

CS 6135
VLSI Physical Design Automation
VLSI實體設計自動化
Fall 2025

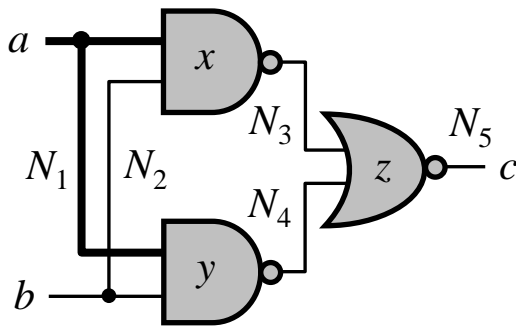
Course Information

- Time: T3T4R3
- Room: Delta 108
- Instructor: Ting-Chi Wang (王廷基), C. L. Liu Building 643 ext. 42963, tcwang@cs.nthu.edu.tw
- TAs: 呂宸漢 (love0416much@gmail.com) & 曾怡茹 (annie.tseng1113@gmail.com), C. L. Liu Building 230, ext. 33570
- Website: eeclass (<https://eeclass.nthu.edu.tw>)
- Material: technical papers selected from major EDA (Electronic Design Automation) conference proceedings and journals
- Grading policy
 - Homework assignments: 75%
 - Final exam: 25% (9:00 – 12:00, 12/16/2025)

Physical Design

- Physical design is the process of converting a circuit netlist into a geometric description (i.e., determining where to put components and how to connect them).
- The description is used to manufacture a chip.
- Physical design is very complicated
 - Millions of components
 - Multiple objectives: area, performance, and power, etc.
 - Multiple constraints: components may not be too close, and wires cannot cross, etc.
- Physical design needs help from Electronic Design Automation (EDA) tools.

Netlist



(*a*: *N*₁)
(*b*: *N*₂)
(*c*: *N*₅)
(*x*: *IN*₁ *N*₁, *IN*₂ *N*₂, *OUT* *N*₃)
(*y*: *IN*₁ *N*₁, *IN*₂ *N*₂, *OUT* *N*₄)
(*z*: *IN*₁ *N*₃, *IN*₂ *N*₄, *OUT* *N*₅)

Pin-Oriented Netlist

(*N*₁: *a*, *x*.*IN*₁, *y*.*IN*₁)
(*N*₂: *b*, *x*.*IN*₂, *y*.*IN*₂)
(*N*₃: *x*.*OUT*, *z*.*IN*₁)
(*N*₄: *y*.*OUT*, *z*.*IN*₂)
(*N*₅: *z*.*OUT*, *c*)

Net-Oriented Netlist

Course Objectives & Topics

- Course objectives
 - Understanding the problems arising in the physical design of VLSI circuits.
 - Understanding various algorithms for automating the physical design process.
- Course topics
 - Introduction
 - Partitioning
 - Floorplanning
 - Placement
 - Routing

Target Audience & Required Background

- Target audience: students who want to be
 - EDA engineers or researchers: development and implementation of EDA tools
 - VLSI designers: designing VLSI chips using EDA tools
- Required background
 - Digital logic design
 - Data structures
 - Algorithms
 - Programming in C or C++

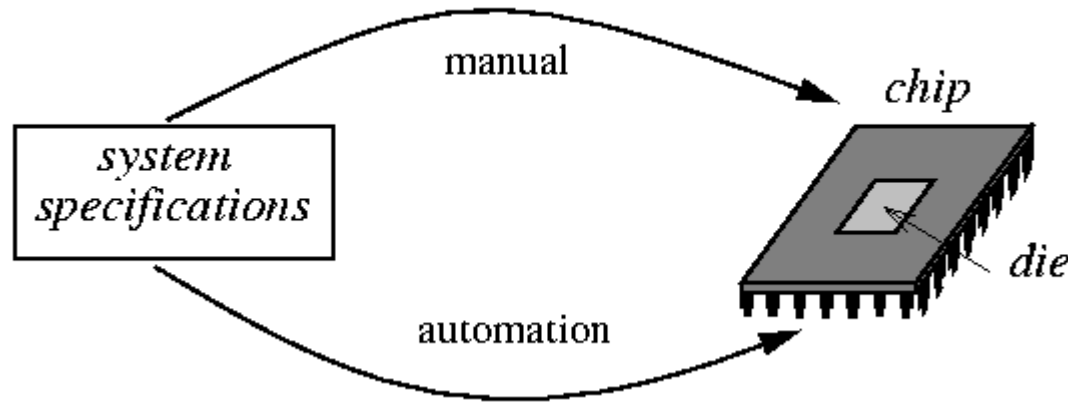
Major Conferences and Journals

- Conferences
 - Design Automation Conference (DAC)
 - International Conference on Computer-Aided Design (ICCAD)
 - Asia and South Pacific Design Automation Conference (ASP- DAC)
 - Design, Automation and Test in Europe (DATE)
 - International Symposium on Physical Design (ISPD)
- Journals
 - IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD)
 - ACM Transactions on Design Automation of Electronic Systems (TODAES)
 - INTEGRATION: The VLSI Journal

Related Books

- A. B. Kahng, J. Lienig, I. L. Markov, J. Hu, *VLSI Physical Design: From Graph Partitioning to Timing Closure*, Springer, 2011.
- *Handbook of Algorithms for Physical Design Automation*, edited by C. J. Alpert, D. P. Mehta, S. Sapatnekar, CRC Press, 2009.
- *Electronic Design Automation: Synthesis, Verification, and Test*, edited by L.-T. Wang, Y.-W. Chang, K.-T. Cheng, Morgan Kaufmann, 2009.
- S. K. Lim, *Practical Problems in VLSI Physical Design Automation*, Springer, 2008
- N. Sherwani, *Algorithms for VLSI Physical Design Automation*, 3rd Edition, Kluwer Academic Publishers, 1999.
- M. Sarrafzadeh and C. K. Wong, *An Introduction to VLSI Physical Design*, McGraw-Hill, 1996.
- S. M. Sait and H. Youssef, *VLSI Physical Design Automation: Theory and Practice*, McGraw-Hill, 1995.

