ECE 2700
Digital Circuits

Logic Simplifications

UtahState University

Today’s Objectives

- To deepen your understanding on how to use Boolean Algebra to simplify/convert/manipulate digital circuits
- To understand the difference between optimization and tradeoff

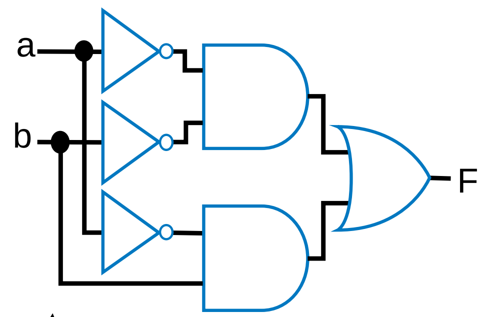
Representations of Boolean Functions

English 1: F outputs 1 when a is 0 and b is 0, or when a is 0 and b is 1.

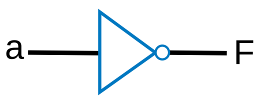
English 2: F outputs 1 when a is 0, regardless of b's value.

Equation 1: F(a,b) = a'b' + a'b

Equation 2: F(a,b) = a'



Circuit 1



Circuit 2

a	b	F
0	0	1
0	1	1
1	0	0
1	1	0

Truth table

The function F

- Only one truth table for any given Boolean function

Truth Table Representations

a	b	F
0	0	
0	1	
1	0	
1	1	

a	b	c	F
0	0	0	
0	0	1	
0	1	0	
0	1	1	
1	0	0	
1	0	1	
1	1	0	
1	1	1	

a	b	c	d	F
0	0	0	0	
0	0	0	1	
0	0	1	0	
0	0	1	1	
0	1	0	0	
0	1	0	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	

- Can be tedious as number of variables exceeds 5

Question (AP)

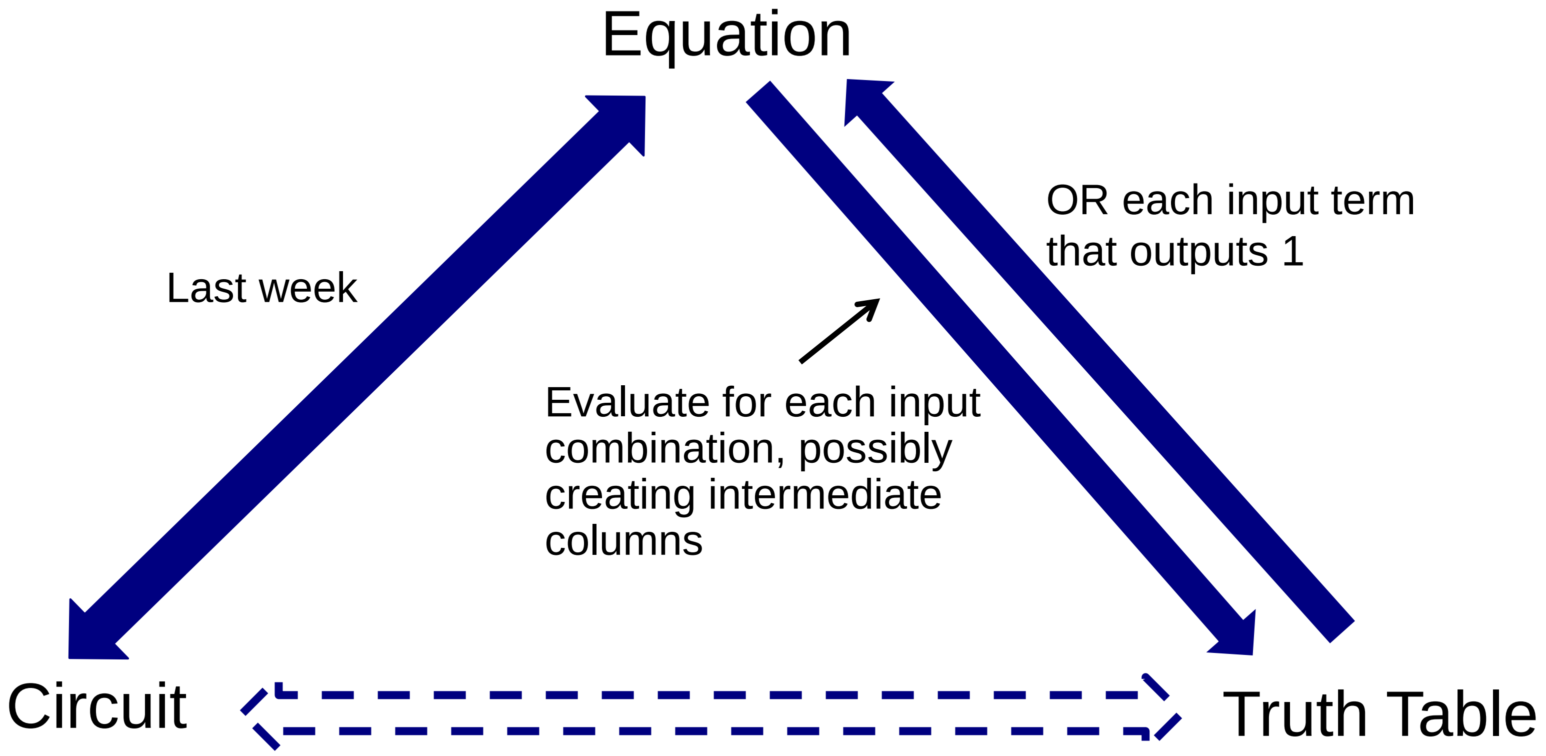
- Using truth tables, are F and G equivalent?
  - $F = xy + xy' + y'z$
  - $G = yz + x'z'$

