

# Gray Codes & Karnaugh Maps

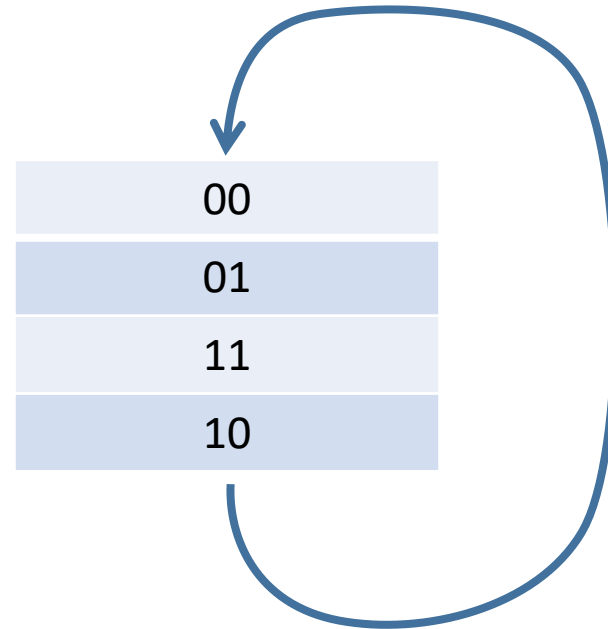
Prof. James L. Frankel  
Harvard University

Version of 10:16 AM 2-Dec-2021  
Copyright © 2021, 2017, 2016 James L. Frankel. All rights reserved.

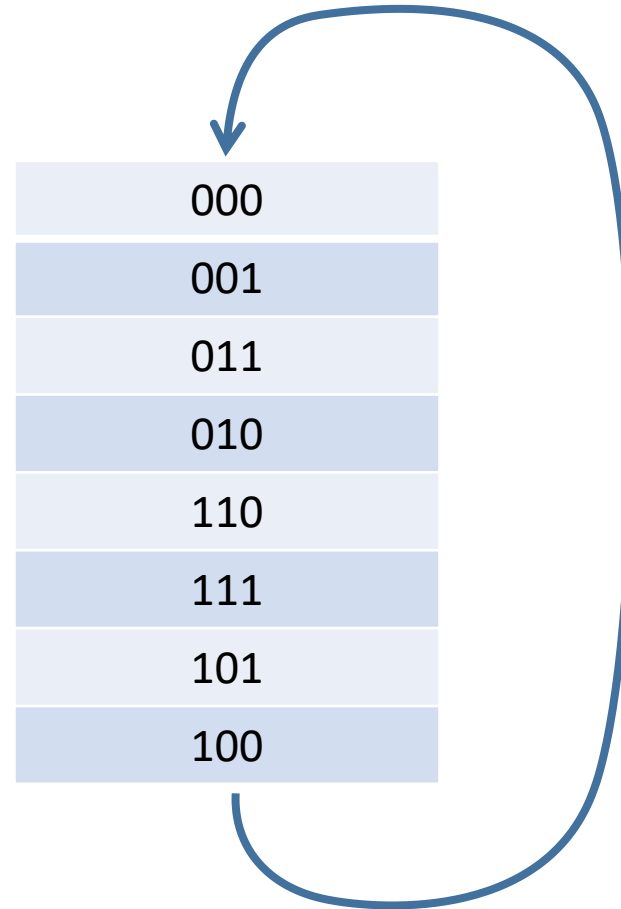
# Gray Code Ordering

- A sequence of  $n$ -bit codes in which only one bit changes at each transition
  - Must include the transition from the last to the first in the sequence as well
  - May be used to ensure that at each transition, there can never be an issue of multiple bits changing at slightly different times
  - In a clocked synchronous system, this is not an issue because we do not look at the signals while they are in transition

# Two-Bit Gray Code Sequence



# Three-Bit Gray Code Sequence



# Gray Code Properties

- Possible to select a power-of-two subsequence of values in a gray code ordering by looking at some of the bits
- This can be used to *cover* some of the values as we'll see

