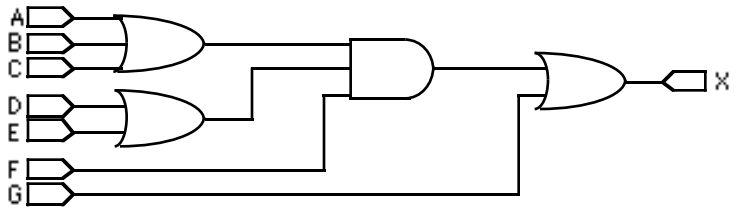
Chapter overview

- Multi-level logic
  - conversion to NAND-NAND and NOR-NOR Networks
  - de Morgan's law and OR-AND translation
  - AND-OR-Invert building blocks
- Time response in combinational networks
  - gate delays and timing waveforms
  - hazards/glitches and how to avoid them

© 1992 Gaetano Borriello, 1994 Steven Burns

Multi-level logic: advantages

- Reduced sum-of-products form:
  - x = A D F + A E F + B D F + B E F + C D F + C E F + G
  - 6 x 3-input AND gates + 1 x 7-input OR gate
  - 25 wires (19 literals plus 6 internal wires)
- Factored form:
  - x = (A + B + C) (D + E) F + G
  - 1 x 3-input OR gate, 2 x 2-input OR gates, 1 x 3-input AND gate
  - 10 wires (7 literals plus 3 internal wires)

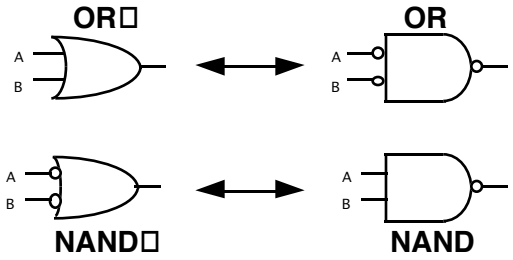


© 1992 Gaetano Borriello, 1994 Steven Burns

Multi-level logic: conversion between forms

- NAND-NAND and NOR-NOR networks
  - de Morgan's law: (A + B)' = A' · B' (A · B)' = A' + B'
  - written differently: A + B = (A' · B')' (A · B) = (A' + B')'
- In other words —
  - OR is the same as NAND with complemented inputs
  - AND is the same as NOR with complemented inputs
  - NAND is the same as OR with complemented inputs
  - NOR is the same as AND with complemented inputs

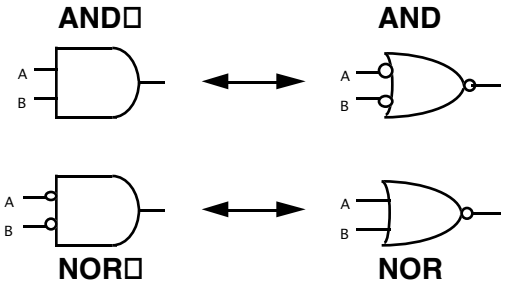
OR/NAND equivalence



© 1992 Gaetano Borriello, 1994 Steven Burns

Multi-level logic: conversion between forms

AND/NOR equivalence

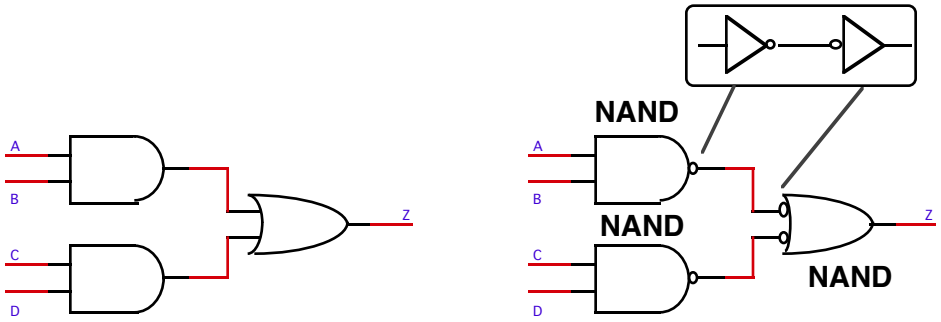


- It is possible to convert from networks with ANDs and ORs to networks with NANDs and NORs by introducing appropriate inversions ("bubbles")
- To preserve logic function, each introduced "bubble" must be matched with a corresponding "bubble"