A. Yes

B. No

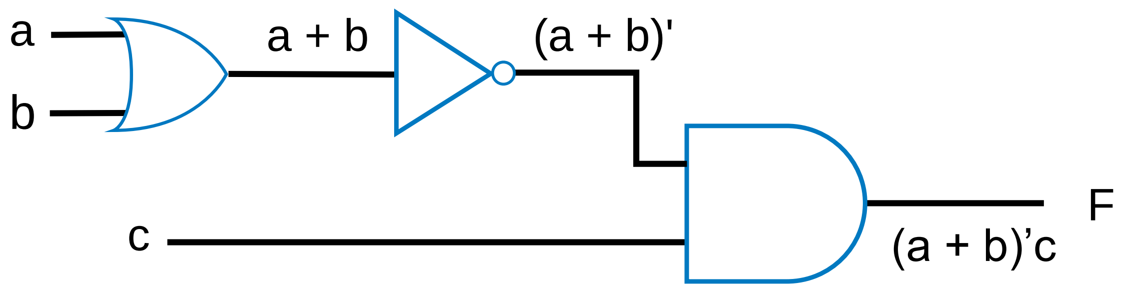
C. ☹

Converting among Representations



Circuit to Truth Table

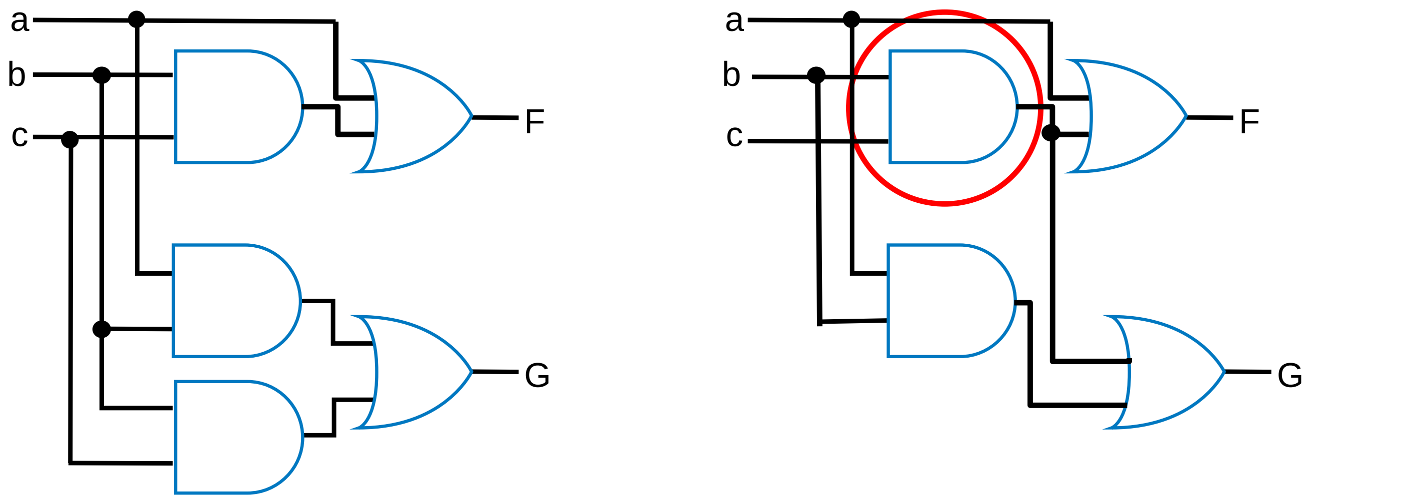
- First convert to circuit to equation, then equation to table



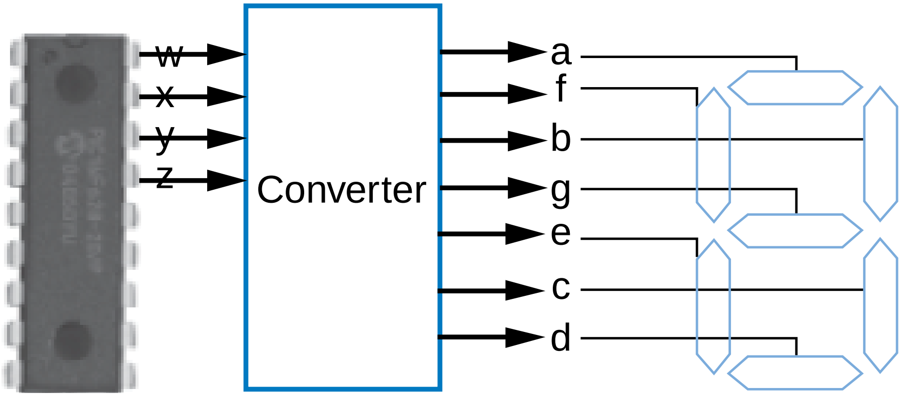
Inputs					Outputs
a	b	c	a + b	(a + b)'	F
0	0	0	0	1	0
0	0	1	0	1	1
0	1	0	1	0	0
0	1	1	1	0	0
1	0	0	1	0	0
1	0	1	1	0	0
1	1	0	1	0	0
1	1	1	1	0	0

