電子設計自動化演算法與實作
Electronic Design Automation Algorithms and Implementation

VLSI Testing
NYCU

陳宏明
Hung-Ming Chen

趙家佐
Mango Chao

Institute of Electronics

National Yang Ming Chiao Tung University

Spring 2024

EDA Courses in DEE

- Undergraduate
  - Data structures
  - Discrete mathematics
  - Algorithms
  - Introduction to EDA
- Graduate
  - Advanced algorithms
  - Special topics in CAD
    (logic synthesis & verification) – front end
  - Physical design automation – back end
  - High level synthesis
  - Testing

Administrative Matters

- **Time/Location:** Monday G@ED101 and Thursday EF@ED116
- **Instructor:** ***Hung-Ming Chen*** and Mango Chao
  - **E-mail:** hmchen@nycu.edu.tw and mango@nycu.edu.tw
  - **URL:** https://vdalab.web.nctu.edu.tw/advisor/
  - **Office:** ED407 ext. 31626
  - **Office Hours:** Monday 11-11:50m (by appointment)
- **Teaching Assistants**
  - 張皓儒 (s410510026.ee10@nycu.edu.tw) and other 3TAs
- **Prerequisites:** Data structures (or discrete math) & logic design
- **Text/Reference Books**
  - L.-T. W, Y.-W. Chang, and K.-T. Cheng, *Electronic Design Automation*, Morgan Kaufmann, 2009. ISBN: 978-0-12-374364-0
  - S. H. Gerez, *Algorithms for VLSI Design Automation*, John Wiley & Sons, 1999. ISBN: 0-471-98489-2.
  - G. De Micheli, *Synthesis and Optimization of Digital Circuits*, McGraw-Hill, Inc., 1994. ISBN: 0-07-01332-2.

Grading Policy

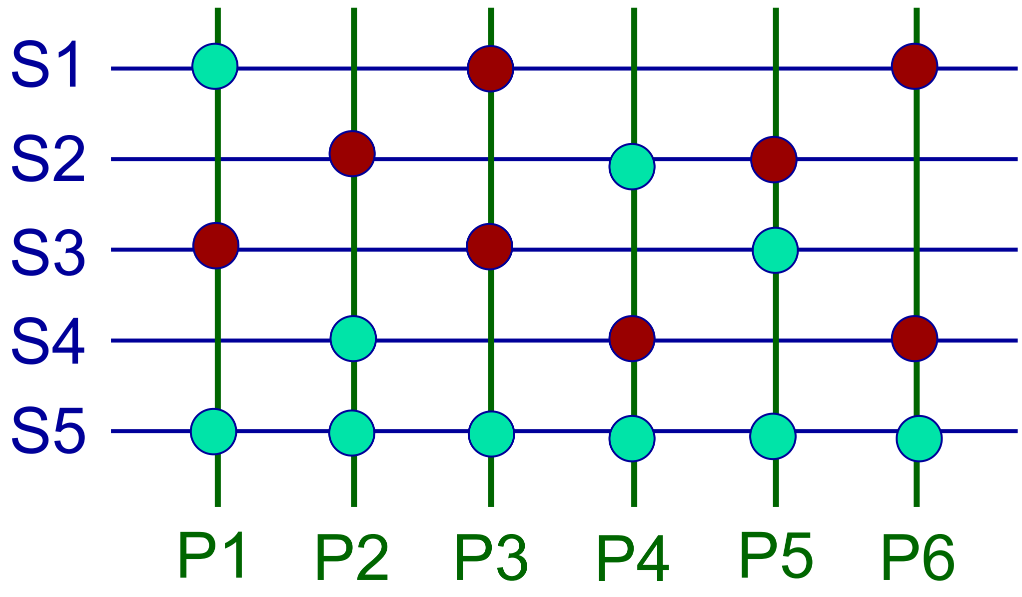
- **Grading Policy:**
  - Four programming assignments/labs: 50%
  - One exam (Late midterm: 30%)
  - Final project: 20% (with up to 10% bonus)
- **Assignments:**
  - Discussions are encouraged
  - Write down solutions individually (no credits for plagiarism)
  - No late assignment due (partial solutions may get partial credits)
- **Test (closed book):**
  - No discussions (no credits for plagiarism).
- **Class webpage:** newe3
- **Academic Honesty: Plagiarism is strongly prohibited**

Final Project Information

- Final project selection
  - Read an EDA paper from DAC/ICCAD/DATE in recent 3 years and write a report for it (>1000 words)
    - Get at most 15 points if only report is provided
    - Verify the paper by yourself can get extra 5~10 points
  - Attend CAD contest this year instead to get higher scores
    - Get 20 points if you submit your work
    - Get extra 5~10 points if your program passes the test (alpha test)
  - The implementation of one CAD topic
    - One past CAD contest problem or discuss with the instructor
    - Get extra points if you can provide better results (compared with prior published works)
  - You should decide before the end of April, submit your decision to TA (register during May)
- You can have 1-2 people for one team
  - If you attend CAD contest international problems, you can ask for 3 people in the team if necessary

Course Objectives

- Study techniques for electronic design automation (EDA), a.k.a. computer-aided design (CAD)
- Study IC technology evolution and their impacts on the development of EDA tools
- **Study problem-solving (-finding) techniques!!!**



Course Contents

- Introduction to VLSI design flow/styles/automation, EDA business(5 hrs) + EDA algorithm review (1 hr)
- Physical design: partitioning, floorplanning, placement, and routing (21 hrs)
- Logic synthesis and verification (9 hrs)
- Timing analysis (3 hrs)
- Testing (6 hrs)
- Introduction to ML/DL EDA (1 hr)
- Simulation (3 hrs)

Introduction to Electronic Design Automation

江蕙如
Hui-Ru Jiang
hrjiang@faculty.nctu.edu.tw
http://eda.ee.nctu.edu.tw/hrjiang

Department of Electronics Engineering
National Chiao Tung University
Hsinchu 300, Taiwan

Spring 2007

Introduction to Computer-Aided Design of VLSI Circuits

台灣大學電機系 張耀文

清華大學電機系 黃錫瑜

交通大學資科系 李毅郎

中央大學電機系 劉建男

元智大學資工系 林榮彬

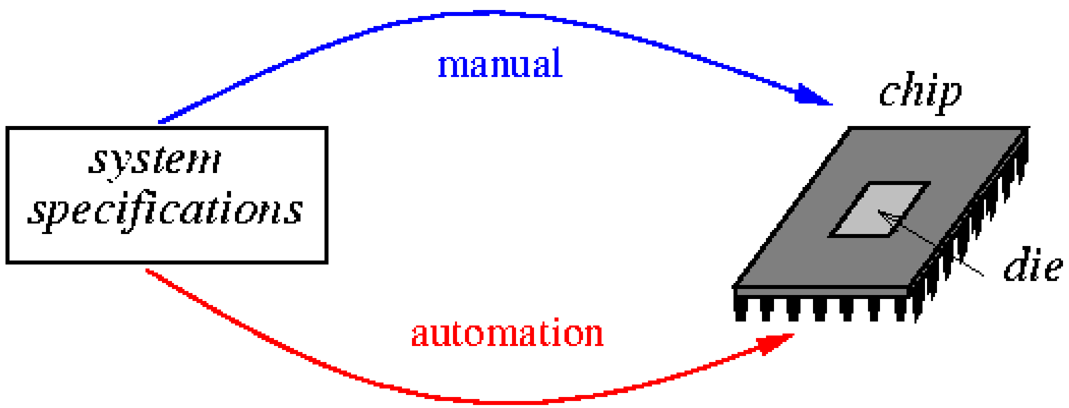
教育部

超大型積體電路與系統設計教育改進計畫

EDA聯盟

Unit 1: Introduction

- Course contents:
  - Introduction to VLSI design flow/methodologies/styles
  - Introduction to VLSI design automation tools
  - The business of EDA
  - Semiconductor technology roadmap and future trends



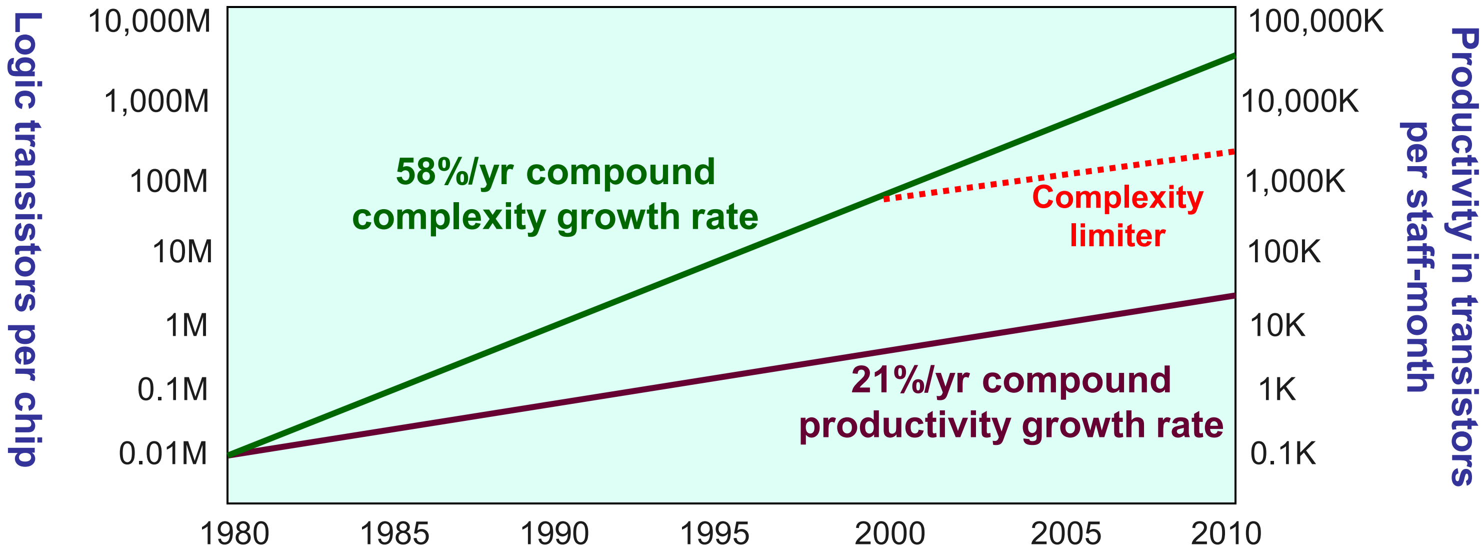
Evolving CAD / EDA

- The Industrial Revolution
  - Application of power-driven machinery to manufacturing (1750—1830)
- The 2nd Industrial Revolution!
  - Application of electronic devices to information processing (1950—present)
- Electronic systems evolve in a fascinating speed
  - Design challenges emerge and shift in this evolution
  - CAD tools exist and evolve along with the design challenges

Mission of CAD / EDA

- CAD tools aim at automating VLSI design process and optimizing **all** design instances (not just a specific design)

Mission of CAD / EDA: Solving Productivity Crisis



Source: DataQuest

- Human factors may limit design more than technology.
- Keys to solve the productivity crisis: CAD (tool & methodology), hierarchical design, abstraction, IP reuse, platform-based design, etc.

Discipline of CAD / EDA

- EDA is a field with rich applications from theoretical computer science, operations research, mathematics as well as physics
  - Algorithms, complexity theory
  - Automata theory, logics, games
  - Probability, statistics
  - Algebra
  - Numerical analysis, matrix computation
  - Device physics
  - …
- EDA is also a paradise for software engineers
  - Modern SAT solvers (e.g., GRASP, Chaff, BerkMin, MiniSAT) are developed greatly due to EDA

Brief History of EDA

- The first generation
  - 1960's: circuit board physical design automation
  - 1970's: simulation, device modeling
    - SPICE (analog simulator)
    - Technology CAD
    - Layout verification and editor
- The second generation
  - 1980's
    - LSI physical design (APR), two-level logic minimization, testing
    - Logic simulator
- The third generation
  - 1990's
    - Multi-level and sequential optimization
    - RTL methodology/synthesis and simulators, verification
- After 2000: system-level synthesis, design for manufacturing, software verification

Analogy for Electronic System Design

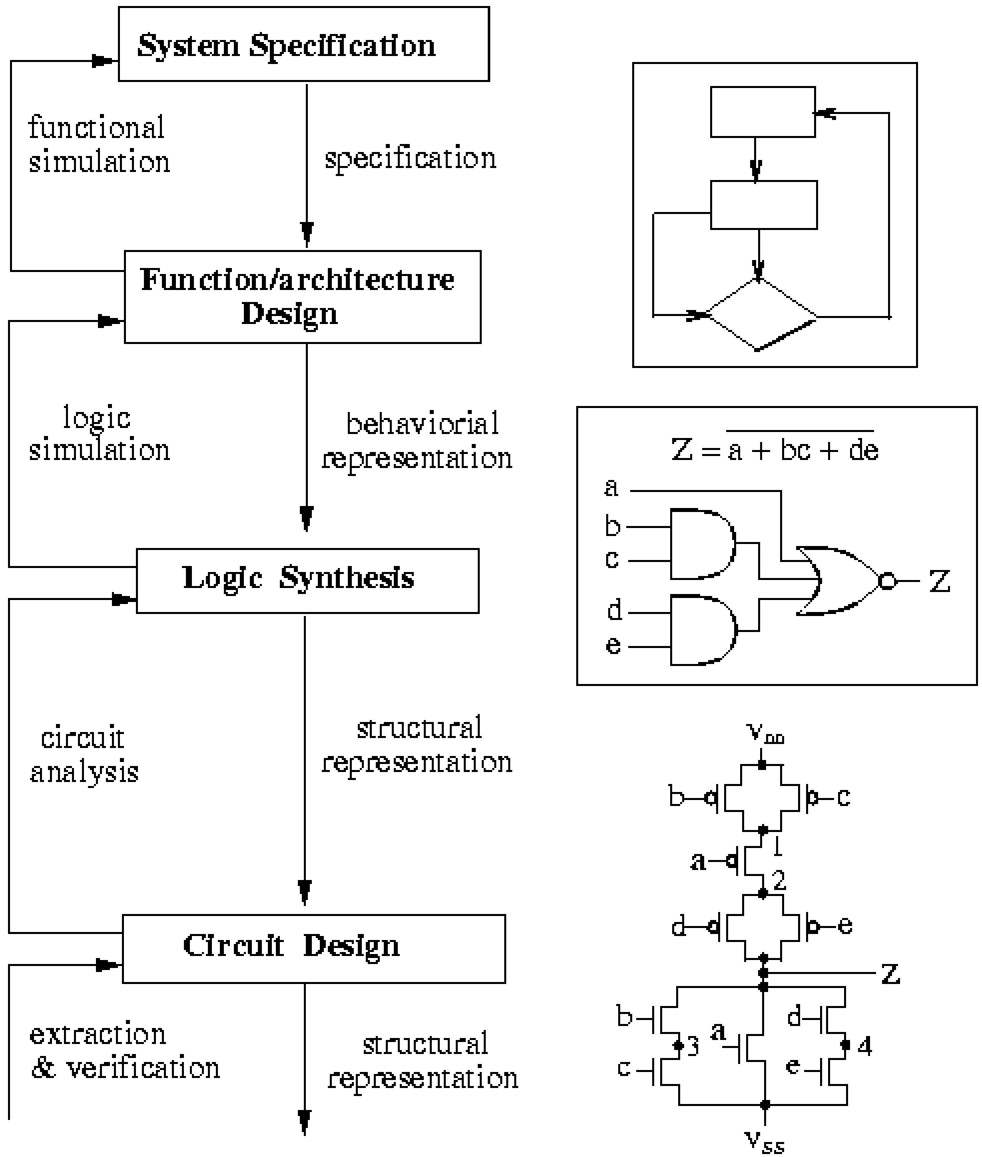
- City design
  - City planners, city architects, civil engineers and city council
- System design
  - System architects, design engineers, layout designers and software engineers/programmers
- For example, system architects decide which portions of the system will be implemented in hardware (islands and buildings), and which will be implemented in software (books of instructions in the library)

