Gray Codes & Karnaugh Maps

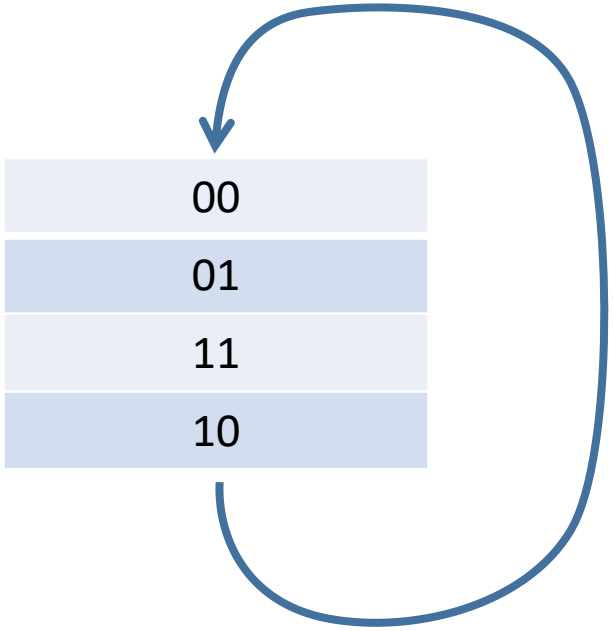
Prof. James L. Frankel
Harvard University

