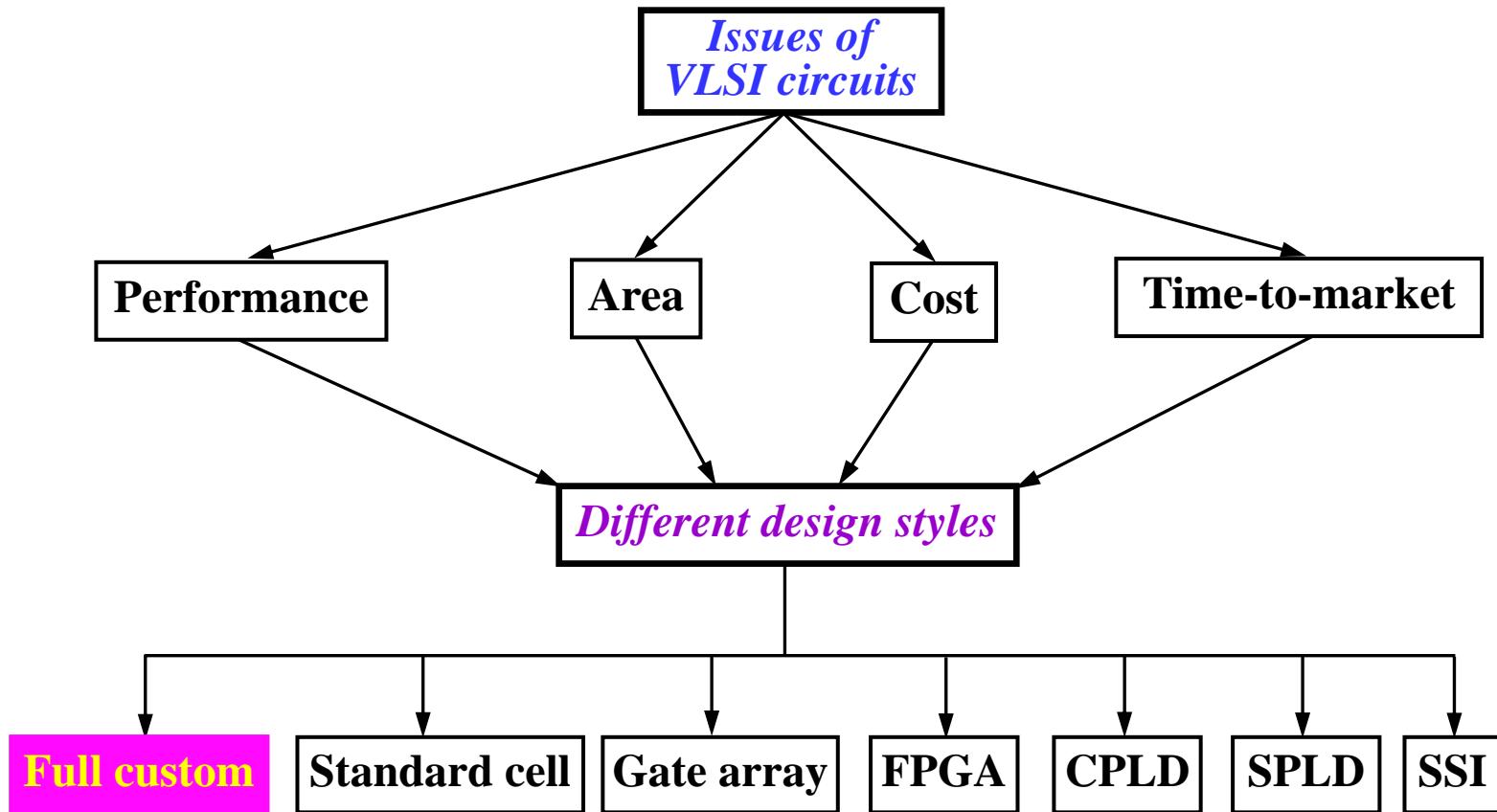




Introduction to VLSI Design

VLSI Design Styles

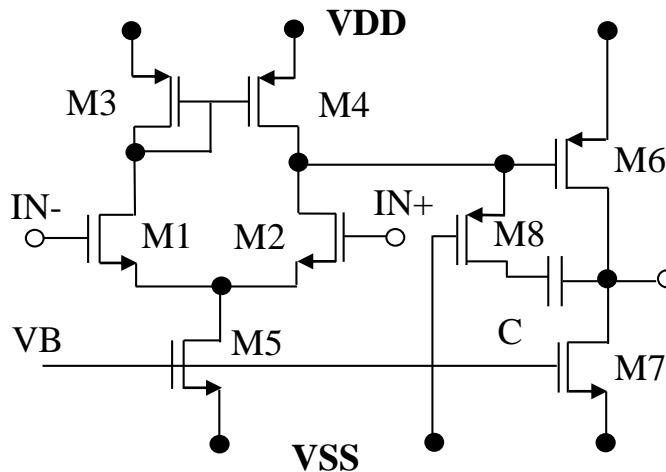


Performance, Area efficiency, Cost, Flexibility

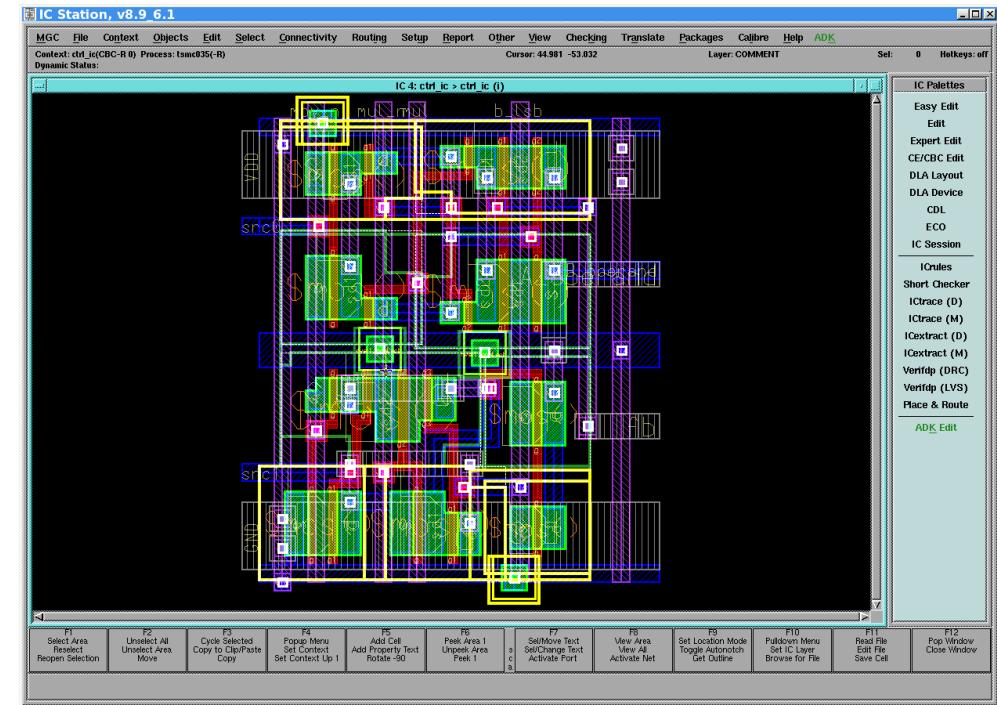
Full-custom Design (1/3)

Full-custom Design (Layout editor)

- ◆ small circuits such as the library cells
- ◆ performance-critical part of a large circuit
- ◆ a circuit such as a microprocessor is to be mass-produced