VLSI Design Considerations

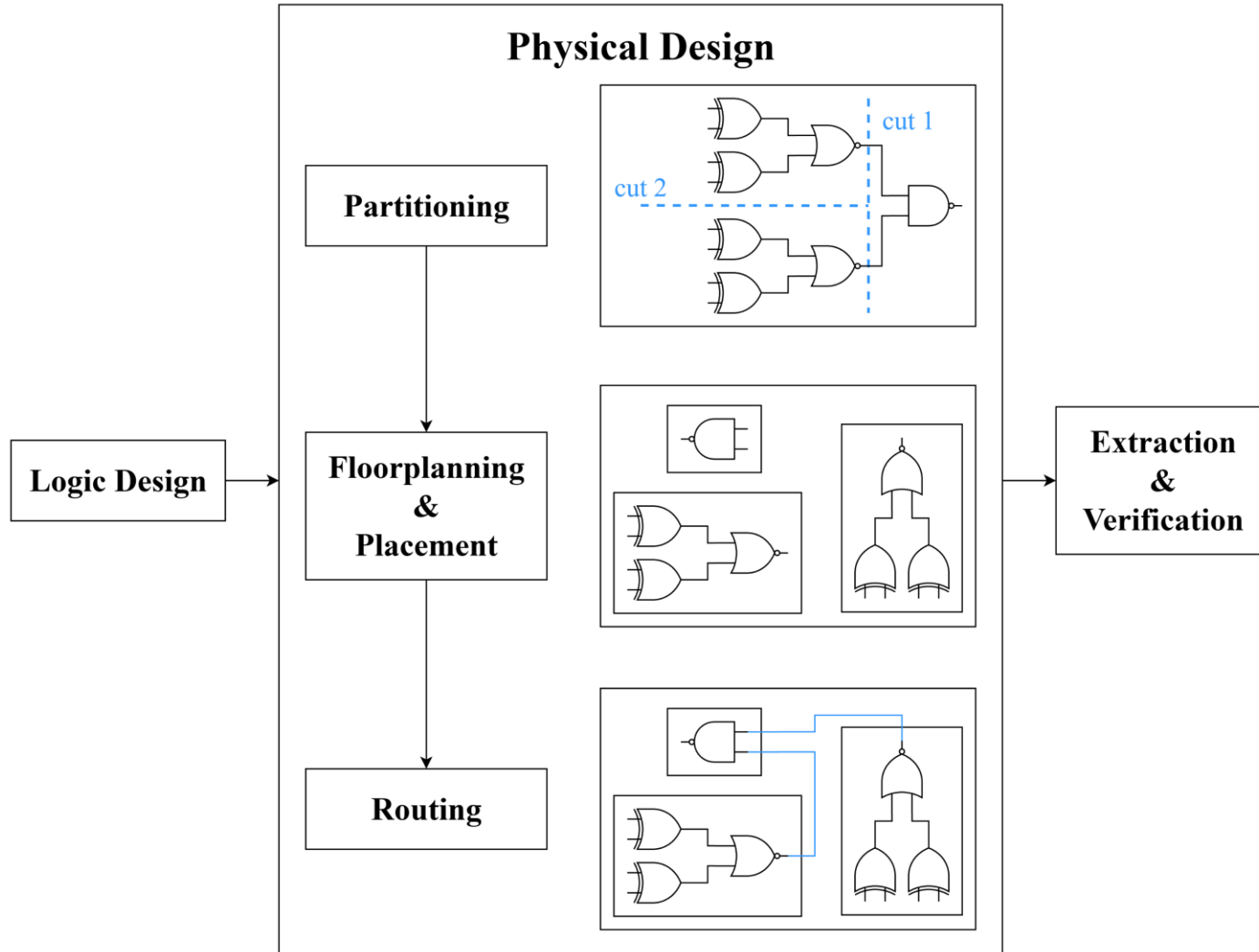


- Several conflicting considerations
 - Design complexity: large number of devices/transistors
 - Performance: optimization requirements for high performance
 - Time-to-market: about a 15% gain for early birds
 - Cost: die area, packaging, testing, etc
 - Others: power consumption, noise, reliability, etc

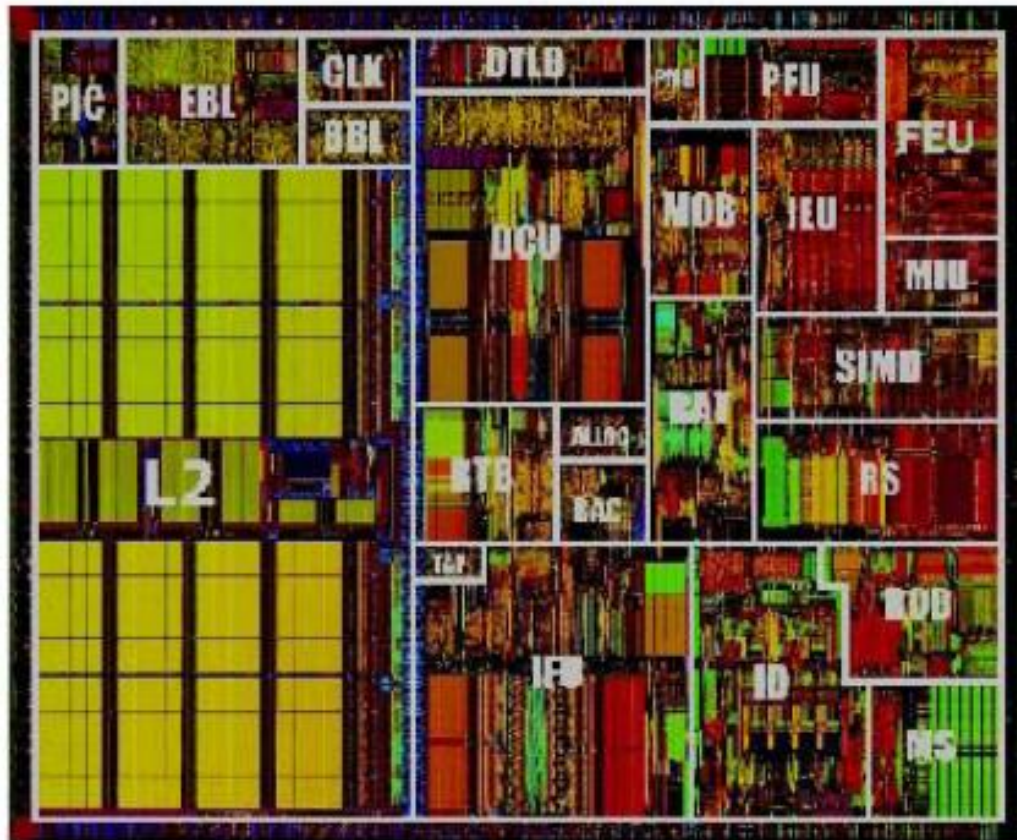
Technology Advancement

- In 1965, Gordon Moore stated that the number of transistors on an IC would double every year. Ten years later, he revised his statement, asserting that they double every 18 months. Since then, this “rule” has been famously known as Moore’s Law.
- Logic capacity doubles per IC at regular intervals.
- Designs are too complicated to be handled manually
 - Solutions
 - EDA
 - Hierarchical design
 - Design reuse
- International Technology Roadmap for Semiconductors ([http: www.itrs2.net/](http://www.itrs2.net/))
- International Roadmap for Devices and Systems (<http://irds.ieee.org/>)