© 1992 Gaetano Borriello, 1994 Steven Burns

Multi-level logic: conversion between forms

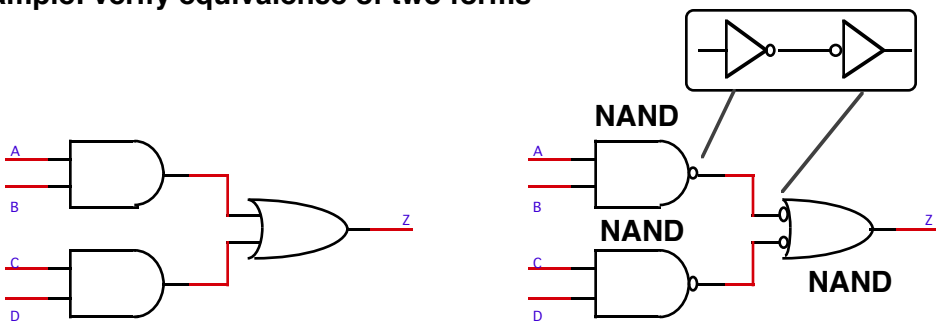
- Example: map AND/OR network to NAND/NAND network



© 1992 Gaetano Borriello, 1994 Steven Burns

Multi-level logic: conversion between forms

- Example: verify equivalence of two forms



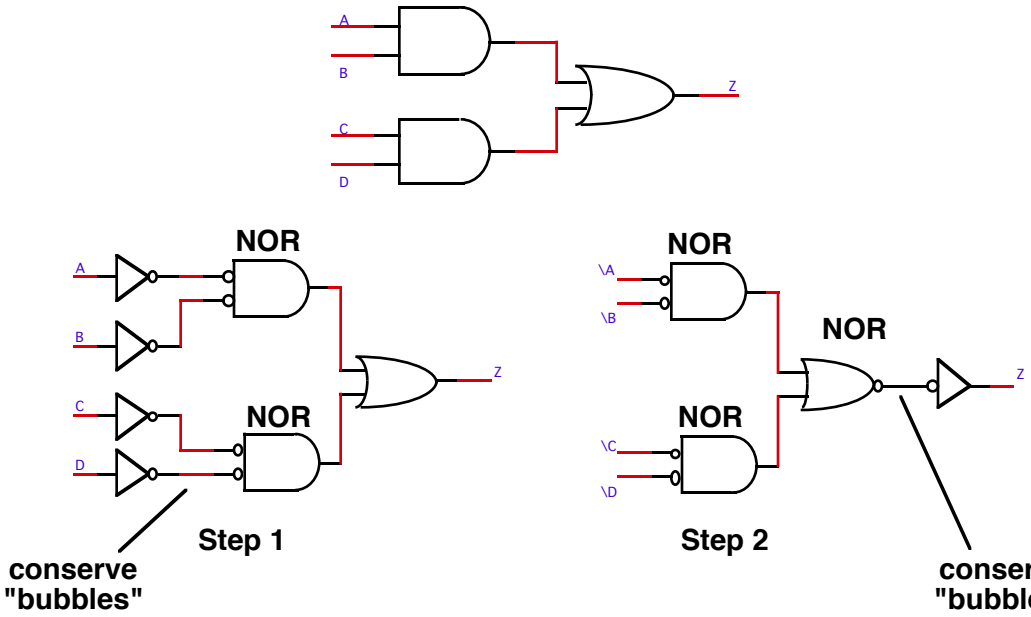
$$Z = [\,(A \cdot B)' \cdot (C \cdot D)'\,]'$$
$$= [\,(A' + B') \cdot (C' + D')\,]'$$
$$= [\,(A' + B')' + (C' + D')'\,]$$
$$= (A \cdot B) + (C \cdot D) \surd$$

this is the easy conversion!

