UC Berkeley
Teaching Professor
Dan Garcia

CS61C

Great Ideas in Computer Architecture
(a.k.a. Machine Structures)

UC Berkeley
Professor
Bora Nikolić

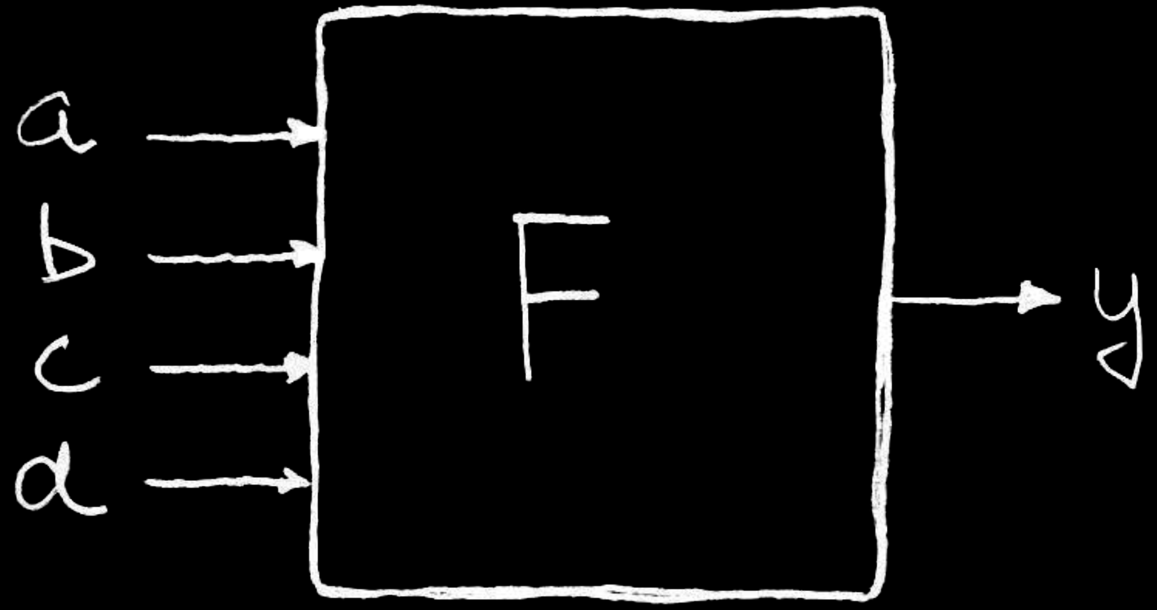
Combinational Logic

Berkeley
UNIVERSITY OF CALIFORNIA

cs61c.org

Truth Tables

Truth Tables



a	b	c	d	y
0	0	0	0	F(0,0,0,0)
0	0	0	1	F(0,0,0,1)
0	0	1	0	F(0,0,1,0)
0	0	1	1	F(0,0,1,1)
0	1	0	0	F(0,1,0,0)
0	1	0	1	F(0,1,0,1)
0	1	1	0	F(0,1,1,0)
0	1	1	1	F(0,1,1,1)
1	0	0	0	F(1,0,0,0)
1	0	0	1	F(1,0,0,1)
1	0	1	0	F(1,0,1,0)
1	0	1	1	F(1,0,1,1)
1	1	0	0	F(1,1,0,0)
1	1	0	1	F(1,1,0,1)
1	1	1	0	F(1,1,1,0)
1	1	1	1	F(1,1,1,1)

How many Fs
(4-input devices)
@Fry's?

