} + \bar{A}B\bar{C} + AB\bar{C}$$





$$Out = \bar{C}$$

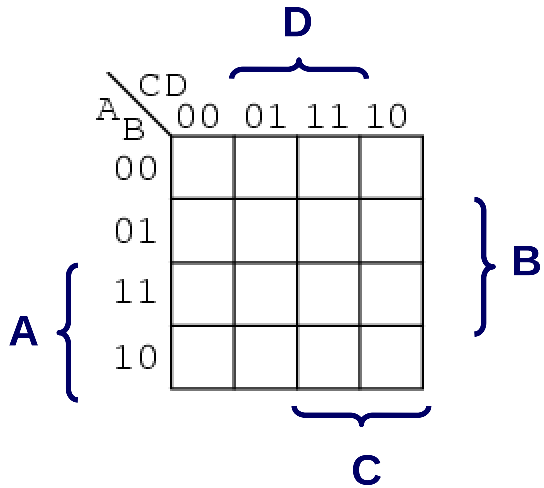
Back to our earlier example…..



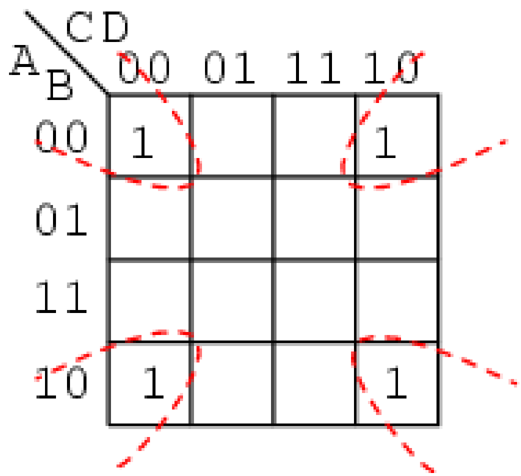
The K-map and the algebraic produce the same result.

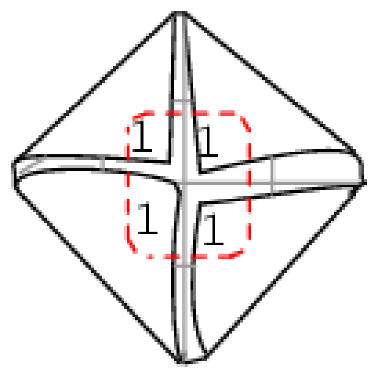
22

Up… up… and let's keep going



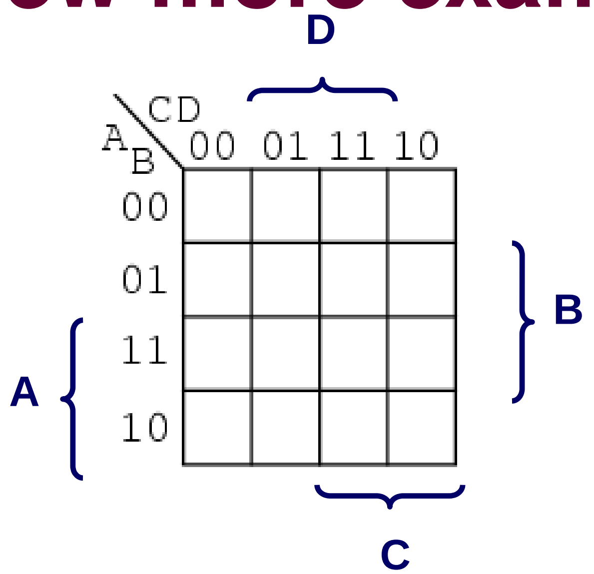
$$Out = \overline{A}\,\overline{B}\,\overline{C}\,\overline{D} + \overline{A}\,\overline{B}\,C\,\overline{D} + A\,\overline{B}\,\overline{C}\,\overline{D} + A\,\overline{B}\,C\,\overline{D}$$