© 1992 Gaetano Borriello, 1994 Steven Burns

Multi-level logic: conversion between forms

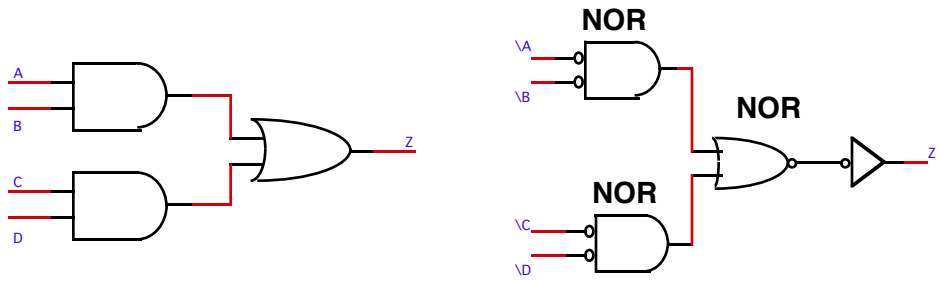
- Example: map AND/OR network to NOR/NOR network



© 1992 Gaetano Borriello, 1994 Steven Burns

Multi-level logic: conversion between forms

- Example: verify equivalence of two forms



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

$= \{ (A' + B') \cdot (C' + D') \}'$

$= (A' + B')' + (C' + D')'$

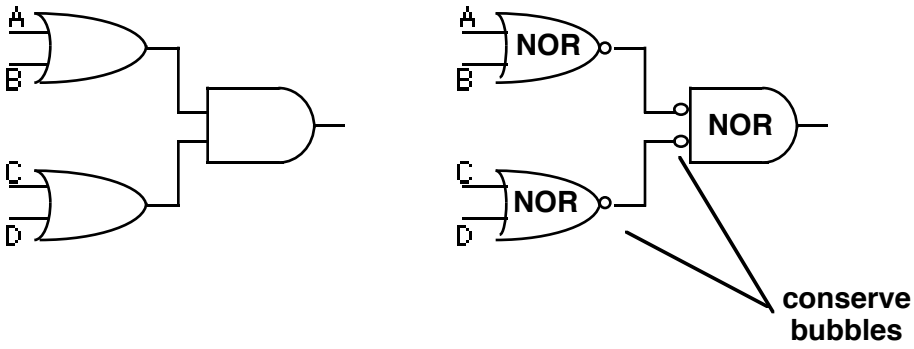
$= (A \cdot B) + (C \cdot D)$ √

this is the hard conversion!
AND/OR to NAND/NAND more natural

© 1992 Gaetano Borriello, 1994 Steven Burns

Multi-level logic: conversion between forms

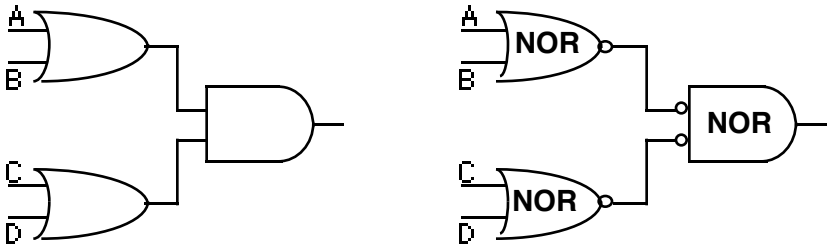
- Example: map OR/AND network to NOR/NOR network



