CALIFORNIA STATE UNIVERSITY FULLERTON™

EGEC 281: Designing with VHDL

Fall 2024

Lecture 5: Boolean Algebra and K-Map

Rakesh Mahto, Ph.D.
***Office**: E 314, California State University, Fullerton*
***Office Hours**: Monday and Wednesday 2:00 - 3:30 pm*
Or by appointment
***Office Hour Zoom Meeting ID**: 894 4126 5483*
***Email**: ramahto@fullerton.edu*
***Phone No**: 657-278-7274*

Example Problem #1

- Lets evaluate XY+Z

X	Y	Z	XY	XY+Z
0	0	0	0.0 = 0	0+0 = 0
0	0	1	0.0 = 0	0+1 = 1
0	1	0	0.1 = 0	0+0 = 0
0	1	1	0.1 = 0	0+1 = 1
1	0	0	1.0 = 0	0+0 = 0
1	0	1	1.0 = 0	0+1 = 1
1	1	0	1.1 = 1	1+0 = 1
1	1	1	1.1 =1	1+1 = 1

Practice Problem #1

- Lets evaluate X+YZ’

X	Y	Z	Z’	YZ’	X+YZ’
0	0	0	1	0.1 = 0	0+0 = 0
0	0	1	0	0.0 = 0	0+0 = 0
0	1	0	1	1.1 = 1	0+1 = 1
0	1	1	0	1.0 = 0	0+0 = 0
1	0	0	1	0.1 = 0	1+0 = 1
1	0	1	0	0.0 = 0	1+0 = 1
1	1	0	1	1.1 = 1	1+1 = 1
1	1	1	0	1.0 = 0	1+0 = 1

Practice Problem #2

- Is XY+Z = X+YZ’

XY+Z		X+YZ’
0+0 = 0		0+0 = 0
0+1 = 1	⟺	0+0 = 0
0+0 = 0	⟺	0+1 = 1
0+1 = 1	⟺	0+0 = 0
0+0 = 0	⟺	1+0 = 1
0+1 = 1		1+0 = 1
1+0 = 1		1+1 = 1
1+1 = 1		1+0 = 1

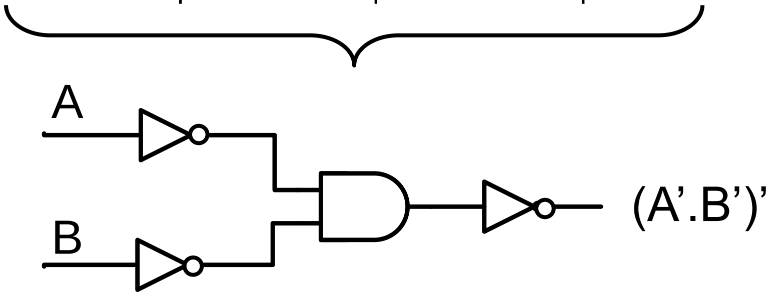
So the answer is NO

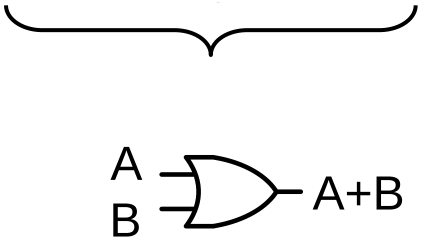
Example Problem #2

- Is A+B = (A’.B’)’?
- Step 1 Complete the truth table for A+B
- St Ta B’)’

A	B	A’	B’	A’.B’	(A’.B’)’
0	0	1	1	1.1 = 1	0
0	1	1	0	1.0 = 0	1
1	0	0	1	0.1 = 0	1
1	1	0	0	0.0 = 0	1

A	B	A+B
0	0	0
0	1	1
1	0	1
1	1	1



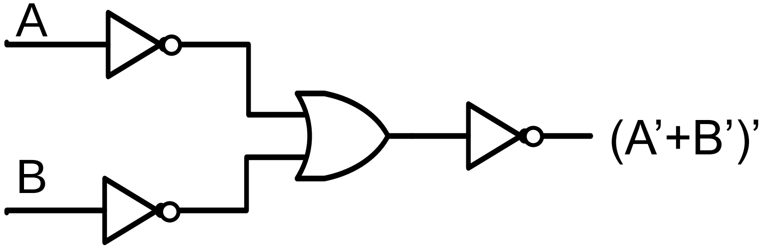


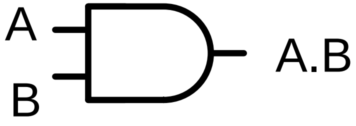
Practice Problem # 3

- Is AB = (A’+B’)’

A	B	A’	B’	A’+B’	(A’+B’)’
0	0	1	1	1+1 = 1	0
0	1	1	0	1+0 = 1	0
1	0	0	1	0+1 = 1	0
1	1	0	0	0+0 = 0	1

A	B	A.B
0	0	0
0	1	0
1	0	0
1	1	1





Practice Problem # 4

- What is the output for the function (X+X’).Y

X	Y	X’	X+X’	(X+X’).Y
0	0	1	0+1 = 1	1.0=0
0	1	1	0+1 = 1	1.1=1
1	0	0	1+0 = 1	1.0=0
1	1	0	1+0 = 1	1.1=1

Why implement the circuit It does not change Y?

Practice Problem #1

Given the following truth table write the truth function for F1?

A	B	C	F1
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

	A	B	C	F1
A’B’C’	0	0	0	0
A’B’C	0	0	1	0
A’BC’	0	1	0	0
A’BC	0	1	1	0
AB’C’	1	0	0	1
AB’C	1	0	1	1
ABC’	1	1	0	1
ABC	1	1	1	0

Truth Function

F1 = AB’C’+ AB’C+ ABC’

Practice Problem #2

Given the following truth table write the truth function for F2?

A	B	C	F2
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	0

	A	B	C	F2
A’B’C’	0	0	0	1
A’B’C	0	0	1	1
A’BC’	0	1	0	1
A’BC	0	1	1	1
AB’C’	1	0	0	0
AB’C	1	0	1	0
ABC’	1	1	0	1
ABC	1	1	1	0

Truth Function

F2 = A’B’C’+ A’B’C
+ A’BC’ + A’BC
+ABC’

Practice Problem #3

Given the following truth table write the truth function for F1?

A	B	C	D	F3
0	0	0	0	1
0	0	0	1	1
0	0	1	0	1
0	0	1	1	1
0	1	0	0	0
0	1	0	1	0
0	1	1	0	1
0	1	1	1	0
1	0	0	0	1
1	0	0	1	1
1	0	1	0	1
1	0	1	1	1
1	1	0	0	0
1	1	0	1	0
1	1	1	0	1
1	1	1	1	0

	A	B	C	D	F3
A’B’C’D’	0	0	0	0	1
A’B’C’D	0	0	0	1	1
A’B’CD’	0	0	1	0	1
A’B’CD	0	0	1	1	1
A’BC’D’	0	1	0	0	0
A’BC’D	0	1	0	1	0
A’BCD’	0	1	1	0	1
A’BCD	0	1	1	1	0
AB’C’D’	1	0	0	0	1
AB’C’D	1	0	0	1	1
AB’CD’	1	0	1	0	1
AB’CD	1	0	1	1	1
ABC’D’	1	1	0	0	0
ABC’D	1	1	0	1	0
ABCD’	1	1	1	0	1
ABCD	1	1	1	1	0

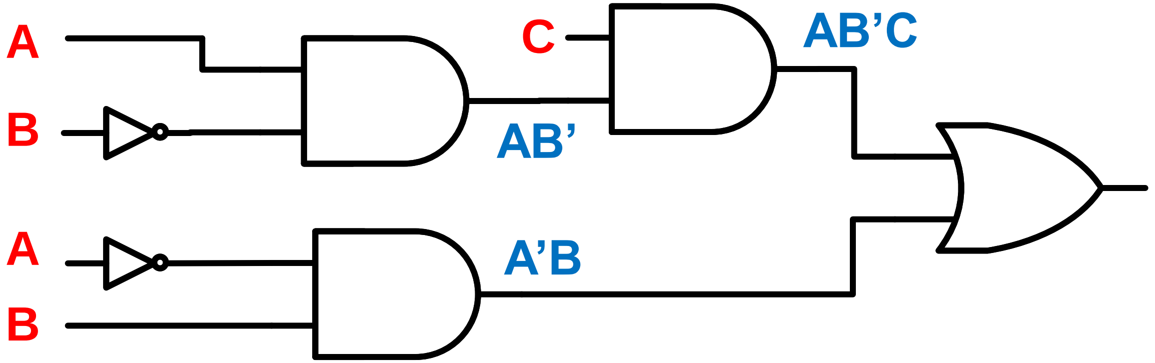
Truth Function

F3 = A’B’C’D’ + A’B’C’D
+ A’B’CD’ + A’B’CD
+ A’BCD’ + AB’C’D
+ AB’CD’ + ABCD’

Expressions/Boolean Functions

- A literal means we need to provide a wire to move the signal to the gate.
  - Each appearance of a variable or its complement in an expression will be referred to as a literal.
  - Thus, each literal in an expression corresponds to a gated input.