# Three-Bit Gray Code Subsequences of Length Four

- Reference the three bits as ABC, for example

- Select A = 0 

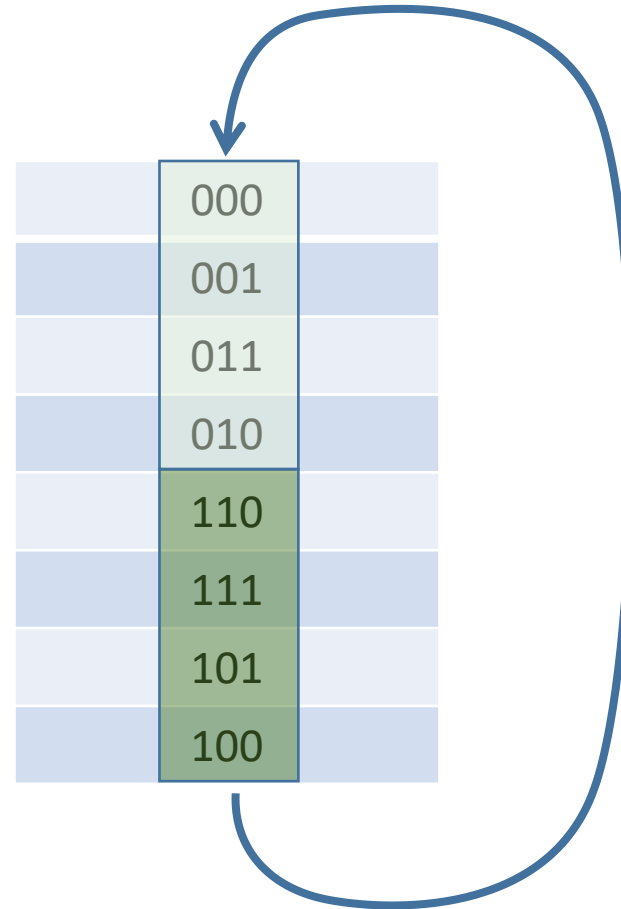
- Select A = 1 

- Select B = 0 

- Select B = 1 

- Select C = 0 

- Select C = 1 



# Three-Bit Gray Code Subsequences of Length Four

- Reference the three bits as ABC, for example

- Select A = 0 

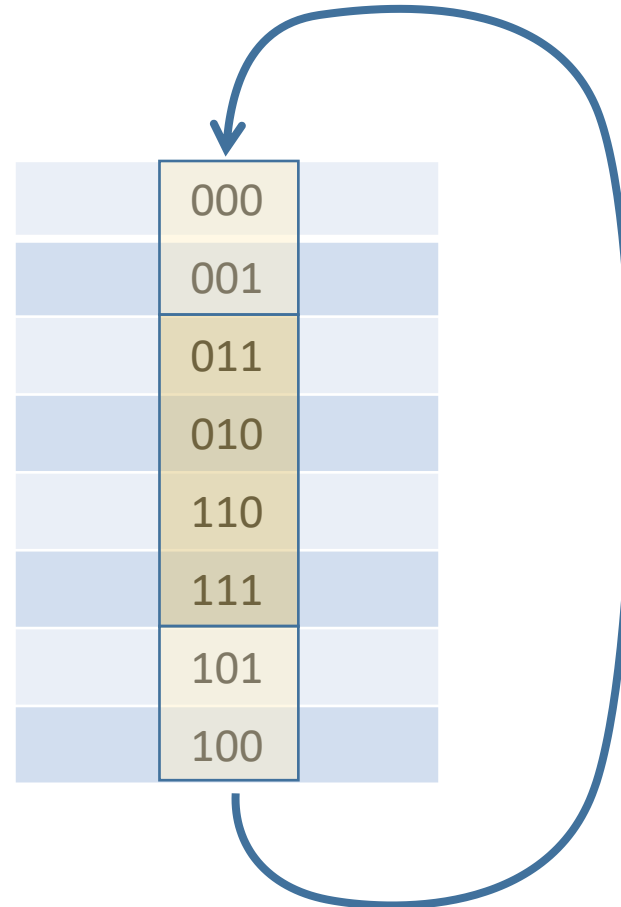