© 1992 Gaetano Borriello, 1994 Steven Burns

Multi-level logic: conversion between forms

- Example: verify equivalence of two forms



$Z = [\,(A + B)' + (C + D)'\,]'$

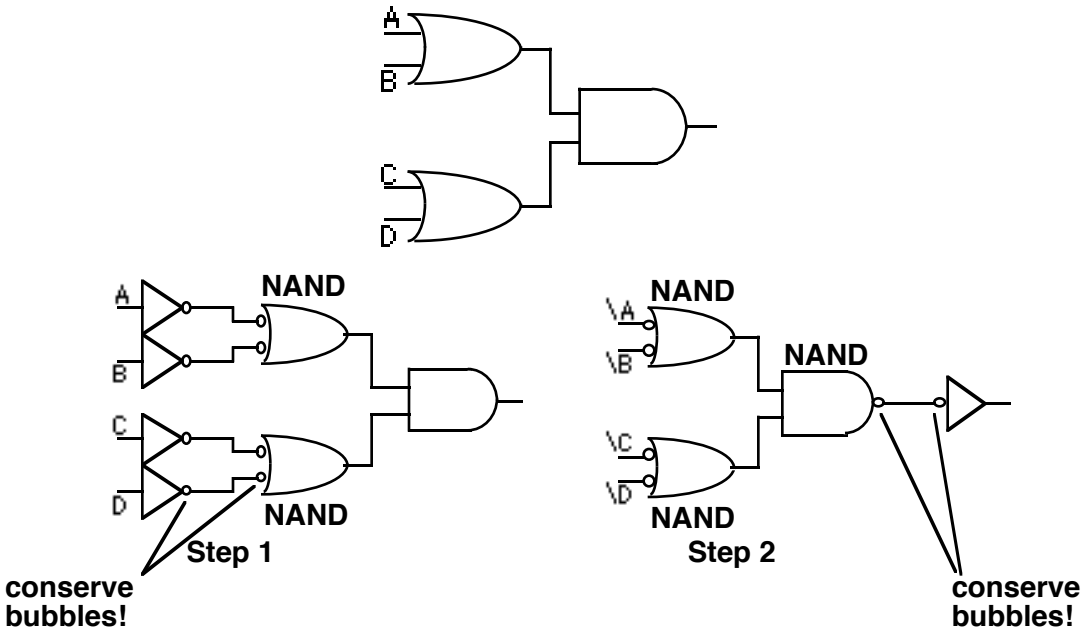
$= (A + B) \cdot (C + D)$ √

this is the easy conversion!

© 1992 Gaetano Borriello, 1994 Steven Burns

Multi-level logic: conversion between forms

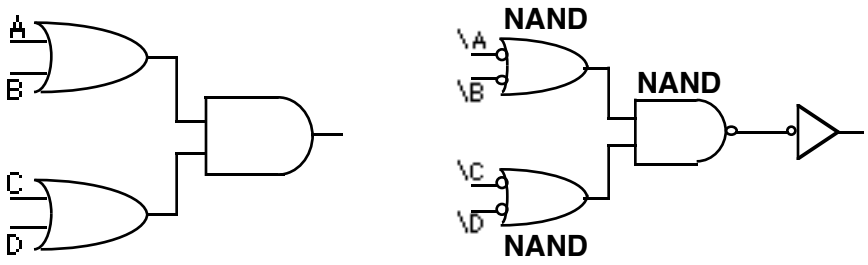
- Example: map OR/AND network to NAND/NAND network



© 1992 Gaetano Borriello, 1994 Steven Burns

Multi-level logic: conversion between forms

- Example: verify equivalence of two forms



$Z = \{ [(A' \cdot B')' \cdot (C' \cdot D')']' \}'$

$= \{ (A' \cdot B') + (C' \cdot D') \}'$

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

$= (A + B) \cdot (C + D)$ √

this is the hard conversion!
OR/AND to NOR/NOR more natural

© 1992 Gaetano Borriello, 1994 Steven Burns

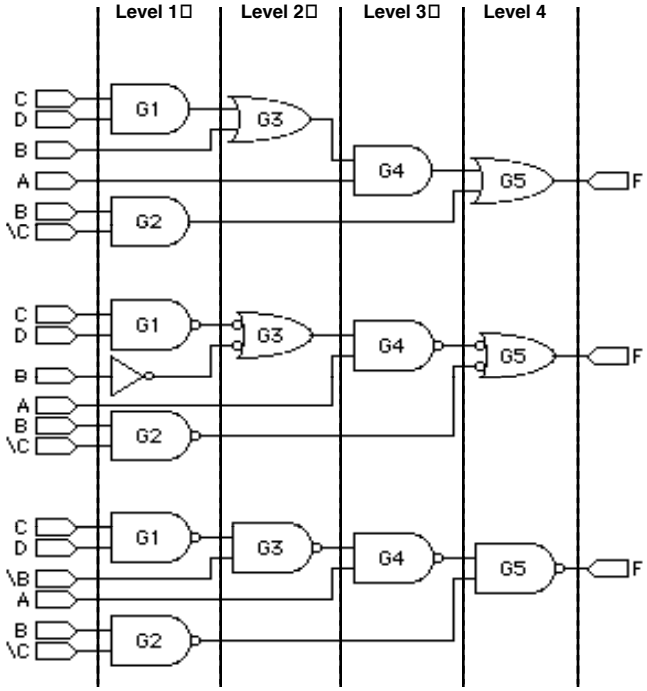
Multi-level logic: conversion between forms