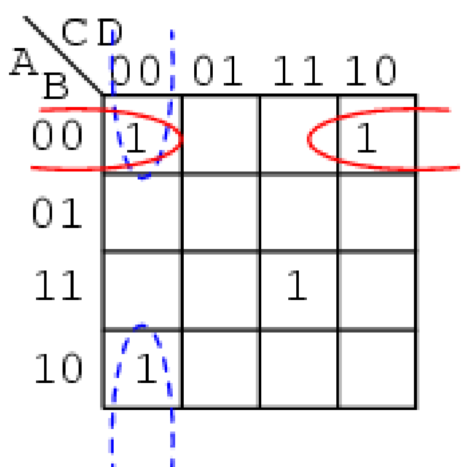


$$Out = \overline{B}\,\overline{D}$$

Few more examples

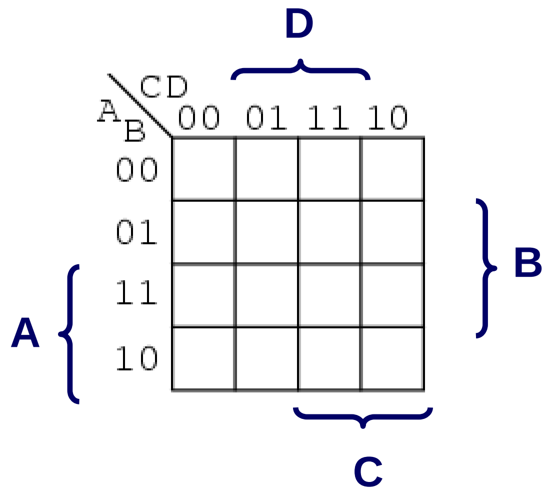


$$\text{Out} = \bar{A}\bar{B}\bar{C}\bar{D} + \bar{A}\bar{B}C\bar{D} + A\bar{B}\bar{C}\bar{D} + ABCD$$

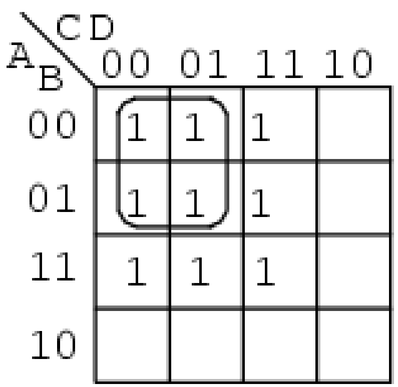


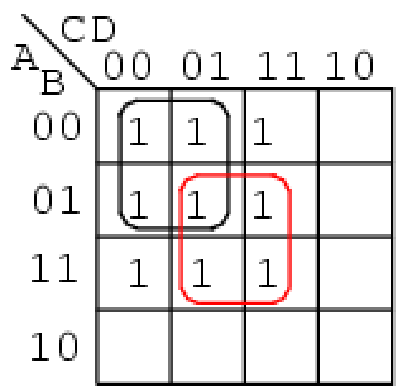
$$\text{Out} = \bar{B}\bar{C}\bar{D} + \bar{A}\bar{B}\bar{D} + ABCD$$

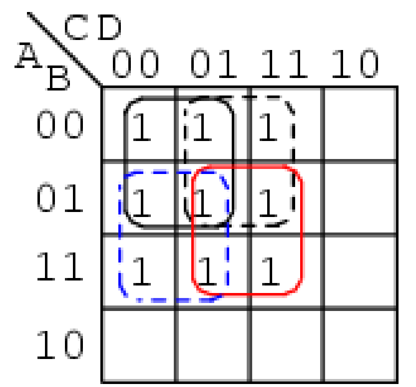
Few more examples



$$Out = \bar{A}\bar{B}\bar{C}\bar{D} + \bar{A}\bar{B}\bar{C}D + \bar{A}\bar{B}CD + \bar{A}B\bar{C}\bar{D} + \bar{A}B\bar{C}D + \bar{A}BCD + AB\bar{C}\bar{D} + AB\bar{C}D + ABCD$$







$$Out = \bar{A}\bar{C} + \bar{A}D + B\bar{C} + BD$$

Don't Care Conditions

- Let F = $AB + \overline{A}\overline{B}$
- Suppose we know that a disallowed input combo is A=1, B=0
- Can we replace F with a simpler function G whose output matches for all inputs we do care about?
- Let H be the function with Don't-care conditions for obsolete inputs

