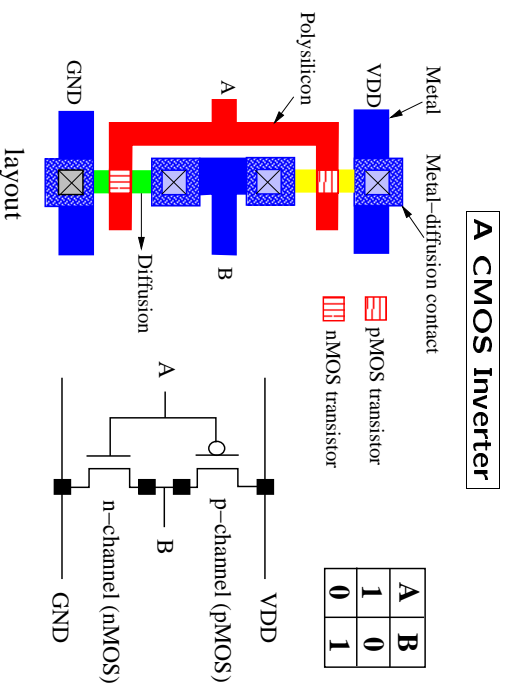


Basics of MOS Devices

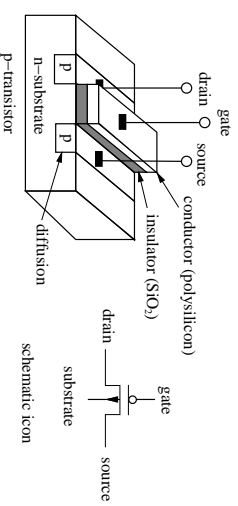
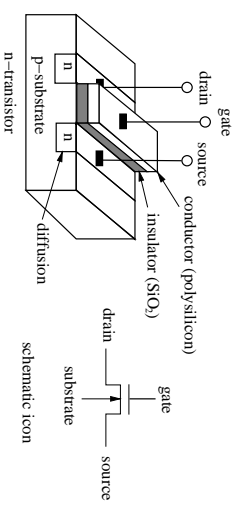
- The most popular VLSI technology: MOS (Metal-Oxide-Semiconductor).
- CMOS (Complementary MOS) dominates nMOS and pMOS, due to CMOS's lower power dissipation, high regularity, etc.
- Physical structure of MOS transistors and their schematic icons: nMOS, pMOS.
- Layout of basic devices:
 - CMOS inverter
 - CMOS NAND gate
 - CMOS NOR gate

