

Introduction to Electronic Design Automation



Jie-Hong Roland Jiang
江介宏

Department of Electrical Engineering
National Taiwan University



Spring 2014

1

Design Automation?



2

Course Info (1/4)

Instructor

Jie-Hong R. Jiang
email: jhjiang@cc.ee.ntu.edu.tw
office: 242, EE2 Building
phone: (02)3366-3685
office hour: 15:00-17:00 Fridays

TA

Chi-Chuan Chuang; Yi-Hsiang Lai
email: chichuan327@gmail.com; r02943086@ntu.edu.tw
phone: (02)3366-9753
office: 526, Ming-Dar Hall
office hour: TBA

Email contact list

NTU email addresses of enrolled students will be used for future contact

Course webpage

<http://cc.ee.ntu.edu.tw/~jhjiang/instruction/courses/spring14-eda/eda.html>
please look up the webpage frequently to keep updated

3

Course Info (2/4)

Grading rules (raw score)

- Homework 40%
- Midterm 25%
- Final Quiz 10%
- Project 25%

(Note that the final grade is based on grading on a curve.)

Homework

- discussions encouraged, but solutions should be written down individually and separately
- 4 assignments in total
- late homework (20% off per day)

Midterm exam/final quiz

- in-class exam

Project

- Team or individual work on selected topics (CAD Contest problems / paper reading / implementation / problem solving, etc.)

Academic integrity: no plagiarism allowed

4

Course Info (3/4)

- Prerequisite
 - Switching circuits and logic design, or by instructor's consent
- Main lecture basis
 - Lecture slides and/or handouts
- Textbook
 - Y.-W. Chang, K.-T. Cheng, and L.-T. Wang (Editors). *Electronic Design Automation: Synthesis, Verification, and Test*. Elsevier, 2009.
- Reference
 - S. H. Gerez. *Algorithms for VLSI Design Automation*. John Wiley & Sons, 1999.

5

Course Info (4/4)

- Objectives:
 - Peep into EDA
 - Motivate interest
 - Learn problem formulation and solving
 - Have fun!

6

FYI

❑ 2014 CAD Contest will be announced soon

- An international event
- Program submission deadline in Sep. 2014
- Award ceremony in ICCAD, Nov. 2014

❑ Previous CAD Contests

- http://cad_contest.cs.nctu.edu.tw/cad13/
- http://cad_contest.cs.nctu.edu.tw/cad12/

7

FAQ

- ❑ What's EDA?
 - What are we concerned about?
 - What's unique in EDA compared to other EE/CS disciplines?
- ❑ What time is good to take *Intro to EDA*?
 - Am I qualified? Do I have enough backgrounds?
- ❑ How's the loading?
 - Program to death!?
- ❑ What kind of skills and domain knowledge can I learn? Other applications?
- ❑ What are the career opportunities?
- ❑ Yet another question?

8

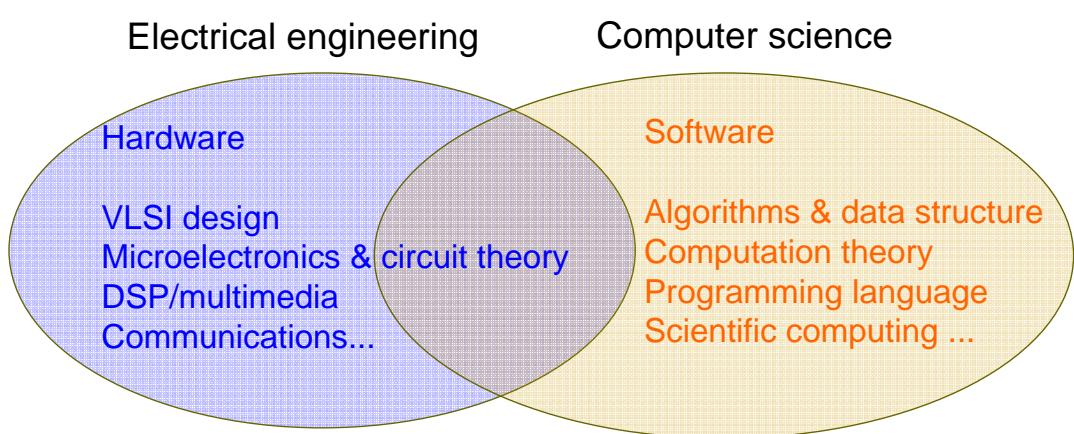
Course Outline

- ❑ Introduction
- ❑ Computation in a nutshell
- ❑ High-level synthesis
- ❑ Logic synthesis
- ❑ Formal verification
- ❑ Physical design
- ❑ Testing
- ❑ Advanced topics

9

Introduction

- ❑ EDA, where HW and SW meet each other



10

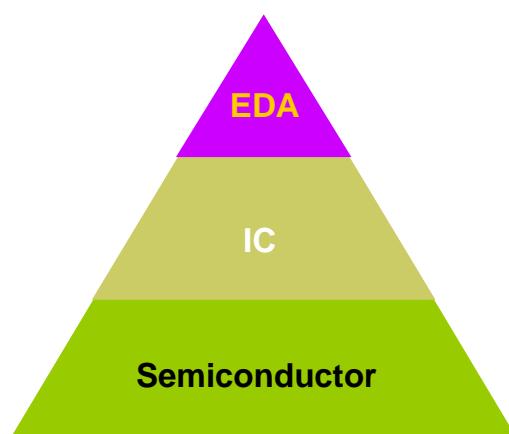
Introduction

- ❑ EDA is concerned about HW/SW design in terms of
 - Correctness
 - Productivity
 - Optimality
 - Scalability

11

Introduction

- ❑ EDA (in a strict sense) and industries
 - Impact - solving a problem may benefit vast electronic designs



12

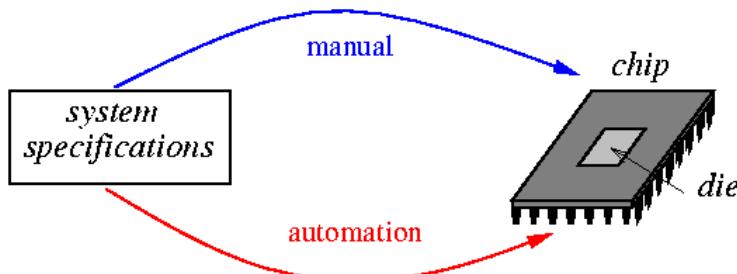
Introduction

□ Today's contents:

- Introduction to VLSI design flow, methodologies, and styles
- Introduction to VLSI design automation tools
- Semiconductor technology roadmap
- CMOS technology

□ Reading:

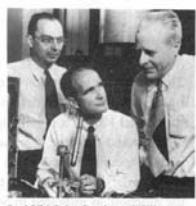
- Chapters 1, 2



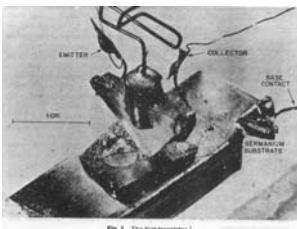
13

Milestones of IC Industry

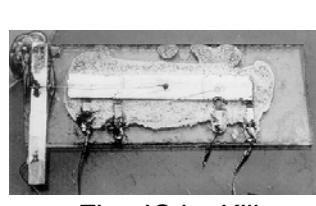
- **1947:** Bardeen, Brattain & Shockley invented the transistor, foundation of the IC industry.
- **1952:** SONY introduced the first transistor-based radio.
- **1958:** Kilby invented integrated circuits (ICs).
- **1965:** Moore's law.
- **1968:** Noyce and Moore founded Intel.
- **1970:** Intel introduced 1 K DRAM.



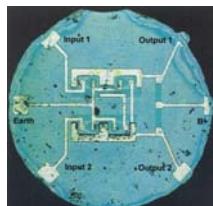
In 1956 John Bardeen, William Shockley and Walter Brattain shared the Nobel Prize in Physics for their discovery of the transistor.



First transistor



First IC by Kilby



First IC by Noyce

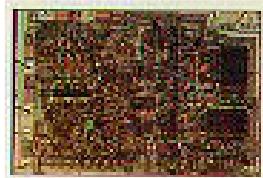
14

Milestones of IC Industry

- **1971:** Intel announced 4-bit 4004 microprocessors (2250 transistors).
- **1976/81:** Apple II/IBM PC.
- **1985:** Intel began focusing on microprocessor products.
- **1987:** TSMC was founded (**fabless** IC design).
- **1991:** ARM introduced its first embeddable RISC IP core (**chipless** IC design).



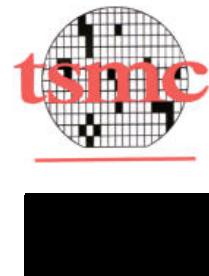
Intel founders



4004



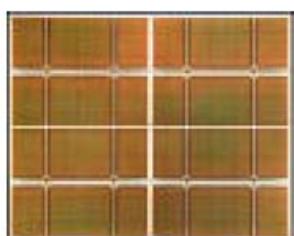
IBM PC



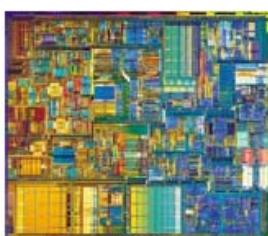
15

Milestones of IC Industry

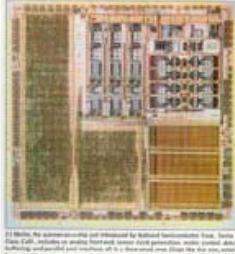
- **1996:** Samsung introduced 1G DRAM.
- **1998:** IBM announces 1GHz experimental microprocessor.
- **1999/earlier:** **System-on-Chip (SoC)** methodology applications.
- **2002/earlier:** **System-in-Package (SiP)** technology
- An Intel P4 processor contains 42 million transistors (1 billion by 2005)
- Today, we produce > 1 billion transistors per person.



