

VLSI Design

Lecture 6: Combinational Logic Gates

Shaahin Hessabi

Department of Computer Engineering

Sharif University of Technology

Adapted with modifications from lecture notes prepared by
author (from Prentice Hall PTR)

Topics

- Combinational logic functions
- Static complementary logic gate structures

Combinational logic expressions

- Combinational logic: function value is a combination of function arguments.
- A logic gate implements a particular logic function.
- Both specification (logic equations) and implementation (logic gate networks) are written in Boolean logic.

Gate design

Why designing gates for logic functions is non-trivial:

- may not have logic gates in the library for all logic expressions;
- a logic expression may map into gates that consume a lot of area, delay, or power.

Boolean algebra terminology

- Function:
 $f = a'b + ab'$
- a is a variable; a and a' are literals.
- ab' is a term.
- A function is irredundant if no literal can be removed without changing its truth value.

Completeness

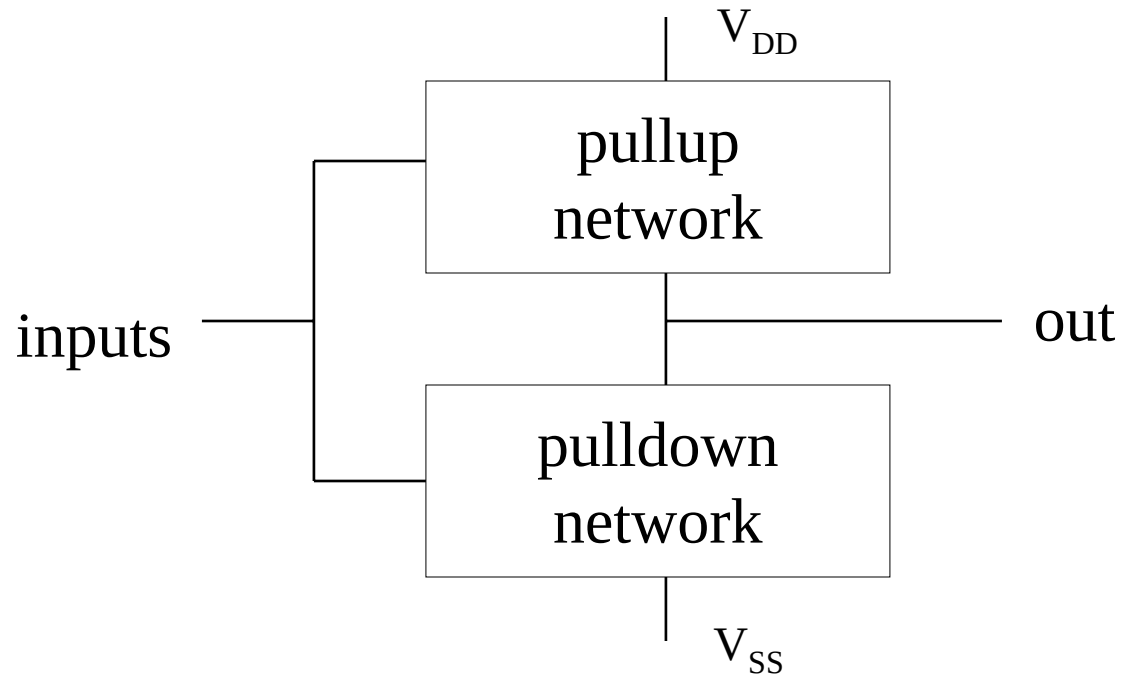
- A set of functions $f_1, f_2, \dots$ is complete iff every Boolean function can be generated by a combination of the functions.
- NAND is a complete set; NOR is a complete set; {AND, OR} is not complete.
- Transmission gates are not complete.
- If your set of logic gates is not complete, you can't design arbitrary logic.