Version of 10:16 AM 2-Dec-2021

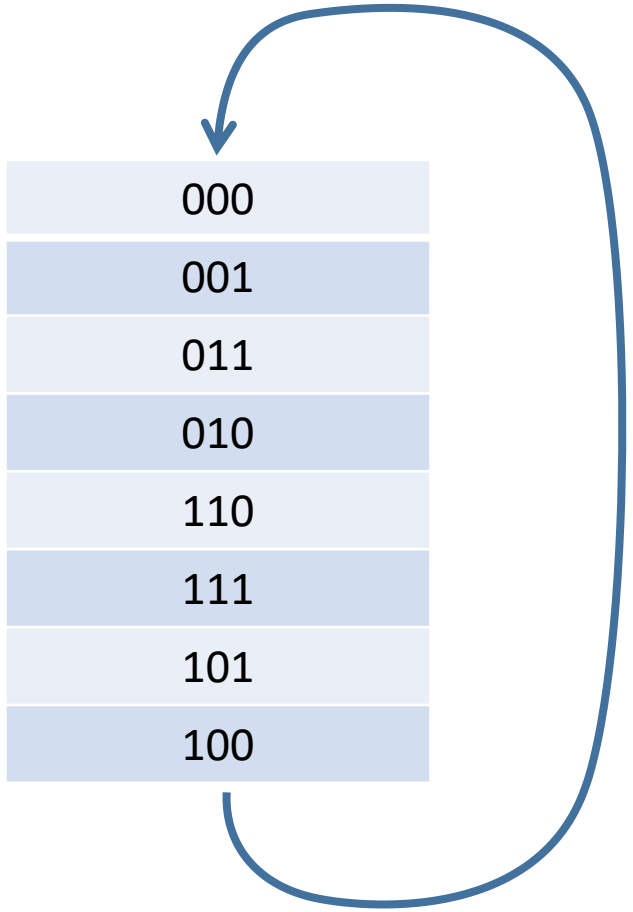
Gray Code Ordering

- 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

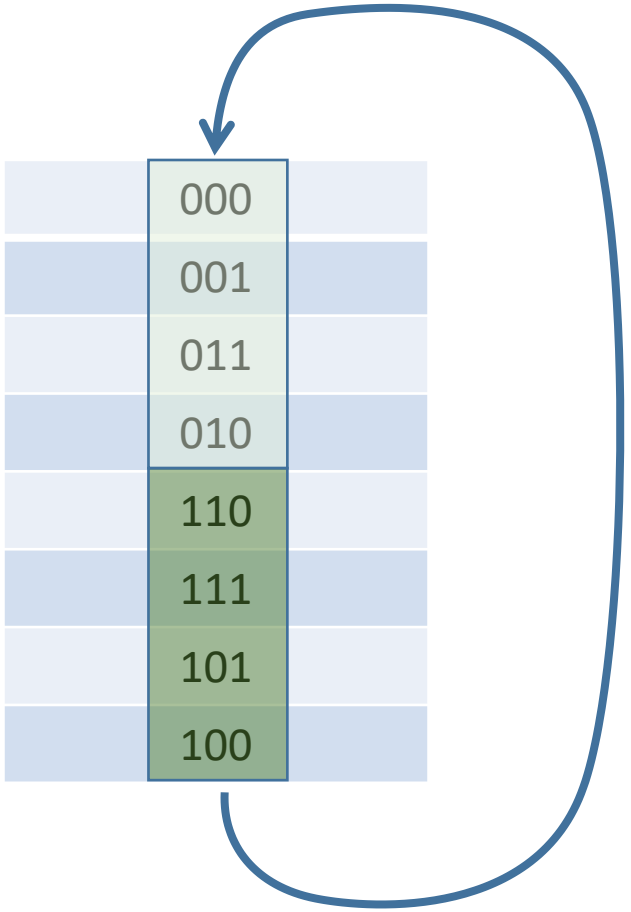
Three-Bit Gray Code Subsequences of Length Four

- 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



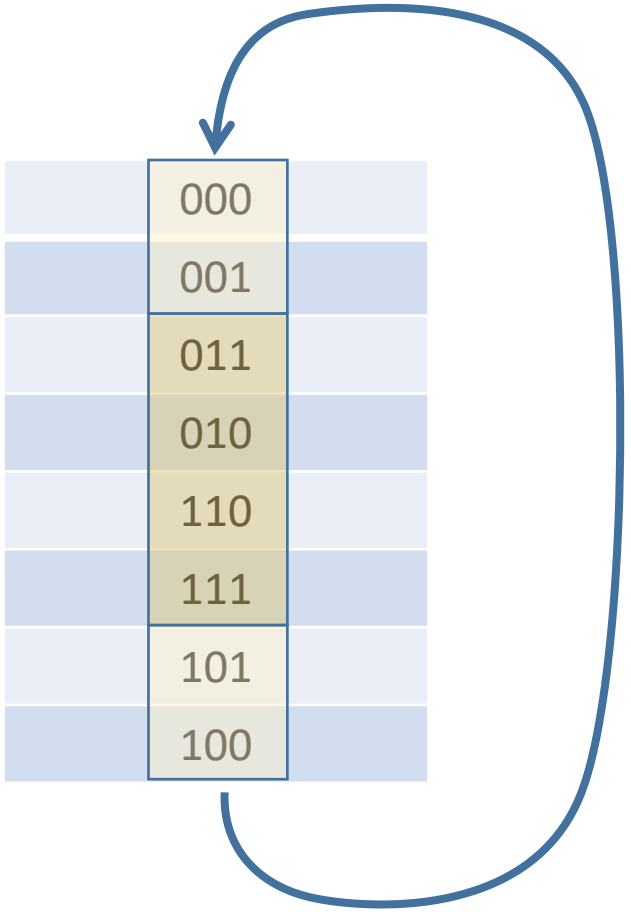
Three-Bit Gray Code Subsequences of Length Four