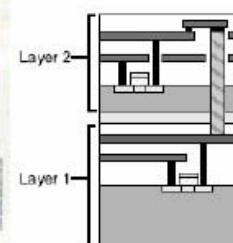
4GB DRAM (2001)



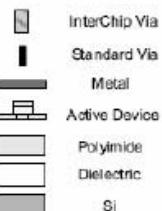
Pentium 4



Scanner-on-chip

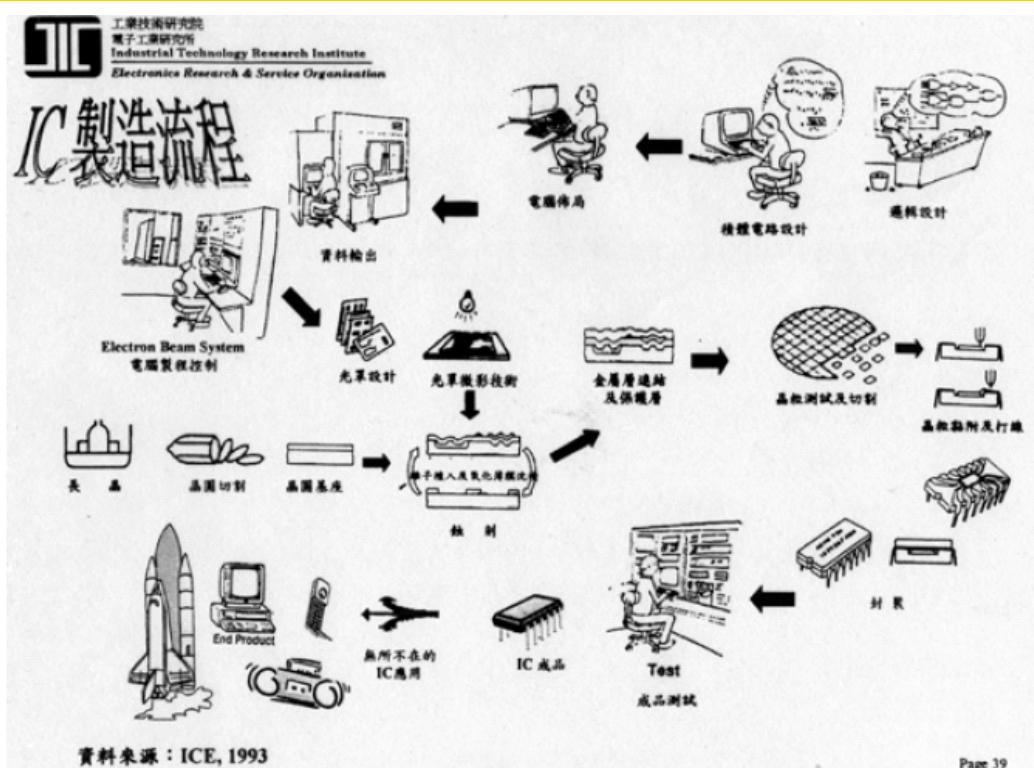


System in Package (SiP)



16

IC Design & Manufacturing Process



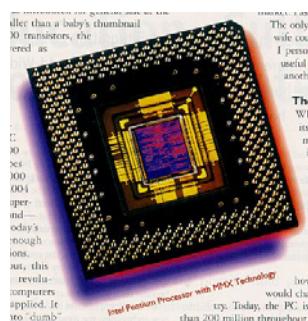
Page 39

17

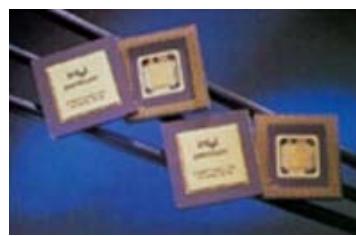
From Wafer to Chip



Figure 9-34
Attachment of leads from the Al pads on the periphery of the chip to posts on the package. (Photograph courtesy of



Mr. Cook's move would greatly reduce the cost of making chips.



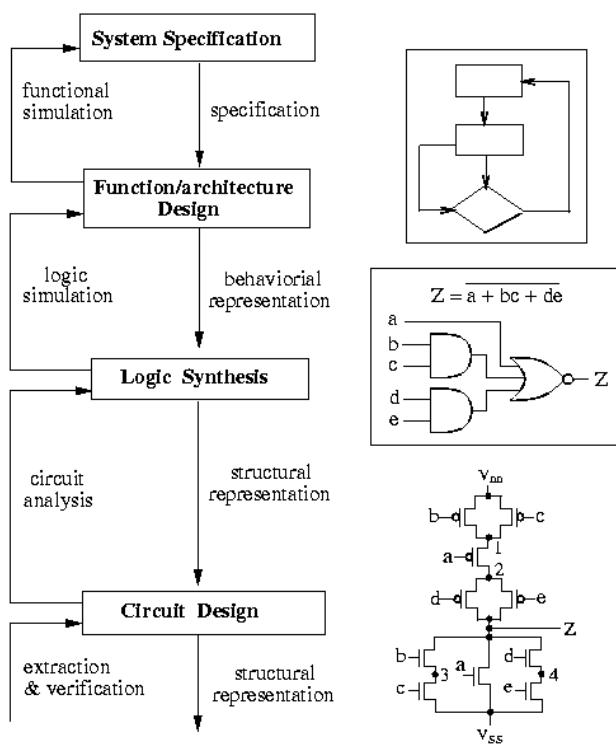
Standard VLSI Design Cycles

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.

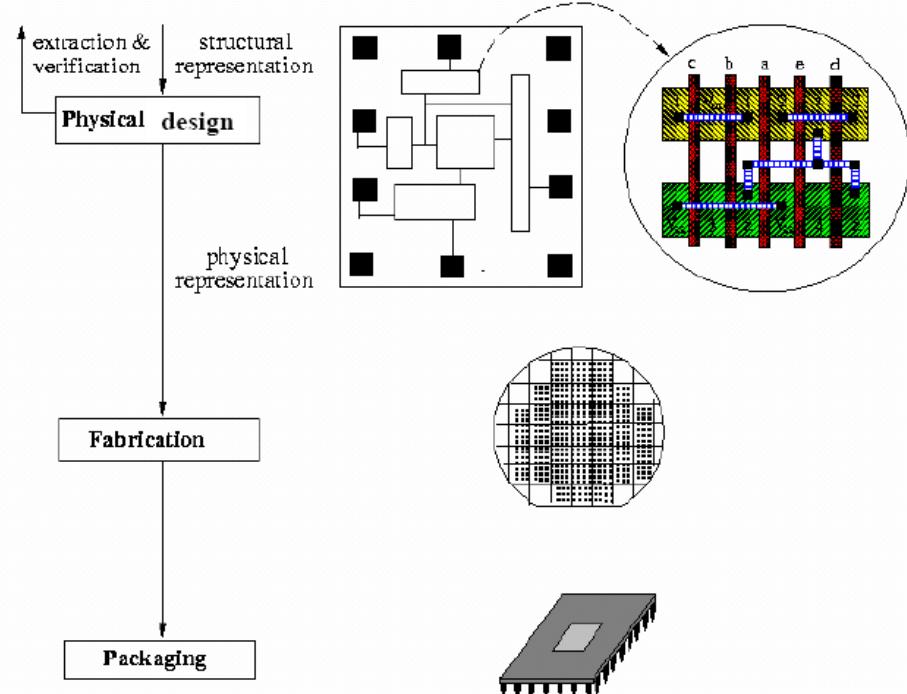
19

VLSI Design Flow



20

VLSI Design Flow



21

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., function, 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).

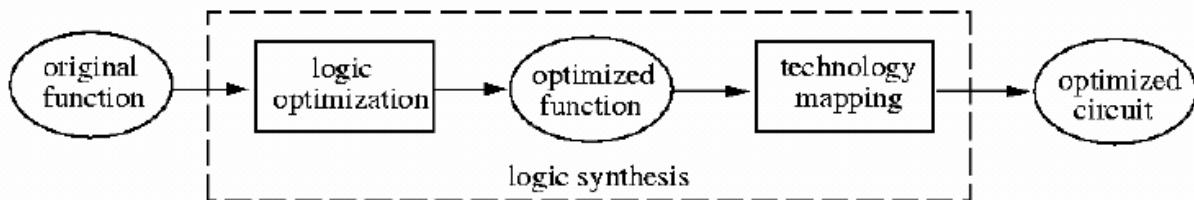
22

Design Issues and Tools

- ❑ System-level design
 - Partitioning into hardware and software, co-design/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:
 - Register-transfer level and logic synthesis
 - Gate-level simulation (functionality, power, etc)
 - Timing analysis
 - Formal verification

23

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.

24

Logic Optimization Examples

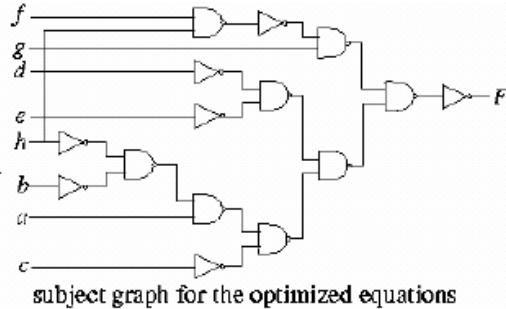
- **Two-level:** minimize the # of product terms.

- $F = \bar{x}_1 \bar{x}_2 \bar{x}_3 + \bar{x}_1 \bar{x}_2 x_3 + x_1 \bar{x}_2 \bar{x}_3 + x_1 \bar{x}_2 x_3 + x_1 x_2 \bar{x}_3 \Rightarrow F = \bar{x}_2 + x_1 \bar{x}_3.$

- **Multi-level:** minimize the #'s of literals, variables.

- E.g., equations are optimized using a smaller number of literals.

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



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

25

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-button* operation

26

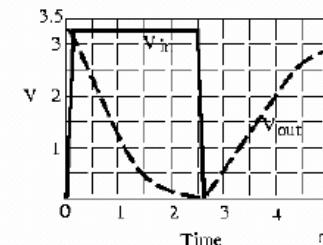
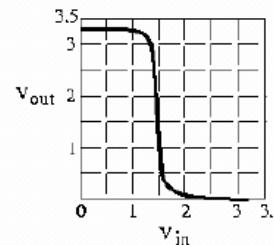
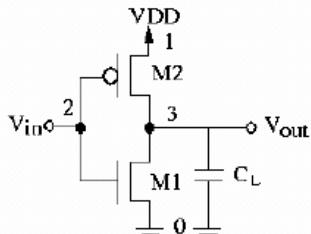
Circuit Simulation

```
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)
```

