Canonical forms

- Last lecture
  - Logic gates and truth tables
  - Implementing logic functions
  - CMOS switches
- Today's lecture
  - deMorgan's theorem
  - NAND and NOR
  - Canonical forms
    - Sum-of-products (minterms)
    - Product-of-sums (maxterms)

de Morgan's theorem

- Replace
  - • with +, + with •, 0 with 1, and 1 with 0
  - All variables with their complements
- Example 1: Z = A'B' + A'C'

  Z' = (A'B' + A'C')'
  = (A'B')' • (A'C')'
  = (A+B) • (A+C)

- Example 2: Z = A'B'C + A'BC + AB'C + ABC'

  Z' = (A'B'C + A'BC + AB'C + ABC')'
  = (A'B'C)' • (A'BC)' • (AB'C)' • (ABC')'
  = (A+B+C') • (A+B'+C') • (A'+B+C') • (A'+B'+C)

NAND and NOR

(X + Y)' = X' • Y'
NOR is equivalent to AND
with inputs complemented

X	Y	X'	Y'	(X + Y)'	X' • Y'
0	0	1	1	1	1
0	1	1	0	0	0
1	0	0	1	0	0
1	1	0	0	0	0

(X • Y)' = X' + Y'
NAND is equivalent to OR
with inputs complemented

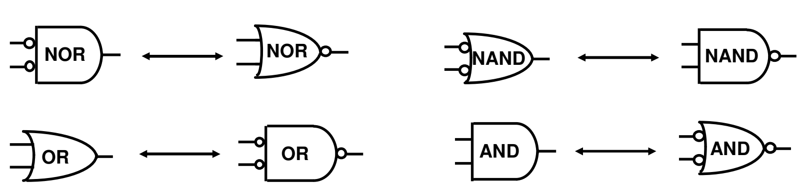
X	Y	X'	Y'	(X • Y)'	X' + Y'
0	0	1	1	1	1
0	1	1	0	1	1
1	0	0	1	1	1
1	1	0	0	0	0

NAND, NOR, and de Morgan's theorum

- de Morgan's
  - Standard form: A'B' = (A + B)' A' + B' = (AB)'
  - Inverted: A + B = (A'B')' (AB) = (A' + B')'

  - AND with complemented inputs ≡ NOR
  - OR with complemented inputs ≡ NAND
  - OR ≡ NAND with complemented inputs
  - AND ≡ NOR with complemented inputs

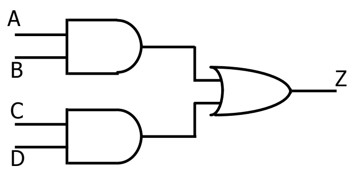
pushing the bubble

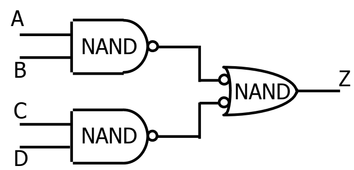


Converting between forms

- Introduce inversions ("bubbles")
  - Introduce bubbles in pairs
    - Conserve inversions
    - Do not alter logic function
- Example
  - AND/OR to NAND/NAND

$Z = AB + CD$
$= (A'+B')'+(C'+D')'$
$= [(A'+B')(C'+D')]'$
$= [(AB)'(CD)']'$

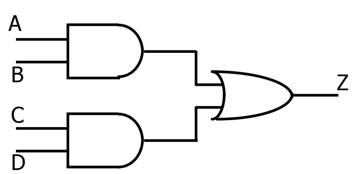


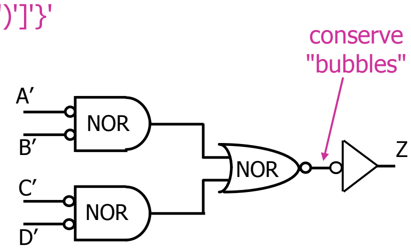


Converting between forms (con't)