Garcia, Nikolić

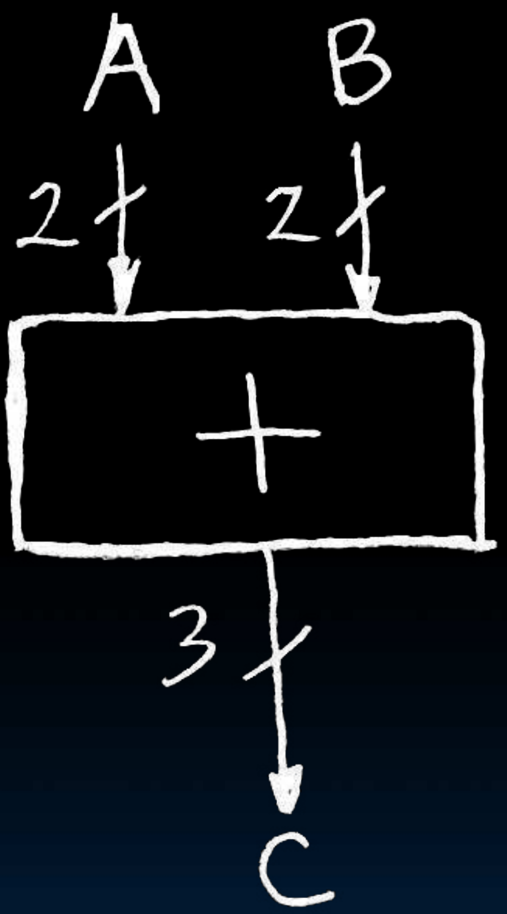
TT Example #1: 1 iff one (not both) a,b=1

a	b	y
0	0	0
0	1	1
1	0	1
1	1	0

⇒

a	y
0	b
1	$\overline{b}$

TT Example #2: 2-bit adder



A	B	C
$a_1 a_0$	$b_1 b_0$	$c_2 c_1 c_0$
00	00	000
00	01	001
00	10	010
00	11	011
01	00	001
01	01	010
01	10	011
01	11	100
10	00	010
10	01	011
10	10	100
10	11	101
11	00	011
11	01	100
11	10	101
11	11	110

How Many Rows?

Garcia, Nikolić

TT Example #3: 32-bit unsigned adder

A	B	C
000 ... 0	000 ... 0	000 ... 00
000 ... 0	000 ... 1	000 ... 01
.	.	.
.	.	.
.	.	.
111 ... 1	111 ... 1	111 ... 10

How Many Rows?

TT Example #4: 3-input majority circuit

a	b	c	y
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

☑

Garcia, Nikolić

Logic Gates

Logic Gates (1/2)

AND

ab	c
00	0
01	0
10	0
11	1

OR

ab	c
00	0
01	1
10	1
11	1

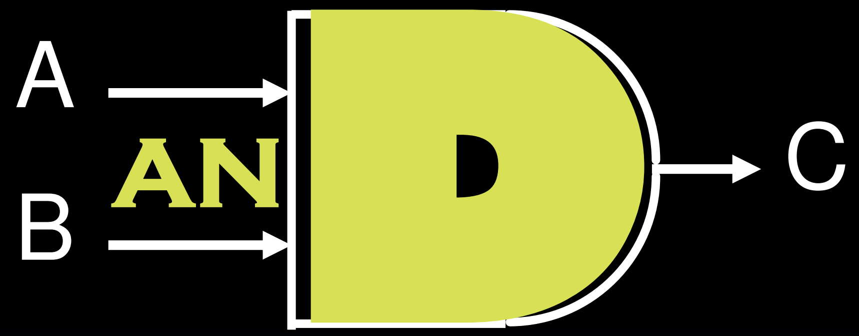
NOT

a	b
0	1
1	0

Garcia, Nikolić

and vs. or … how to recall which is which

and gate symbol



a	b	y
0	0	0
0	1	0
1	0	0
1	1	1

Logic Gates (2/2)