Physical Design Flow

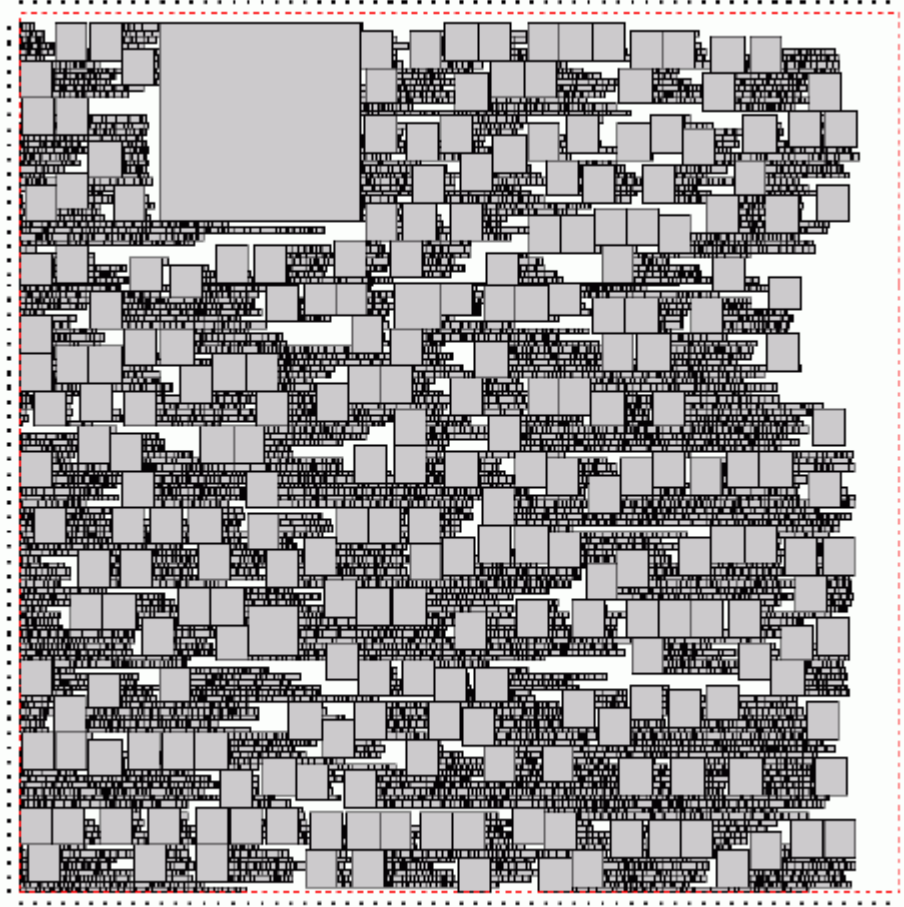


Floorplan Example



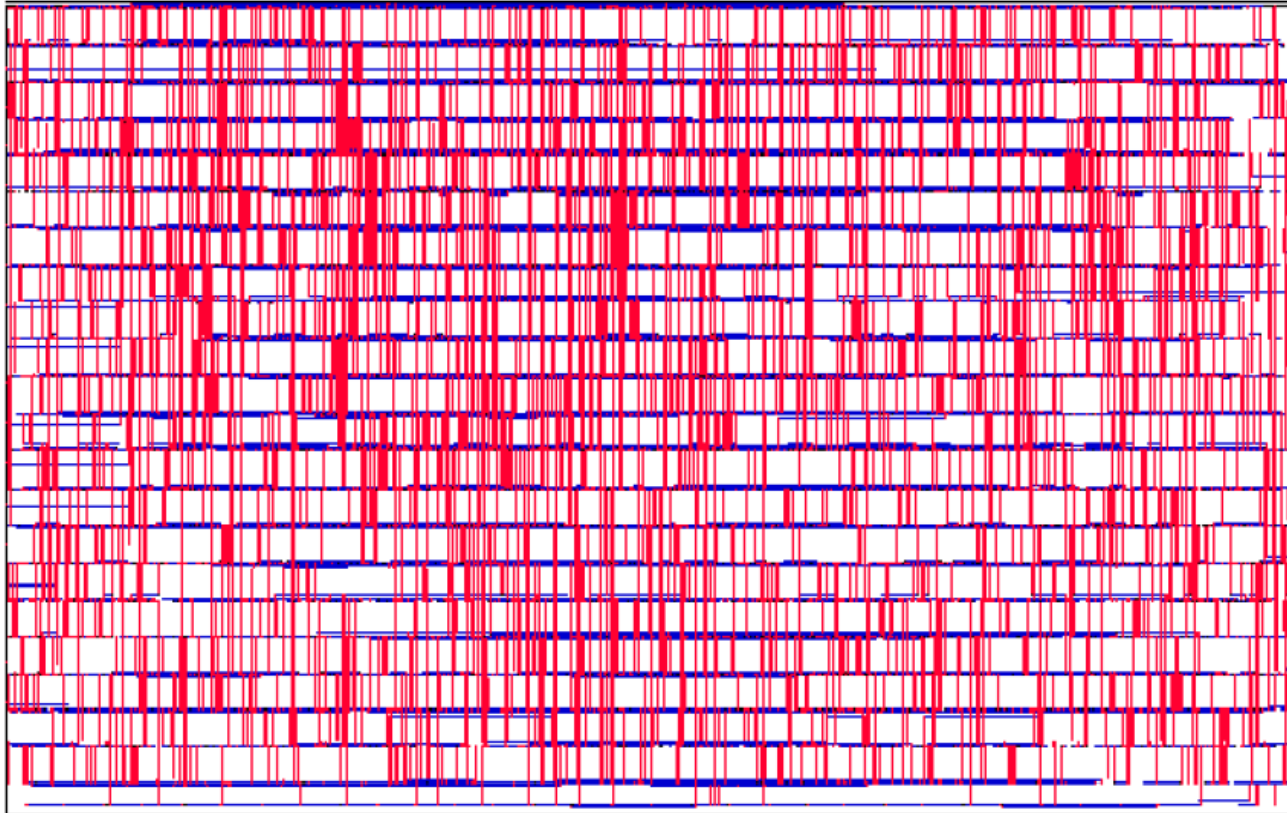
Intel
Pentium 4

Placement Example



12,752 cells and 247 macros

Routing Example

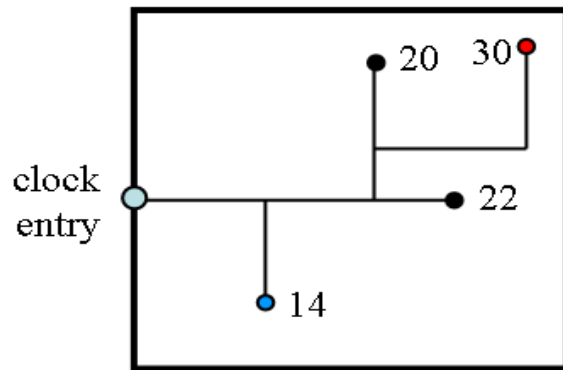


A two-layer routing example with 8,109 nets.

All horizontal wires are routed on one layer, and so are vertical ones.

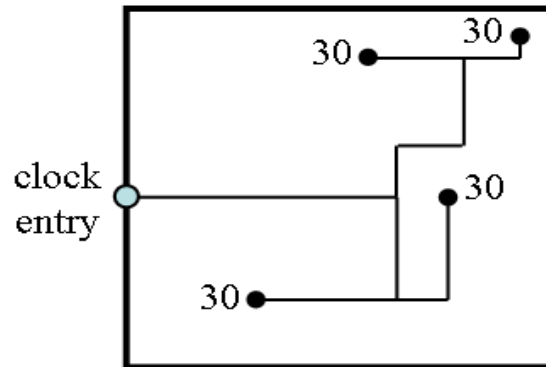
Clock Tree Synthesis (CTS)

- Clock skew: the maximum difference in the arrival times of the clock signals at the circuit components
 - should be minimized to avoid the idleness of the component with an earlier clock signal arrival.



clock skew = $30 - 14 = 16$

(a)



clock skew = 0

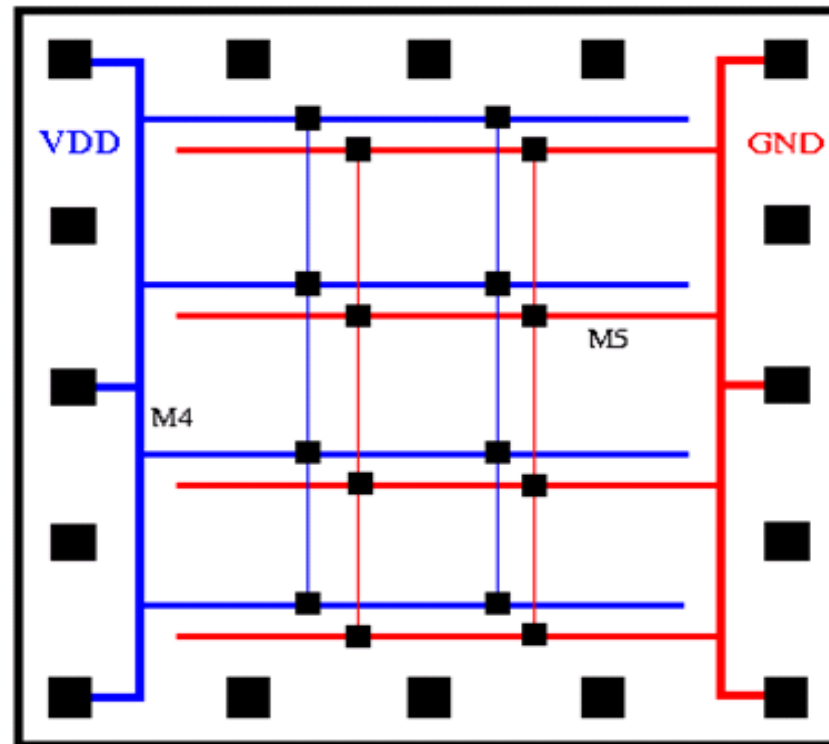
(b)

(a) a skew of 16 units and the maximum delay of 30 units.

