Overview

- Last lecture
  - deMorgan's theorem
  - NAND and NOR
  - Canonical forms
    - Sum-of-products (minterms)
    - Product-of-sums (maxterms)
- Today's lecture
  - Logic simplification
    - Boolean cubes
    - Karnaugh maps

Logic-function simplification

- Key tool: The uniting theorem → A(B'+B) = A
- The approach:
  - Find subsets of the ON-set where some variables don't change (the A's above) and others do (the B's above)
  - Eliminate the changing variables (the B's)

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

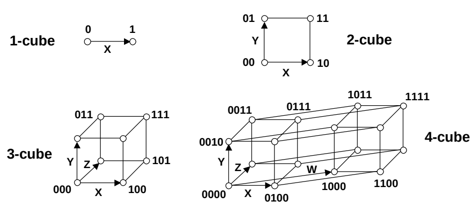
A has the same value in both on-set rows ⇒ keep A

B has a different value in the two rows ⇒ eliminate B

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

Boolean cubes

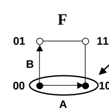
- *Visualize* when we can apply the uniting theorem
  - n input variables = n-dimensional "cube"



Mapping truth tables onto Boolean cubes

- ON set = solid nodes
- OFF set = empty nodes

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



Subcube (a line) comprises two nodes

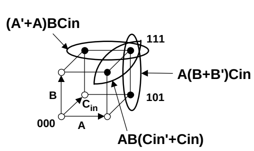
A varies within the subcube; B does not

This subcube represents the literal B'

Logic minimization using Boolean cubes

- Uniting theorem = find reduced-dimensionality subcubes
- Example: Binary full-adder carry-out logic
  - On-set is covered by the OR of three 2-D subcubes

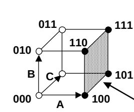
A	B	Cin	Cout
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1



Cout = BCin+AB+ACin

M-dimensional cubes in n-dimensional space

- In a 3-cube (three variables):
  - A 0-cube (a single node) yields a term in 3 literals
  - A 1-cube (a line of two nodes) yields a term in 2 literals
  - A 2-cube (a plane of four nodes) yields a term in 1 literal
  - A 3-cube (a cube of eight nodes) yields a constant term "1"



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

On-set forms a square (a 2-D cube)

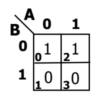
A is asserted (true) and unchanging
B and C vary

This subcube represents the literal A

Karnaugh maps

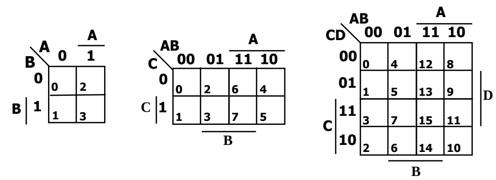
- Flat representation of Boolean cubes
  - Easy to use for 2– 4 dimensions
  - Hard for 4 – 6 dimensions
  - Virtually impossible for 6+ dimensions
    - Use CAD tools
- Help visualize adjacencies
  - On-set elements that have one variable changing are adjacent
    - Unlike a truth-table
  - Visual way to apply the uniting theorem

	A	B	F
0	0	0	1
1	0	1	0
2	1	0	1
3	1	1	0



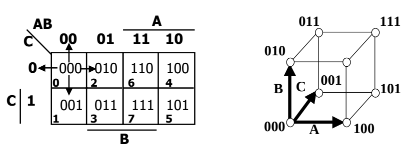
K-map cell numbering

- Gray–code: Only one bit changes between cells
  - Example: 00 → 01 → 11 → 10
- Layout for 2 – 4 dimension K-maps:

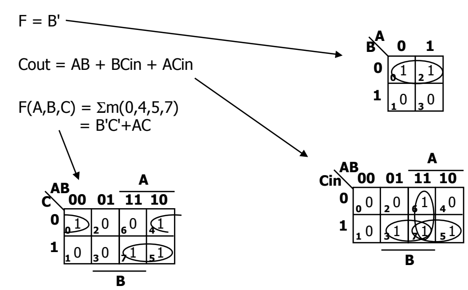


Adjacencies

- Wrap–around at edges
  - First column to last column
  - Top row to bottom row

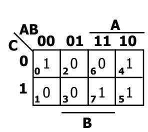


K-map minimization: 2 and 3 variables



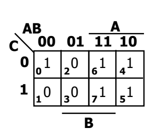
K-map minimization (con't)

- Obtain the complement by covering 0s with subcubes



F(A,B,C) = Σm(0,4,5,7)
= B'C'+AC

F'(A,B,C) = Σm(1,2,3,6)
= A'C + BC'

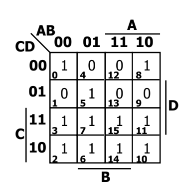


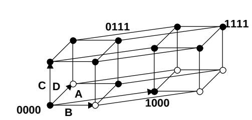
F(A,B,C) = ???

F'(A,B,C) = ???

K-map minimization: 4 variables

- Minimize F(A,B,C,D) = Σm(0,2,3,5,6,7,8,10,11,14,15)
  - Find the least number of subcubes, each as large as possible, that cover the ON-set

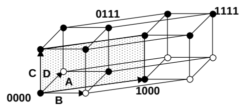




Karnaugh map: 4-variable example (con't)

- Minimize F(A,B,C,D) = Σm(0,2,3,5,6,7,8,10,11,14,15)
- Answer: F = C+A'BD+B'D'

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



K-map class examples

F(A,B,C,D) = Σm(0,3,7,8,11,15)

F(A,B,C,D) = ???
F'(A,B,C,D) = ???

CD \ AB	00	01	11	10
00				
01				
11				
10				

F(A,B,C) = Σm(0,3,6,7)

F(A,B,C) = ???
F'(A,B,C) = ???

C \ AB	00	01	11	10
0				
1				