Traditional VLSI Design Cycles

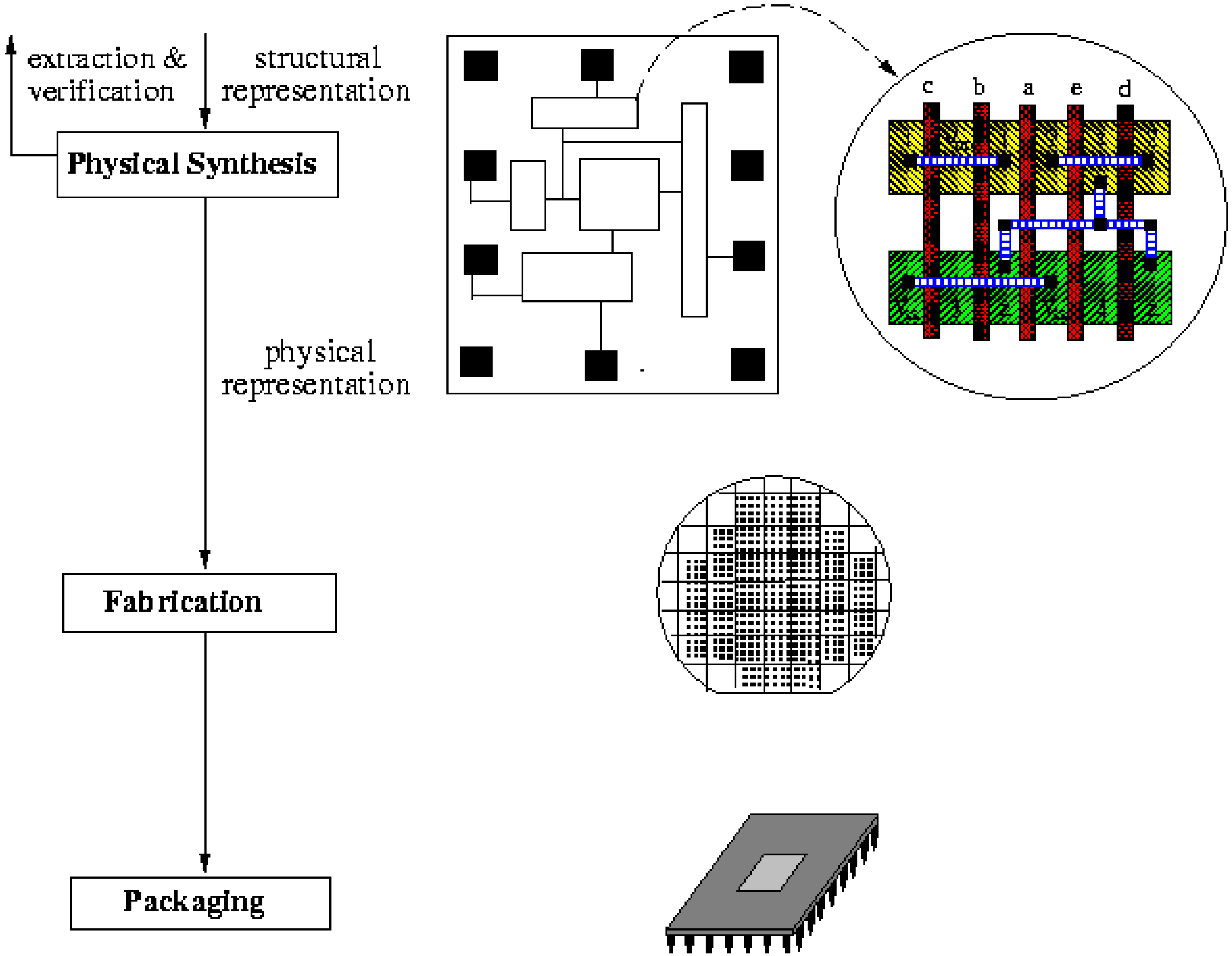
1. System specification
2. Functional design
3. Logic synthesis
4. Circuit design
5. Physical design and verification
6. Fabrication
7. Packaging

- Other tasks involved: testing, simulation, etc.
- Design metrics: area, speed, power dissipation, noise, design time, testability, etc.
- Design revolution: interconnect (not gate) delay dominates circuit performance in deep submicron era.
  - Interconnects are determined in physical design.
  - Shall consider interconnections in early design stages.

Traditional VLSI Design Flow



Traditional VLSI Design Flow (Cont'd)



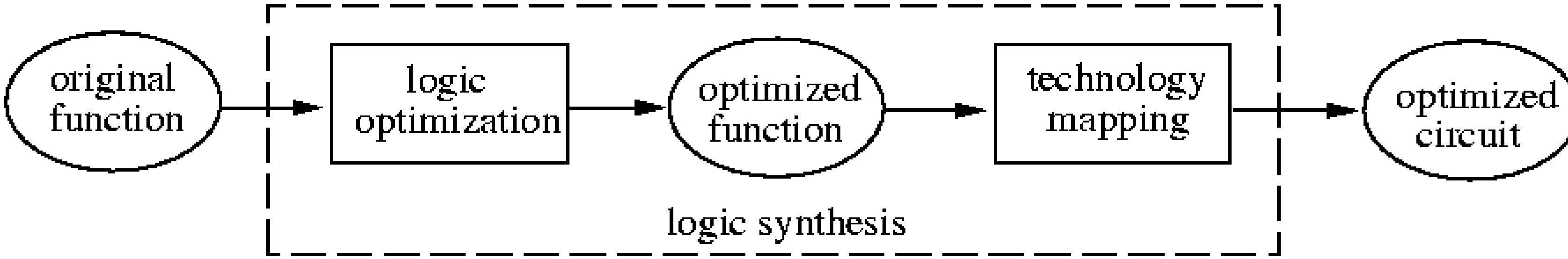
Design Actions

- **Synthesis:** increasing information about the design by providing more detail (e.g., logic synthesis, physical synthesis).
- **Analysis:** collecting information on the quality of the design (e.g., timing analysis).
- **Verification:** checking whether a synthesis step has left the specification intact (e.g., layout verification).
- **Optimization:** increasing the quality of the design by rearrangements in a given description (e.g., logic optimizer, timing optimizer).
- **Design Management:** storage of design data, cooperation between tools, design flow, etc. (e.g., database).

Design Issues and Tools

- System-level design
  - Partitioning into hardware and software, co-design, co-simulation, etc.
  - Cost estimation, design-space exploration
- Algorithmic-level design
  - Behavioral descriptions (e.g. in Verilog, VHDL)
  - High-level simulation
- From algorithms to hardware modules
  - High-level (or architectural) synthesis
- Logic design
  - Schematic entry
  - Register-transfer level and logic synthesis
  - Gate-level simulation (functionality, power, etc)
  - Timing analysis
  - Formal verification

Logic Design/Synthesis

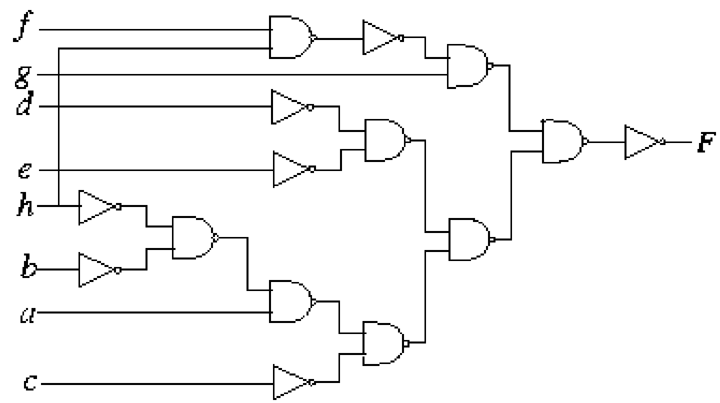


- **Logic synthesis** programs transform Boolean expressions into logic gate networks in a particular library
- Optimization goals: minimize area, delay, power, etc
- **Technology-independent** optimization: logic optimization
  – Optimizes Boolean expression equivalent.
- **Technology-dependent** optimization: **technology mapping/library binding**
  – Maps Boolean expressions into a particular cell library.

Logic Optimization Examples

- **Two-level:** minimize the # of product terms.
  - $F = \bar{x}_1\bar{x}_2\bar{x}_3 + \bar{x}_1\bar{x}_2 x_3 + x_1\bar{x}_2\bar{x}_3 + x_1\bar{x}_2 x_3 + x_1 x_2 \bar{x}_3 \Rightarrow F = \bar{x}_2 + x_1\bar{x}_3$.
- **Multi-level:** minimize the #'s of literals, variables.
  - E.g., equations are optimized using a smaller number of literals.

$$
\begin{aligned}
t1 &= a + b\ c; \\
t2 &= d + e; \\
t3 &= a\ b + d; \\
t4 &= t1\ t2 + f\,g; \\
t5 &= t4\ h + t2\ t3; \\
F &= t5';
\end{aligned}
\quad \xrightarrow{\text{logic optimization}} \quad
\begin{aligned}
t1 &= d + e; \\
t2 &= b + h; \\
t3 &= a\ t2 + c; \\
t4 &= t1\ t3 + f\,g\,h;
\end{aligned}
$$



subject graph for the optimized equations

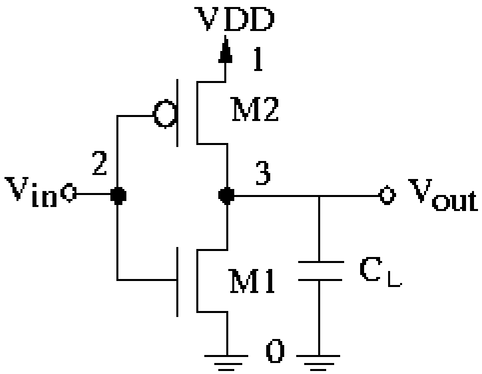
- Methods/CAD tools: Quine-McCluskey method (exponential-time exact algorithm), Espresso (heuristics for two-level logic), MIS (heuristics for multi-level logic), Synopsys, etc.

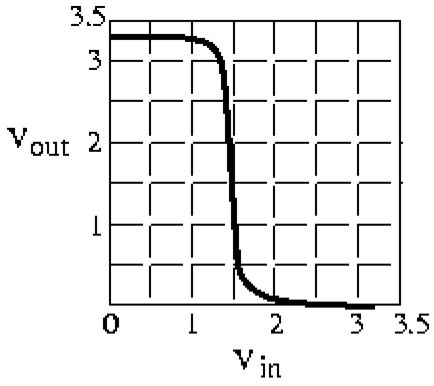
Design Issues and Tools (cont'd)

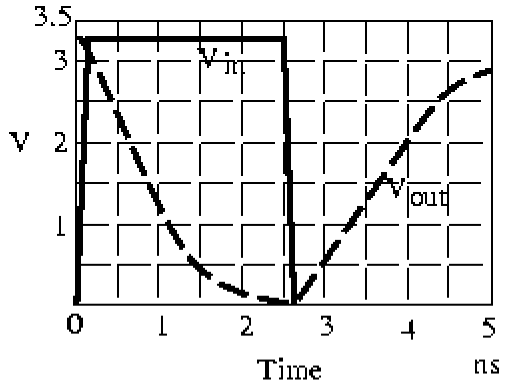
- Transistor-level design
  - Switch-level simulation
  - Circuit simulation
- Physical (layout) design
  - Partitioning
  - Floorplanning and Placement
  - Routing
  - Layout editing and compaction
  - Design-rule checking
  - Layout extraction
- Design management
  - Data bases, frameworks, etc.
- Silicon compilation: *from algorithm to mask patterns*
  - The *idea* is approached more and more, but still far away from a single *push-buttom* operation

Circuit Simulation of a CMOS Inverter (0.6 μm)

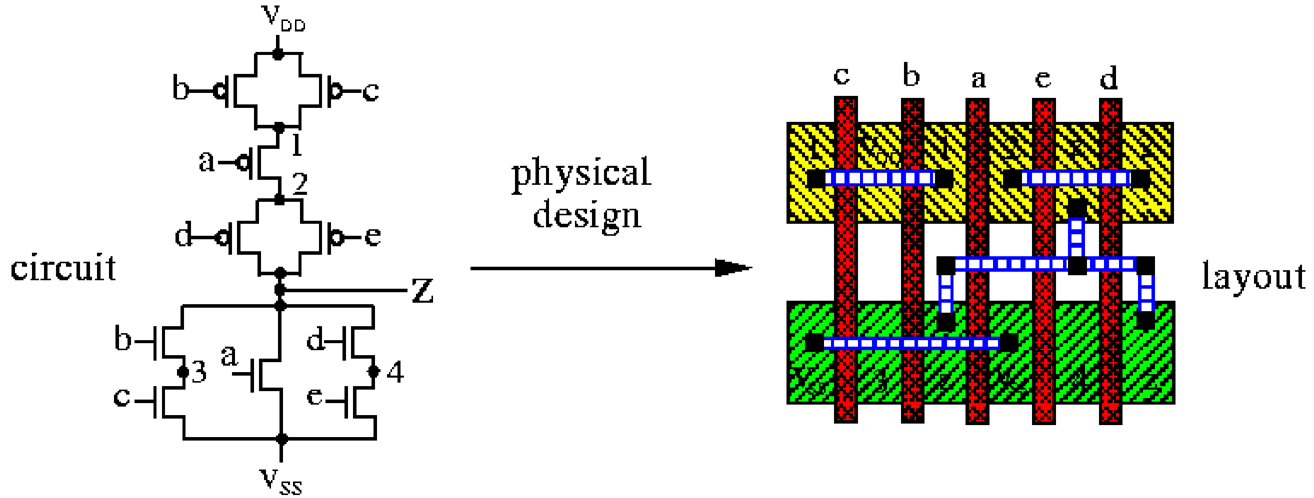
```
M1 3 2 0 0 nch W=1.2u L=0.6u AS=2.16p PS=4.8u AD=2.16p PD=4.8u
M2 3 2 1 1 pch W=1.8u L=0.6u AS=3.24p PS=5.4u AD=3.24p PD=5.4u
CL 3 0 0.2pF

VDD 1 0 3.3
VIN 2 0 DC 0 PULSE (0 3.3 0ns 100ps 100ps 2.4ns 5ns)

.LIB '../mod_06' typical

.OPTION NOMOD POST INGOLD=2 NUMDGT=6 BRIEF
.DC VIN 0V 3.3V 0.001V
.PRINT DC V(3)
.TRAN 0.001N 5N
.PRINT TRAN V(2) V(3)
.END
```





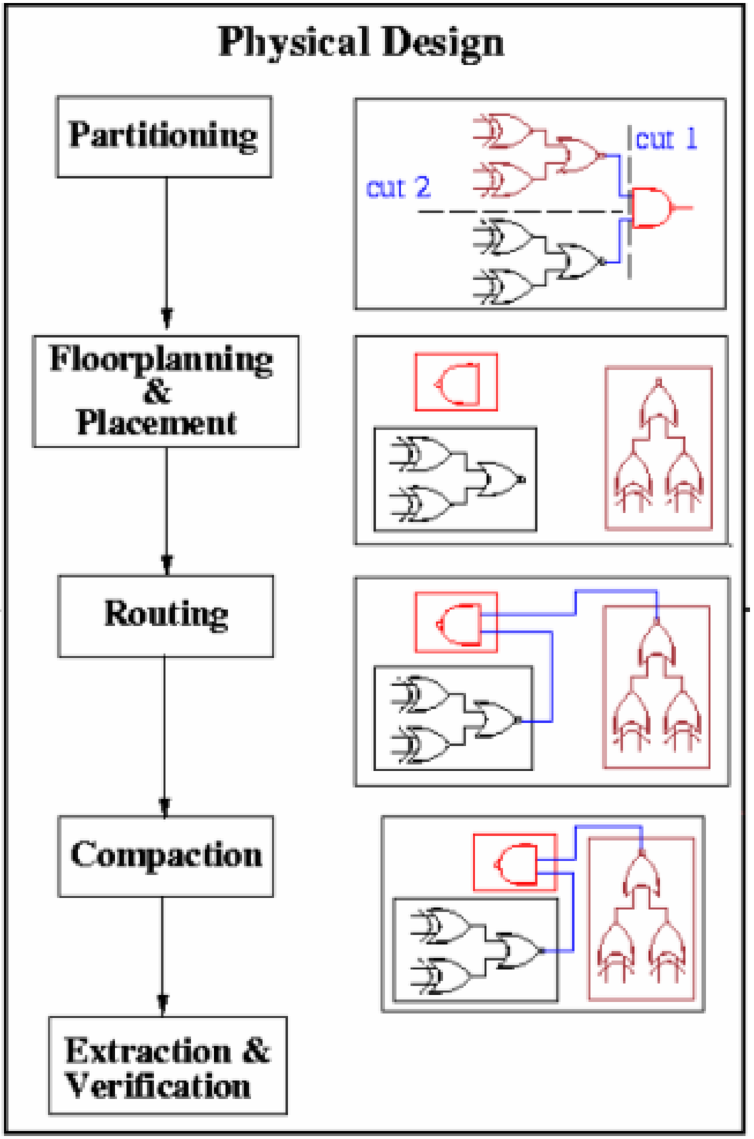


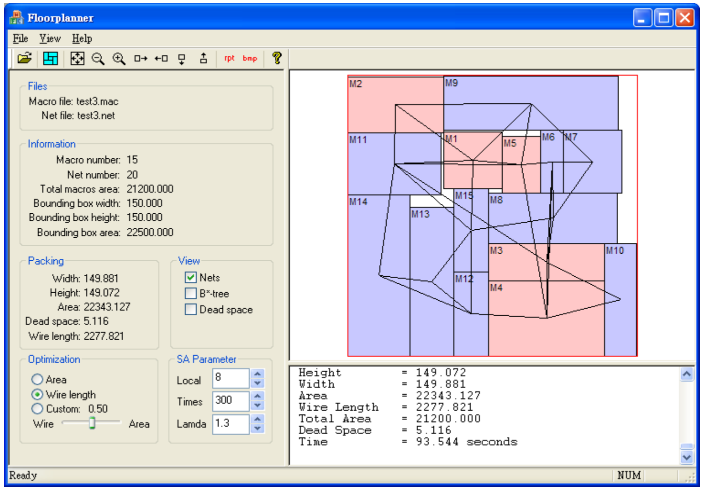
Physical Design



- Physical design converts a circuit description into a geometric description.
- The description is used to manufacture a chip.
- Physical design cycle:
  1. Logic partitioning
  2. Floorplanning and placement
  3. Routing
  4. Compaction
- Others: circuit extraction, timing verification and design rule checking