A	B	F	H	G
0	0	1	1	1
0	1	0	0	0
1	0	0	X	1
1	1	1	1	1

Inputs will not occur

$$G = AB + \overline{B}$$

- Both F & G are appropriate functions for H
- G can substitute for F for valid input combinations

Don't Cares can Greatly Simplify Circuits

Sometimes "don't cares" greatly simplify circuitry

		D	D		
	1	X	X	X	
	X	1	X	X	B
A	0	0	1	X	B
A	0	0	X	1	
			C	C	

$\overline{A}\,\overline{B}\,\overline{C}\,\overline{D} + \overline{A}B\overline{C}D + ABCD + A\overline{B}C\overline{D}$ vs. $\overline{A} + C$

Formal Definition of Minterms

e.g., Minterms for 3 variables A,B,C

A	B	C	minterm
0	0	0	m0 $\bar{A}\bar{B}\bar{C}$
0	0	1	m1 $\bar{A}\bar{B}C$
0	1	0	m2 $\bar{A}B\bar{C}$
0	1	1	m3 $\bar{A}BC$
1	0	0	m4 $A\bar{B}\bar{C}$
1	0	1	m5 $A\bar{B}C$
1	1	0	m6 $AB\bar{C}$
1	1	1	m7 ABC

- A product term in which all variables appear once, either complemented or uncomplemented (i.e., an entry in the truth table).
- Each minterm evaluates to 1 for exactly one variable assignment, 0 for all others.
- Denoted by mX where X corresponds to the variable assignment for which mX = 1.

Minterm Example

(variables appear once in each minterm)

A	B	C	F	$\overline{F}$	minterm
0	0	0	1	0	m0 $\overline{A}\,\overline{B}\,\overline{C}$
0	0	1	1	0	m1 $\overline{A}\,\overline{B}C$
0	1	0	1	0	m2 $\overline{A}B\overline{C}$
0	1	1	0	1	m3 $\overline{A}BC$
1	0	0	1	0	m4 $A\overline{B}\,\overline{C}$
1	0	1	1	0	m5 $A\overline{B}C$
1	1	0	0	1	m6 $AB\overline{C}$
1	1	1	0	1	m7 ABC

$$F = \overline{A}\,\overline{B}\,\overline{C} + \overline{A}\,\overline{B}C + \overline{A}B\overline{C} + A\overline{B}\,\overline{C} + A\overline{B}C$$

$$= m0 + m1 + m2 + m4 + m5$$

$$= \sum m(0,1,2,4,5)$$

$$\overline{F} = \overline{A}BC + AB\overline{C} + ABC$$

$$= m3 + m6 + m7$$

$$= \sum m(3,6,7)$$

Minterm Example

Circuits may be specified in the form $\sum \mathrm{m}(0, 2, 4, 5, 6, 7, 13, 15)$, where the indices correspond to positions in a Karnaugh map:

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

Minterm Example

Circuits may be specified in the form $\sum \mathrm{m}(0, 2, 4, 5, 6, 7, 13, 15)$, where the indices correspond to positions in a Karnaugh map:

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

Formal Definition of Maxterms

A	B	C	maxterm	
0	0	0	M0	$A+B+C$
0	0	1	M1	$A+B+\overline{C}$
0	1	0	M2	$A+\overline{B}+C$
0	1	1	M3	$A+\overline{B}+\overline{C}$
1	0	0	M4	$\overline{A}+B+C$
1	0	1	M5	$\overline{A}+B+\overline{C}$
1	1	0	M6	$\overline{A}+\overline{B}+C$
1	1	1	M7	$\overline{A}+\overline{B}+\overline{C}$

- A sum term in which all variables appear once, either complemented or uncomplemented.
- Each maxterm evaluates to 0 for exactly one variable assignment, 1 for all others.
- Denoted by MX where X corresponds to the variable assignment for which MX = 0.