- Select A = 1 

- Select B = 0 

- Select B = 1 

- Select C = 0 

- Select C = 1 



# Three-Bit Gray Code Subsequences of Length Four

- Reference the three bits as ABC, for example

- Select A = 0 

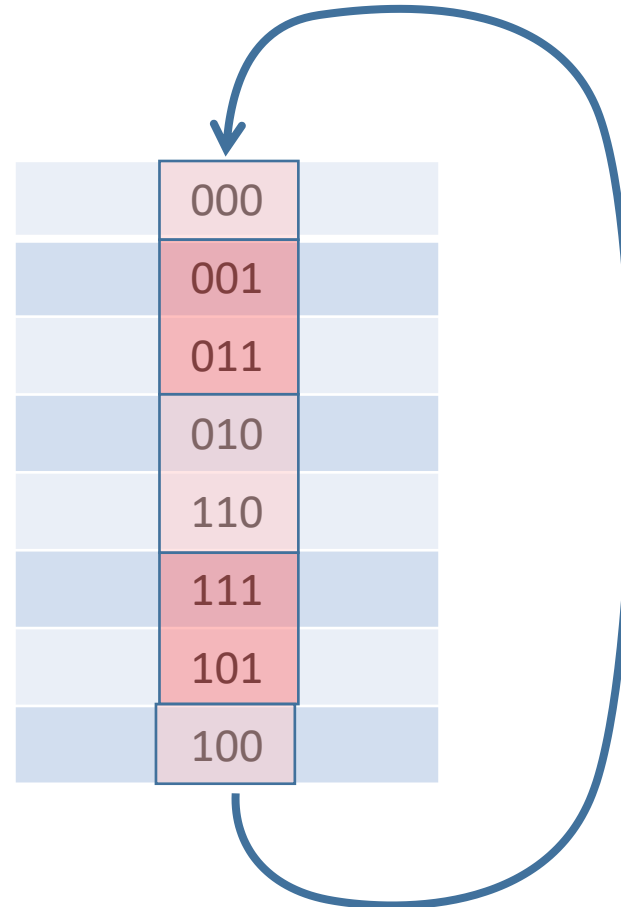
- Select A = 1 

- Select B = 0 

- Select B = 1 

- Select C = 0 





- Select C = 1 

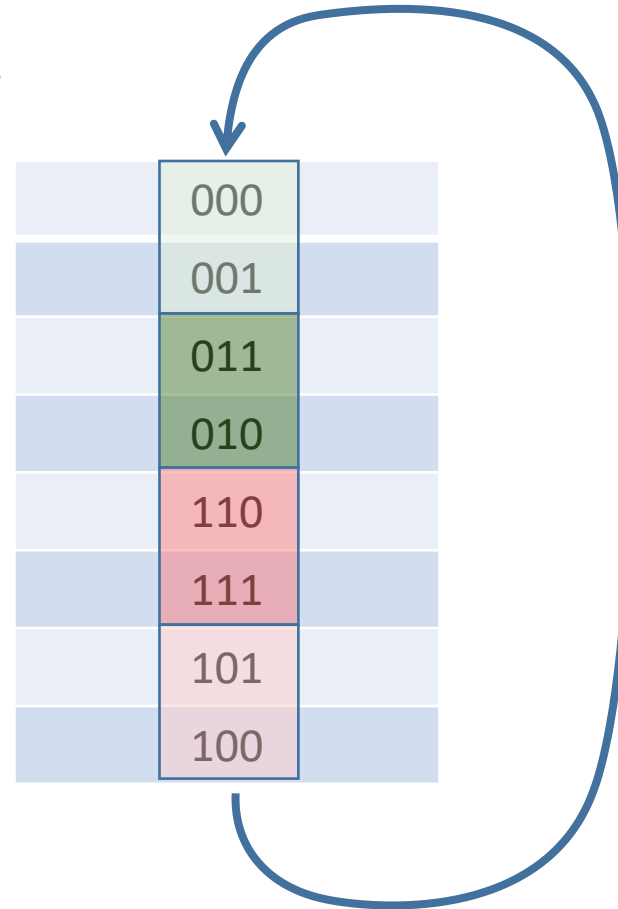