Static complementary gates

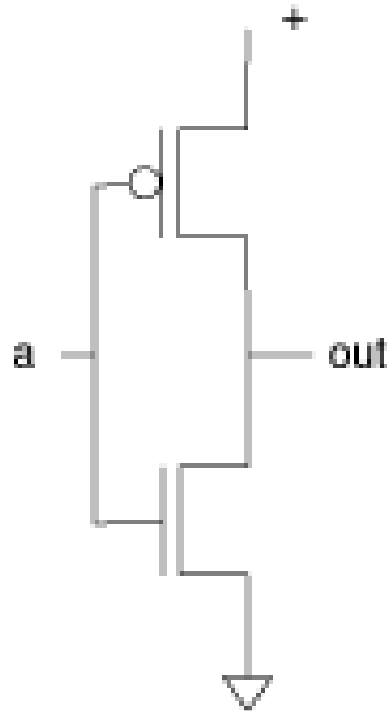
- Complementary: have complementary pullup (p-type) and pulldown (n-type) networks.
- Static: do not rely on stored charge.
- Simple, effective, reliable; hence very popular.

Static complementary gate structure

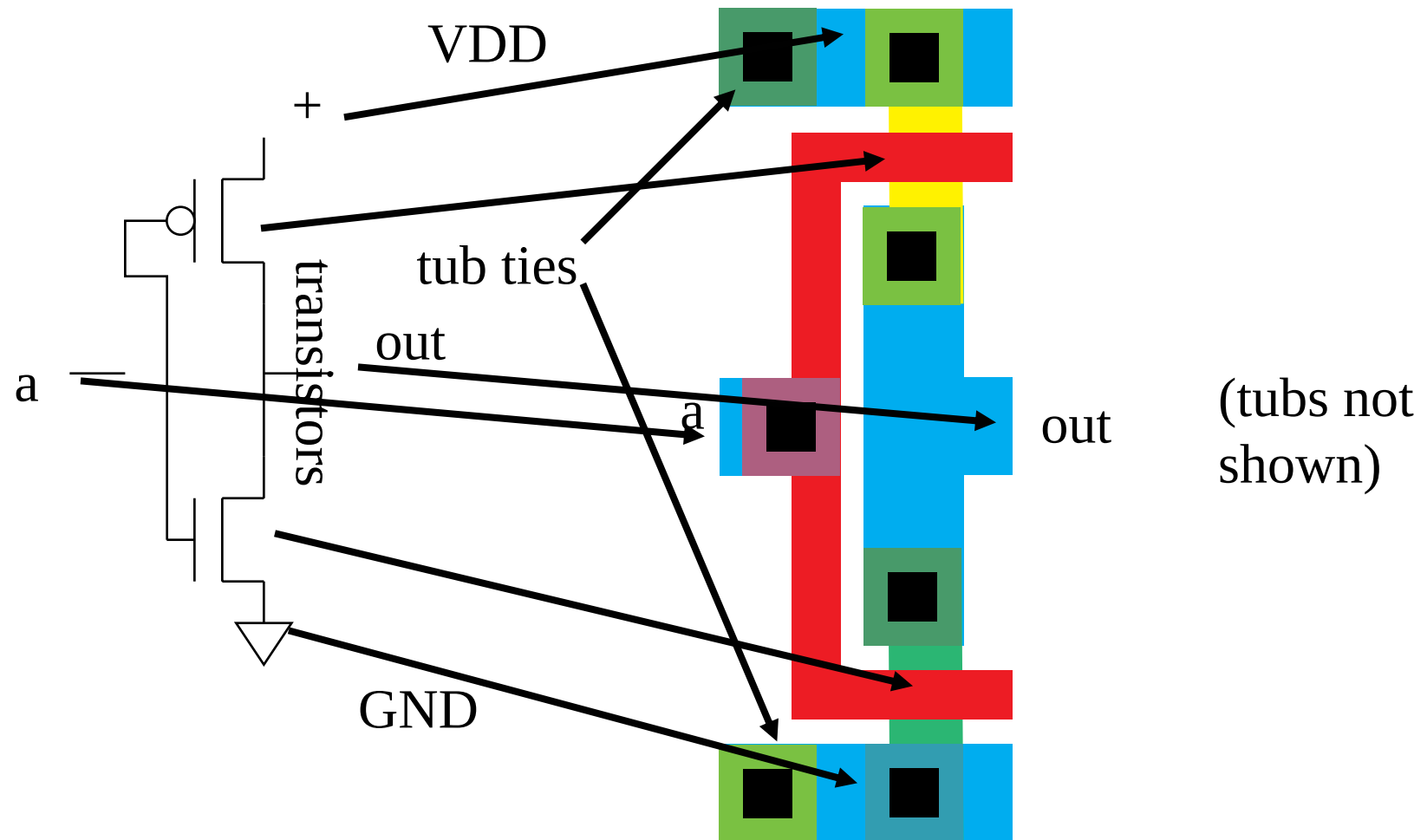
Pullup and pulldown networks:



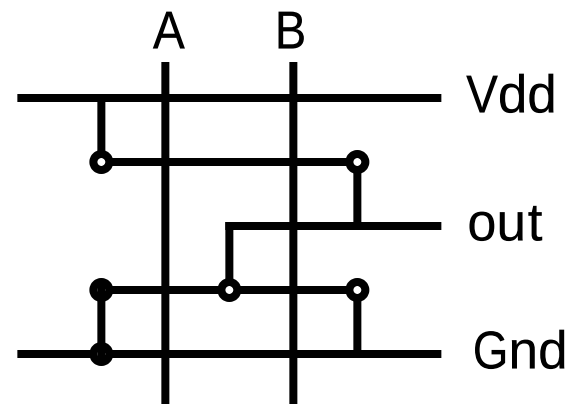
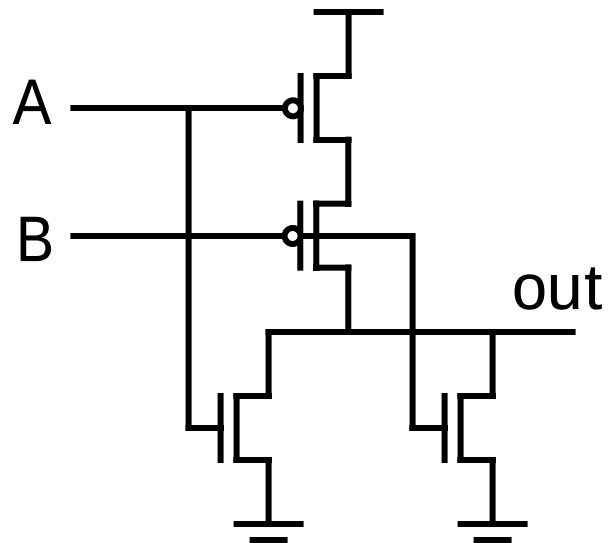
Inverter