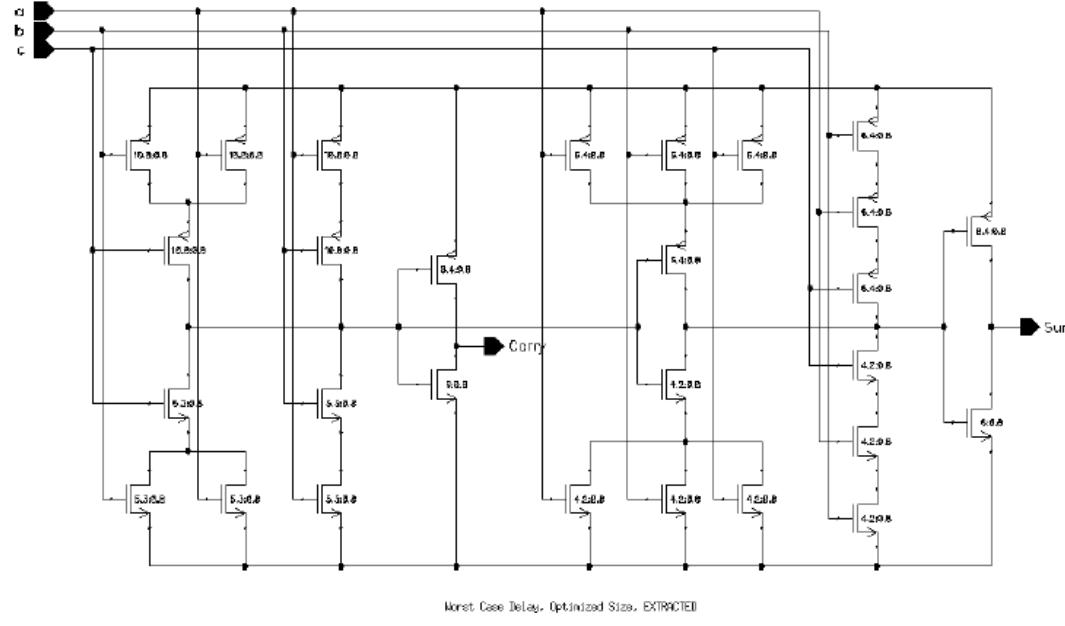


Schematic

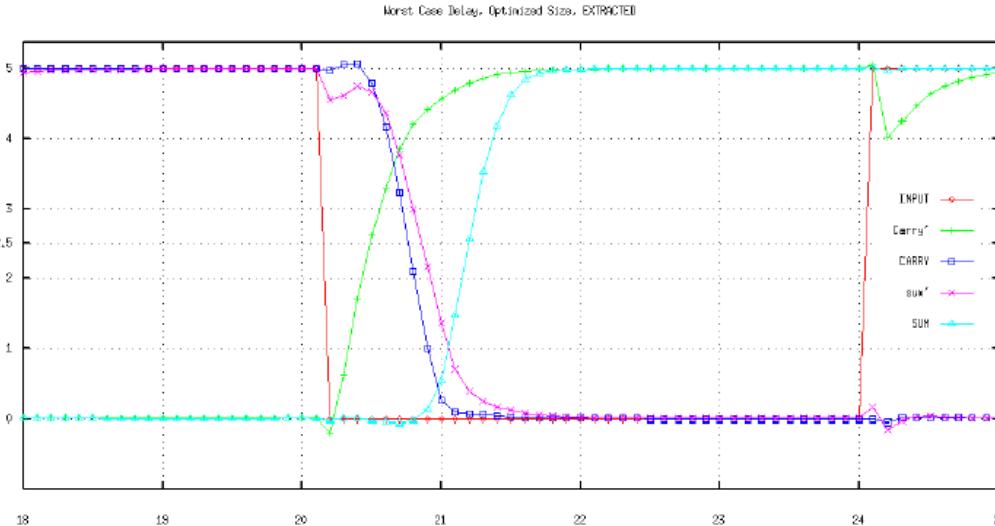


Layout

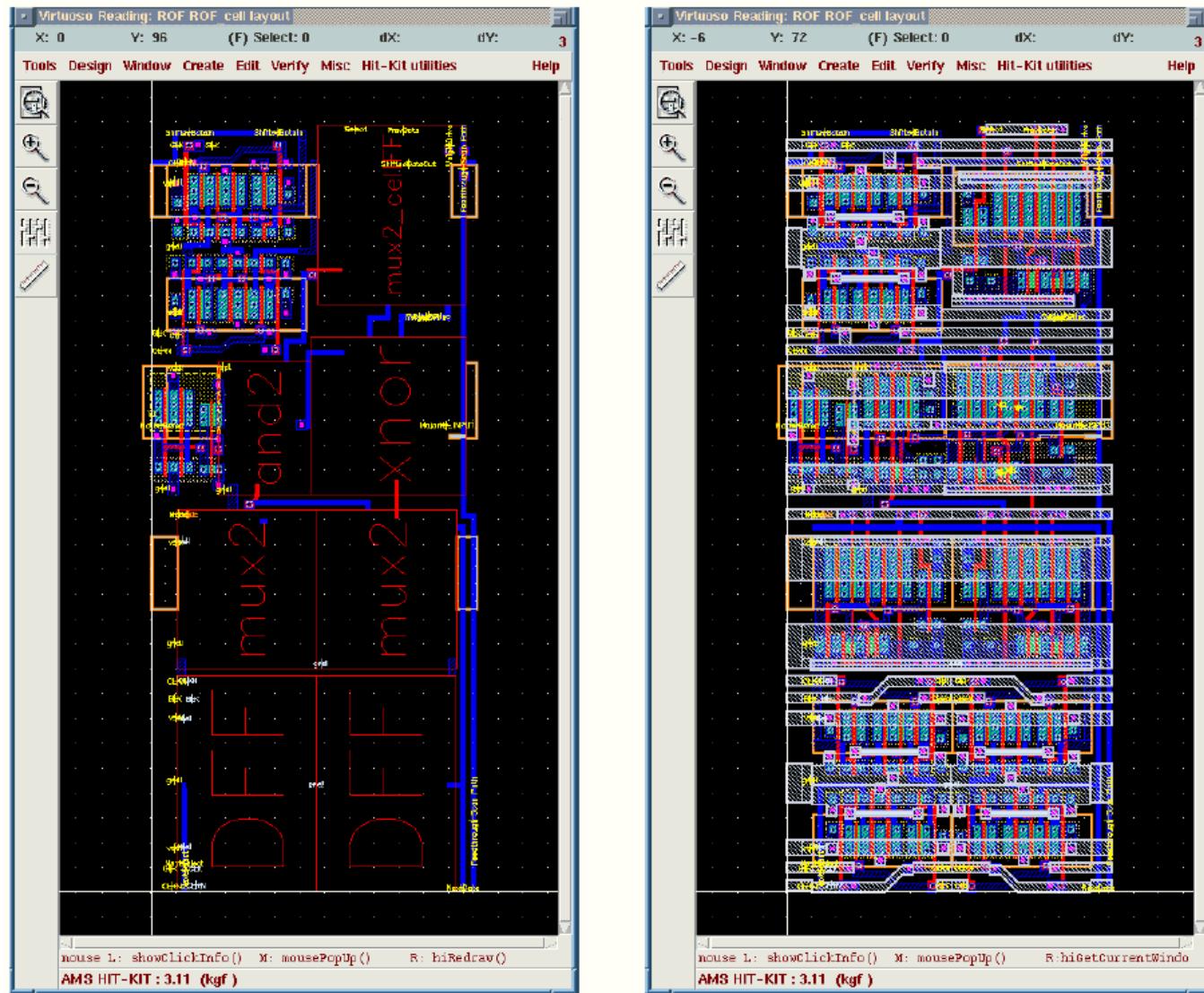
Full-custom Design (2/3)