Physical Design Flow



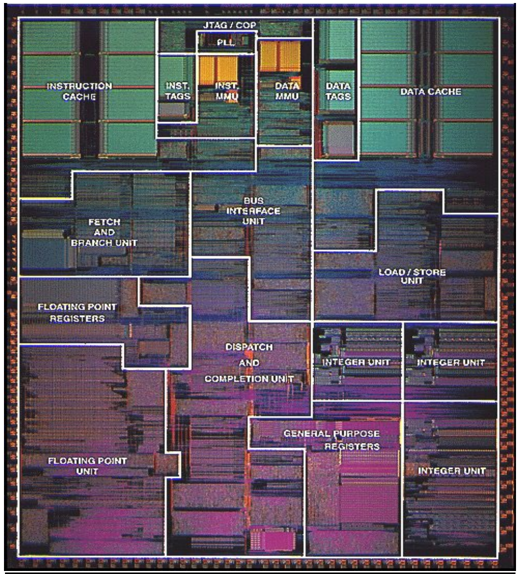


B*-tree based floorplanning system

A routing system

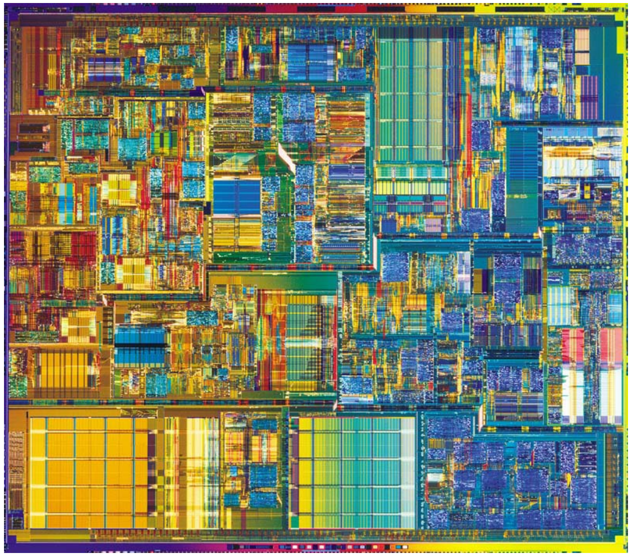
Floorplan Examples

PowerPC 604

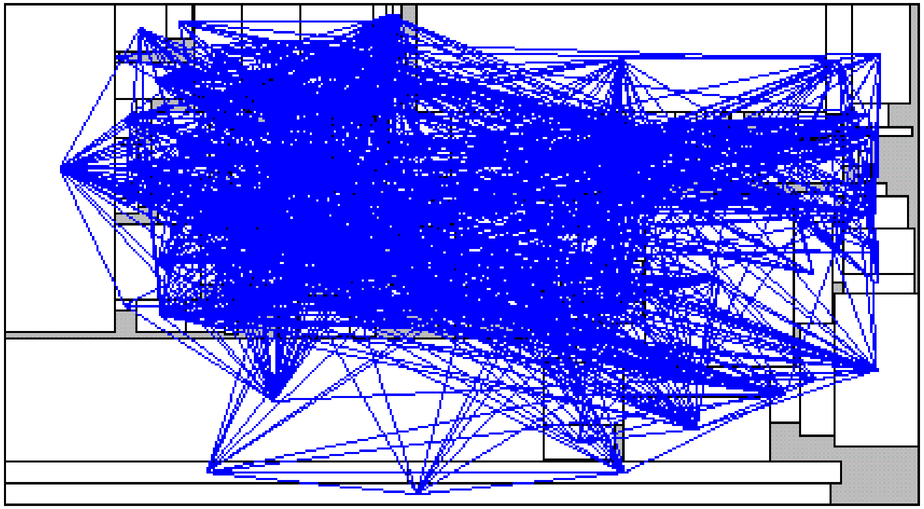


Motorola's PowerPC™ 604 RISC Microprocessor

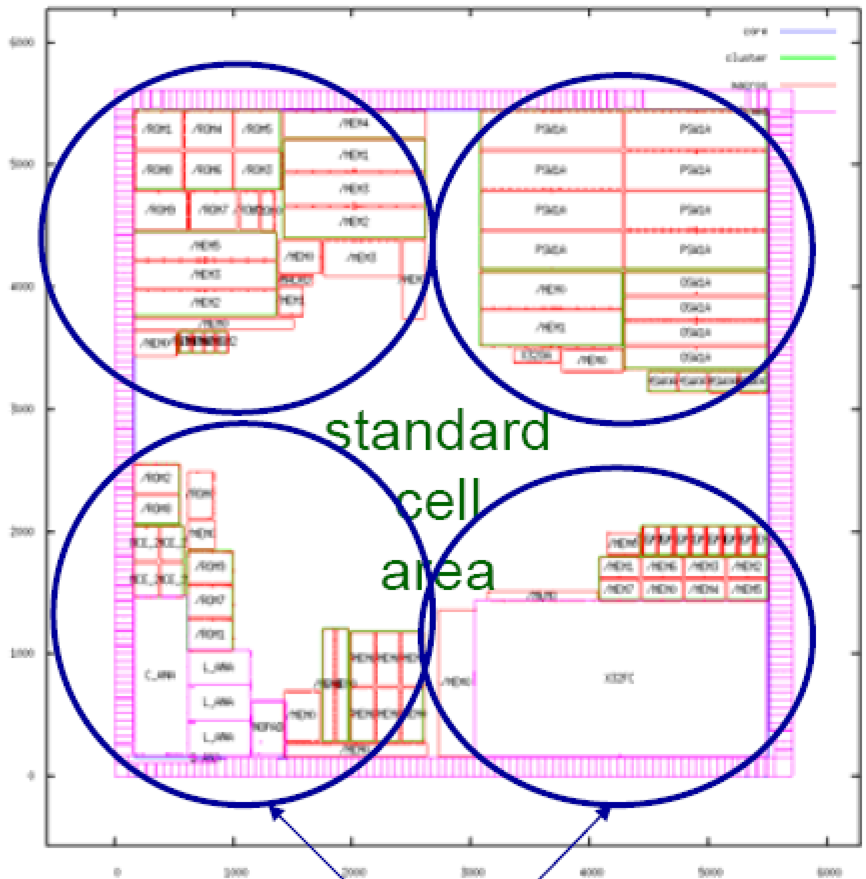
Intel Pentium 4



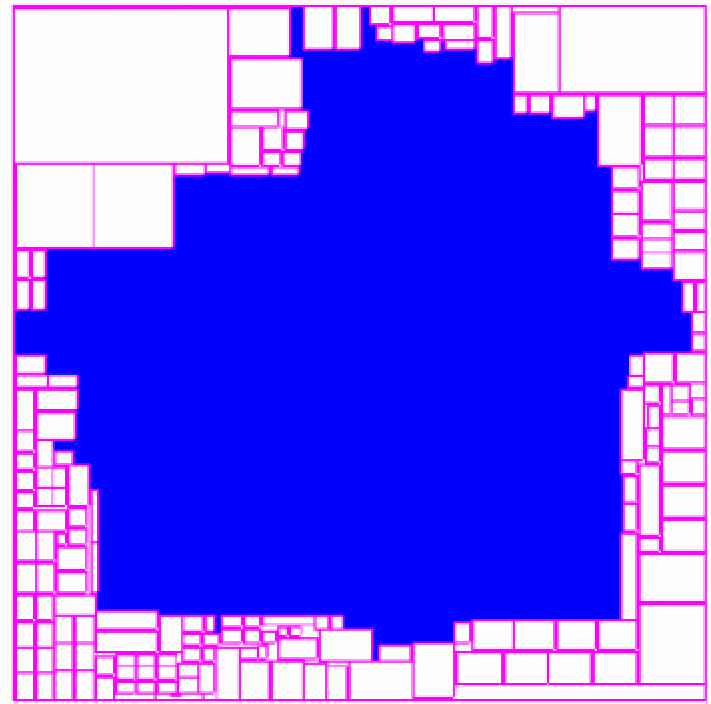
A floorplan with interconnections



Mixed Macro/Cell Placement



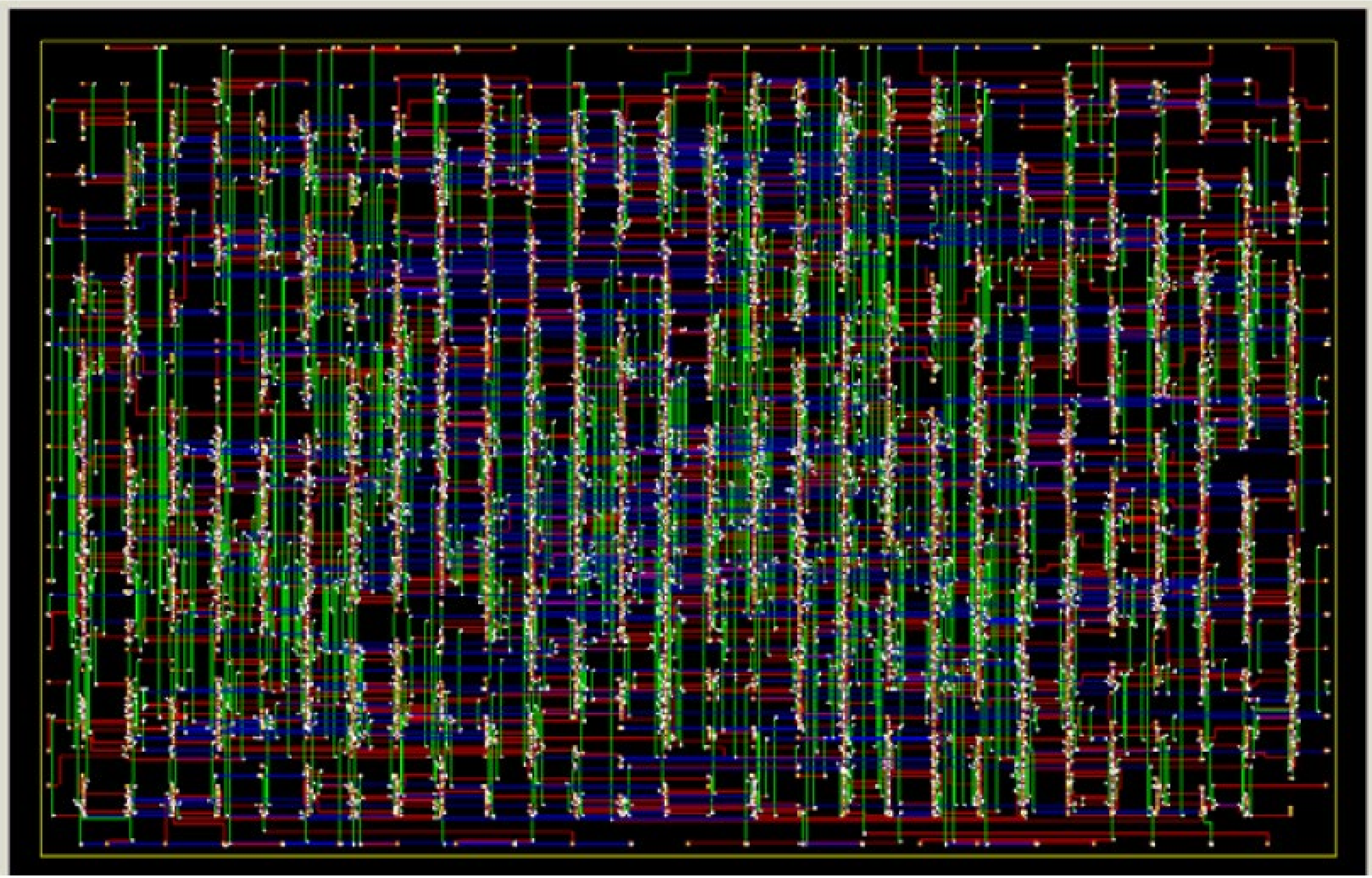
macro areas
placed by 4 B*-trees



Mchip4
1.32 million cells + 380 macros

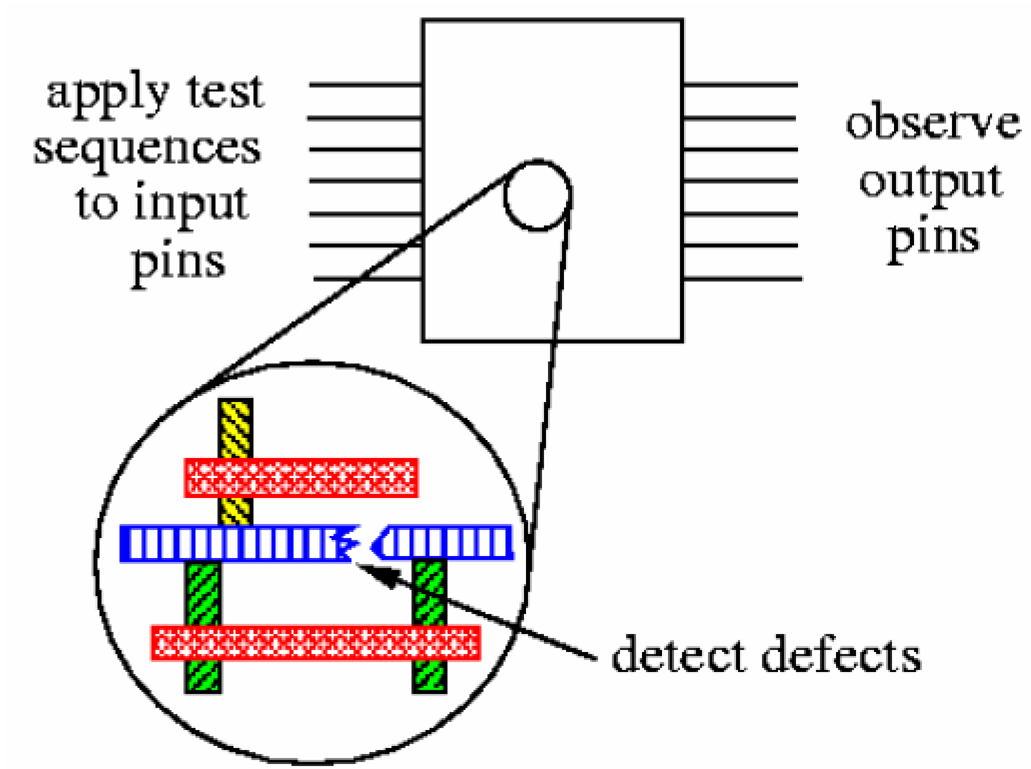
Routing with Via Insertion Example

- Circuit: s5378; 3-layer, 4818 nets



Testing

- Verify if a design was manufactured correctly



The Business of EDA: Introduction

- The size of EDA industry
  - In 2002, the total revenue was about USD$4 billion dollars
    - The semiconductor industry revenue: $140 billion (2012 $303 billion)
    - There were about 200 companies and about 50 different kinds of tools
- How do the EDA tools help in IC design?
  - The tools make the designers much more productive

The Business of EDA: ROI

- EDA user return on investment
  - EDA tools have a potentially huge ROI for the Original Equipment Manufacturer (OEM) electronics product companies
  - EDA tools can create big revenue gains by increasing the life of electronic products and getting to market earlier
- EDA vendor return on investment
  - We can typically sell a tool for anywhere from $10K to $200K per copy, depending on how critical it is
  - There is a large difference between the development cost and the value to the user
- Examples on the blackboard

The Business of EDA: Business Models

- New EDA tools adoption in design houses
  - Reference flow
- Licensing models
  - Perpetual vs. Subscription
- Mergers and acquisitions
  - Usually with lawsuits
- Application service provider model
  - Through the Internet on a usage basis, cloud model
- Design services business
  - Including selling re-usable design blocks, namely IP
  - What is the most profitable IP in the market? Guess
- Foundry services
  - Especially for physical design issues

Existing EDA Tools Chronicles (1/2)

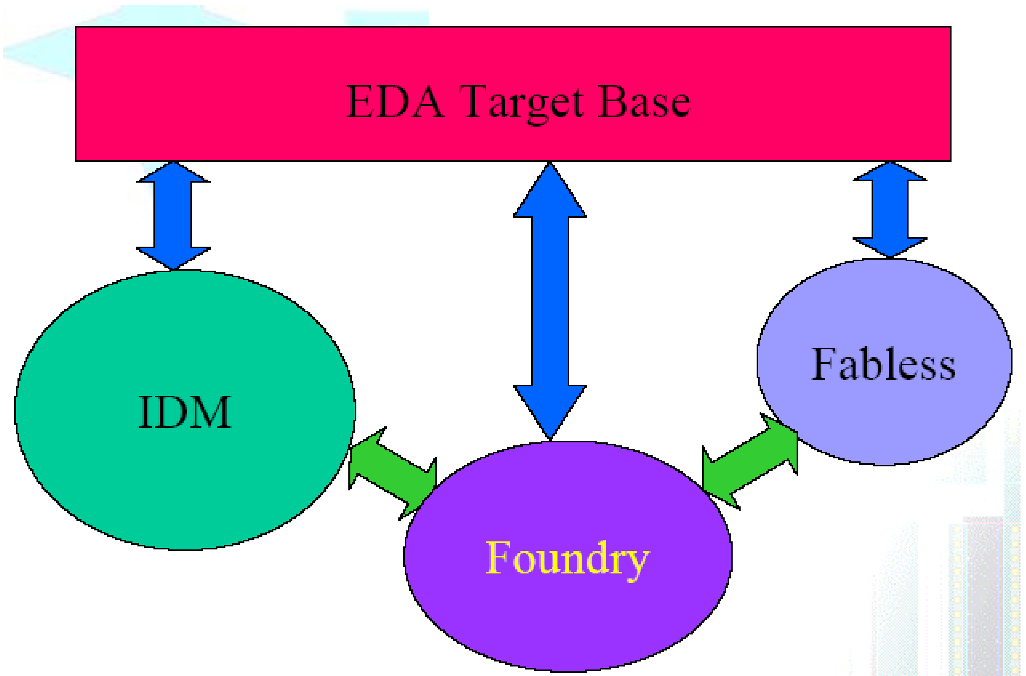
Year	Design Tools	Company
1950 ~ 1960	Manual design	
1965 ~ 1975	Layout editors Automatic routers (for PCB) Efficient partitioning algorithm	
1975 ~ 1985	Automatic placement tools Well defined phase of design of circuits Significant theoretical development in all phases	Applicon Calma Computervision
1985 ~ 1990	Performance driven placement and routing tools Parallel algorithms for physical design Significant development in underlying graph theory Combinatorial optimization problems for layouts	Daisy Mentor Valid

