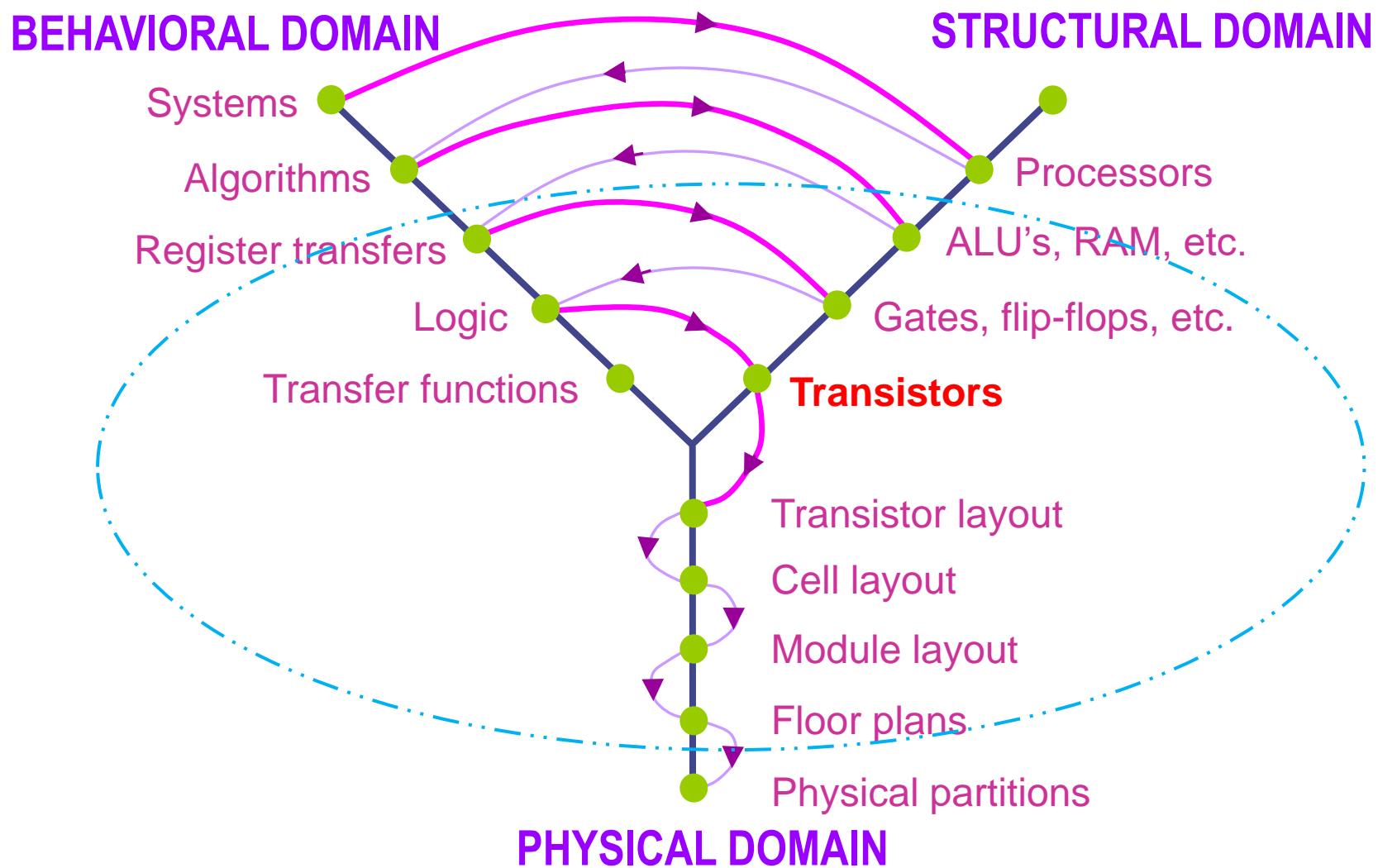




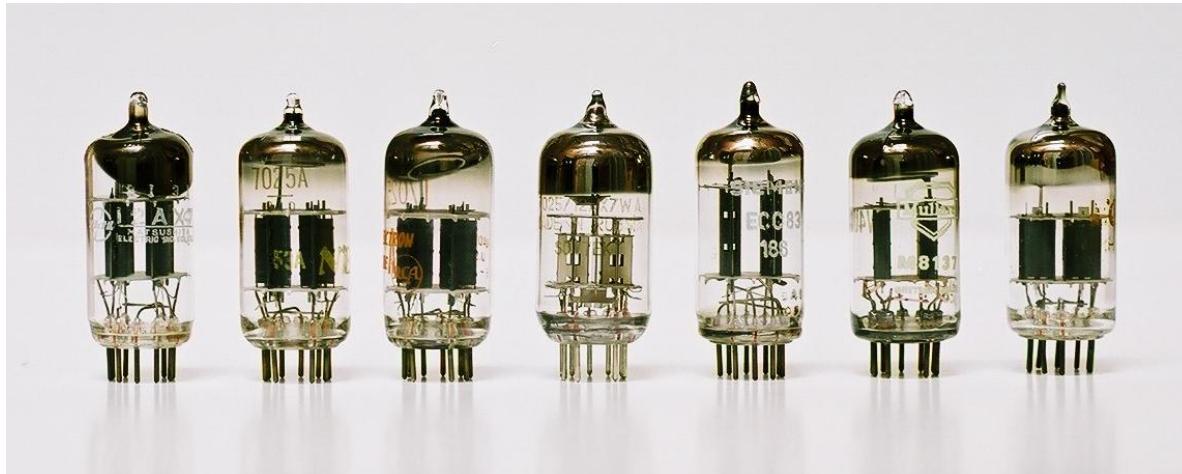
Introduction to VLSI Design

Top-down Structural Design



Electronics Industry (1/5)

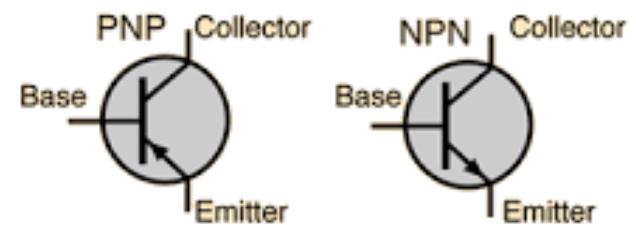
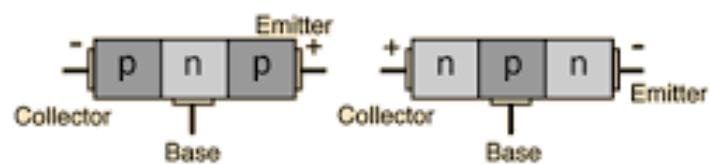
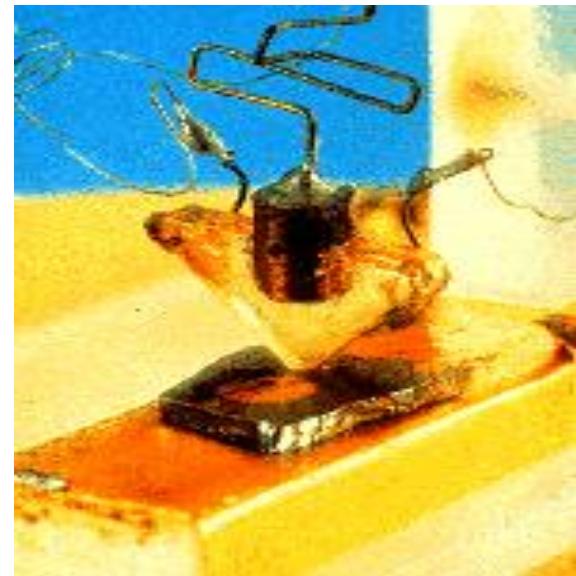
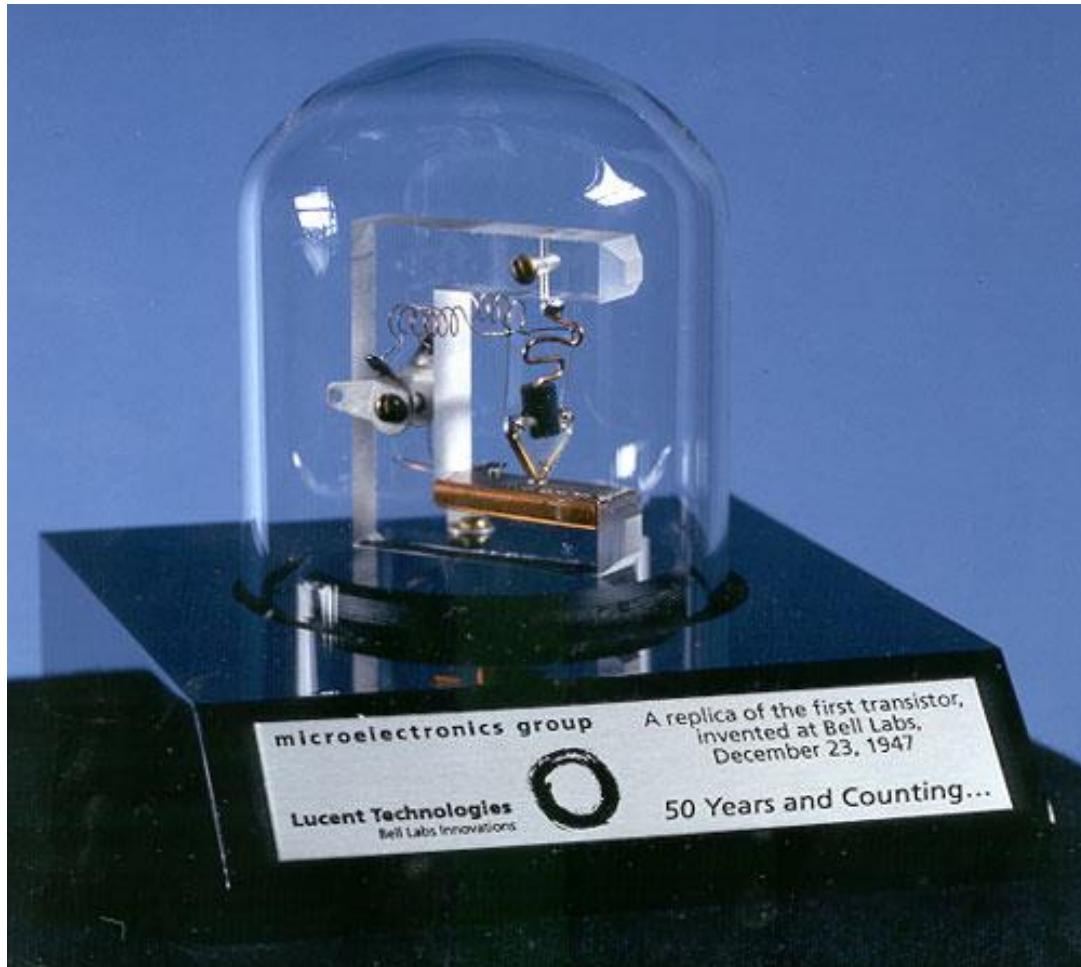
■ Vacuum Tube (1904: Diodes, 1907: Triodes)



Electronics Industry (2/5)

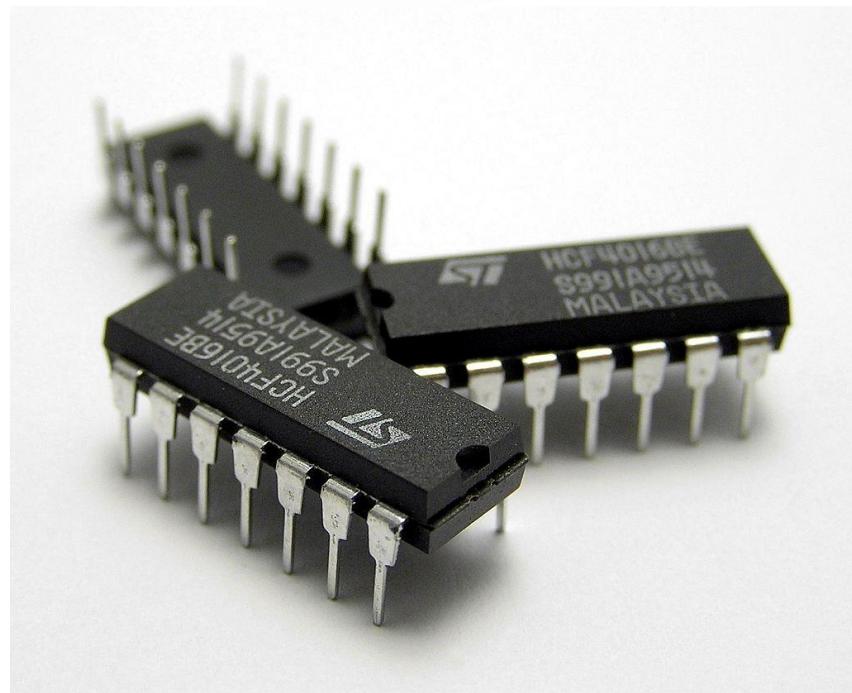
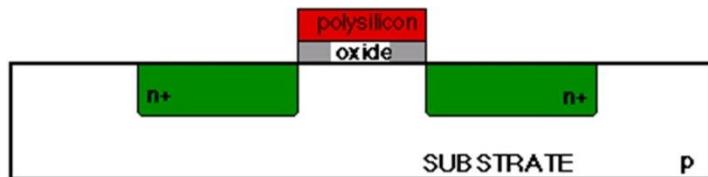
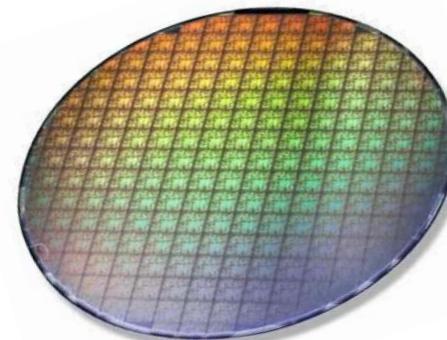
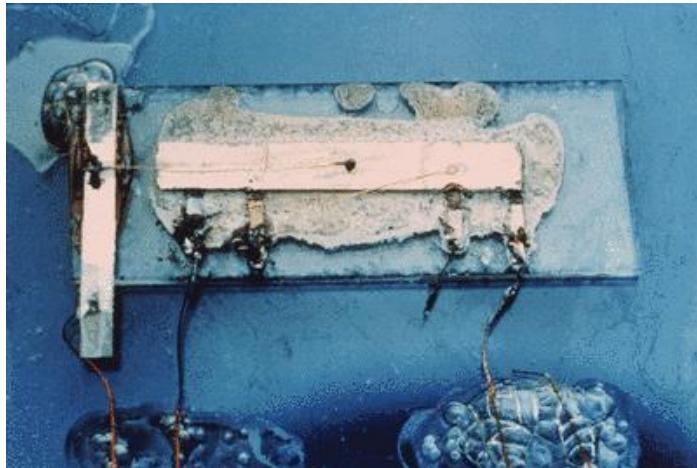


Transistor (1947)



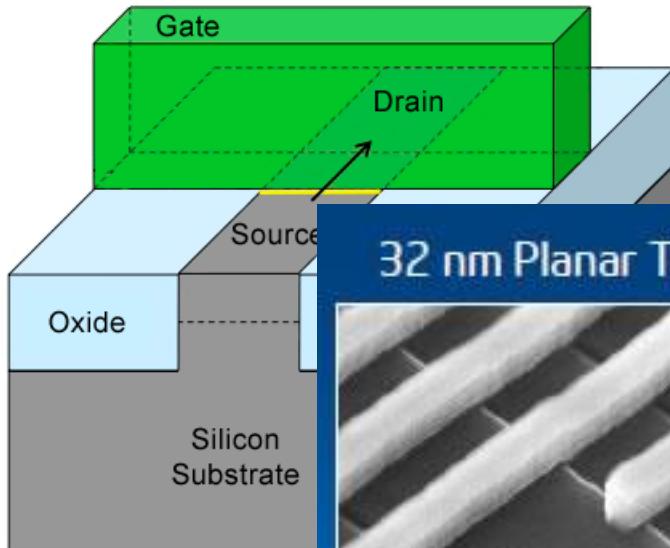
Electronics Industry (3/5)

Integrated Circuits (ICs, 1958)



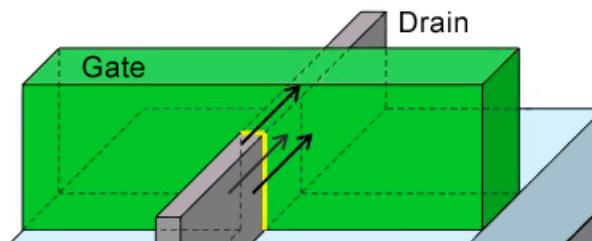
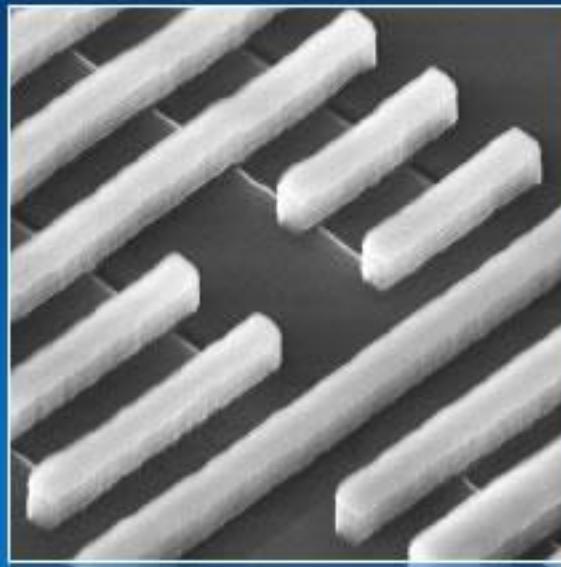
Electronics Industry (4/5)

- Multiple Independent Gate Field Effect Transistor (MIGFET)
 - ◆ Tri-gate transistor (Intel), FinFET (TSMC)

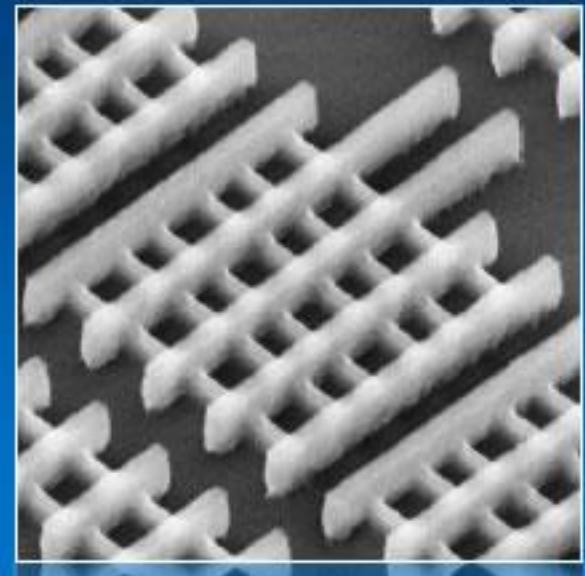


2D Planar Str

32 nm Planar Transistors

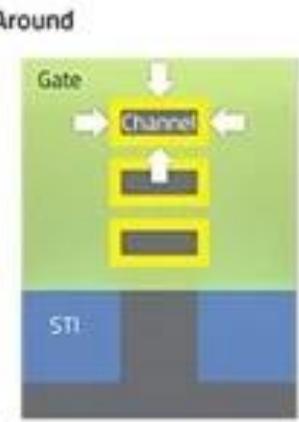
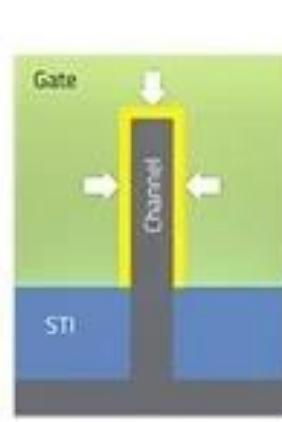
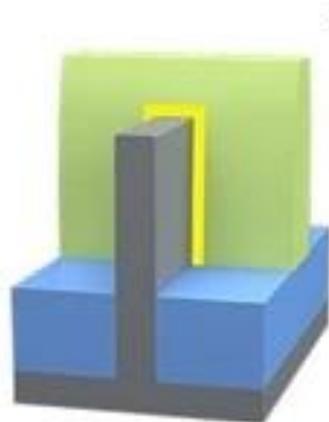
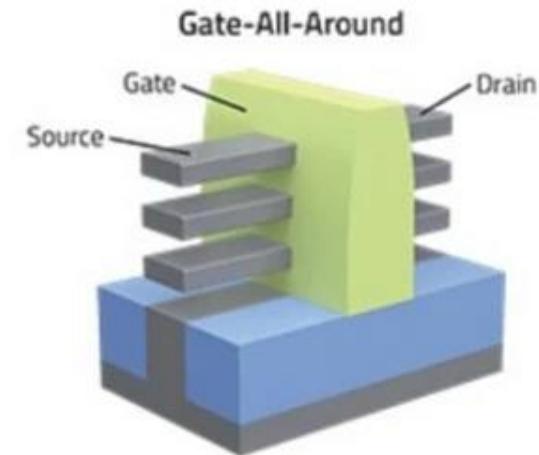
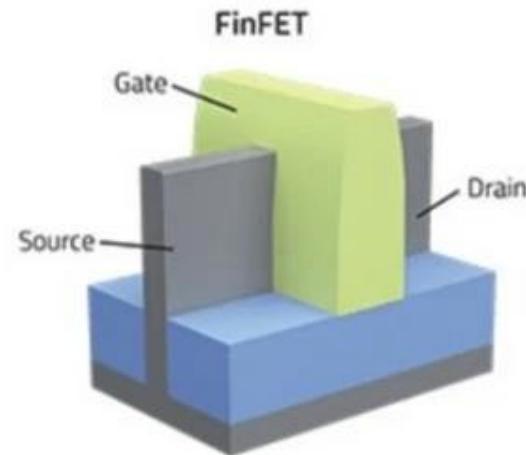
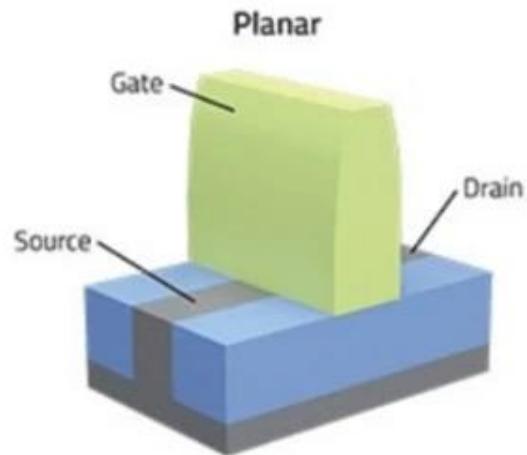


22 nm Tri-Gate Transistors



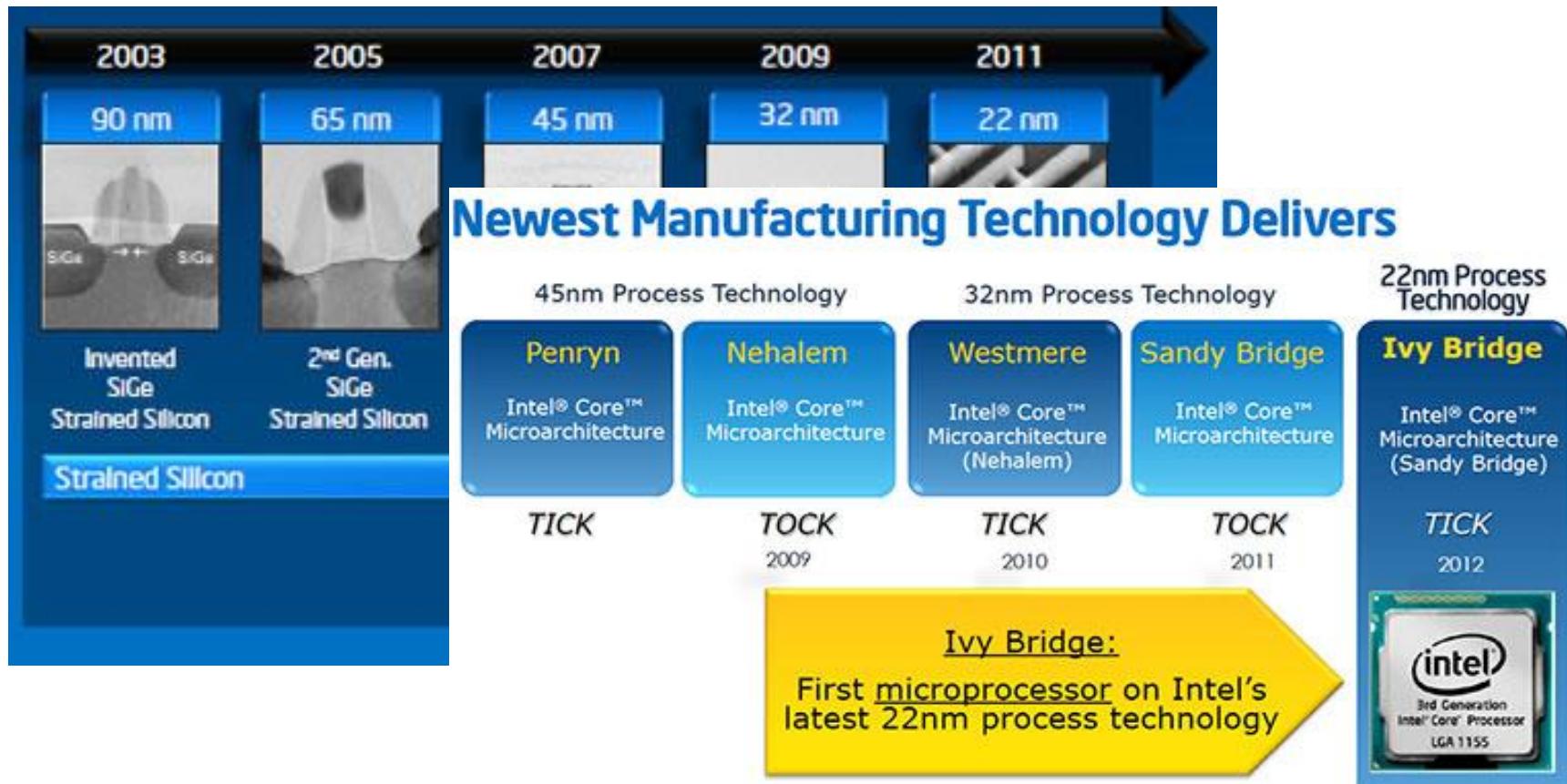
Electronics Industry (5/5)