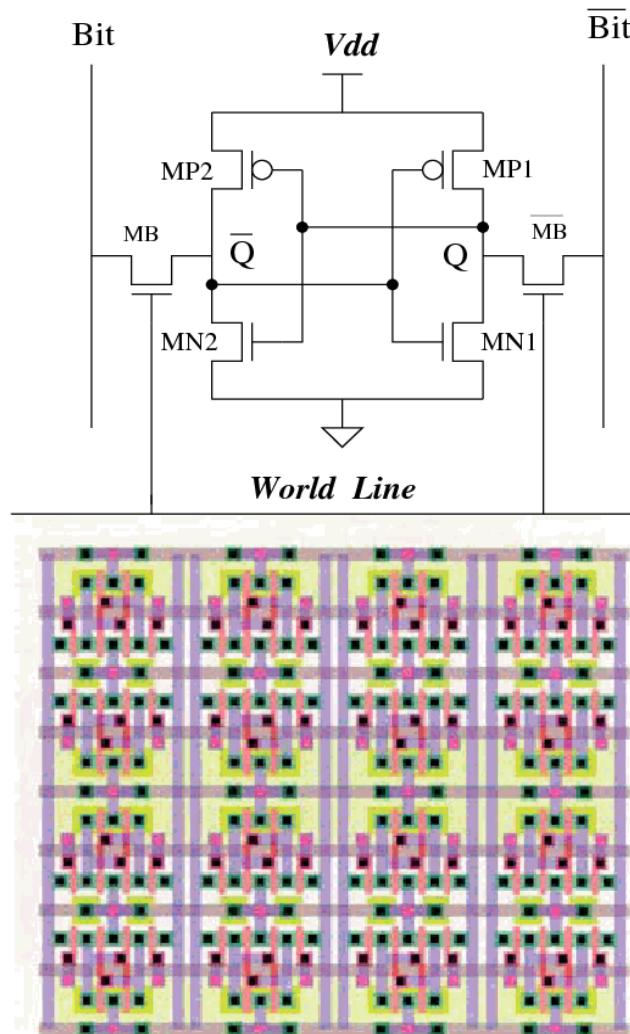
1-bit full-adder schematic



SPICE simulation results



Regularity and Modularity



Full Custom SRAM Cell Design

Full-custom Design (3/3)



■ Design Rule Check (DRC)

- ◆ detect violations of design rules

■ Layout Versus Schematic (LVS)

- ◆ determine whether a particular layout corresponds to the original schematic or circuit diagram of the design

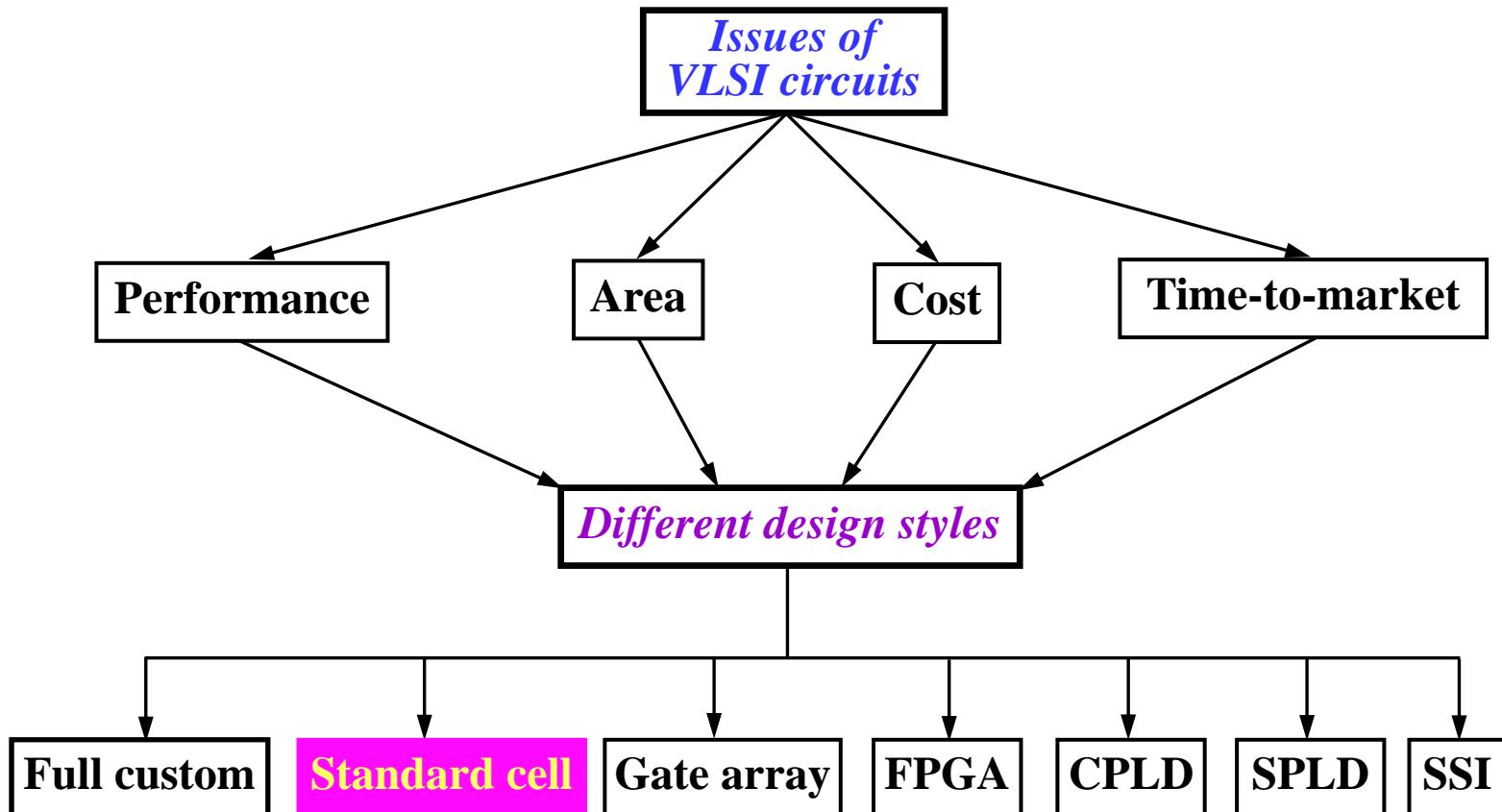
■ Circuit Extraction

- ◆ input the **mask patterns**, construct a **circuit of transistors, resistors and capacitances** that can be simulated at the circuit or switch level
- ◆ find out how **parasitic capacitances and resistors** affect the circuit behavior (timing behavior)

■ Layout compaction

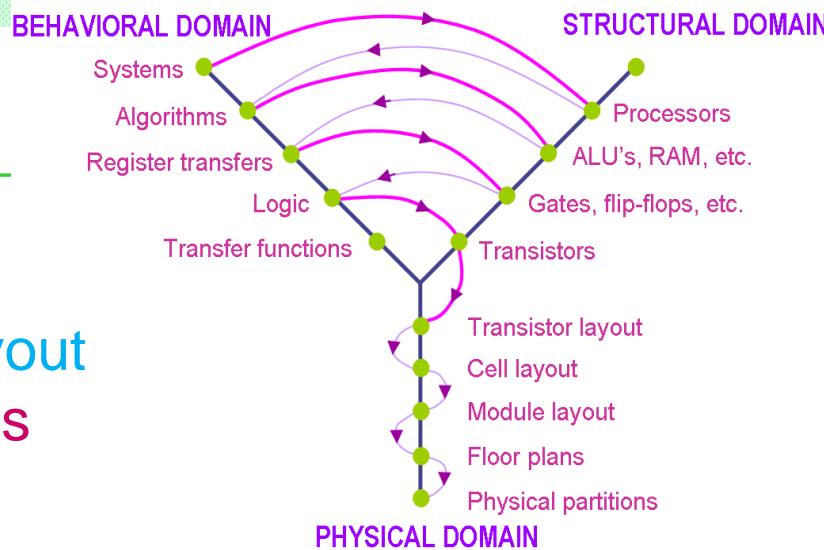
- ◆ symbolic layout
 - ◆ widths and distances of mask patterns **are irrelevant**
- ◆ take the symbolic description, assign widths to all patterns and space the patterns such that all **design rules** are satisfied

VLSI Design Styles



Performance, Area efficiency, Cost, Flexibility

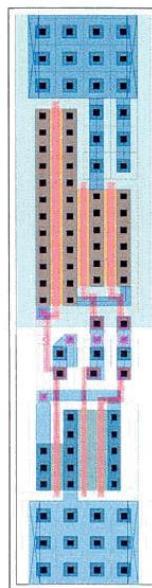
Standard-cell Approach (1/2)