Existing EDA Tools Chronicles (2/2)

Year	Design Tools	Company
1990 ~ 1995	Over-the-Cell Routing tools Three dimensional interconnect based physical design Synthesis tools mature and gain widespread acceptance	Avanti(Synopsys) Cadence Synopsys
1995 ~ Present	Interconnect design and Modeling dominates physical design Process related tools (reliability, electro-migration)	Magma (Synopsys) Monterey Verplex (Cadence) SPC(Cadence) Numerical (Synopsys) Springsoft (Synopsys) EverCAD (Mentor)

IC/SoC Industry Segments in Taiwan

- Integrated Device Manufacturers (IDM)
- Foundries
- Fabless Semiconductor Companies
- EDA Vendors

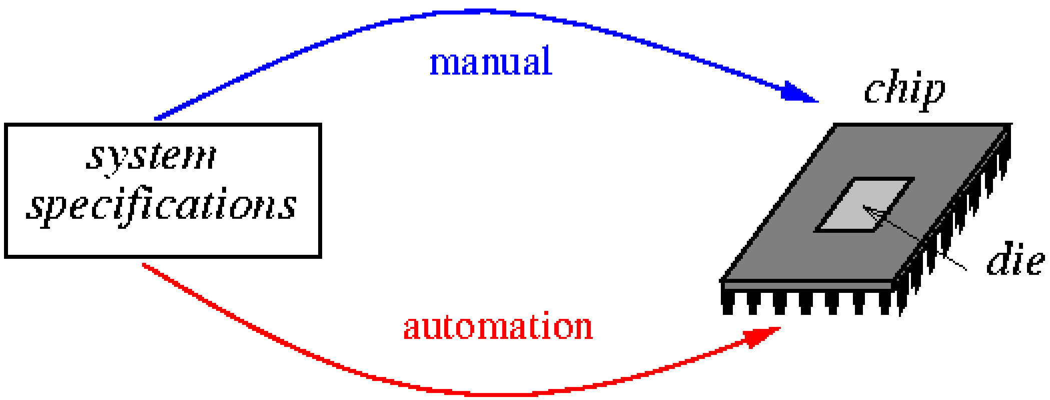


Courtesy from Prof. Chi-Wei Liu, NCTU EE

What Techniques Do We Need in EDA Business?

- The best of both worlds
- Software background
  - Algorithms and programming language
  - Data structures play important roles
- Hardware background
  - Device modeling
  - Interconnect modeling
  - Manufacture related issues
  - Circuit design and synthesis
- Combinatorial optimization
  - Unconstrained and constrained optimization
  - Mathematical programming
- Validation and verification techniques
  - Formal methods

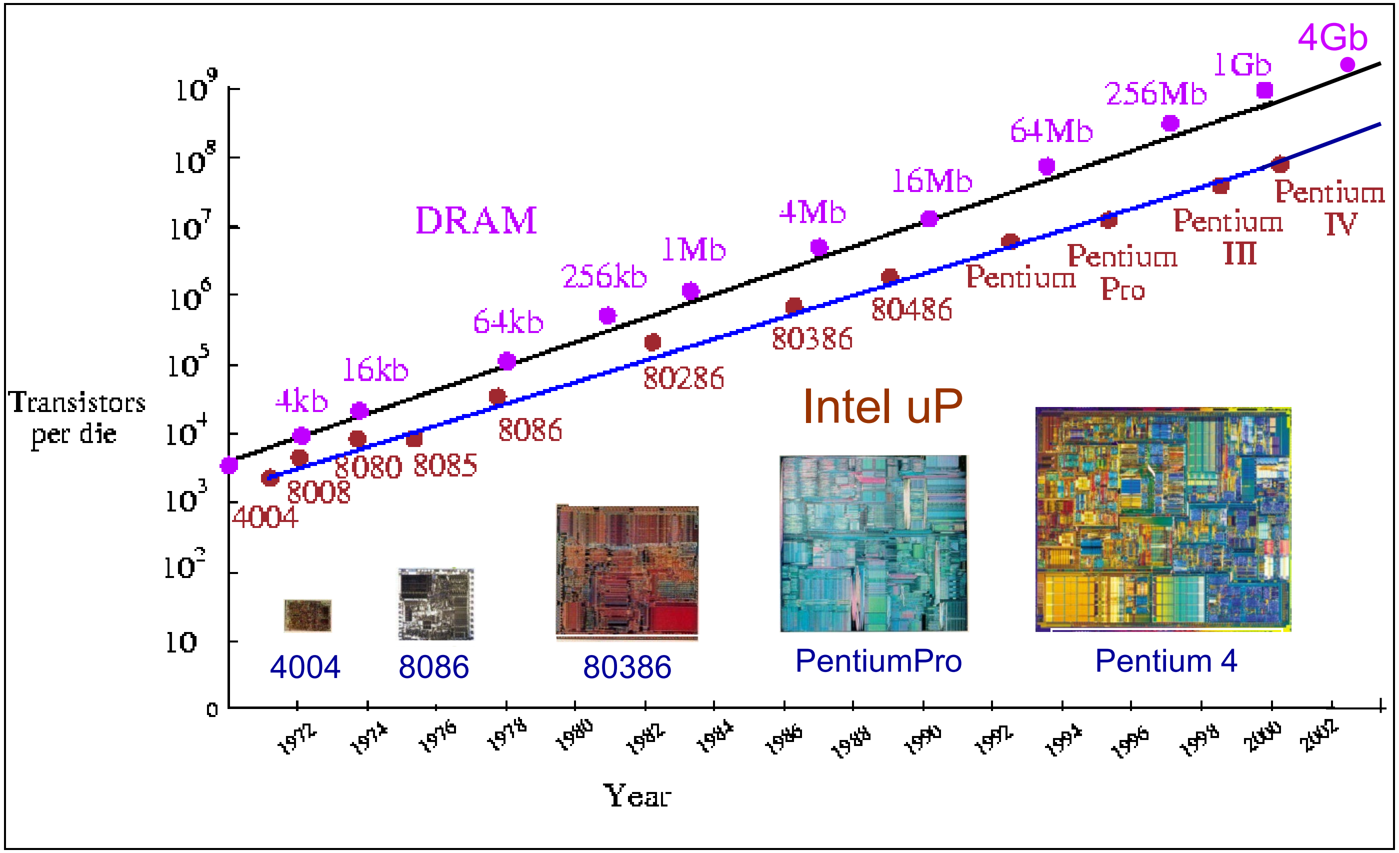
IC Design Considerations



- 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, signal integrity (noise, etc), testability, reliability, manufacturability, etc.

"Moore's" Law: Driving Technology Advances

- Logic capacity doubles per IC at a regular interval.
- Moore: Logic capacity doubles per IC every two years (1975).
- D. House: Computer performance doubles every 18 months (1975)



Technology Roadmap for Semiconductors

Year	1997	1999	2002	2005	2008	2011	2014
Technology node (nm)	250	180	130	100	70	50	35
On-chip local clock (GHz)	0.75	1.25	2.1	3.5	6.0	10	16.9
Microprocessor chip size (mm^2)	300	340	430	520	620	750	901
Microprocessor transistors/chip	11M	21M	76M	200M	520M	1.40B	3.62B
Microprocessor cost/transistor ($\times 10^{-8}$ USD)	3000	1735	580	255	110	49	22
DRAM bits per chip	256M	1G	4G	16G	64G	256G	1T
Wiring level	6	6–7	7	7–8	8–9	9	10
Supply voltage (V)	1.8–2.5	1.5–1.8	1.2–1.5	0.9–1.2	0.6–0.9	0.5–0.6	0.37–0.42
Power (W)	70	90	130	160	170	175	183

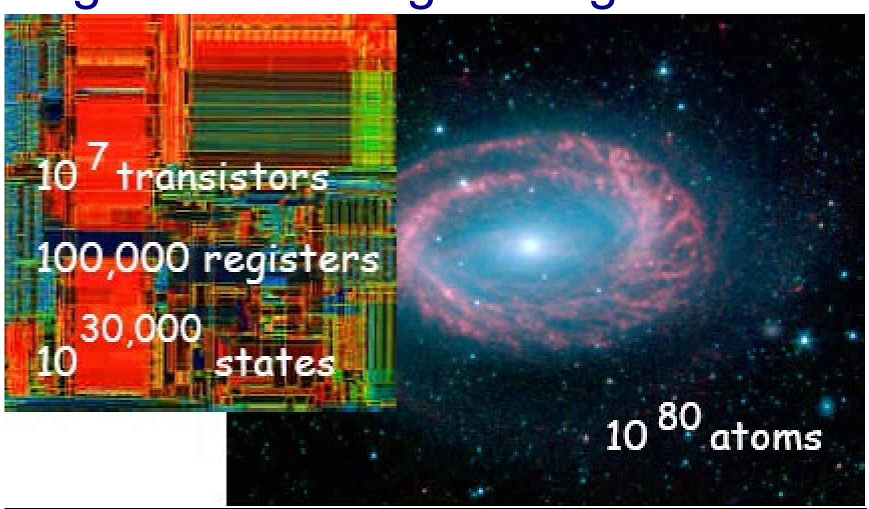
- Source: International Technology Roadmap for Semiconductors (ITRS), Nov. 2002. http://www.itrs.net/ntrs/publntrs.nsf.
- Deep submicron technology: node (**feature size**) < 0.25 μm.
- Nanometer Technology: node < 0.1 μm.

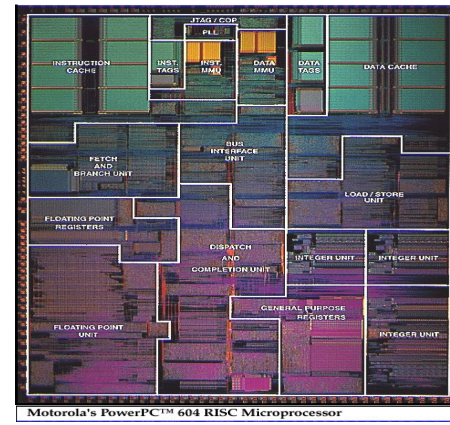
Nanometer Design Challenges

- In 2007, feature size ≈ 65 *nm*, μP frequency ≈ 5 GHz, die size ≈ 600 mm^2, μ P transistor count per chip ≈ 500M, wiring level ≈ 9 layers, supply voltage ≈ 0.9 V, power consumption ≈ 170 W.
  - **Chip complexity**
    - **Effective design and verification methodology? More efficient optimization algorithms? Time-to-market?**
  - **Power consumption**
    - **Power & thermal issues?**
  - **Supply voltage**
    - **Signal integrity (noise, IR drop, etc)?**
  - **Feature size, dimension**
    - **Sub-wavelength lithography (impacts of process variation)? noise? wire coupling? reliability? manufacturability? 3D layout?**
  - **Frequency**
    - **Interconnect delay? electromagnetic field effects? Timing closure?**

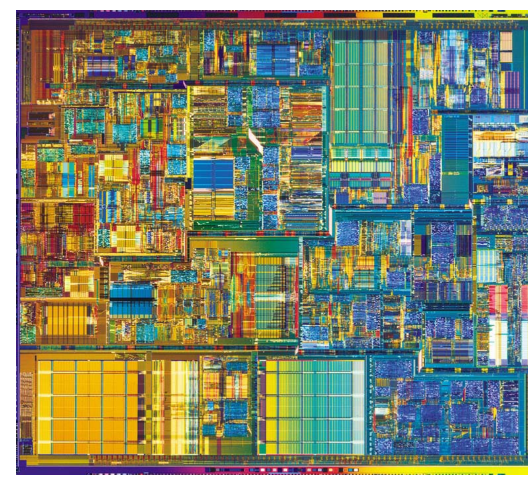
Design Complexity Increases Dramatically!!

- Design issues
  - Design space exploration
  - More efficient optimization algorithms
- Verification issues
  - State explosion problem
  - For modern designs, about 60%-80% of the overall design time was spent on verification; 3-to-1 head count ratio between verification engineers and logic designers





