Boolean Equivalence

A	B	C	$(A\cdot B)\cdot C$
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	$A\cdot(B\cdot C)$
0	0	0	
0	0	1	
0	1	0	?
0	1	1	
1	0	0	
1	0	1	
1	1	0	
1	1	1	

Are these the same?

Do they have exactly the same output values for exactly the same input values? Check ALL input values.

If so, equivalent: $(A\cdot B)\cdot C = A\cdot(B\cdot C)$

Basic Algebraic Properties

equivalent functions

Proof by truth table.

logical constant = TRUE for all inputs.

$1\cdot P = P$

proof

same function

P	$1\cdot P$
0	0
1	1

=

P	P
0	0
1	1

$0\cdot P = 0$

$P\cdot P = P$

proof

P	$P\cdot P$
0	0
1	1

=

P	P
0	0
1	1

$P\cdot\overline{P} = 0$

proof

P	$P\cdot\overline{P}$
0	0
1	0

always 0

$A\cdot B = B\cdot A$

$(A\cdot B)\cdot C = A\cdot(B\cdot C)$

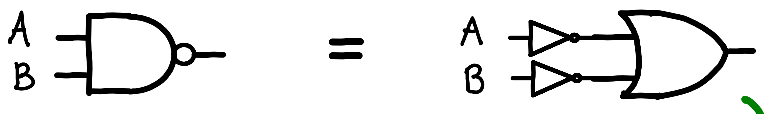
Handy tricks: DeMorgan's Laws and Duality (Aside: can you prove Duality?)
(1) prove algebraic properties or do algebraic manipulations,
(2) convert one type of logic gate to another.

DeMorgan's Laws

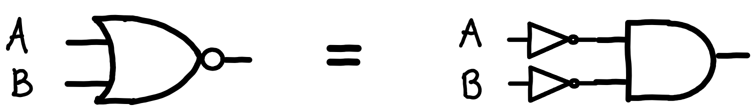
$$\overline{A \cdot B} = \bar{A} + \bar{B}$$

A	B	$\overline{A \cdot B}$
0	0	1
0	1	1
1	0	1
1	1	0

A	B	$\bar{A} + \bar{B}$
0	0	1
0	1	1
1	0	1
1	1	0



DUAL



$$\overline{A + B} = \bar{A} \cdot \bar{B}$$

DUALITY

$$\left\{ \begin{matrix} \cdot \\ + \\ 0 \\ 1 \end{matrix} \right\} \rightarrow \left\{ \begin{matrix} + \\ \cdot \\ 1 \\ 0 \end{matrix} \right\}$$

starting with an identity, gives an identity.

$$1 \cdot P = P \rightarrow 0 + P = P$$

$$0 \cdot P = 0 \rightarrow 1 + P = 1$$

$$(A+B)C = AC + BC$$

$$\Big\downarrow \text{dual}$$

$$AB + C = (A+C)(B+C)$$

Can We Build ANY k-input function? YES ===> OR k-input minterm functions.

A	B	C	f
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	0

=

A	B	C	m_0
0	0	0	1
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	0

+

A	B	C	m_3
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	0

+

A	B	C	m_6
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	1
1	1	1	0

$$f(A,B,C) = m_0(A,B,C) + m_3(A,B,C) + m_6(A,B,C)$$

Actually, we ONLY know how to build 2-input functions (NAND, NOR) and NOT (AND and OR).

PROVE that OR and AND are associative: then we can use 2-input gates.

MAXTERMS

We found a complete orthogonal set of functions, minterms.

Are there other complete orthogonal sets? YES.

A	B	C	f
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

six minterms

But

A	B	C	NOT(f)
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	0

only 2 minterms

$\bar{A}\bar{B}\bar{C}$

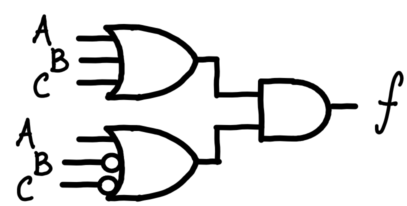
$\bar{A}BC$

Simplify by noticing

NOT(NOT(f)) == f

$$f = \bar{\bar{f}} = \overline{\bar{A}\bar{B}\bar{C} + \bar{A}BC} \overset{\text{DeMorgan}}{=} \overline{\bar{A}\bar{B}\bar{C}} \cdot \overline{\bar{A}BC}$$

$$\overset{\text{DeMorgan}}{=} \underbrace{(A+B+C)}_{\text{maxterms}} \cdot \underbrace{(A+\bar{B}+\bar{C})}_{\text{maxterms}}$$



Maxterms are also a complete orthogonal set.