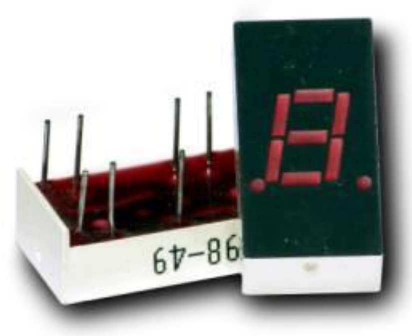
Multiple-Output Circuits

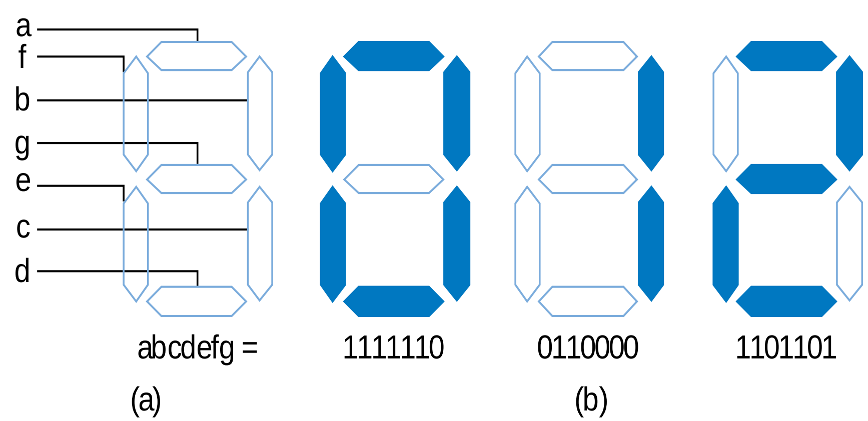
- Can give each a separate circuit, or can share gates
- Example: F = a + bc, G = ab + bc



Multiple-Output Example







Boolean Equations in Canonical Form

- (Standard form)
- Regular algebra: how do you know if the following equations are equivalent?
  - $F = 5x^2 + 4x + 4$
  - $G = 3x^2 + 4x + 2x^2 + 3 + 1$
- Boolean algebra: create sum of minterms
  - Minterm: product term with every literal appearing exactly once, in true or complemented form

Minterms for Three Variables

x	y	z	Mystery Function	Minterm
0	0	0		x'y'z'
0	0	1		x'y'z
0	1	0		x'yz'
0	1	1		x'yz
1	0	0		xy'z'
1	0	1		xy'z
1	1	0		xyz'
1	1	1		xyz

- F = x'y'z' + x'yz + xy'z' + xyz'
  - F is in canonical form (sum of minterms)

Writing a Boolean Eqn. in Canonical Form

- Steps
  - Obtain sum of product terms by multiplying-out equation
  - Expand each term until all terms are minterms

- Can be handy when you need to compare two equations

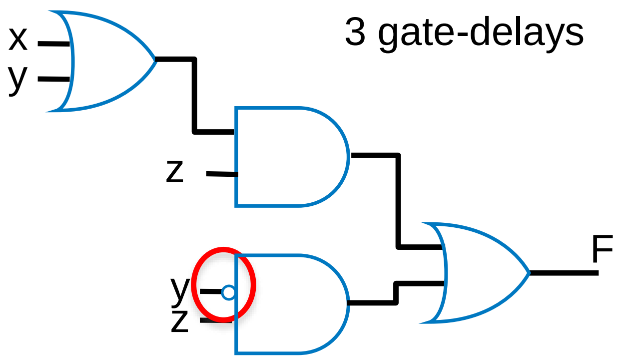
Building Better Circuits

- Two important design criteria
  - Delay – the time from inputs changing to new correct stable output
  - Size – the number of transistors
  - Ideally, we want a small circuit with a small delay
- For quick estimation, assume
  - Every gate has delay of “1 gate-delay”
  - Every gate input requires 2 transistors
    - Ignore inverters

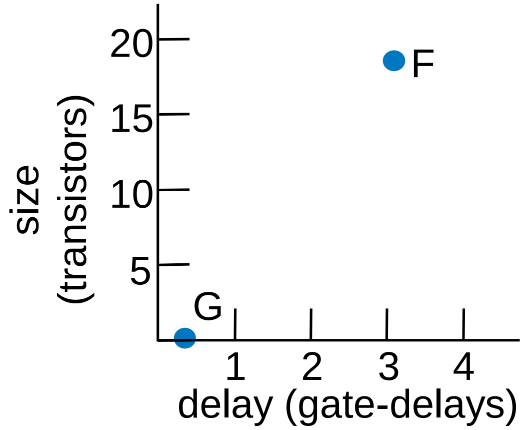
Optimization

- $F = (x + y) z + y'z$
- $G = z$

16 transistors
3 gate-delays



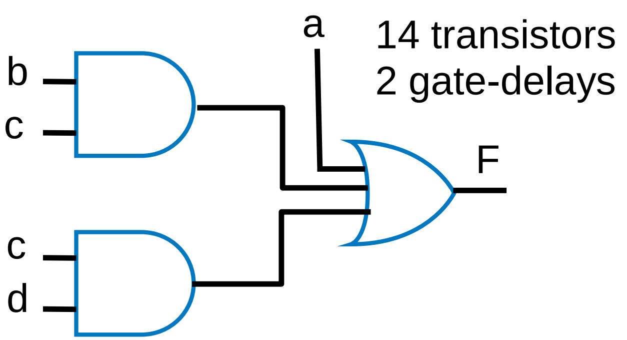
0 transistors
No gate-delay (only wire delay)