Gates are the Boolean operations



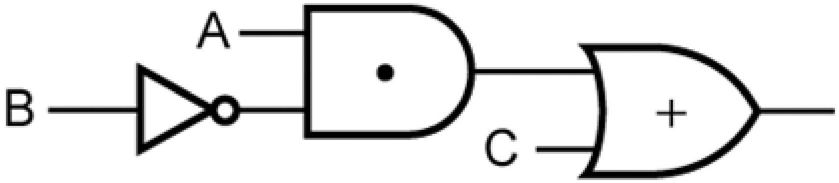
Truth Function

F = AB'C+A'B

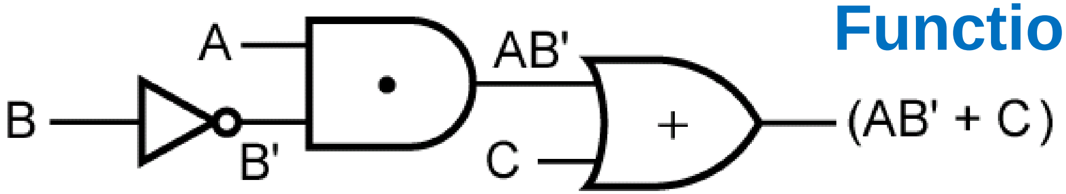
AB' C+A' B: 3 variables, 5 literals, 6 Gates

Practice Problem #4

What is this systems SOM Function and What is the number of variable, literals, and gates?



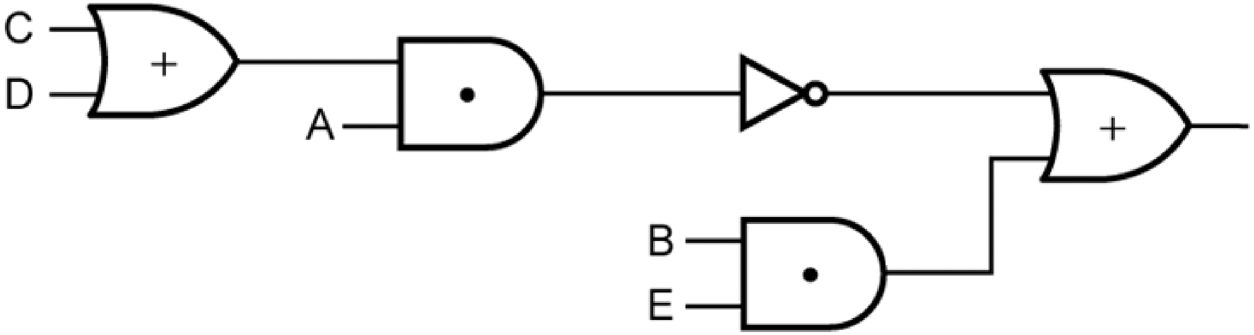
Truth Function

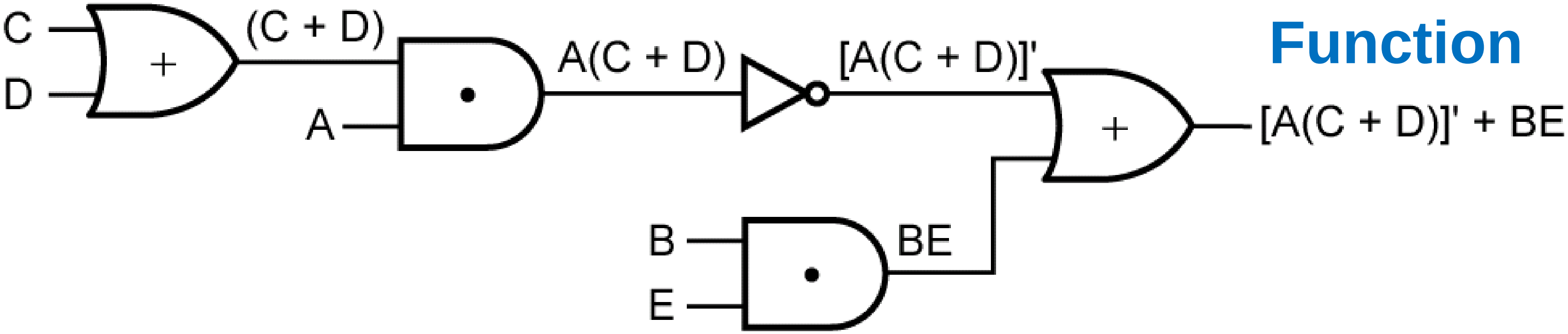


AB’+C: 3 variables, 3 literals, 3 Gates

Practice Problem #5

What is this systems SOM Function and What is the number of variable, literals, and gates?





Truth Function

(A(C+D)’+BE: 5 variables, 5 literals, 5 Gates

Practice Problem #6

Given the following truth table Draw the Gate Logic?

A	B	C	F1
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

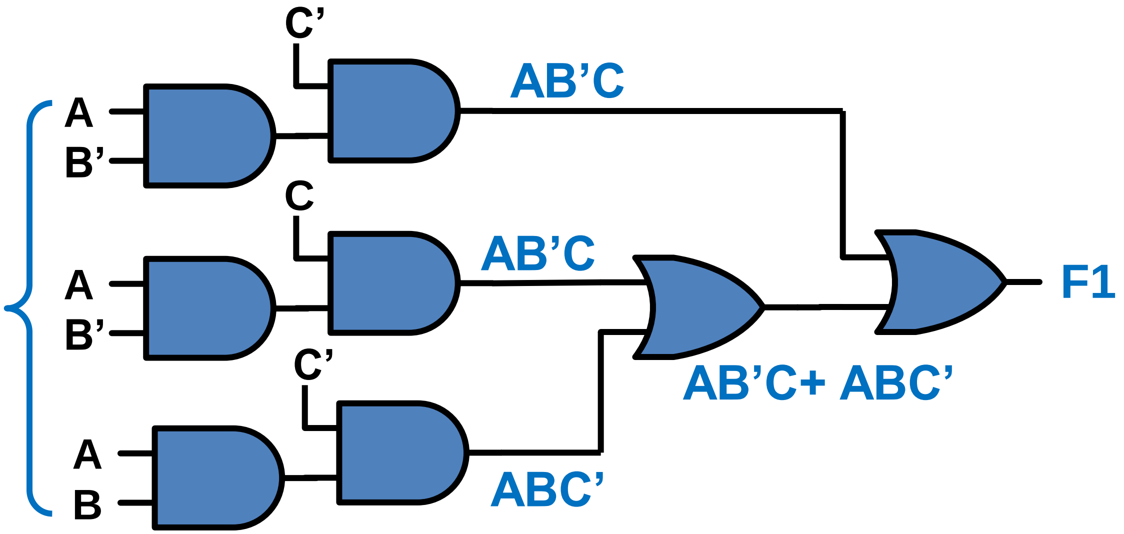
	A	B	C	F1
A’B’C’	0	0	0	0
A’B’C	0	0	1	0
A’BC’	0	1	0	0
A’BC	0	1	1	0
AB’C’	1	0	0	1
AB’C	1	0	1	1
ABC’	1	1	0	1
ABC	1	1	1	0

Truth Function

F1 = AB’C’+ AB’C+ ABC’

Truth Function

F1 = AB’C’+ AB’C+ ABC’



Practice Problem #7

Given the following truth table write the truth function for F2?

A	B	C	F2
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	0

→

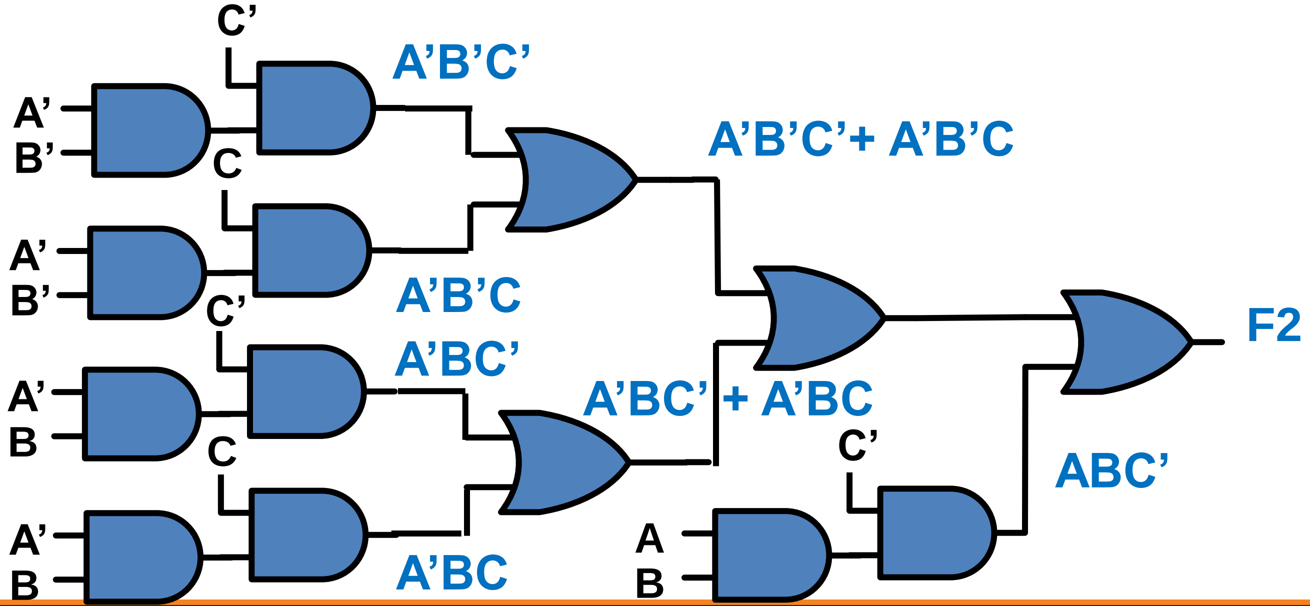
	A	B	C	F2
A’B’C’	0	0	0	1
A’B’C	0	0	1	1
A’BC’	0	1	0	1
A’BC	0	1	1	1
AB’C’	1	0	0	0
AB’C	1	0	1	0
ABC’	1	1	0	1
ABC	1	1	1	0

Truth Function

F2 = A’B’C’+ A’B’C
+ A’BC’ + A’BC
+ ABC’

Truth Function

F2 = A’B’C’+ A’B’C
+ A’BC’ + A’BC’
+ABC’



Why Worry about Minterm and Maxterms

Boolean functions can generally be simplified by using the algebraic techniques. However, two problems arise when algebraic procedures are used:

1. **The procedures are difficult to apply in a systematic way.**
2. **It is difficult to tell when you have arrived at a minimum solution.**

The Karnaugh map method is generally faster and easier to apply than other simplification methods.