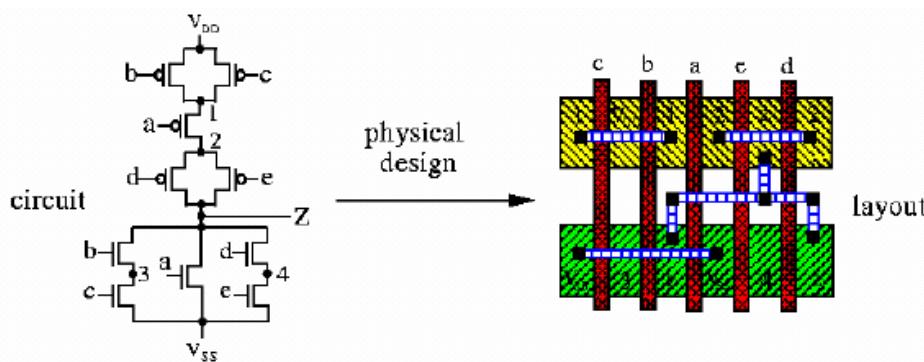
```
.LIB '../mod_06' typical
```

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



27

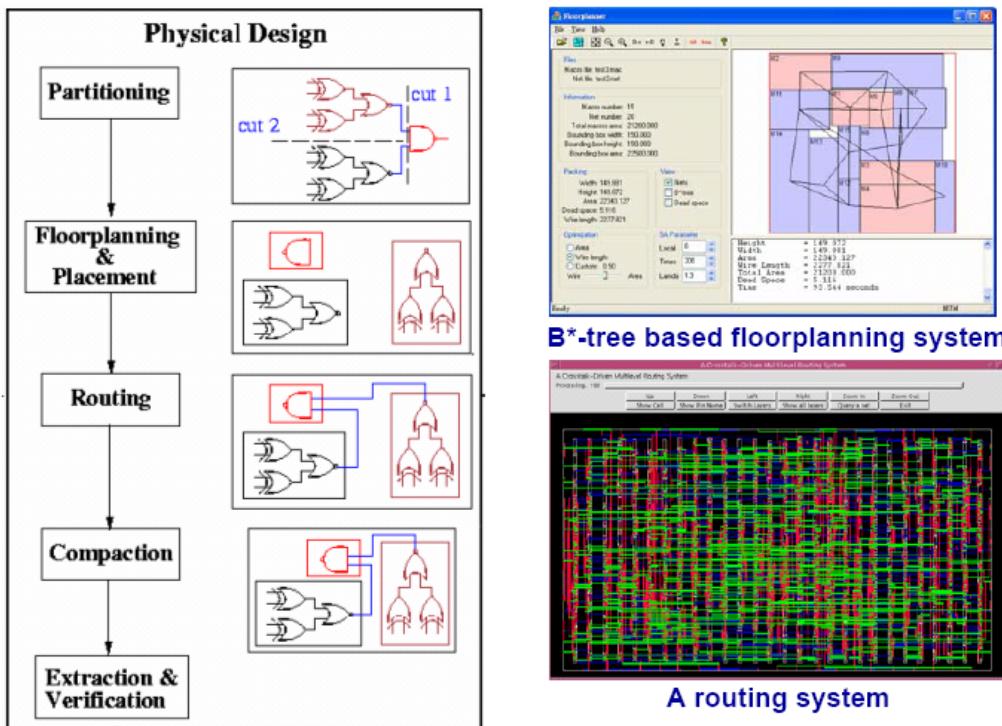
Physical Design



- ❑ 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

28

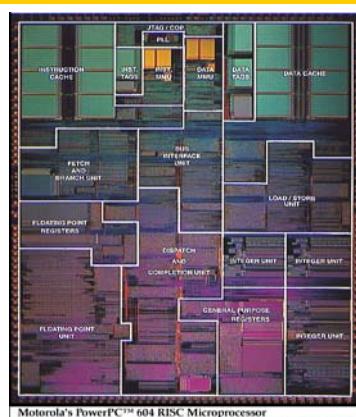
Physical Design Flow



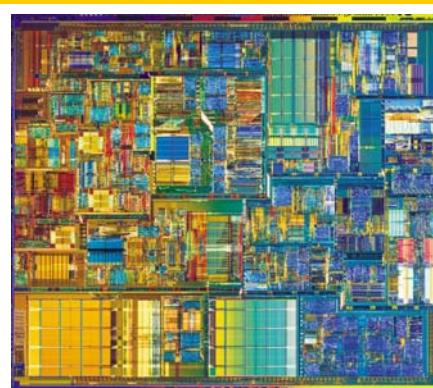
29

Floorplan Examples

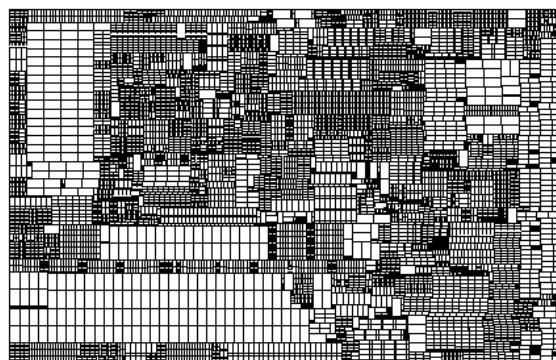
PowerPC 604



Pentium 4



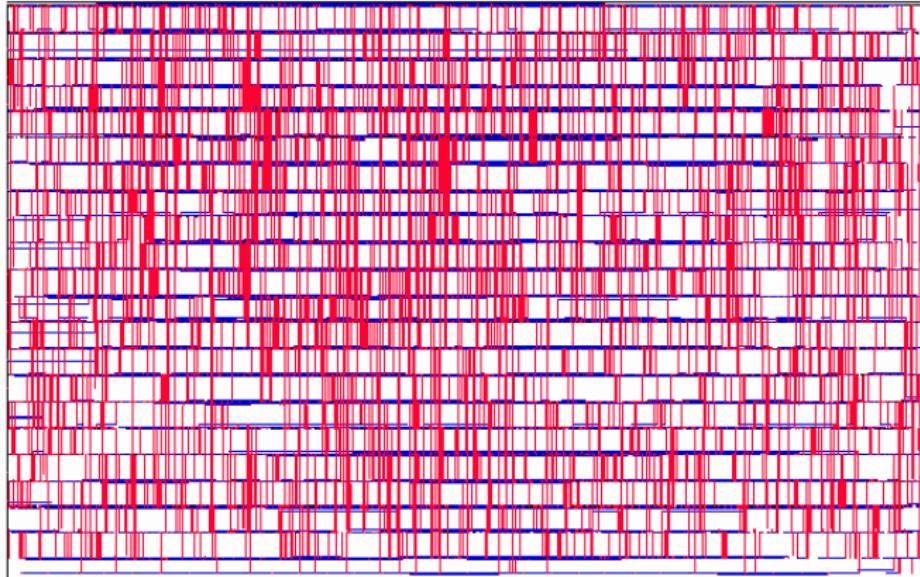
A floorplan with 9800 blocks



30

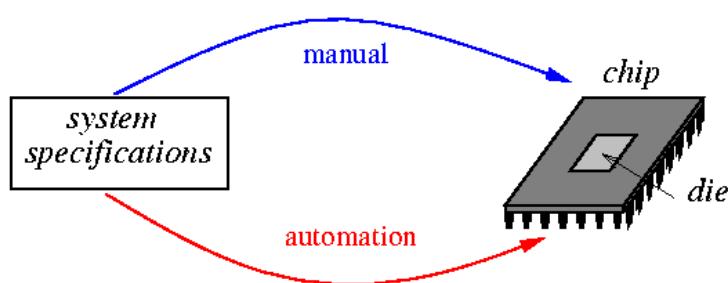
Routing Example

- 0.18um technology, two layers, pitch = 1 um, 8109 nets



31

IC Design Considerations

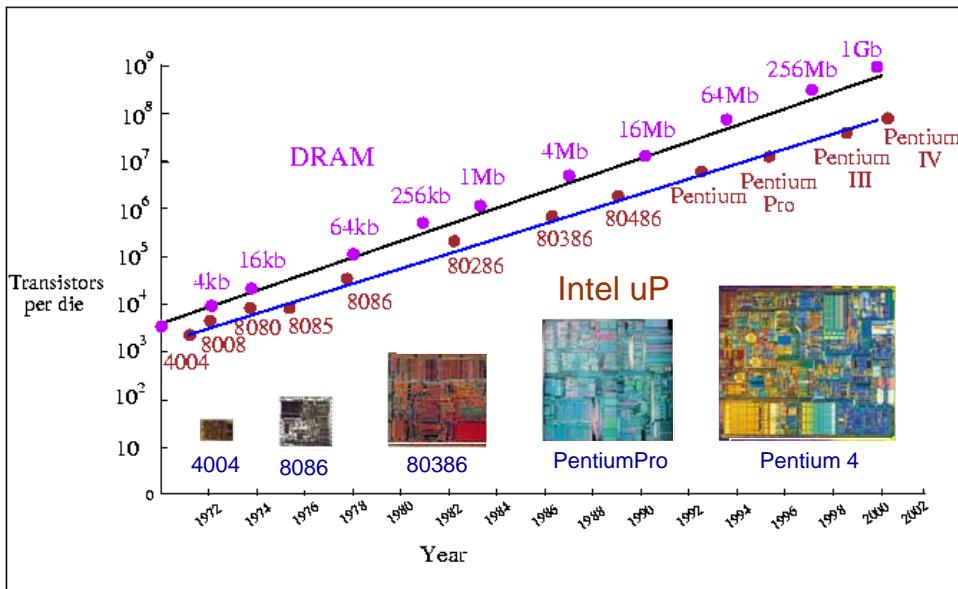


- Several conflicting considerations:
 - Design complexity:** large number of devices/transistors
 - Performance:** optimization requirements for high performance
 - Time-to-market:** about a 15% gain for early birds
 - Cost:** die **area**, packaging, testing, etc.
 - Others: power, signal integrity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

32

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)



33

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 (x10 ⁻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, Nov, 2002. <http://www.itrs.net/ntrs/publntrs.nsf>
- Deep submicron technology: node (**feature size**) < 0.25 μm
- Nanometer Technology: node < 0.1 μm

34

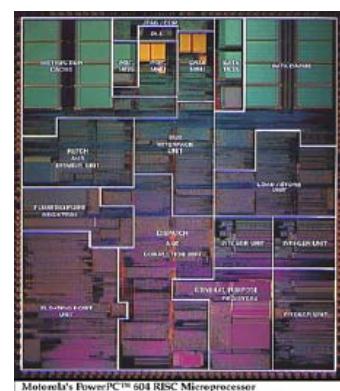
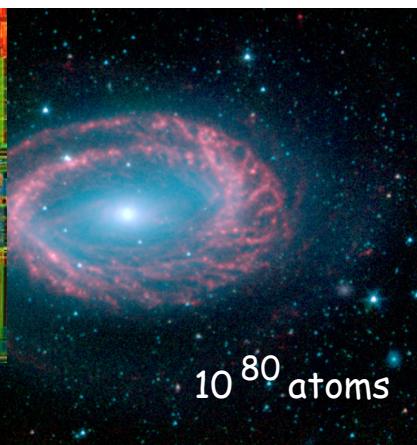
Nanometer Design Challenges

- ❑ In 2005, feature size $\approx 0.1 \mu m$, μP frequency ≈ 3.5 GHz, die size $\approx 520 \text{ mm}^2$, μP transistor count per chip $\approx 200M$, wiring level ≈ 8 layers, supply voltage ≈ 1 V, power consumption ≈ 160 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?