- 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



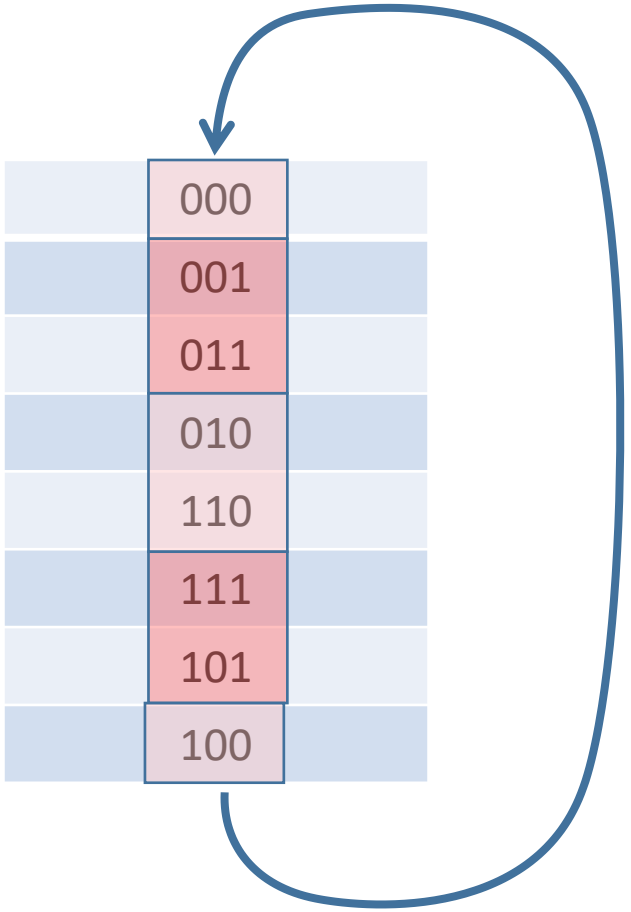
Three-Bit Gray Code Subsequences of Length Four