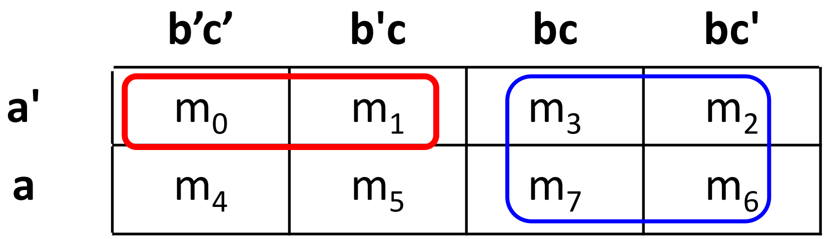
Karnaugh Maps

	b’c’	b'c	bc	bc'
a'	m_0	m_1	m_3	m_2
a	m_4	m_5	m_7	m_6

Want to combine the largest number of cells possible to eliminate as many variables as possible, implies that:

Combine 2 cells, eliminate 1 variable
Combine 4 cells, eliminate 2 variables
Combine 2^n, eliminate n variables

a’b’c’+a’b’c
a’b’(c’+c)
a’b’(1)
a’b’



	b’c’	b'c	bc	bc'
a'	m_0	m_1	m_3	m_2
a	m_4	m_5	m_7	m_6

a’bc+a’bc’+abc+abc’
a’(bc+bc’)+a(bc+bc’)
(a’+a)(bc’+bc)
b (c+c’)
b

Minterm and Maxterm Expansions

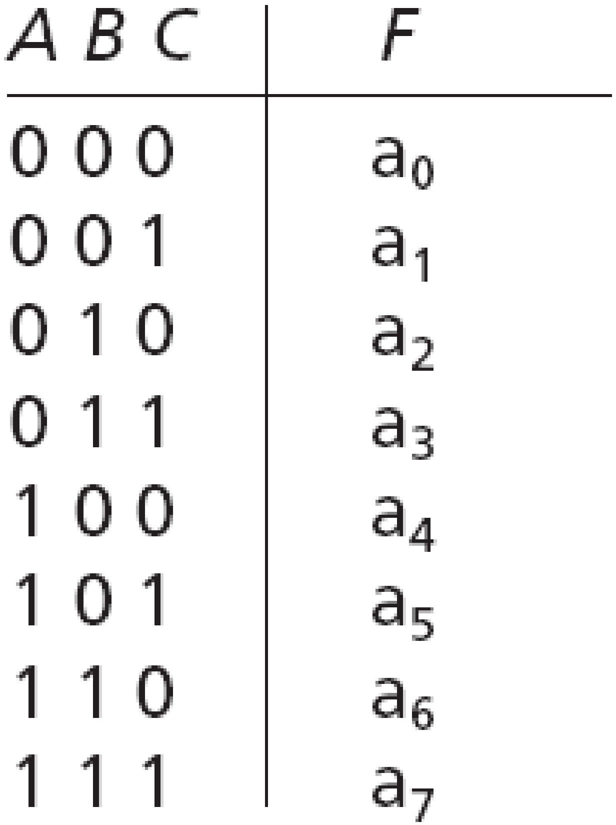
$$f = A'BC + AB'C' + AB'C + ABC' + ABC$$

Each of the terms in the above equation is referred to as a minterm. In general, a *minterm* of *n* variables is a product of *n literals* (A, A’, B, B’, C, and C’) in which each variable appears exactly once in either true or complemented form, but not both.

In General a *literal* is a variable or its complement

General Truth Table for Three Variables

Table represents a truth table for a general function of three variables. Each a_i is a constant with a value of 0 or 1.



A B C	F
0 0 0	a_0
0 0 1	a_1
0 1 0	a_2
0 1 1	a_3
1 0 0	a_4
1 0 1	a_5
1 1 0	a_6
1 1 1	a_7

$$F = a_0 m_0 + a_1 m_1 + a_2 m_2 + \cdots + a_7 m_7 = \sum_{i=0}^{7} a_i m_i$$

General Minterm and Maxterm Expansions

We can write the minterm expansion for a general function is as follows; $a_i = 1$ or 0: (Don't forget when $a_i = 0$, a term to dropout since $0.m_i = 0$)

$$F = a_0m_0 + a_1m_1 + a_2m_2 + \cdots + a_7m_7 = \sum_{i=0}^{n} a_im_i \qquad (4\text{-}12)$$

$$m_0 \Leftrightarrow A'B' or\ A'B'C' or\ A'B'C'D', \qquad \cdots \qquad , \qquad m_n \Leftrightarrow AB or\ ABC or\ ABCD$$

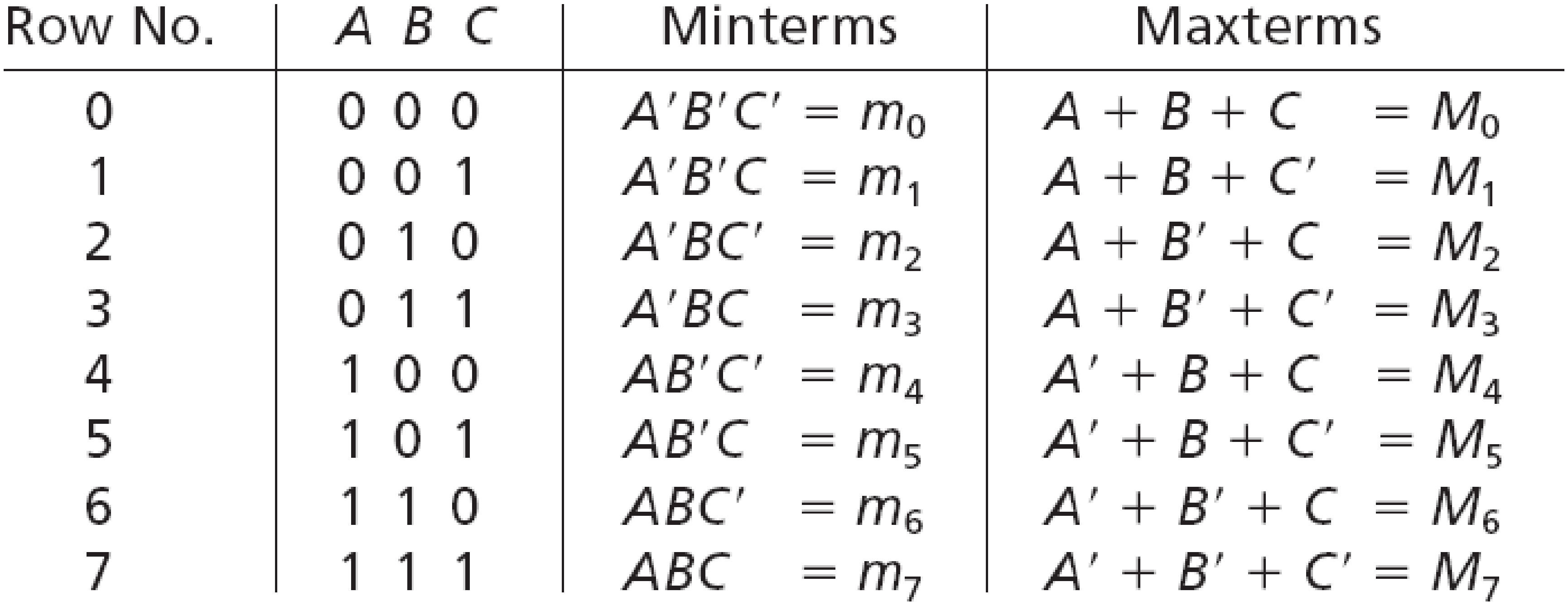
The maxterm expansion for a general function variables is; $a_i = 1$ or 0: (Don't forget when $a_i = 1$, a term to dropout since $1+m_i = 1$ and $1.(\,..) = (\,..\,)$)

$$F = (a_0 + M_0)(a_1 + M_1)(a_2 + M_2) \cdots (a_7 + M_7) = \prod_{i=0}^{n} (a_i + M_i) \qquad (4\text{-}13)$$

$$M_0 \Leftrightarrow A' + B'\ or\ \ A' + B' + C'\ \ or\ \ A' + B' + C' + D', \qquad \cdots$$

$$M_n \Leftrightarrow A + B\ or\ A + B + C\ or\ A + B + C + D$$

Minterms and Maxterms for Three Variables



Row No.	A B C	Minterms	Maxterms
0	0 0 0	$A'B'C' = m_0$	$A + B + C = M_0$
1	0 0 1	$A'B'C = m_1$	$A + B + C' = M_1$
2	0 1 0	$A'BC' = m_2$	$A + B' + C = M_2$
3	0 1 1	$A'BC = m_3$	$A + B' + C' = M_3$
4	1 0 0	$AB'C' = m_4$	$A' + B + C = M_4$
5	1 0 1	$AB'C = m_5$	$A' + B + C' = M_5$
6	1 1 0	$ABC' = m_6$	$A' + B' + C = M_6$
7	1 1 1	$ABC = m_7$	$A' + B' + C' = M_7$

Input **Output: Minterms & Maxterms**

minterm Expansions

Minterm expansion for a function is unique. Look at the equation below, it can be rewritten in terms of m-notation as:

A	B	C	F	Term	Coeff	Expansion
0	0	0	0	m_0	$a_0=0$	0.A’.B’.C’ = 0
0	0	1	0	m_1	$a_1=0$	0.A’.B’.C = 0
0	1	0	0	m_2	$a_2=0$	0.A’.B.C = 0
0	1	1	1	m_3	$a_3=1$	1.A’.B.C = A’.B.C
1	0	0	1	m_4	$a_4=1$	1.A.B’.C’ = A.B’.C’
1	0	1	1	m_5	$a_5=1$	1.A.B’.C = A.B’.C
1	1	0	1	m_6	$a_6=1$	1.A.B.C’ = A.B.C’
1	1	1	1	m_7	$a_7=1$	1.A.B.C = A.B.C

$$f(A, B, C) = 0 + 0 + 0 + A'BC + AB'C' + AB'C + ABC' + ABC$$

OR

$$f(A, B, C) = m_3 + m_4 + m_5 + m_6 + m_7$$

$$f(A, B, C) = \sum m(3, 4, 5, 6, 7)$$

Minterm Expansions

Find the *minterm* expansion of $F(a,b,c,d) = a'(b' + d) + acd' = \underbrace{a'b'} + \underbrace{a'd'} + \underbrace{acd'}.$

A	B	C	D	*F(a, b, c, d)*	Terms
0	0	0	0	1+1	m_0
0	0	0	1	1	m_1
0	0	1	0	1+1	m_2
0	0	1	1	1	m_3
0	1	0	0	1	m_4
0	1	0	1	0	M_5
0	1	1	0	1	m_6
0	1	1	1	0	M_7
1	0	0	0	0	M_8
1	0	0	1	0	M_9
1	0	1	0	1	m_{10}
1	0	1	1	0	M_{11}
1	1	0	0	0	M_{12}
1	1	0	1	0	M_{13}
1	1	1	0	1	m_{14}
1	1	1	1	0	M_{15}

These are not *minterms* since a *minterm* should have 4 literals

$$F(A, B, C, D) = \Sigma\, m(0, 1, 2, 3, 4, 6, 10, 11)$$

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

Maxterm Expansions

Maxterm expansion for a function is unique. Look at the equation below, it can be rewritten in terms of m-notation as:

A	B	C	F	Term	Coeff	Expansion
0	0	0	0	M_0	$a_0=0$	0+(A+B+C) = A+B+C
0	0	1	0	M_1	$a_1=0$	0+(A+B+C’) = A+B+C’
0	1	0	0	M_2	$a_2=0$	0+(A+B’+C) = A+B’+C
0	1	1	1	M_3	$a_3=1$	1+(A+B’+C’) = 1
1	0	0	1	M_4	$a_4=1$	1+(A’+B+C) = 1
1	0	1	1	M_5	$a_5=1$	1+(A’+B+C’) = 1
1	1	0	1	M_6	$a_6=1$	1+(A’+B’+C) = 1
1	1	1	1	M_7	$a_7=1$	1+(A’+B’+C’) = 1

$$F\,(A, B, C) = (A+B+C)(A+B+C')(A+B'+C)(1)(1)(1)(1)(1)$$

$$= (A+B+C)(A+B+C')(A+B'+C)$$

OR

$F(A, B, C) = M_0\, M_1\, M_2$ $\quad F(A, B, C) = \prod M(0, 1, 2)$

Minterm Expansions

Find the *minterm* expansion of $F(a,b,c,d) = a'(b' + d') + acd' = \underbrace{a'b'} + \underbrace{a'd'} + \underbrace{acd'}.$

A	B	C	D	*F(a, b, c, d)*	Terms
0	0	0	0	1+1	m_0
0	0	0	1	1	m_1
0	0	1	0	1+1	m_2
0	0	1	1	1	m_3
0	1	0	0	1	m_4
0	1	0	1	0	M_5
0	1	1	0	1	m_6
0	1	1	1	0	M_7
1	0	0	0	0	M_8
1	0	0	1	0	M_9
1	0	1	0	1	m_{10}
1	0	1	1	0	M_{11}
1	1	0	0	0	M_{12}
1	1	0	1	0	M_{13}
1	1	1	0	1	m_{14}
1	1	1	1	0	M_{15}

These are not *maxterms* since a *maxterm* should have 4 literals and be product of sums

$$F(A, B, C, D) = \prod M(5, 7, 8, 9, 11, 12, 13, 15)$$

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

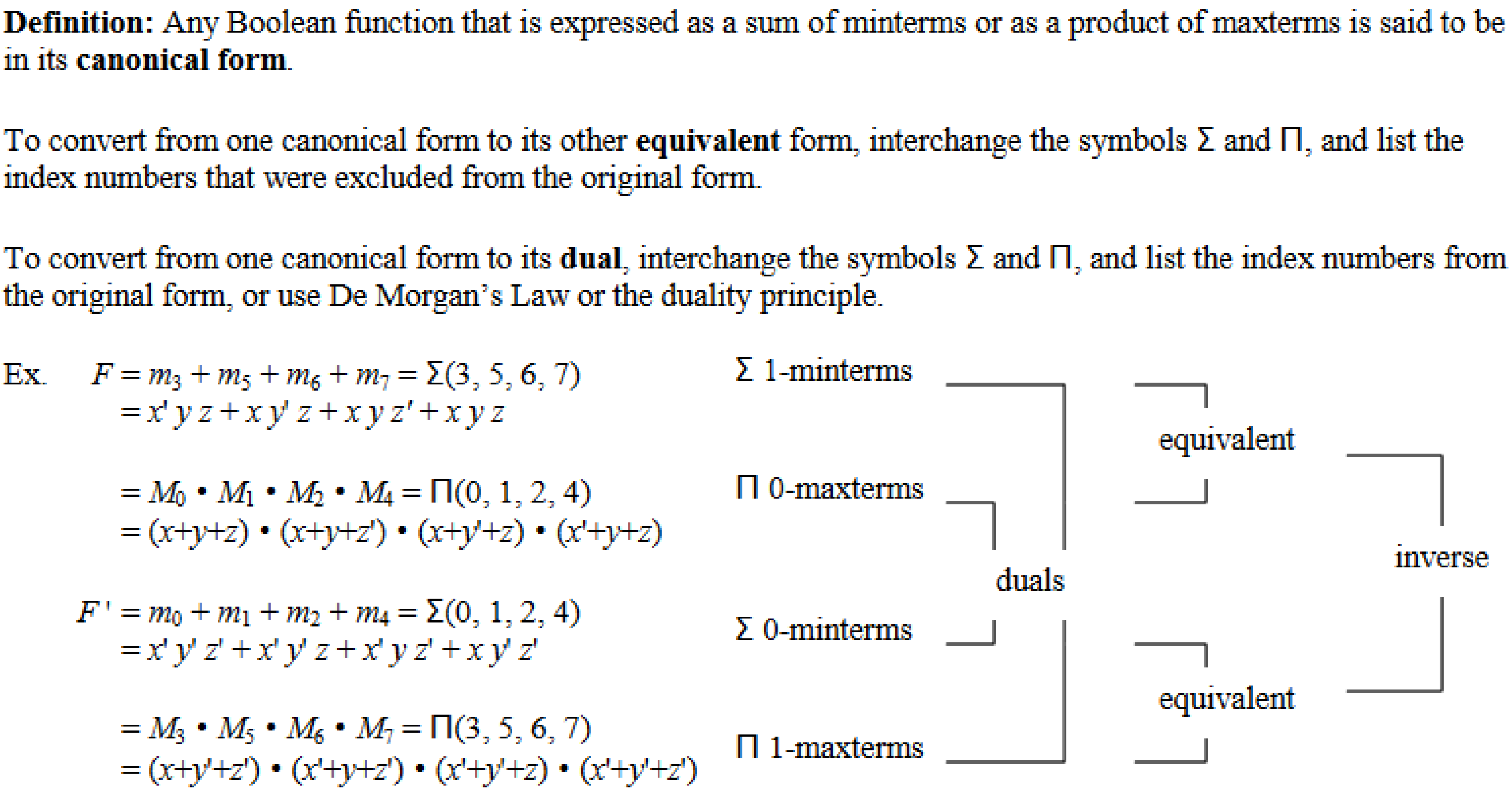
Note: Since we know

$$F(A, B, C, D) = \sum m(0, 1, 2, 3, 4, 6, 10, 11)$$

We can write,

$$F(A, B, C, D) = \prod M(5, 7, 8, 9, 11, 12, 13, 15)$$

Moving Between



Definition: Any Boolean function that is expressed as a sum of minterms or as a product of maxterms is said to be in its **canonical form**.

To convert from one canonical form to its other **equivalent** form, interchange the symbols Σ and Π, and list the index numbers that were excluded from the original form.

To convert from one canonical form to its **dual**, interchange the symbols Σ and Π, and list the index numbers from the original form, or use De Morgan's Law or the duality principle.