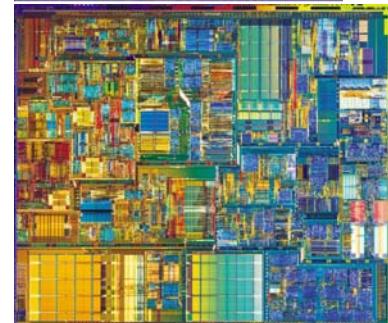
35

Design Complexity Challenges

- ❑ 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



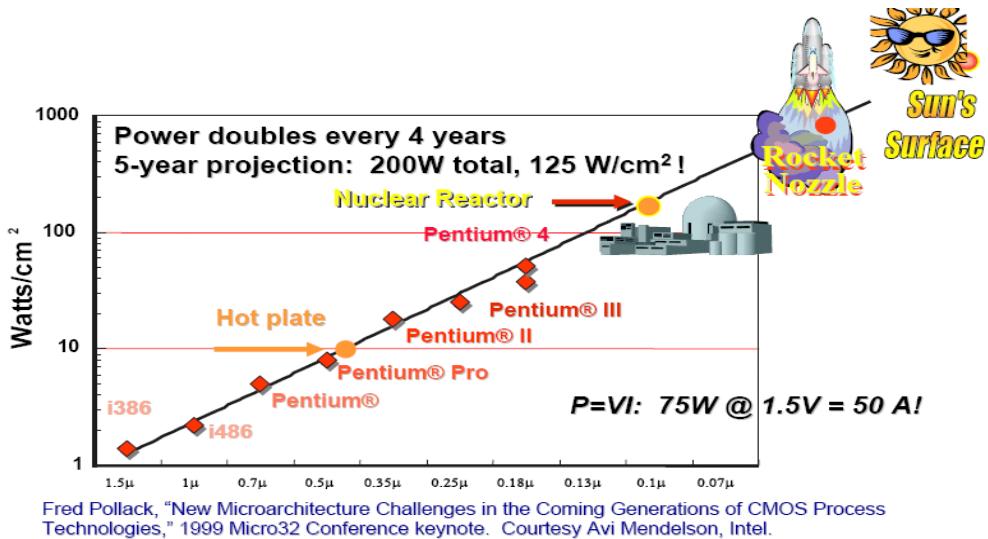
PowerPC 604



36

Power Dissipation Challenges

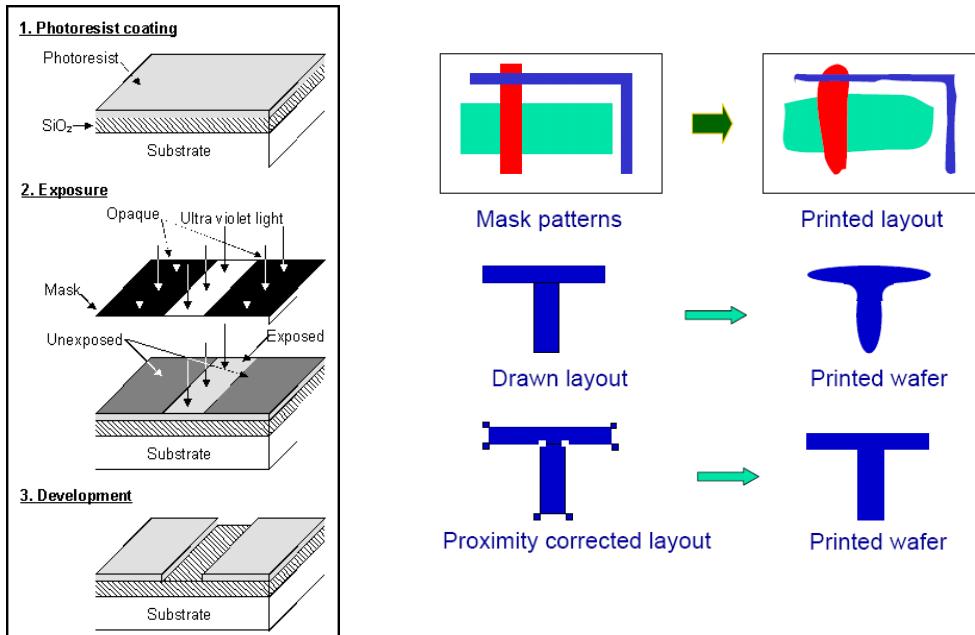
- Power density increases exponentially!



37

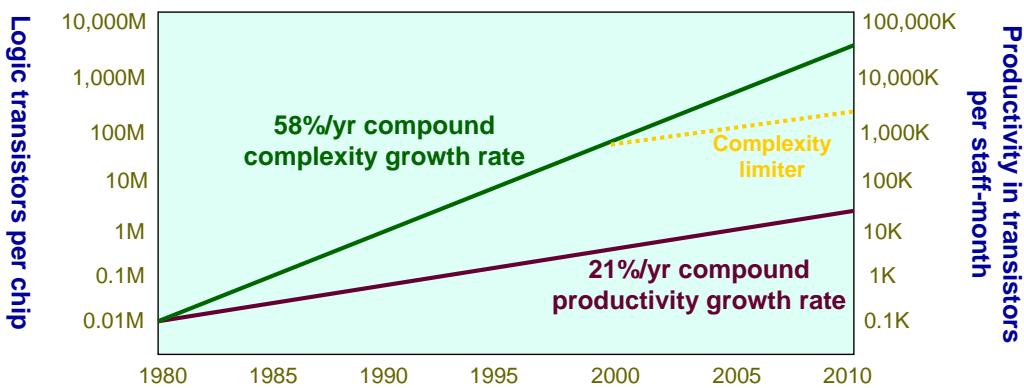
Semiconductor Fabrication Challenges

- Feature-size shrinking approaches physical limitation



38

Design Productivity Challenges

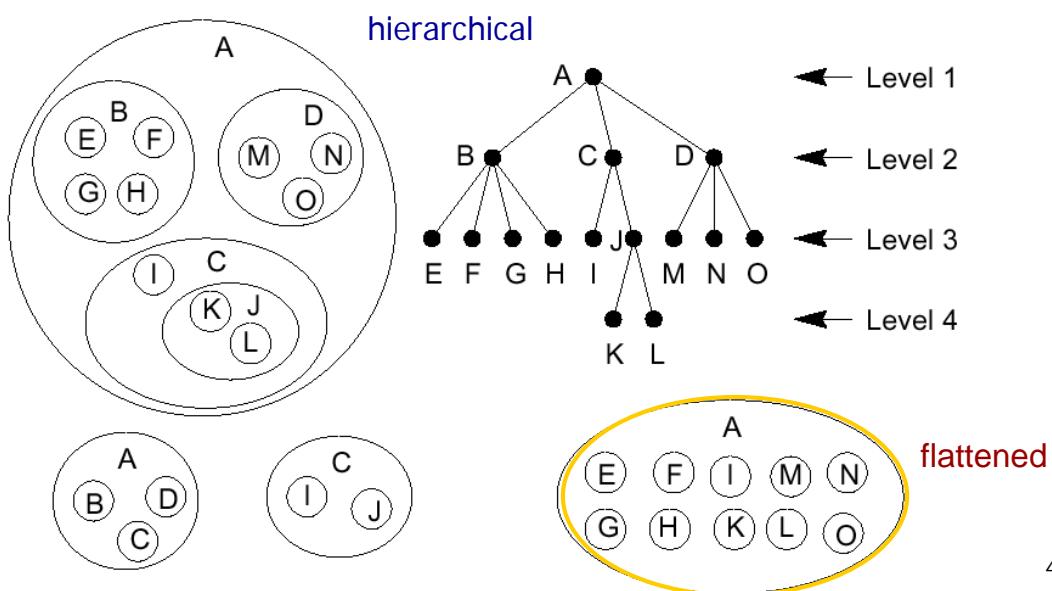


- ❑ Human factors may limit design more than technology
- ❑ Keys to solve the productivity crisis: **hierarchical design, abstraction, CAD (tool & methodology)**, IP reuse, etc.

39

Cope with Complexity

- ❑ **Hierarchical design**
 - Design cannot be done in one step \Rightarrow partition the design hierarchically
 - *Hierarchy*: something is composed of simpler things

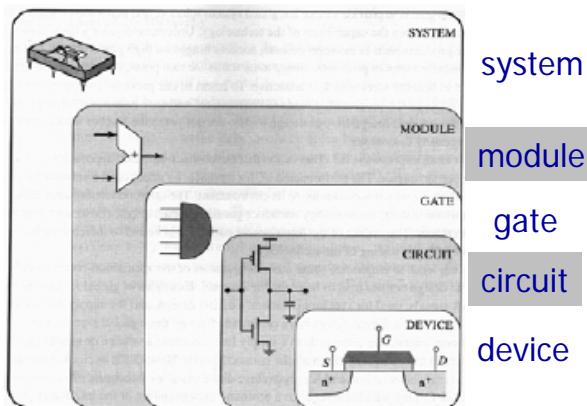


40

Cope with Complexity

❑ Abstraction

- Trim away unnecessarily detailed info at proper abstract levels
- Design domains:
 - ❑ *Behavioral*: black box view
 - ❑ *Structural*: interconnection of subblocks
 - ❑ *Physical*: layout properties
 - ❑ Each design domain has its own hierarchy



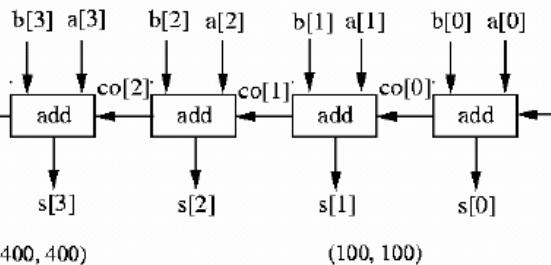
41

Three Design Views

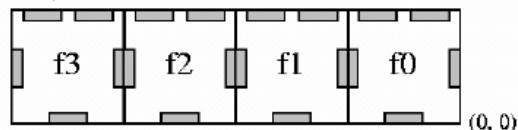
Behavior

```
module add4 (s, c4, ci, a, b);
  input [3:0] a, b;
  input ci;
  output [3:0] s;
  output c4;
  wire [2:0] co;
  add f0 (co[0], s[0], a[0], b[0], ci);
  add f1 (co[1], s[1], a[1], b[1], co[0]);
  add f2 (co[2], s[2], a[2], b[2], co[1]);
  add f3 (c4, s[3], a[3], b[3], co[2]);
endmodule
```