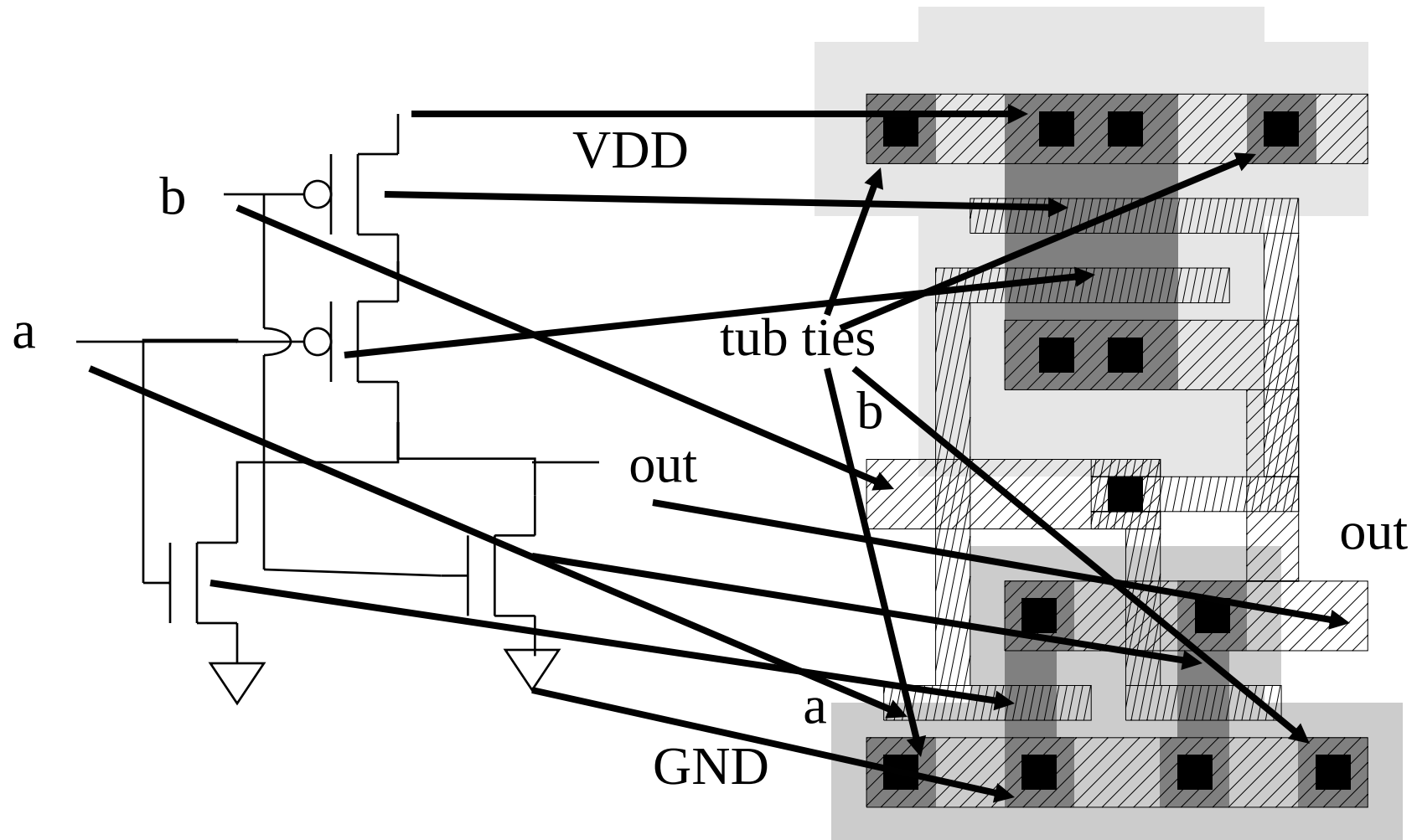
Inverter layout



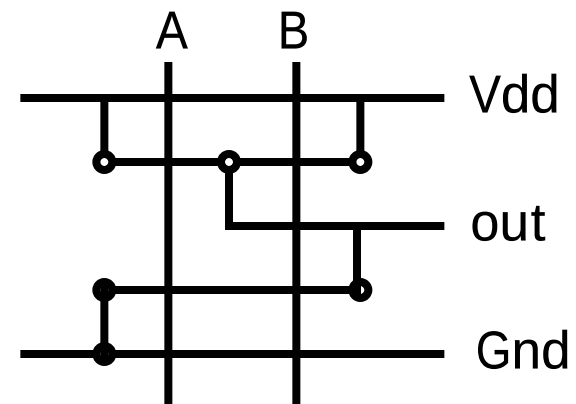
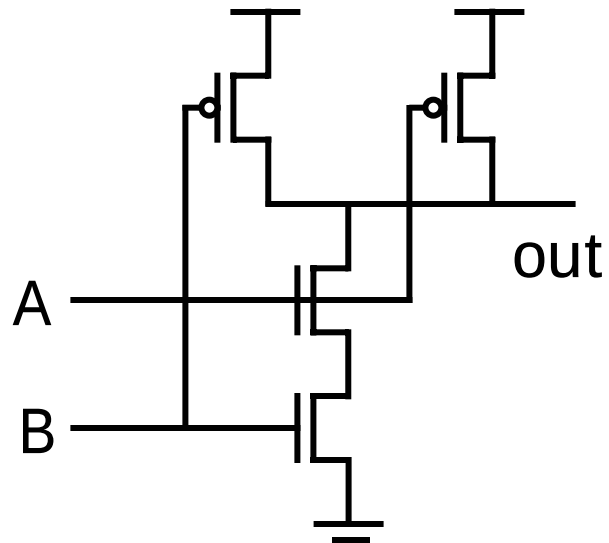
CMOS NOR