Gate-All-Around Field Effect Transistor (GAAFET)



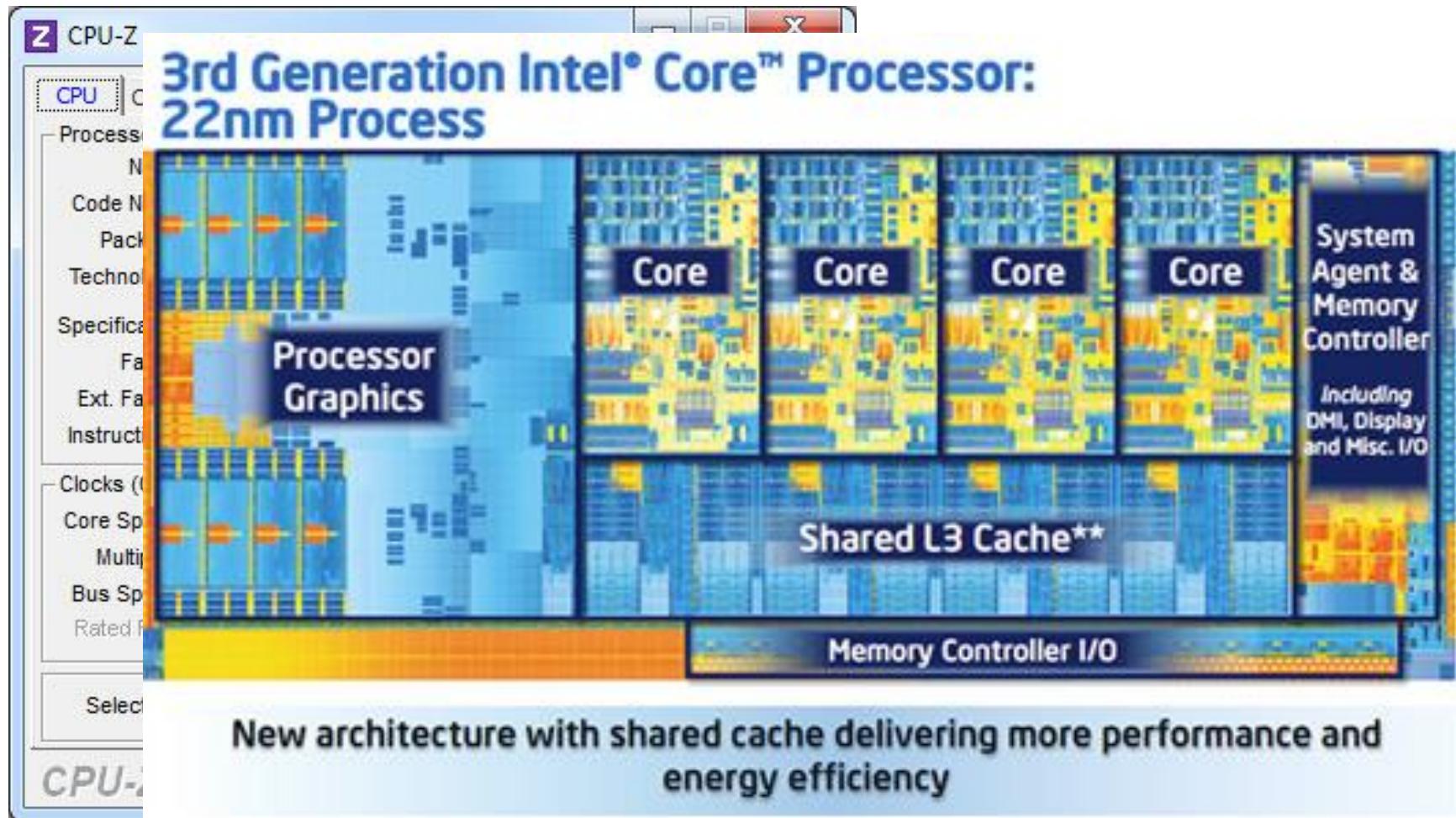
3D Transistor (1/4)

■ Tri-gate transistor (Intel)



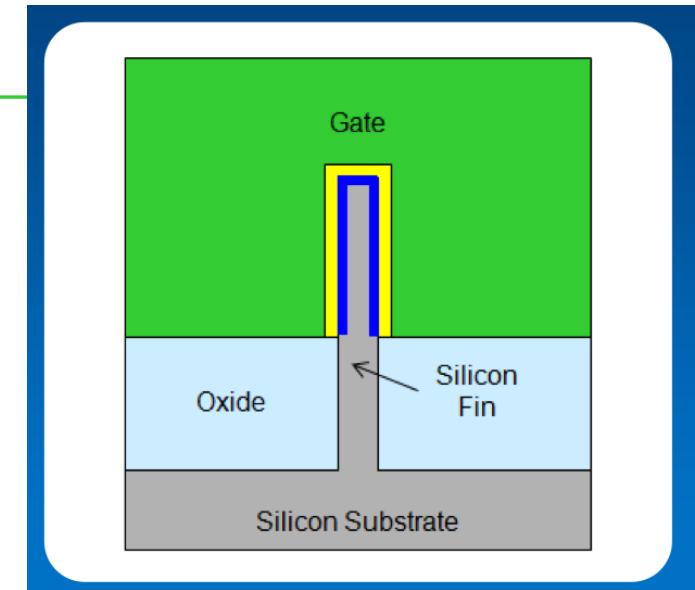
Cadence of Innovation delivers New Microprocessor Efficiency on the 22nm Process

3D Transistor (2/4)



Quad Core die with Intel® HD Graphics 4000 shown above
Transistor count: 1.4Billion Die size: 160mm²
** Cache is shared across all 4 cores and processor graphics

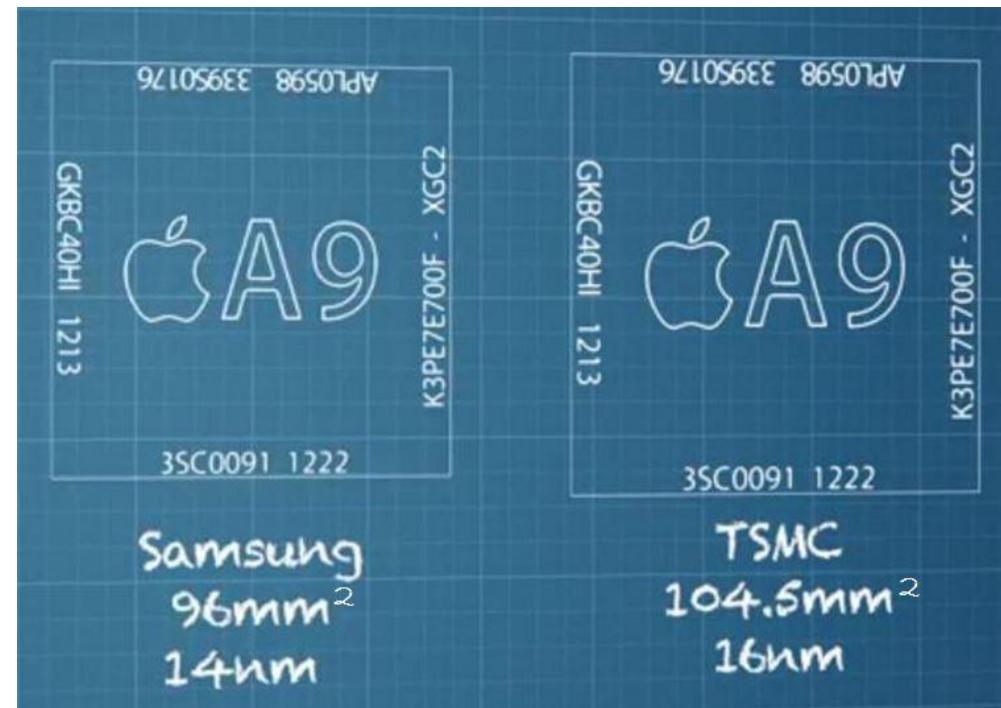
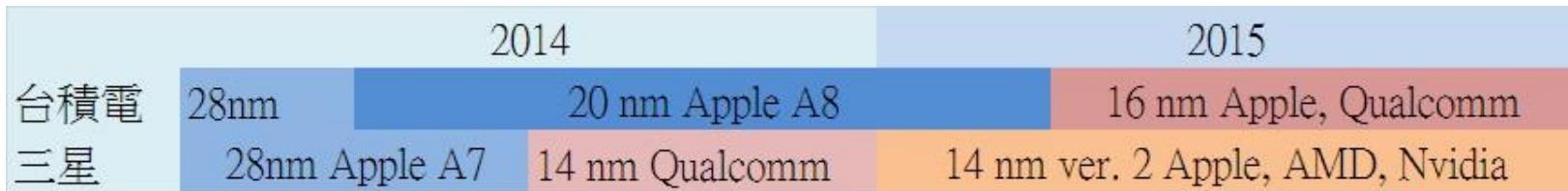
3D Transistor (3/4)



| 台積電先進製程布局 | | | | |
|-----------|------------------------|------------------------|----------------|----------------|
| 製程世代 | 20奈米 | 16奈米 | 10奈米 | 7奈米 |
| 電晶體架構 | 平面 | 3D FinFET | 3D FinFET | 3D FinFET |
| 量產時間 | 2014年第2季 | 2014年第3季 | 2016年下半年 | 2017至2018年 |
| 主要生產據點 | 南科12吋廠 Fab 14第五至第六期 | 南科12吋廠 Fab 14第七至第六期 | 中科Fab 15第五到第六期 | 中科Fab 15第五到第六期 |

資料來源：台積電及法人機構
編示祥 / 製表

3D Transistor (4/4)

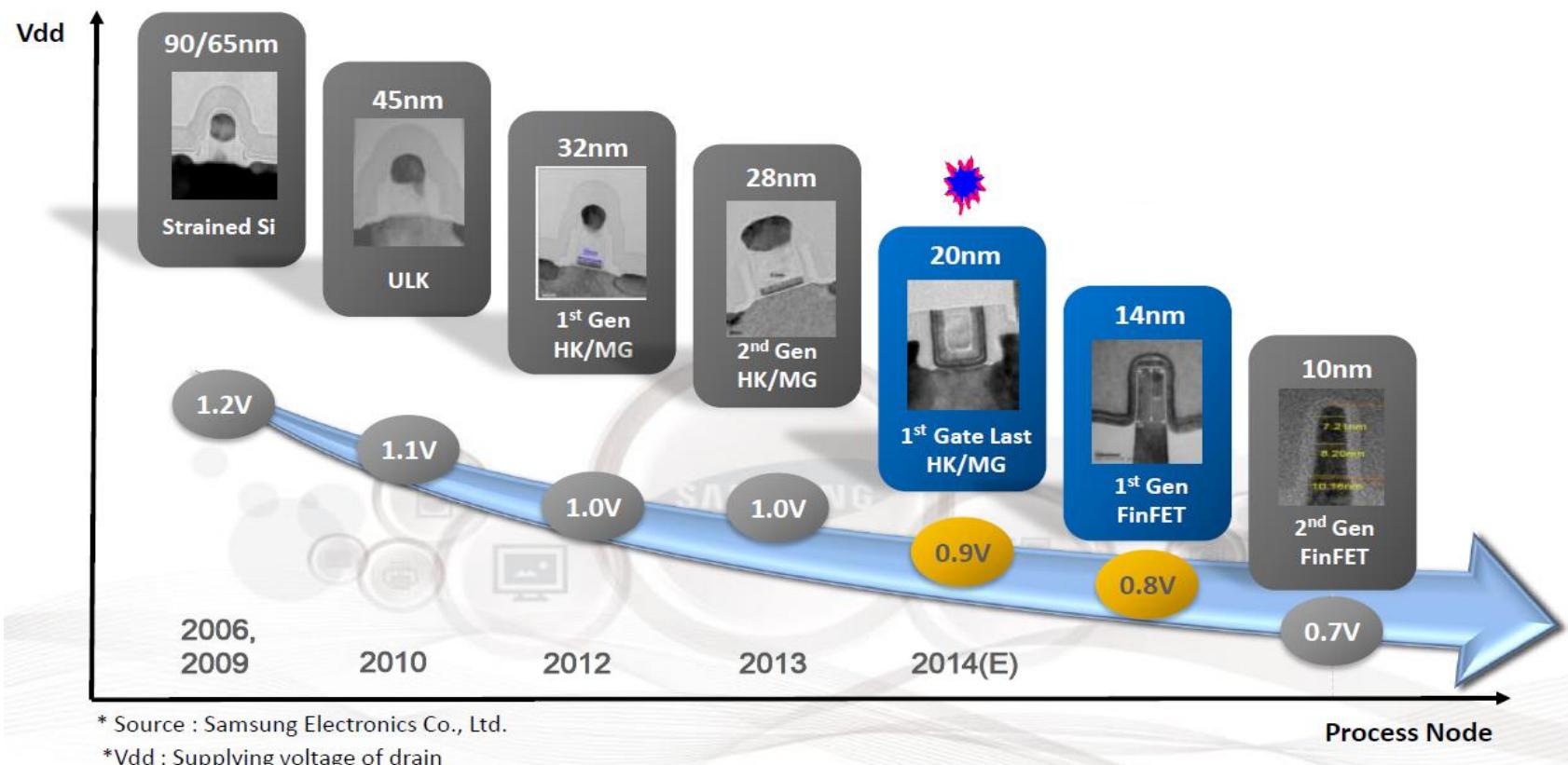


Evolution of the Integrated Circuit (1/4)

- 1947: W. Shockley, J. Bardeen, and W. H. Brattain of Bell Laboratories invented the **transistor**
- 1958: Kilby invented **integrated circuits (ICs)**
 - ◆ Metal-oxide-semiconductor field-effect transistor (MOSFET)
 - ◆ Began to become commercially available in the **early 1960s**
- 1965: Gordon Moore proposed **Moore's law**
 - ◆ The effectiveness of semiconductor technology (the maximum number of transistors on a chip) approximately **doubles** every **eighteen months**

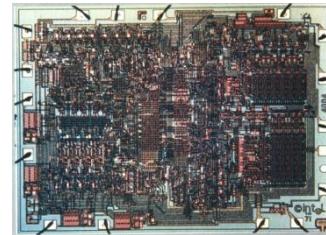
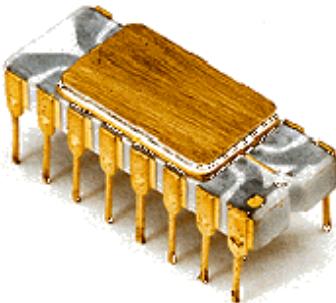
Moore's Law

- Successive generations of technology have used a scaling $S = \sqrt{2}$, doubling the number of transistors per unit area
- A scaling factor (S) reduces device dimensions as $1/S$



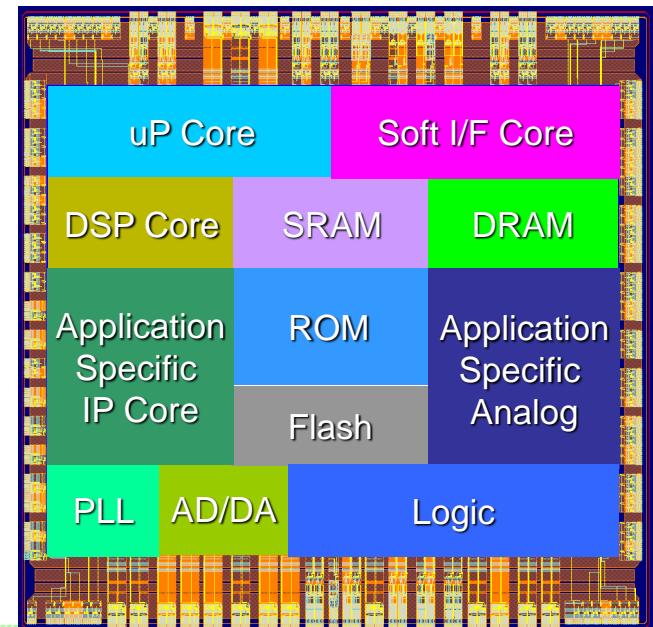
Evolution of the Integrated Circuit (2/4)

- 1968: Noyce and Moore founded Intel
- 1971: Intel announced 4-bit 4004 microprocessors (2250 transistors)
- 1976/81: Apple II /IBM PC (technology driver)
- 1984: Xilinx invented FPGA's
- 1985: Intel began focusing on microprocessor products
- 1987: TSMC was founded (fabless IC design)



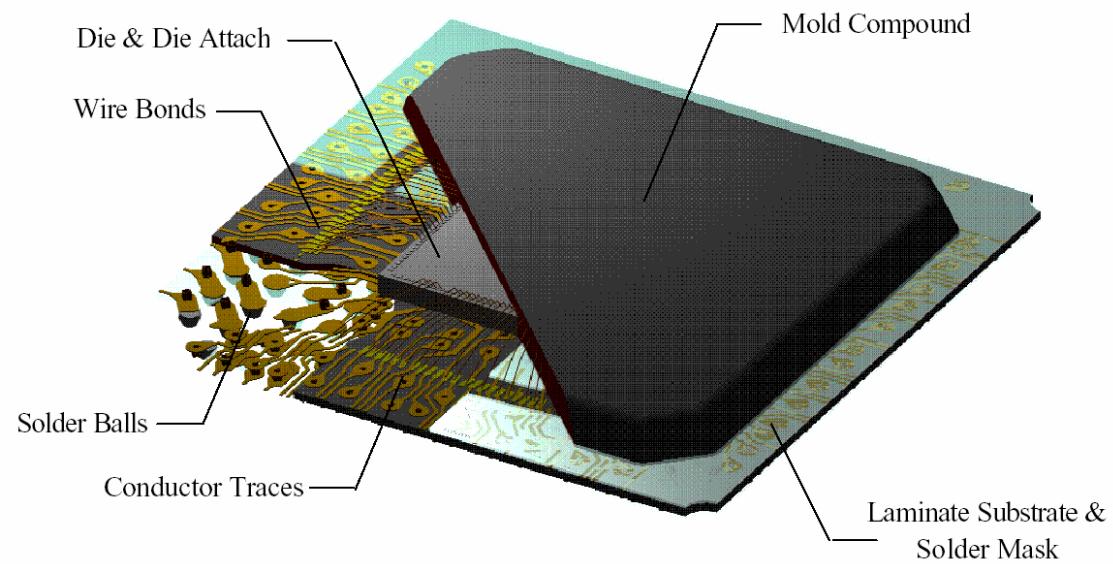
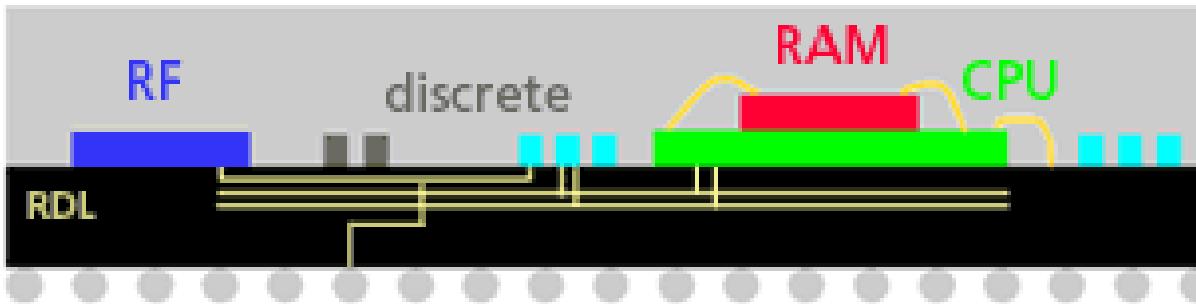
Evolution of the Integrated Circuit (3/4)

- 1991: ARM introduced its first embeddable RISC IP core (chipless IC design)
- 1996: Samsung introduced 1G DRAM
- 1998: IBM announces 1GHz experimental microprocessor
- 1999: System-on-Chip (SoC) applications
- 2002: System-in-Package (SiP) technology
- 3D IC
- 3D Transistor

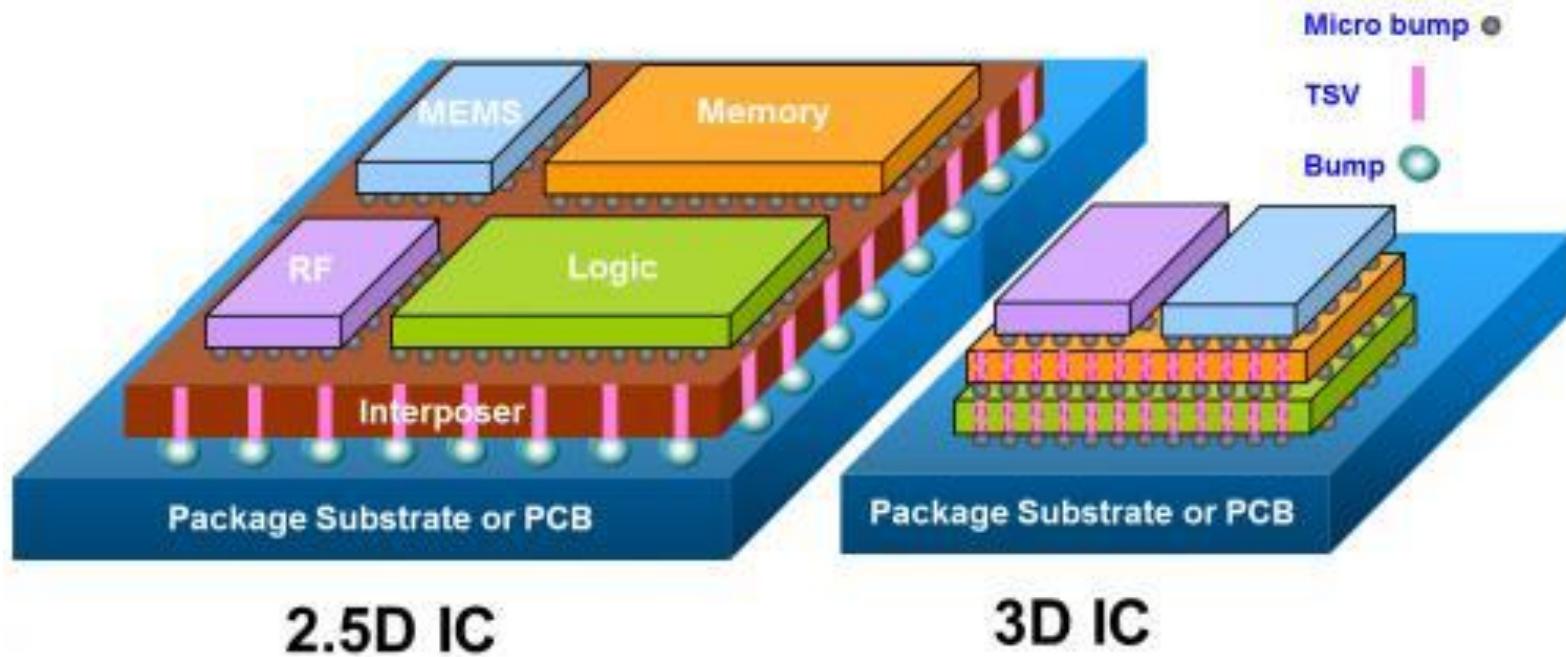


System-in-Package (SiP)