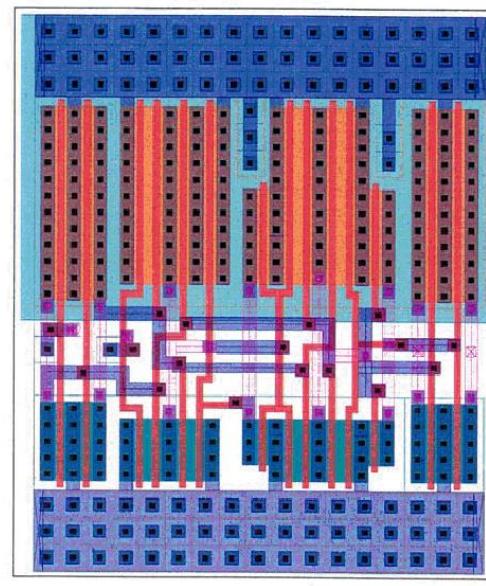
- Convert a circuit into a **physical layout** using **a library of basic logic circuits (standard cells)** in the process
- Standard-cell libraries are commercially available for specific fabrication process
- A library usually contains basic logic gates (inverter, NAND, NOR, etc.)
- The **behavioral representation** of a design is **synthesized** into a **structural representation** that consists of only **standard-cells** available in the library
- A **layout** is then created by **arranging** and **interconnecting** the standard-cells required in the design

Standard-cell Approach (2/2)

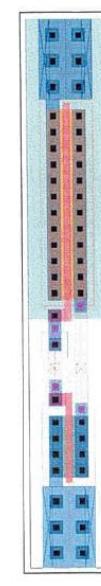
- Compared to the full-custom approach, the complexity of designing a circuit is greatly reduced
- Standard-cells in a library are typically designed to
 - have an identical height (standard height)
 - the widths are determined by their functions
- Example



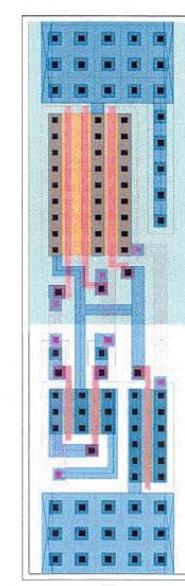
AND



DFF

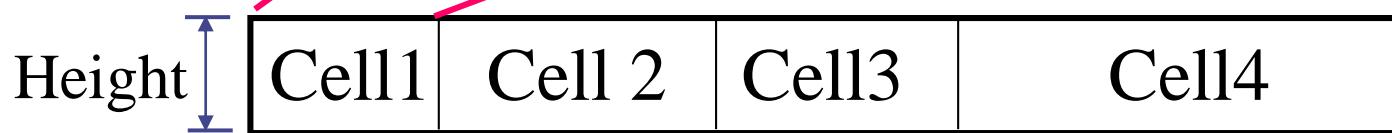
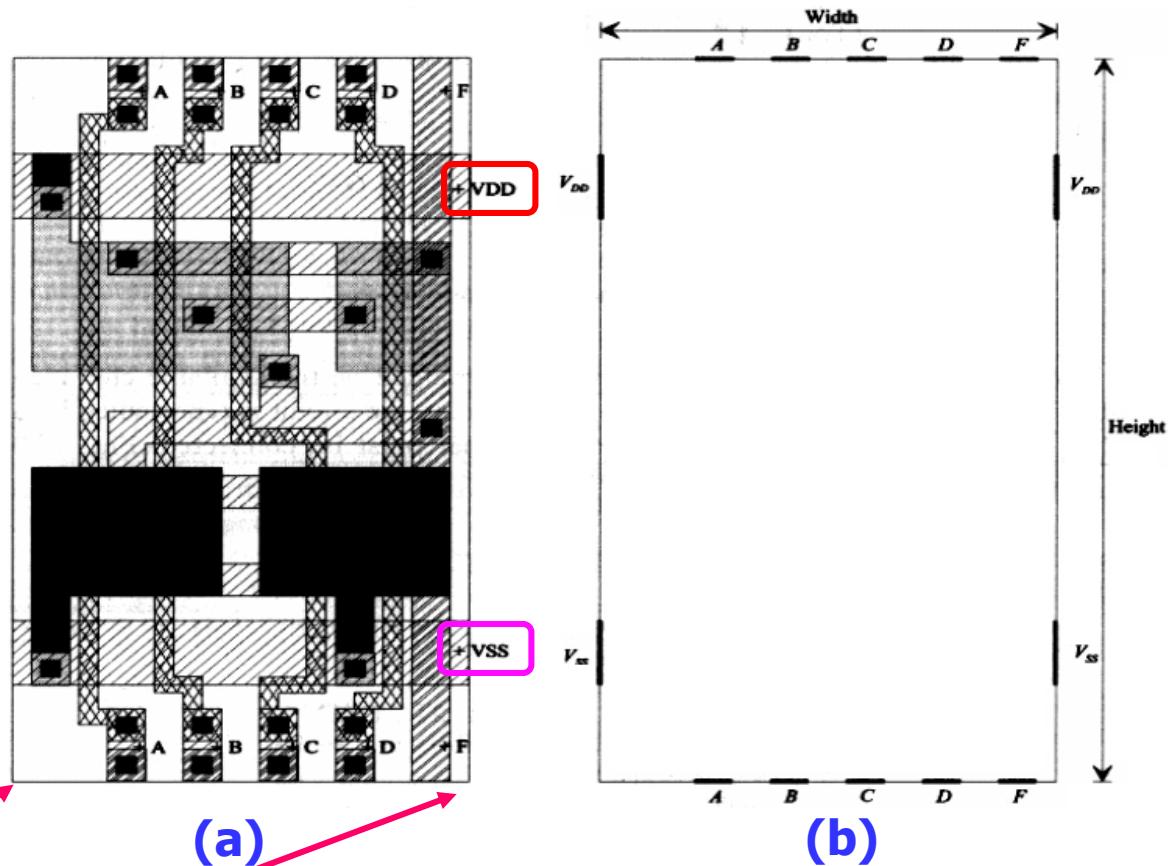