Ex.

$$F = m_3 + m_5 + m_6 + m_7 = \Sigma(3, 5, 6, 7)$$
$$= x'\,y\,z + x\,y'\,z + x\,y\,z' + x\,y\,z$$

$$= M_0 \cdot M_1 \cdot M_2 \cdot M_4 = \Pi(0, 1, 2, 4)$$
$$= (x+y+z) \cdot (x+y+z') \cdot (x+y'+z) \cdot (x'+y+z)$$

$$F' = m_0 + m_1 + m_2 + m_4 = \Sigma(0, 1, 2, 4)$$
$$= x'\,y'\,z' + x'\,y'\,z + x'\,y\,z' + x\,y'\,z'$$

$$= M_3 \cdot M_5 \cdot M_6 \cdot M_7 = \Pi(3, 5, 6, 7)$$
$$= (x+y'+z') \cdot (x'+y+z') \cdot (x'+y'+z) \cdot (x'+y'+z')$$

Σ 1-minterms

Π 0-maxterms

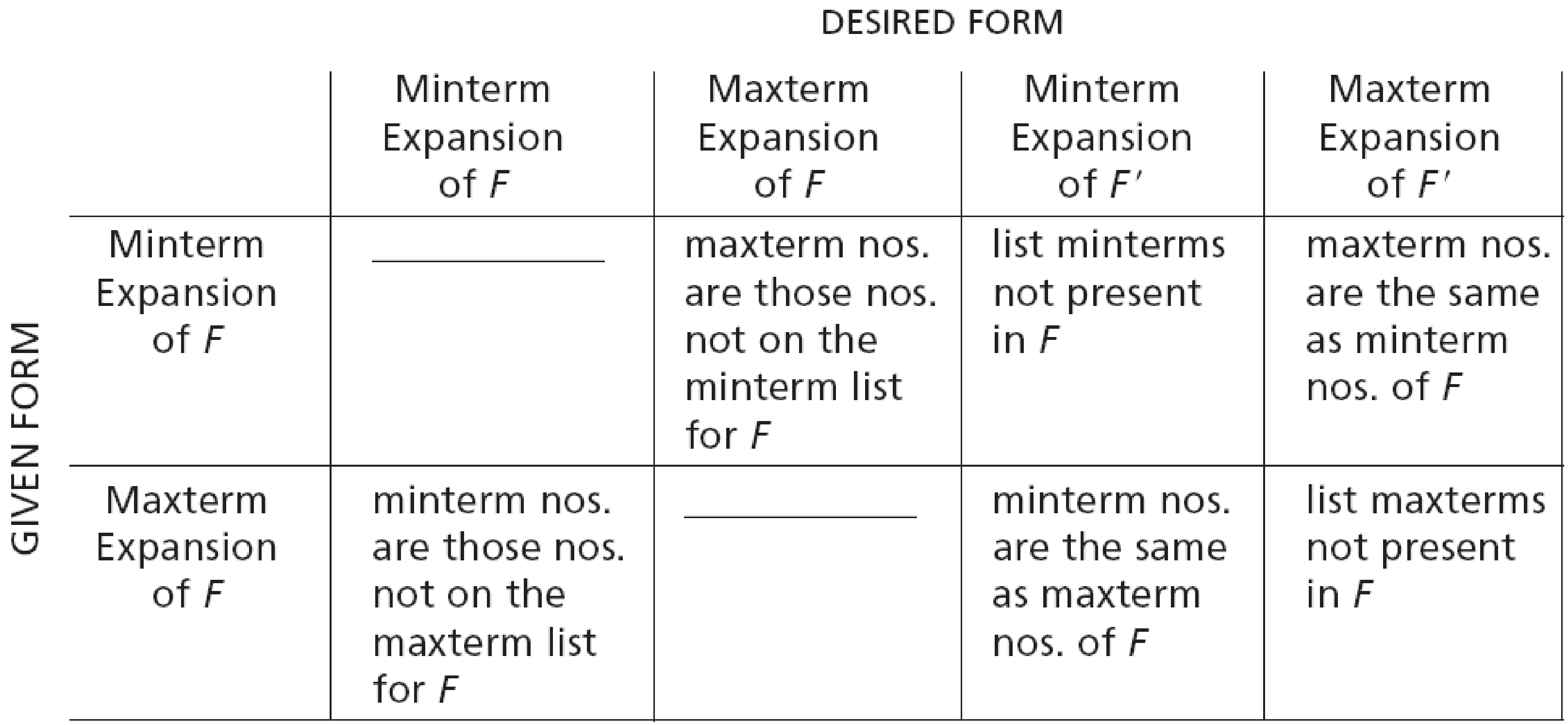
Σ 0-minterms

Π 1-maxterms

http://www.cs.ucr.edu/~ehwang/courses/cs120a/minterms.pdf

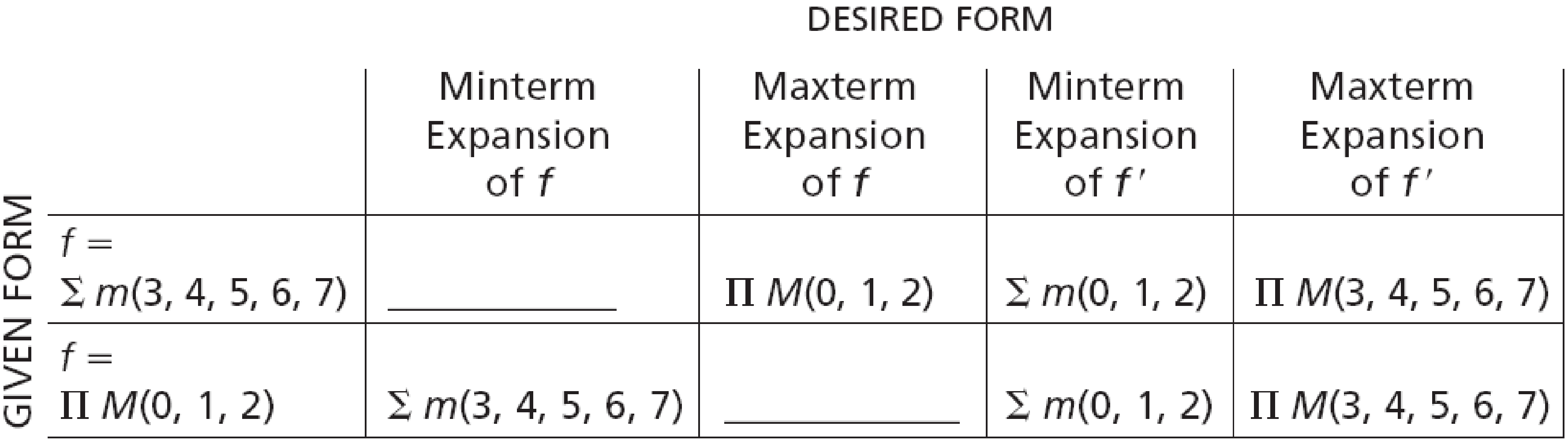
Conversion of Forms

Summarizes the procedures for conversion between minterm and maxterm expansions of F and F′



		DESIRED FORM			
		Minterm Expansion of *F*	Maxterm Expansion of *F*	Minterm Expansion of *F*′	Maxterm Expansion of *F*′
GIVEN FORM	Minterm Expansion of *F*	———	maxterm nos. are those nos. not on the minterm list for *F*	list minterms not present in *F*	maxterm nos. are the same as minterm nos. of *F*
	Maxterm Expansion of *F*	minterm nos. are those nos. not on the maxterm list for *F*	———	minterm nos. are the same as maxterm nos. of *F*	list maxterms not present in *F*

Example



		DESIRED FORM			
		Minterm Expansion of f	Maxterm Expansion of f	Minterm Expansion of f'	Maxterm Expansion of f'
GIVEN FORM	$f = \Sigma\, m(3, 4, 5, 6, 7)$	____________	$\Pi\, M(0, 1, 2)$	$\Sigma\, m(0, 1, 2)$	$\Pi\, M(3, 4, 5, 6, 7)$
	$f = \Pi\, M(0, 1, 2)$	$\Sigma\, m(3, 4, 5, 6, 7)$	____________	$\Sigma\, m(0, 1, 2)$	$\Pi\, M(3, 4, 5, 6, 7)$

Minterm Order

A	C	B	D	Terms
0	0	0	0	m_0
0	0	0	1	m_1
0	**0**	**1**	0	m_4
0	**0**	**1**	1	m_5
0	**1**	**0**	0	m_2
0	1	0	1	m_3
0	1	1	0	m_6
0	1	1	1	m_7
1	0	0	0	m_8
1	0	0	1	m_9
1	0	1	0	m_{12}
1	0	1	1	m_{13}
1	1	0	0	m_{10}
1	1	0	1	m_{11}
1	1	1	0	m_{14}
1	1	1	1	m_{15}