Maxterm Example

A	B	C	maxterm	F
0	0	0	M0 $A+B+C$	1
0	0	1	M1 $A+B+\overline{C}$	1
0	1	0	M2 $A+\overline{B}+C$	1
0	1	1	M3 $A+\overline{B}+\overline{C}$	0
1	0	0	M4 $\overline{A}+B+C$	1
1	0	1	M5 $\overline{A}+B+\overline{C}$	1
1	1	0	M6 $\overline{A}+\overline{B}+C$	0
1	1	1	M7 $\overline{A}+\overline{B}+\overline{C}$	0

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

$$= (M3)\ (M6)\ (M7)$$

$$= \prod M(3,6,7)$$

Maxterm Example

Then we can find the usual sum of products:

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

$$F = BD + \bar{A}\bar{D}$$

Maxterm Example

Or the product of sums:

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

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

Converting Between Canonical Forms

	F	$\overline{F}$
Minterms (SOP)	$\sum m(F = 1)$	$\sum m(F = 0)$
Maxterms (POS)	$\prod M(F = 0)$	$\prod M(F = 1)$

DeMorgans: same terms

Product of Sums Example

WX \ YZ	0 0	0 1	1 1	1 0
0 0	1	1	0	1
0 1	1	0	0	0
1 1	1	0	0	0
1 0	1	1	0	1

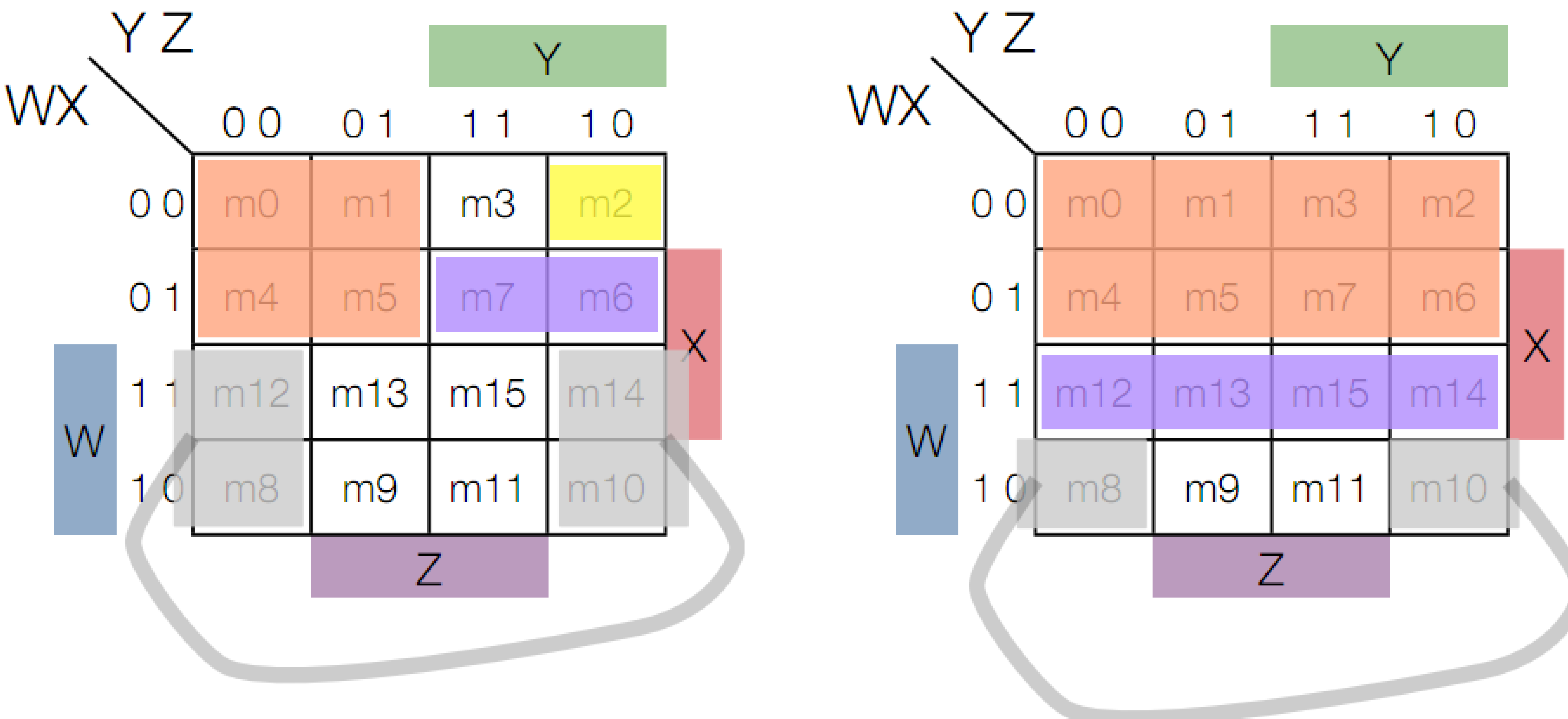
$$\overline{F} = YZ + XZ + YX$$

DeMorgan's

$$F = (\overline{Y}+\overline{Z})(\overline{Z}+\overline{X})(\overline{Y}+\overline{X})$$