# Three-Bit Gray Code Subsequences of Length Two





- Reference the three bits as ABC, for example

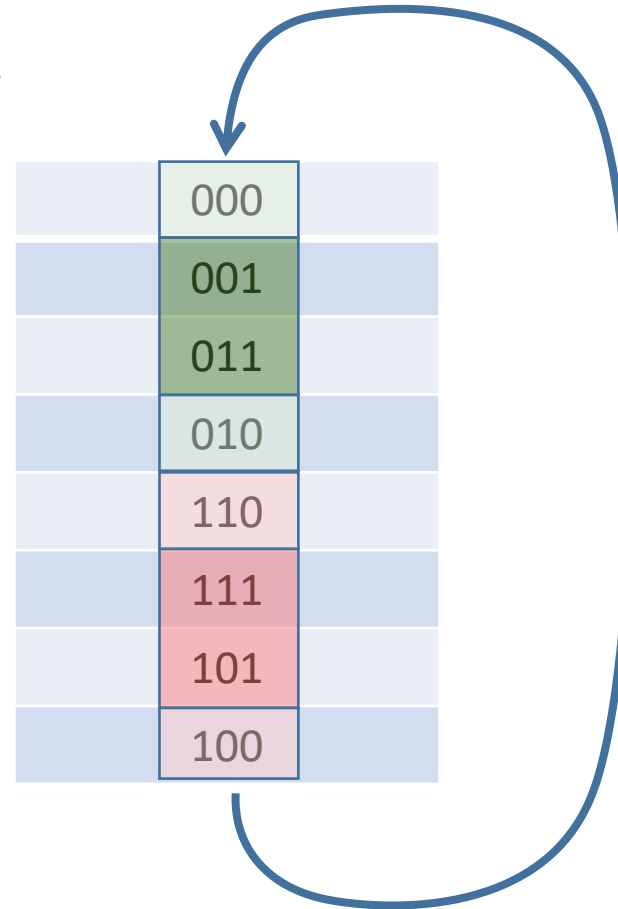
- Select AB = 00 
- Select AB = 01 
- Select AB = 11 
- Select AB = 10 



# Three-Bit Gray Code Subsequences of Length Two





- Reference the three bits as ABC, for example

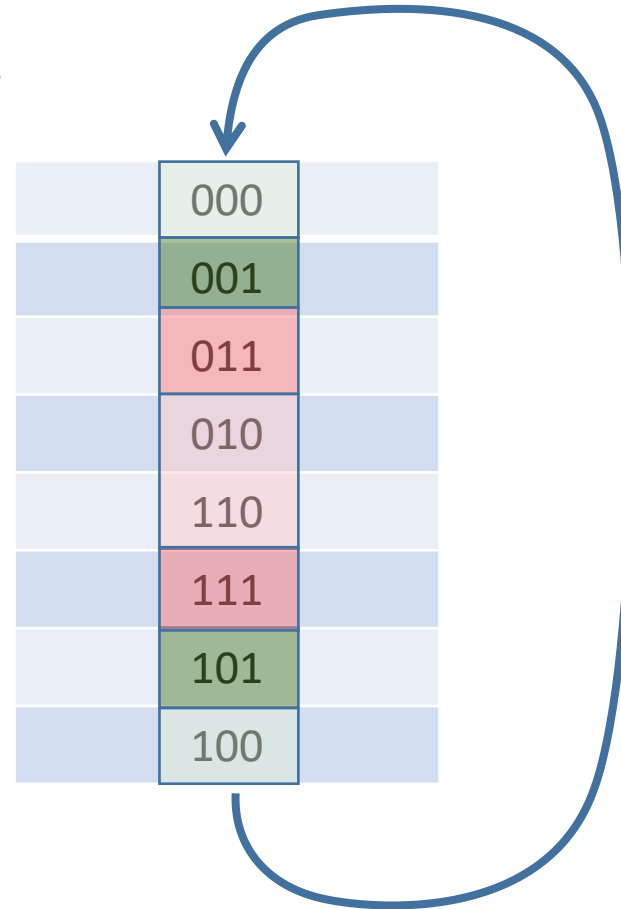
- Select AC = 00 
- Select AC = 01 
- Select AC = 11 
- Select AC = 10 