25

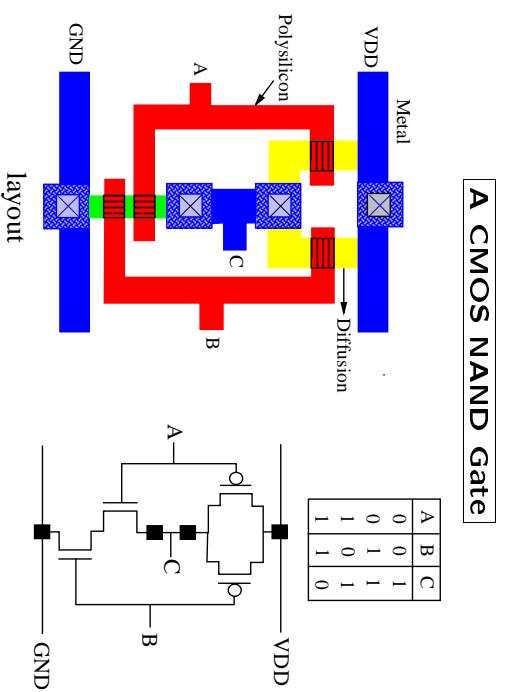


27

MOS Transistors

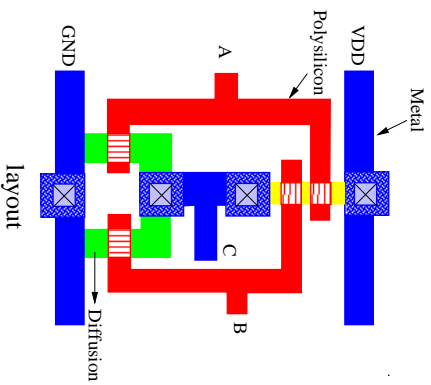


26

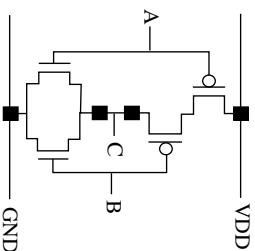


28

A CMOS NOR Gate



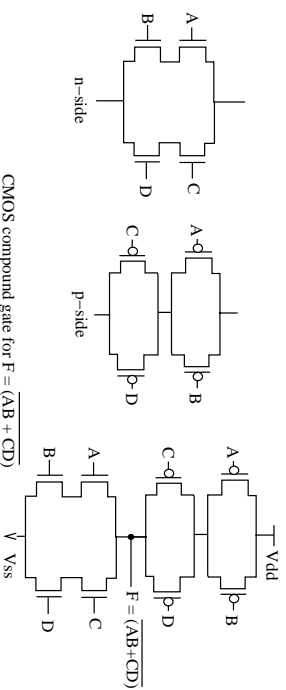
A	B	C
0	0	1
0	1	0
1	0	0
1	1	0



29

Construction of Compound Gates

- Example: $F = \overline{A \cdot B + C \cdot D}$.
- Step 1: Invert F to derive n -network ($\overline{F} = A \cdot B + C \cdot D$).
- Step 2: Make connections of transistors:
 - AND \iff Series connection; OR \iff Parallel connection.
- Step 3: Expand F to derive p -network ($F = \overline{A \cdot B + C \cdot D} = \overline{A \cdot B} \cdot \overline{C \cdot D} = (\overline{A} + \overline{B}) \cdot (\overline{C} + \overline{D})$); each input is inverted.
- Step 4: Make connections of transistors (same as Step 2).
- Step 5: Connect the n -network to GND (typically, 0V) and the p -network to VDD (5V, 3.3V, or 2.5V).



30

Layout Design Rules

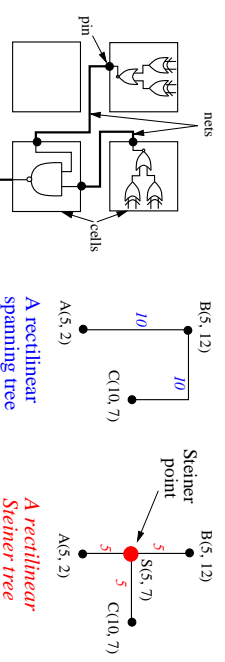
- A circuit is laid out according to a set of **geometric design rules**.
- 3 basic design rules: Wire width, Wire separation, Contact rule.
- Design rule examples

Diffusion region width	2 λ
Polysilicon region width	2 λ
Diffusion-Diffusion spacing	3 λ
Poly-Poly spacing	2 λ
Contact extension	λ
Metal	3 λ

λ : \approx device feature size (Year 1995: 0.35 μm ; Year 1997: 0.25 μm ; Year 2000: 0.18 μm)

Basic Definitions and Notation

- **Cell**: a logic block used to build larger circuits.
- **Pin**: a wire (metal or polysilicon) to which another external wire can be connected.
- **Net**: a collection of pins which must be electrically connected.
- **Netlist**: a list of all nets in a circuit.
- **Manhattan distance**: If two points (pins) are located at coordinates (x_1, y_1) and (x_2, y_2) , the Manhattan distance between them is given by $d_{12} = |x_1 - x_2| + |y_1 - y_2|$.
- **Rectilinear spanning tree**: a spanning tree that connects its pins using Manhattan paths.
- **Steiner tree**: a tree that connects its pins and additional points (**Steiner points**) are permitted to be used for the connection.



31

32

Computational Complexity

- Two concerns: **time complexity** and **space complexity**.
- **Big-Oh** notation: $f(n) = O(g(n))$ if there exist constants n_0 and c such that for all $n > n_0$, $f(n) \leq c \cdot g(n)$.
- Run-time comparison: 1000 MIPS (x886, Yr: 2000), 1 instr. /op.

Time	Big-Oh	$n = 10$	$n = 100$	$n = 10^3$	$n = 10^6$
500	$O(1)$	5×10^{-7} sec	5×10^{-7} sec	5×10^{-7} sec	5×10^{-7} sec
$3n$	$O(n)$	3×10^{-8} sec	3×10^{-7} sec	3×10^{-6} sec	0.003 sec
$n \log n$	$O(n \log n)$	3×10^{-8} sec	2×10^{-7} sec	3×10^{-6} sec	0.006 sec
n^2	$O(n^2)$	1×10^{-7} sec	1×10^{-5} sec	0.001 sec	16.7 min
n^3	$O(n^3)$	1×10^{-6} sec	0.001 sec	1 sec	3×10^6 cent.
2^n	$O(2^n)$	1×10^{-6} sec	3×10^{17} cent.	∞	∞
$n!$	$O(n!)$	0.003 sec	∞	∞	∞

- Polynomial-time complexity: $O(p(n))$, where n is the problem size and $p(n)$ is a polynomial function of n .
- **P (NP)**: Those can be **solved (checked)** in polynomial time
 - $P = NP$?
- Edmonds: **Tractable** (in **P**) vs. **Intractable** problems

Algorithmic Paradigms

- **Exhaustive search**: Search the entire solution space.
- **Branch and bound**: A search technique with pruning.
- **Greedy approach**: Pick one with a locally optimal solution at each step.
- **Dynamic programming**: Partition a problem into a collection of sub-problems, the sub-problems are solved, and then the original problem is solved by combining the solutions. (Applicable when the sub-problems are **NOT independent**).