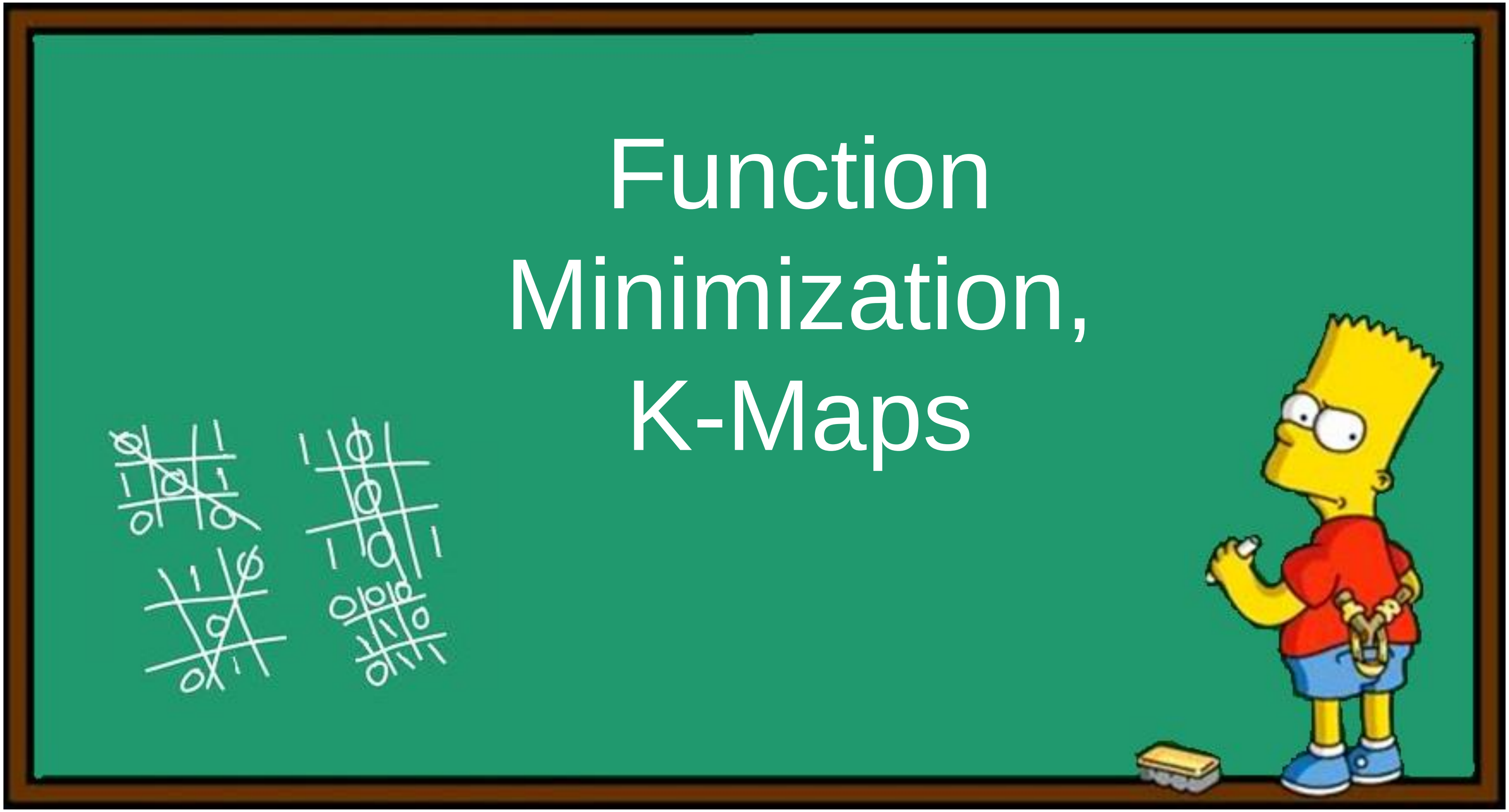
A	B	C	D
0	0	0	0
0	0	0	1
0	**1**	**0**	0
0	**1**	**0**	1
0	**0**	**1**	0
0	0	1	1
0	1	1	0
0	1	1	1
1	0	0	0
1	0	0	1
1	0	1	0
1	0	1	1
1	1	0	0
1	1	0	1
1	1	1	0
1	1	1	1

EGCP 281 convention:

Minterms are assigned based on the A, B, C, D column order only

If the equation is given as F(A, C, B, D) it is easier to lay the table out as you see on the left,

However, the minterms are assigned based on the table on the right.



Function Minimization, K-Maps

Karnaugh Maps

Switching/Boolean functions can generally be simplified by using the algebraic techniques. However, two problems arise when algebraic procedures are used:

1. The procedures are difficult to apply in a systematic way.
2. It is difficult to tell when you have arrived at a minimum solution.

The Karnaugh map method is generally faster and easier to apply than other simplification methods.

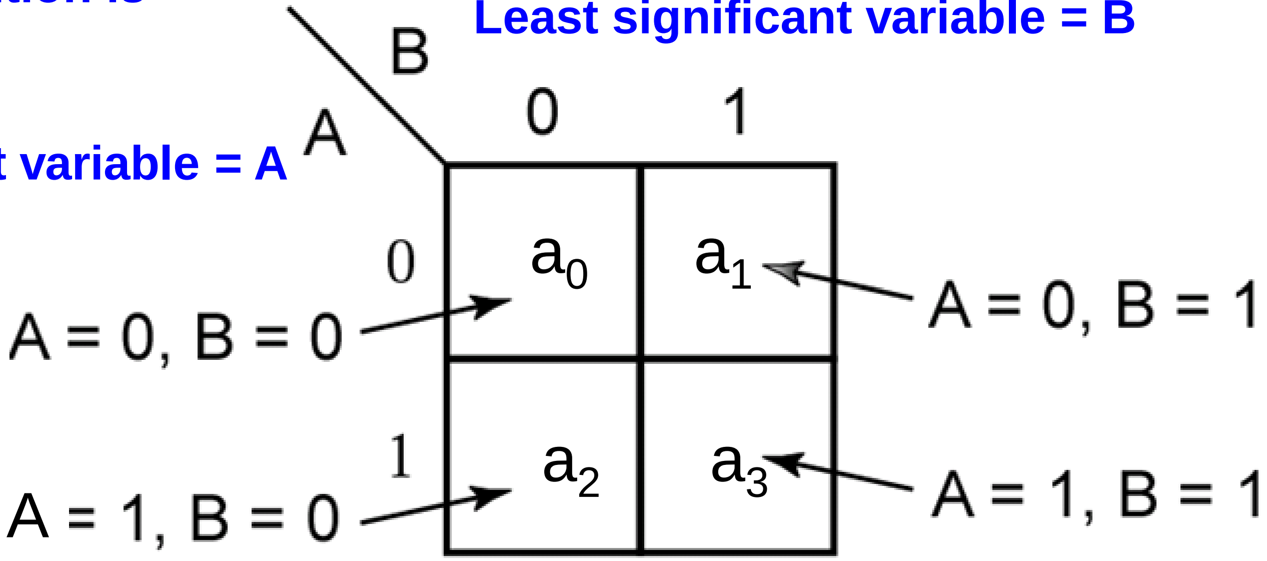
Two- Variable Karnaugh Maps

Just like a truth table, the Karnaugh map of a function specifies the value of the function for every combination of values of the independent variables.

ORDER Does not matter BUT the convention is

Least significant variable = B

Most significant variable = A



A \ B	0	1
0	a_0	a_1
1	a_2	a_3

A'B' ⇨ A = 0, B = 0

m_0

AB' ⇨ A = 1, B = 0

m_2

A = 0, B = 1 ⇦ A'B

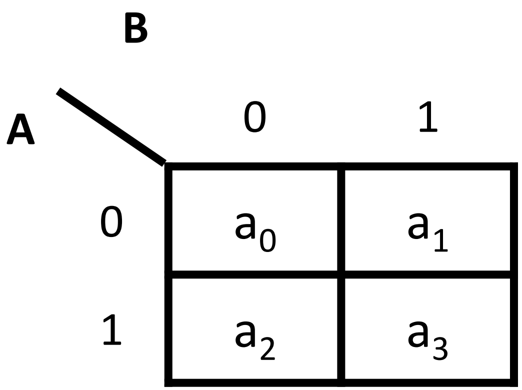
m_2

A = 1, B = 1 ⇦ AB

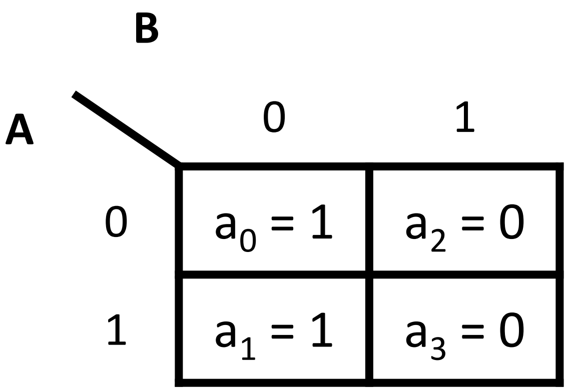
m_3

Example: Two – Variable K-map

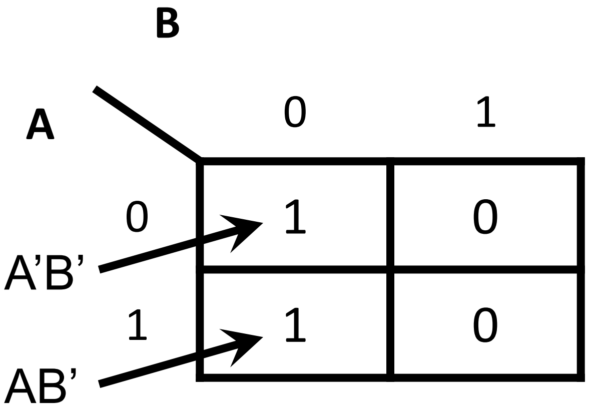
A	B	F
0	0	1
0	1	0
1	0	1
1	1	0



	0	1
0	a_0	a_1
1	a_2	a_3



	0	1
0	$a_0 = 1$	$a_2 = 0$
1	$a_1 = 1$	$a_3 = 0$



	0	1
0	1	0
1	1	0

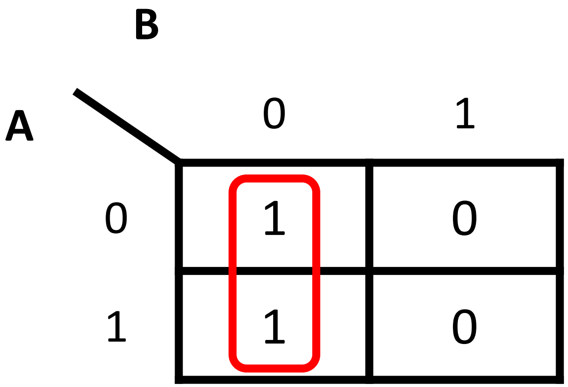
F(A,B) = A'B'+AB'

Why it works

B'(A'+A)

B'(1)

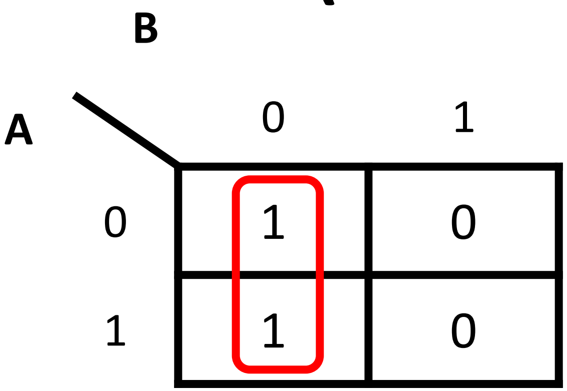
B'