Structural

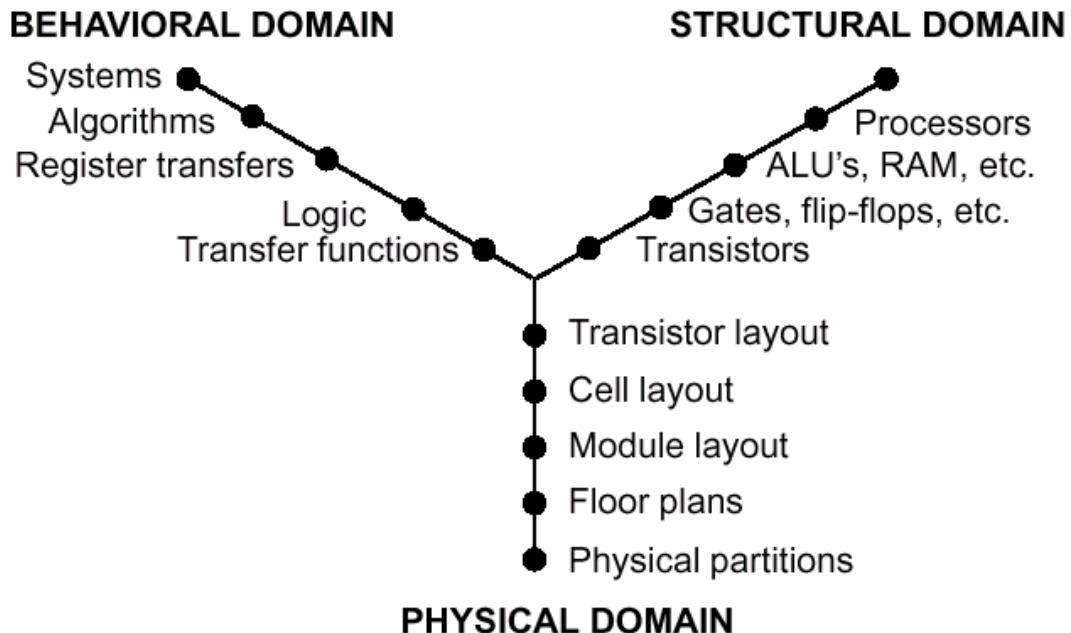


Physical



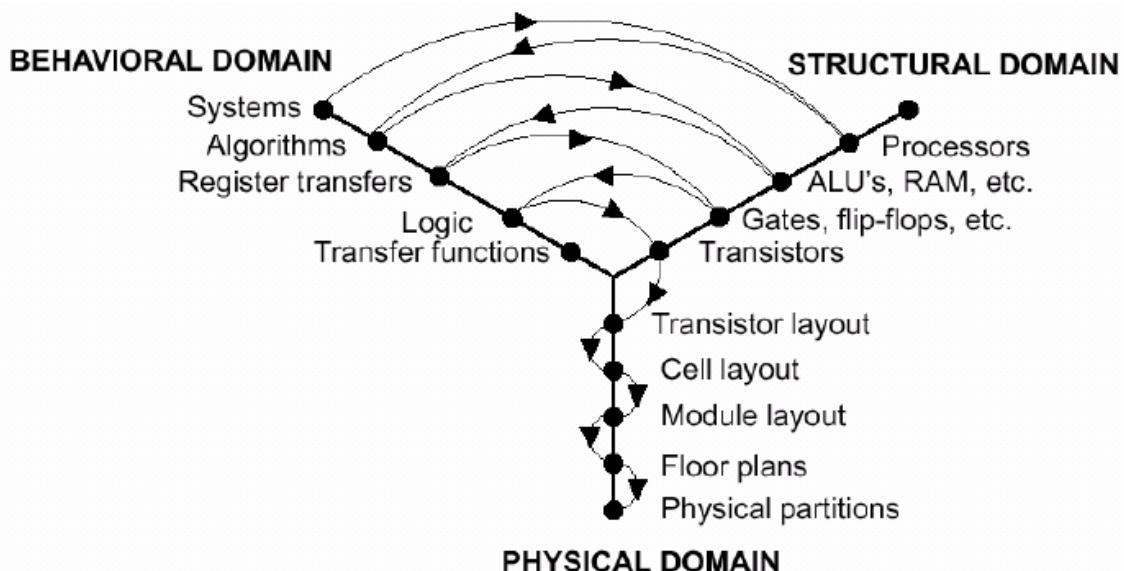
42

Gajski's Y-Chart



43

Top-Down Structural Design



44

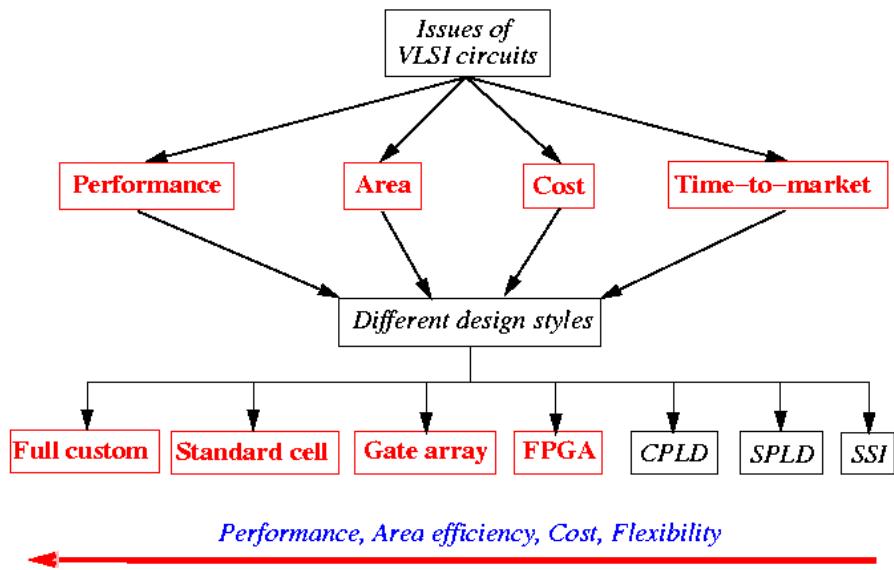
Design Styles

- There are various design styles:
 - Full custom, standard cell, sea of gates, FPGA, etc.
- Why having different design styles?

45

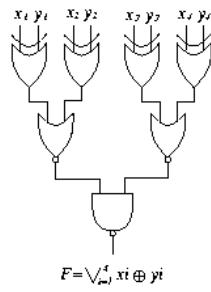
Design Styles

- Specific design styles shall require specific CAD tools

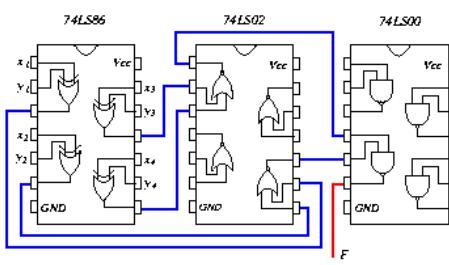


46

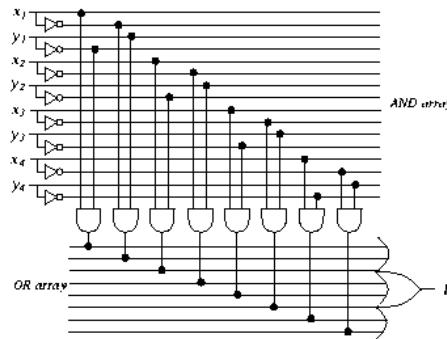
SSI/SPLD Design Style



(a) 4-bit comparator.



(b) SSI implementation.

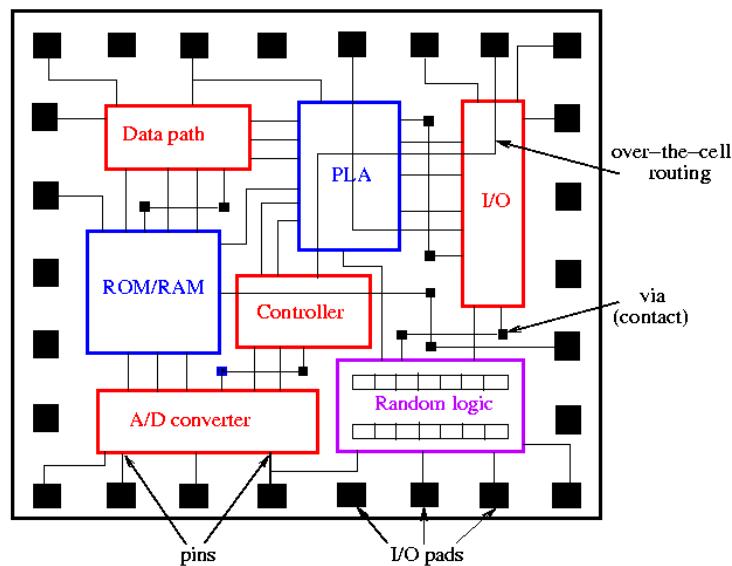


(c) SPLD (PLA) implementation.

47

Full Custom Design Style

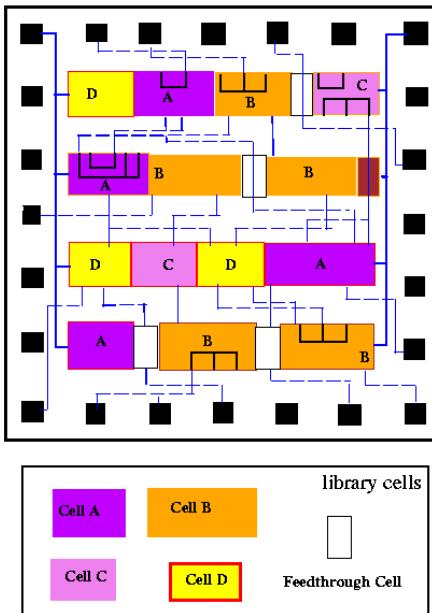
- Designers can control the shape of all mask patterns
- Designers can specify the design up to the level of individual transistors