XOR

ab	c
00	0
01	1
10	1
11	0

NAND

ab	c
00	1
01	1
10	1
11	0

NOR

ab	c
00	1
01	0
10	0
11	0

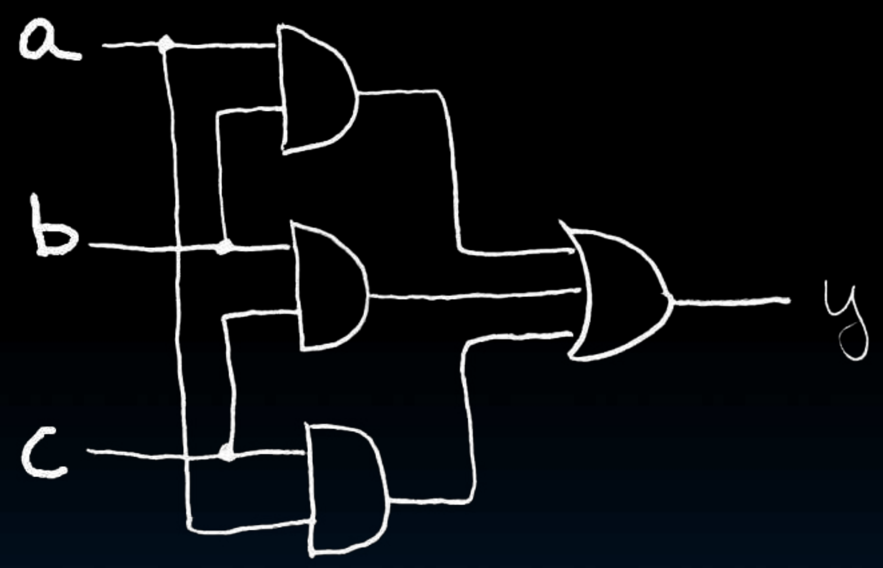
2-input gates extend to n-inputs

- N-input XOR is the only one which isn't so obvious
- It's actually simple…
  - XOR is a 1 iff the # of 1s at its input is odd

a	b	c	y
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

Truth Table → Gates (e.g., majority circ.)

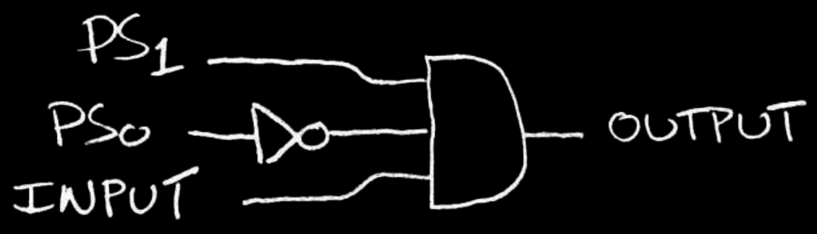
a	b	c	y
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1



Garcia, Nikolić

Truth Table → Gates (e.g., FSM circuit)