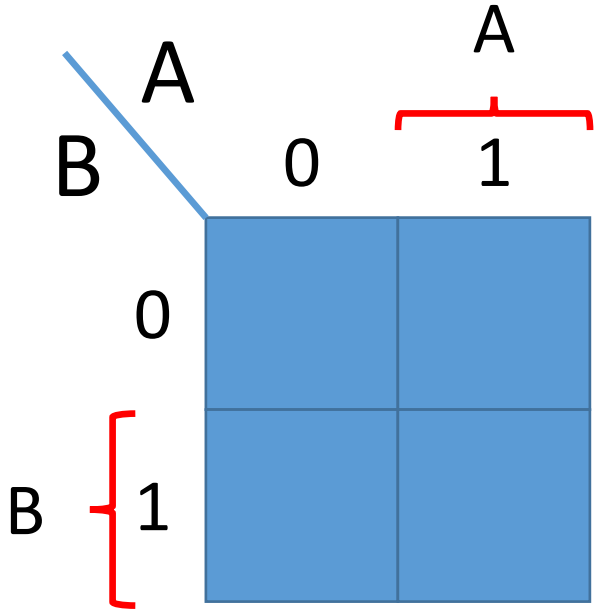
# Three-Bit Gray Code Subsequences of Length Two

- Reference the three bits as ABC, for example

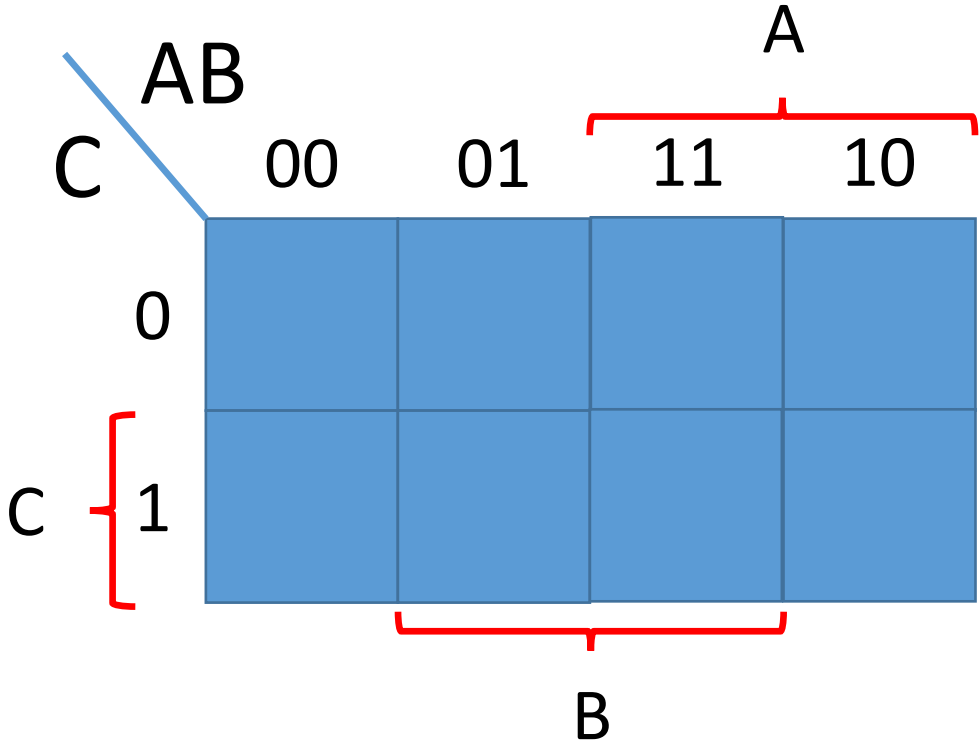
- Select BC = 00 
- Select BC = 01 
- Select BC = 11 
- Select BC = 10 