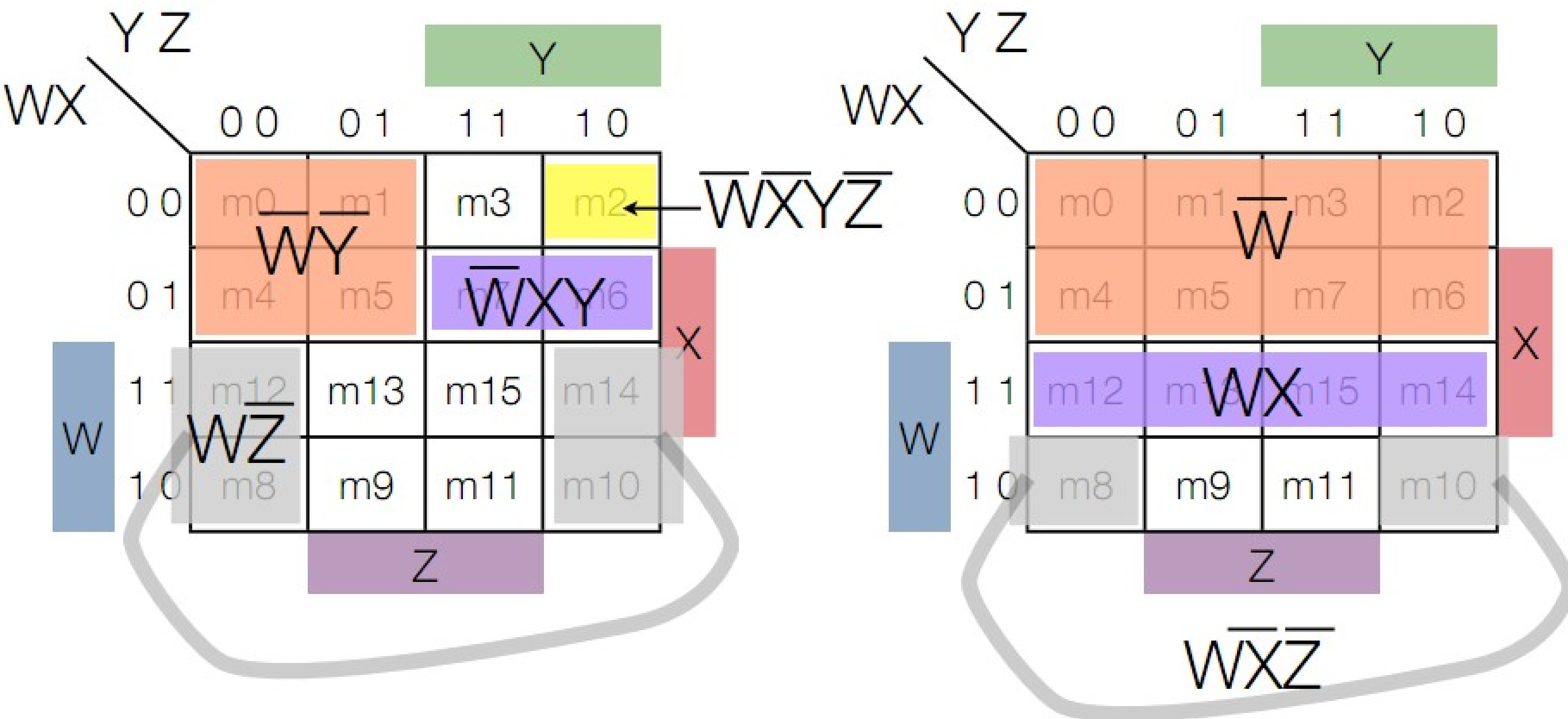
K-maps and Implicants

- **Implicant:** a product term, which, viewed in a K-Map is a $2^i \times 2^j$ size "rectangle" (possibly wrapping around) where i=0,1,2, j=0,1,2



Implicants

- **Implicant:** a product term, which, viewed in a K-Map is a $2^i \times 2^j$ size "rectangle" (possibly wrapping around) where i=0,1,2, j=0,1,2



Note: bigger rectangles = fewer literals

More Implicant Terminology

Implicant: product term, which when viewed in a K-map, is a rectangle of 1s

Prime implicant: an implicant not contained in another implicant

Essential prime implicant: a prime implicant that is the only prime implicant to cover some minterm

Example

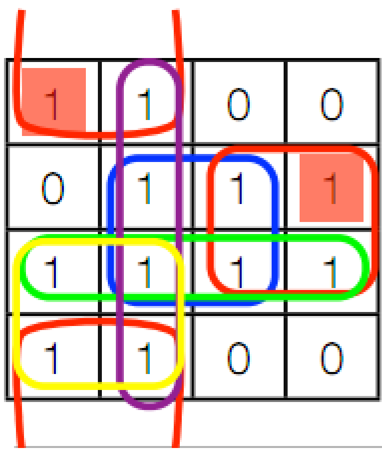
- List all of the prime implicants for this function