48

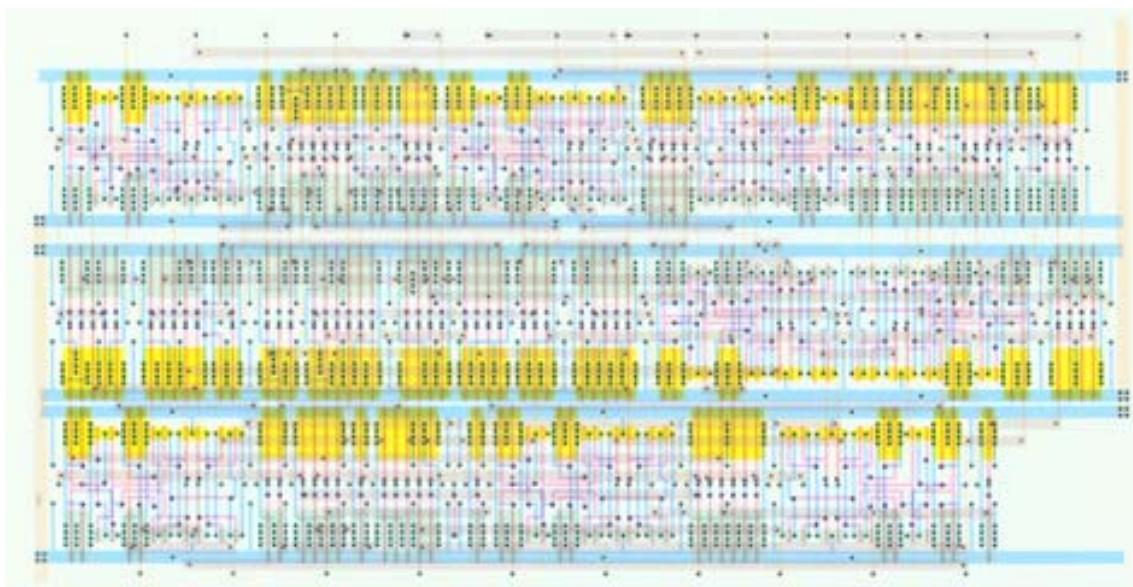
Standard Cell Design Style

- Selects pre-designed cells (of same height) to implement logic



49

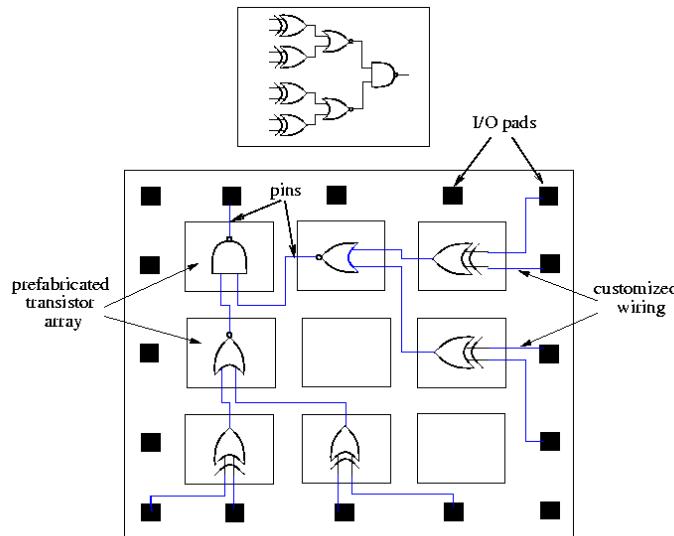
Standard Cell Example



50

Gate Array Design Style

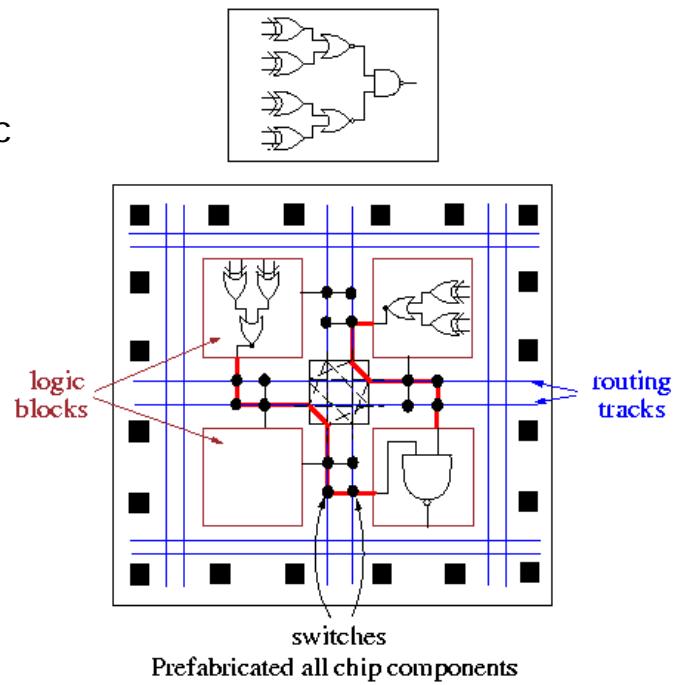
- ❑ Prefabricates a transistor array
- ❑ Needs wiring customization to implement logic



51

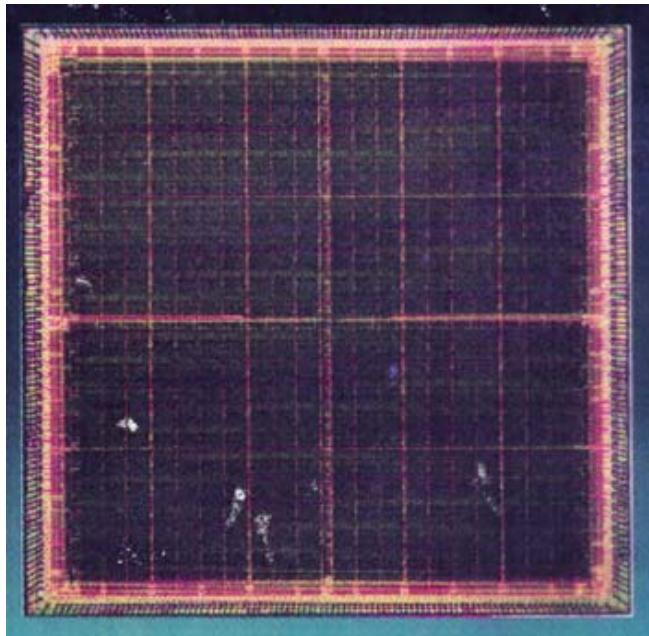
FPGA Design Style

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



52

Array-Based FPGA Example



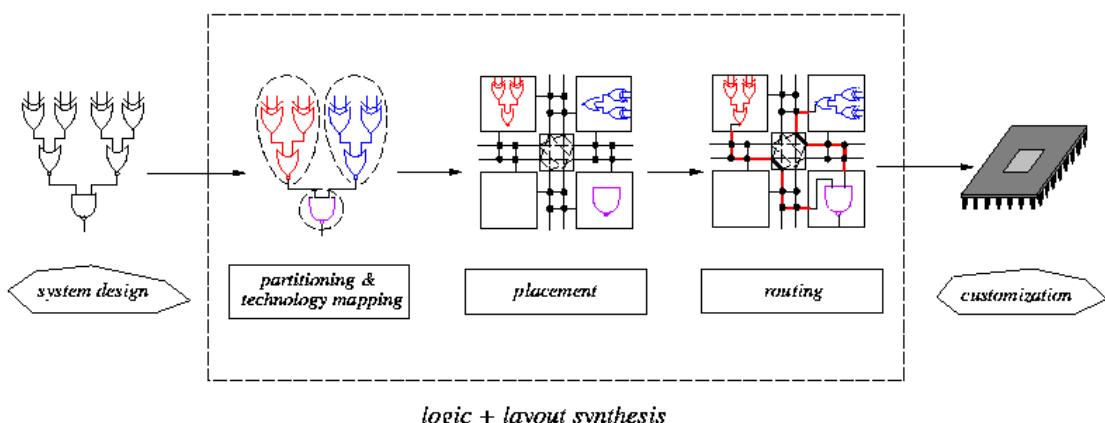
Lucent 15K ORCA FPGA

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

53

FPGA Design Process

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



54

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.

55

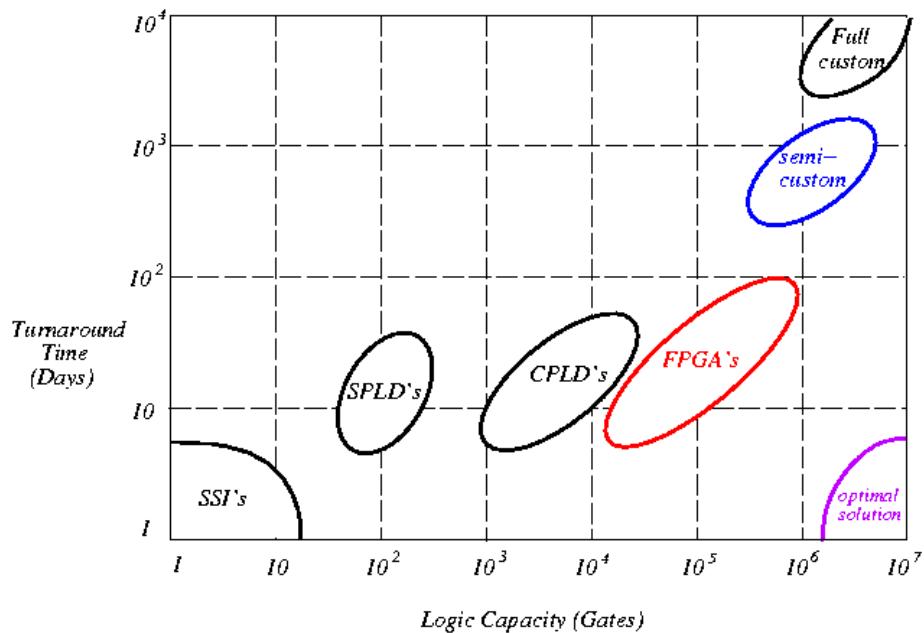
Comparisons of Design Styles

	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

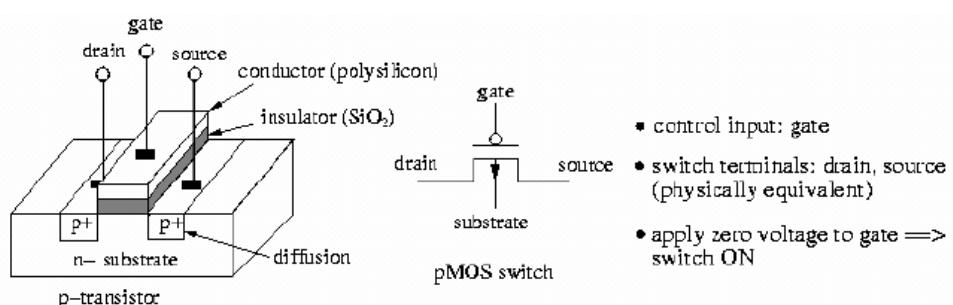
56

Design Style Trade-offs

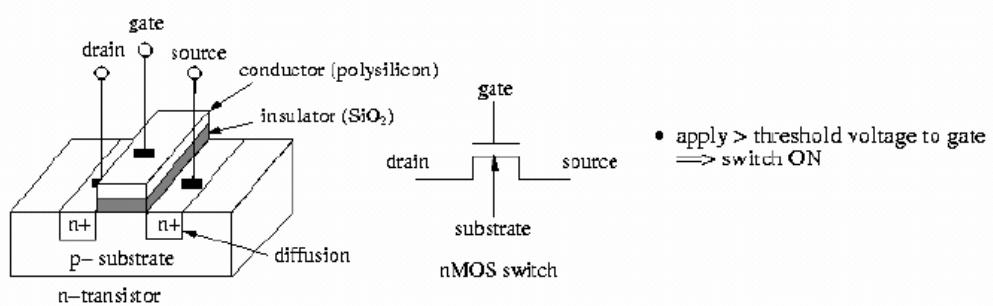


57

MOS Transistors



The pMOS switch passes signal "1" well.