	0	1
0	1	0
1	1	0

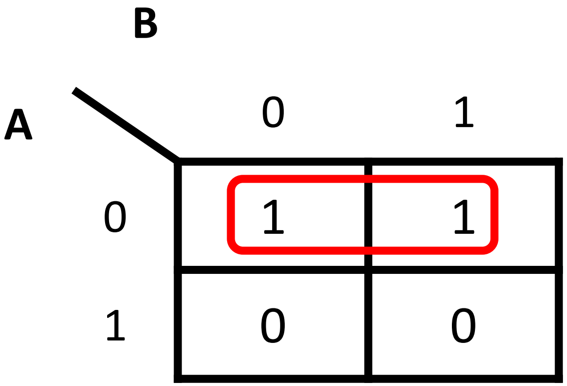
Since A changes and B' does not the answer is B'

Karnaugh Maps
(2 – Variable MINTERM Rules)



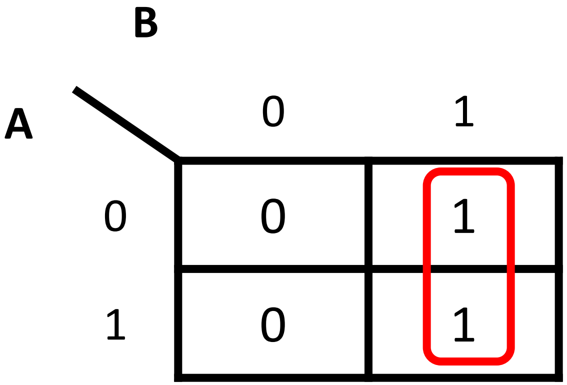
A \ B	0	1
0	1	0
1	1	0

$F(A,B) = A'B'+AB' = (A'+A)B' = B'$



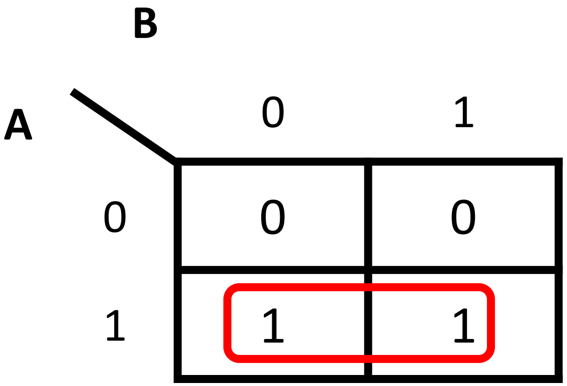
A \ B	0	1
0	1	1
1	0	0

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



A \ B	0	1
0	0	1
1	0	1

$F(A,B) = A'B+AB = (A'+A)B = B$

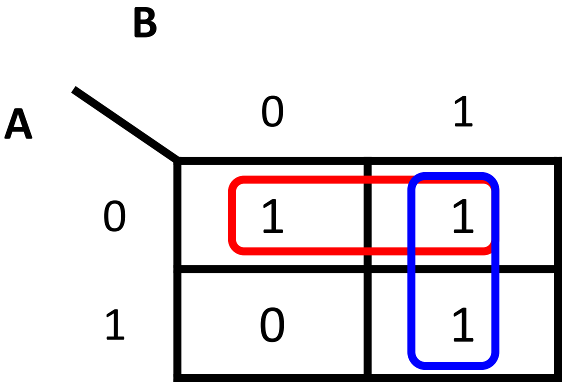


A \ B	0	1
0	0	0
1	1	1

$F(A,B) = AB'+AB = A(B'+B) = A$

How to Do 2 Groups

- Now suppose F(A,B) = A’B+A’B’ +AB, the Karnaugh Map looks like



A \ B	0	1
0	1	1
1	0	1

Since A’ does not change in the RED terms but B does then A’ remains and B is eliminated.

Since B does not change in the BLUE terms but A does then B remains and A is eliminated.

F(A,B) = A’B’ + A’B + AB
= A’B’ + A’B + A’B + AB
= A’(B’ + B) + (A’ + A)B
= A’(1) + (1)B
= A’ + B

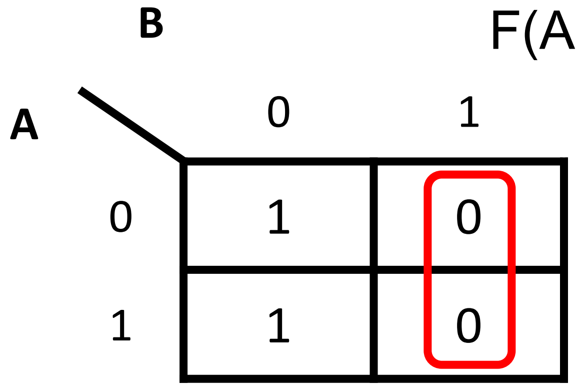
So,

F(A,B) = A’B+A’B’ +AB = A’ + B

Rules for Combining Squares (Simplification, Elimination)

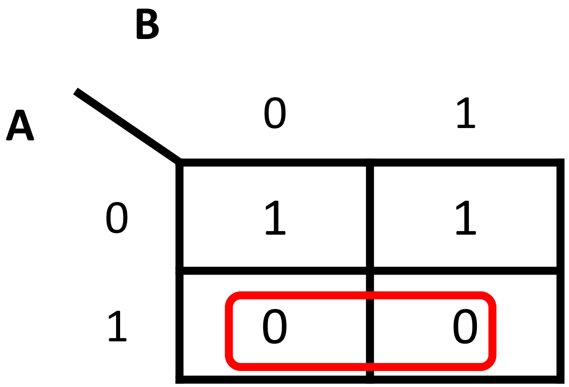
1. Can combine 2^k adjacent cells; where 0≤k≤n and n is the number of variables; implies that
   - Can combine 1, 2, 4, 8, 16, etc, cells; a power of 2
   - Cannot combine 3, 5, 6, 7, 9, etc cells
2. Want to combine the largest number of cells possible to eliminate as many variables as possible, implies that:
   - Combine 2 cells, eliminate 1 variable
   - Combine 4 cells, eliminate 2 variables
   - Combine 8 cells, eliminate 3 variables
   - Combine 2^n, eliminate n variables
3. Once a minterm (maxterm) is used we can use it again
4. Avoid redundancy!!!

Karnaugh Maps
(2 – Variable MAXTERM Rules)



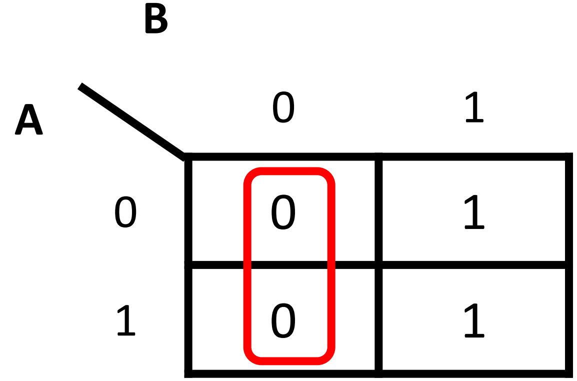
	0	1
0	1	0
1	1	0

F(A,B) = (A+B')(A'+B')
= AA'+AB'+B'A'+B'B'
= 0 + AB' +B'A' + 0
= AB' +B'A'
= (A + A')B'
= (1)B'
= B'



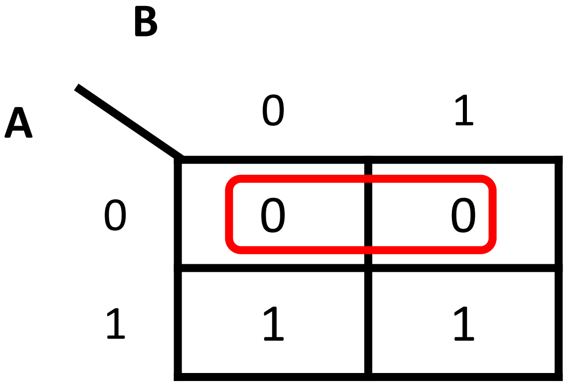
	0	1
0	1	1
1	0	0

F(A,B) = A'



	0	1
0	0	1
1	0	1

F(A,B) = B



	0	1
0	0	0
1	1	1

F(A,B) = A

Any 2 maxterms in adjacent squares that are ANDed will cause removal of the different variable

Looking Back

- Now suppose F(A,B) = A’B+A’B’ +AB, the Karnaugh Map looks like

A \ B	0	1
0	1	1
1	$\boxed{0}$	1

So,

$$F(A,B) = M_2 = A' + B$$

$$\begin{aligned} F(A,B) &= A'B' + A'B + AB \\ &= A'B' + A'B + A'B + AB \\ &= A'(B' + B) + (A' + A)B \\ &= A'(1) + (1)B \\ &= A' + B \end{aligned}$$

Some times it easier to work the maxterms than the minterms and vice versa

Examples: 2 – Variable K-Maps

EX: simplify the function

X \ Y	0	1
0	0	1
1	1	0

F(X,Y) = XY' + X' Y ⇒ **XOR gate**

These 2 configurations that result in selecting the 2 minterms by themselves.

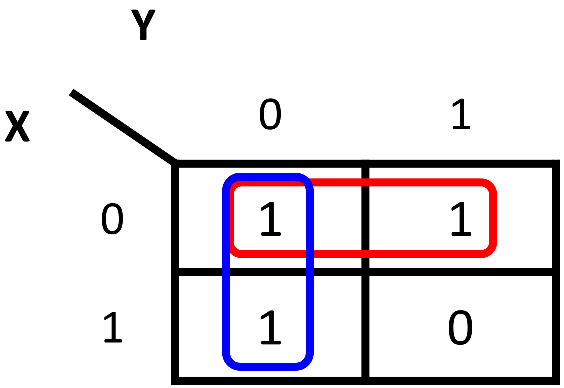
These Represent an

X \ Y	0	1
0	1	0
1	0	1

F(X,Y) = X'Y + XY' ⇒ **XNOR gate**

Practice Problem

- Simplify the function $F(X,Y) = X'Y' + XY' + X'Y$



Y

X	0	1
0	1	1
1	1	0

Since X' does not change in the RED terms but Y does then X' remains and Y is eliminated.