- Is any of them an essential prime implicant?
- What is a simplified expression for this function?

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

Example

- Step 1: Identify all PIs and essential PIs
- Step 2: Include all Essential PIs in the circuit (Why?)
- Step 3: If any 1-valued minterms are uncovered by EPIs, choose PIs that are "big" and do a good job covering
- Selection Rule: a heuristic for usually choosing "good" PIs: choose the PIs that minimize overlap with one another and with EPIs

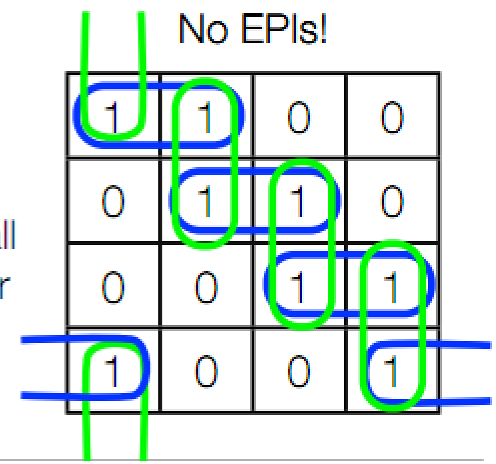
Red bounds are EPIs (solo-covered minterm shown in red)



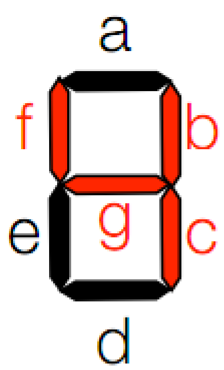
Also need (purple or blue) and (yellow or green)

All blue PIs or all green PIs cover

No EPIs!