(b) zero skew and 30-unit delay.

Power/Ground Delivery Networks

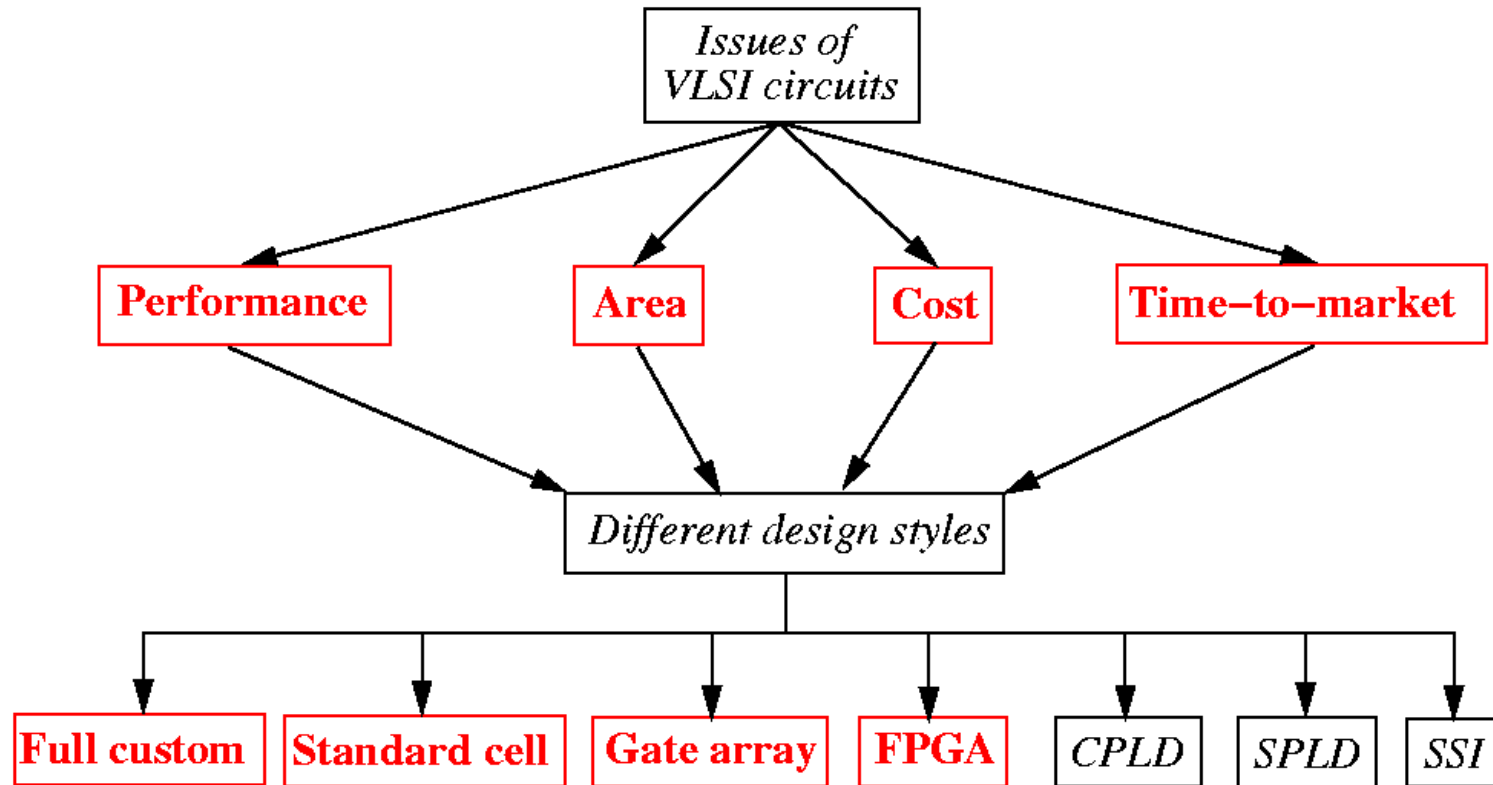
- Power/ground networks are laid out on metal layers.



Design Styles

- Restricting design styles to reduce complexity.
- Choosing design style according to design time, performance, size and cost, etc.

Design Styles (Cont'd)

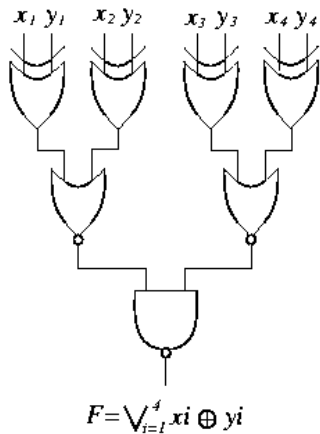


Performance, Area efficiency, Cost, Flexibility

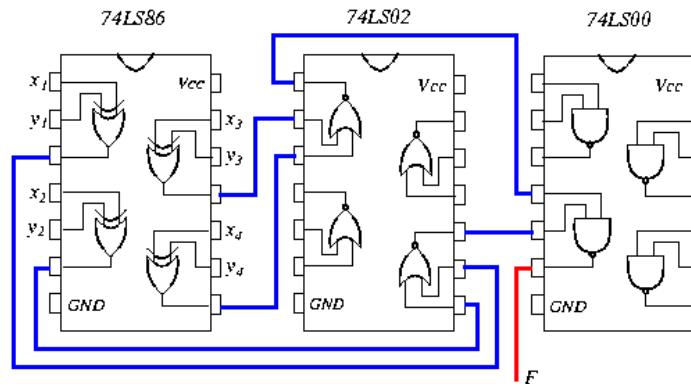
Design Styles (Cont'd)

- Full Custom
 - Fewest constraints
- Standard Cell
 - A library of cells designed in (multiples of) a fixed height
 - A design consisting of rows of cells
- Gate Array
 - A design consisting of an array of identical pre-fabricated cells with transistors but no connections
 - Requiring intra-cell routing (creating a cell by connecting certain transistors) and inter-cell routing (connecting cells to form nets from the netlist)
 - Fabricating routing layers after chip-specific requirements are known
- Field Programmable Gate Array (FPGA)
 - Pre-fabricated cells and interconnects
 - Programmable cells and interconnects

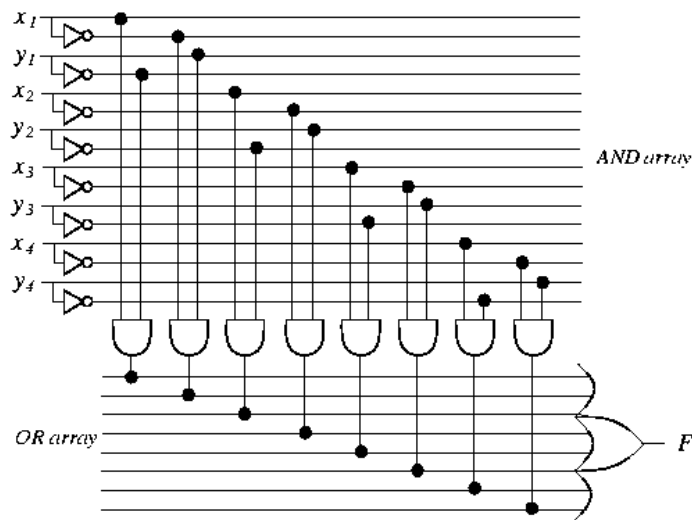
SSI & SPLD Designs



(a) 4-bit comparator.