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

original AND-OR Network

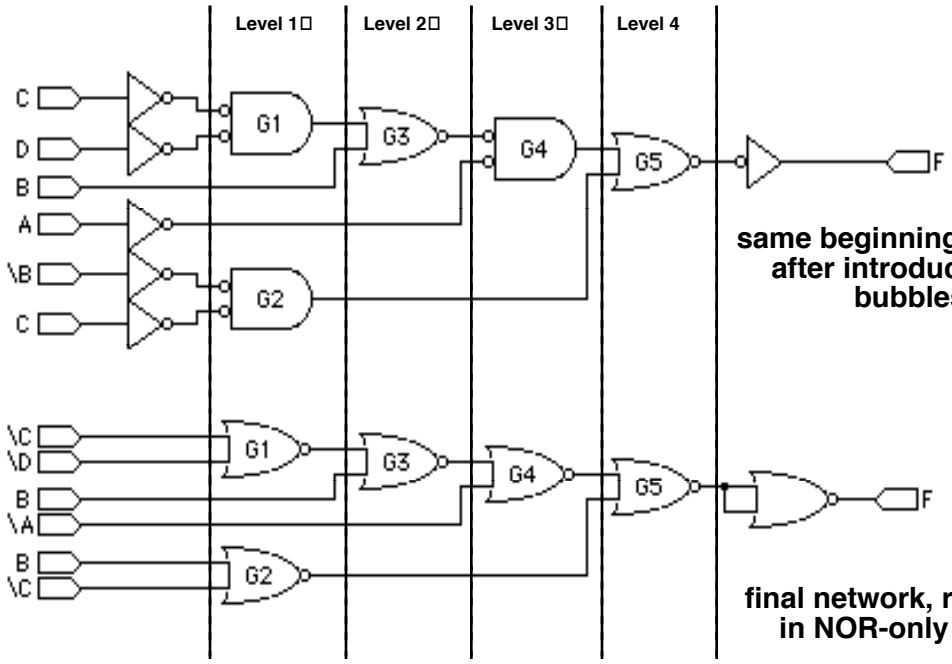
introduction and conservation of bubbles

redrawn in terms of conventional NAND gates



© 1992 Gaetano Borriello, 1994 Steven Burns

Multi-level logic: conversion between forms



same beginning network after introduction of bubbles

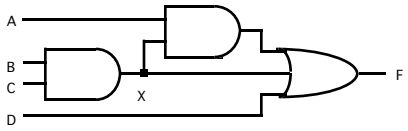
final network, redrawn in NOR-only form

© 1992 Gaetano Borriello, 1994 Steven Burns

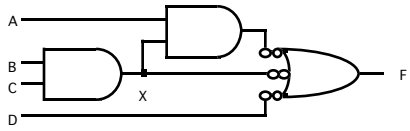
Multi-level logic: conversion between forms

- **Conversion example**

(a)



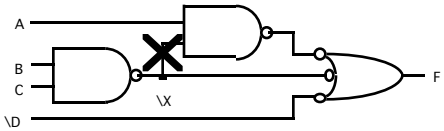
original circuit



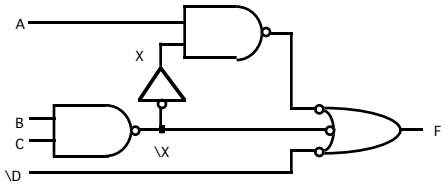
(b)

add double bubbles at inputs

(c)



distribute bubbles
some mismatches



(d)