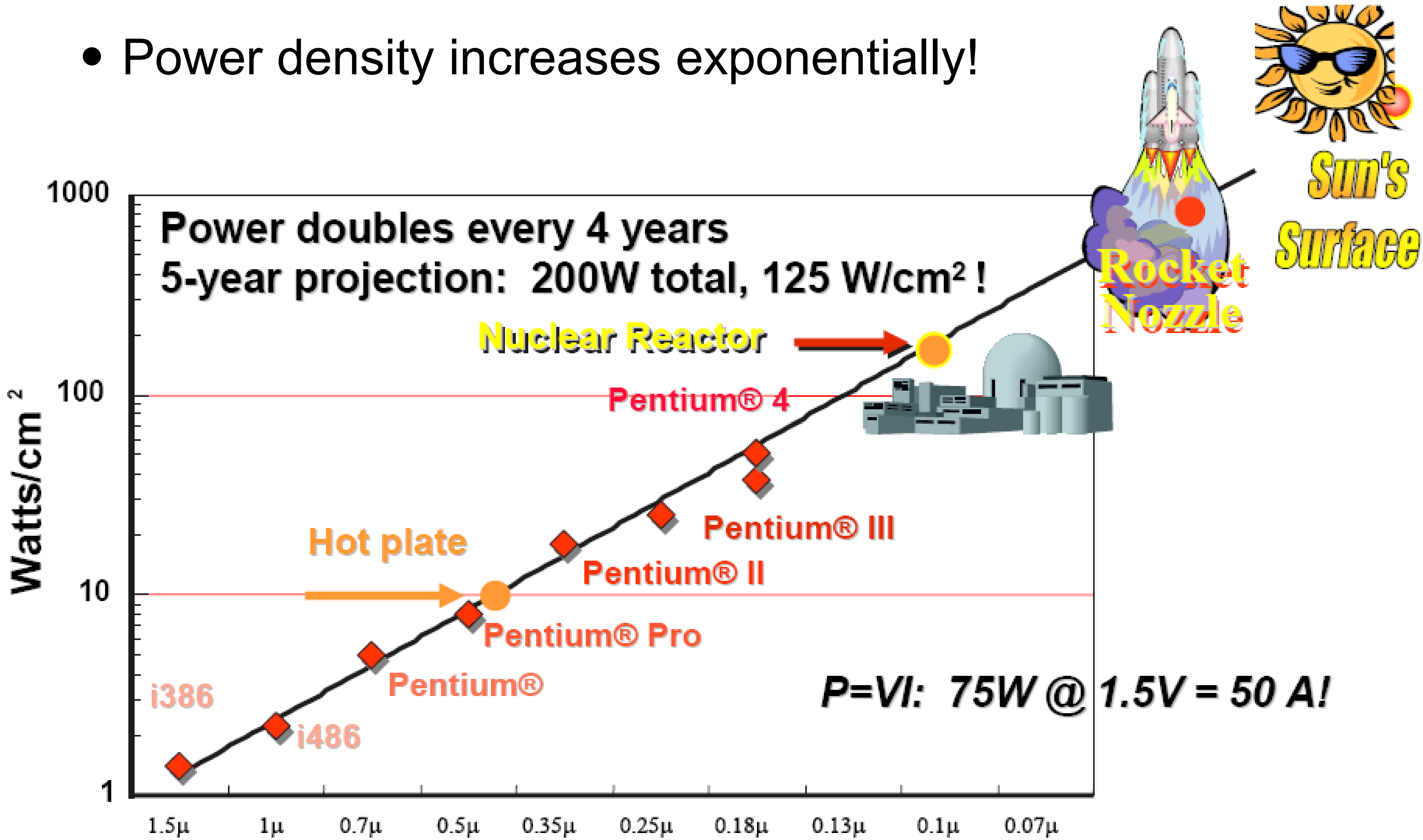
PowerPC 604



Intel Pentium 4

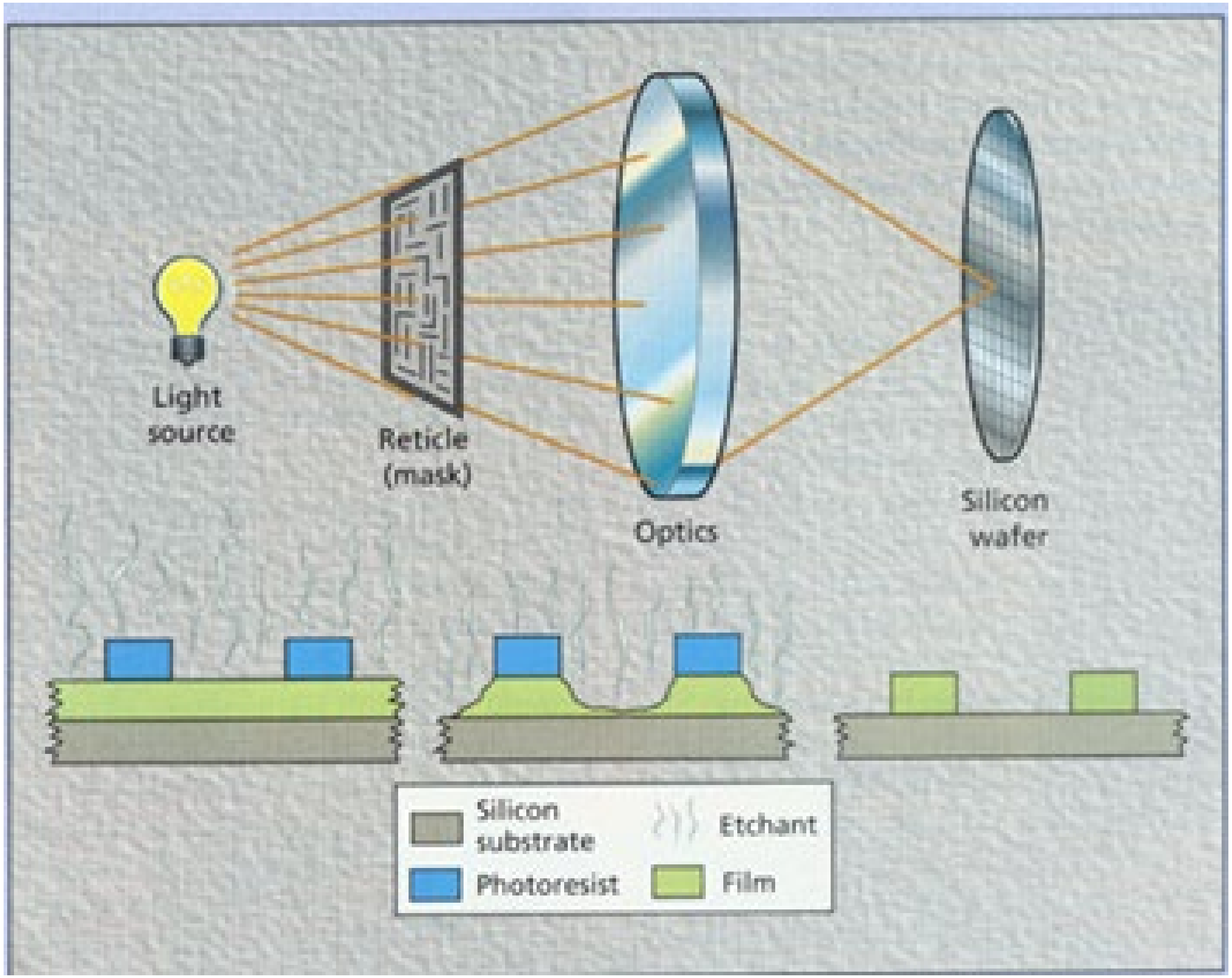
Power/Thermal Is Another Big Problem!!

- Power density increases exponentially!



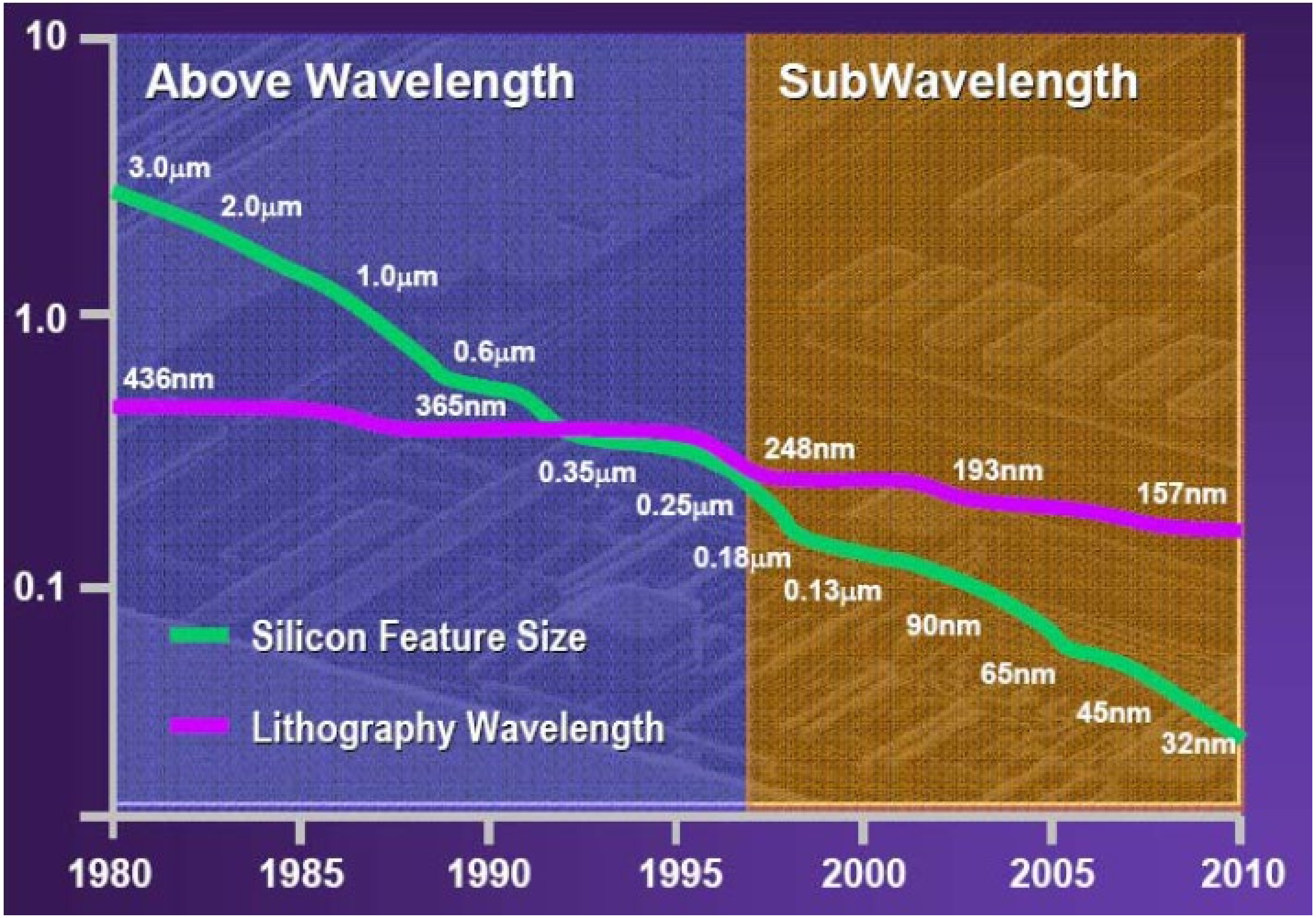
Fred Pollack, "New Microarchitecture Challenges in the Coming Generations of CMOS Process Technologies," 1999 Micro32 Conference keynote. Courtesy Avi Mendelson, Intel.

Lithography Process

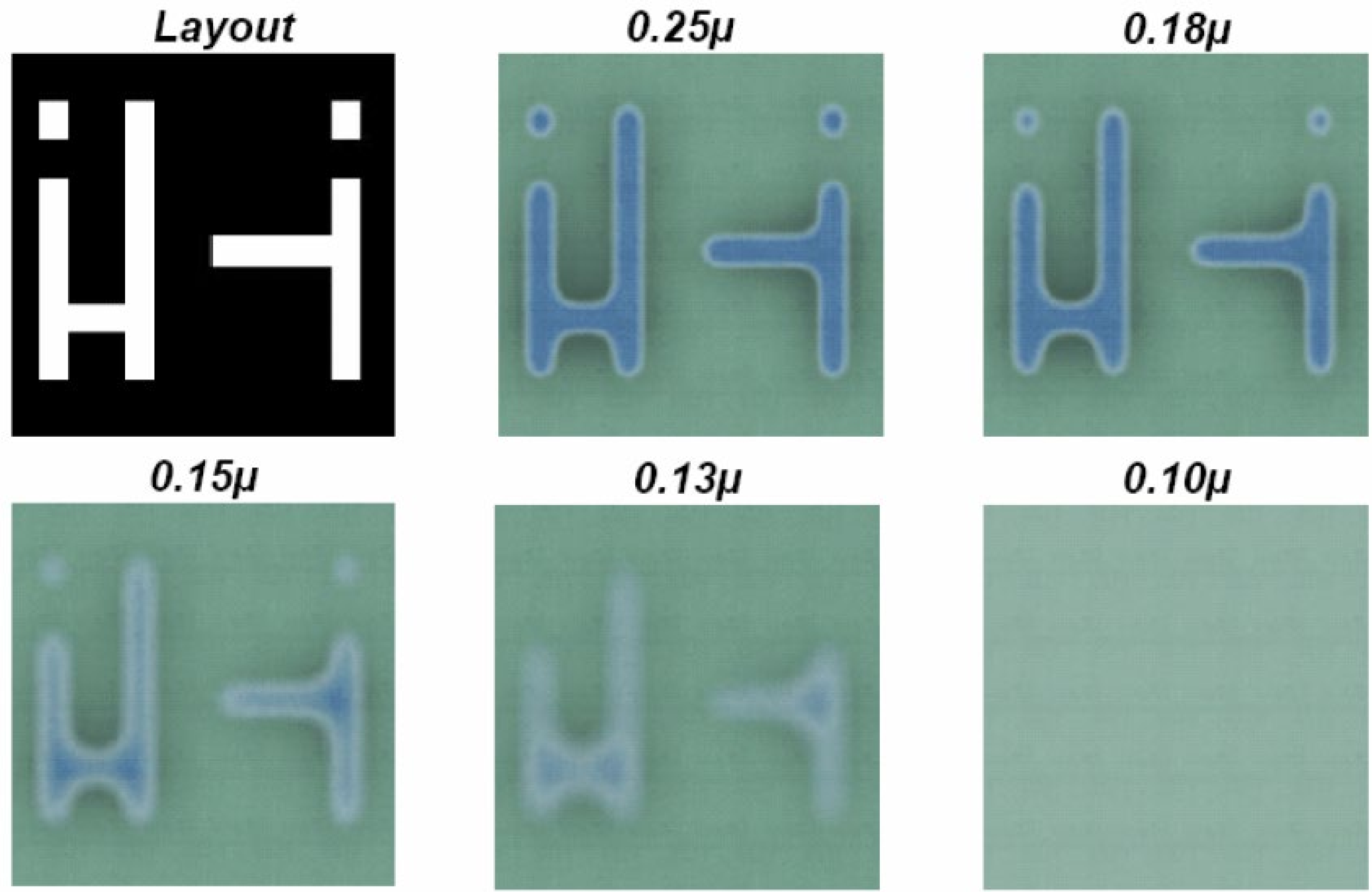


Subwavelength Lithography Gap

- Printed feature size is smaller than the wavelength of the light shining through the mask

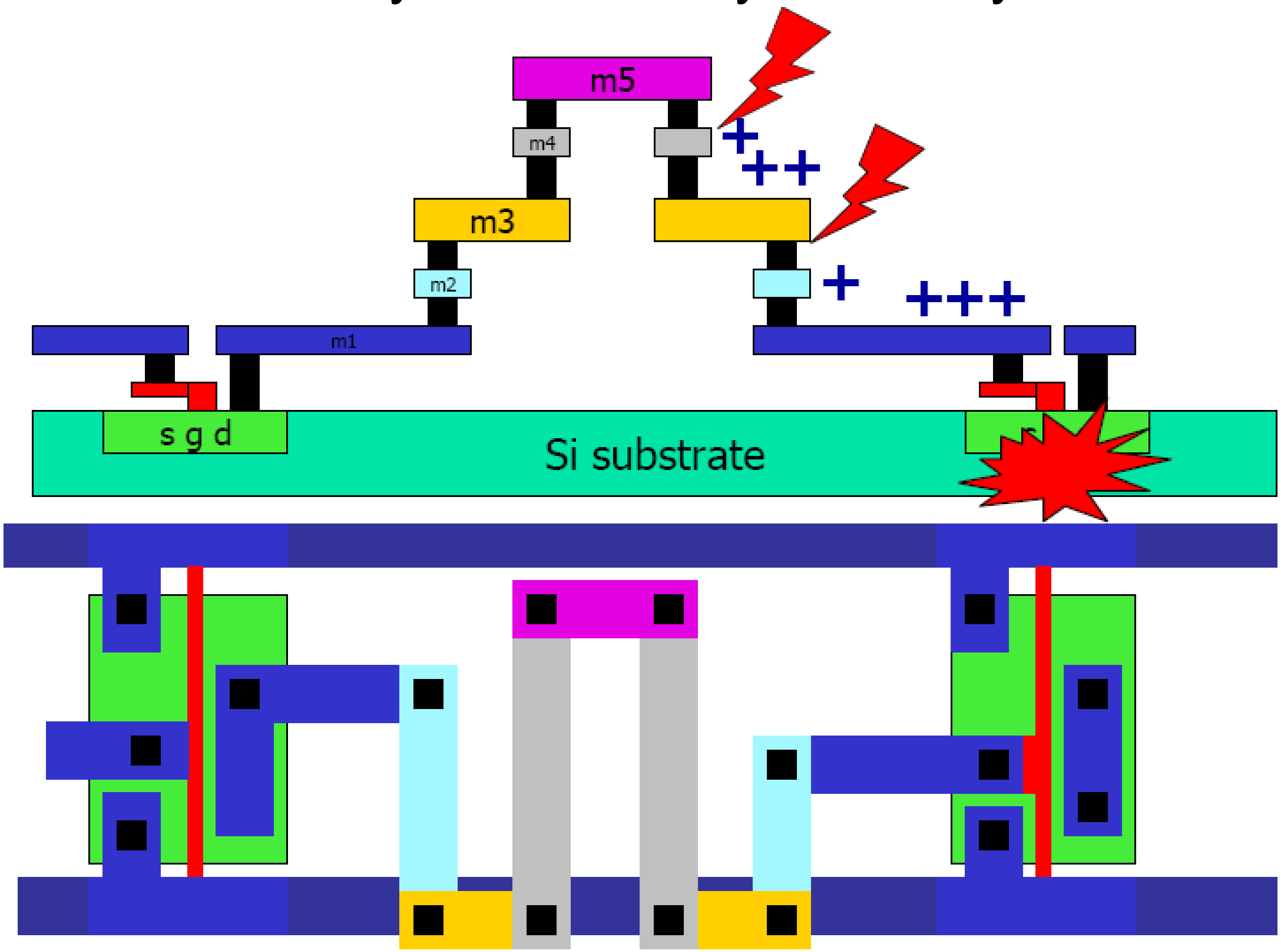


Tale of Disappearing Silicon



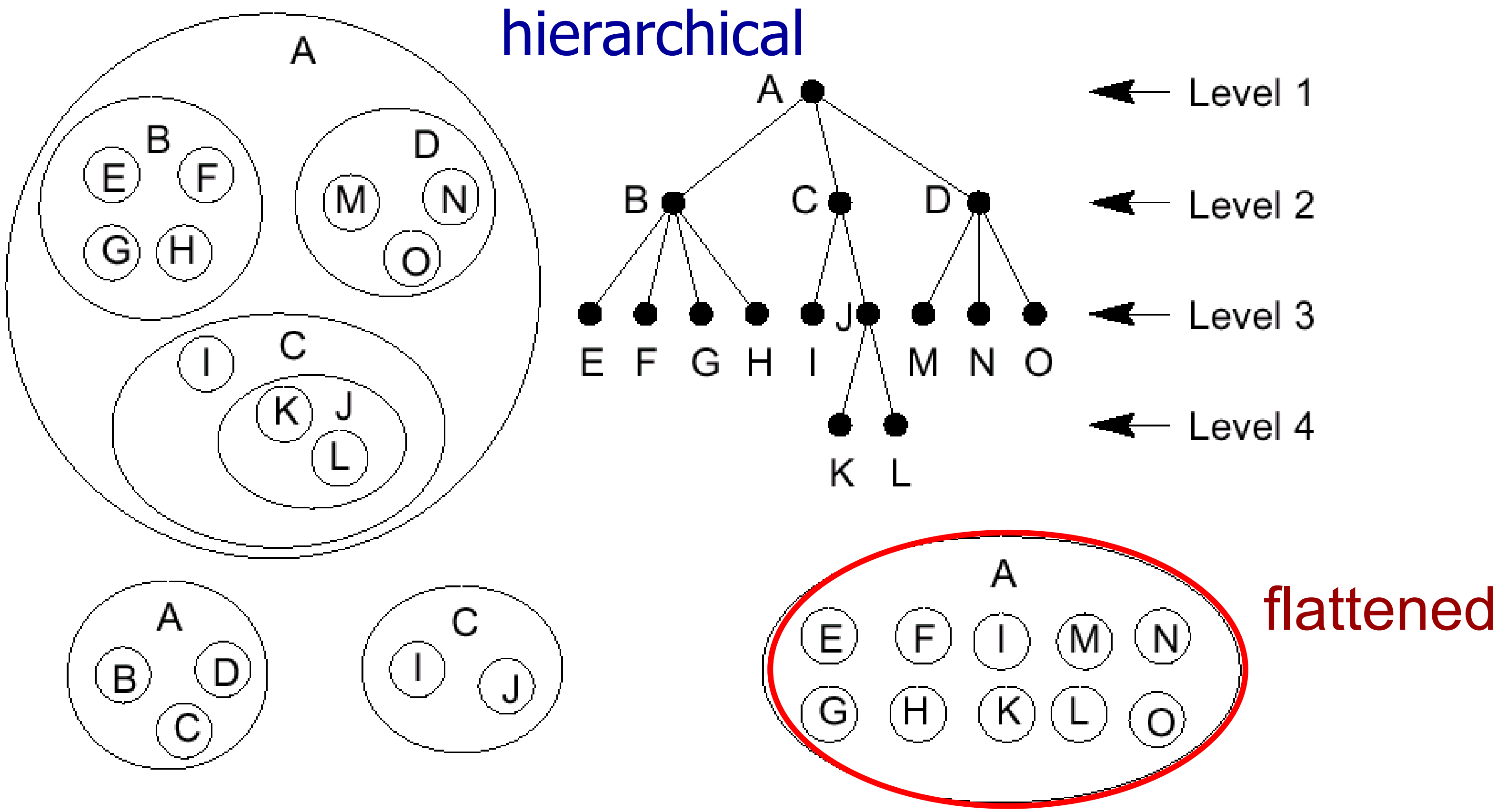
Manufacturability Becomes a 1st-Order Effect!!