OR-AND Tree

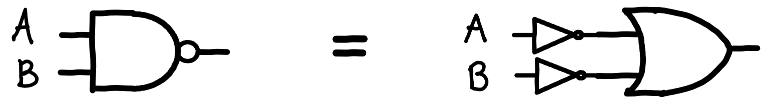
AND-OR and OR-AND Trees can express any function:

====> { AND, OR, NOT } is a complete set of logical primitives

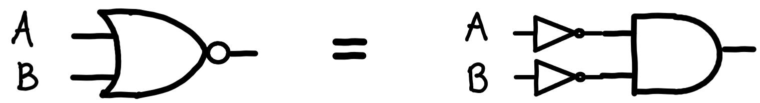
Simplifying Logic Circuits

DeMorgan's Laws can be used to change NAND to NOR and vice versa

$\overline{A \cdot B} = \bar{A} + \bar{B}$

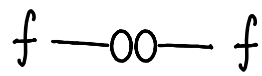


$\overline{A + B} = \bar{A} \cdot \bar{B}$

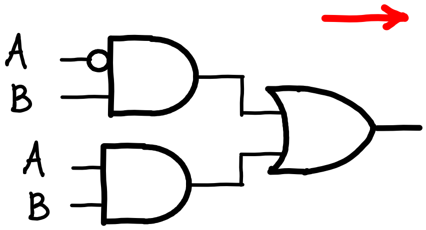


push bubble to inputs

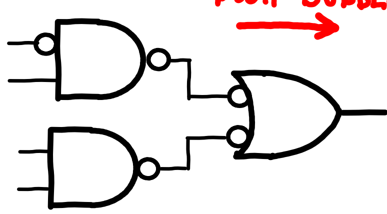
Combine with $f = \bar{\bar{f}}$



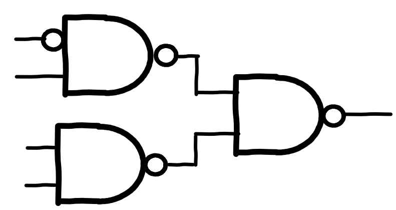
AND-OR ⇒ NAND-NAND



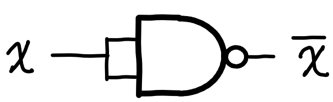
ADD BUBBLES



PUSH BUBBLES



NAND can do NOT

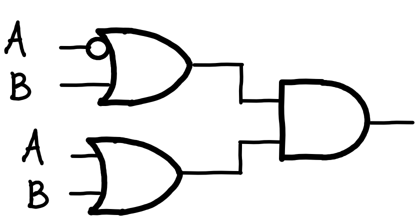


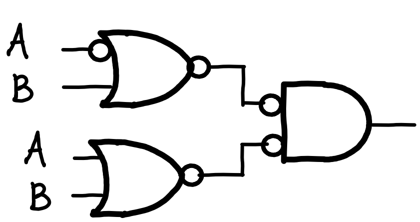
A	B	NAND
0	0	1
0	1	1
1	0	1
1	1	0

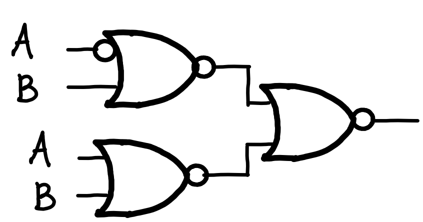
NAND alone is a complete logic set.

OR-AND ⇒ NOR-NOR

NOR-NOR is also complete.







Karnaugh Maps are truth tables ===> **See simpler logic**.
Similar to extracting minterms, but not orthogonal.
Can be done algebraically, but usually harder.
Is there a general procedure for minimizing circuits? (Is it computable?)

A	B	f
0	0	f_0
0	1	f_1
1	0	f_2
1	1	f_3

$\rightarrow$

A \ B	0	1
0	f_0	f_1
1	f_2	f_3

IF

f_0 == f_1 == 1

THEN

f **== 1 when A = 0 (B doesn't matter)**

A	B	f
0	0	1
0	1	1
1	0	0
1	1	1

$\rightarrow$

A \ B	0	1
0	1	1
1	0	1

$$\bar{A} + B = f$$

Find a collection of terms that:

1. covers all ones

2. has the fewest terms,
each term covers as many 1s as possible

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

$\bar{A}\bar{C}$ (B changes, but f doesn't)

B (A and C change, but f doesn't)

AC (B changes, but f doesn't)

$$f = \bar{A}\bar{C} + B + AC$$

General Algorithm:

Quine-McClusky