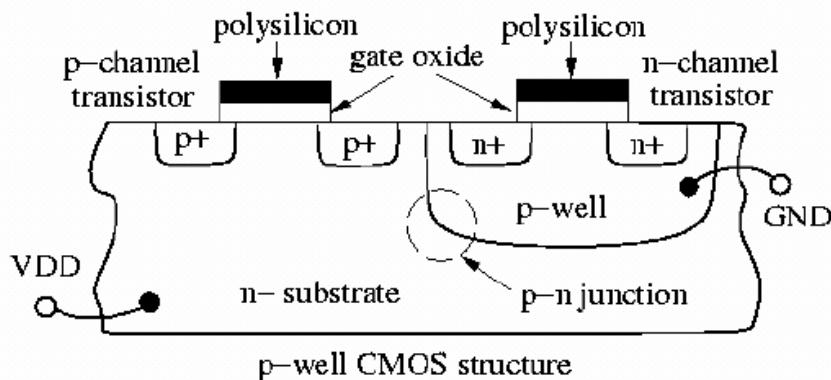


The nMOS switch passes signal "0" well.

58

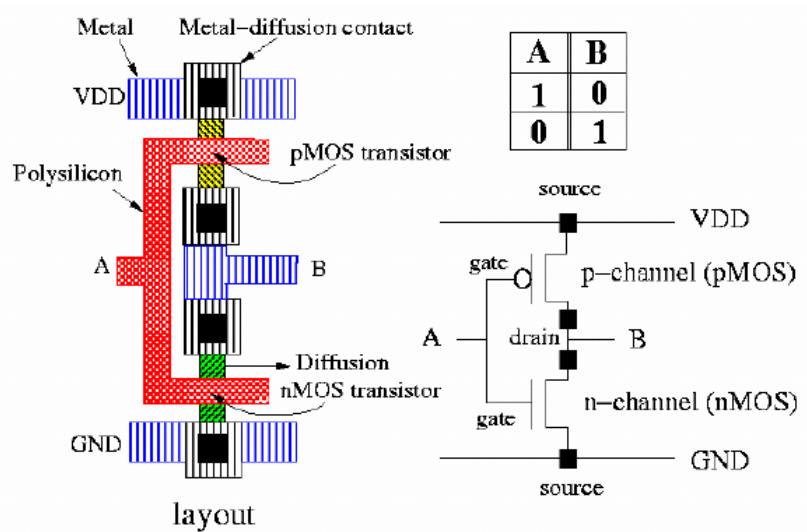
Complementary MOS (CMOS)

- The most popular VLSI technology (v.s. BiCMOS, nMOS)
- CMOS uses both *n*-channel and *p*-channel transistors
- Advantages: lower power dissipation, higher regularity, more reliable performance, higher noise margin, larger fanout, etc.
- Each type of transistor must sit in a material of the complementary type (the reverse-biased diodes prevent unwanted current flow)



59

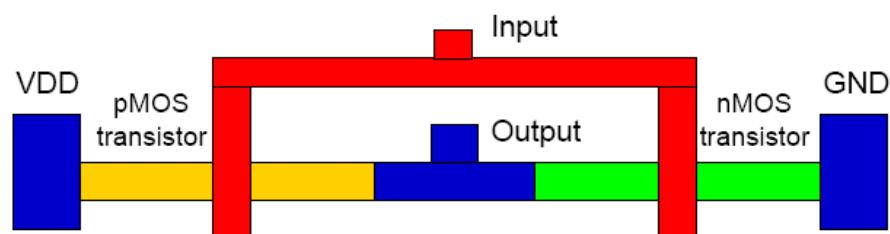
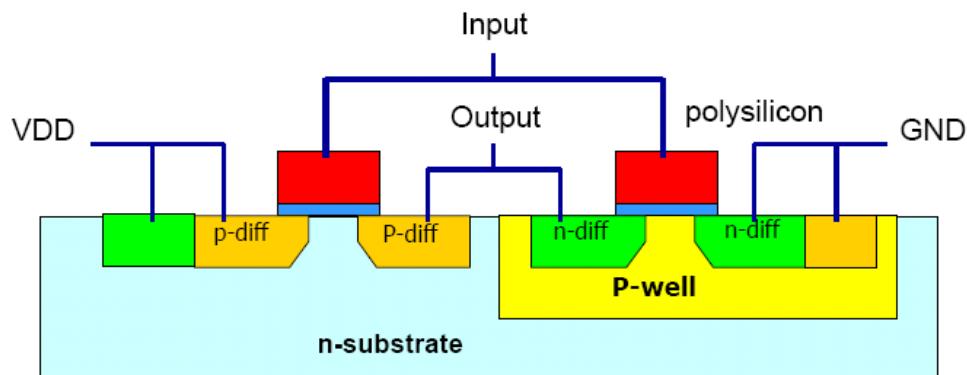
CMOS Inverter



Legend:
Metal 1: blue Polysilicon: red p-diffusion: yellow (p-well: light yellow)
Metal 2: brown Contact/via: black n-diffusion: green (n-well: light green)

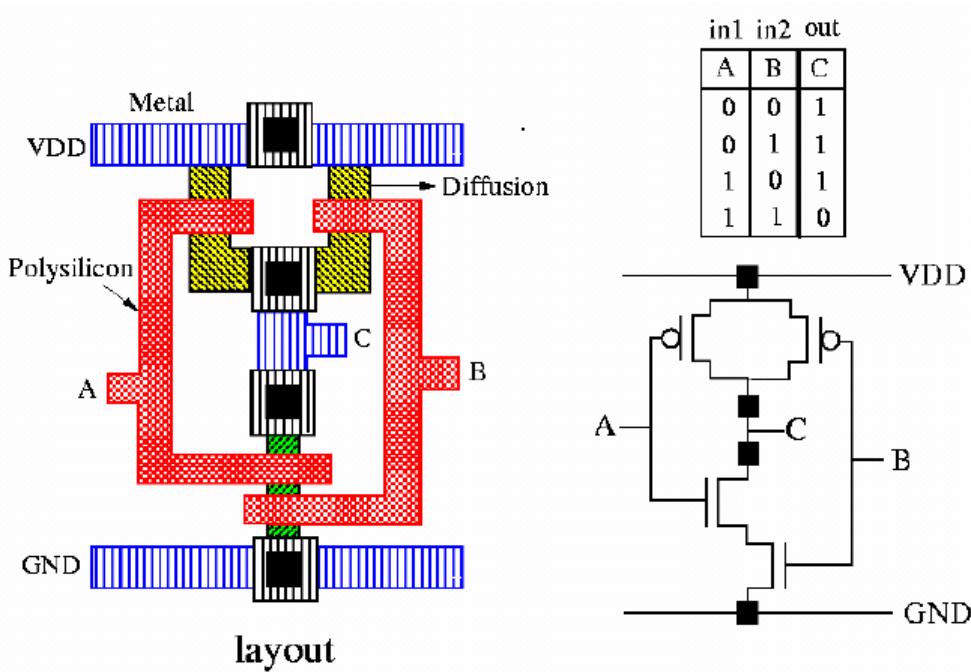
60

CMOS Inverter Cross Section



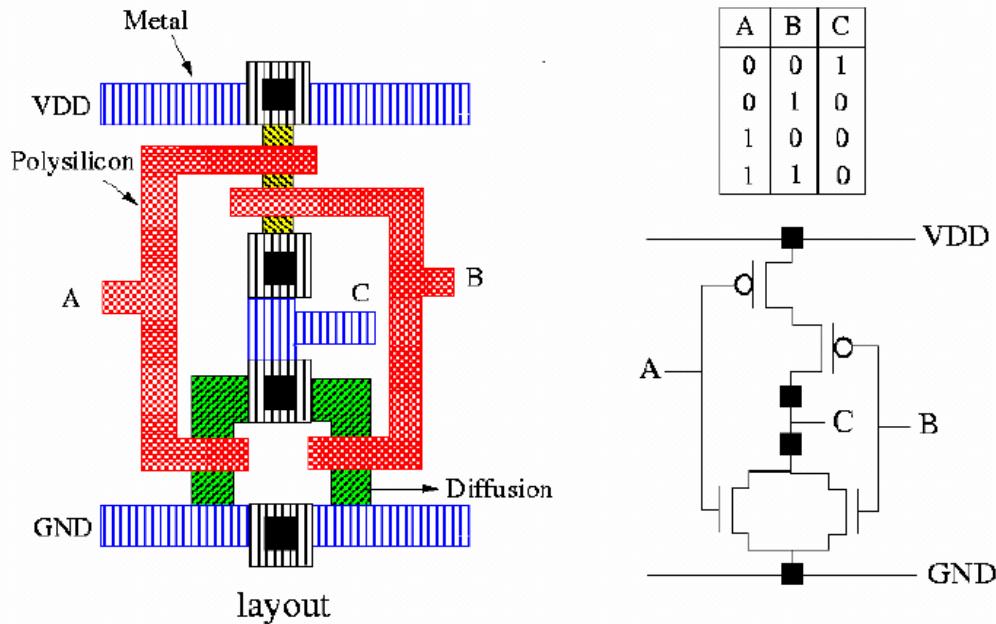
61

CMOS NAND Gate



62

CMOS NOR Gate



63

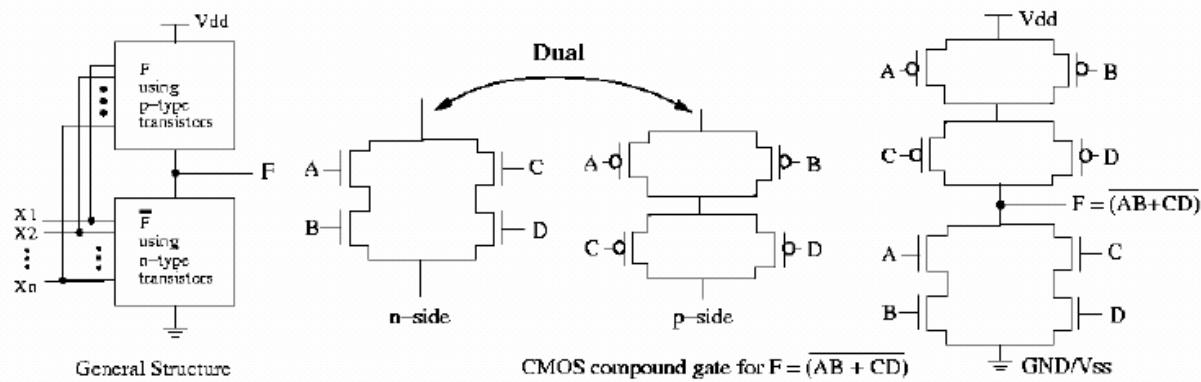
Basic CMOS Logic Library

Name	Distinctive shape	Algebraic equation	Cost (# of transistors)	Scaled gate delay (ps)
AND		$F = XY$	6	24
OR		$F = X + Y$	6	24
NOT (inverter/repeater)		$F = \bar{X}$	2	10
Buffer (driven/repeater)		$F = X$	4	20
NAND		$F = \bar{X}\bar{Y}$	4	14
NOR		$F = \bar{X} + \bar{Y}$	4	14
Exclusive-OR (XOR)		$F = X\bar{Y} + \bar{X}Y = X \oplus Y$	14	42