- Manufacturability and reliability with 9-layer metal?



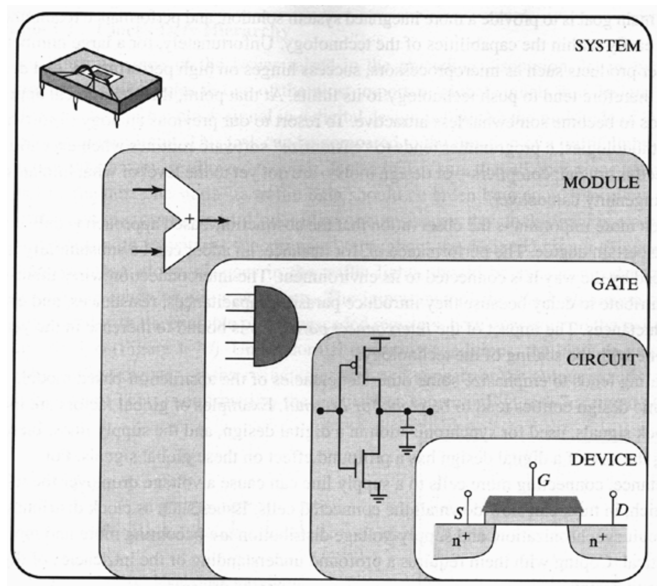
Go Back to Design

- *Hierarchy:* something is composed of simpler things.
- Design cannot be done in one step ⇒ partition the design hierarchically.



Abstraction

- ***Abstraction:*** when looking at a certain level, you don't need to know all details of the lower levels.



system
module
gate
circuit
device

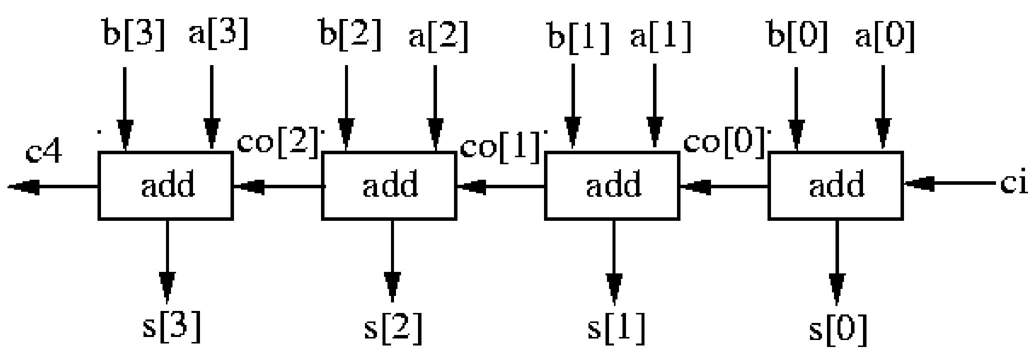
- Design domains:
  - Behavioral: functionality of components
  - Structural: connectivity between components
  - Physical: layout description
- Each design domain has its own hierarchy.

Three Design Views

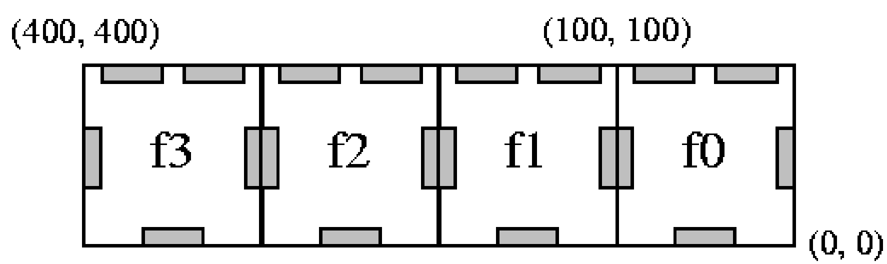
Behavior

```
Module add4 (s, c4, ci, a, b);
    input [3:0] a, b;
    input ci;
    output [3:0] s;
    output c4;
  assign {c4,s} = a + b + ci;
endmodule
```

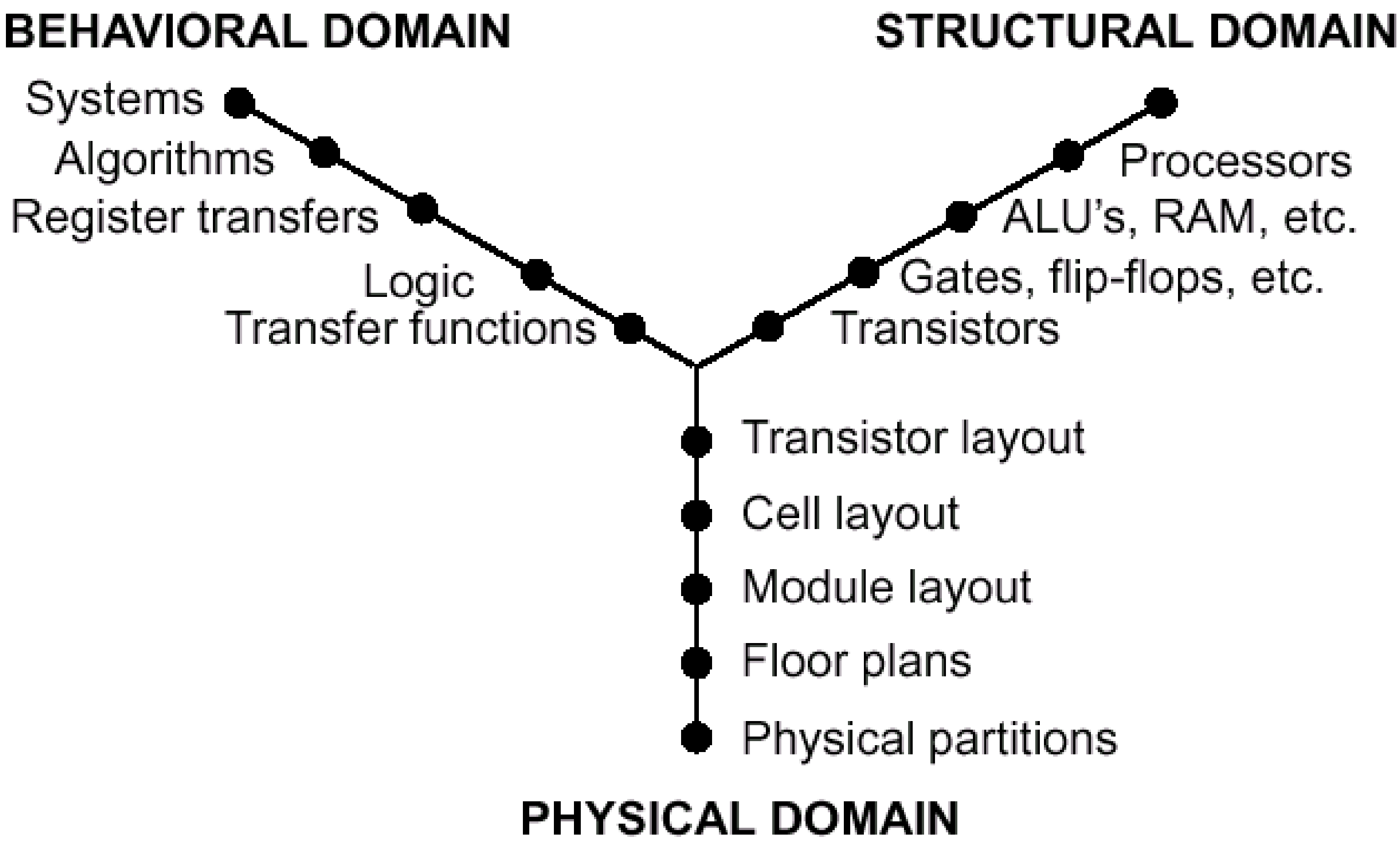
Structural



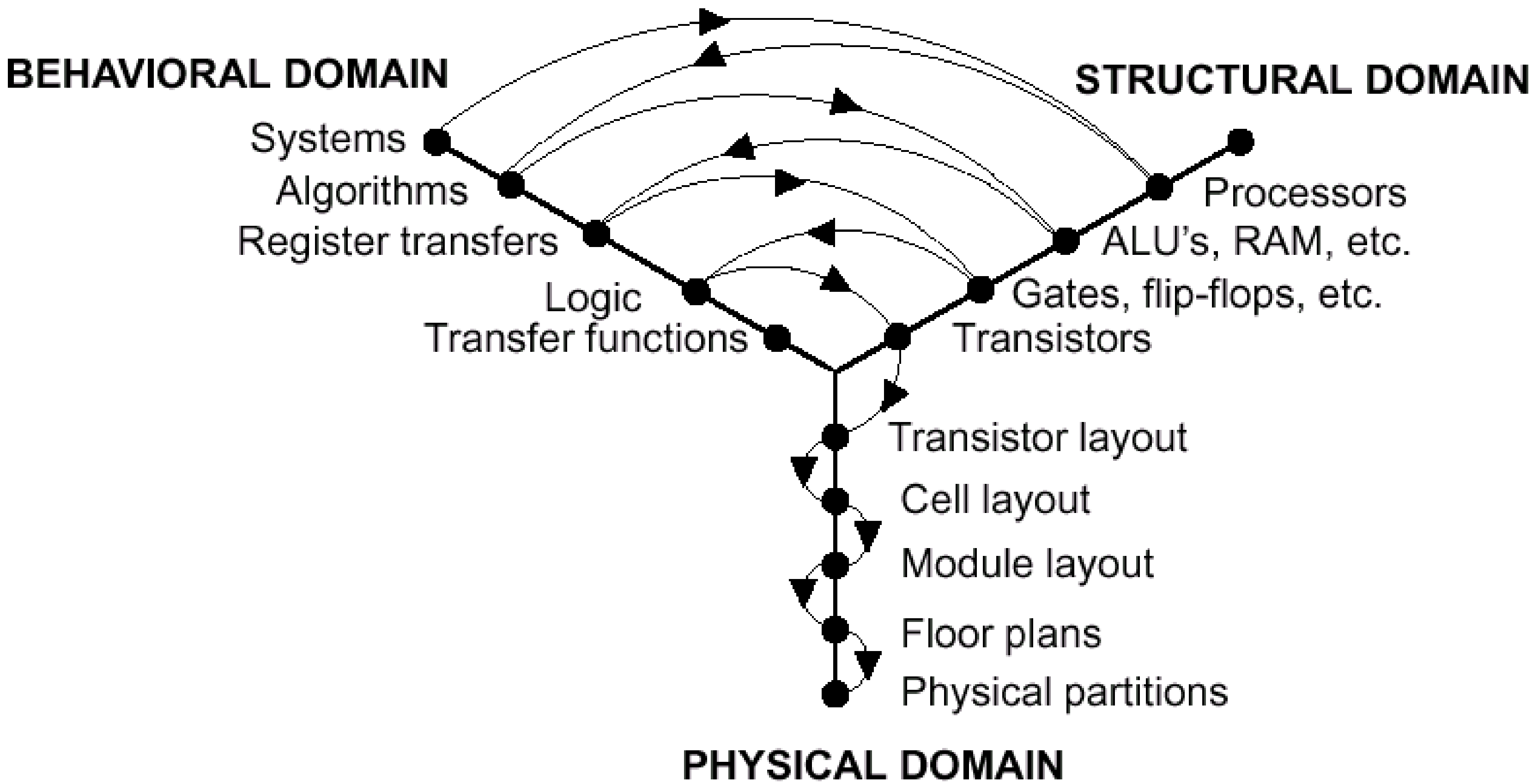
Physical



Gajski's Y-Chart

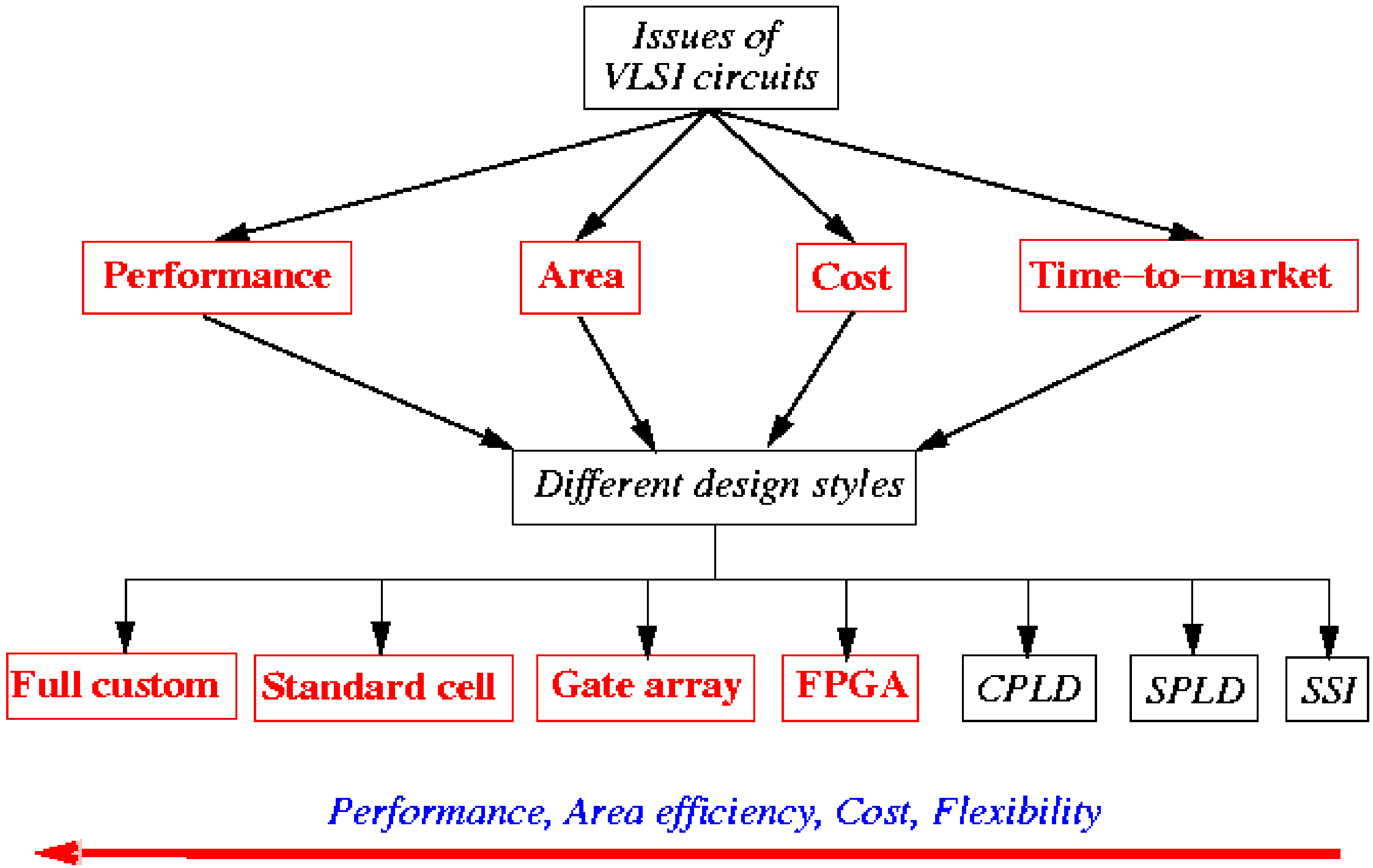


Top-Down Structural Design

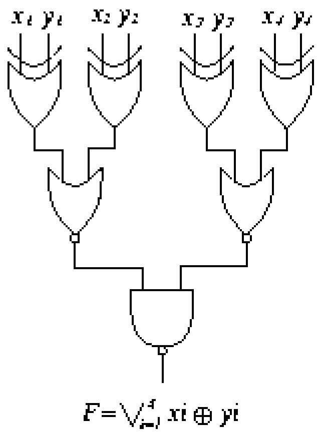


Design Styles

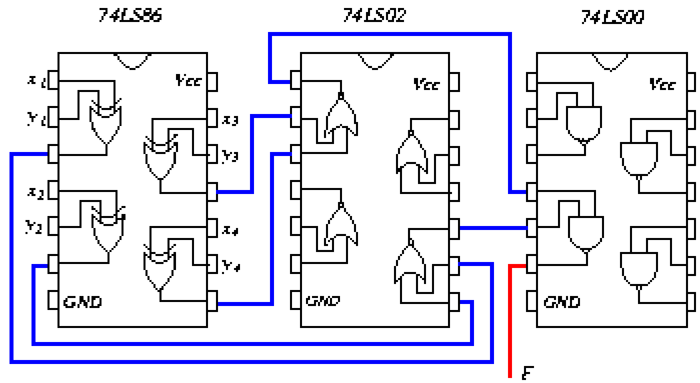
- Specific design styles shall require specific CAD tools