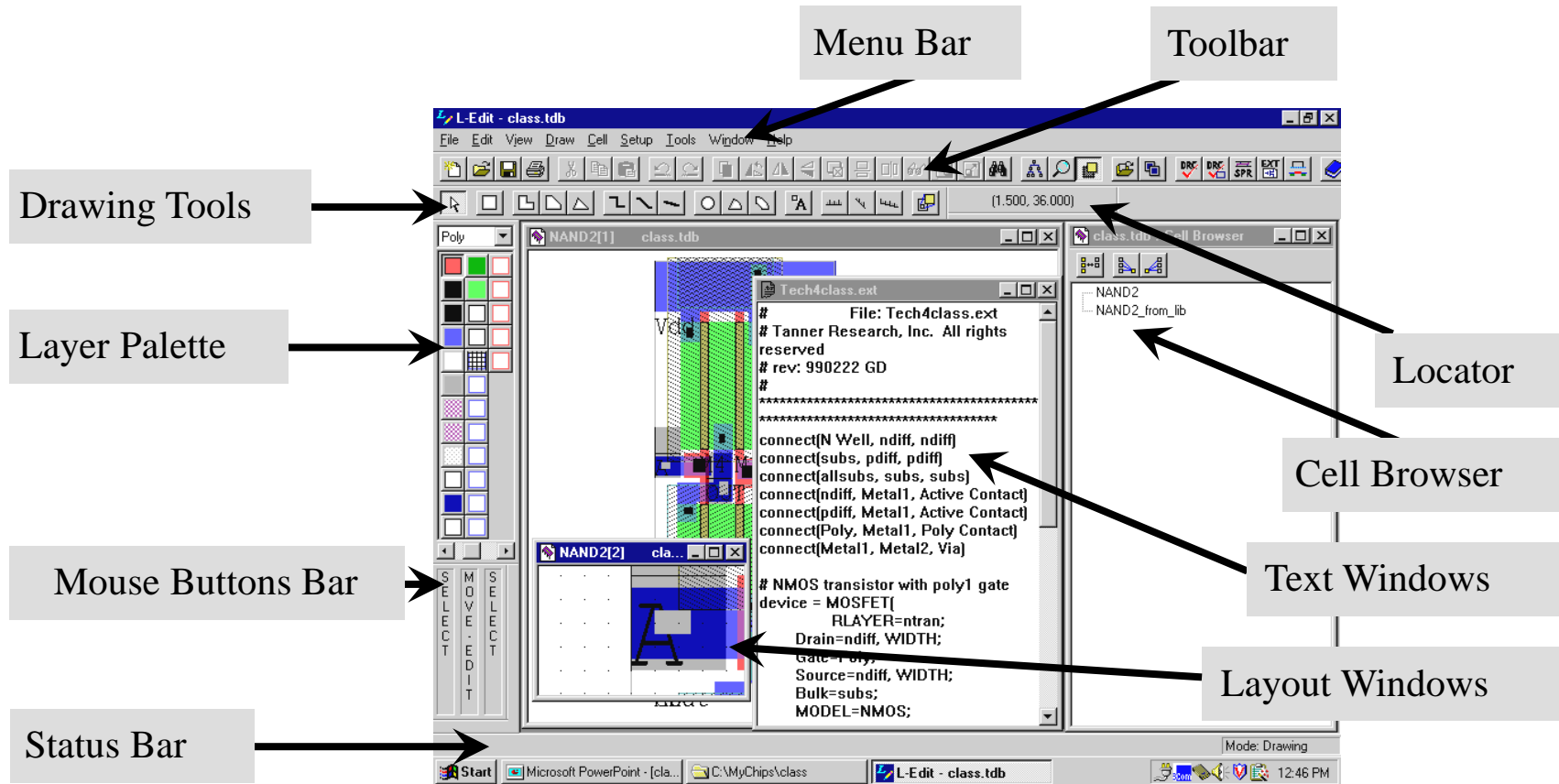
(b) SSI implementation.



(c) SPLD (PLA) implementation.