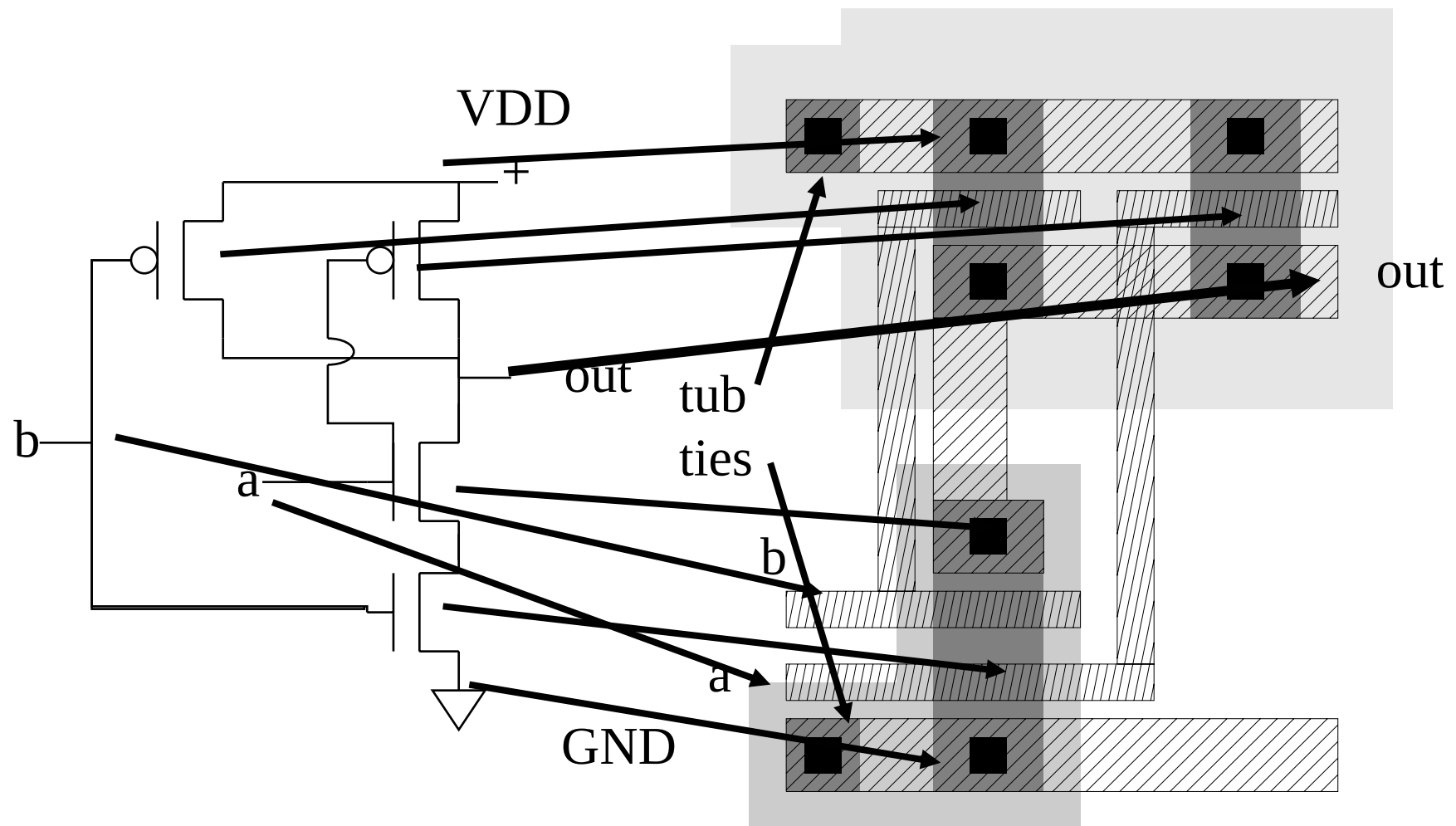
NOR layout



CMOS NAND



NAND layout



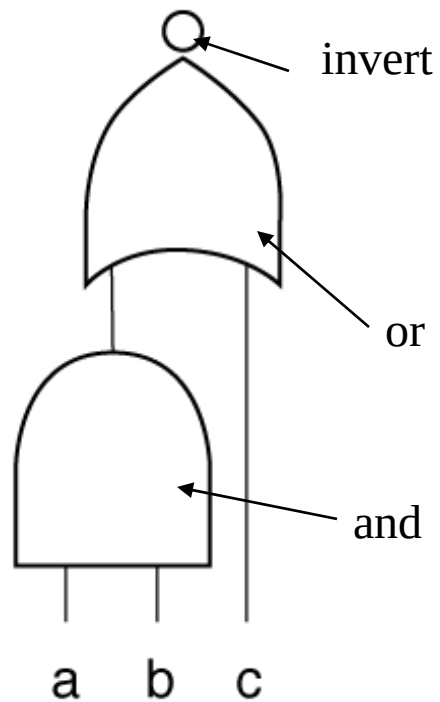
AOI/OAI gates

- AOI = and/or/invert; OAI = or/and/invert.
- Implement larger functions.
- Pullup and pulldown networks are compact: smaller area, higher speed than NAND/NOR network equivalents.
- AOI312: and 3 inputs, and 1 input (dummy), and 2 inputs; or together these terms; then invert.