SSI/SPLD Design Style

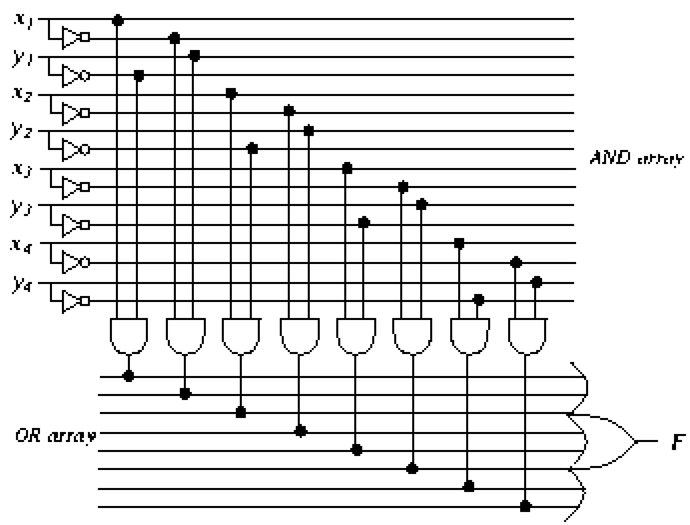


(a) 4–bit comparator.



(b) SSI implementation.

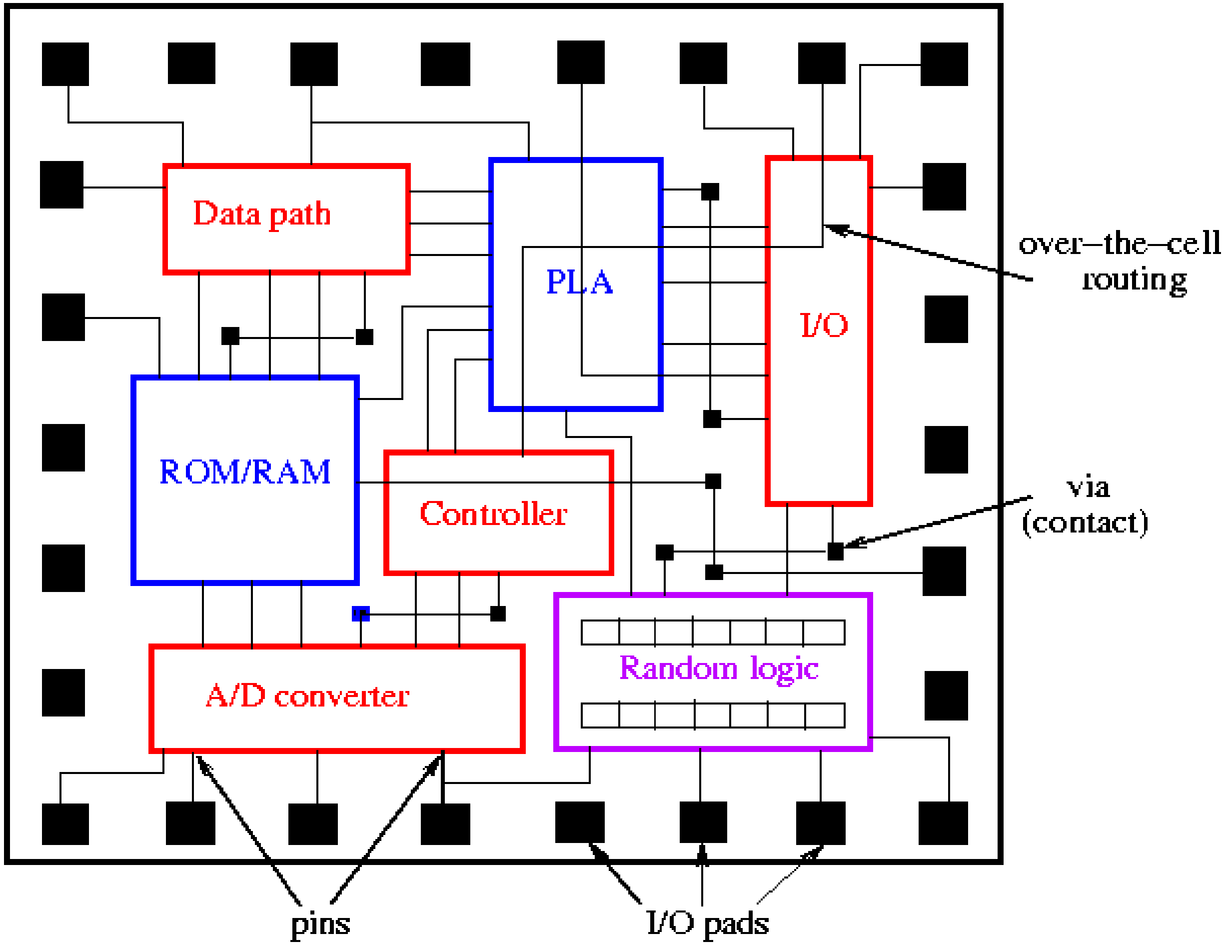
SSI: Small Scaled Integrated circuits



(c) SPLD (PLA) implementation.

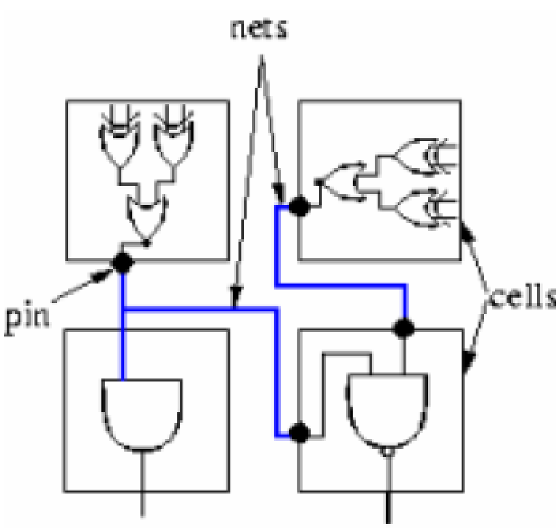
Full Custom Design Style

- Designers can control the shape of all mask patterns.
- Designers can specify the design up to the level of individual transistors.



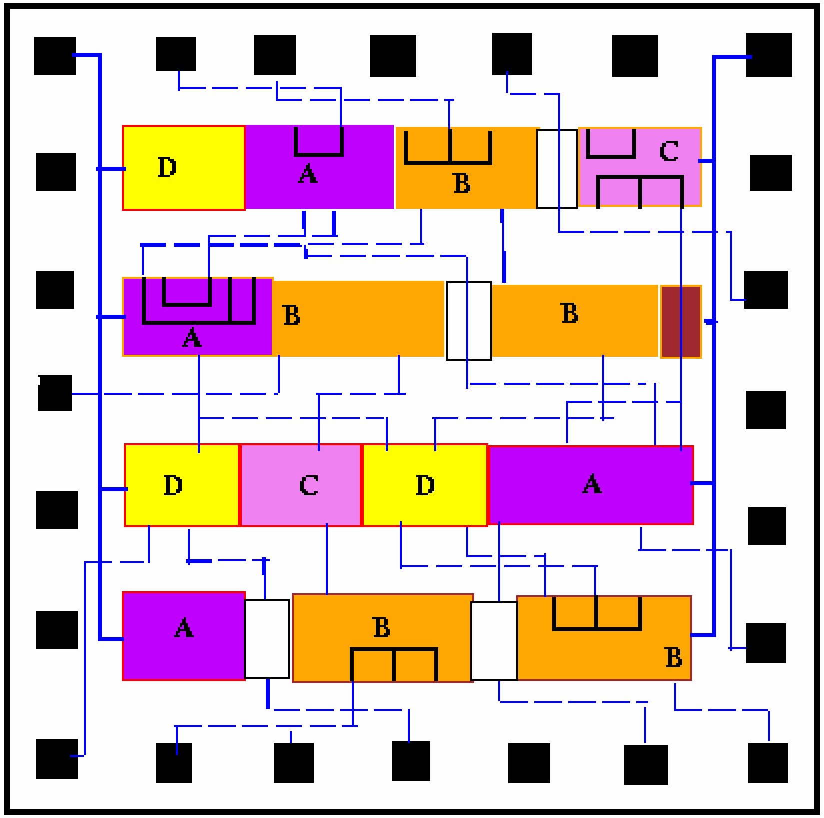
Terminology

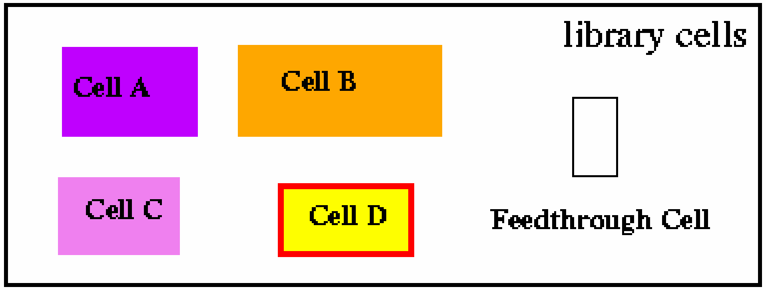
- **Cell**: a logic block used to build larger circuits.
- **Pin**: a wire (metal or polysilicon) to which another external wire can be connected.
- **Nets**: a collection of pins which must be electrically connected.
- **Netlist**: a list of all nets in a circuit.



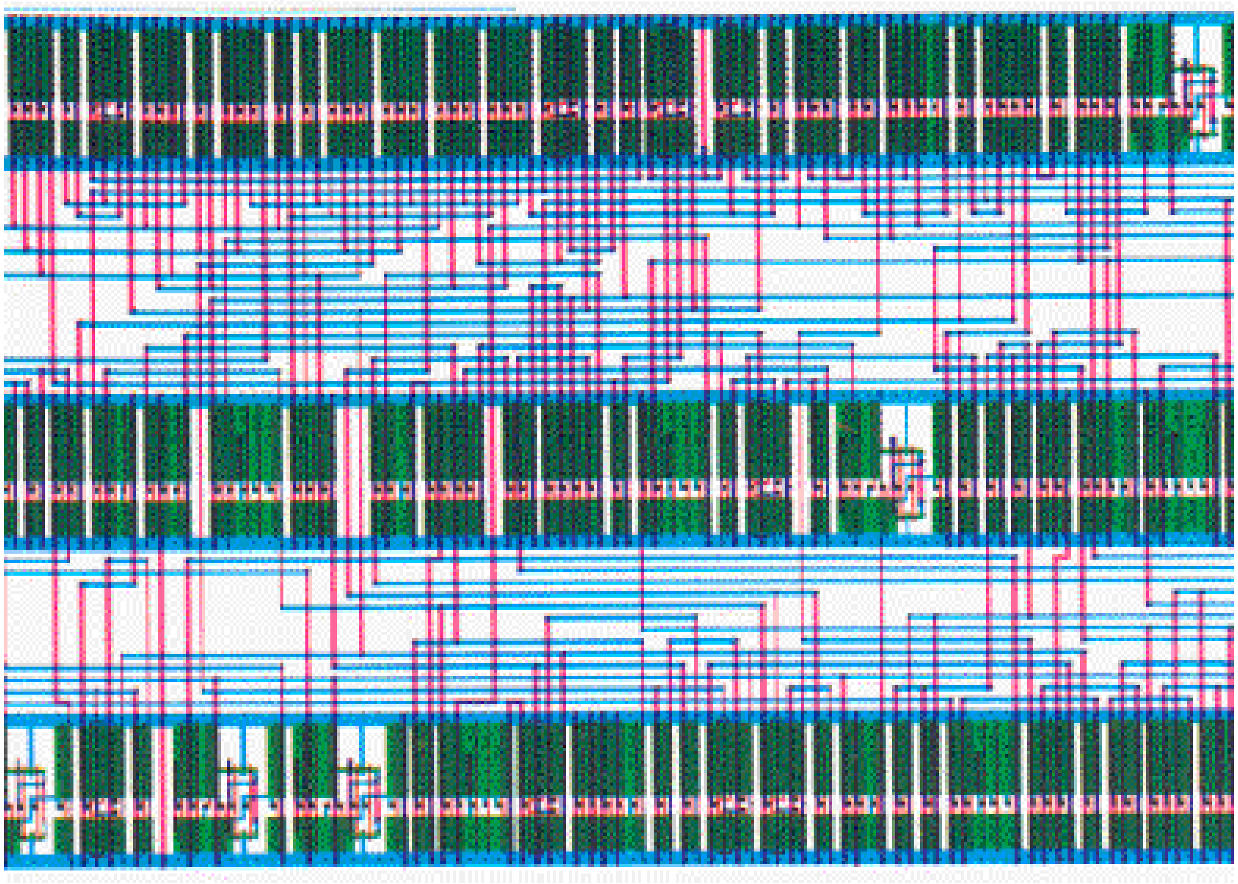
Standard Cell Design Style

- Select pre-designed cells (of same height) to implement logic
- Characterize and store cells in library