PS	Input	NS	Output
00	0	00	0
00	1	01	0
01	0	00	0
01	1	10	0
10	0	00	0
10	1	00	1



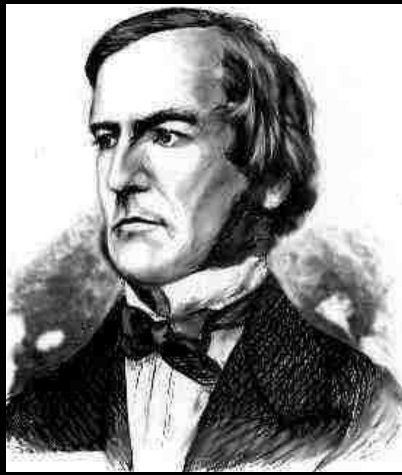
or equivalently…

Garcia, Nikolić

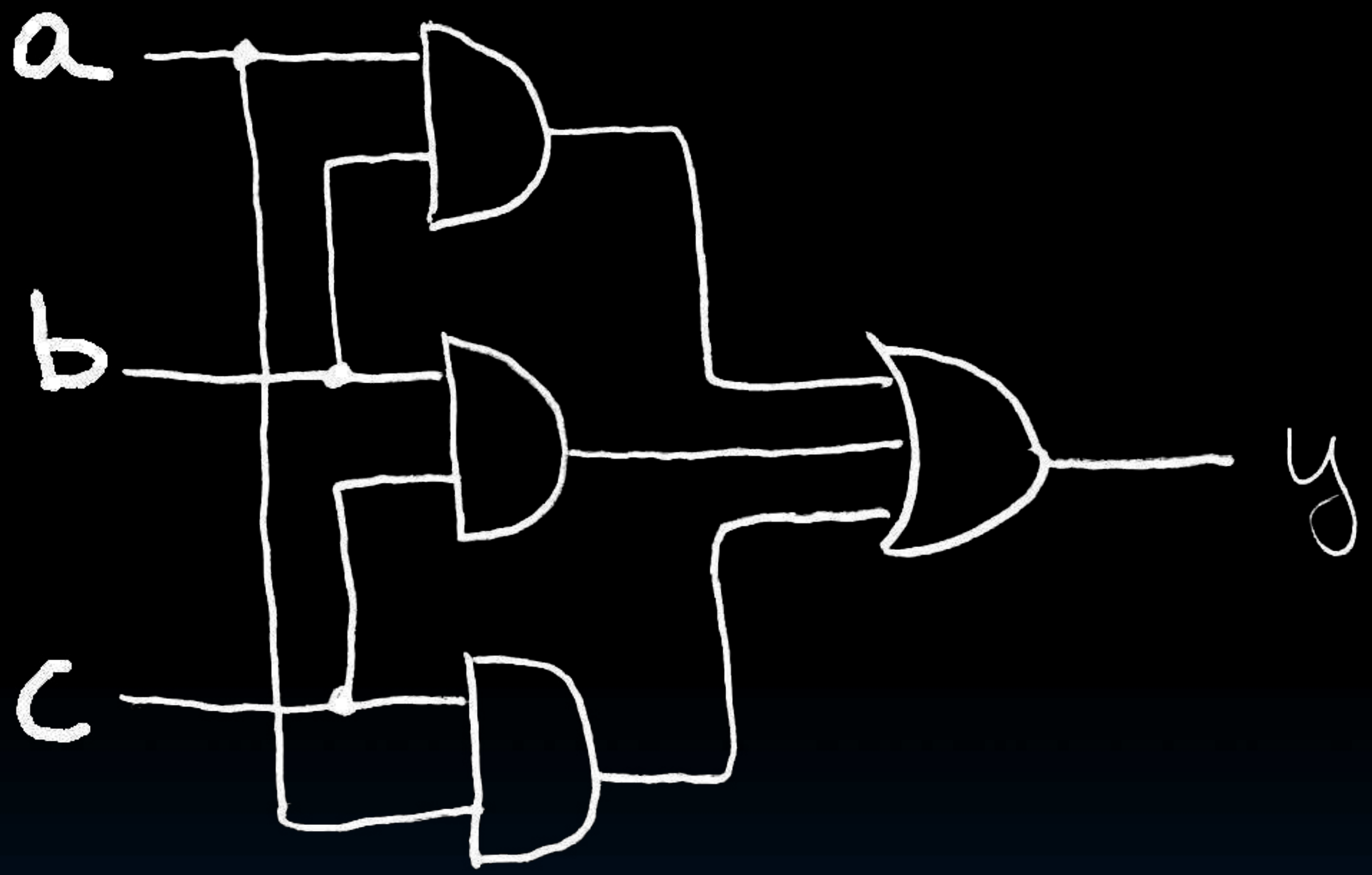
Boolean Algebra

Boolean Algebra

- George Boole, 19th Century mathematician
- Developed a mathematical system (algebra) involving logic
  - later known as “Boolean Algebra”
- Primitive functions: AND, OR and NOT
- Power of Boolean Algebra
  - there’s a one-to-one correspondence between circuits made up of AND, OR and NOT gates and equations in BA
- $+$ means OR, $\cdot$ means AND, $\overline{x}$ means NOT



Boolean Algebra (e.g., for majority fun.)



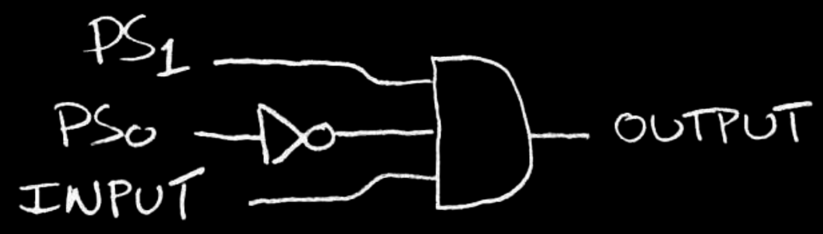
$$y = a \cdot b + a \cdot c + b \cdot c$$