SiP System in Package



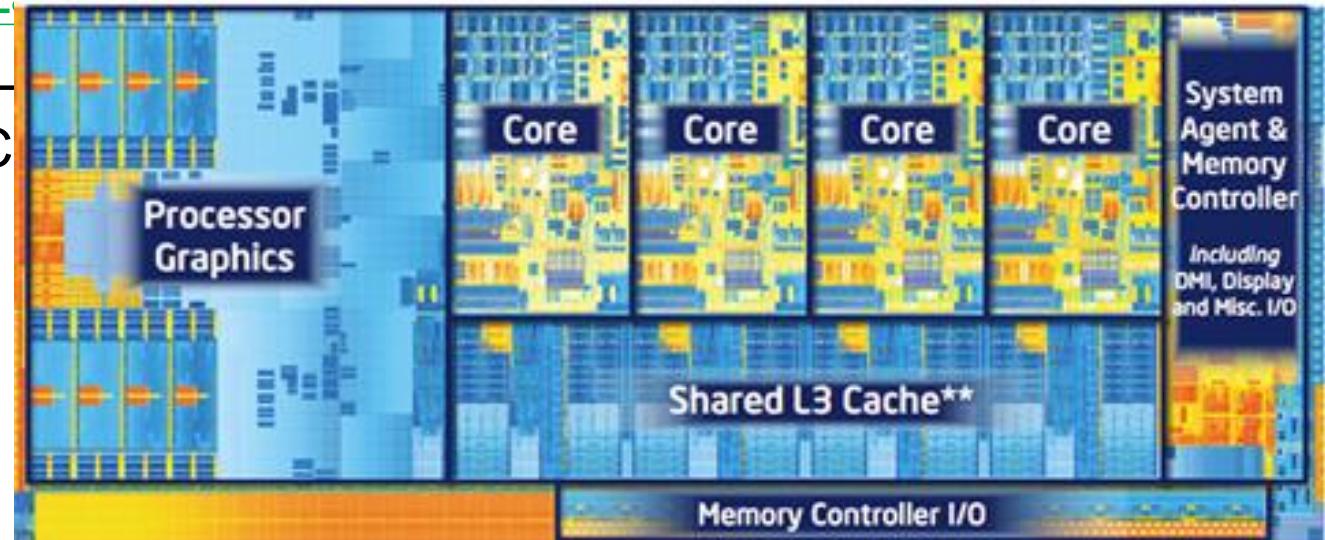
3D IC



Evolution of the Integrated Circuit (4/4)

- The number of components integrated on a single chip
 - SSI (Small Scale Integration): < 10 transistors
 - MSI (Medium Scale Integration): 10 ~ 1,000 transistors
 - LSI (Large Scale Integration): 1,000 ~ 10,000 transistors
 - VLSI (Very Large Scale Integration): 10,000 ~ 1,000,000 transistors
 - ULSI (Ultra Large Scale Integration): > 1,000,000 transistors
 - System on Chip (SoC)

3rd Generation Intel® Core™ Processor: 22nm Process



New architecture with shared cache delivering more performance and energy efficiency

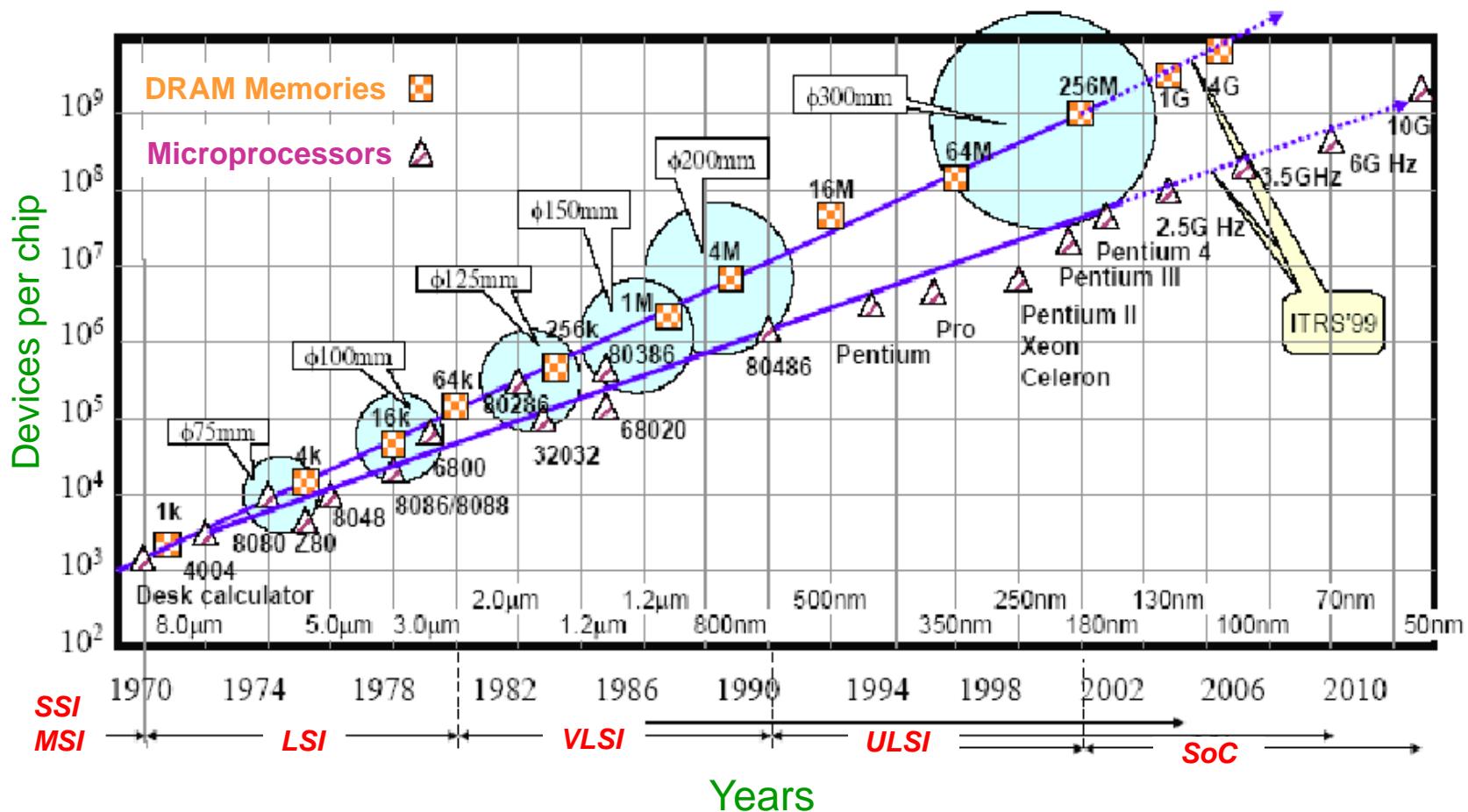
Quad Core die with Intel® HD Graphics 4000 shown above

Transistor count: 1.4Billion

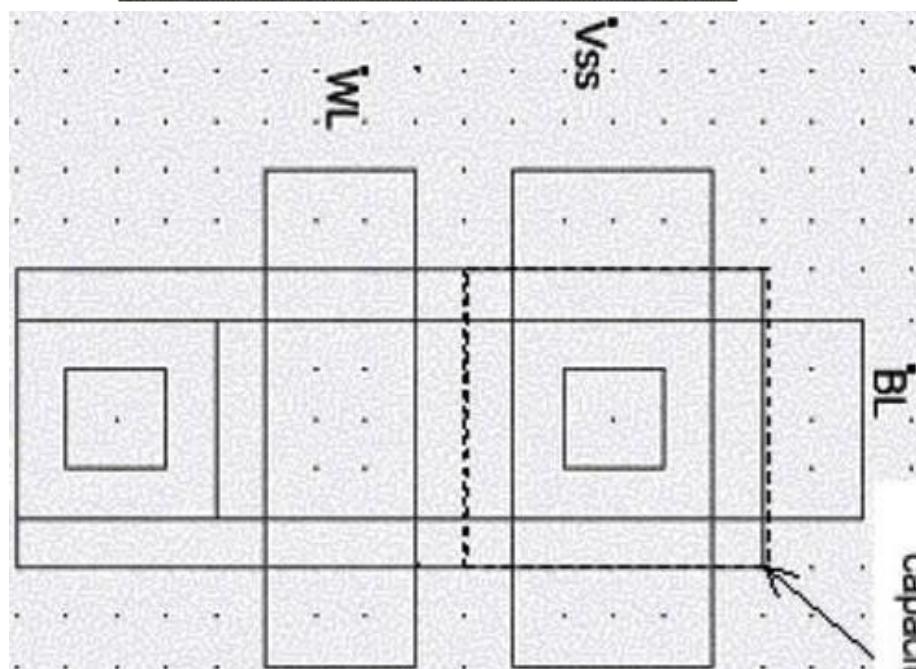
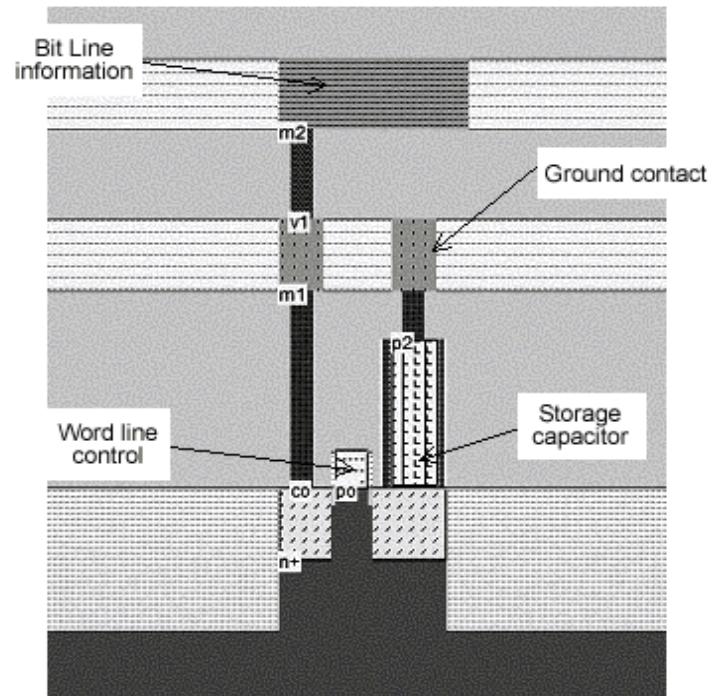
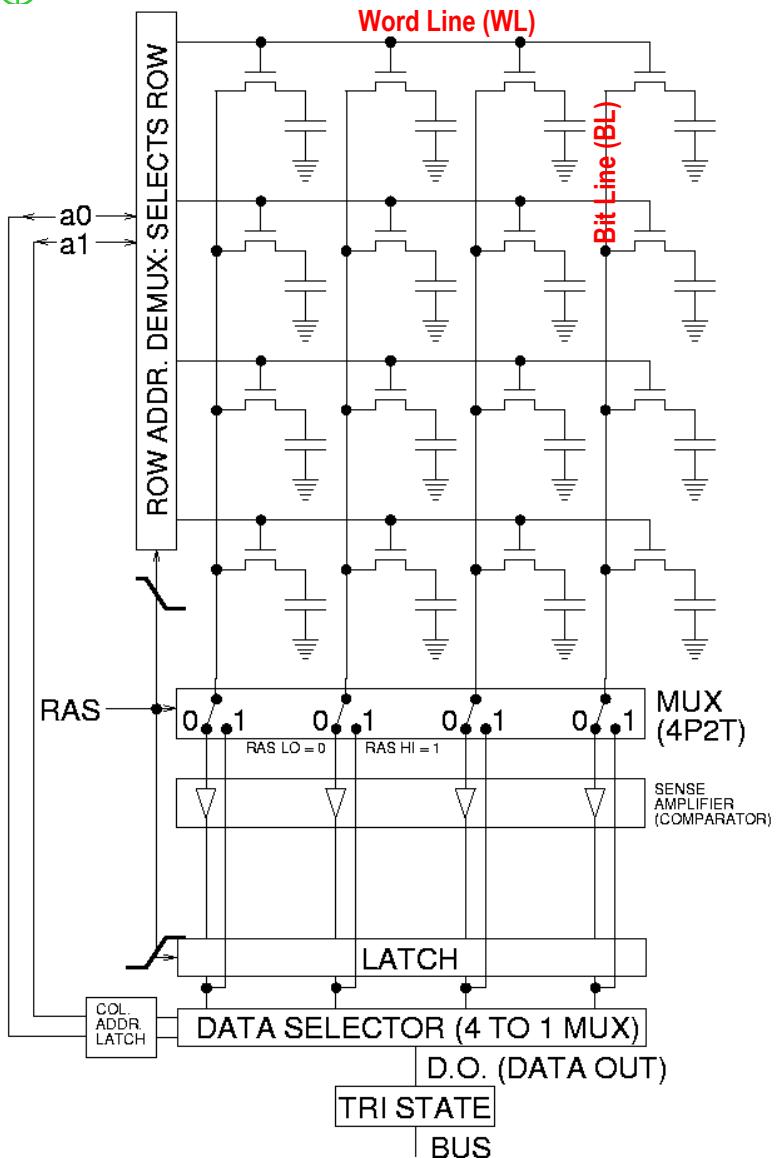
Die size: 160mm²

** Cache is shared across all 4 cores and processor graphics

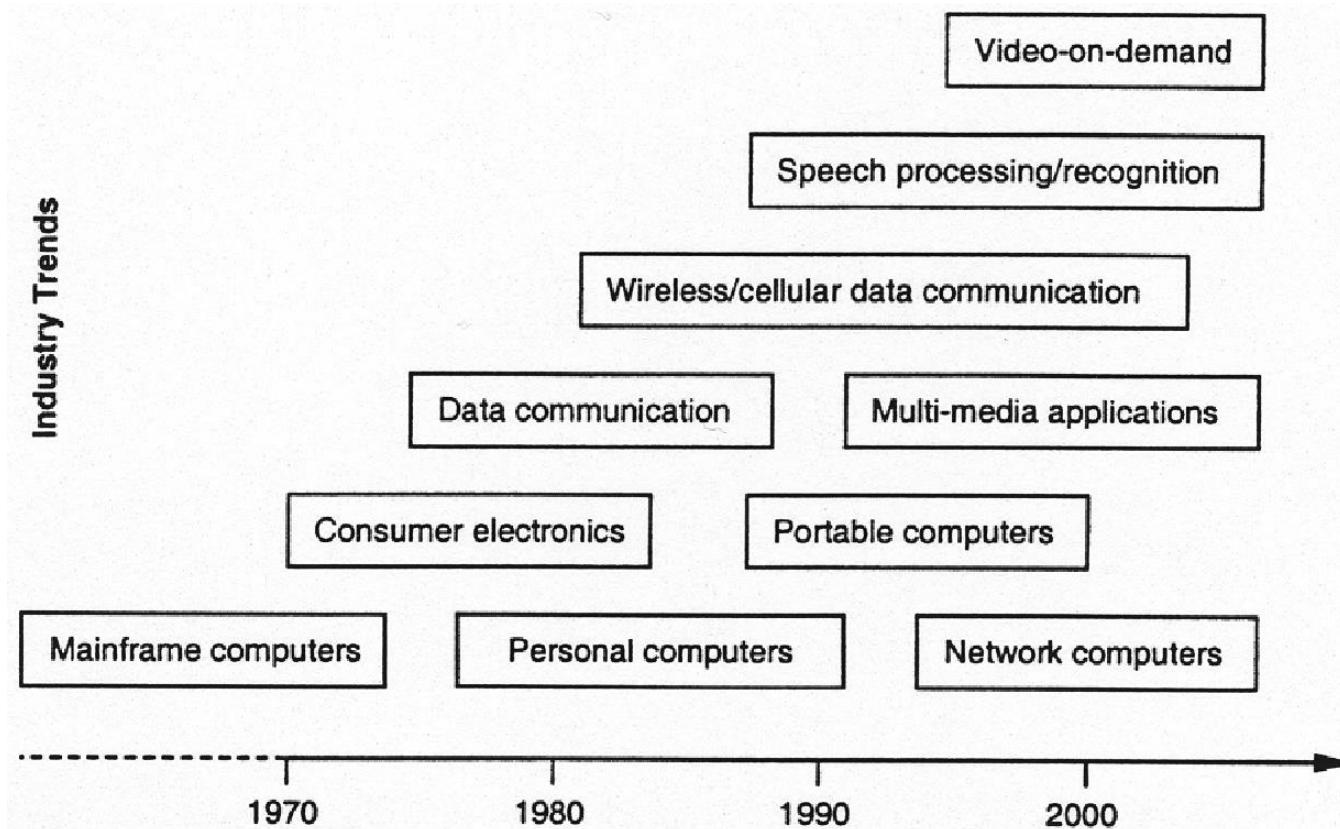
IC Design History



Dynamic RAM Cells



Industry Trends (1/3)



**Large
Centralized
Expensive**



**Small / Portable
Distributed
Inexpensive**

Industry Trends (2/3)



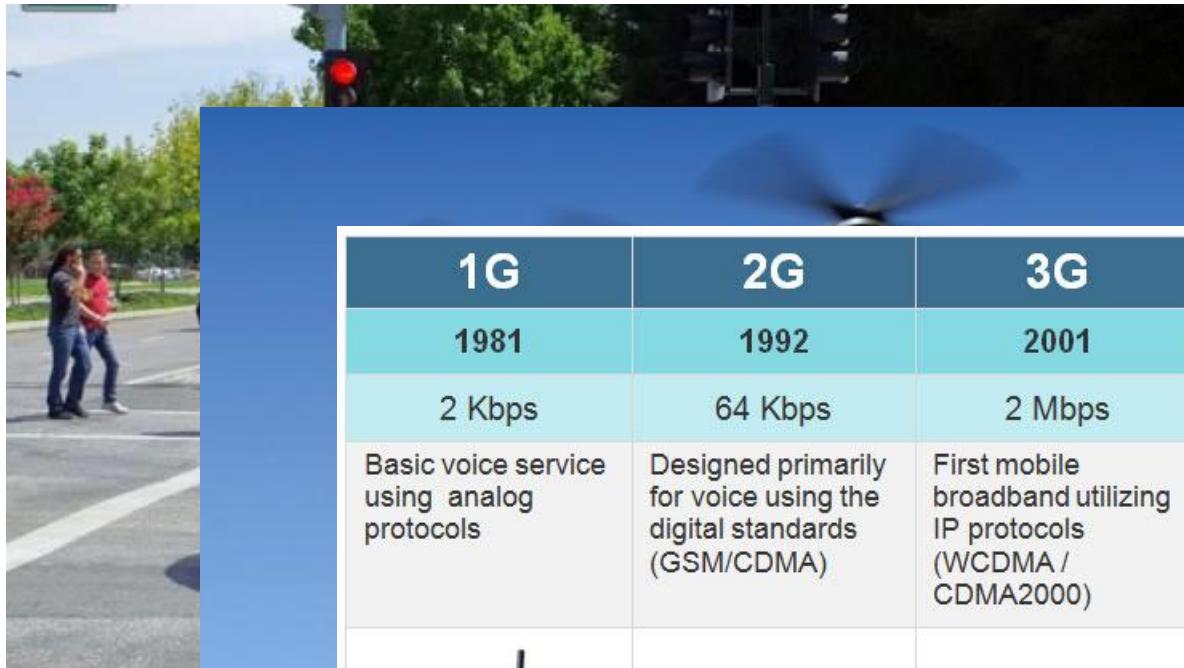
High performance
Low power dissipation
Wireless capability
etc...



More portable, wearable, and more powerful devices
for ubiquitous and pervasive computing ...

Industry Trends (3/3)

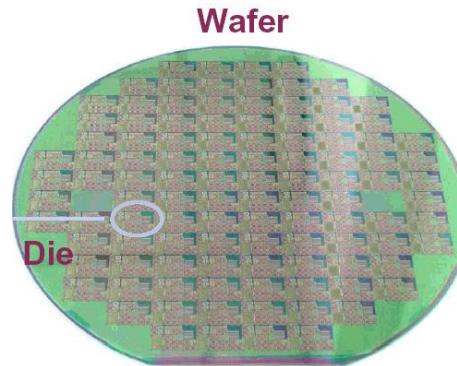
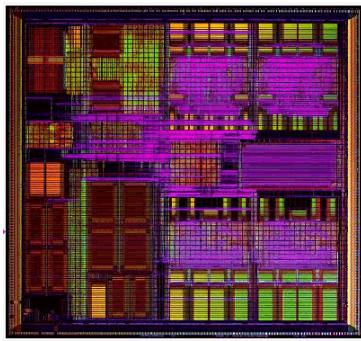
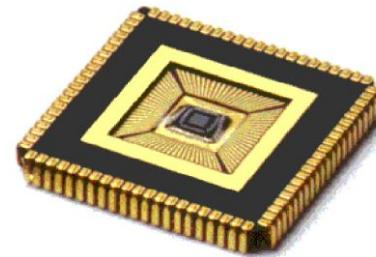
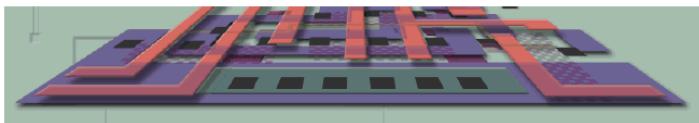
- IoT (Internet of Things)
 - 5G wireless networks + AI



| 1G | 2G | 3G | 4G | 5G |
|--|---|--|---|---|
| 1981 | 1992 | 2001 | 2010 | 2020(?) |
| 2 Kbps | 64 Kbps | 2 Mbps | 100 Mbps | 10 Gbps |
| Basic voice service using analog protocols | Designed primarily for voice using the digital standards (GSM/CDMA) | First mobile broadband utilizing IP protocols (WCDMA / CDMA2000) | True mobile broadband on a unified standard (LTE) | 'Tactile Internet' with service-aware devices and fiber-like speeds |
| | | | | |

IC Design and Manufacture

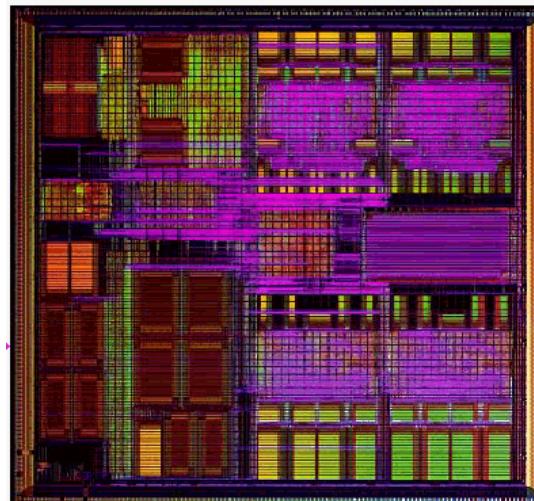
Design → Mask → Fabrication → Package → Testing



VLSI Design



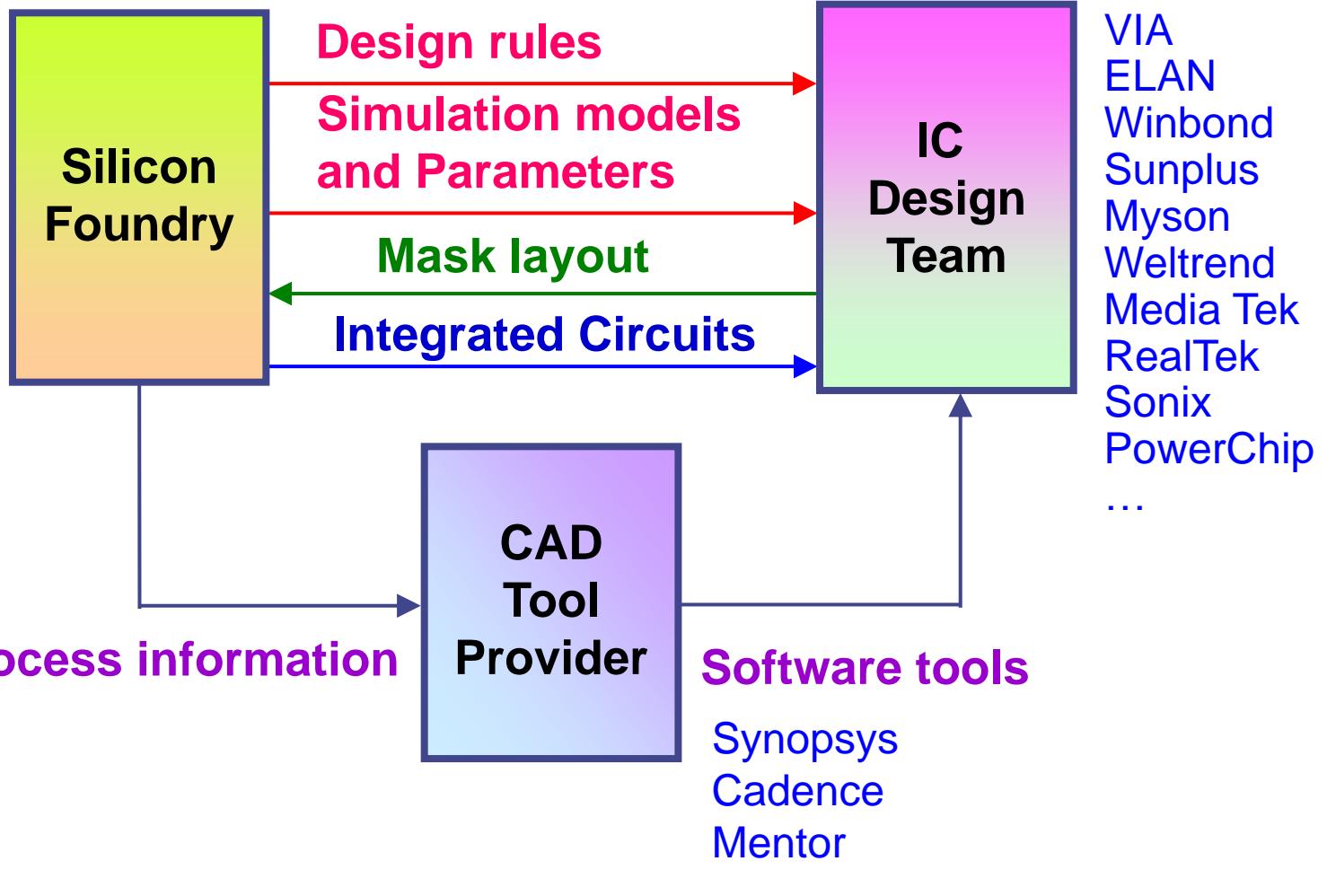
- Creation of a VLSI circuit involves a large number of activities
 - ◆ logic design, modeling, simulation, testing, and fabrication
 - ◆ a broad spectrum of knowledge ranging is required
- Rapid advancement of manufacturing capability has apparently enabled the growth of integration density
 - ◆ the methodology specifically developed for the design of VLSI circuit



VLSI Design Methodology

- Developed in the late 1970s released the **IC designers** from the details of semiconductor manufacturing
- Silicon foundry, IC design team, and CAD tool provider
 - ◆ **Design Rules**: a set of dimensional rules (e.g., minimum line widths, minimum spacings, etc.) determined by considering the physical limitations of a fabrication process
 - ◆ **Simulation Models** of the active and passive devices fabricated by a certain process are developed and made available to the logic designers
 - ◆ **Mask Layouts**: represent how components and connections should be formed on a semiconductor wafer
- Divide-and-conquer: the most important design principle
- **Hierarchical design methodology**

TSMC
UMC
...