INV



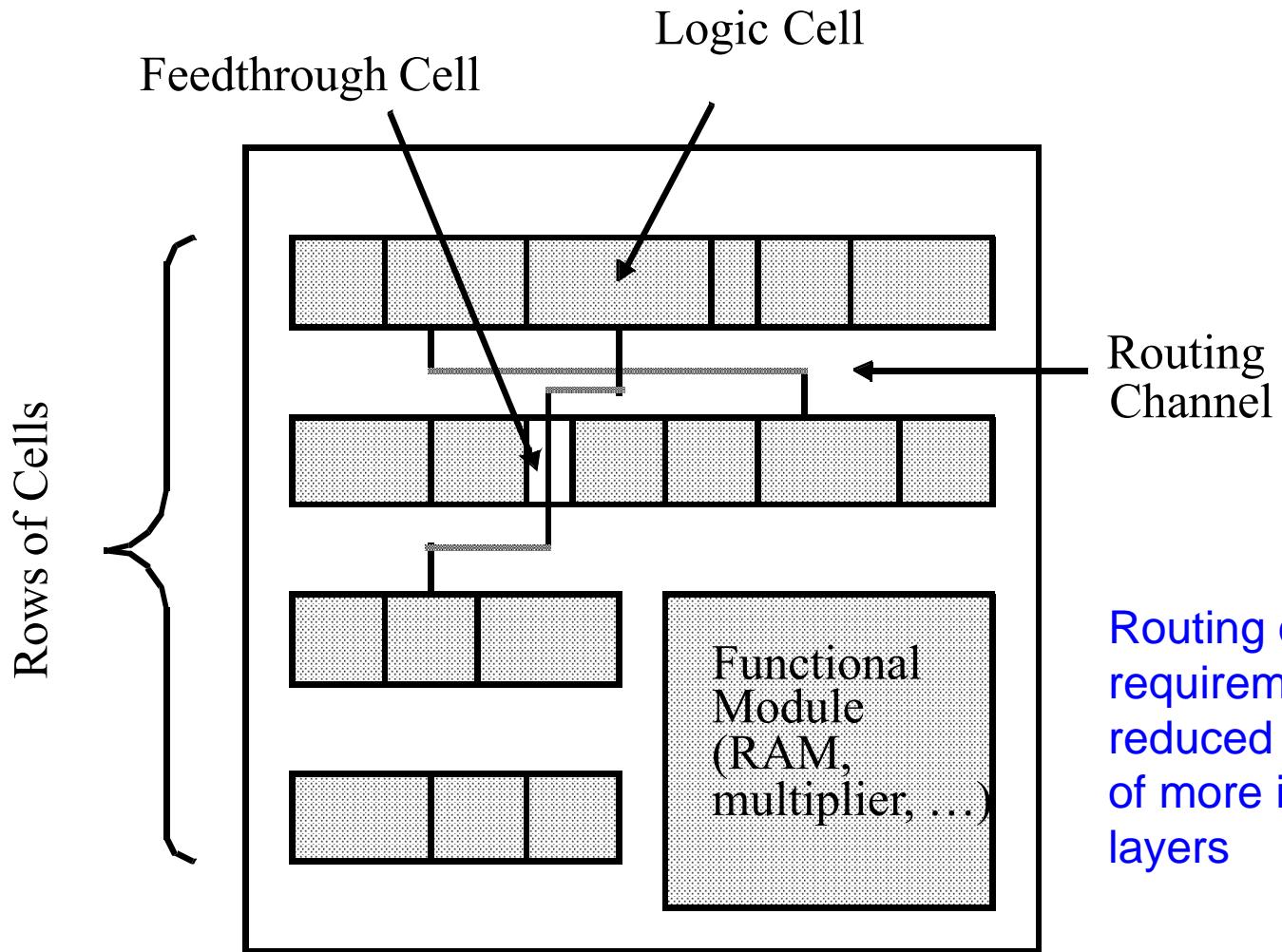
XOR

- (a) Physical design
- (b) Designer's view of a standard cell

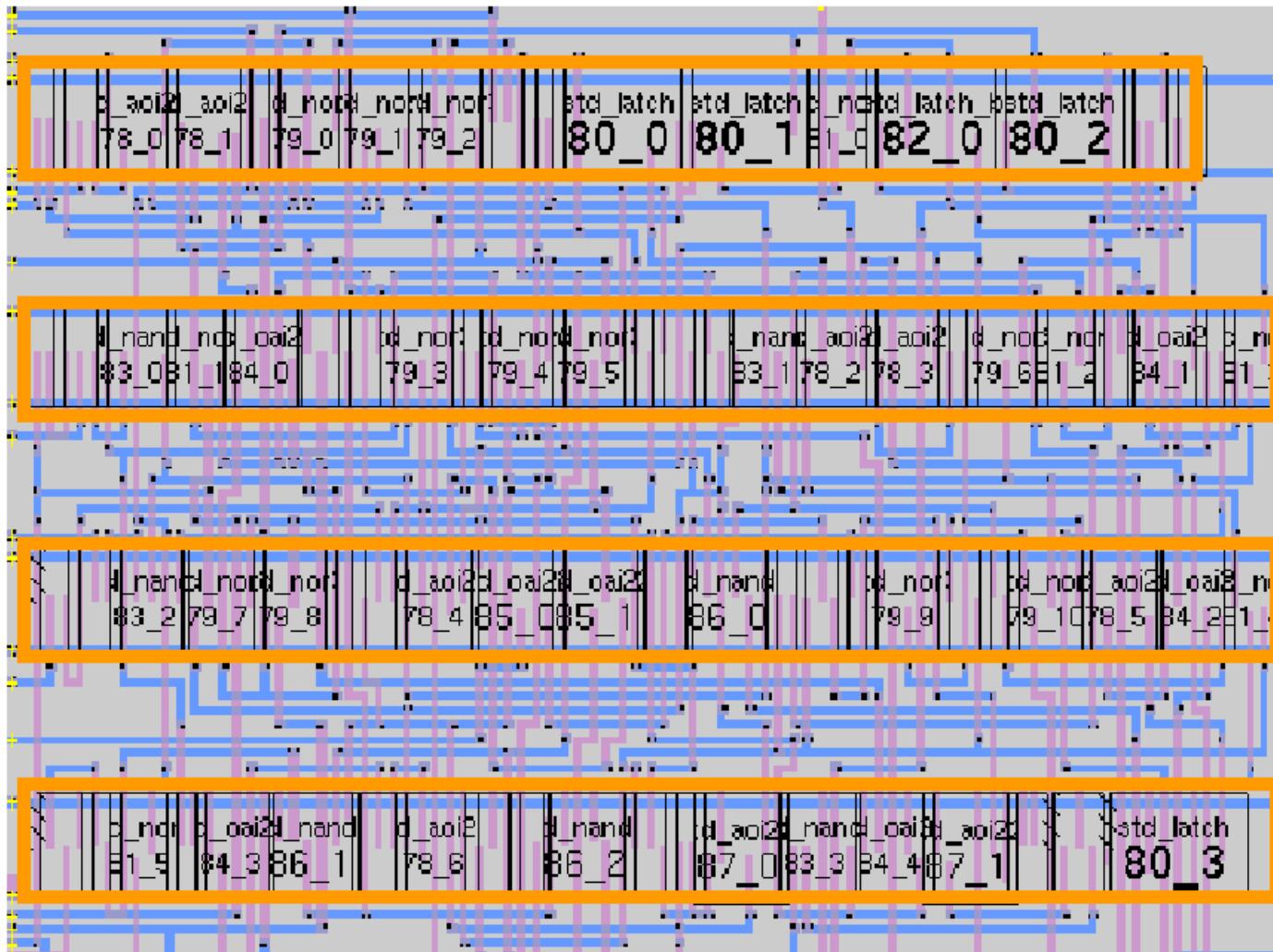


Arrangement of standard-cells in a row

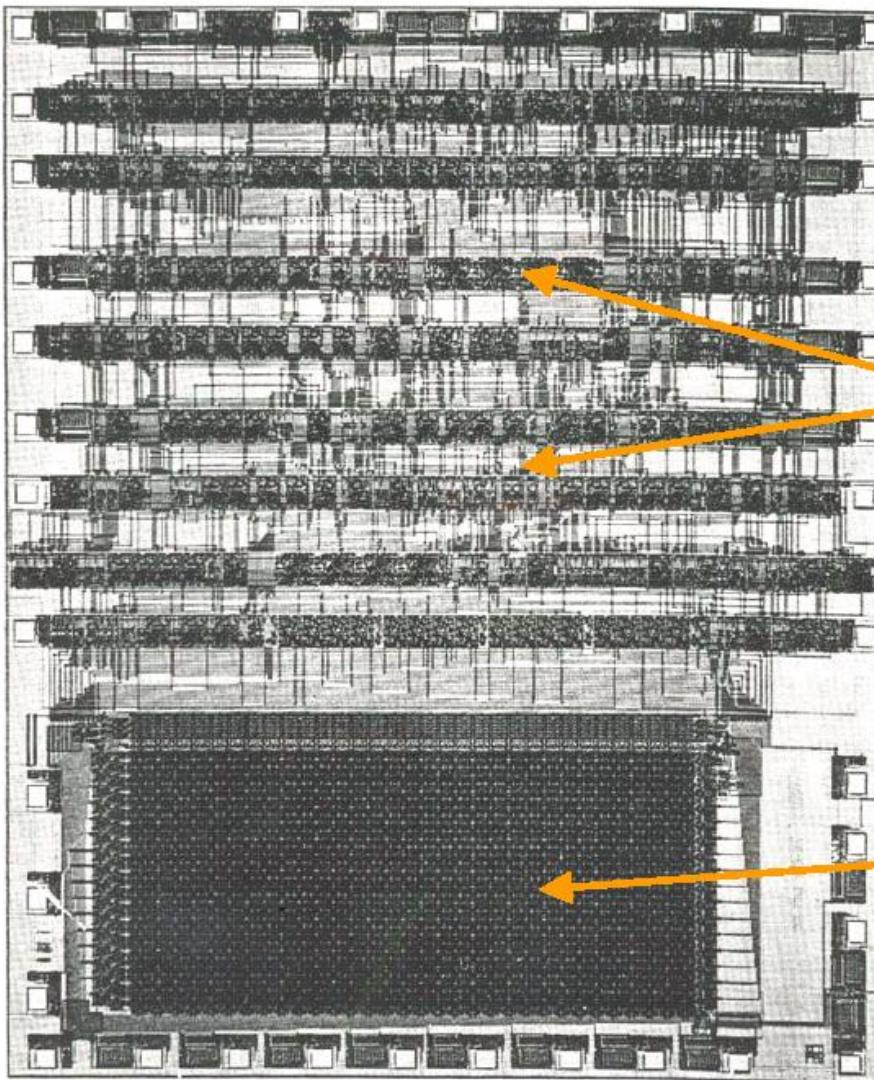
Cell-Based Design



Example of Standard Cells (1/3)