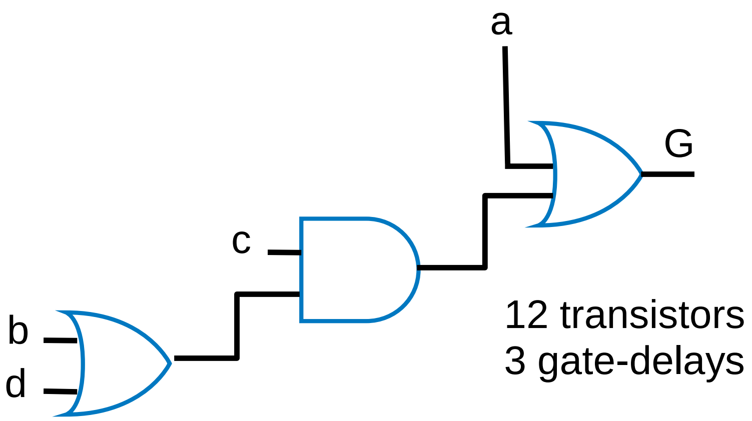
Transforming F to G represents an ***optimization***: Better in all criteria of interest

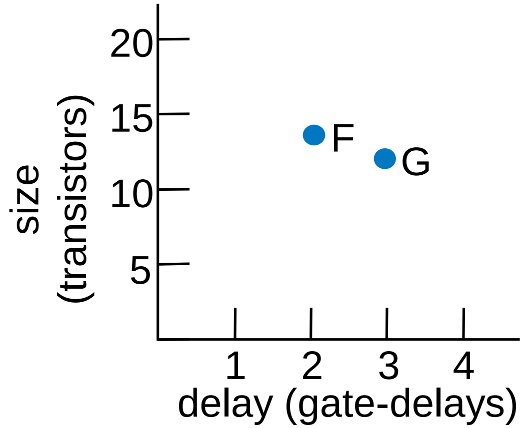
Tradeoff

- F = a + bc + cd



- G = a + c (b + d)





Transforming F to G represents an ***tradeoff***: Some criteria better, others worse

Question (AP)

- How many transistors does the following circuit have?
  - $F = xy + x'y' + w$
  - 10
  - 12
  - 14
  - 16
  - ☹

Question (AP)

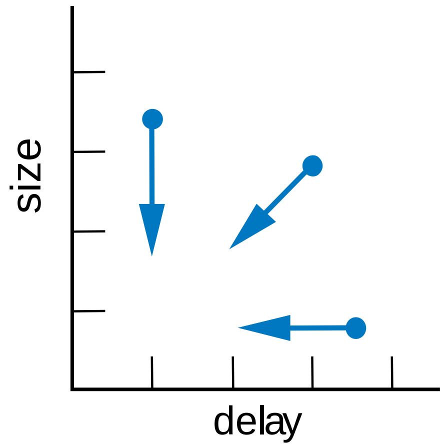
- What is the number of gate delays for the following circuit assuming each gate has 2 inputs?
  – F = xy + x'y' + w

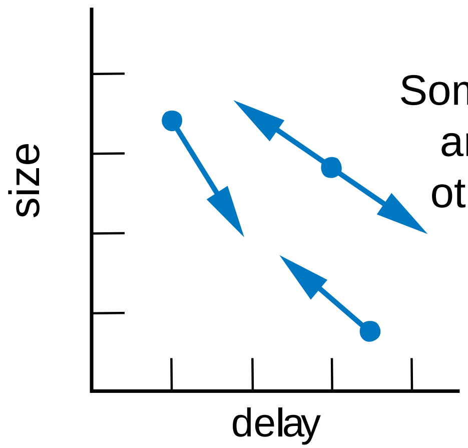
A. 0
B. 1
C. 2
D. 3
E. 4

Optimization vs. Tradeoff

Optimizations

All criteria of interest are improved (or at least kept the same)