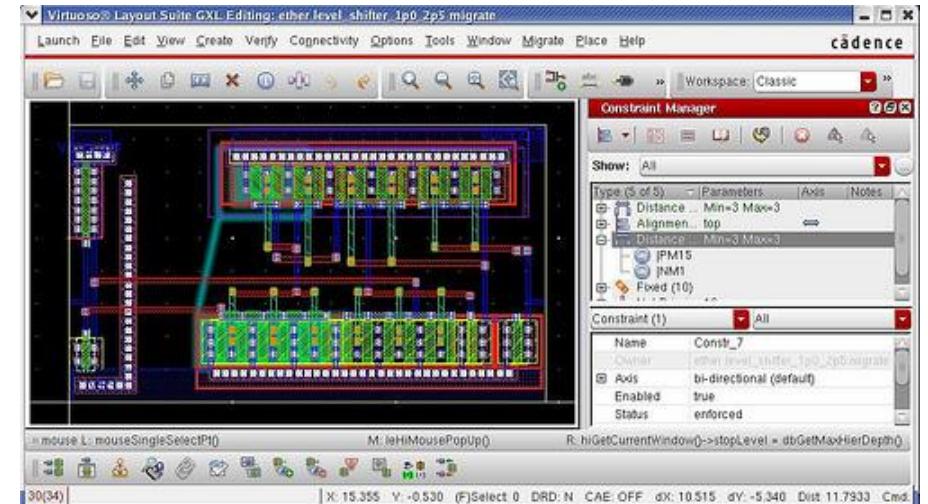
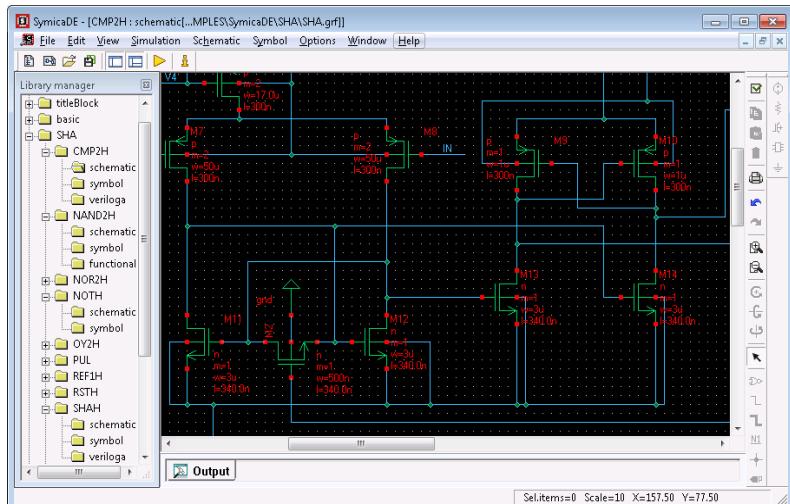
Relationship between a silicon foundry, IC design team,
and a CAD tool provider



Fabrication



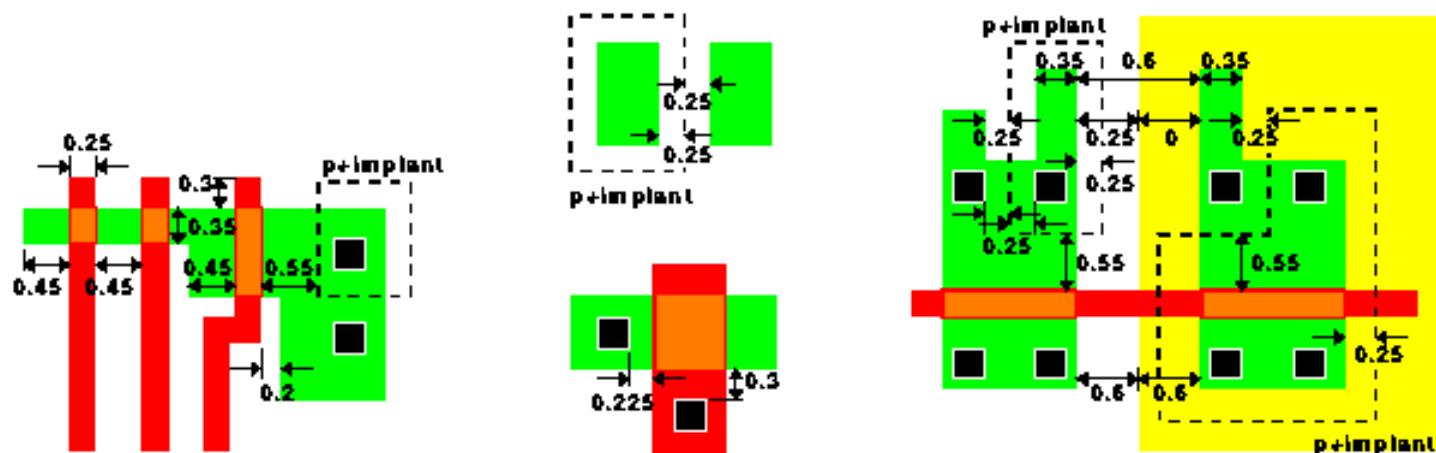
Design



CAD tools

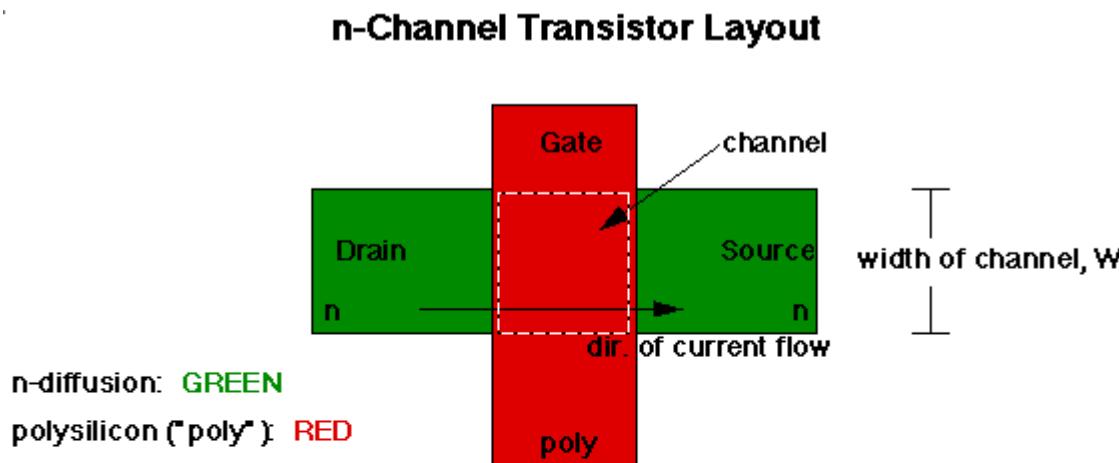
Design Rules

- A set of **dimensional rules** (e.g., minimum line widths, minimum spacings, etc.) determined by considering the physical limitations of a fabrication process

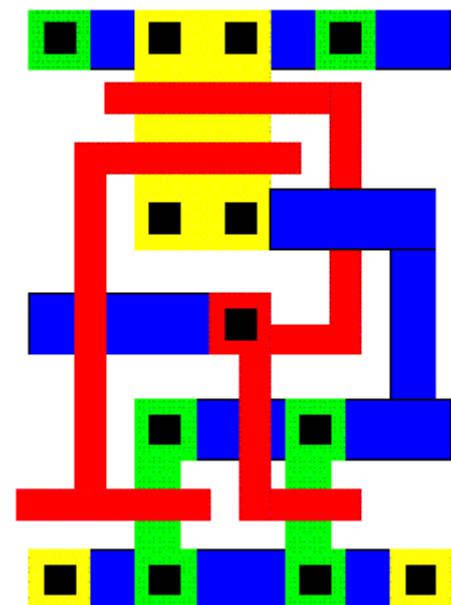
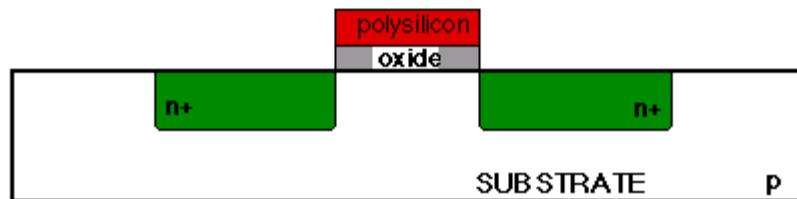


Mask Layout

- Represent how components and connections should be formed on a semiconductor wafer



NOTE: Source and drain are physically indistinguishable



NOR gate layout



Technology (1/4)

- Modern IC design methodology separates the designers from the semiconductor processing tasks
- No urgent need to select a specific IC fabrication technology at the initial phase of the design
- A general knowledge of the fabrication process is sufficient at this point
- Major materials to build integrated circuit
 - ◆ silicon (Si), germanium (Ge), and gallium arsenic (GaAs)
- Two types of silicon based device
 - ◆ the bipolar junction transistor (**BJT**) and the metal-oxide-semiconductor field-effect transistor (**MOSFET**)

Technology (2/4)

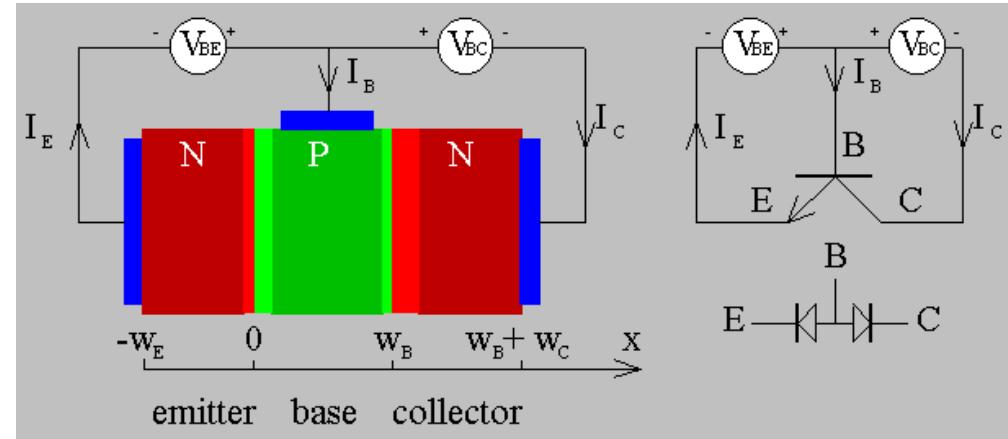
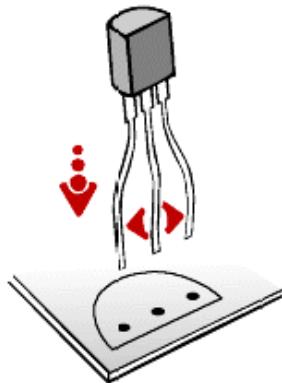
■ Complementary MOSFET (CMOS) technology

- ◆ employ both n-channel and p-channel transistors to form logic circuits
- ◆ dominate digital logic ICs

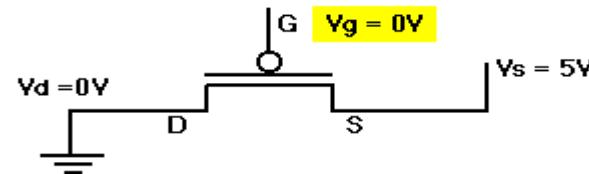
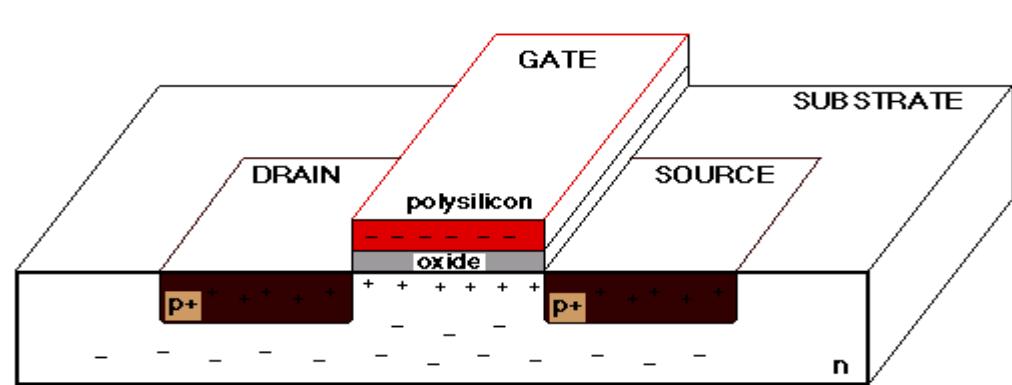
■ CMOS technology has the following advantages over the bipolar technology

- ◆ the nature of a CMOS circuit allows it to operate with **low power**
 - ⊕ a higher integration density is possible
- ◆ a MOSFET occupies a **smaller area** than a BJT
 - ⊕ higher circuit density
- ◆ a MOSFET has a very high input impedance and can be modeled as a **switch**
 - ⊕ simplify the design and analysis of CMOS logic circuits
- ◆ CMOS circuits have the **largest logic swings** and thus excellent noise margins

BJT:



MOSFET:



Technology (3/4)

Logic circuits based on BJTs

- ◆ transistor-transistor logic (TTL) and emitter-coupled logic (ECL)
- ◆ advantages: **high speed** and **large current capability**

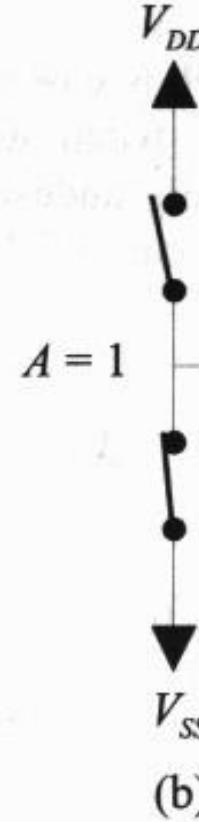
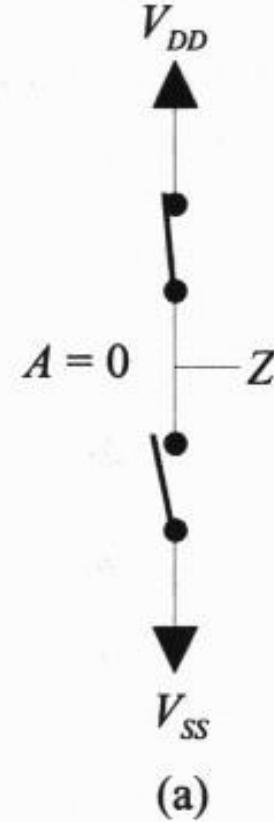
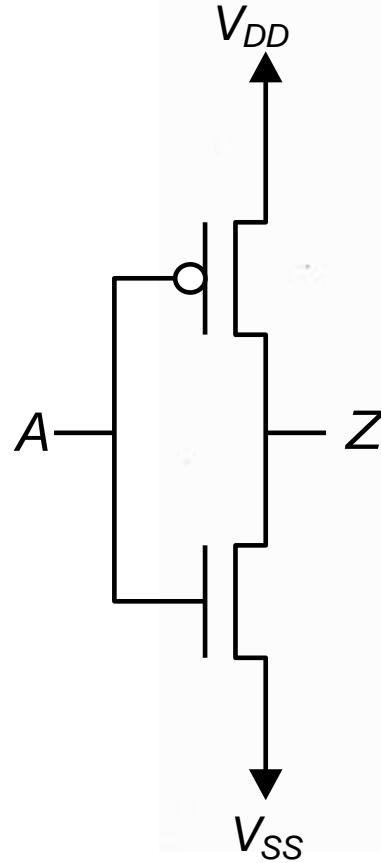
BiCMOS technology

- ◆ BJTs and MOSFETs coexists in the same circuit
- ◆ offer the **high speed** of BJTs and the **low power** advantages of CMOS technology

CMOS technology

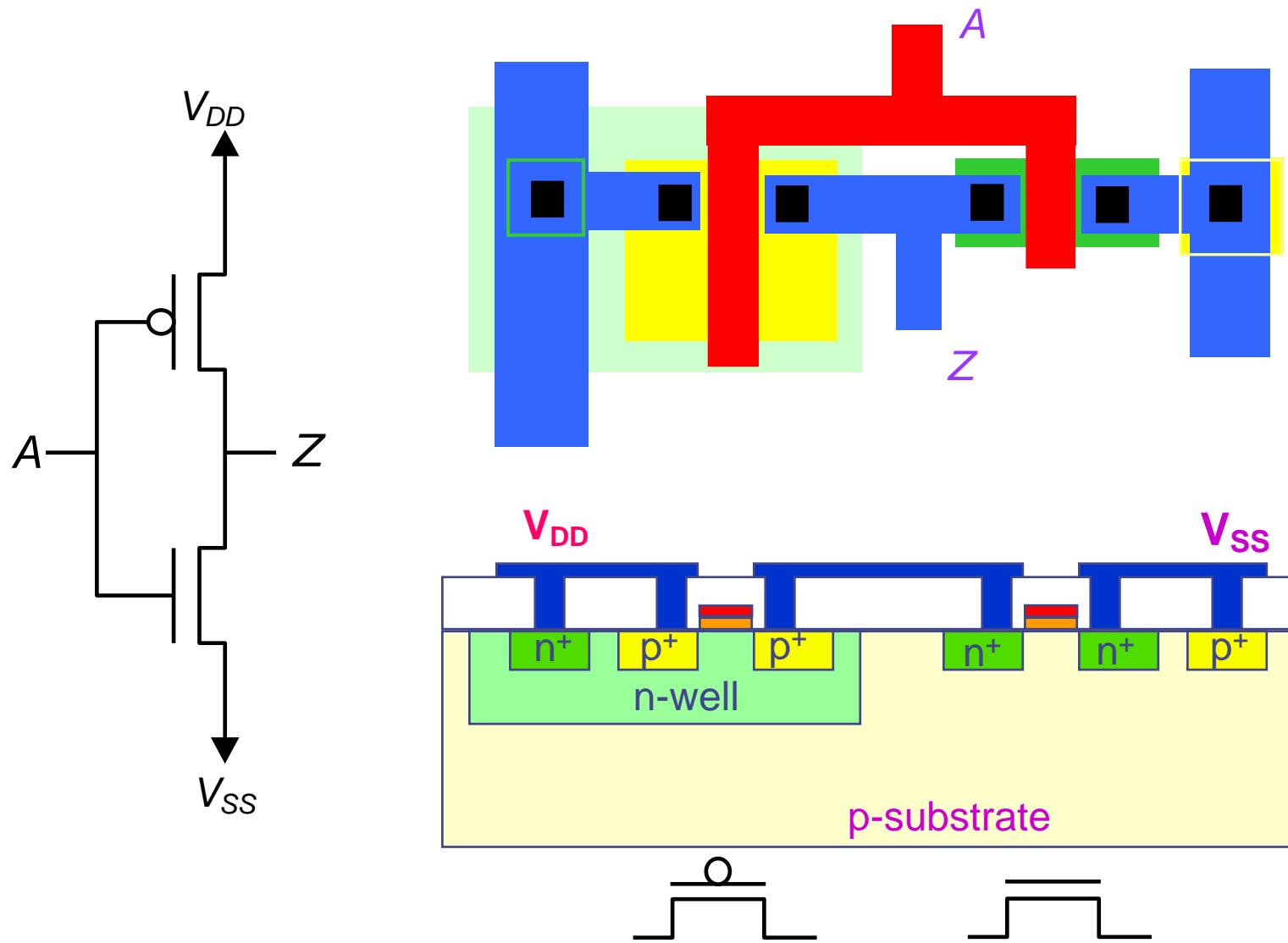
- ◆ n-channel and p-channel MOSFETs to coexist on the same substrate
- ◆ **n-well CMOS technology:**
 - ⊕ p-type wafer as the substrate in which n-channel transistor can be formed
 - ⊕ p-channel transistors are formed in a n-well
- ◆ p-well CMOS technology
- ◆ twin-well CMOS technology