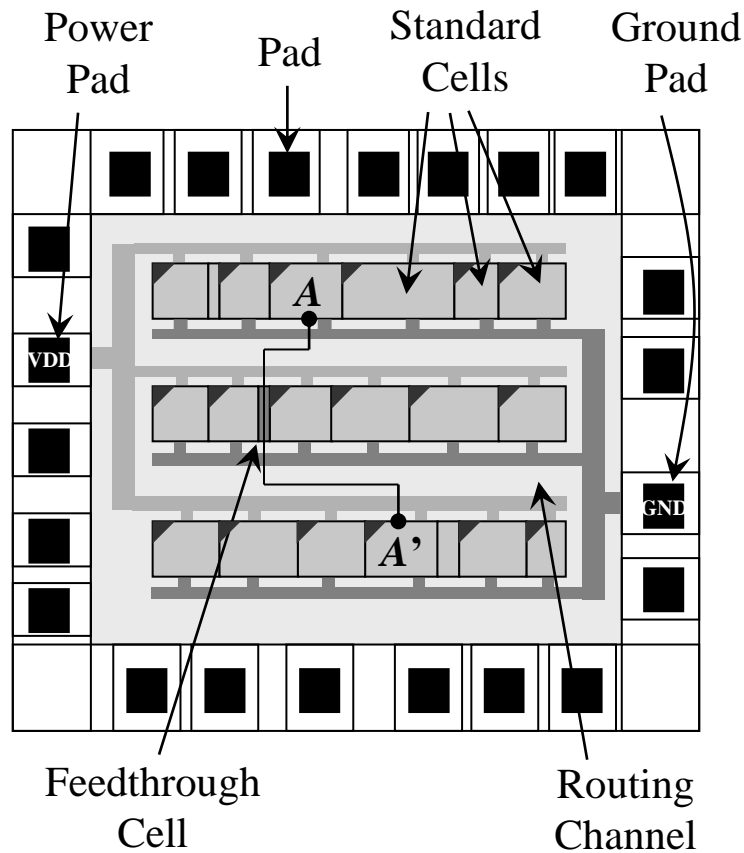
Full Custom Design

- Layout editor is an essential tool



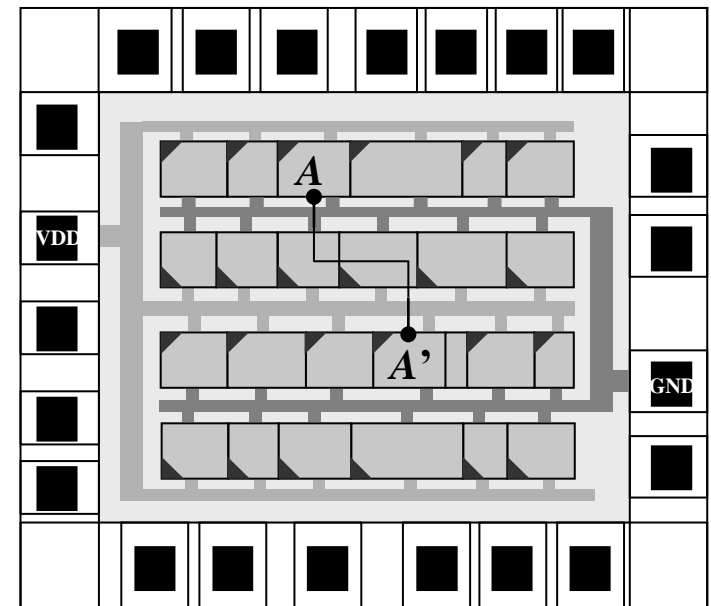
Standard Cell Design

Using feedthrough cells



Unit 1

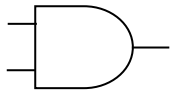
Using over-the-cell (OTC) routing



22

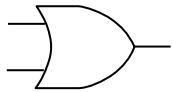
Common Digital Cells

AND



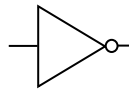
IN1	IN2	OUT
0	0	0
1	0	0
0	1	0
1	1	1

OR



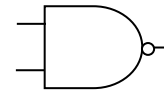
IN1	IN2	OUT
0	0	0
1	0	1
0	1	1
1	1	1

INV



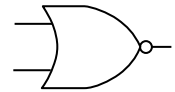
IN	OUT
0	1
1	0

NAND

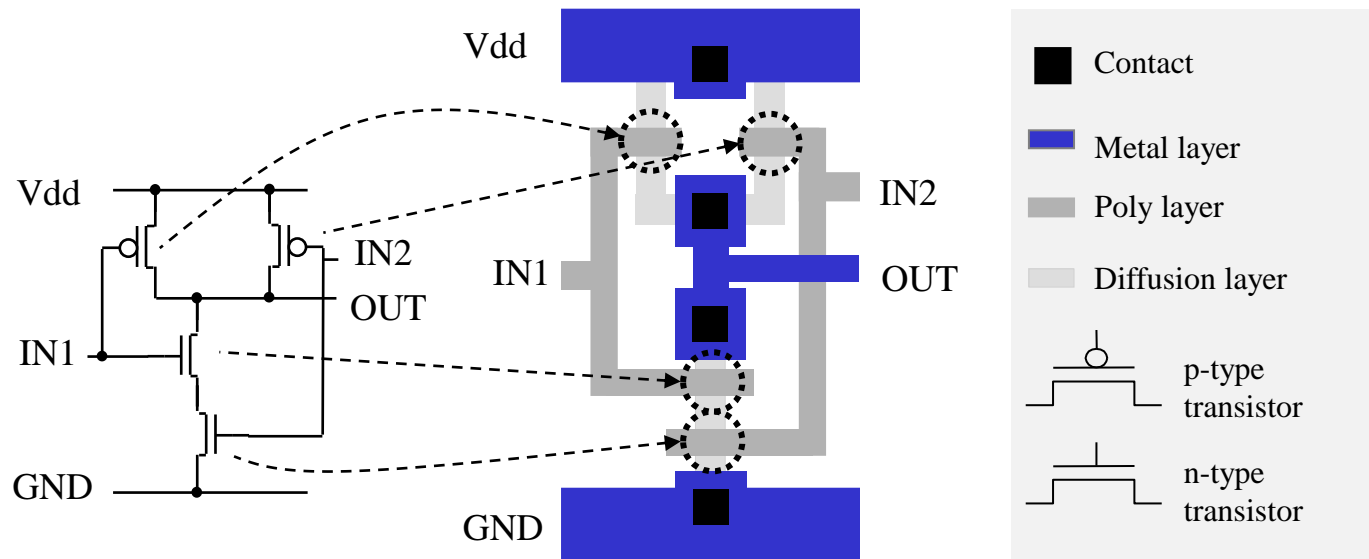
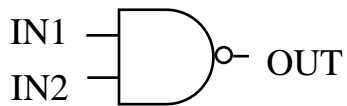


IN1	IN2	OUT
0	0	1
1	0	1
0	1	1
1	1	0

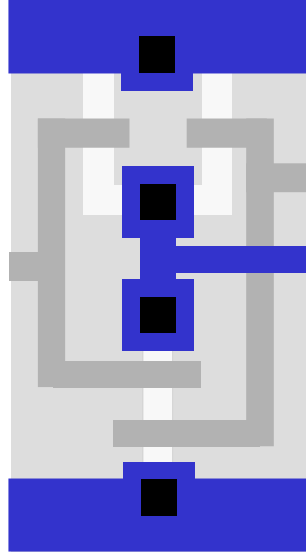
NOR



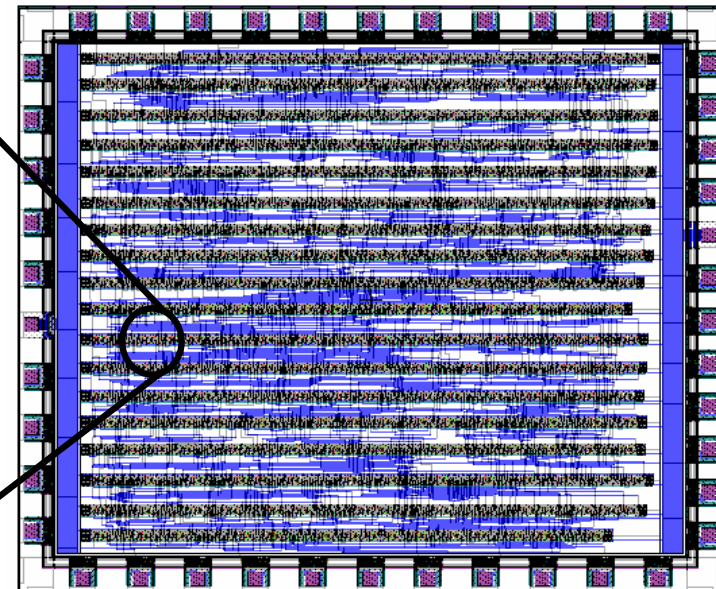
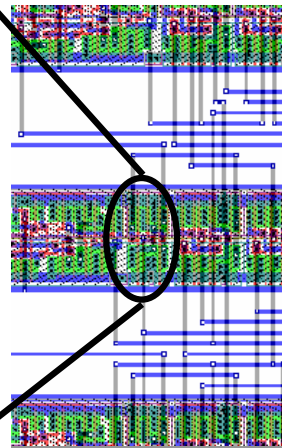
IN1	IN2	OUT
0	0	1
1	0	0
0	1	0
1	1	0