64

Construction of Compound Gates (1/2)

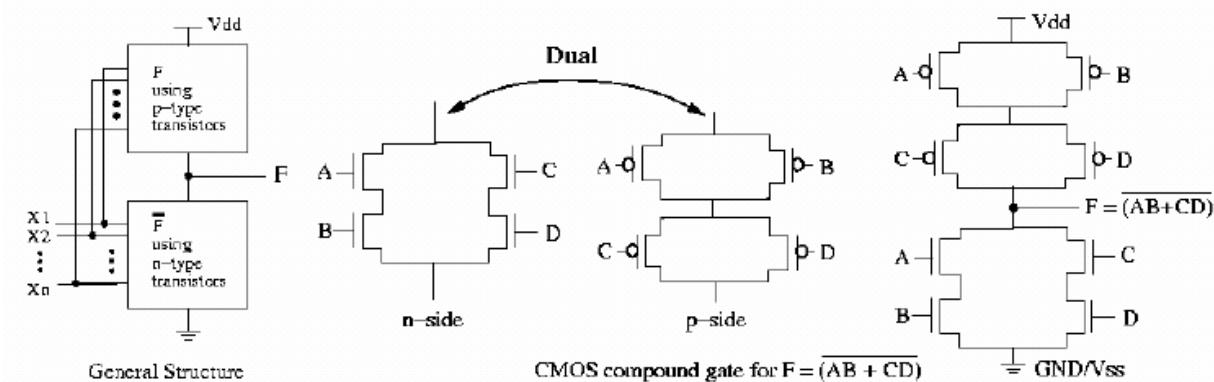
- Example: $F = \overline{A \cdot B + C \cdot D}$
- Step 1 (n-network): Invert F to derive n-network
 - $(\overline{F} = A \cdot B + C \cdot D)$
- Step 2 (n-network): Make connections of transistors:
 - AND \Leftrightarrow Series connection
 - OR \Leftrightarrow Parallel connection



65

Construction of Compound Gates (2/2)

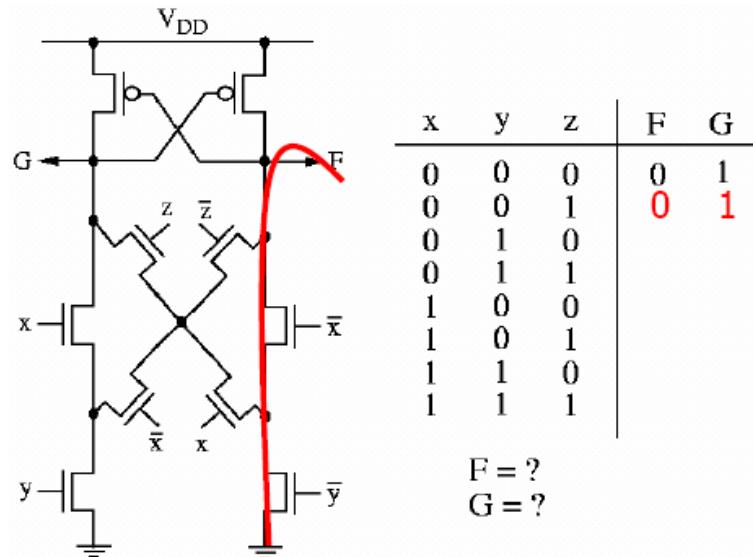
- Step 3 (p-network): Expand F to derive p-network
 - $(F = \overline{AB + CD} = \overline{AB} \cdot \overline{CD} = (\overline{A} + \overline{B}) \cdot (\overline{C} + \overline{D}))$
 - each input is inverted
- Step 4 (p-network): Make connections of transistors (same as Step 2).
- Step 5: Connect the n-network to GND (typically, 0V) and the p-network to VDD (5V, 3.3V, or 2.5V, etc).



66

Complex CMOS Gate

- The functions realized by the n and p networks must be complementary, and one of the networks must conduct for every input combination
- Duality is not necessary



67

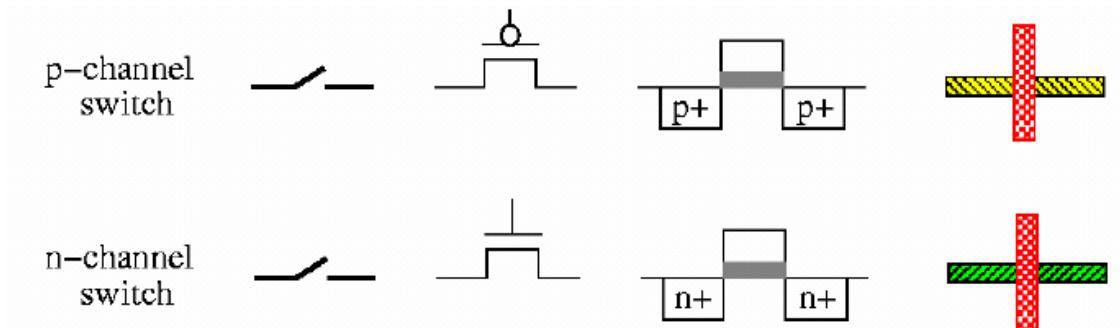
CMOS Properties

- There is always a path from one supply (VDD or GND) to the output.
- There is never a path from one supply to the other. (This is the basis for the low power dissipation in CMOS--virtually no static power dissipation.)
- There is a momentary drain of current (and thus power consumption) when the gate switches from one state to another.
 - Thus, CMOS circuits have dynamic power dissipation.
 - The amount of power depends on the switching frequency.

68

Stick Diagram

- ❑ Intermediate representation between the transistor level and the mask (layout) level.
- ❑ Gives topological information (identifies different layers and their relationship)
- ❑ Assumes that wires have no width.
- ❑ Possible to translate stick diagram automatically to layout with correct **design rules**.



69

Stick Diagram

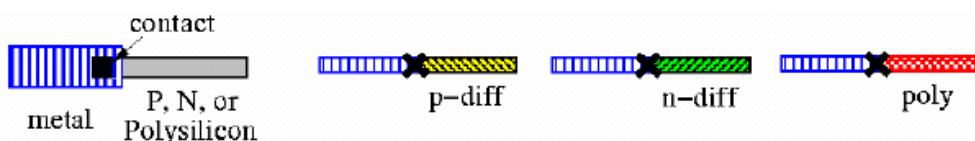
- ❑ When the same material (on the same layer) touch or cross, they are connected and belong to the same electrical node.



- ❑ When **polysilicon** crosses N or P **diffusion**, an N or P transistor is formed.
 - ❑ Polysilicon is drawn on top of diffusion.
 - ❑ Diffusion must be drawn connecting the source and the drain.
 - ❑ Gate is automatically self-aligned during fabrication.



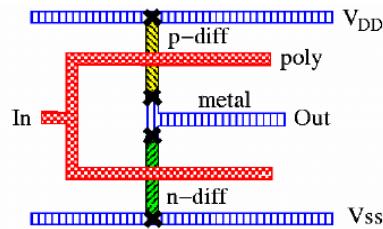
- ❑ When a metal line needs to be connected to one of the other three conductors, a **contact** cut (**via**) is required.



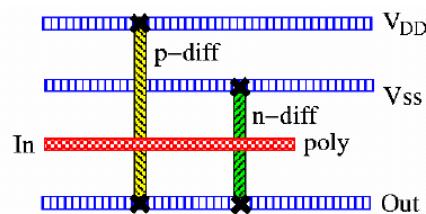
70

CMOS Inverter Stick Diagram

□ Basic layout

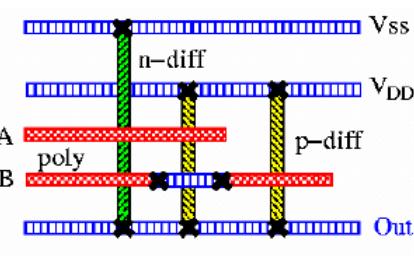
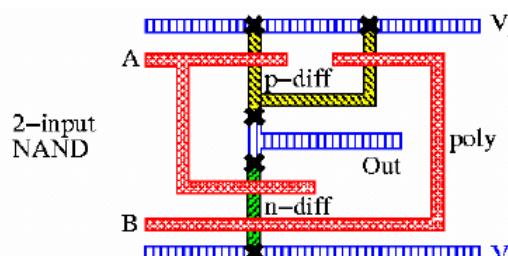


□ More area efficient layout

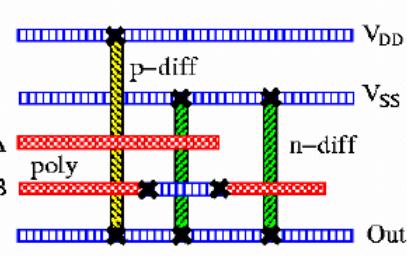
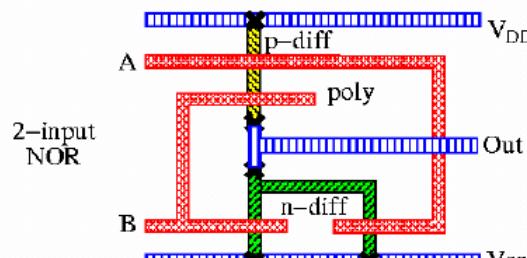


71

CMOS NAND/NOR Stick Diagram



area efficient? contacts?



72

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.

73

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

74

SCMOS Design Rules