Basic CMOS gate: Inverter



Switch network implementing an inverter

Basic CMOS gate: Inverter



Basic CMOS gate: NAND

Description

$Y = 1$ if A is 0 or B is 0

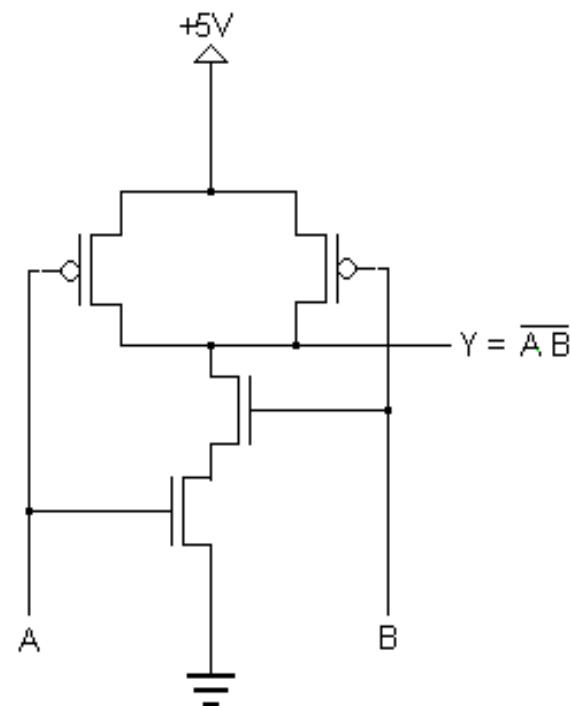
Gate



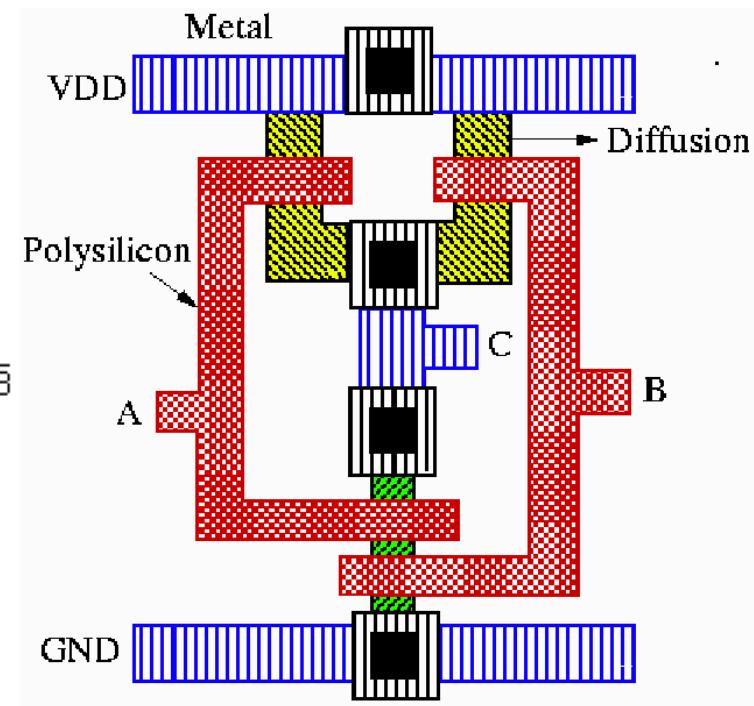
Truth Table

| A | B | Y |
|---|---|---|
| 0 | 0 | 1 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

MOS Transistors



Schematic view



Layout view

Basic CMOS gate: NOR

Description

$Y = 1$ if both A and B are 0

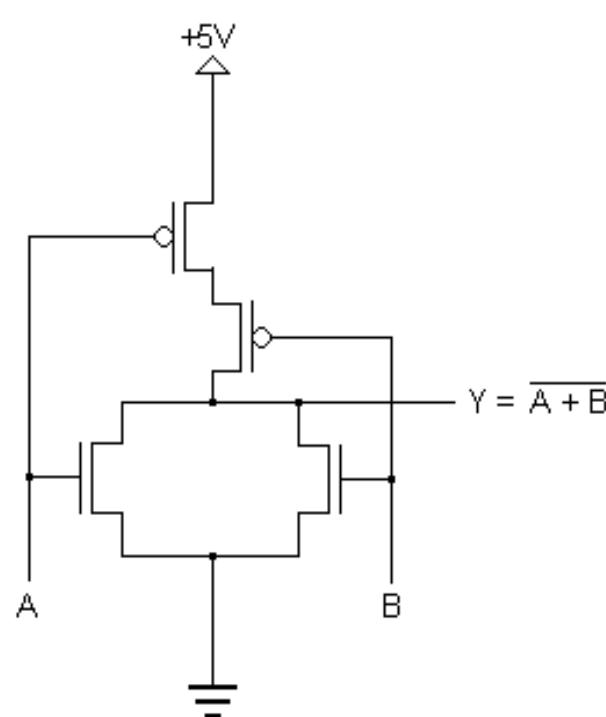
Gate



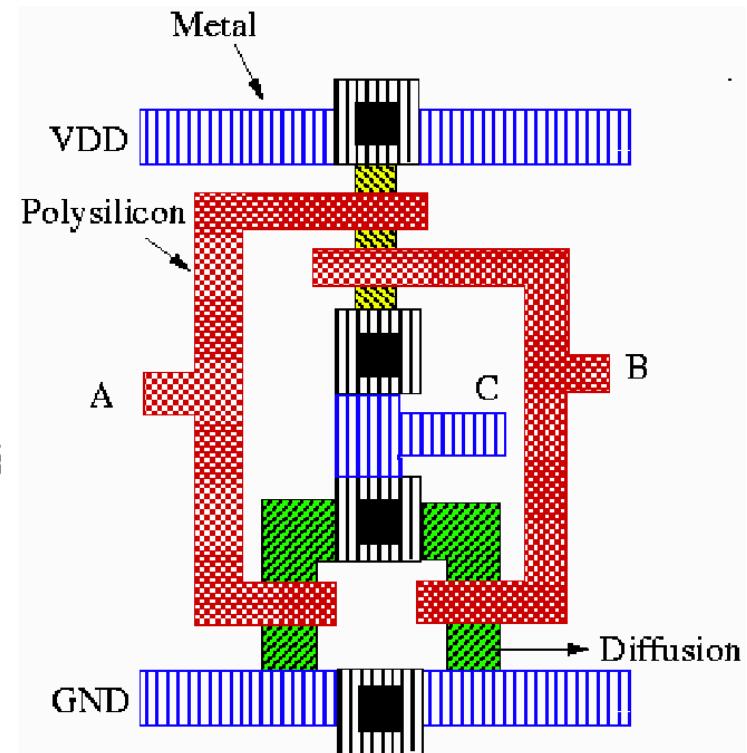
Truth Table

| A | B | Y |
|---|---|---|
| 0 | 0 | 1 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 0 |

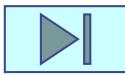
MOS Transistors



Schematic view



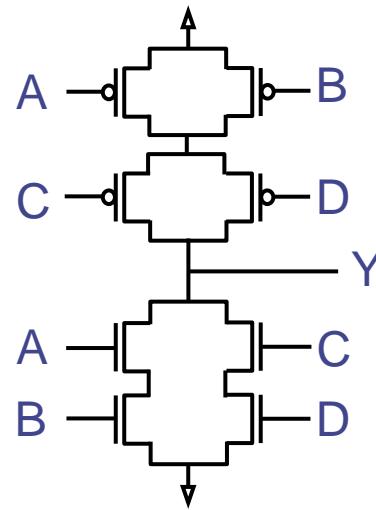
Layout view



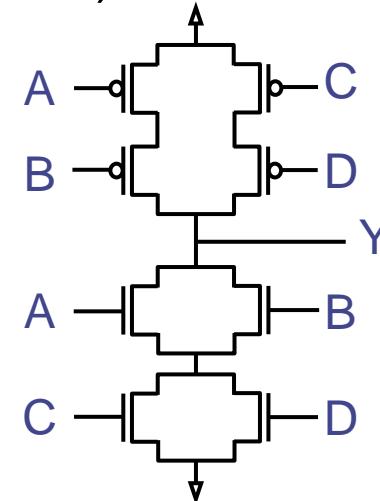
Complex Gates



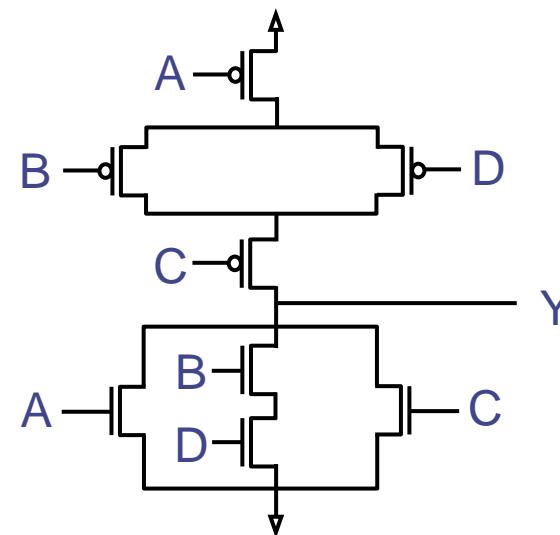
■ $Y = \overline{AB+CD}$



■ $Y = \overline{(A+B)(C+D)}$



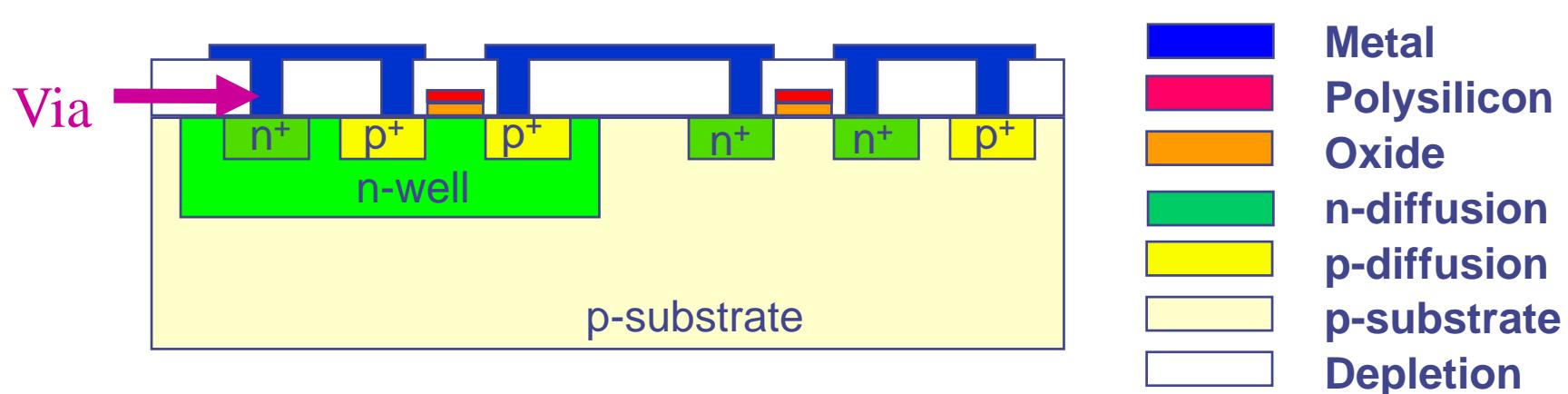
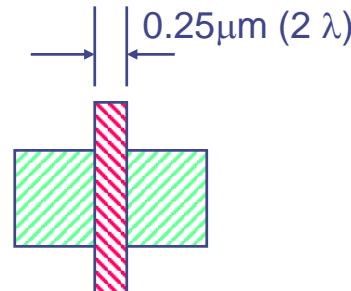
■ $Y = \overline{A+BD+C}$



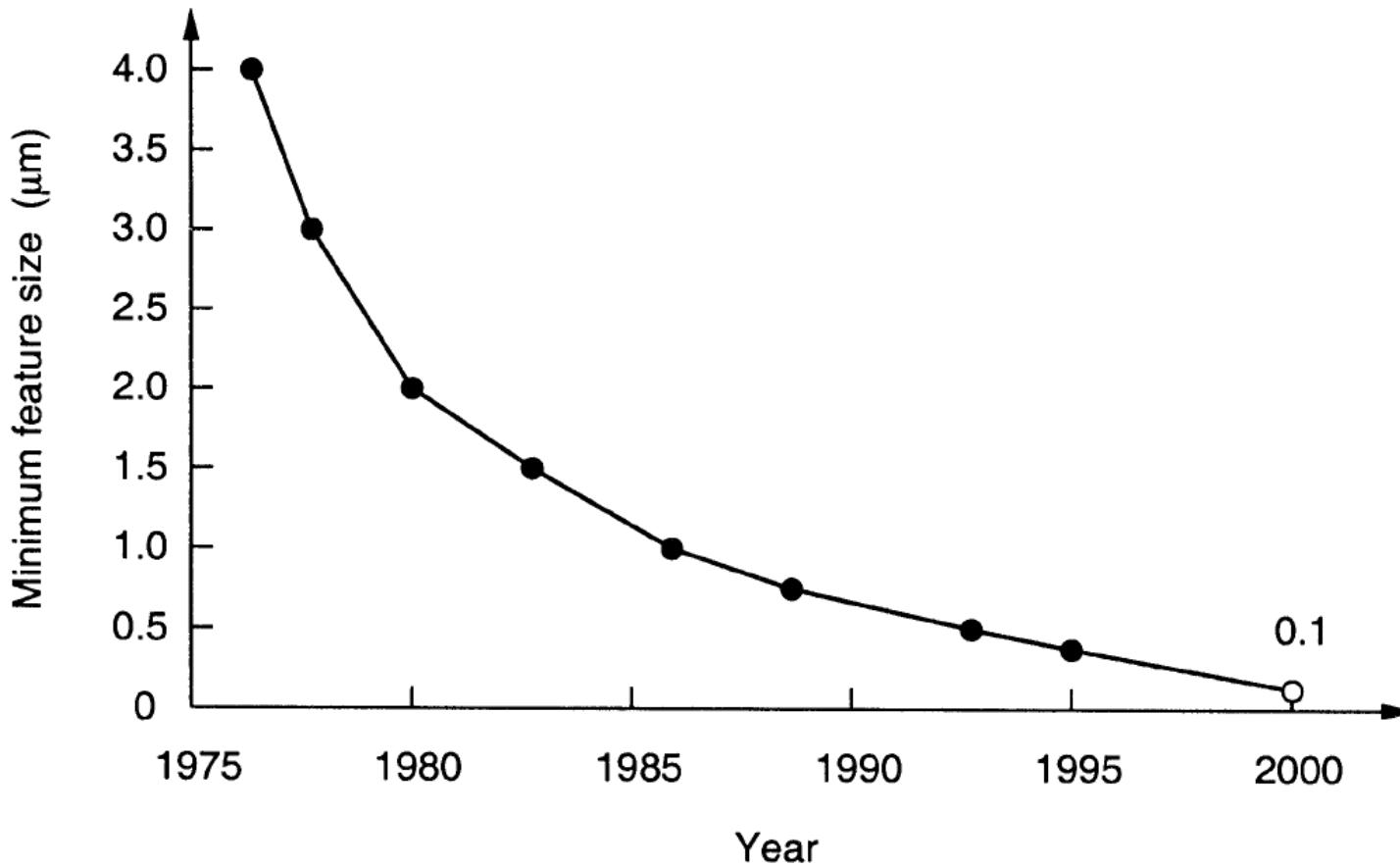
Technology (4/4)

- Two important properties of a selected CMOS technology
 - ◆ **minimum manufacturable feature size**, number of available routing layers
- The minimum feature size determines the shortest transistor channel length possible
 - ◆ the shortest transistor channel length is about $0.25 \mu\text{m}$ in a $0.25 \mu\text{m}$ -CMOS technology
 - ◆ smaller feature size: faster transistors and a higher circuit density
- The number of available routing layers for component interconnects
 - ◆ routing layers insulated from each other by silicon dioxide (SiO_2) allow a wire to cross over another line without making undesired contacts
 - ◆ **vias**: connecting paths between two routing layers typical CMOS process: 3 to 7 routing layers

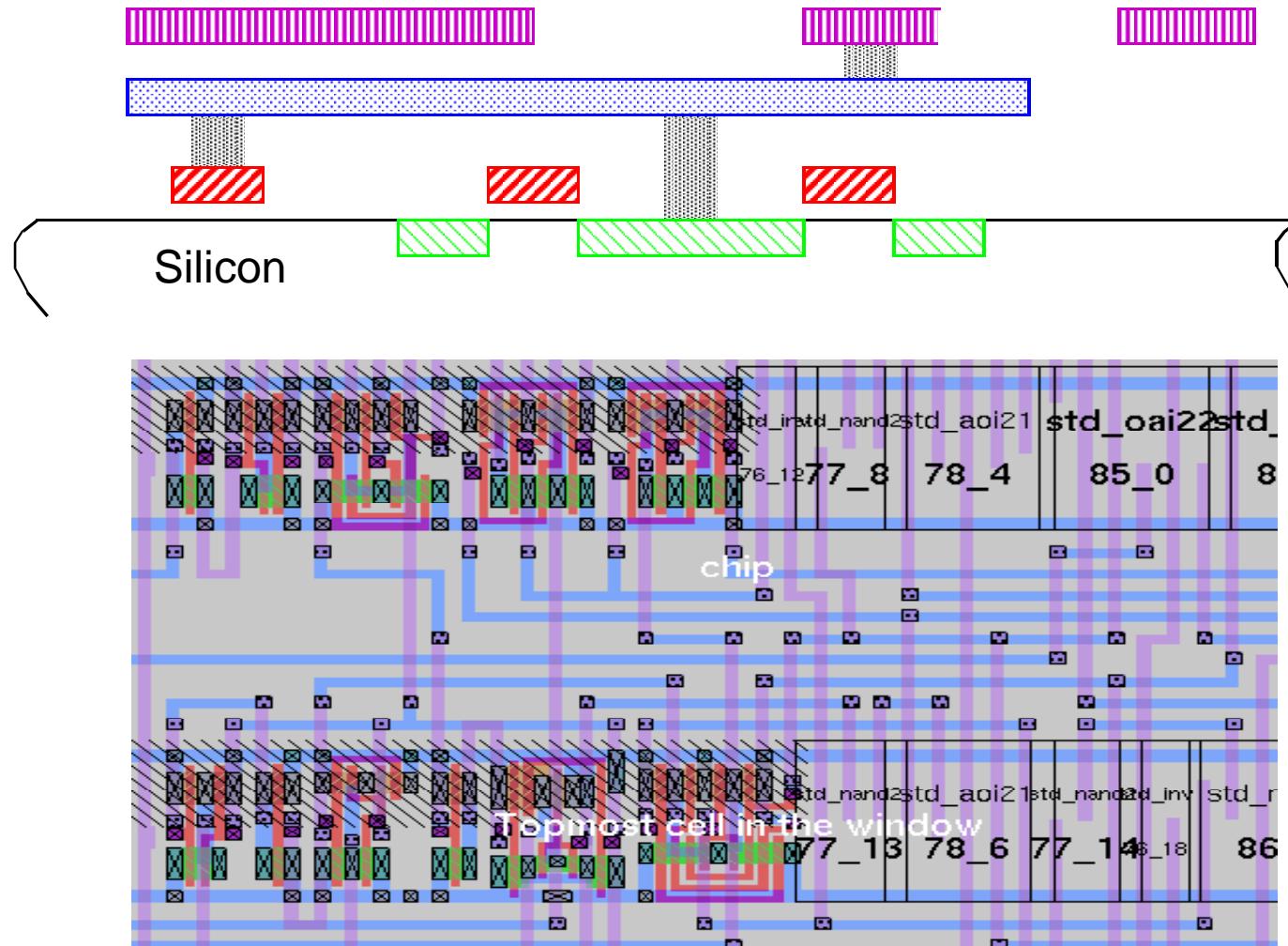
Minimum feature size



Evolution of Minimum Feature Size

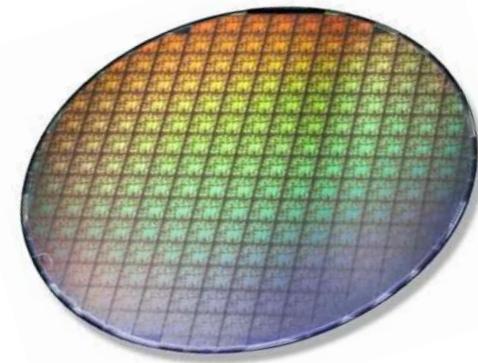


Layer and Via



VLSI Design Problem (1/2)

- Realize a given **specification** on **silicon**
- Optimize the following entities
 - ❖ Area (less silicon, high yield)
 - ❖ Speed (design constraint)
 - ❖ Power dissipation (cooling, battery)
 - ❖ Design time (CAD tools)
 - ❖ Testability (minimize the time spent to test a single chip)
- Combine all these entities into a single cost function
 - ◆ The complexity is too high
 - ◆ Optimization cannot be done in one step
 - ◆ **Partition** problem and optimize subproblems



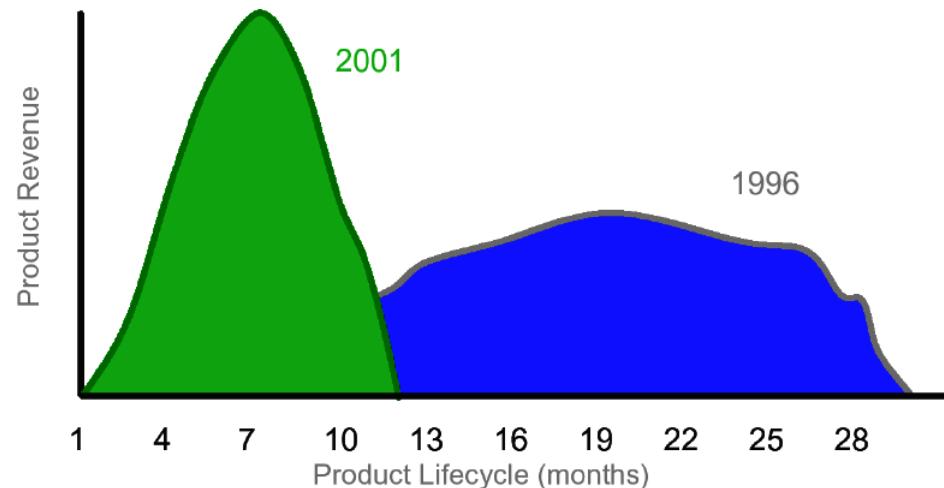
VLSI Design Problem (2/2)



Chip complexity

| <u>Year</u> | <u>Technology</u> | <u>Chip Complexity</u> | <u>Frequency</u> | <u>Staff</u> | <u>Staff Cost*</u> |
|-------------|-------------------|------------------------|------------------|--------------|--------------------|
| 1997 | 250 nm | 13 M Tr. | 400 | 210 | 90 M |
| 1998 | 250 nm | 20 M Tr. | 500 | 270 | 120 M |
| 1999 | 180 nm | 32 M Tr. | 600 | 360 | 160 M |
| 2002 | 130 nm | 130 M Tr. | 800 | 800 | 360 M |

Products have a shorter life-cycle



Hierarchical Design Methodology (1/2)



■ Hierarchy

- ◆ Something is composed of simpler things
- ◆ Show the structure of a design at different levels of description

■ Abstraction

- ◆ Hide the lower level details when looking at a certain level

■ This decomposition

- ◆ Continues until the basic building blocks of a VLSI circuit

■ Regularity

- ◆ The hierarchical decomposition of a large system should result in not only simple, but also similar blocks, as much as possible
- ◆ Regularity usually reduces the number of different modules that need to be designed and verified, at all levels of abstraction

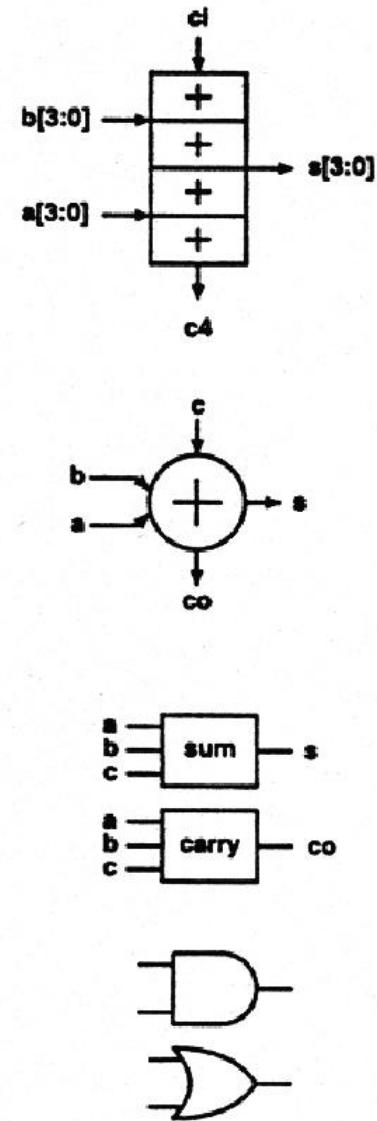
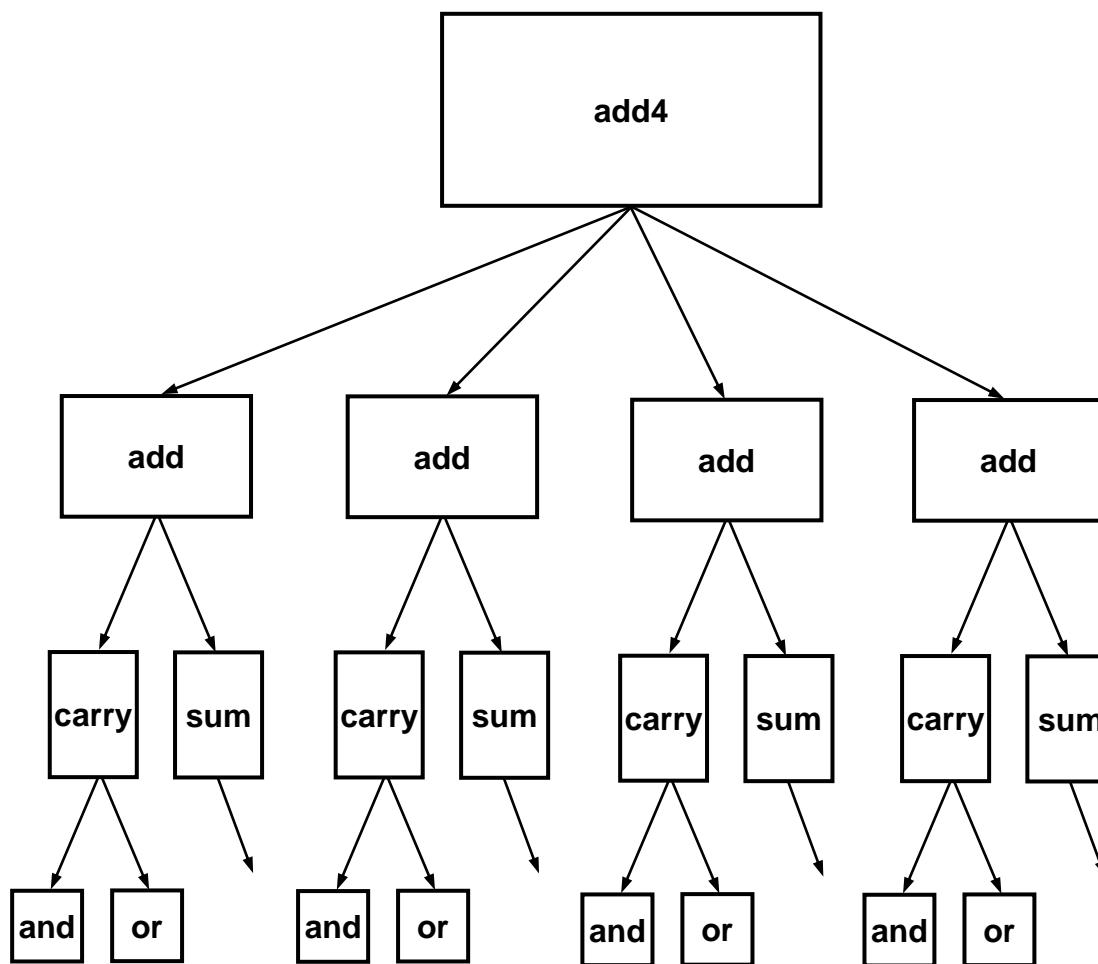
Hierarchical Design Methodology (2/2)