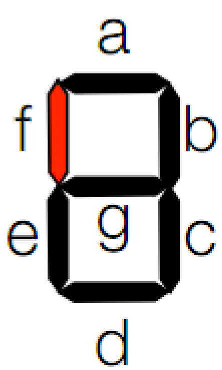
Design Example



Input					Output						
Va	W	X	Y	Z	a	b	c	d	e	f	g
0	0	0	0	0	1	1	1	1	1	1	0
1	0	0	0	1	0	1	1	0	0	0	0
2	0	0	1	0	1	1	0	1	1	0	1
3	0	0	1	1	1	1	1	1	0	0	1
4	0	1	0	0	0	1	1	0	0	1	1
5	0	1	0	1	1	0	1	1	0	1	1
6	0	1	1	0	1	0	1	1	1	1	1
7	0	1	1	1	1	1	1	0	0	0	0
8	1	0	0	0	1	1	1	1	1	1	1
9	1	0	0	1	1	1	1	0	0	1	1
X	1	0	1	0	X	X	X	X	X	X	X
X	1	0	1	1	X	X	X	X	X	X	X
X	1	1	0	0	X	X	X	X	X	X	X
X	1	1	0	1	X	X	X	X	X	X	X
X	1	1	1	0	X	X	X	X	X	X	X
X	1	1	1	1	X	X	X	X	X	X	X

e.g., what outputs "lights up" when input V=4?

Design Example



For what values does output f "light up" for?

Y=1

1	0	0	0
1	1	0	1
X	X	X	X
1	1	X	X

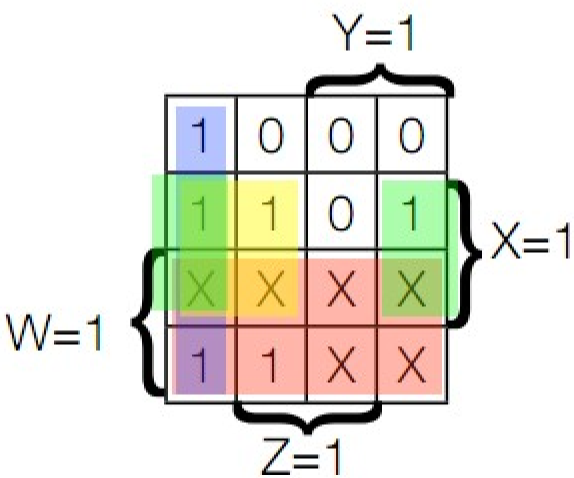
X=1

W=1

Z=1

	Input				Output						
Va	W	X	Y	Z	a	b	c	d	e	f	g
0	0	0	0	0	1	1	1	1	1	1	0
1	0	0	0	1	0	1	1	0	0	0	0
2	0	0	1	0	1	1	0	1	1	0	1
3	0	0	1	1	1	1	1	1	0	0	1
4	0	1	0	0	0	1	1	0	0	1	1
5	0	1	0	1	1	0	1	1	0	1	1
6	0	1	1	0	1	0	1	1	1	1	1
7	0	1	1	1	1	1	1	0	0	0	0
8	1	0	0	0	1	1	1	1	1	1	1
9	1	0	0	1	1	1	1	0	0	1	1
X	1	0	1	0	X	X	X	X	X	X	X
X	1	0	1	1	X	X	X	X	X	X	X
X	1	1	0	0	X	X	X	X	X	X	X
X	1	1	0	1	X	X	X	X	X	X	X
X	1	1	1	0	X	X	X	X	X	X	X
X	1	1	1	1	X	X	X	X	X	X	X

Design Example



$$f = W + \overline{Y}\,\overline{Z} + X\overline{Z} + \overline{X}\,\overline{Y} = W + (X+\overline{Y})\overline{Z} + \overline{X}\,\overline{Y}$$