- Reference the three bits as ABC, for example

- Select A = 0
- Select A = 1

- Select B = 0
- Select B = 1

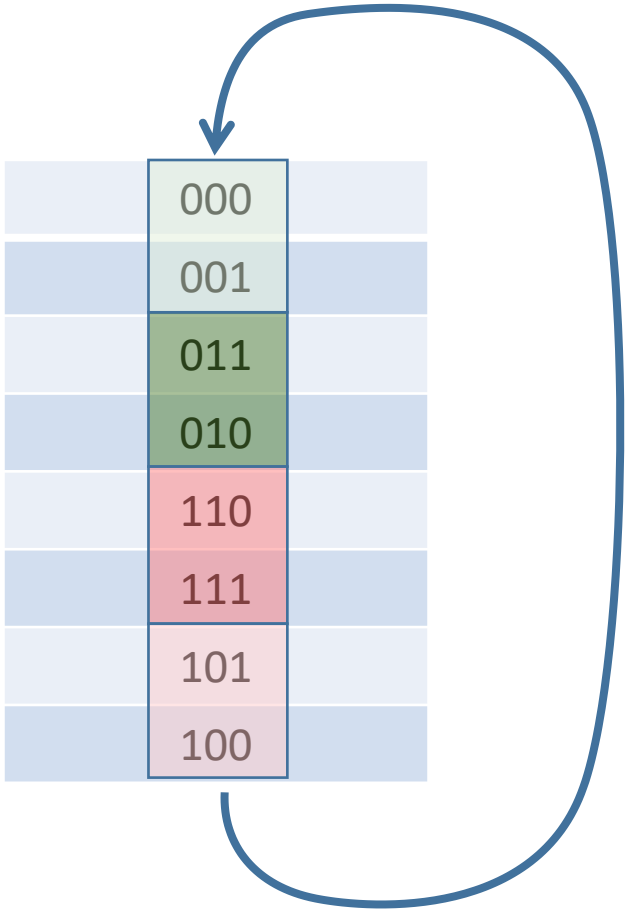
- Select C = 0
- Select C = 1



Three-Bit Gray Code Subsequences of Length Two

- Reference the three bits as ABC, for example

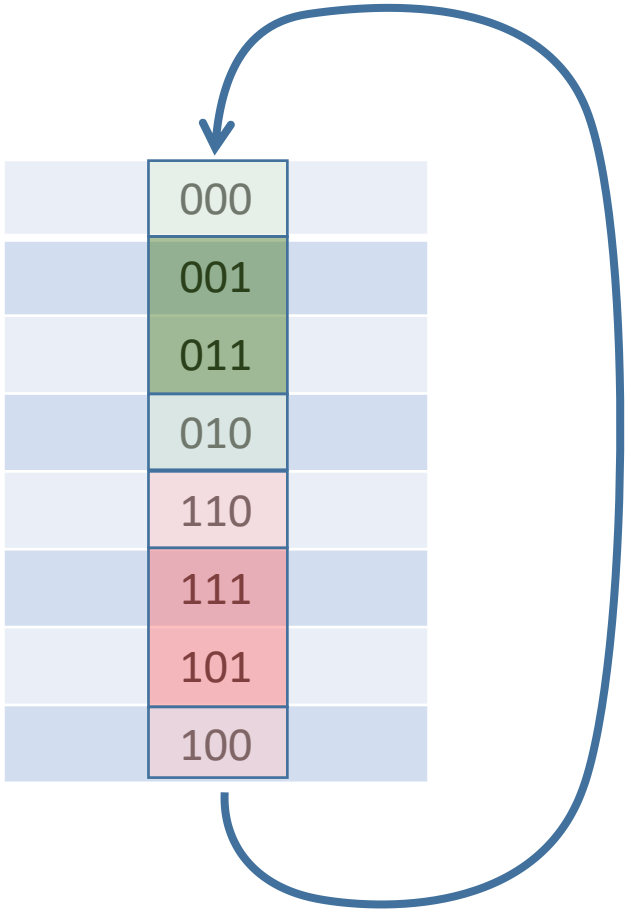
- Select AB = 00
- Select AB = 01
- Select AB = 11
- Select AB = 10



Three-Bit Gray Code Subsequences of Length Two

- Reference the three bits as ABC, for example

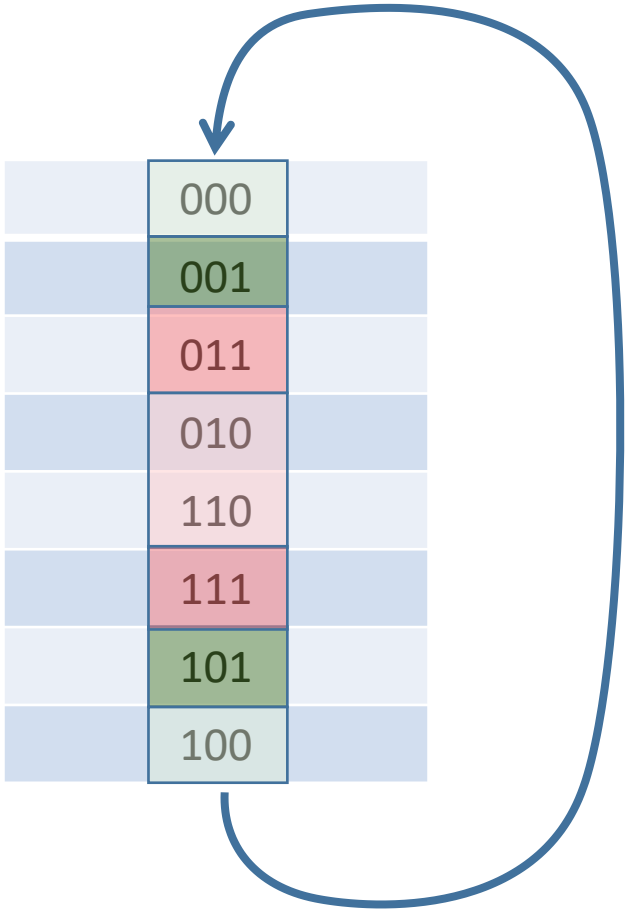
- Select AC = 00
- Select AC = 01
- Select AC = 11
- Select AC = 10



