The nMOS switch passes "0" well.

A CMOS NAND gate

The pMOS switch passes "1" well.

A CMOS INVERTER

MOS Transistors

CMOS NOR gate
- CMOS NAND gate
- CMOS Inverter

- Layout of basic devices:
  - MOS: PMOS, NMOS
  - Physical structure of MOS transistors and their schematic icons:
    - CMOS (Complementary MOS) dominates MOS and CMOS, due to CMOS's lower power dissipation, higher frequency, etc.
- The most popular VLSI technology: MOS (Metal-Oxide-Semiconductor)
Basic Definitions and Notation

Cmos compound gate, $f = (A \lor C) \land (A \lor D)$

VDD (5V, 3.3V, 2.5V)

Step 1: Connect the network to GND (0V) and the network to

Step 2: Make connections of transistors (same as Step 1)

Example: $f = F \lor G \lor H \lor I$

Step 1: Insert P to derive network (P)$ \lor F \lor G \lor H \lor I$

Step 2: Make connections of transistors

Example: $f = F \lor G \lor H \lor I$

A CMOS NOR gate

Design Rule Examples

1. Basic design rules: Wire width, wire separation, contact rule.

2. A circuit is laid out according to a set of geometric design rules.

Layout Design Rules

1. Pad layout

2. Dimension region width

3. Poly-Poly shorting spacing

4. Contact extension
Dynamic Programming

- [ ] Greedy: First-Fit Decreasing (FFD) \((\text{FFD}(p) \leq \text{OPT}(p))\)

Example: Bin packing

**Goal:** Pack all items minimizing \# of bins used \((\text{NP}-hard)\)

\(m \leq \min \{\text{set of bins}\} \quad \text{such that} \quad \text{each item fits}\)

The Bin-Packing Problem: \(I = \{(w_1, h_1), \ldots, (w_n, h_n)\}\), where

- Elements: **Tetrahedron** (in \(P\)) vs. **Inteplatable problems**
- **NP**-Hard:
  - \(p = \text{NP}(\text{Extended})\)
  - \(p \in \text{NP}(\text{Extended})\) if a polynomial-time algorithm exists

- NP-Completeness and NP-Hardness

- P (\(\text{Polynomial-Time}\) algorithms)
- \(\text{NP}\) (\(\text{Non-Polynomial-Time}\) algorithms)
- \(\text{NP}\)-Hard: those that can be solved (checked) in polynomial time
- \(\text{P} = \text{NP}\): \(\text{NP}\) is a polynomial function of \(n\)
- \(\text{P} \neq \text{NP}\): \(\text{NP}\)-hard problems and \(\text{NP}\)-complete problems

- \(\text{NP}\)-Complete: \(1, 2, 3, 4, 5, 6, 7, \ldots\)

- \(\text{NP}\)-Hardness}

- \(\text{NP}\)-Completeness And \(\text{NP}\)-Hardness
Step 1: Set $x = |I|$ to the lower bound of the # of bins.

Step 2: Otherwise, set $|I| + 1$ to $x$. Go to step 2.

Step 3: If the solution exists, the # of bins required is $|I|$. Then exit.

Step 2: Use the LP to find a feasible solution.

ILP Formulation for Bin Packing

Mathematical Programming: Use Integer Linear Programming

Solving ILP

1. Find a solution using LP, then search for the smallest feasible $|I|$.