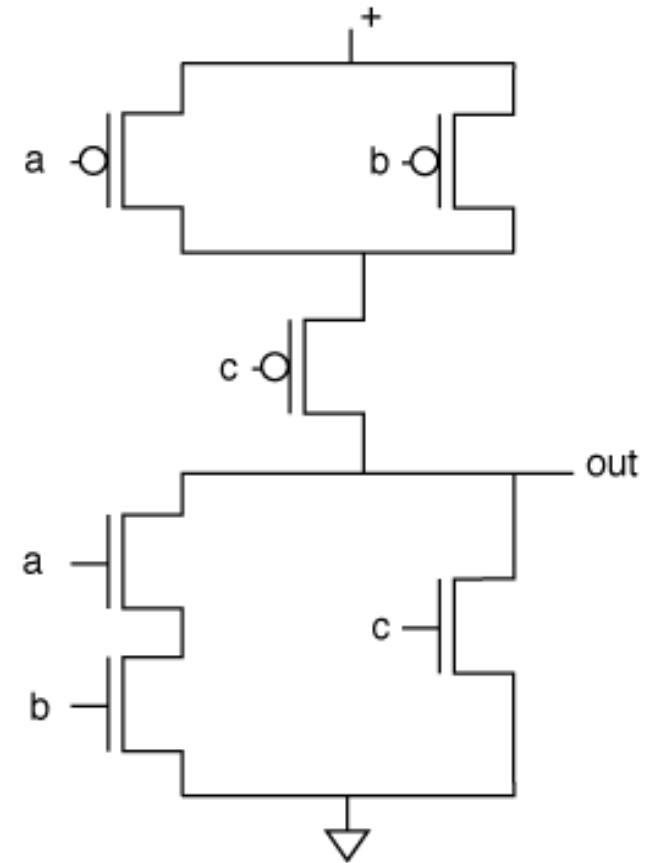
AOI example

$$\text{out} = [ab+c]':$$

symbol



circuit



Pullup/pulldown network design

- Pullup and pulldown networks are duals.
- To design one gate, first design one network, then compute dual to get other network.
- Example: design network which pulls down when output should be 0, then find dual to get pullup network.

Dual network construction