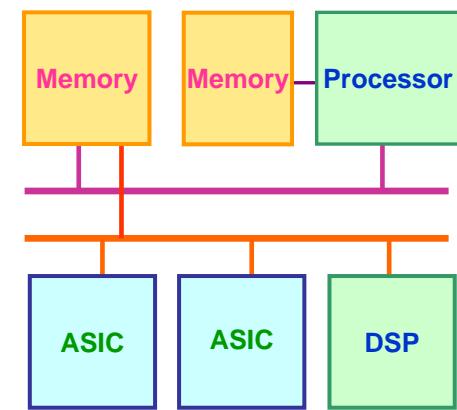
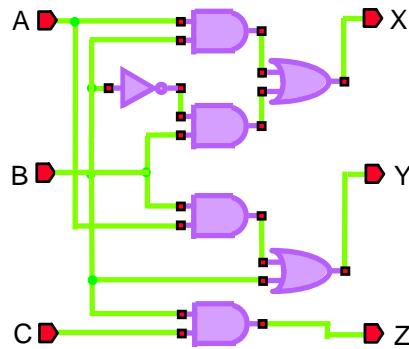
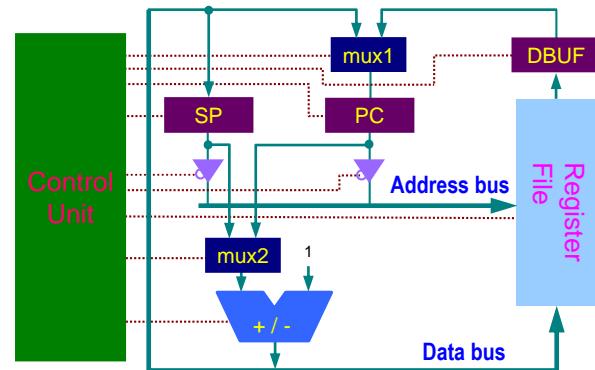
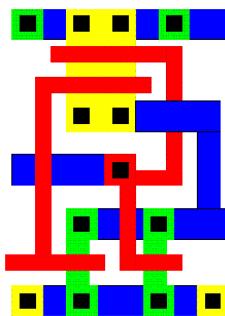
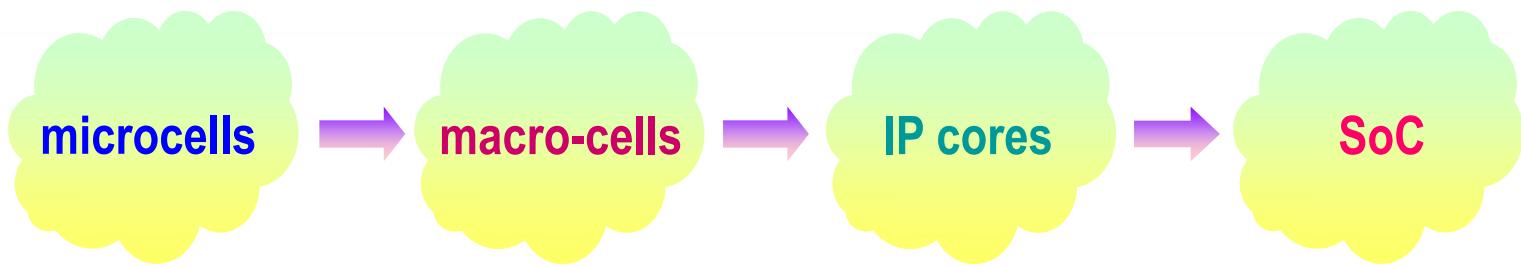


■ Design Methodology

- ◆ Determine the proper subset of abstraction levels, synthesis tasks, the order in which they are executed, and the type of computer-aided design (CAD) tools used in the design process
- ◆ Top-down
- ◆ Bottom-up
- ◆ Meet in the middle

Hierarchy of a 4-bit Carry Ripple Adder



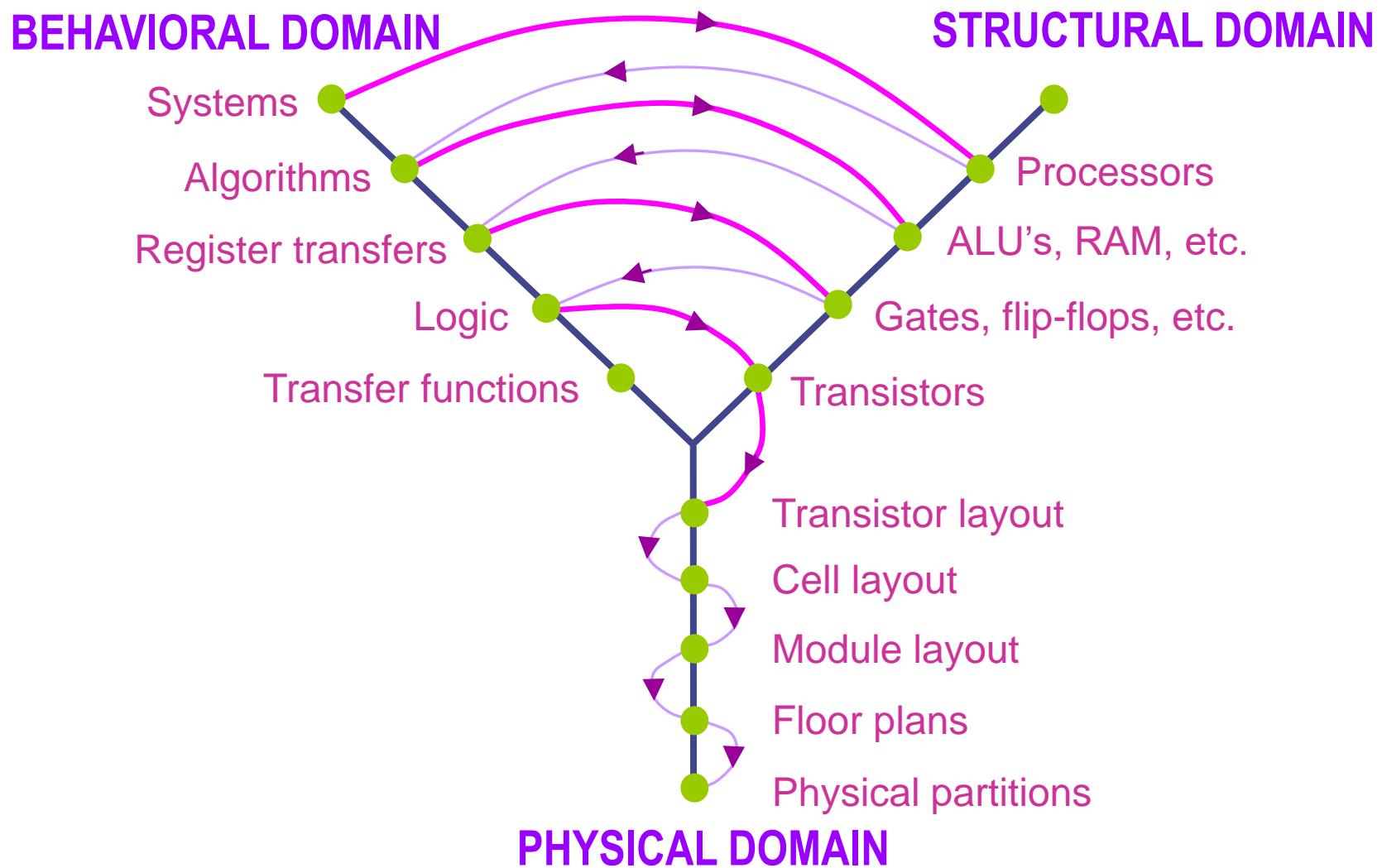




The Design Domains

- A single hierarchy is **not sufficient** to properly describe the VLSI design process
- Design Domains
 - ◆ **Behavioral** or functional representation
 - ❖ specifies the **behavior** or the **function** of a design without any implementation information
 - ◆ **Structural** representation
 - ❖ specifies the implementation of a design in terms of **components** and their **interconnections**
 - ◆ **Physical** representation
 - ❖ specifies the **physical characteristics** of the design and can be thought of as a **blueprint** for manufacturing
- Each design domain has its own hierarchy

Top-down Structural Design

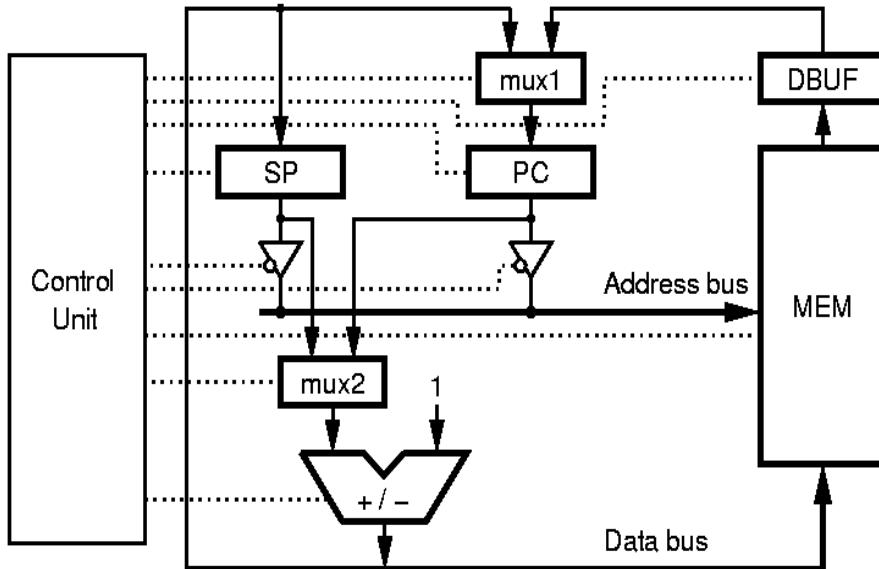


```

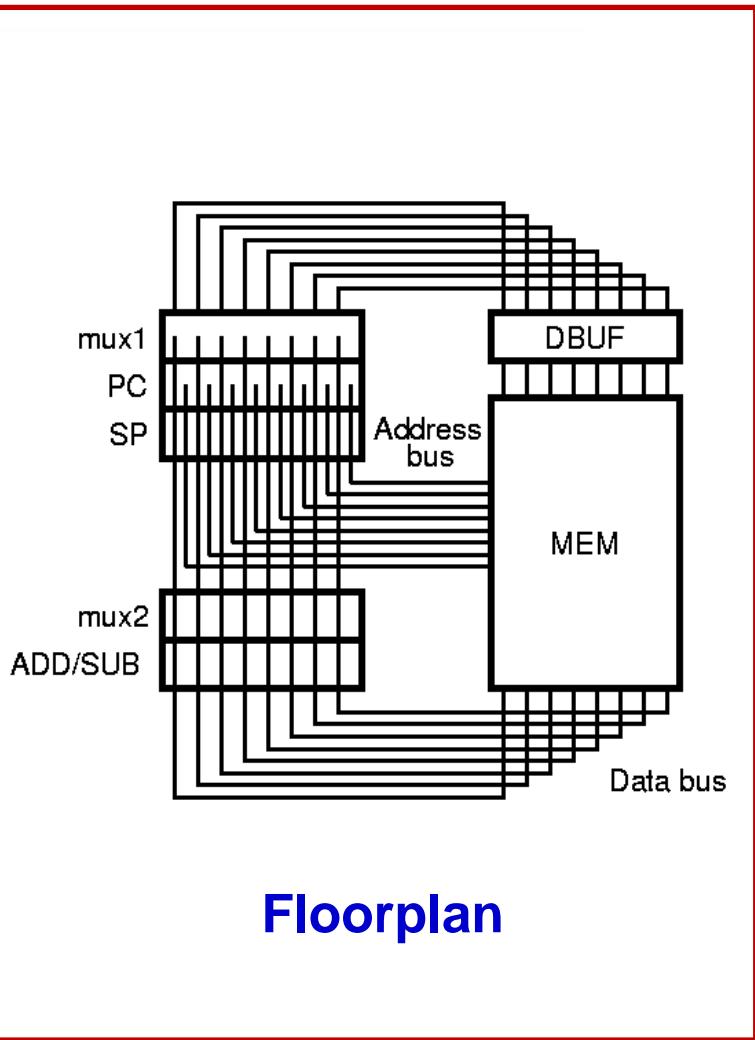
if IR(3) = '0' then
    PC      := PC + 1;
else
    DBUF   := MEM(PC);
    MEM(SP) := PC + 1;
    SP     := SP - 1;
    PC      := DBUF;
end if;