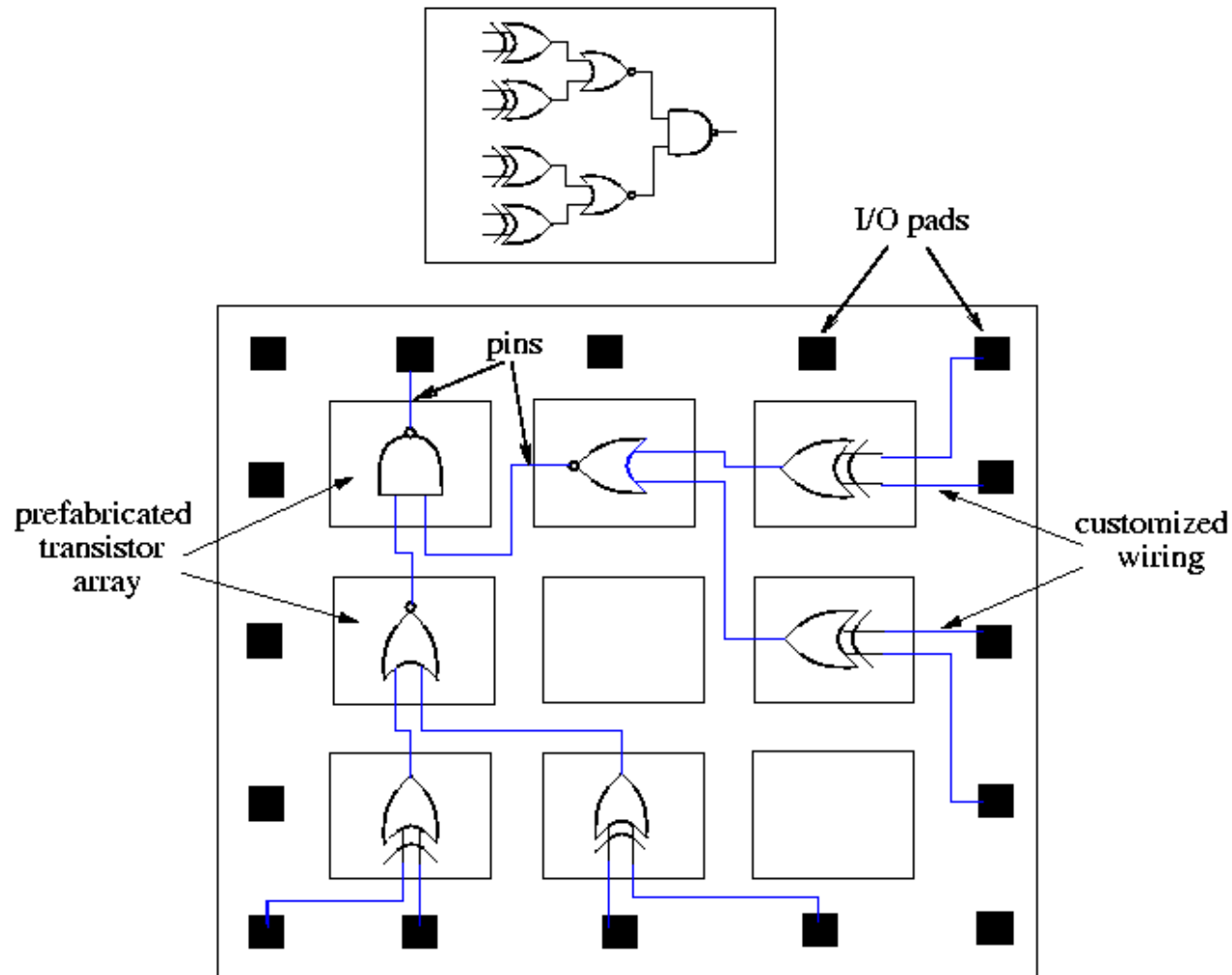
Power (Vdd)-Rail



Ground (GND)-Rail

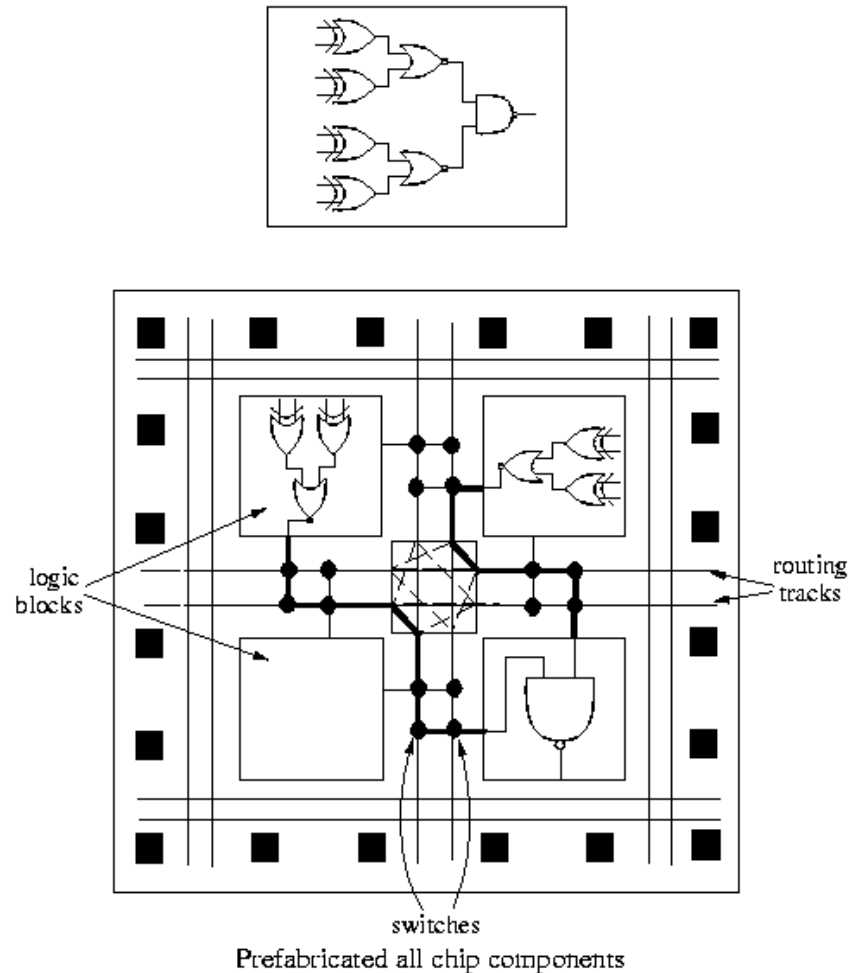


Gate Array Design



FPGA Design

- Illustrated by a symmetric array-based FPGA:



Comparisons of Design Styles

	Full custom	Standard cell	Gate array	FPGA	SPLD
Cell size	variable	fixed height*	fixed	fixed	fixed
Cell type	variable	variable	fixed	programmable	programmable
Cell placement	variable	in row	fixed	fixed	fixed
Interconnections	variable	variable	variable	programmable	programmable

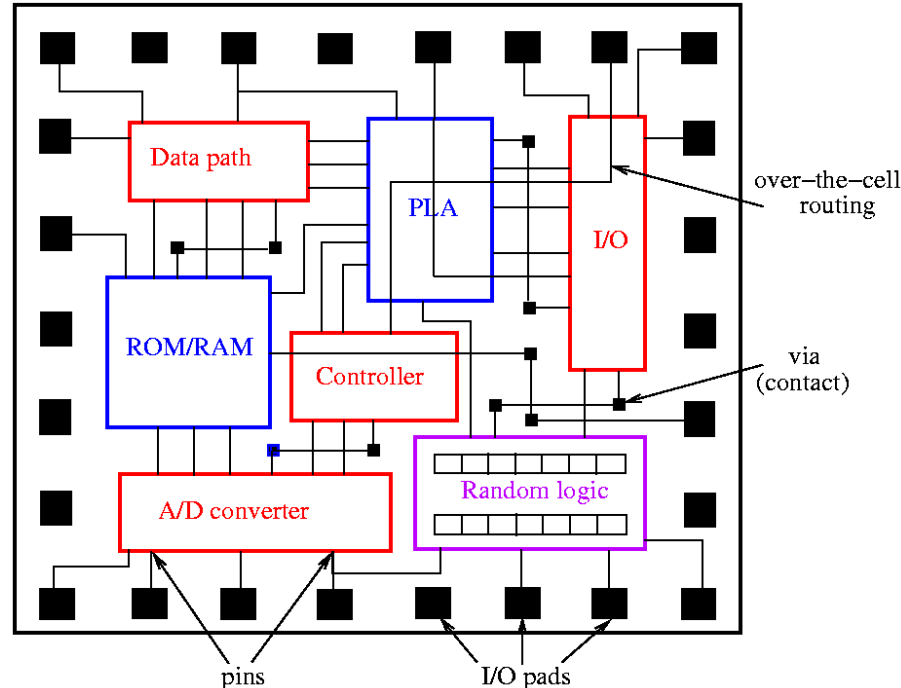
* Uneven height cells are also used.