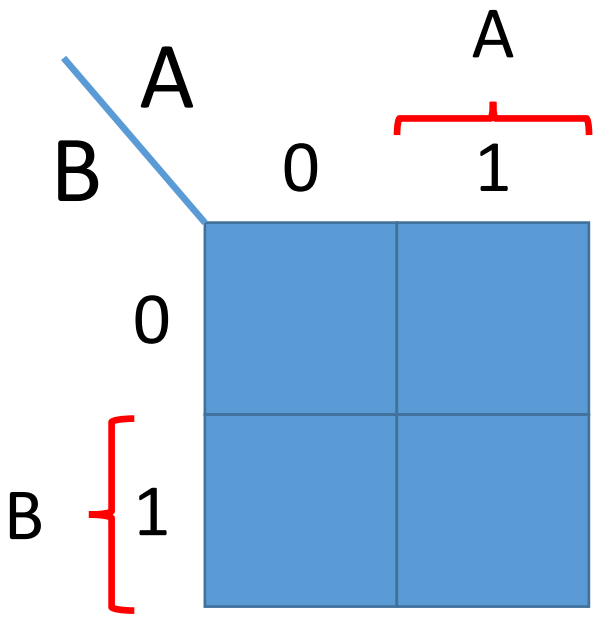
Three-Bit Gray Code Subsequences of Length Two

- Reference the three bits as ABC, for example

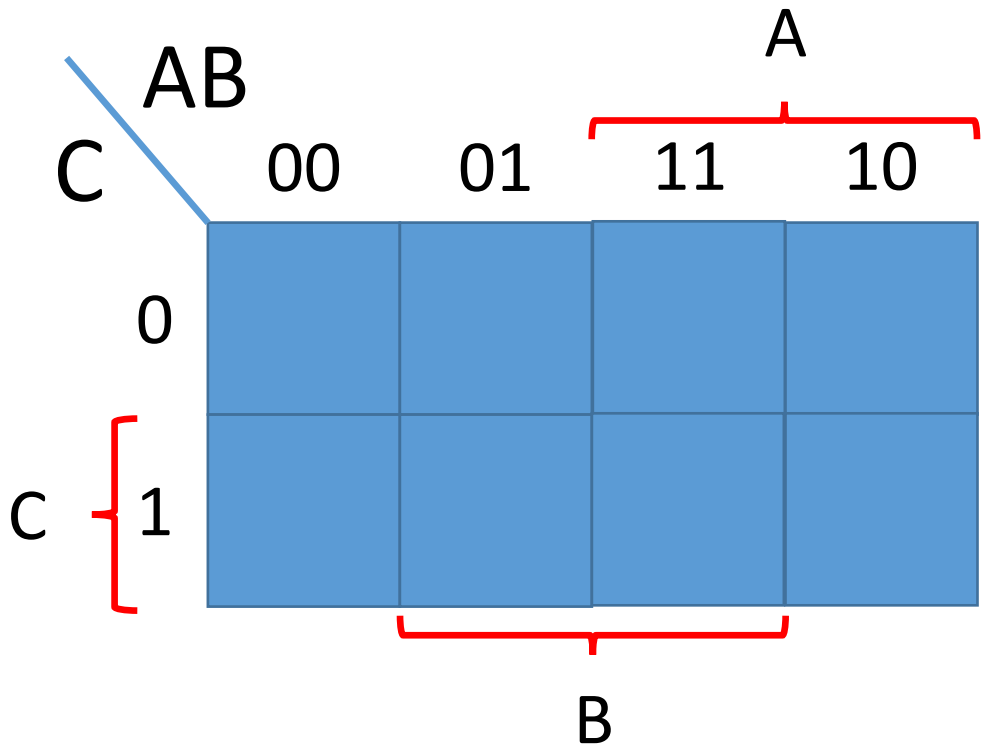
- Select BC = 00
- Select BC = 01
- Select BC = 11
- Select BC = 10