```

Behavior



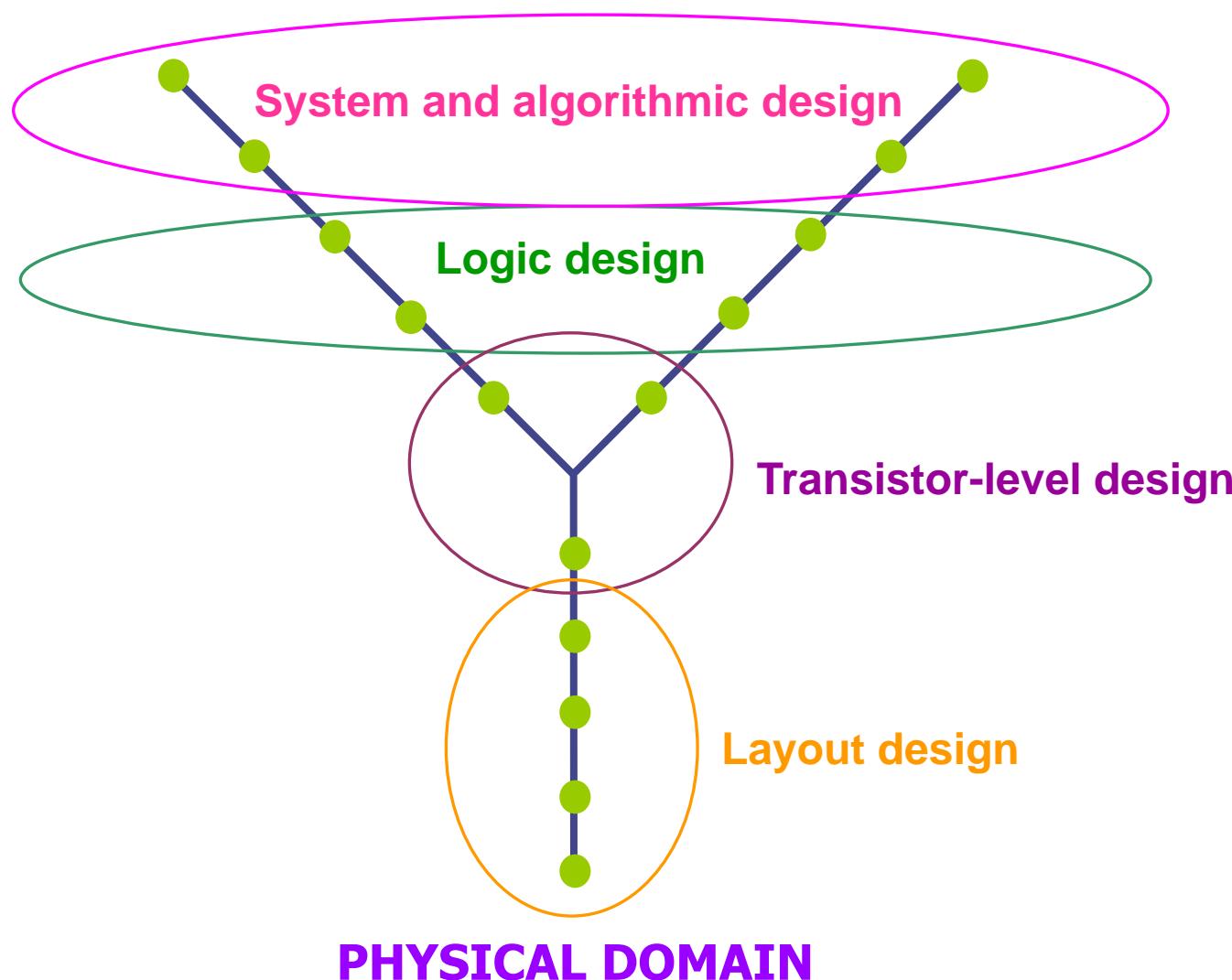
Structure



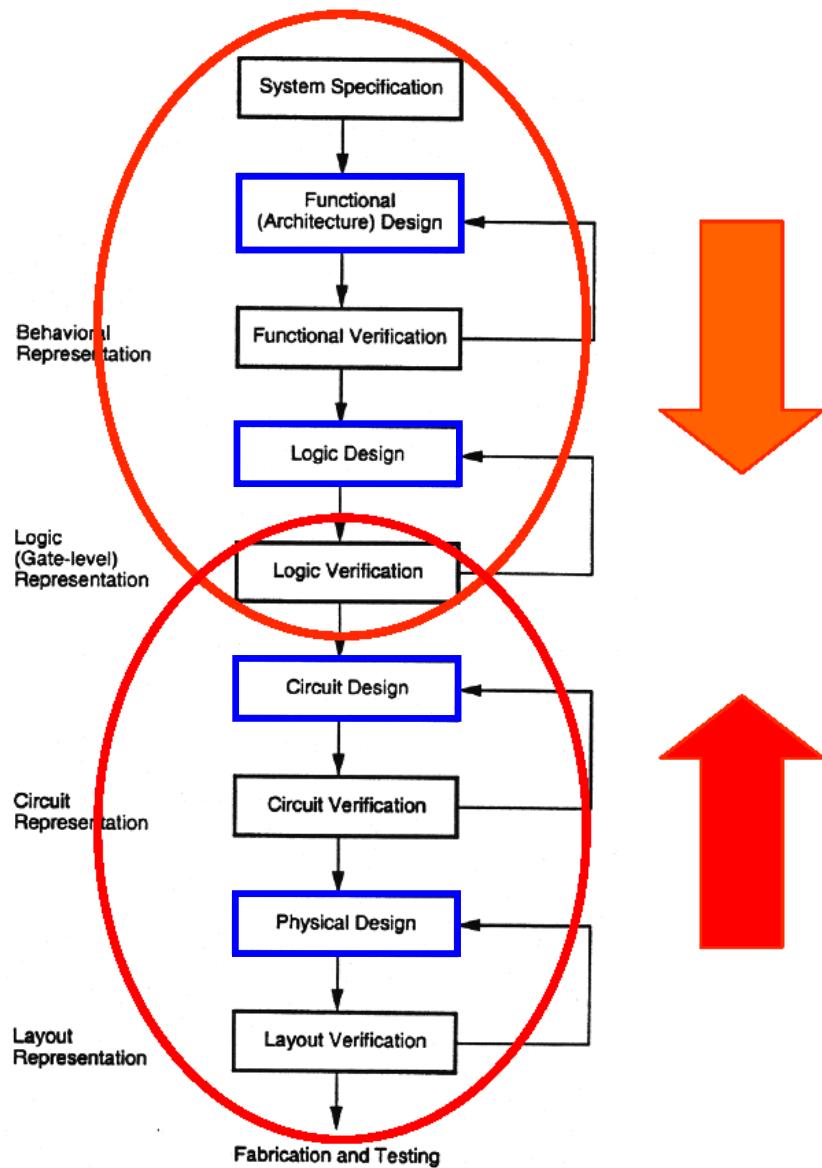
Floorplan

BEHAVIORAL DOMAIN

STRUCTURAL DOMAIN



Meet in the Middle

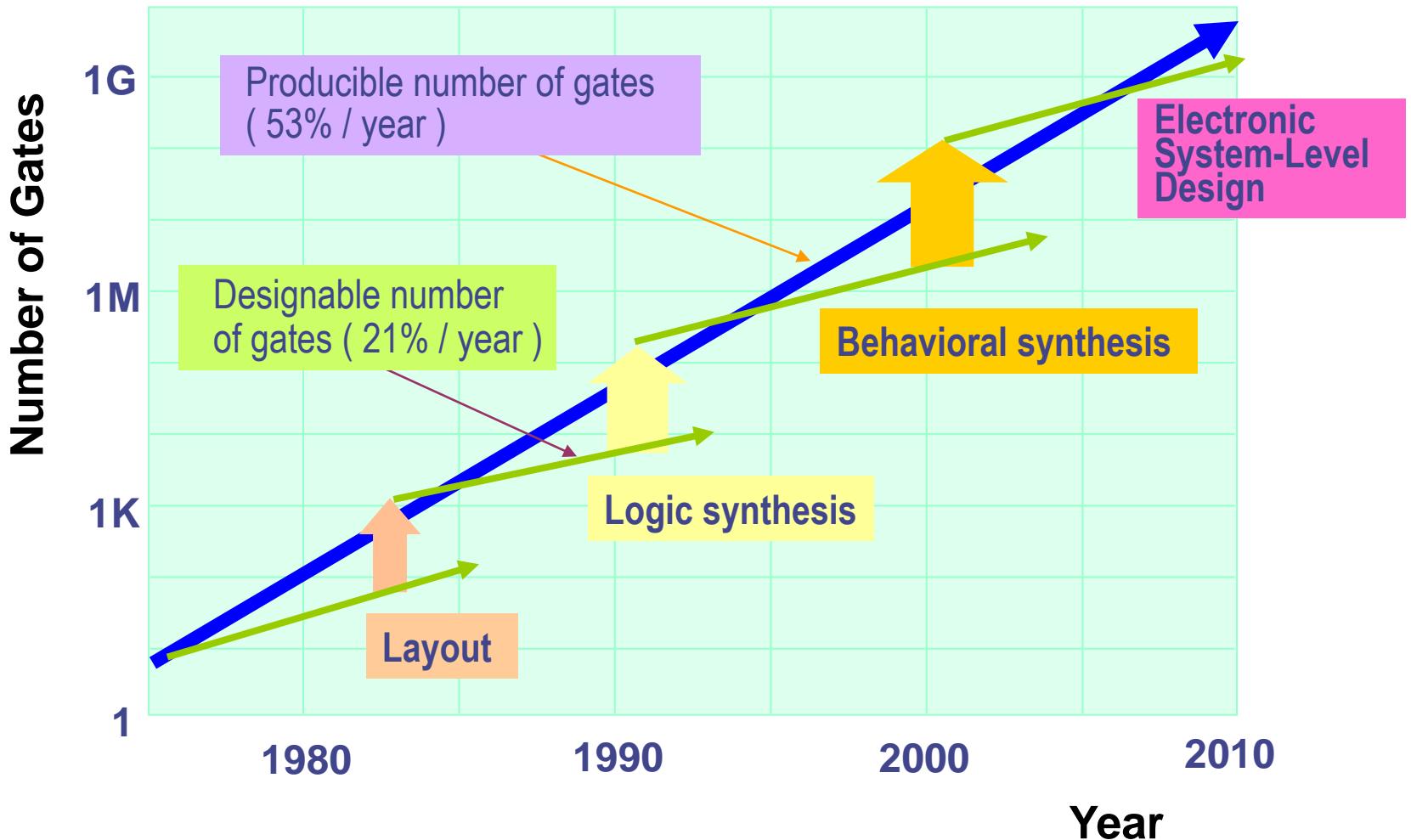


Simplified VLSI Design Flow

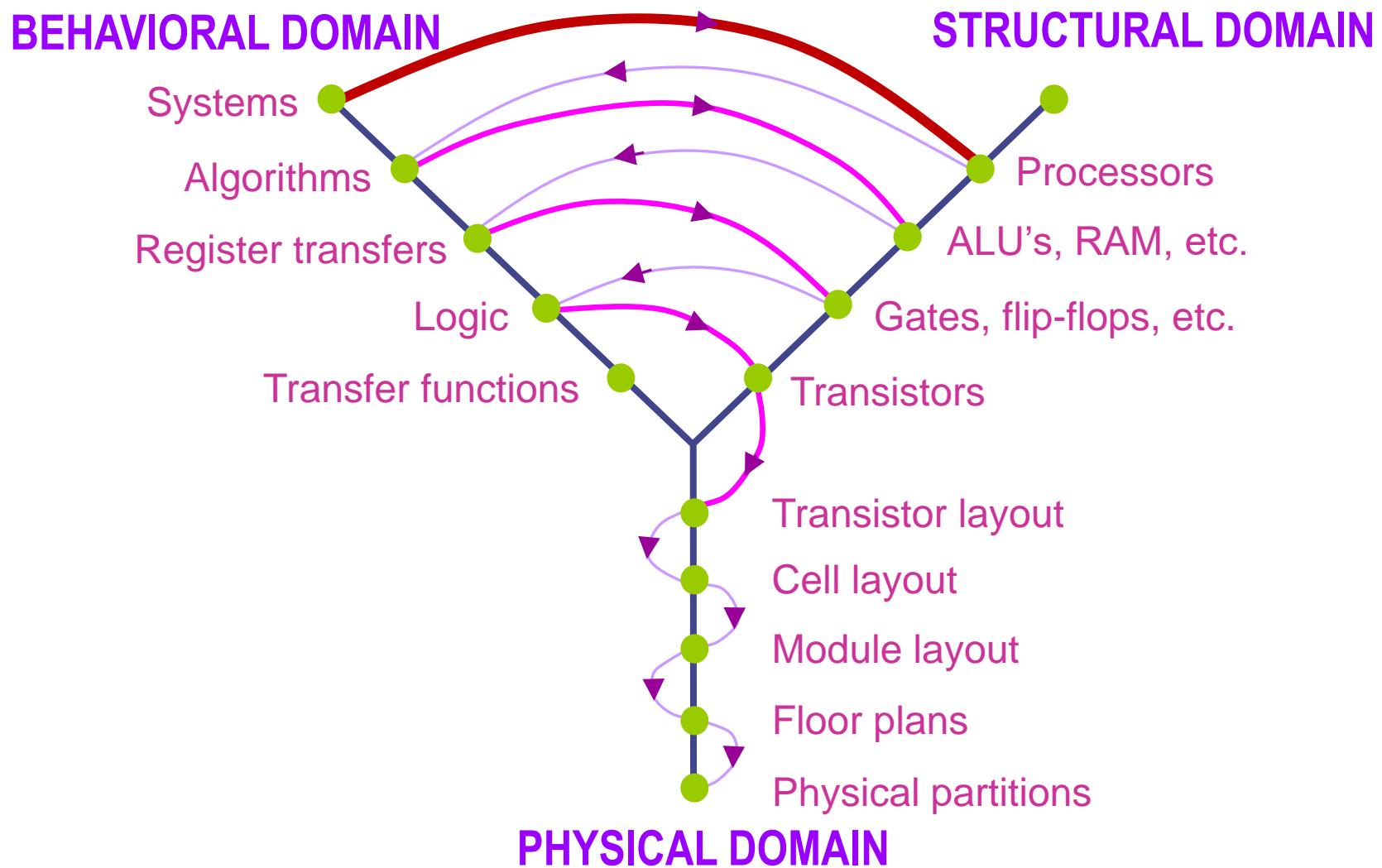
Top-down

Bottom-up

Electronic Design Automation (EDA) tools



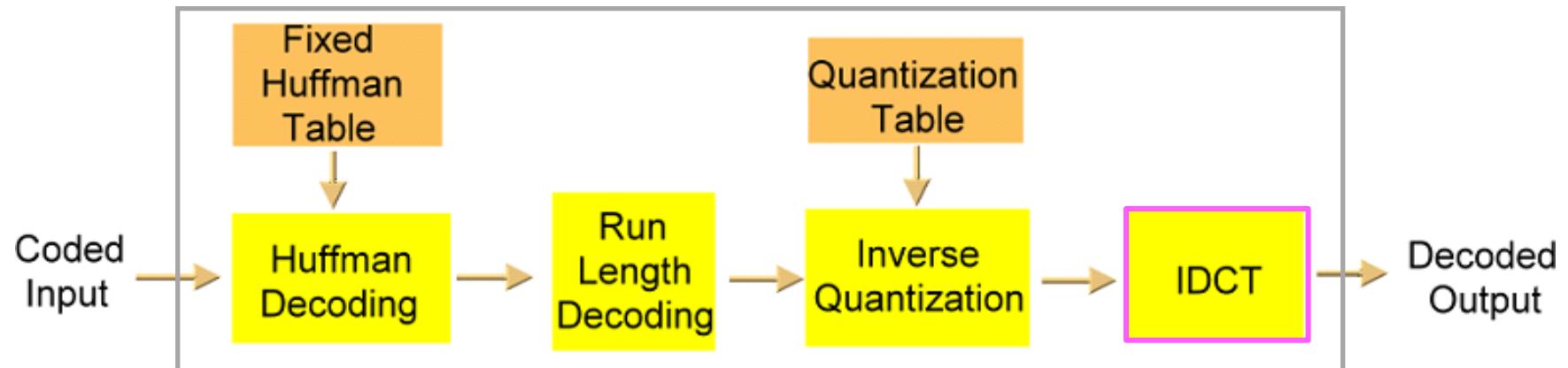
Top-down Structural Design



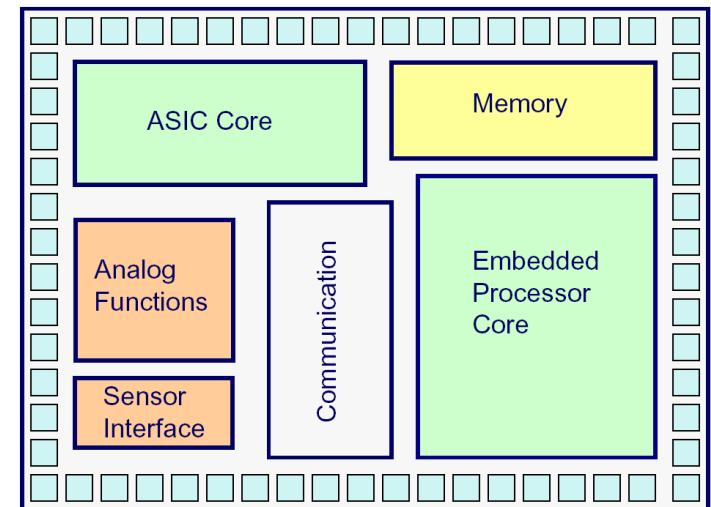
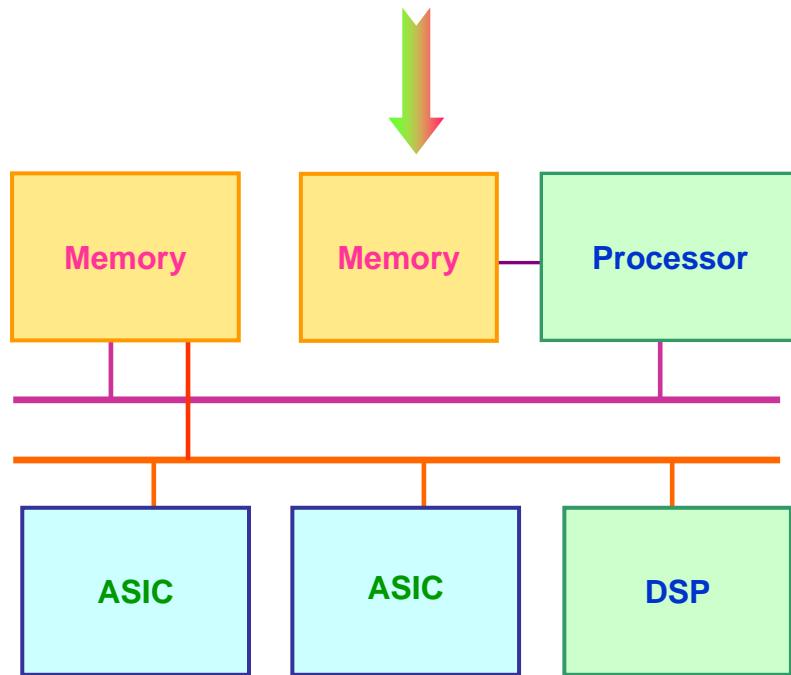


System-level Design

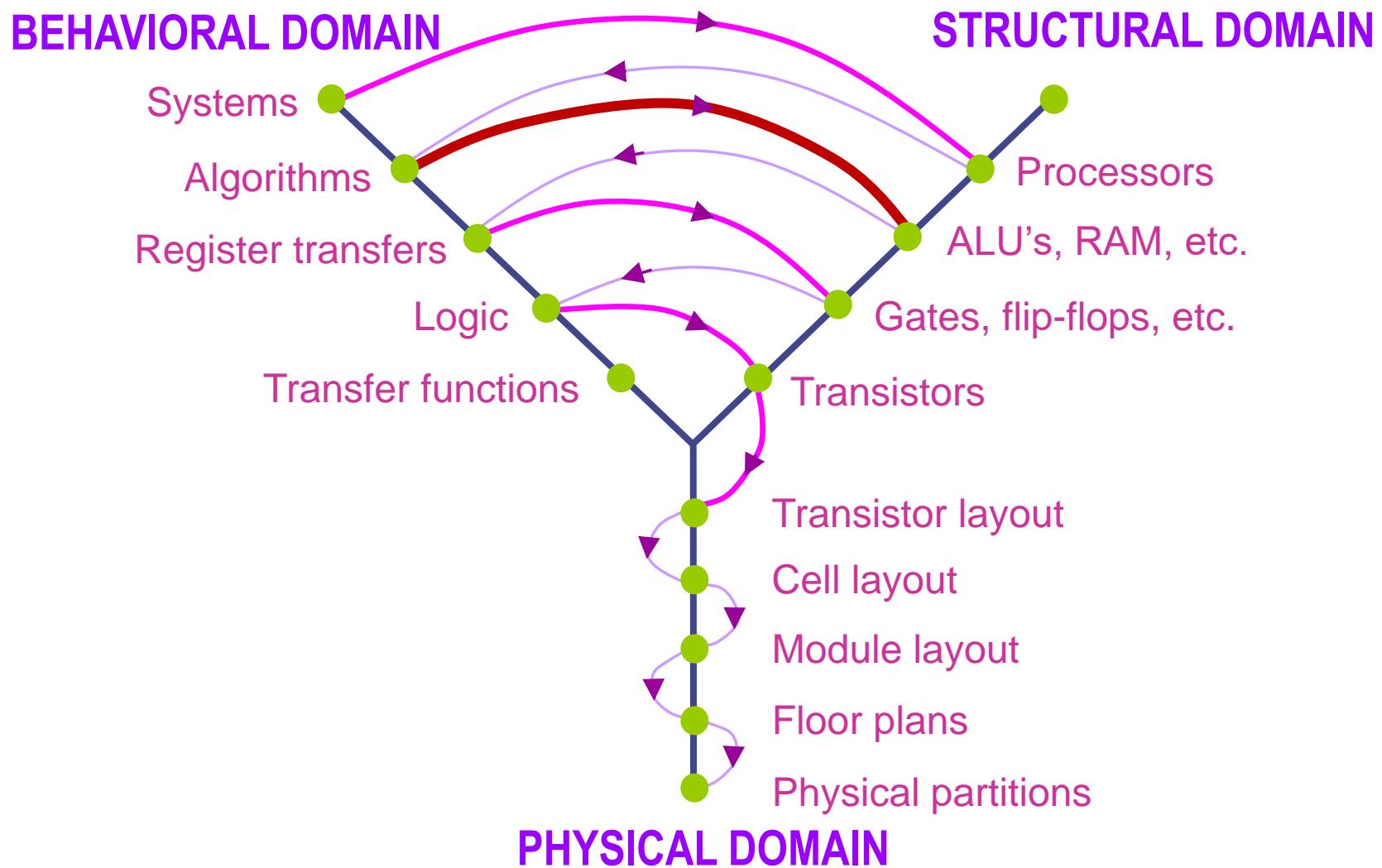
- Design starting from a **system specification**
 - ◆ Normally not result in a single ASIC
- **Hardware-software co-design**
 - ◆ Partition the initial specification in hardware and software parts
 - ◆ The parts with the **highest frequencies** are the most likely to be realized in hardware
- Cost estimation, design-space exploration
- Code generation
 - ◆ Map the high-level descriptions of the software to the low-level instructions of the programmable hardware (a CAD problem)
- Hardware-software co-simulation
 - ◆ Verify the correctness of the result of co-design
 - ◆ Can cope simultaneously with descriptions of hardware and software



JPEG decoder

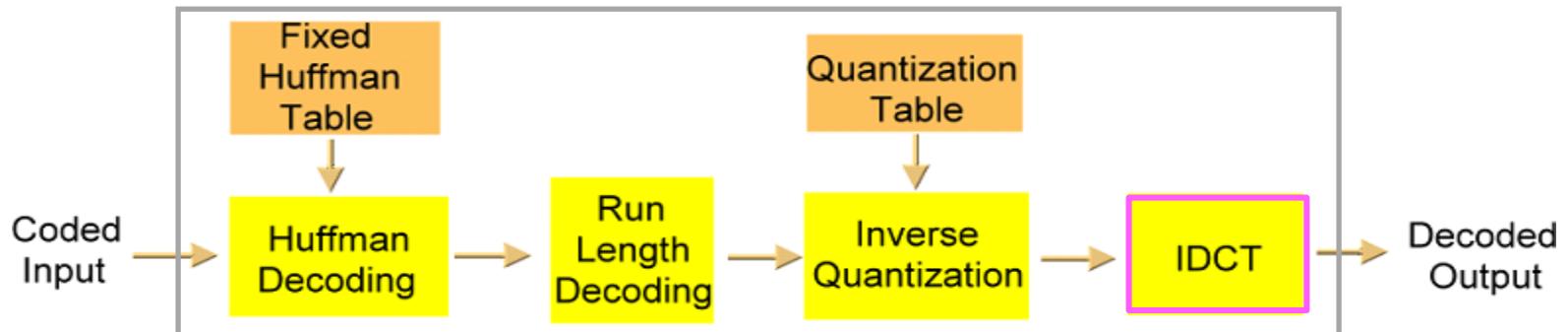


Top-down Structural Design



Algorithmic-level Design

- Specification (behavioral description)
 - ◆ General-purpose programming languages (e.g. C or Pascal, sequential execution)
 - ◆ Hardware description language (e.g. VHDL or Verilog, parallel execution)
 - ⊕ Simulation, automatic synthesis, formal verification
- High-level simulation
 - ◆ Help in the detection of errors in the specification
 - ◆ Allow the comparison of the highest-level description with more detailed versions of the design

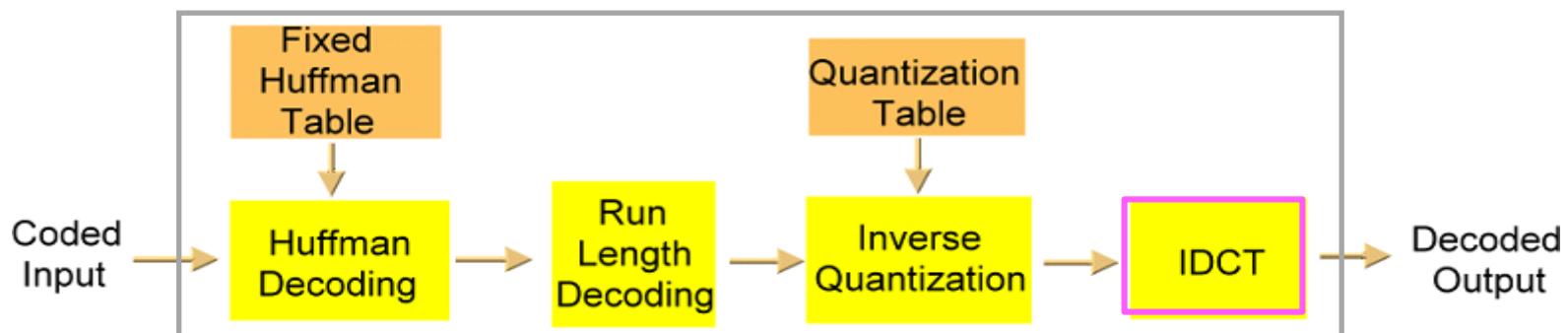
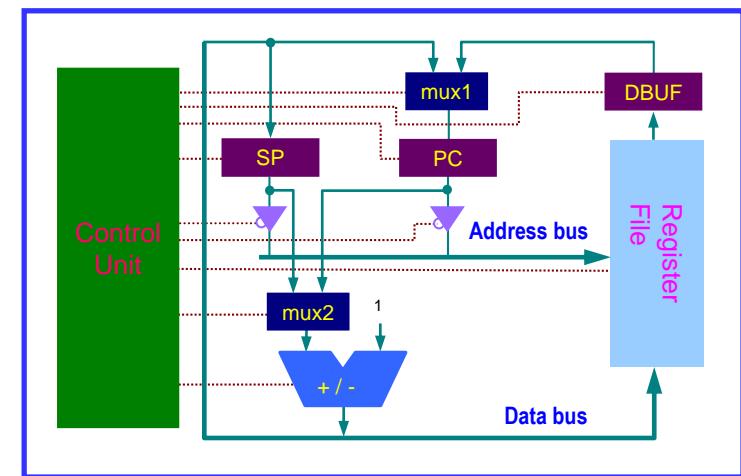


High-level (or Behavioral) Synthesis

High-level (or Behavioral) synthesis

- Synthesis from the **algorithmic behavioral level** to **structural descriptions** consisting of arithmetic hardware elements, memories and wiring

```
if IR(3) = '0' then
    PC      := PC + 1;
else
    DBUF   := MEM(PC);
    MEM(SP) := PC + 1;
    SP     := SP - 1;
    PC      := DBUF;
end if;
```

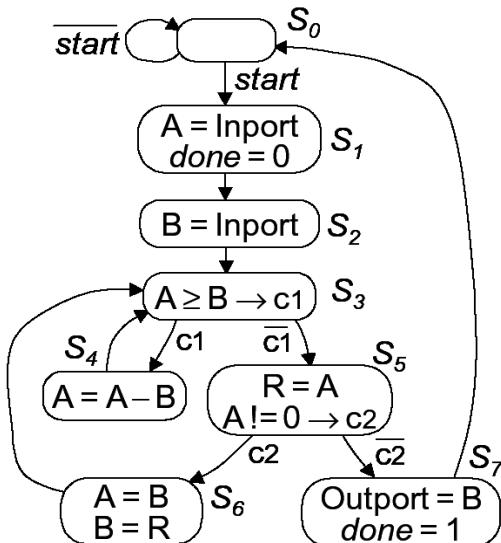
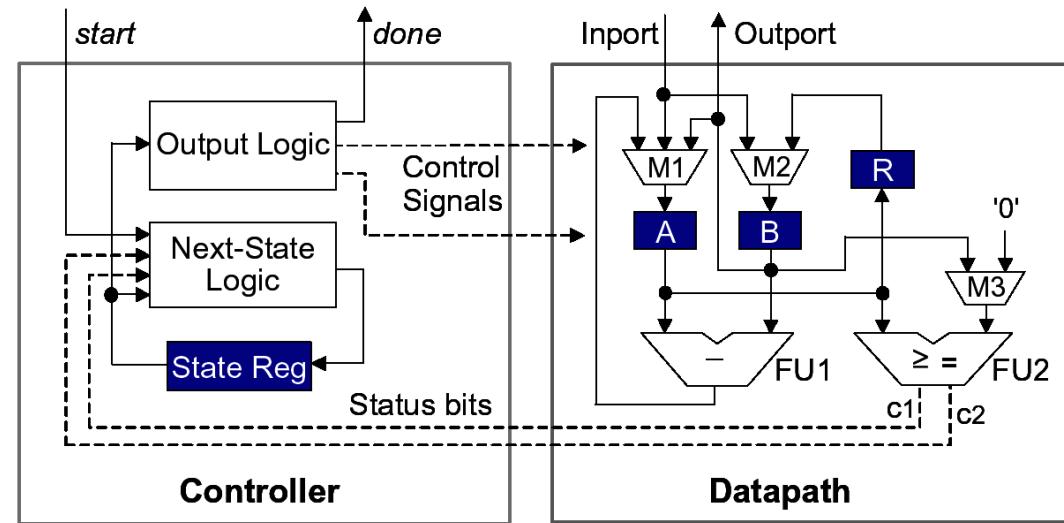


An Example of Behavioral Synthesis

```

A = Import;
B = Import;
done = 0;
Repeat: while (A ≥ B)
         A = A - B;
         R = A;
         if (A != 0) {
             A = B;
             B = R;
             goto Repeat;
         }
         else goto End;
End:   Outport = B;
       done = 1;

```



| PS | Input | NS | Control signals | | | | | | |
|-------|-------------|-------|-----------------|---|---|--------|--------|-----|------|
| | | | A | B | R | M1 | M2 | M3 | done |
| S_0 | $start = 0$ | S_0 | - | - | - | - | - | - | 0 |
| | $start = 1$ | S_1 | | | | | | | |
| S_1 | - | S_2 | 1 | - | - | Import | - | - | 0 |
| S_2 | - | S_3 | 0 | 1 | - | - | Import | - | 0 |
| S_3 | c1 = 1 | S_4 | 0 | 0 | 0 | - | - | B | 0 |
| | c1 = 0 | S_5 | | | | | | | |
| S_4 | - | S_3 | 1 | 0 | 0 | FU1 | - | - | 0 |
| S_5 | c2 = 1 | S_6 | 0 | 0 | 1 | - | - | '0' | 0 |
| | c2 = 0 | S_7 | | | | | | | |
| S_6 | - | S_3 | 1 | 1 | 0 | B | R | - | 0 |
| S_7 | - | S_0 | 0 | 0 | 0 | - | - | - | 1 |