	Full custom	Standard cell	Gate array	FPGA	SPLD
Fabrication time	— — —	— —	+	+++	++
Packing density	+++	++	+	— —	— — —
Unit cost in large quantity	+++	++	+	— —	—
Unit cost in small quantity	— — —	— —	+	+++	++
Easy design and simulation	— — —	— —	—	++	+
Easy design change	— — —	— —	—	++	++
Accuracy of timing simulation	—	—	—	+	++
Chip speed	+++	++	+	—	— —

+ desirable; — not desirable

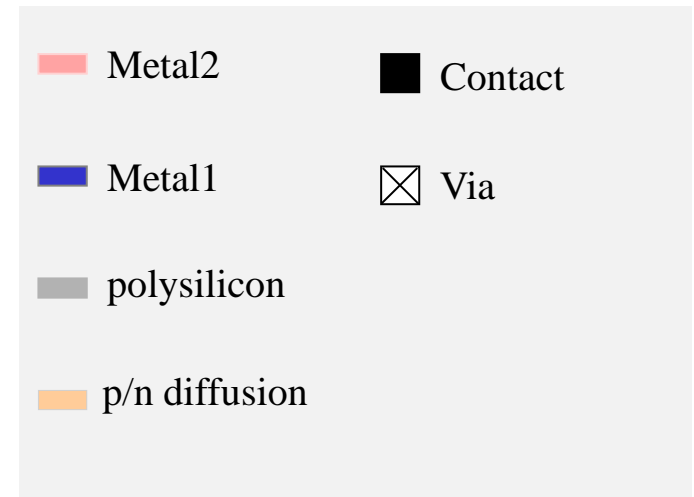
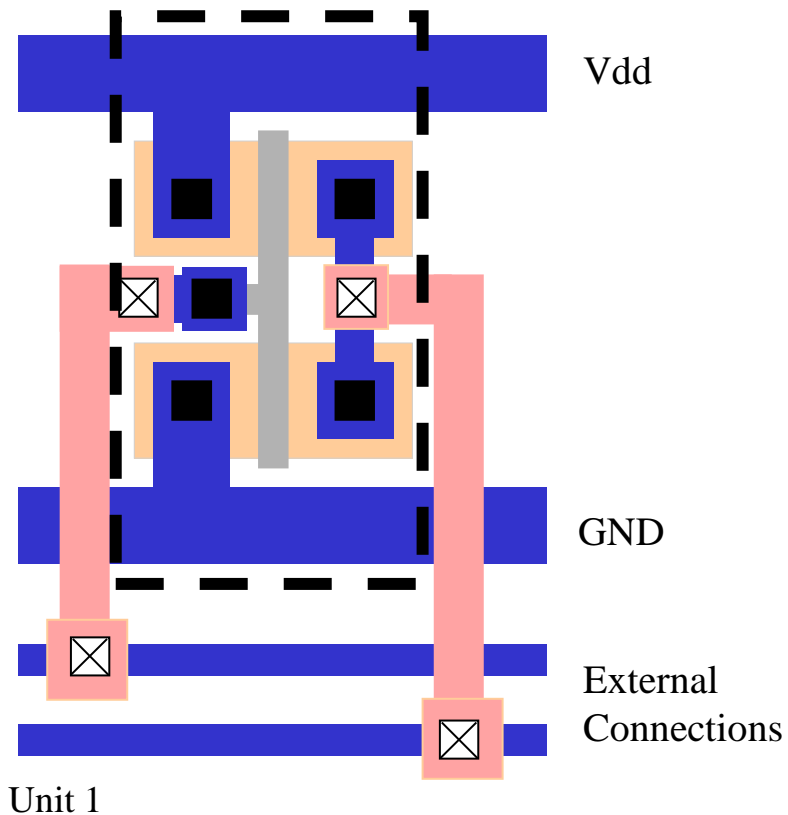
Design with Macro Cells

- Macro cells
 - Are typically large pieces of logic that perform a reusable functionality
 - Range from simple (a couple of standard cells) to highly complex (the scale of an embedded processor or memory block)
 - Vary greatly with respect to shapes and sizes
 - Can be placed anywhere in most cases



Layout Layers

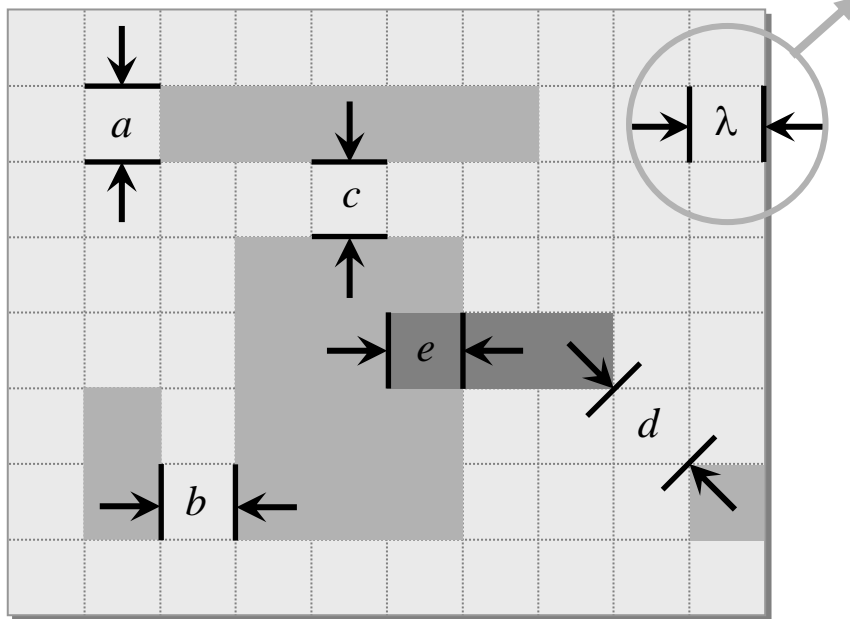
- An inverter cell with external connections



Design Rules

- Size rules (such as minimum width)
 - The dimensions of any component (shape), e.g., length of a boundary edge or area of the shape, cannot be smaller than given minimum values.
 - These values vary across different metal layers.
- Separation rules (such as minimum separation)
 - Two shapes, either on the same layer or on adjacent layers, must be a minimum (rectilinear or Euclidean diagonal) distance apart.
- Overlap rules (such as minimum overlap)
 - Two connected shapes on adjacent layers must have a certain amount of overlap due to inaccuracy of mask alignment to previously-made patterns on the wafer.

Design Rules (Cont'd)



λ : smallest meaningful technology-dependent unit of length

Minimum Width: a

Minimum Separation: b, c, d

Minimum Overlap: e

Constraint Types

- During physical design optimization, three types of constraints must be satisfied:
 - Technology constraints
 - Enabling fabrication for a specific technology node and derived from technology restrictions
 - Examples: minimum layout widths and spacing values between layout shapes
 - Electrical constraints
 - Ensuring the desired electrical behavior of a design
 - Examples: meeting maximum timing constraints for signal delay and staying below maximum coupling capacitances
 - Geometry (design methodology) constraints
 - Introduced to reduce the overall complexity of the design process
 - Examples: preferred wiring directions during routing, and the placement of standard cells in rows