- Example: AND/OR network to NOR/NOR

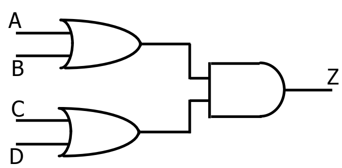
$Z = AB+CD$
$= (A'+B')'+(C'+D')'$
$= [(A'+B')'+(C'+D')']''$
$= \{[(A'+B')'+(C'+D')']'\}'$

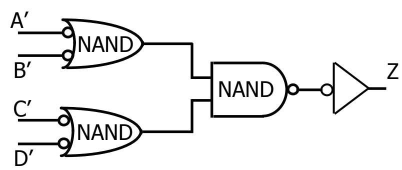




Converting between forms (con't)

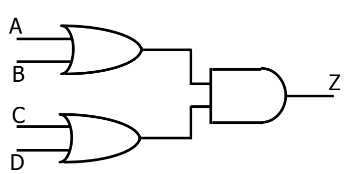
- Example: OR/AND to NAND/NAND

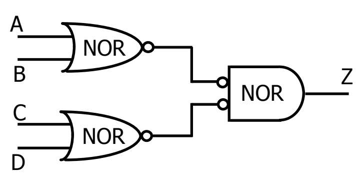




Converting between forms (con't)

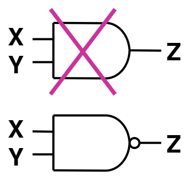
- Example: OR/AND to NOR/NOR



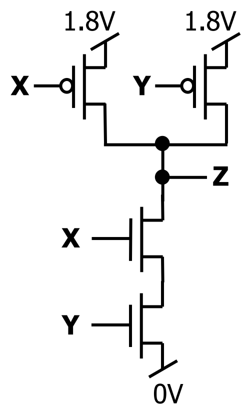


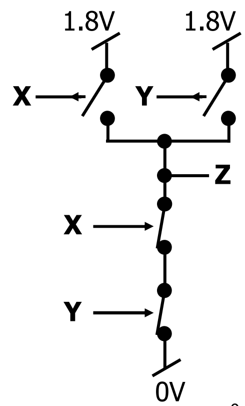
Why convert between forms?

- CMOS logic gates are inverting
  - Get NAND, NOR, NOT
  - Don’t get AND, OR, Buffer



X	Y	Z
0	0	1
0	1	1
1	0	1
1	1	0





Canonical forms

- Canonical forms
  - Standard forms for Boolean expressions
  - Unique algebraic signatures
  - Generally not the simplest forms
    - Can be minimized
  - Derived from truth table
- Two canonical forms
  - Sum-of-products (minterms)
  - Product-of-sum (maxterms)

Sum-of-products canonical form

- Also called disjunctive normal form
  - Commonly called a **minterm expansion**

A	B	C	F	F'
0	0	0	0	1
0	0	1	1	0
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	0

minterm

001 011 101 110 111

F = A'B'C + A'BC + AB'C + ABC' + ABC

F' = A'B'C' + A'BC' + AB'C'

Minterms

- Variables appears exactly once in each minterm
  - In true or inverted form (but not both)

A	B	C	minterms	
0	0	0	A'B'C'	m0
0	0	1	A'B'C	m1
0	1	0	A'BC'	m2
0	1	1	A'BC	m3
1	0	0	AB'C'	m4
1	0	1	AB'C	m5
1	1	0	ABC'	m6
1	1	1	ABC	m7

short-hand notation

F in canonical form:

F(A,B,C) = Σm(1,3,5,6,7)
= m1 + m3 + m5 + m6 + m7
= A'B'C+A'BC+AB'C+ABC'+ABC

canonical form → minimal form

F(A,B,C) = A'B'C+A'BC+AB'C+ABC+ABC'
= (A'B'+A'B+AB'+AB)C+ABC'
= ((A' + A)(B' + B))C + ABC'
= ABC' + C
= AB + C

Product-of-sums canonical form

- Also called conjunctive normal form
  - Commonly called a **maxterm expansion**

maxterm

000 010 100

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

A	B	C	F	F'
0	0	0	0	1
0	0	1	1	0
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	0