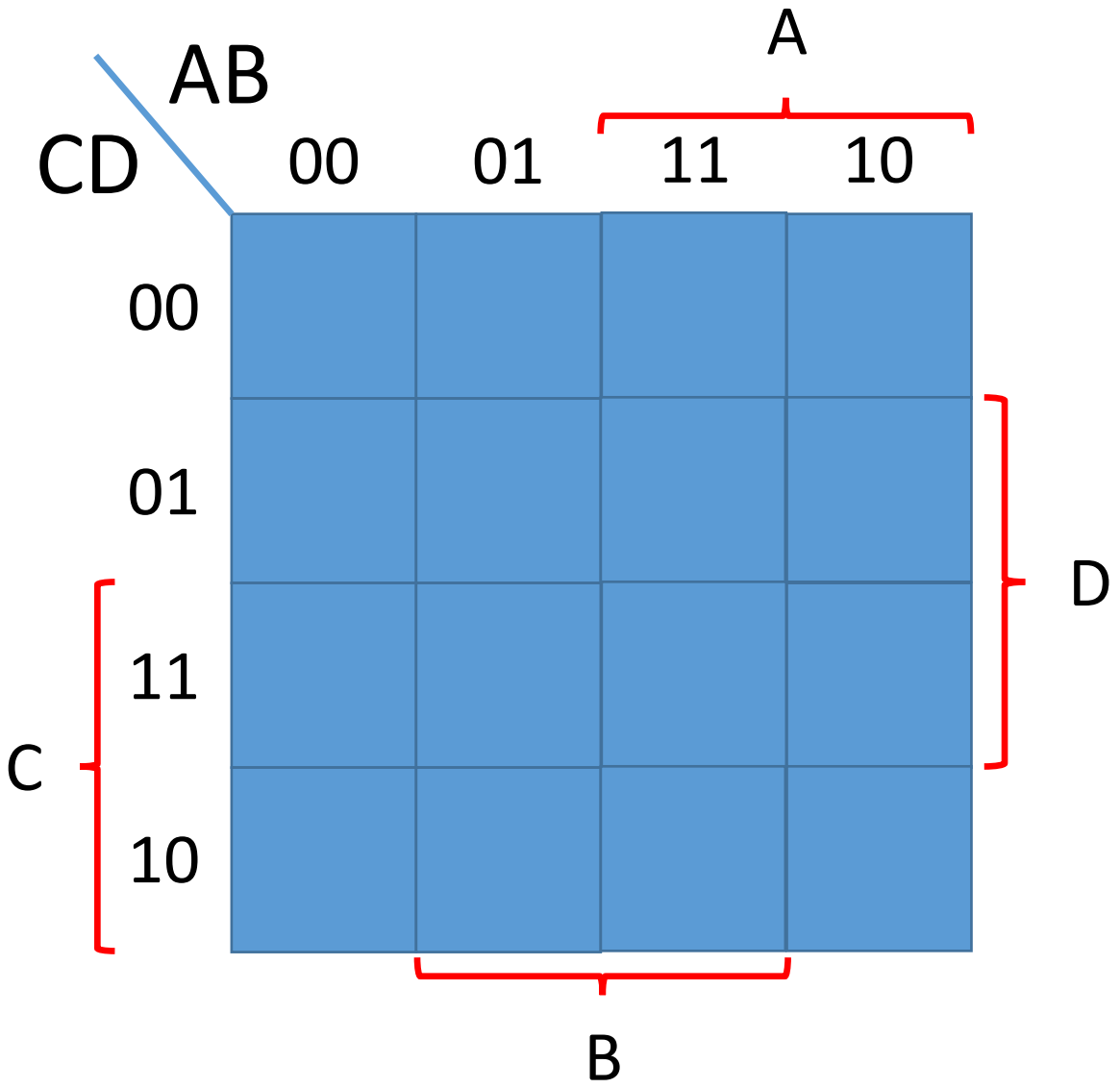
Two Literal Karnaugh (or K-) Map



Three Literal Karnaugh (or K-) Map



Four Literal Karnaugh (or K-) Map

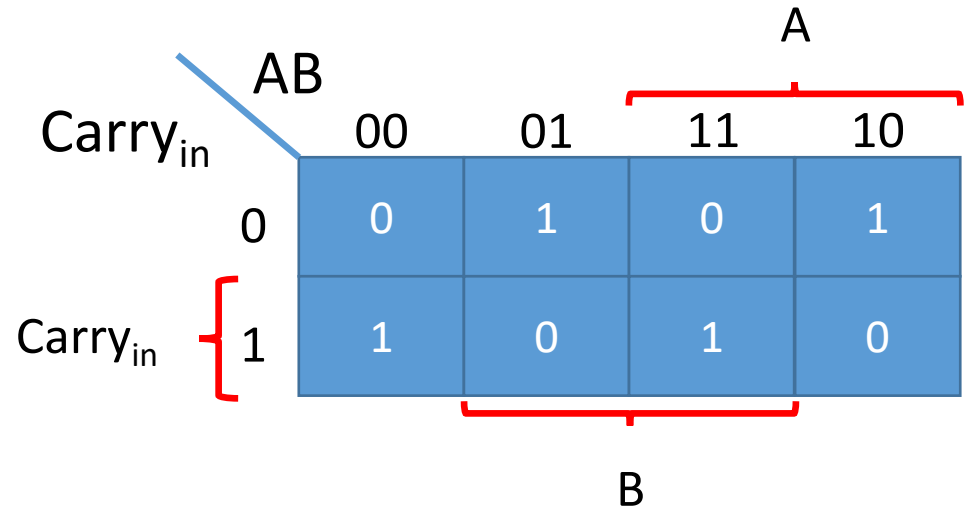