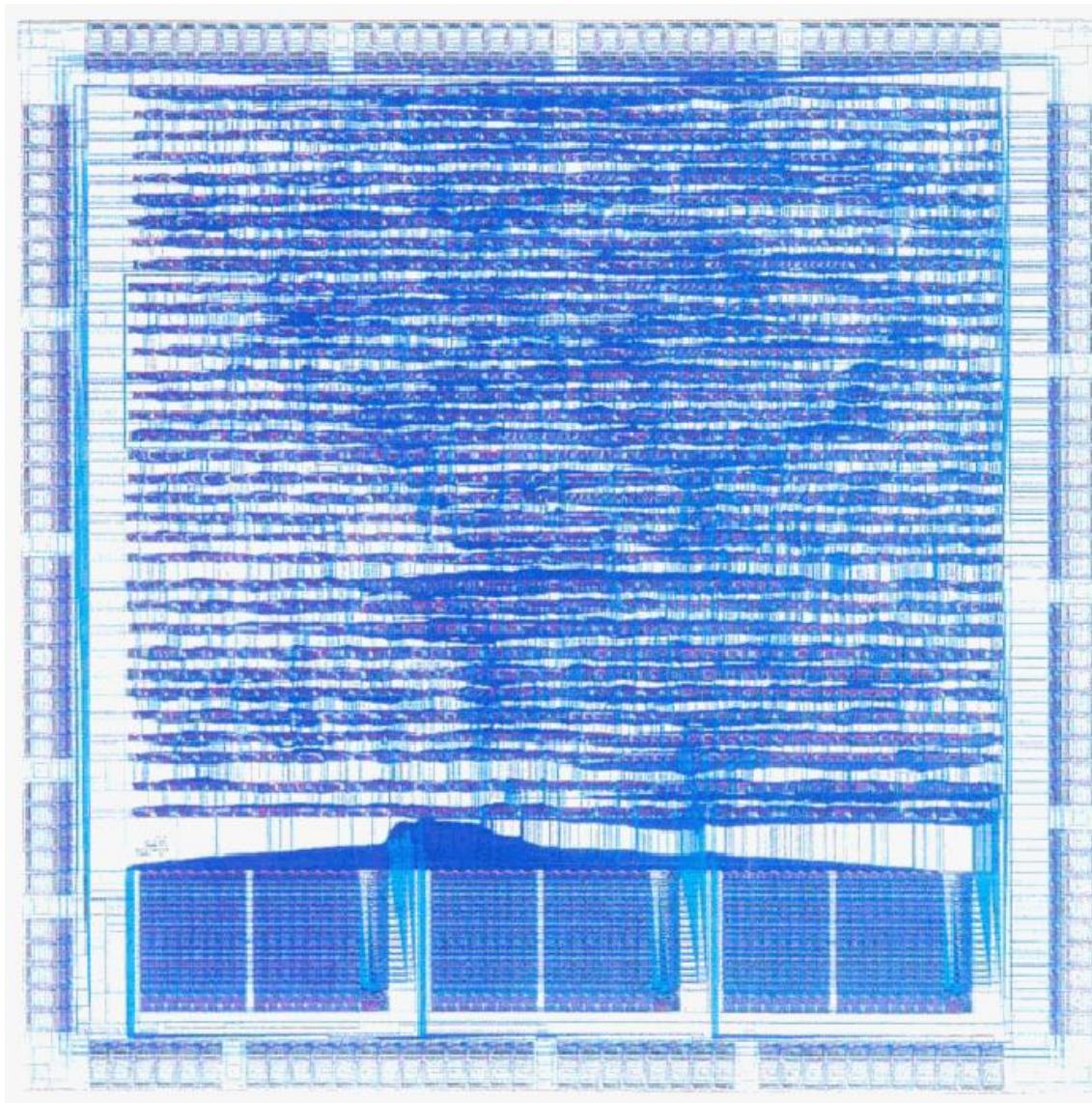
Example of Standard Cells (2/3)



Rows of standard cells with routing channels between them

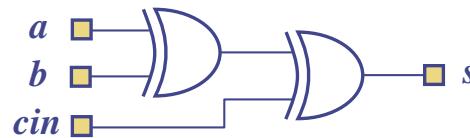
Memory array

Example of Standard Cells (3/3)



Floorplanning, Placement, and Routing (1/2)

- The building blocks of a circuit have to be arranged on the available chip area
- At this point, the circuit is represented by a netlist
 - a list of interconnections (i.e., nets) that must be connected
 - logic diagram of a circuit and its netlist



```
(a, XOR[1].IN1)
(b, XOR[1].IN2)
(XOR[1].OUT, XOR[2].IN1)
(cin, XOR[2].IN2)
(s, XOR[2].OUT)
(XOR[1].Vdd, XOR[2].Vdd)
(XOR[1].GND, XOR[2].GND)
```

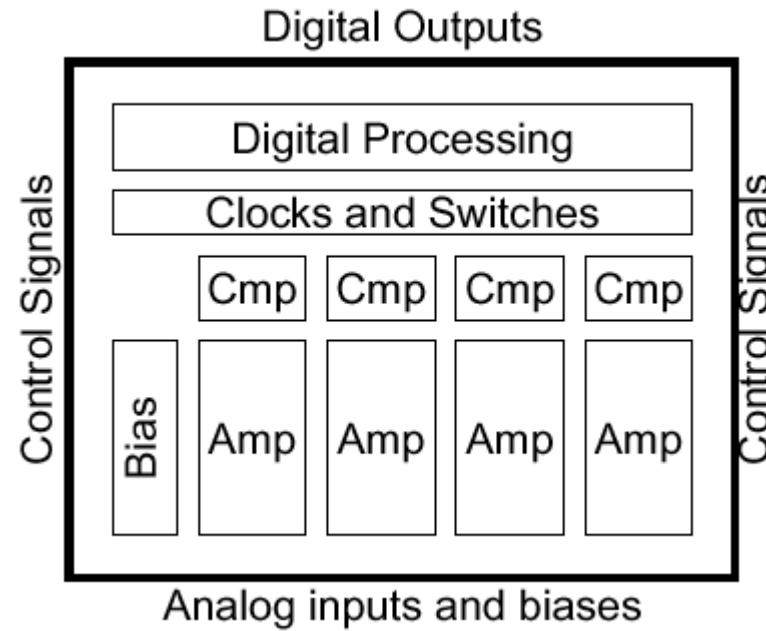
Floorplanning, Placement, and Routing (2/2)