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

Maxterms

- Variables appears exactly once in each maxterm
  - In true or inverted form (but not both)

A	B	C	maxterms	
0	0	0	A+B+C	M0
0	0	1	A+B+C'	M1
0	1	0	A+B'+C	M2
0	1	1	A+B'+C'	M3
1	0	0	A'+B+C	M4
1	0	1	A'+B+C'	M5
1	1	0	A'+B'+C	M6
1	1	1	A'+B'+C'	M7

short-hand notation

F in canonical form:

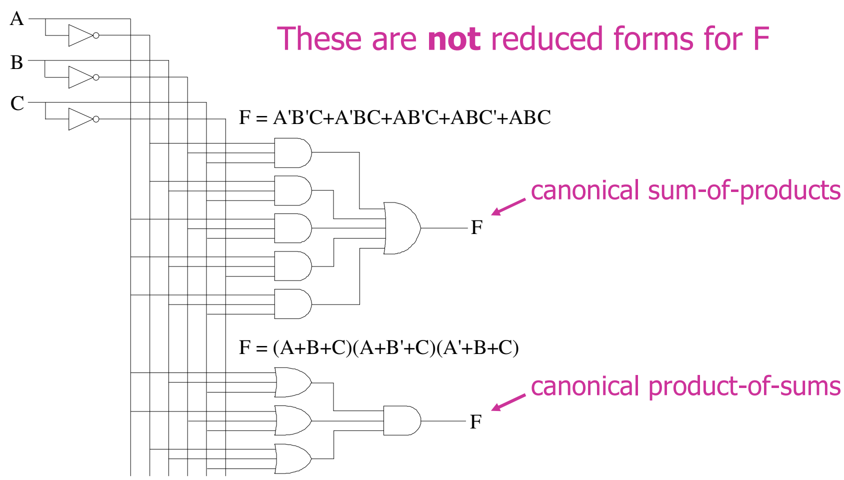
F(A,B,C) = ΠM(0,2,4)
= M0 • M2 • M4
= (A+B+C)(A+B'+C)(A'+B+C)

canonical form → minimal form

F(A,B,C) = (A+B+C)(A+B'+C)(A'+B+C)
= (A+B+C)(A+B'+C)•
(A+B+C)(A'+B+C)
= (A + C)(B + C)

Canonical implementations of F = AB + C

These are **not** reduced forms for F



SOP, POS, and de Morgan's theorem

- Sum-of-products
  - F' = A'B'C' + A'BC' + AB'C'
- Apply de Morgan's to get POS
  - (F')' = (A'B'C' + A'BC' + AB'C')'
  - F = (A+B+C)(A+B'+C)(A'+B+C)
- Product-of-sums
  - F' = (A+B+C')(A+B'+C')(A'+B+C')(A'+B'+C)(A'+B'+C')
- Apply de Morgan's to get SOP
  - (F')' = ((A+B+C')(A+B'+C')(A'+B+C')(A'+B'+C)(A'+B'+C'))'
  - F = A'B'C + A'BC + AB'C + ABC' + ABC

Conversion between canonical forms

- Minterm to maxterm
  - Use maxterms that aren't in minterm expansion
  - $F(A,B,C) = \sum m(1,3,5,6,7) = \prod M(0,2,4)$
- Maxterm to minterm
  - Use minterms that aren't in maxterm expansion
  - $F(A,B,C) = \prod M(0,2,4) = \sum m(1,3,5,6,7)$
- Minterm of F to minterm of F'
  - Use minterms that don't appear
  - $F(A,B,C) = \sum m(1,3,5,6,7)$ $F'(A,B,C) = \sum m(0,2,4)$
- Maxterm of F to maxterm of F'
  - Use maxterms that don't appear
  - $F(A,B,C) = \prod M(0,2,4)$ $F'(A,B,C) = \prod M(1,3,5,6,7)$