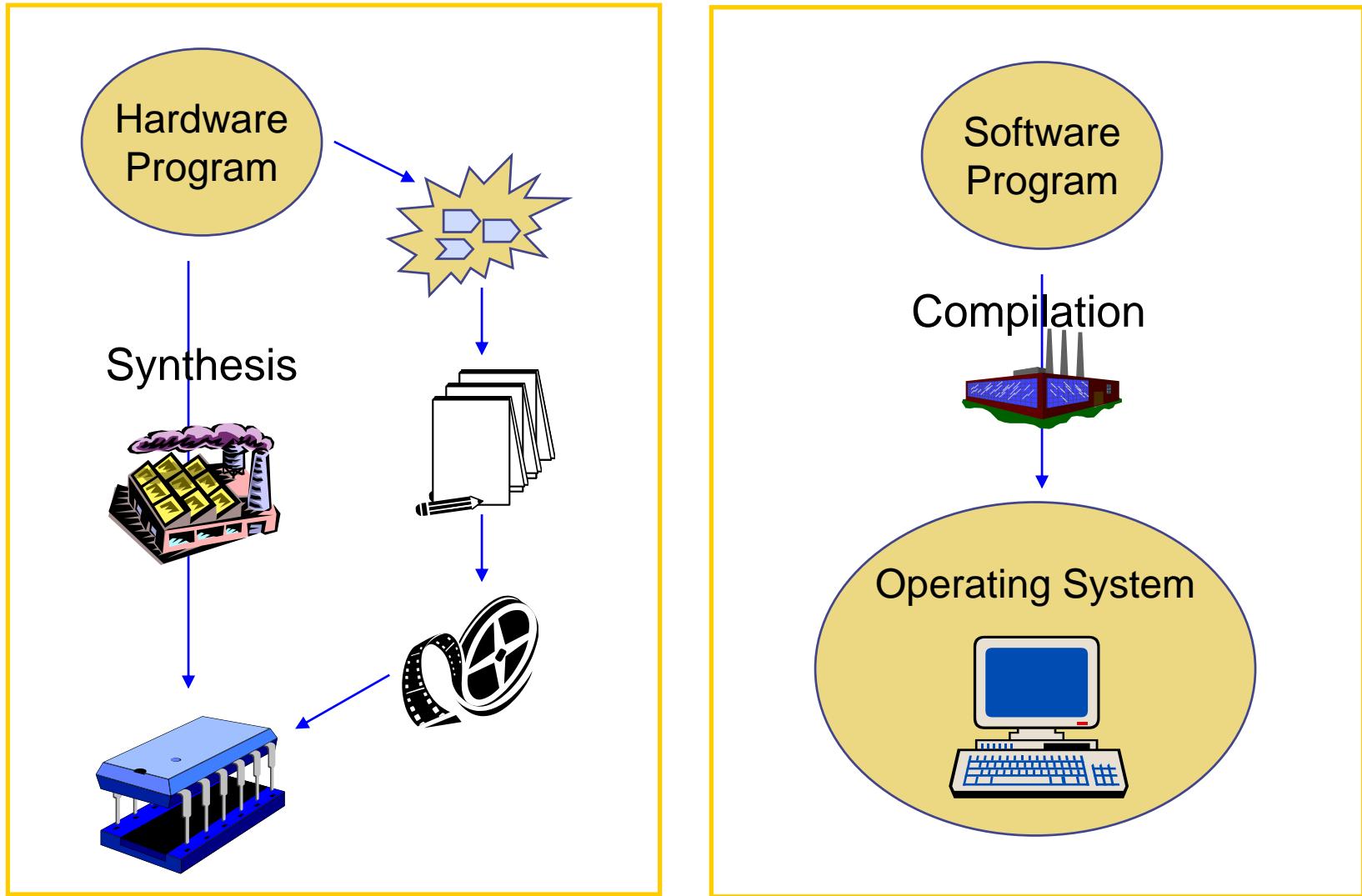
Silicon Compiler



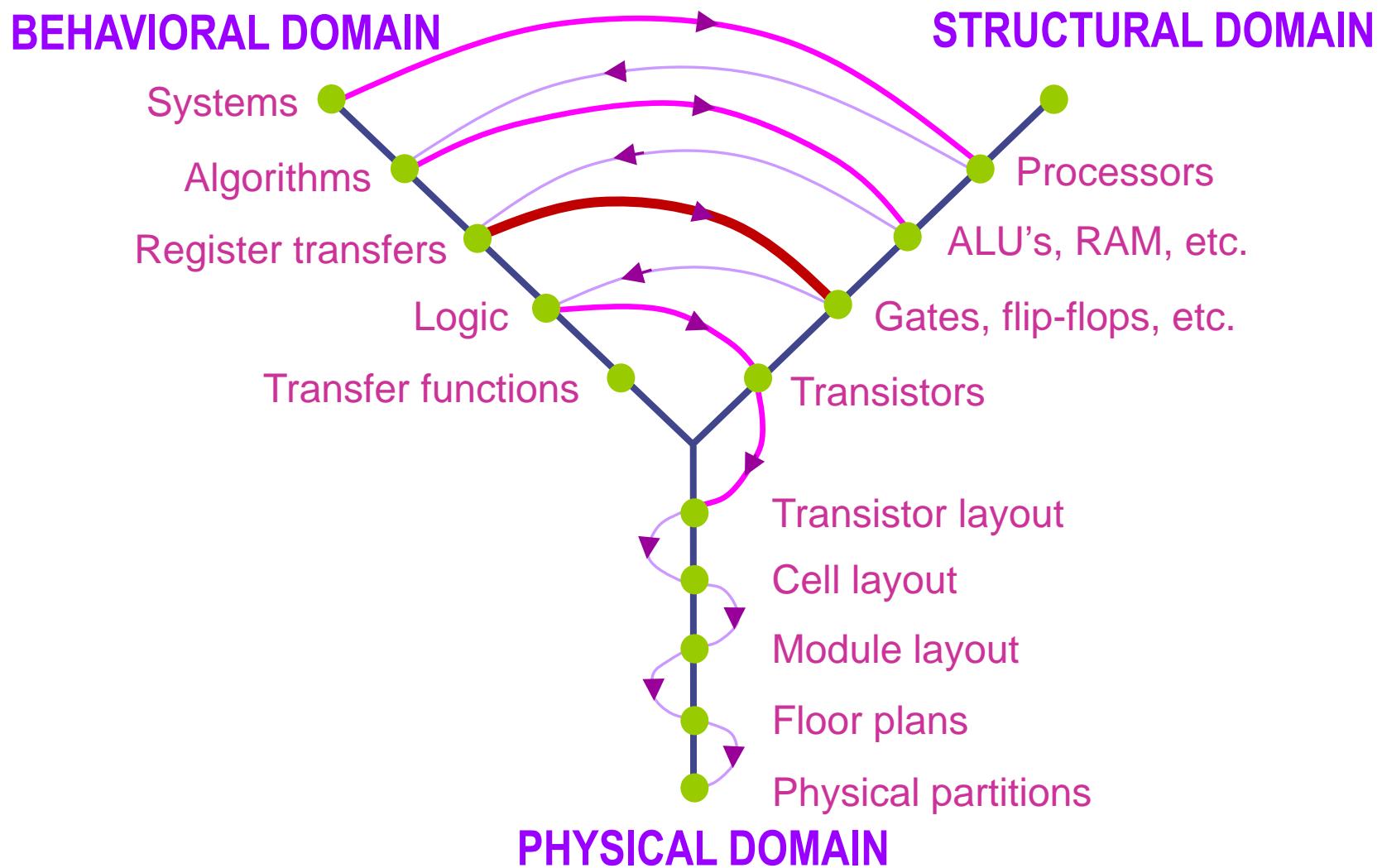
Silicon compiler

- ◆ Analogy with compilers for programming languages: from **algorithm** to **machine instructions**
- ◆ In VLSI: from **algorithm** to **mask patterns**
- ◆ Current practice:
 - ❖ this ideal is approached more and more by the **synthesis tools** (e.g. VHDL synthesis)
 - ❖ still far away from a single push-button operation

HDL's vs. SDL's: Realization



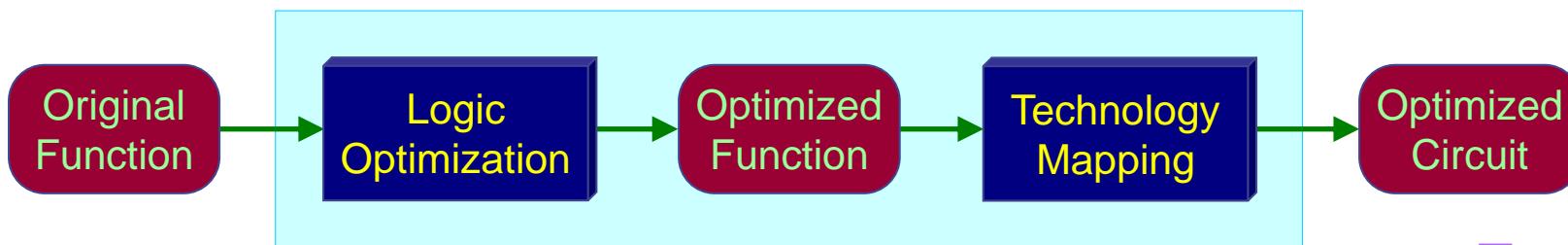
Top-down Structural Design



Logic Design



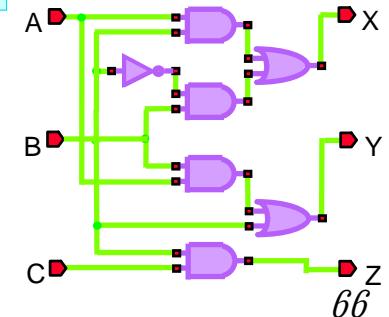
- Schematic entry
- Logic synthesis
 - ◆ Translates **RTL** design to **gate-level** design
- Gate-level simulation (functionality, power, etc.)
- Technology mapping (actually available library cells)
- Timing analysis (the maximum delay paths)
- Formal verification



$$X = AB + AB'C + A'BC$$

$$Y = \dots$$

$$Z = \dots$$



Logic Synthesis

- Transform Boolean expressions into logic gate networks in a particular library
 - ◆ Two-level combinational logic (a sum of products or a product of sums, implemented as PLAs)
 - ◆ Multilevel combinational logic (random logic)
 - ◆ Sequential logic (state encoding, etc.)
- 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



- Two-level: minimize the number of product terms

$$F = \overline{x_1} \overline{x_2} \overline{x_3} + \overline{x_1} \overline{x_2} x_3 + x_1 \overline{x_2} x_3 \Rightarrow F = \overline{x_1} \overline{x_2} + \overline{x_2} x_3$$

- Multi-level: minimize the number of literals, variables

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

$$\begin{aligned}t1 &= a+bc \\t2 &= d+e \\t3 &= ab+d \\t4 &= t1 t2 + fg \\t5 &= t4h + t2 t3 \\F &= t5'\end{aligned}$$



$$\begin{aligned}t1 &= d+e \\t2 &= b+h \\t3 &= at2+c \\t4 &= t1 t3 + fgh \\F &= t4'\end{aligned}$$

- Methods/CAD tools

- ◆ Quine-McCluskey method (exponential-time exact algorithm), Espresso (heuristics for two-level logic), MIS (heuristics for multi-level logic), Synopsys, etc.

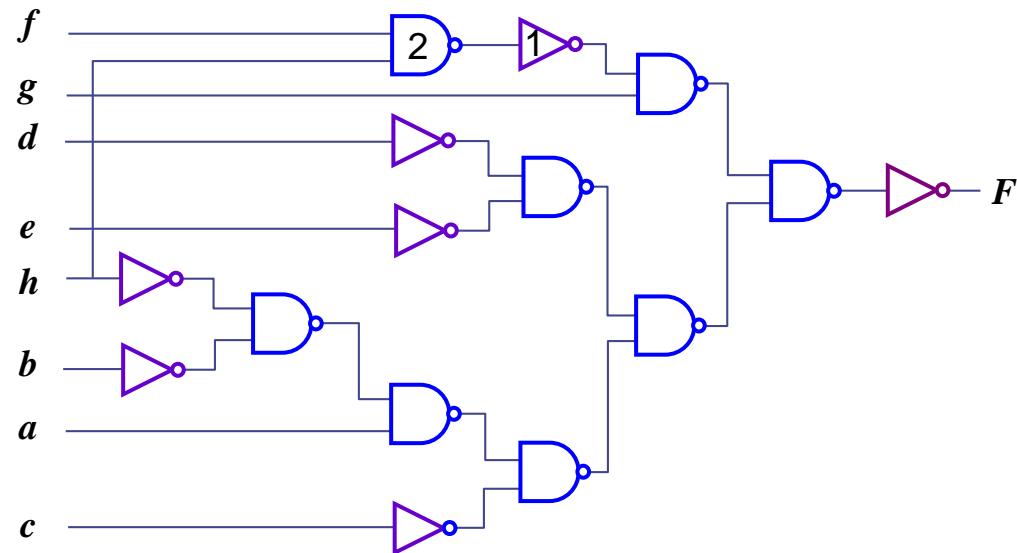
Technology Mapping

- Technology Mapping: The optimization problem of finding a **minimum cost covering** of the subject graph by choosing from the collection of pattern graphs for all
- A cover is a collection of pattern graphs such that every node of the subject graph is contained in one (or more) of the pattern graphs
- The cover is further constrained so that each input required by a pattern graph is actually an output of some other pattern graph

Trivial Covering

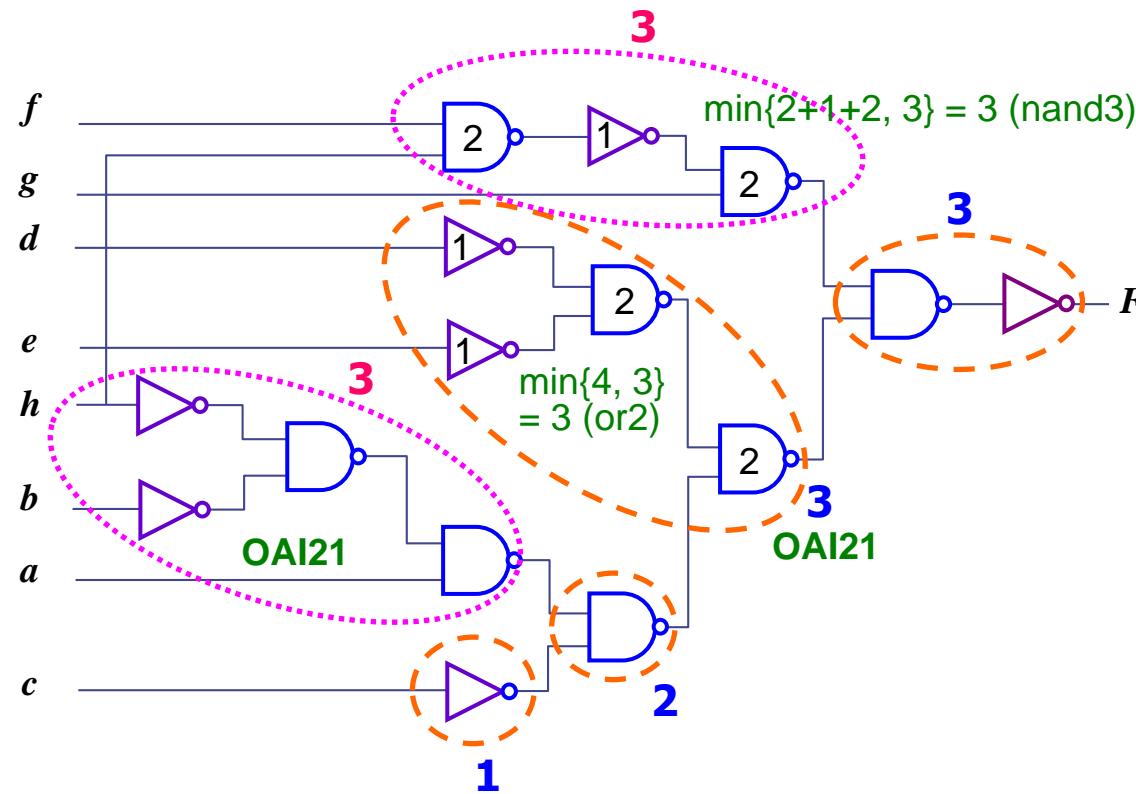
- Mapped into 2-input NANDs and 1-input inverters
- 8 2-input NAND-gates and 7 inverters for an area cost of 23

$$\begin{aligned}t1 &= d+e \\t2 &= b+h \\t3 &= a \cdot t2 + c \\t4 &= t1 \cdot t3 + f \cdot g \cdot h \\F &= t4'\end{aligned}$$

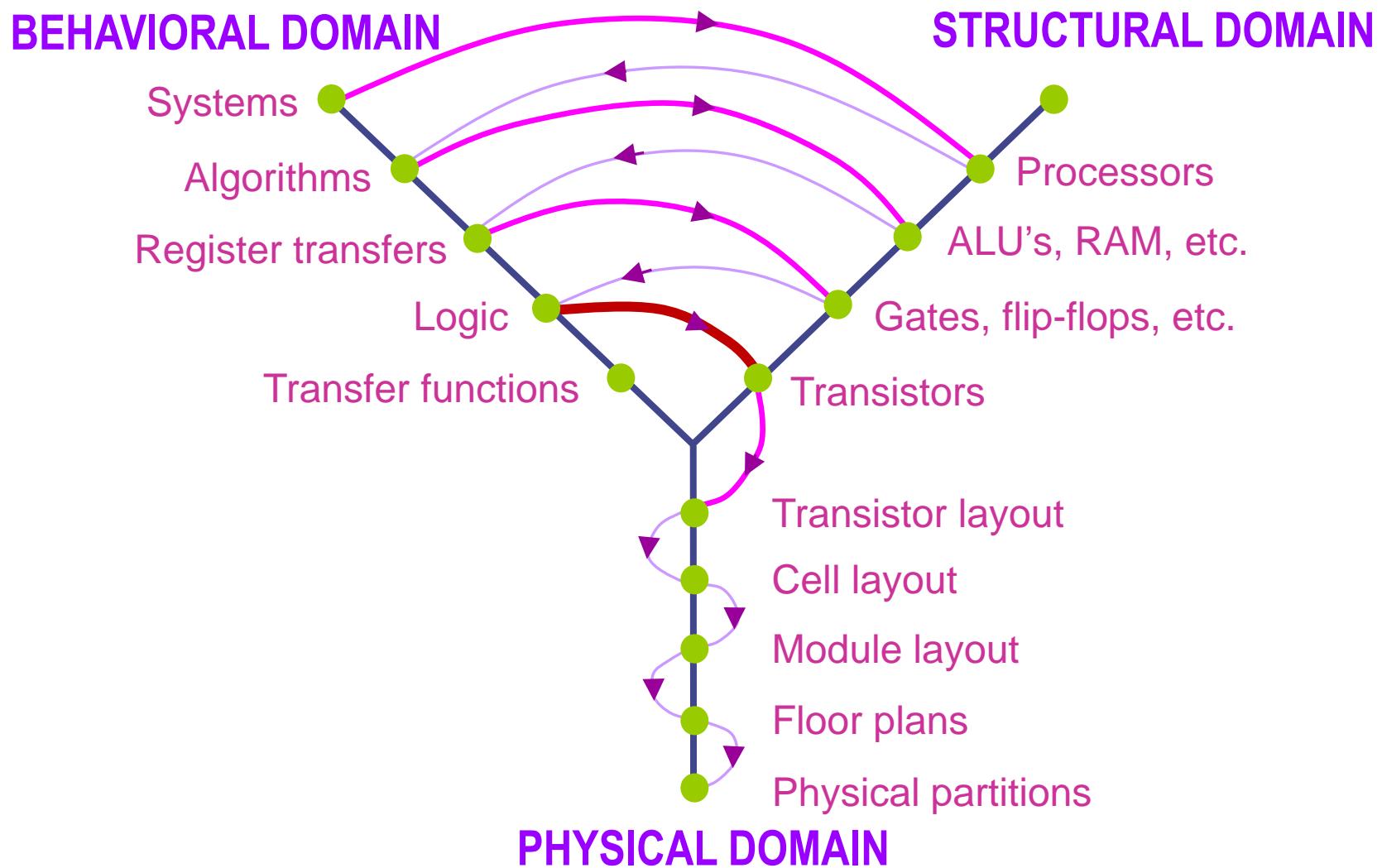


Best Covering

- A best covering with an area of 15
- Obtained by the dynamic programming approach



Top-down Structural Design



Transistor-level design



■ Switch-level simulation

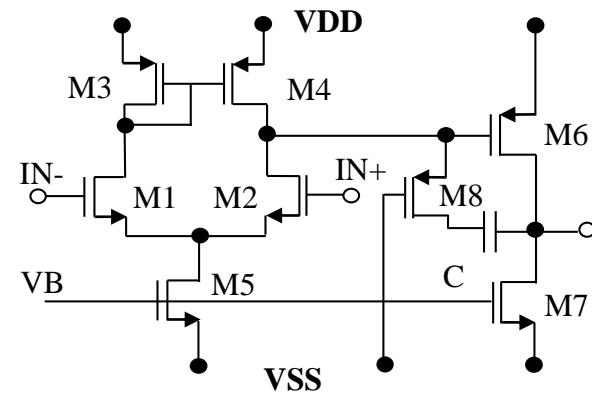
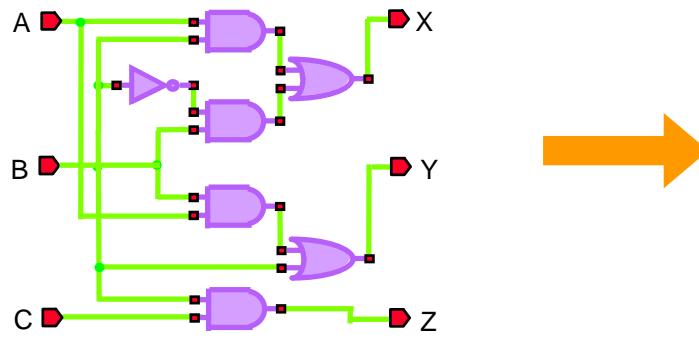
- Transistors are modeled as ideal bidirectional switches

■ Timing-level simulation

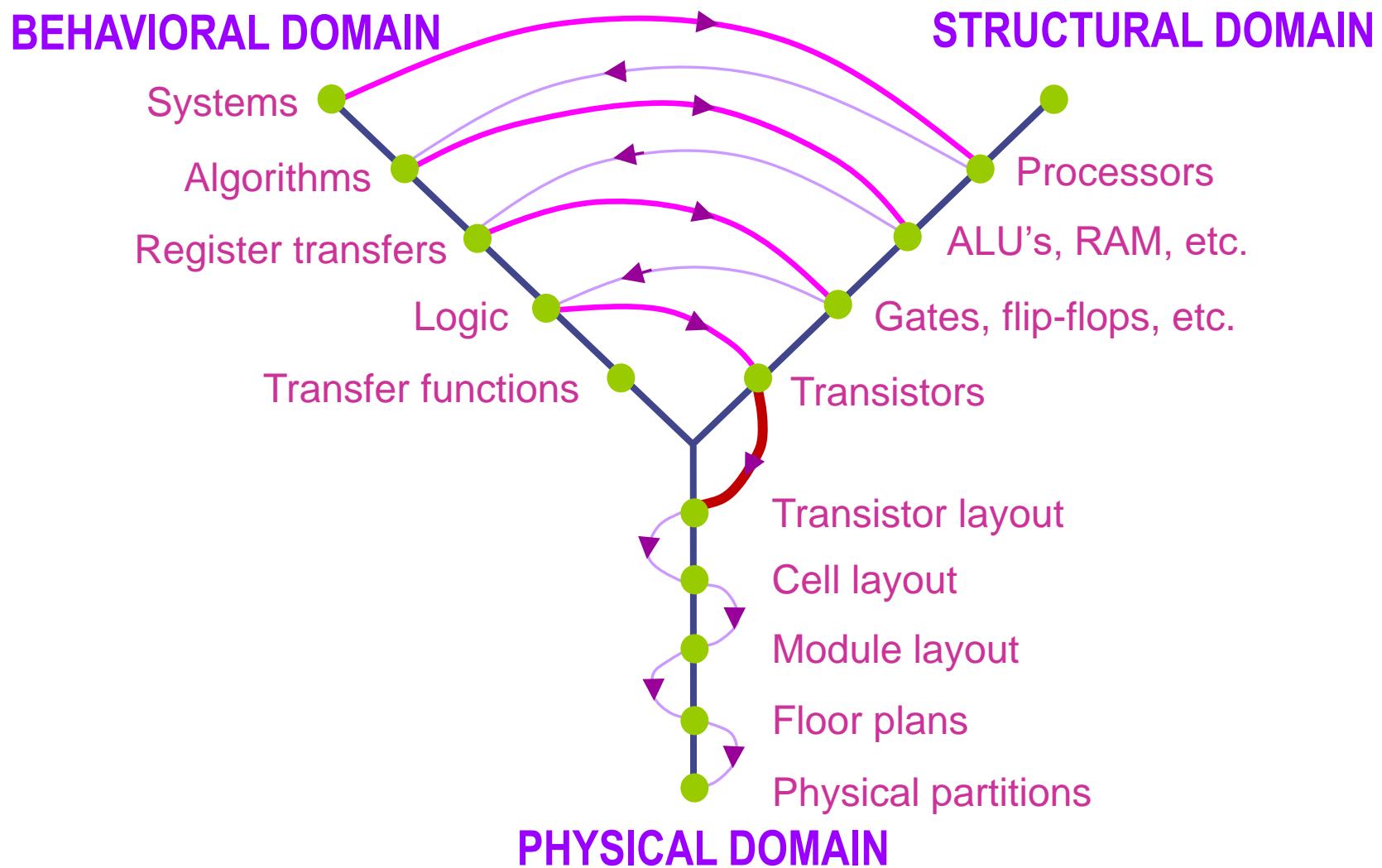
- ◆ Analog signals are considered, but transistors have simple models

■ Circuit-level simulation

- More accurate models of the transistors are used
 - Involve nonlinear differential equations for currents and voltages



Top-down Structural Design



Physical Design

- Is concerned with the creation of **mask layouts**
- Create a **layout** of geometrical entities indicating the **transistor dimensions, locations, and their connections**