Observations about Karnaugh Maps

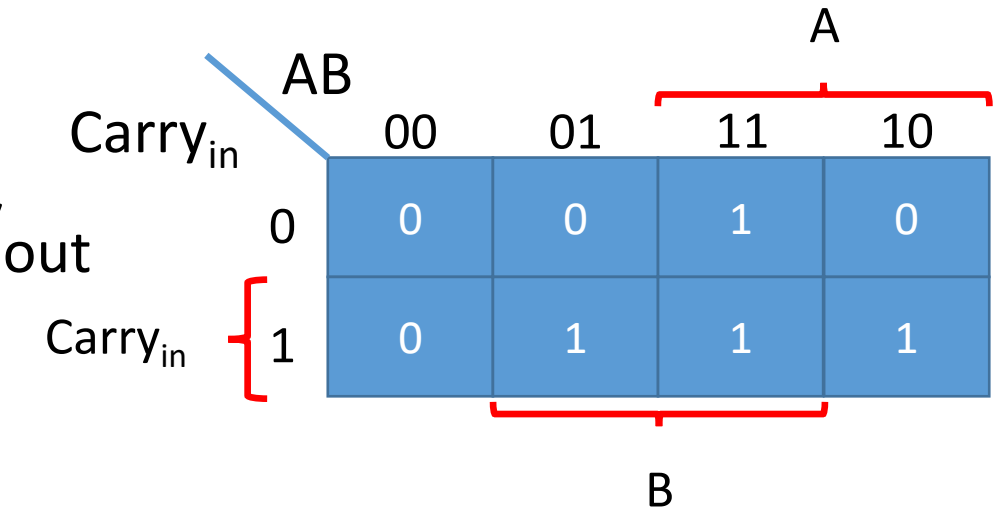
- The K-Map graphically exhibits the adjacencies that are present in the gray code ordering