- **Hierarchical approach**: Divide-and-conquer.

- **Mathematical programming**: A System of solving an objective function under constraints.

- **Simulated annealing**: An adaptive, iterative, non-deterministic algorithm.

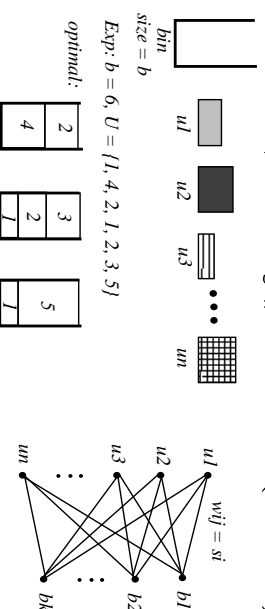
- **Genetic algorithm**: A population of solutions is stored and allowed to evolve through successive generations via mutation, crossover, etc.

NP-Completeness and NP-Hardness

- **Decision problems**: those having a yes/no solution.
 - Given a graph $G = (V, E)$, is there a spanning tree with a cost at most k ?
- **Optimization problems**: those finding a legal configuration such that its cost is minimum (or maximum).
 - Given a graph $G = (V, E)$, find a minimum spanning tree of G .
- **Optimal solutions/costs**, **Optimal (exact)** algorithms (Attention: optimal \neq exact in the theoretic computer science community).
- Π is **NP-complete if: (worst-case analyses for decision problems)**
 1. **NP**: $\Pi \in NP$
 2. **NP-Hard**: $\Gamma' \leq_p \Pi$ for every $\Gamma' \in NP$.
- Coping with NP-hard (optimization) problems:
 - **Exhaustive search**: (feasible only when the problem size is small)
 - **Restriction**: Work on some subset of the original problem.
 - **Pseudo-polynomial time algorithms**:
 - **Approximation algorithms**:
 - **Heuristics**:
 - **Randomization**:

Example: Bin Packing

- **The Bin-Packing Problem Π** : Items $U = \{u_1, u_2, \dots, u_n\}$, where u_i is of an integer size s_i ; set B of bins, each with capacity b .
- **Goal**: Pack all items, minimizing # of bins used. (**NP-hard!**)



- Greedy: First-Fit Decreasing (FFD) ($FFD(\Pi) \leq \frac{11}{9} OPT(\Pi) + 4$)
- Dynamic Programming? Hierarchical Approach? Genetic Algorithm? ...

ILP Formulation for Bin Packing

- 0-1 variable: $x_{ij} = 1$ if item u_i is placed in bin b_j ; 0 otherwise.

$$\max \sum_{(i,j) \in E} w_{ij} x_{ij}$$

subject to

$$\sum_{V \in I'} w_{ij} x_{ij} \leq b_j, \forall j \in B \quad /* \text{capacity constraint} */ \quad (1)$$

$$\sum_{V \in B} x_{ij} = 1, \forall i \in U \quad /* \text{assignment constraint} */ \quad (2)$$

$$\sum_{ij} x_{ij} = n \quad /* \text{completeness constraint} */ \quad (3)$$

$$x_{ij} \in \{0, 1\} \quad /* 0, 1 \text{ constraint} */ \quad (4)$$

- **Step 1:** Set $|B|$ to the lower bound of the # of bins.
- **Step 2:** Use the ILP to find a feasible solution.
- **Step 3:** If the solution exists, the # of bins required is $|B|$. Then exit.
- **Step 4:** Otherwise, set $|B| \leftarrow |B| + 1$. Goto Step 2.

VLSI Design Automation Conferences/Journals

- Important Conferences:
 - IEEE/ACM Int'l Conference on Computer-Aided Design (ICCAD)
 - ACM/IEEE Design Automation Conference (DAC)
 - ACM Int'l Symposium on Physical Design (ISPD)
 - ACM Int'l Symposium on Field Programmable Gate Arrays (FPGA)
 - IEEE Int'l Conference on Computer Design (ICCD)
 - IEEE Int'l Symposium on Circuits and Systems (ISCAS)
 - IEEE Custom Integrated Circuits Conference (CICC)
 - Others: VLSI Design/CAD Symposium/Taiwan; Design, Automation and Test in Europe (DATE); ASP-DAC
- Important Journals:
 - IEEE Transactions on Computer-Aided Design (TCAD)
 - ACM Transactions on Design Automation of Electronic Systems (TO-DAES)
 - IEEE Transactions on VLSI Systems (TVLSI)
 - IEEE Transactions on Computers (TC)
 - IEEE Transactions on Circuits and Systems (TCS)

- INTEGRATION: The VLSI Journal
- Algorithmica