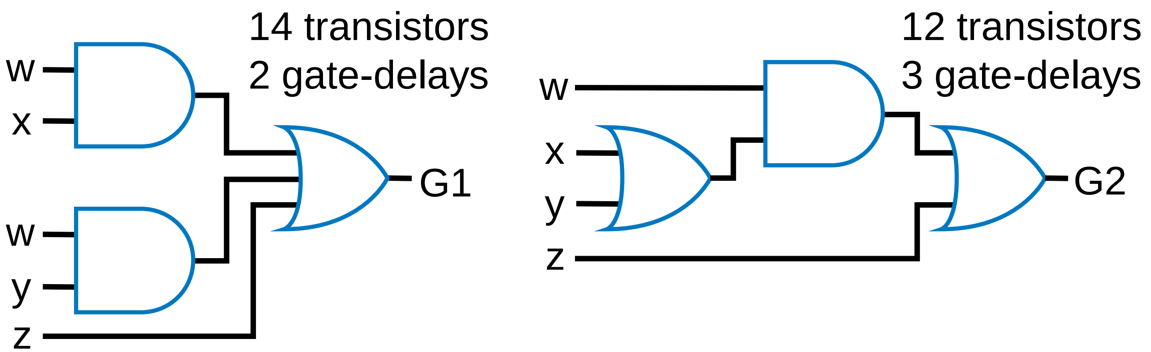
Tradeoffs

Some criteria of interest are improved, while others are worsened

- We obviously prefer optimizations, but often must accept tradeoffs
  - You can’t build a car that is the most comfortable, and has the best fuel efficiency, and is the fastest – you have to give up something to gain other things

How To Determine Which Options Is Better?