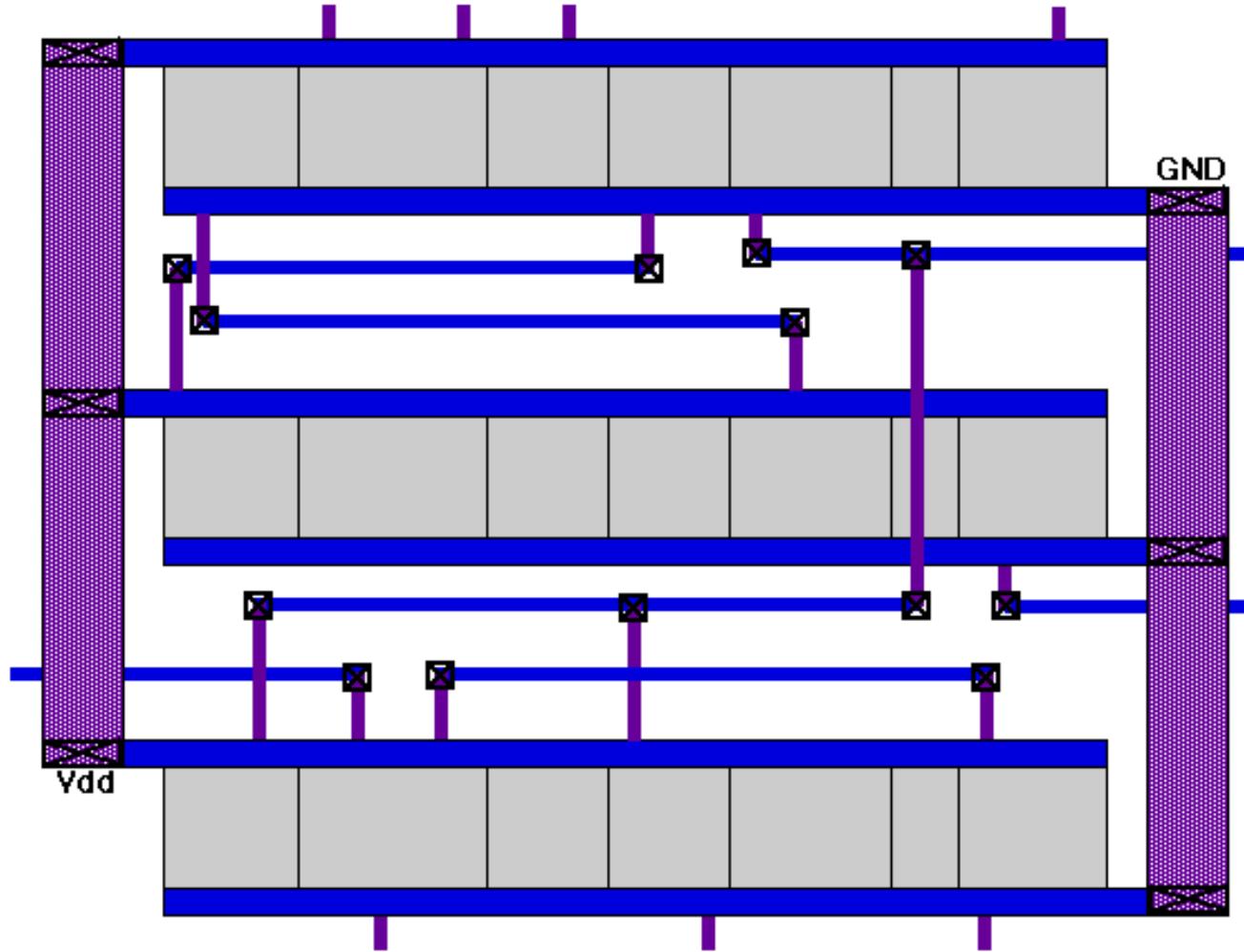
- Floorplanning or **placement** process
 - ◆ the task of **arranging the building blocks** on the layout
 - ◆ attempt to determine the **best location** for each building block
- **Floorplanning**
 - ◆ is reserved to describe the task of a **rough placement**
 - ◆ the **exact shapes** of the building blocks are still being determined
- The criteria for judging a placement result
 - ◆ the overall **area** of the circuit
 - ◆ the **estimated interconnection lengths** (which determine **propagation delays**)
- **Routing** process
 - ◆ take the result of a placement and automatically complete the **interconnections**

Floorplanning: Placement Blocks

- Example:



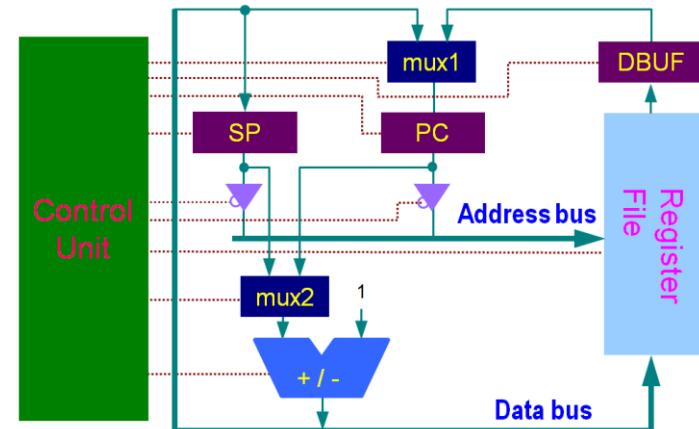
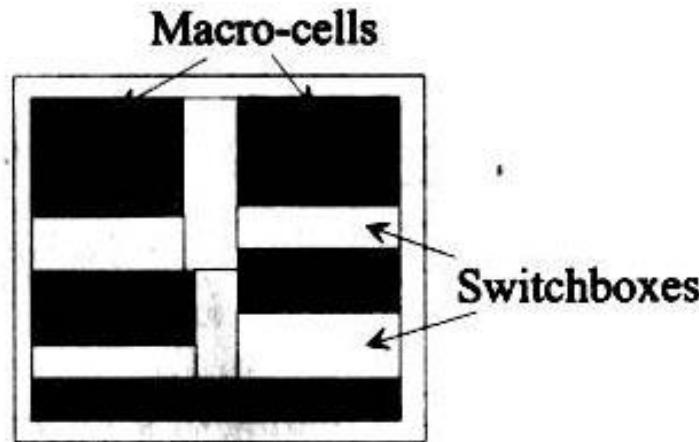
- **Separates analog and digital as much as possible**, divides board into analog and digital sides, uses normally inactive digital control signals and supplies between analog and digital pins.



Placement and routing of standard-cells

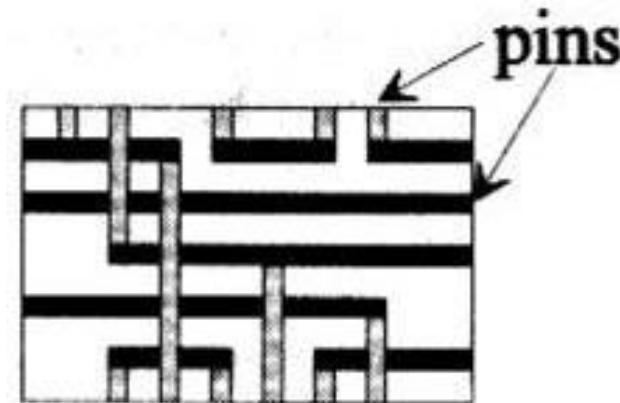
Macro-cell (Custom Cell) (1/2)

- A building block with a more complex functionality
- No dimensional restriction posed on its height and width
- The design of a macro-cell is more flexible
- Candidates for **macro-cells**: registers, register files, arithmetic-logic units, multipliers, random-access memories, read-only memories, ...
 - ◆ design macro-cell with the full-custom approach for optimal results
 - ◆ standard-cells can also be assembled into macro-cells
- Building blocks can no longer be efficiently placed in rows



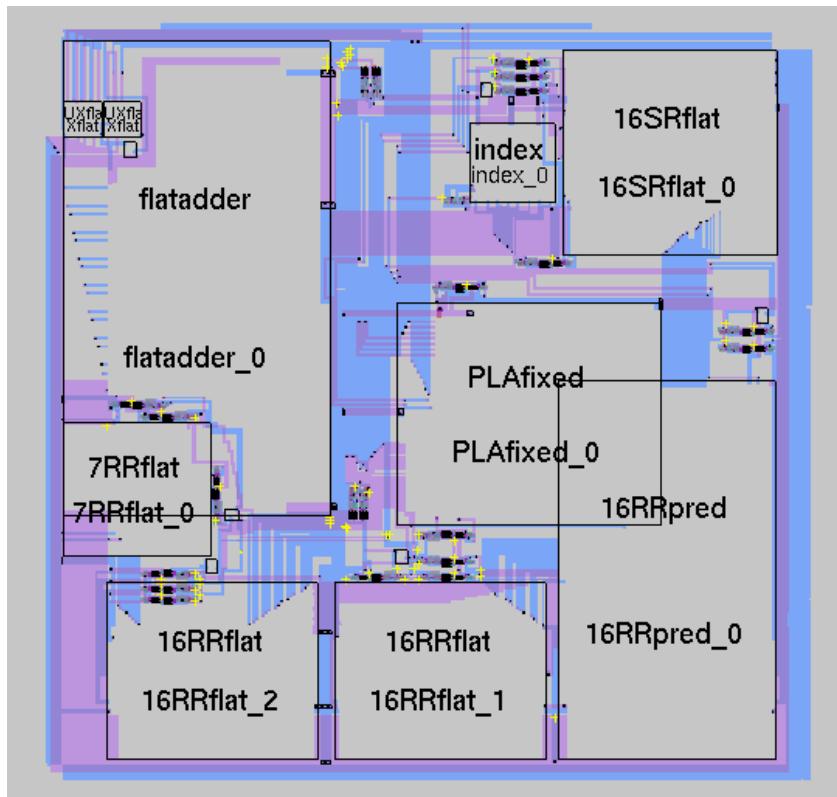
Macro-cell (Custom Cell) (2/2)