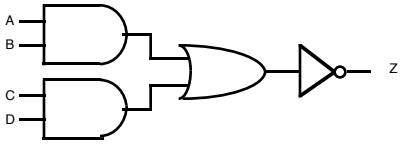
insert inverters to fix mismatches

© 1992 Gaetano Borriello, 1994 Steven Burns

Multi-level logic: AND-OR-Invert

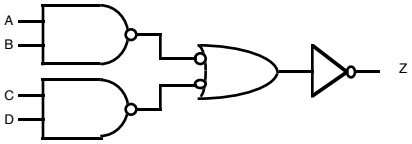
- **AOI function: three stages of logic — AND, OR, Invert**
  - **multiple gates "packaged" as a single circuit block**

logical concept



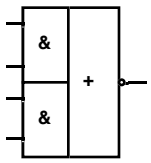
AND OR Invert

possible implementation

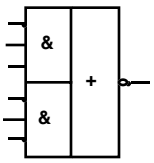


NAND NAND Invert

2x2 AOI schematic symbol



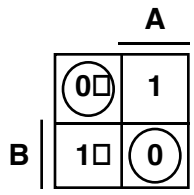
3x2 AOI schematic symbol



© 1992 Gaetano Borriello, 1994 Steven Burns

Multi-level logic: AND-OR-Invert

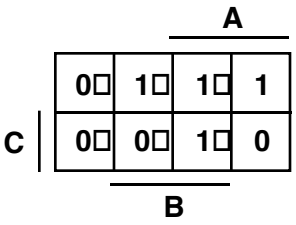
- General procedure to place in AOI form:
  - compute the complement of the function in sum-of-products form
  - by circling the 0s in the Karnaugh map
- Example: XOR implementation — A xor B = A' B + A B'
  - AOI form: F = (A' B' + A B)'



© 1992 Gaetano Borriello, 1994 Steven Burns

Multi-level logic: AND-OR-Invert

- Example:



F = B C' + A C' + A B
F' = A' B' + A' C + B' C
Implemented by 2-input 3-stack AOI gate

F = (A + B) (A + C') (B + C')
F' = (B' + C) (A' + C) (A' + B')
Implemented by 2-input 3-stack OAI gate

- Example: 4-bit equality function

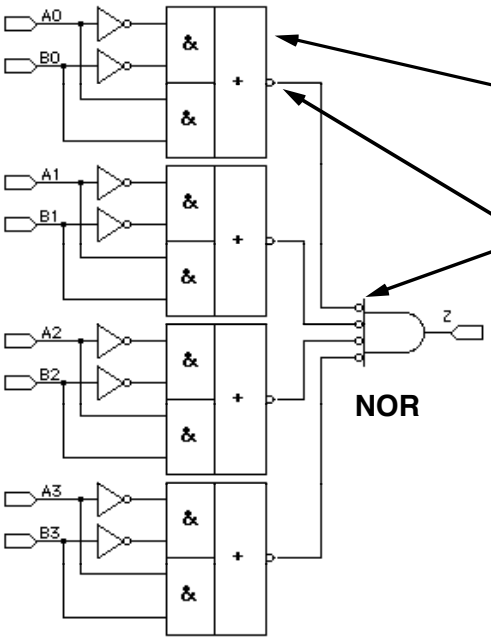
Z = (A0 B0 + A0' B0') (A1 B1 + A1' B1') (A2 B2 + A2' B2') (A3 B3 + A3' B3')

Each implemented in single 2x2 AOI gate

© 1992 Gaetano Borriello, 1994 Steven Burns

Multi-level logic: AND-OR-Invert

- Example: AOI implementation of 4-bit equality function



high if A0 ≠ B0
low if A0 = B0
A = B active low

conservation of bubbles

NOR

if all inputs are low
(asserted in negative logic)
then Ai = Bi, i=0,...,3
output Z asserted

© 1992 Gaetano Borriello, 1994 Steven Burns

Multi-level logic: summary

- Advantages
  - circuits may be smaller
  - gates have smaller fan-in
  - circuits may be faster
- Disadvantages
  - more difficult to design
  - tools for optimization are not as mature or as good
  - analysis is more complex

© 1992 Gaetano Borriello, 1994 Steven Burns