Full Adder K-Maps

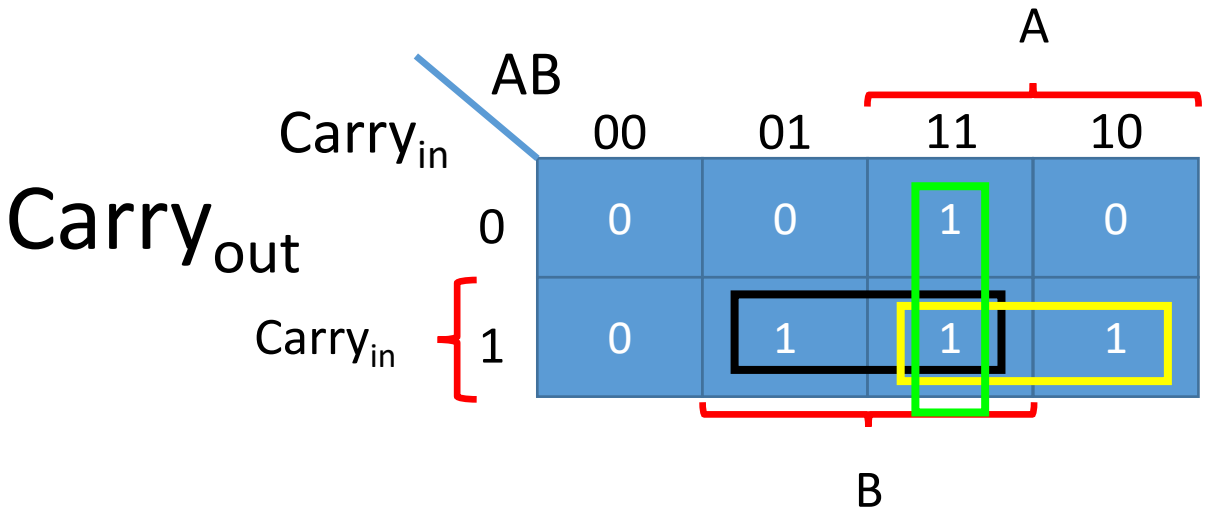
Sum



Carry$_{out}$



Find Covering Rectangles

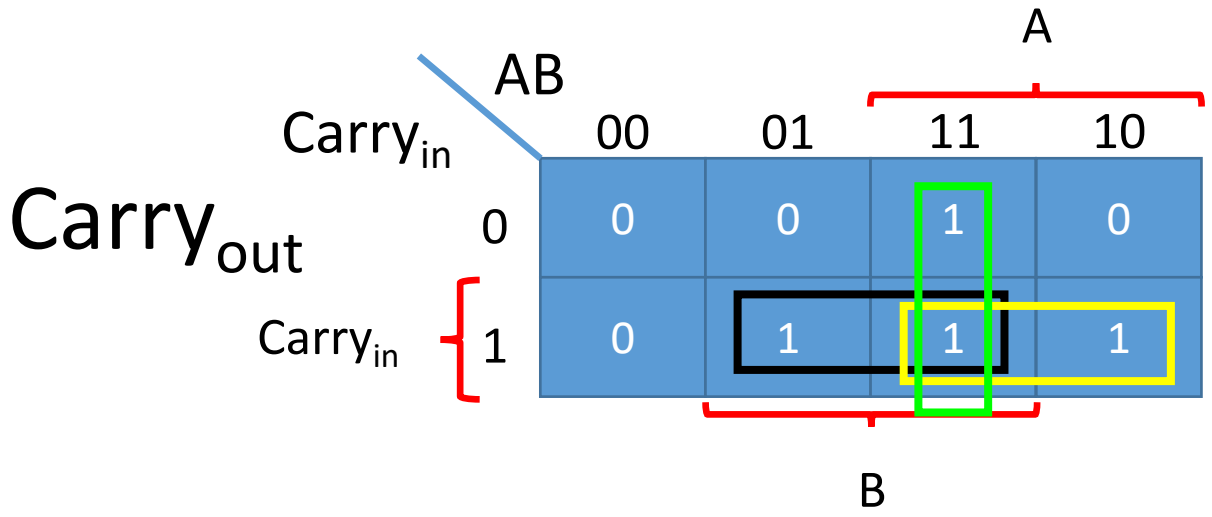


BCarry$_{in}$

ACarry$_{in}$

AB

Create Simplified Expression



BCarry$_{in}$

ACarry$_{in}$

AB

$$Carry_{out} = AB + ACarry_{in} + BCarry_{in}$$

Compare Sum-of-Products and Simplified Expressions

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

- Simplified
  - $Carry_{out} = AB + ACarry_{in} + BCarry_{in}$
  - Three terms
  - Each with two literals