$$y = ab + ac + bc$$

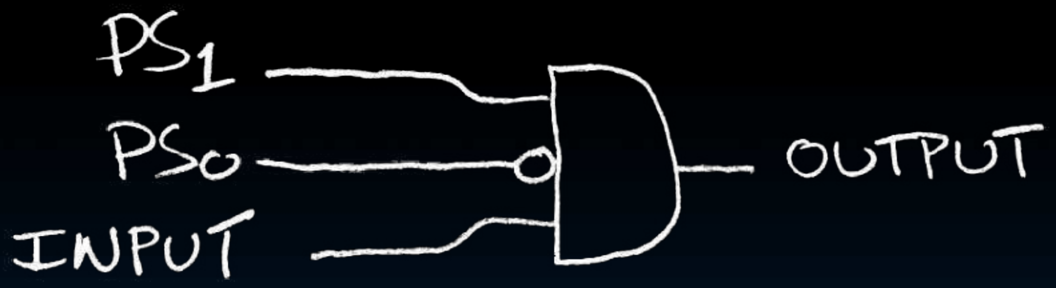
Garcia, Nikolić

Boolean Algebra (e.g., for FSM)

PS	INPUT	NS	OUTPUT
00	0	00	0
00	1	01	0
01	0	00	0
01	1	10	0
10	0	00	0
10	1	00	1

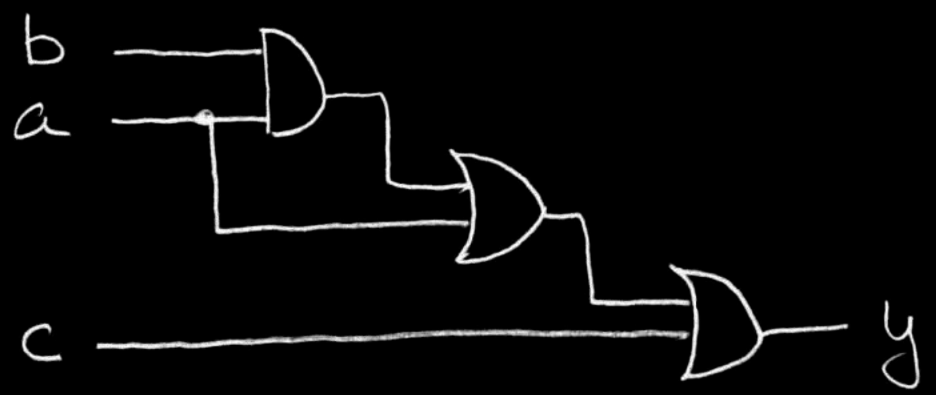


or equivalently…



$$\text{OUTPUT} = PS_1 \cdot \overline{PS_0} \cdot \text{INPUT}$$

BA: Circuit & Algebraic Simplification



original circuit

$$\downarrow$$

$$y = ab + a + c$$

equation derived from original circuit

$$\downarrow$$

$$\begin{aligned} & ab + a + c \\ &= a(b+1) + c \\ &= a(1) + c \\ &= a + c \end{aligned}$$

algebraic simplification

BA also great for circuit verification
Circ X = Circ Y? Use BA to prove!

$$\downarrow$$

simplified circuit

☑

Garcia, Nikolić

Laws of Boolean Algebra

Laws of Boolean Algebra

$x \cdot \overline{x} = 0$	$x + \overline{x} = 1$	complementarity
$x \cdot 0 = 0$	$x + 1 = 1$	laws of 0's and 1's
$x \cdot 1 = x$	$x + 0 = x$	identities
$x \cdot x = x$	$x + x = x$	idempotent law
$x \cdot y = y \cdot x$	$x + y = y + x$	communitive law
$(xy)z = x(yz)$	$(x + y) + z = x + (y + z)$	associativity
$x(y + z) = xy + xz$	$x + yz = (x + y)(x + z)$	distribution
$xy + x = x$	$(x + y)x = x$	uniting theorem
$\overline{x \cdot y} = \overline{x} + \overline{y}$	$\overline{(x + y)} = \overline{x} \cdot \overline{y}$	DeMorgan's Law

Garcia, Nikolić

Boolean Algebraic Simplification Example

$$
\begin{aligned}
y &= ab + a + c \\
&= a(b+1) + c && \textit{distribution, identity} \\
&= a(1) + c && \textit{law of 1's} \\
&= a + c && \textit{identity}
\end{aligned}
$$

Garcia, Nikolić

Canonical Forms

Canonical forms (1/2)

	abc	y
$\overline{a} \cdot \overline{b} \cdot \overline{c}$	000	1
$\overline{a} \cdot \overline{b} \cdot c$	001	1
	010	0
	011	0
$a \cdot \overline{b} \cdot \overline{c}$	100	1
	101	0
$a \cdot b \cdot \overline{c}$	110	1
	111	0