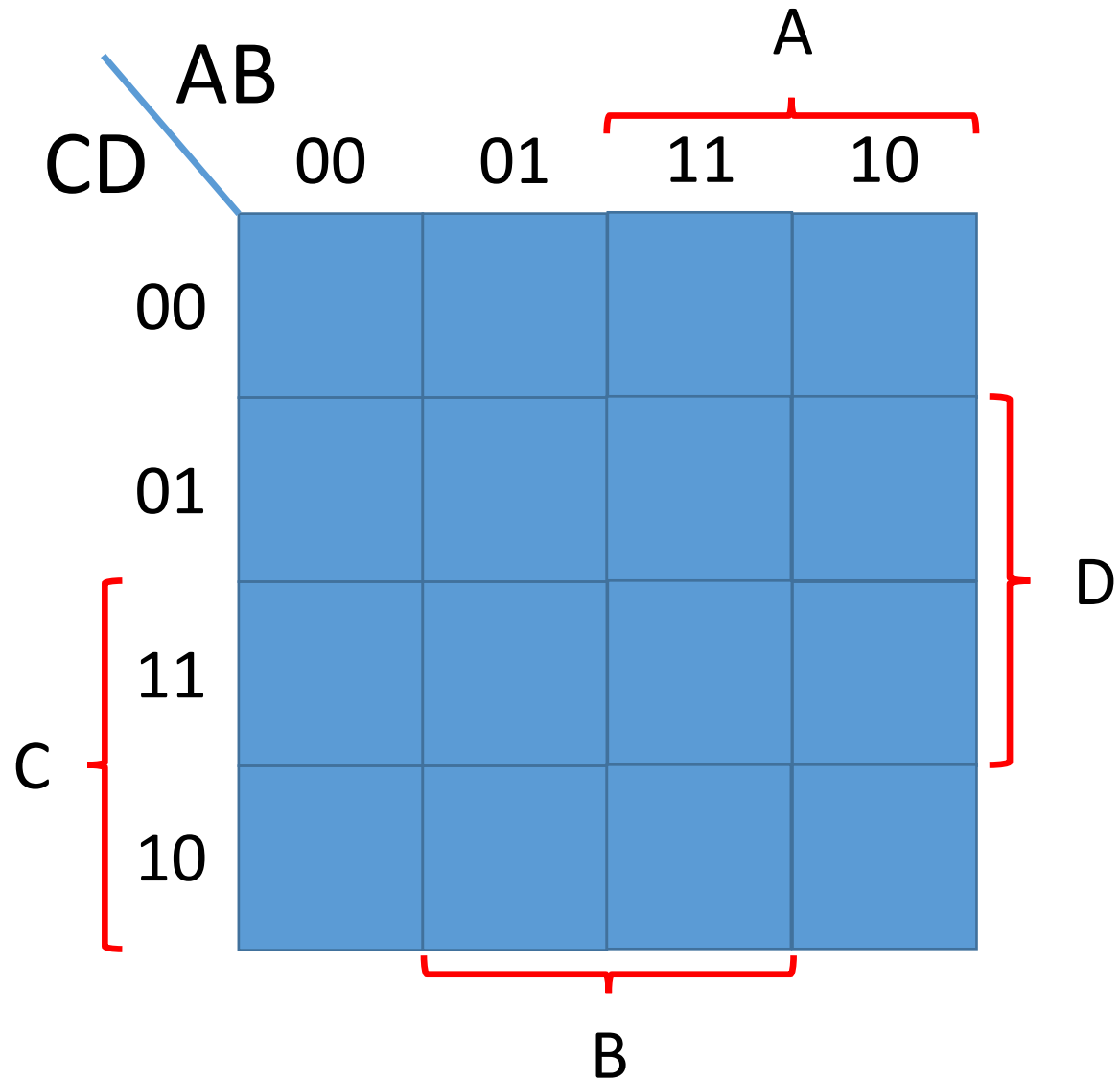
# Two Literal Karnaugh (or K-) Map



# Three Literal Karnaugh (or K-) Map



# Four Literal Karnaugh (or K-) Map



# Observations about Karnaugh Maps

- The K-Map graphically exhibits the adjacencies that are present in the gray code ordering