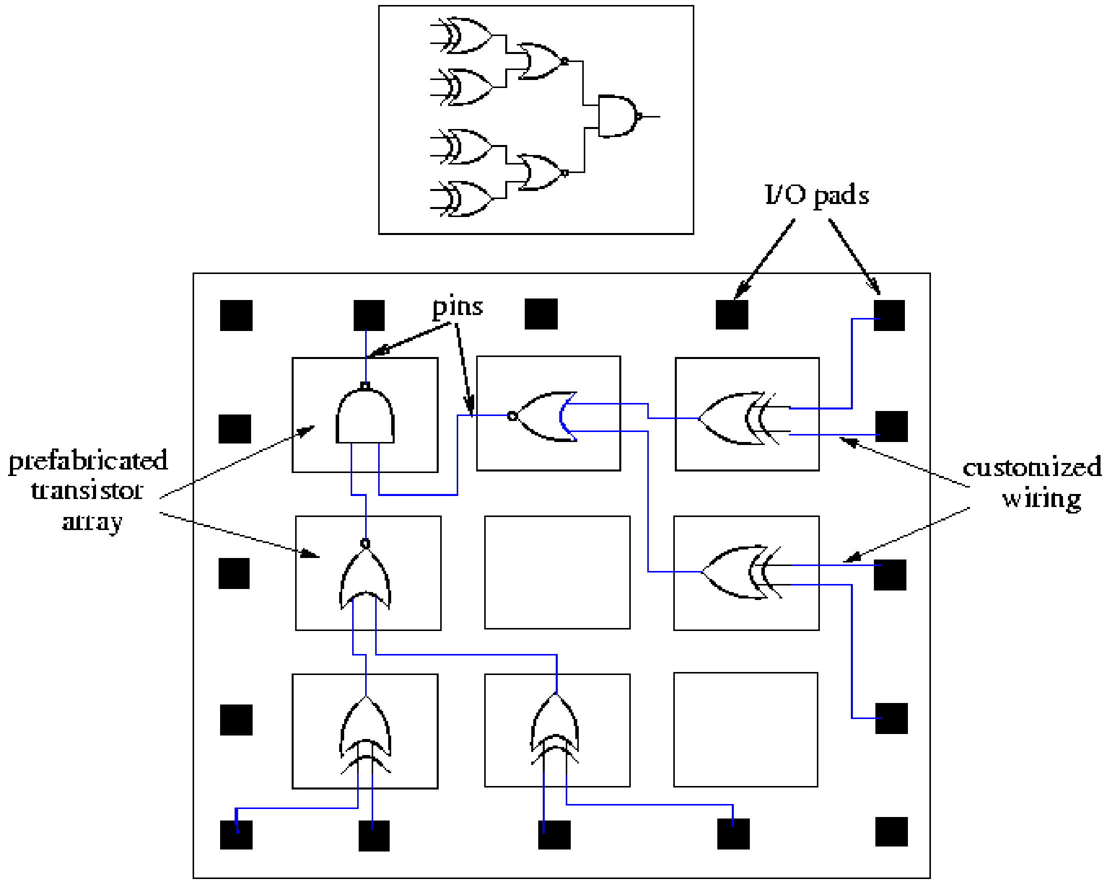
Standard Cell Example



Courtesy Newton/Pister, UC-Berkeley

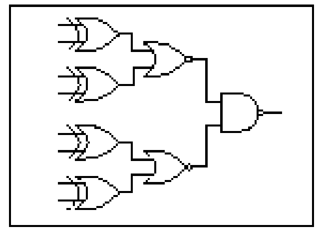
Gate Array Design Style

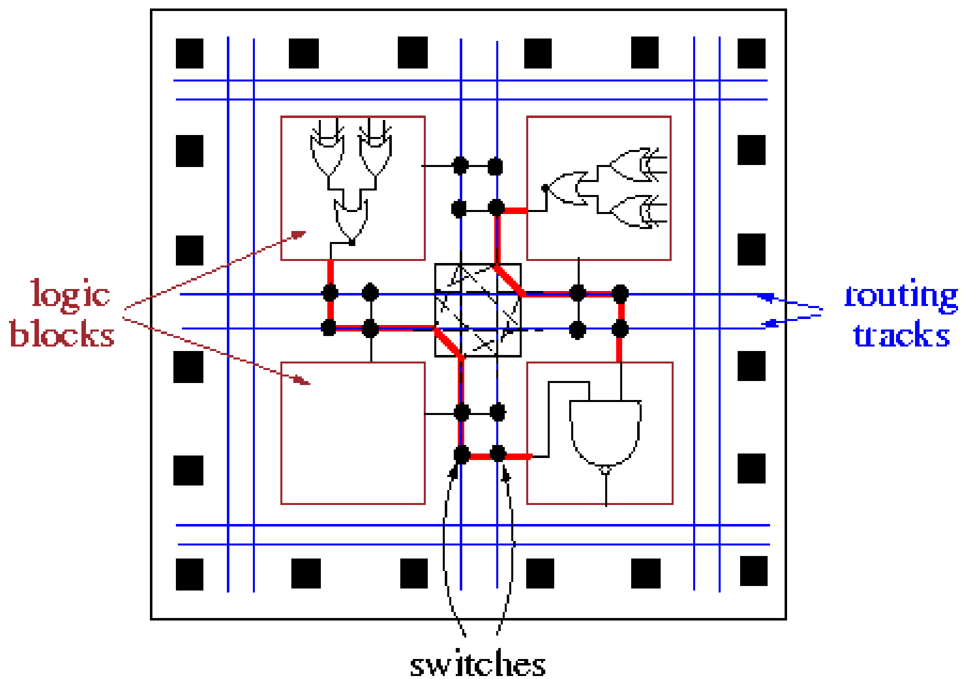
- Prefabricates a transistor array
- Needs wiring customization to implement logic



FPGA Design Style

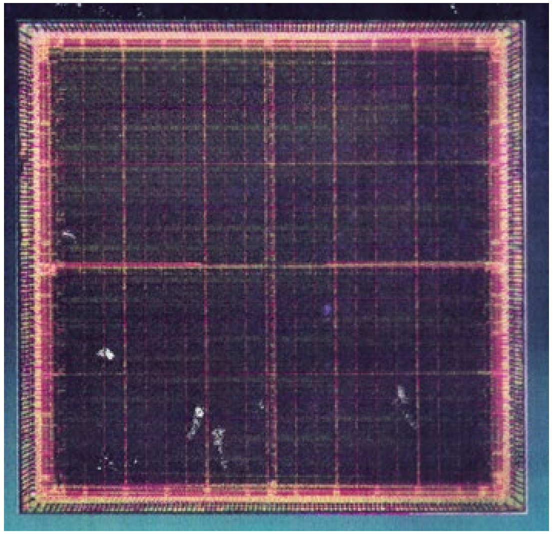
- Logic and interconnects are both prefabricated
- Illustrated by a symmetric array-based FPGA





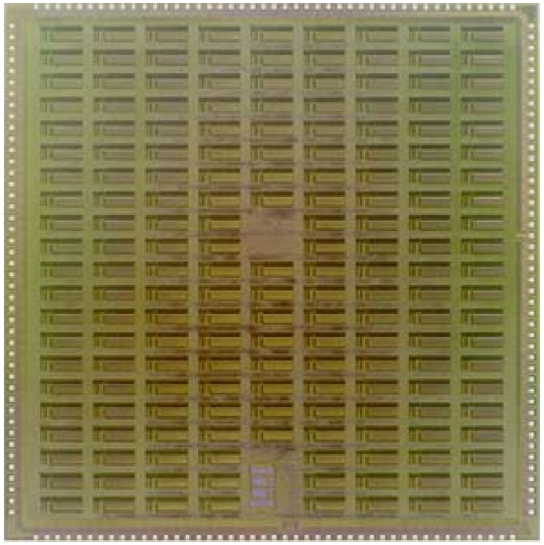
Prefabricated all chip components

Array-Based FPGA Example



Lucent Technologies 15K ORCA FPGA, 1995

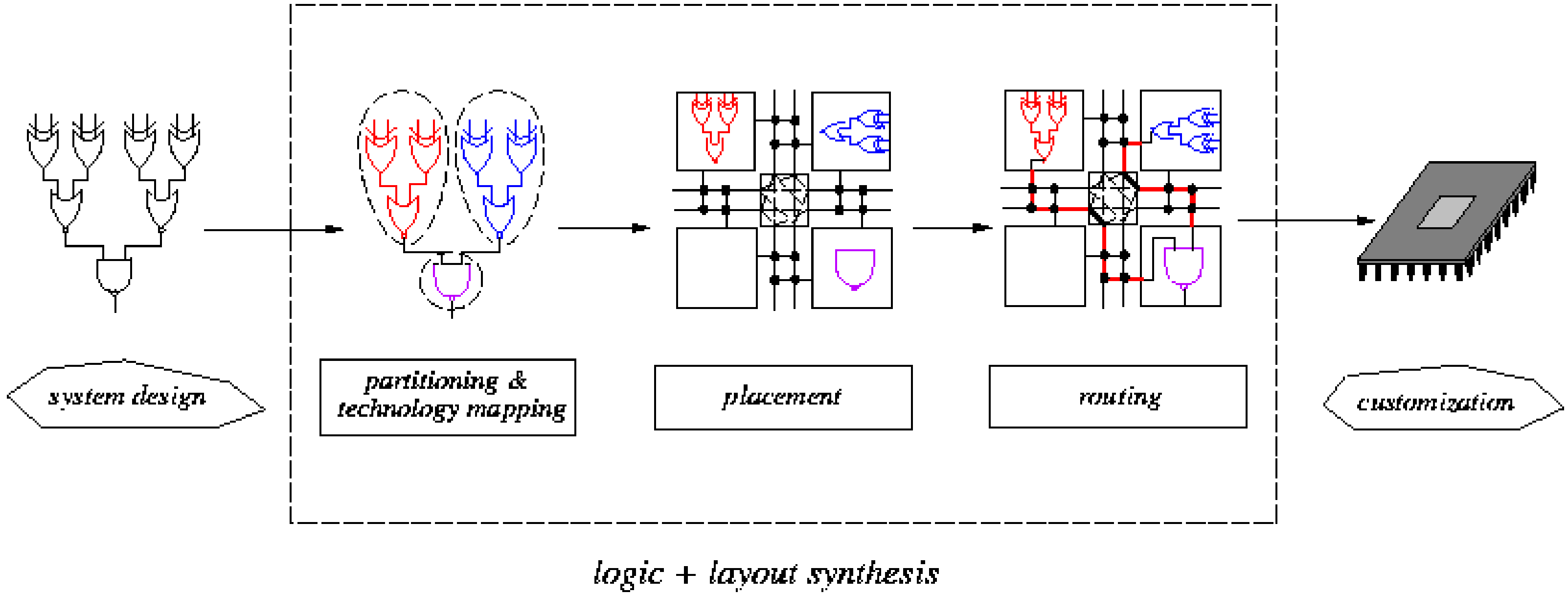
- 0.5 um 3LM CMOS
- 2.45 M Transistors
- 1600 Flip-flops
- 25K bit user RAM
- 320 I/Os



Fujitsu's non-volatile Dynamically Programmable Gate Array (DPGA), 2002

FPGA Design Process

- Illustrated by a symmetric array-based FPGA
- No fabrication is needed



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

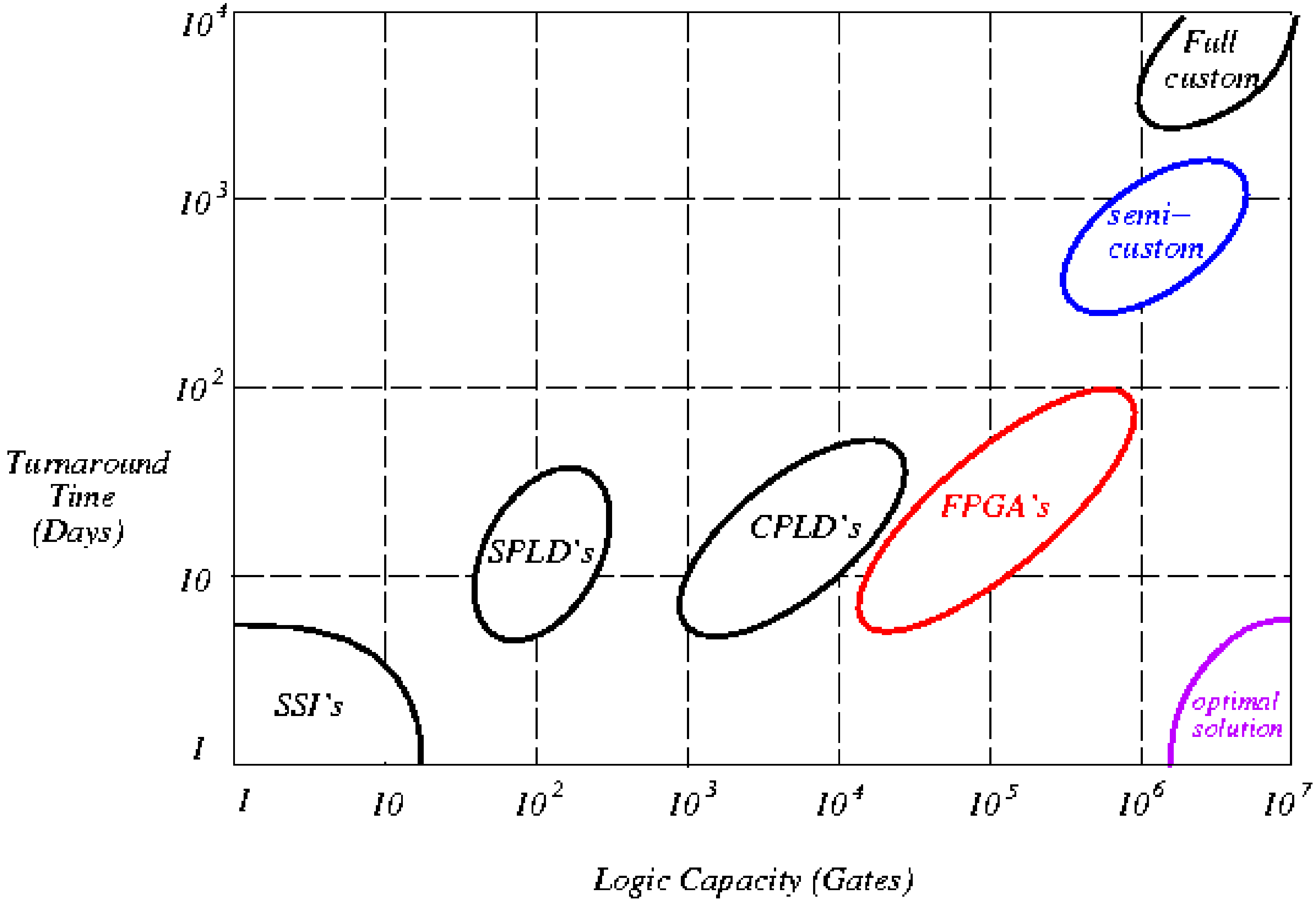
* Uneven height cells are also used.

Comparisons of Design Styles

	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

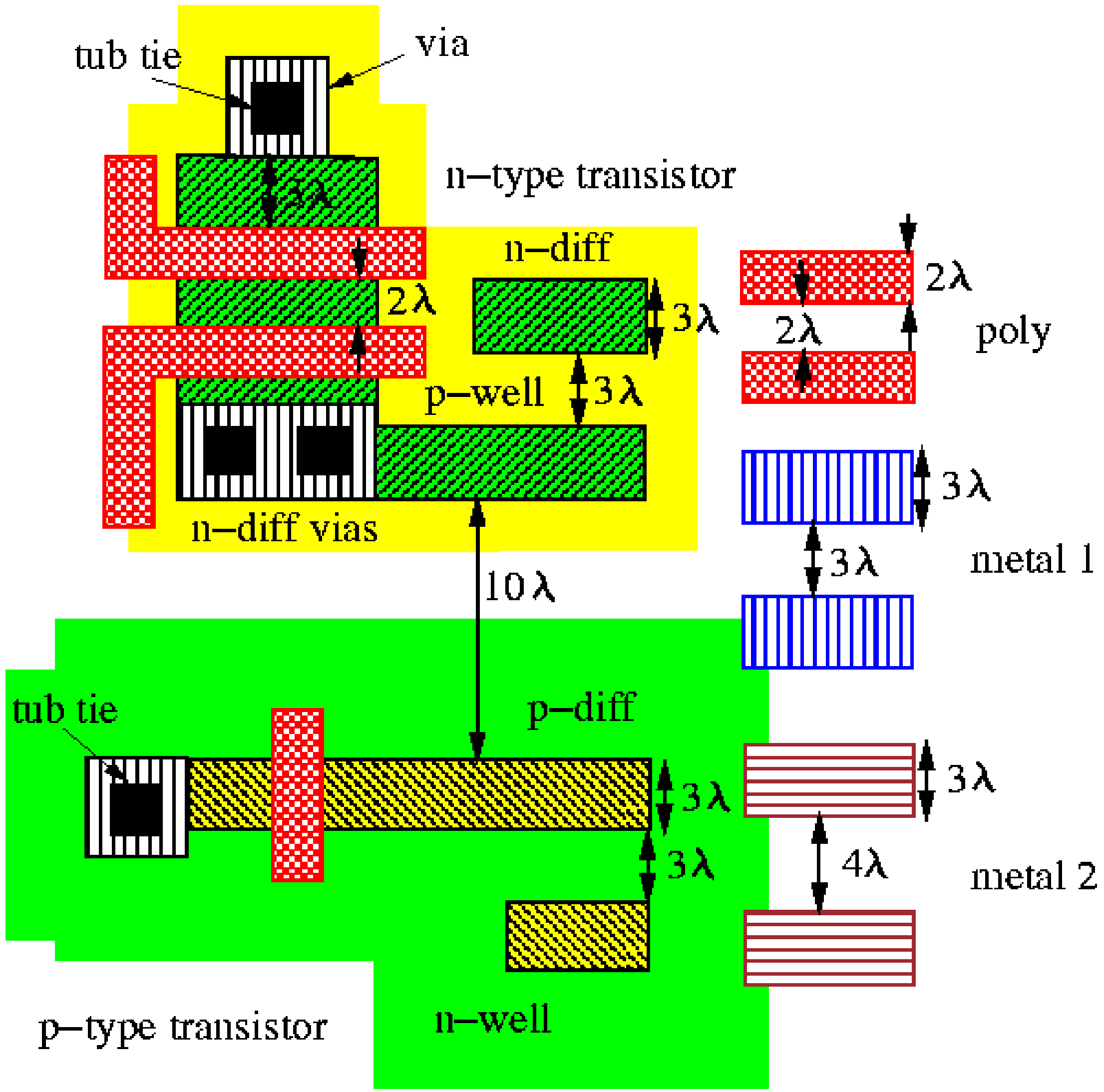
Design Style Trade-offs



Design Rules

- Layout rules are used for preparing the masks for fabrication.
- Fabrication processes have inherent limitations in accuracy.
- Design rules specify geometry of masks to optimize yield and reliability (trade-offs: area, yield, reliability).
- Three major rules:
  - **Wire width:** Minimum dimension associated with a given feature.
  - **Wire separation:** Allowable separation.
  - **Contact:** overlap rules.
- Two major approaches:
  - **“Micron” rules:** stated at micron resolution.
  - λ **rules:** simplified micron rules with limited **scaling** attributes.
- λ may be viewed as the size of minimum feature.
- Design rules represents a tolerance which insures very high probability of correct fabrication (not a hard boundary between correct and incorrect fabrication).
- Design rules are determined by experience.

Example: SCMOS Design Rules



MOSIS Layout Design Rules

- MOSIS design rules (SCMOS rules) are available at http://www.mosis.org.
- 3 basic design rules: Wire width, wire separation, contact rule.
- MOSIS design rule examples

R1	Min active area width	3 λ
R3	Min poly width	2 λ
R4	Min poly spacing	2 λ
R5	Min gate extension of poly over active	2 λ
R8	Min metal width	3 λ
R9	Min metal spacing	3 λ
R10	Poly contact size	2 λ
R11	Min poly contact spacing	2 λ