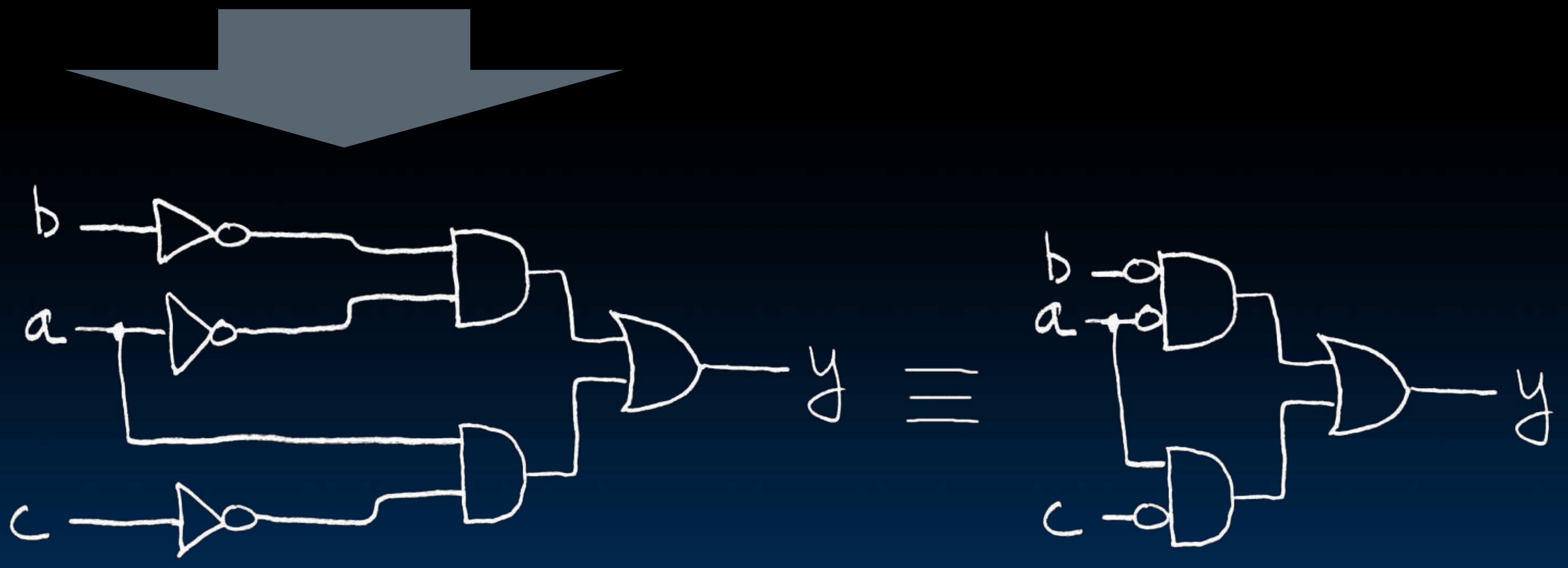
Sum-of-products
(ORs of ANDs)

$$y = \overline{a}\overline{b}\overline{c} + \overline{a}\overline{b}c + a\overline{b}\overline{c} + ab\overline{c}$$

Garcia, Nikolić

Canonical forms (2/2)

$$
\begin{aligned}
y &= \overline{a}\,\overline{b}\,\overline{c} + \overline{a}\,\overline{b}c + a\overline{b}\,\overline{c} + ab\overline{c} & \\
&= \overline{a}\,\overline{b}(\overline{c} + c) + a\overline{c}(\overline{b} + b) & \textit{distribution} \\
&= \overline{a}\,\overline{b}(1) + a\overline{c}(1) & \textit{complementarity} \\
&= \overline{a}\,\overline{b} + a\overline{c} & \textit{identity}
\end{aligned}
$$



Garcia, Nikolić

"And In conclusion…"

- Pipeline big-delay CL for faster clock
- Finite State Machines extremely useful
  - You'll see them again in (at least) 151A, 152 & 164
- Use this table and techniques we learned to transform from 1 to another