# Full Adder K-Maps

Sum

Carry <sub>in</sub>	AB		A	
	00	01	11	10
0	0	1	0	1
1	1	0	1	0

B

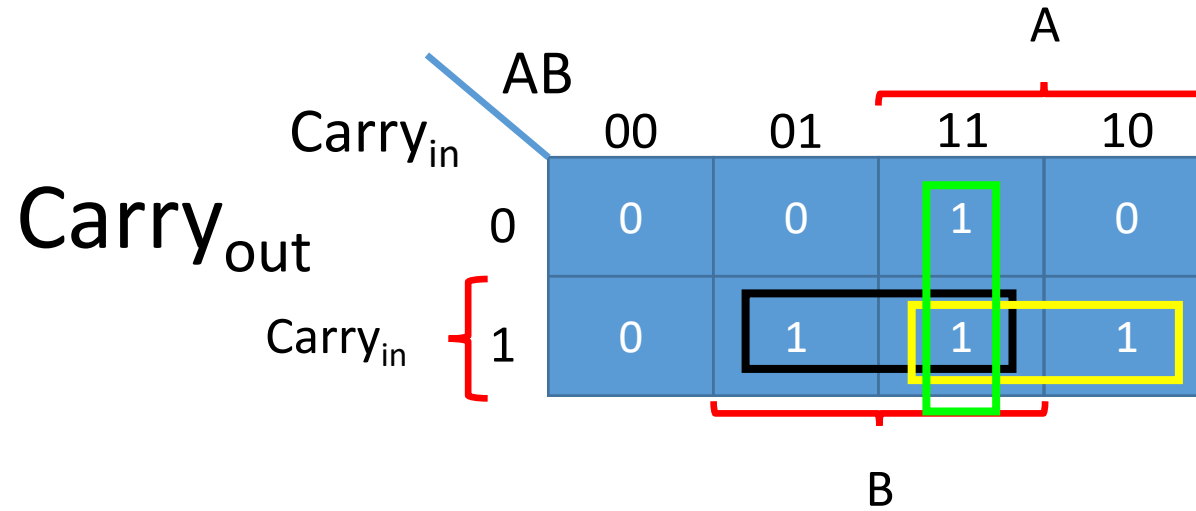
Carry<sub>out</sub>

Carry <sub>in</sub>	AB		A	
	00	01	11	10
0	0	0	1	0
1	0	1	1	1

B



# Find Covering Rectangles



# Create Simplified Expression

		AB		A	
		00	01	11	10
Carry <sub>out</sub>	0	0	0	1	0
	1	0	1	1	1

Carry<sub>in</sub> is indicated by a blue line pointing to the top-left cell (0,0) and a red bracket on the left side of the bottom row.

Red brackets indicate groupings: one above the top row (columns 11 and 10), and one below the bottom row (columns 01, 11, and 10).

Colored boxes highlight specific cells: a green box around the cell (0,11), a black box around the cell (1,01), and a yellow box around the cells (1,01), (1,11), and (1,10).

 BCarry<sub>in</sub>

 ACarry<sub>in</sub>

 AB

$$\text{Carry}_{\text{out}} = AB + \text{ACarry}_{\text{in}} + \text{BCarry}_{\text{in}}$$

# Compare Sum-of-Products and Simplified Expressions

- Sum-of-Products

- $\text{Carry}_{\text{out}} = ABCarry_{\text{in}} + (\sim A)BCarry_{\text{in}} + A(\sim B)Carry_{\text{in}} + AB(\sim Carry_{\text{in}})$
- Four terms
- Each with three literals

- Simplified

- $\text{Carry}_{\text{out}} = AB + ACarry_{\text{in}} + BCarry_{\text{in}}$
- Three terms
- Each with two literals