- Usually specified by your client, boss, etc…



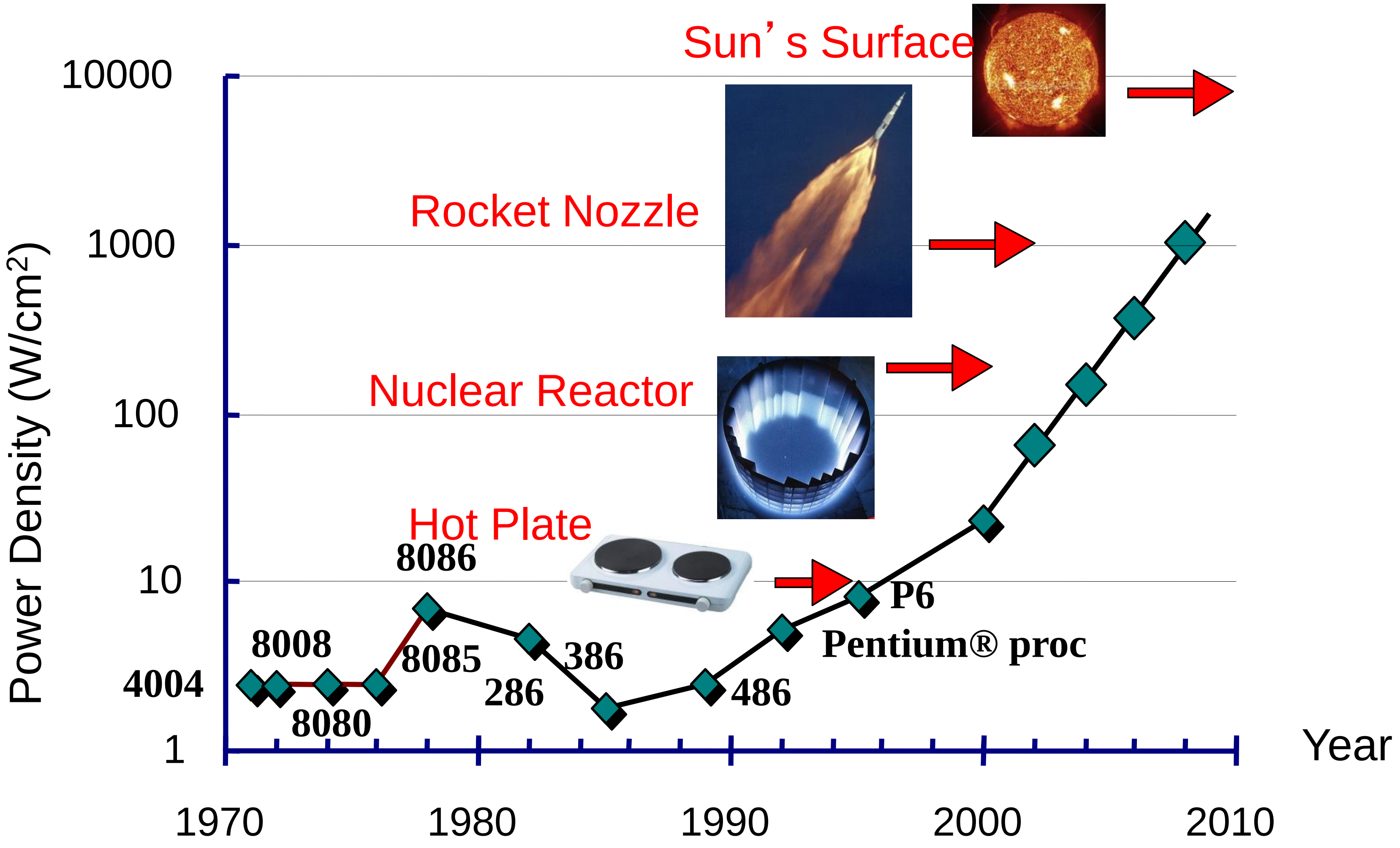
G1 = wx + wy + z

G2 = w(x+y) + z

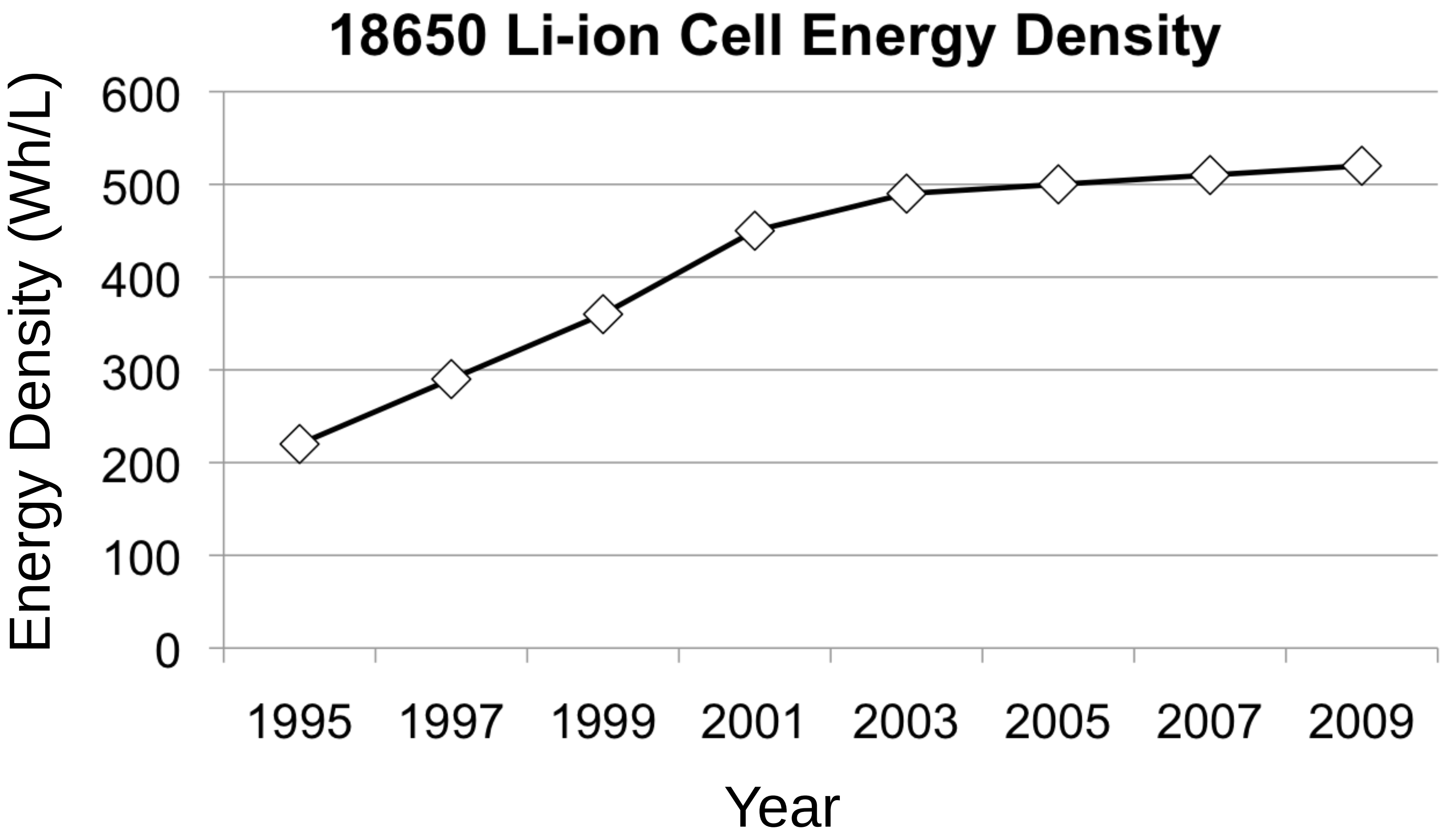
Other Design Criteria

- Power
- Temperature
- Many others…

Temperature Concern



Battery Capacity



Our Focus

- Two-level size optimization using algebraic methods
  - Two-level circuit (ORed AND gates) with fewest transistors
  - Though transistors getting cheaper (Moore's Law), still cost something
- Remember sum-of-products?
  - $F = abc + abc'$ is sum-of-products
    $G = w(xy + z)$ is not
- Transform sum-of-products equation to have fewest literals and terms

Algebraic Two-Level Size Optimization (1)