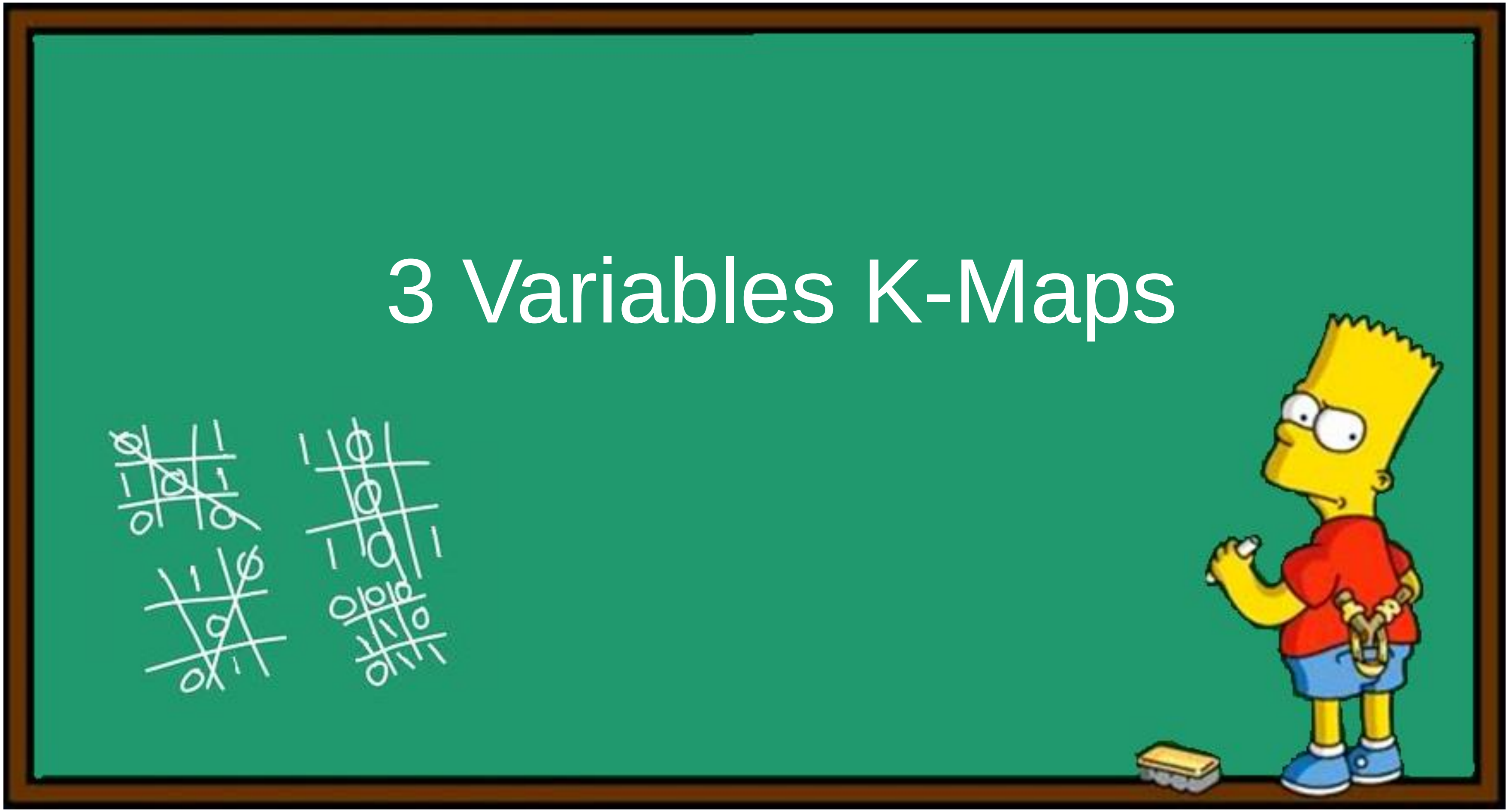
Since Y' does not change in the BLUE terms but X does then Y' remains and X is eliminated.

$$
\begin{aligned}
F(X,Y) &= X'Y' + X'Y + XY' \\
&= X'Y' + X'Y + XY' + X'Y' \\
&= X'(Y' + Y) + (X + X')Y' \\
&= X'(1) + (1)Y' \\
&= X' + Y'
\end{aligned}
$$

So,

$$
\begin{aligned}
F(X, Y) &= X'Y' + XY' + X'Y \\
&= X'+Y'
\end{aligned}
$$

3 Variables K-Maps



Three- Variable Karnaugh Maps

BC A	00	01	11	10
0	a_0	a_1	a_3	a_2
1	a_4	a_5	a_7	a_6

As with the 2 variable K-MAP the truth table is moved mapped into the table as shown

A	B	C	F	Term	Coeff
0	0	0	0	m_0	a_0
0	0	1	0	m_1	a_1
0	1	0	0	m_2	a_2
0	1	1	1	m_3	a_3
1	0	0	1	m_4	a_4
1	0	1	1	m_5	a_5
1	1	0	1	m_6	a_6
1	1	1	1	m_7	a_7

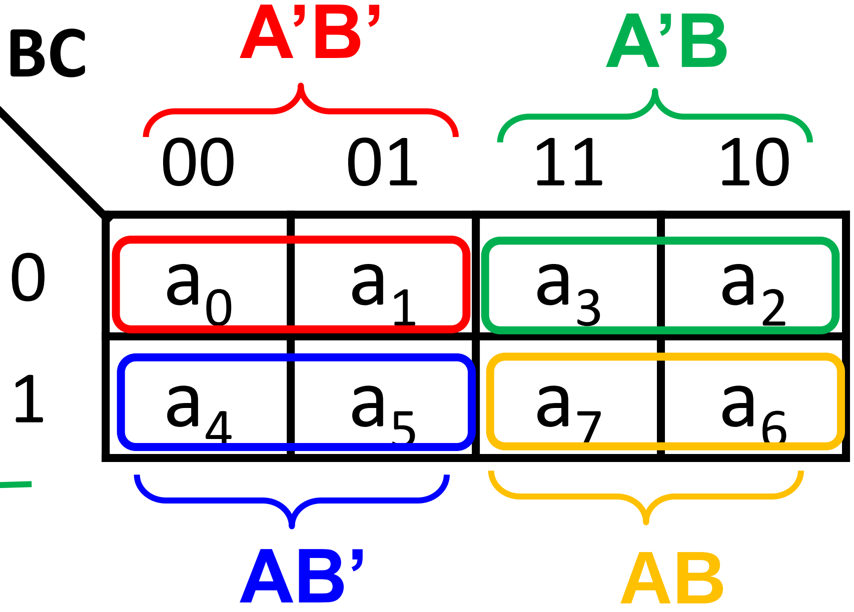
Gray Code

A	B	C	Terms
0	0	0	m_0
0	0	1	m_1
0	1	1	m_3
0	1	0	m_2
1	1	0	m_6
1	1	1	m_7
1	0	1	m_5
1	0	0	m_4

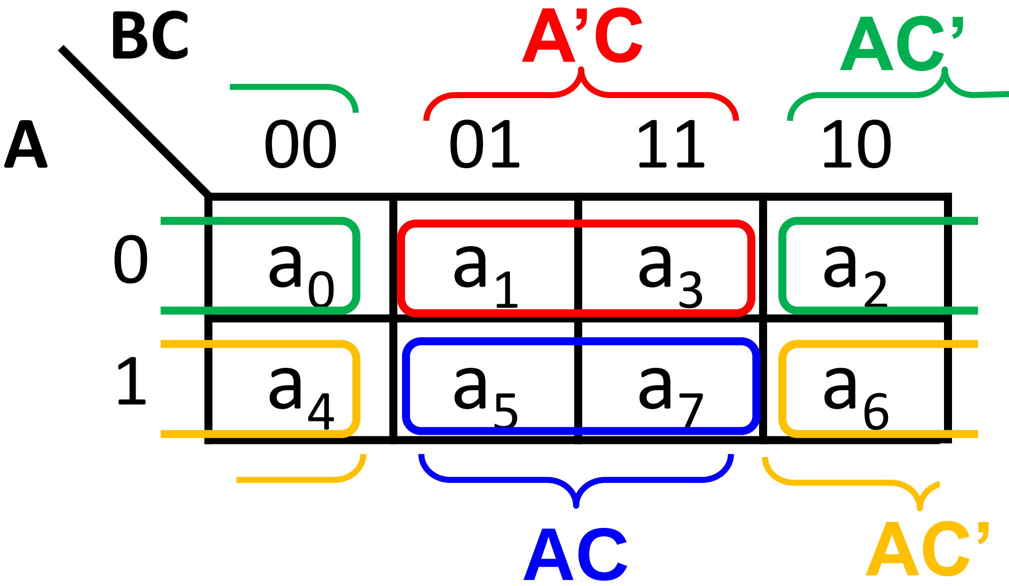
The table uses a gray code for the columns so that the Absorbtion property holds

Groupings of 2

So a group of 2 variables across to adjacent columns eliminates the variable that changes between columns.



A \ BC	00	01	11	10
0	a_0	a_1	a_3	a_2
1	a_4	a_5	a_7	a_6



A \ BC	00	01	11	10
0	a_0	a_1	a_3	a_2
1	a_4	a_5	a_7	a_6

Q&A