- Multiply out to sum-of-products, then…
- Apply following as much as possible
  - a**b** + a**b'** = a(**b + b'**) = a***1** = a
    - Combining terms to eliminate a variable
    - (Formally called the “Uniting theorem”)
- Sometimes after combining terms, can combine resulting terms

F = ab’c’ + ab’c + abc’ + abc

F = ab’(c’ + c) + ab(c’ + c)

F = ab’***1** + ab***1**

F = ab’ + ab

F = a(b’ + b)

F = a***1**

F = a

Algebraic Two-Level Size Optimization (2)

- Duplicating a term sometimes helps
  - Doesn't change function
  - c + d = c + d + d = c + d + d + d + d ...

F = ab'c' + ab'c + abc' + abc + a'bc

F = ab'c' + ab'c + abc' + abc + a'bc **+ abc**

F = ab'(c' + c) + ab(c' + c) + (a' + a)bc

F = ab'***1** + ab***1** + **1***bc

F = ab' + ab + bc

F = a(b' + b) + bc

F = a***1** + bc

F = a + bc

Algebraic Two-Level Size Optimization (3)

- Sometimes, it may be hard to see opportunities to simplify equations
  - There are no hard and fast rules on how to simplify an expression
  - Can we guarantee success?

- Use a pictorial form of truth table!

Today's Objectives

- To deepen your understanding on how to use Boolean Algebra to simplify/convert/manipulate digital circuits
- To understand the difference between optimization and tradeoff

Question

- What's the canonical form of the following Boolean equation?

  $F = abc + bc$