- **Switchboxes**: the space between the building blocks is partitioned into rectangular regions
- Routing of macro-cells is carried out in two steps
 - ◆ **Global routing**: determine approximately how the nets should be connected **through** the switchboxes
 - ◆ **Detailed routing** (**switchbox routing**): operate similarly to channel routing except pins are located on all four sides of a switchbox
 - ◆ Switchbox routing example

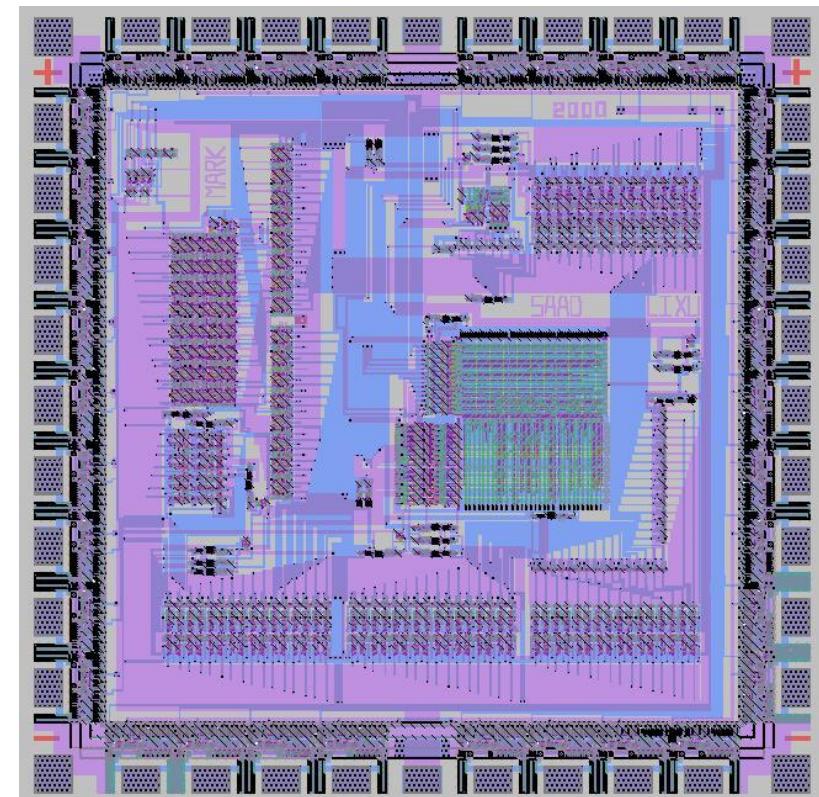


Switchbox routing example

Global Routing & Detail Routing



Global routing



Final Layout

Clock Distribution

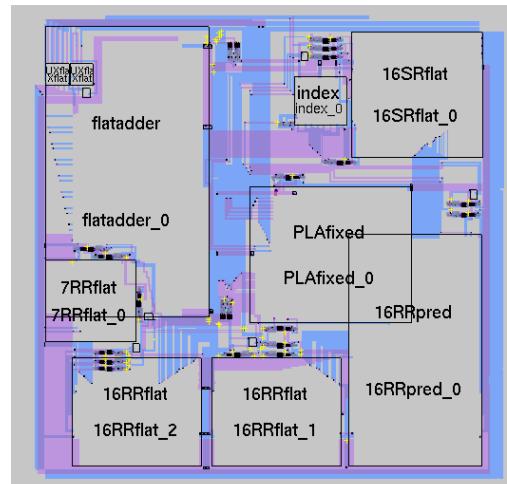


Clocking

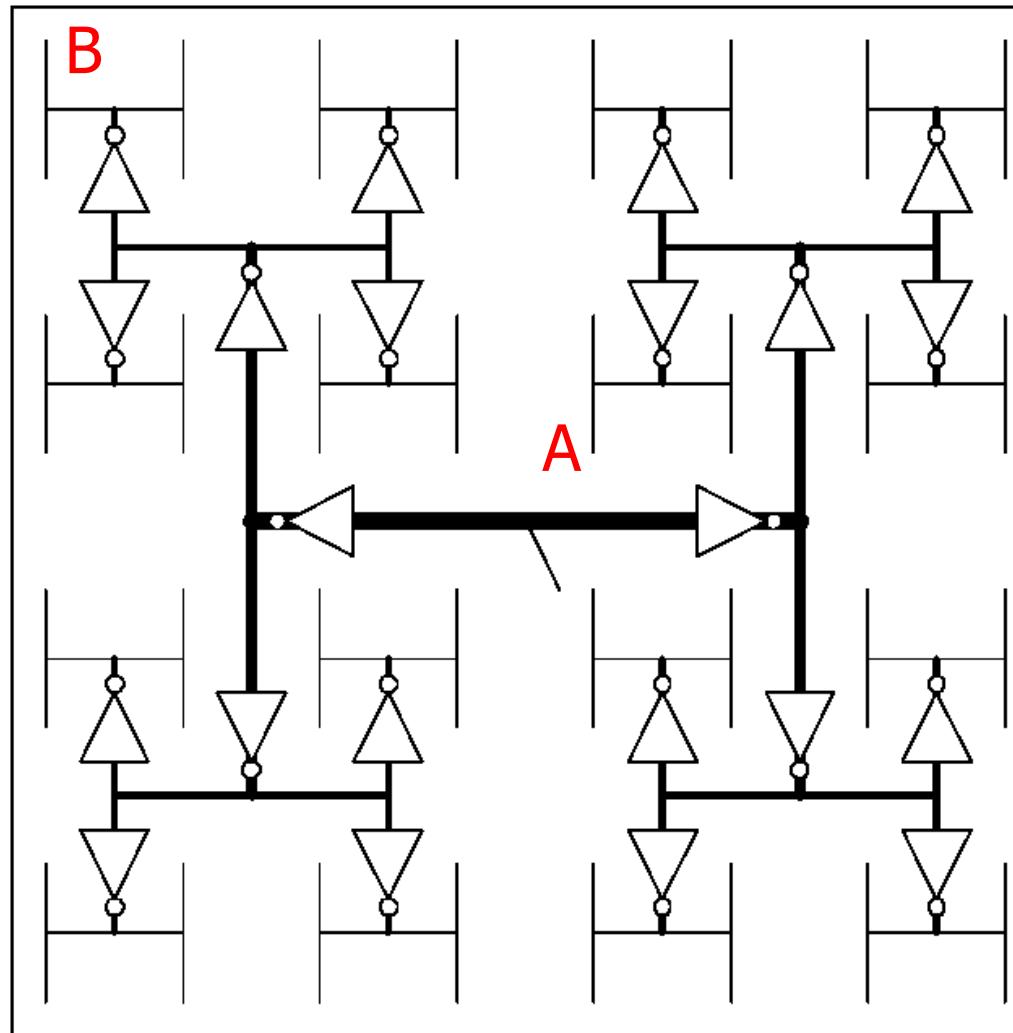
- ◆ Synchronous systems use a clock to keep operations in sequence
- ◆ Clock must be distributed to all the sequencing elements

Clock Distribution

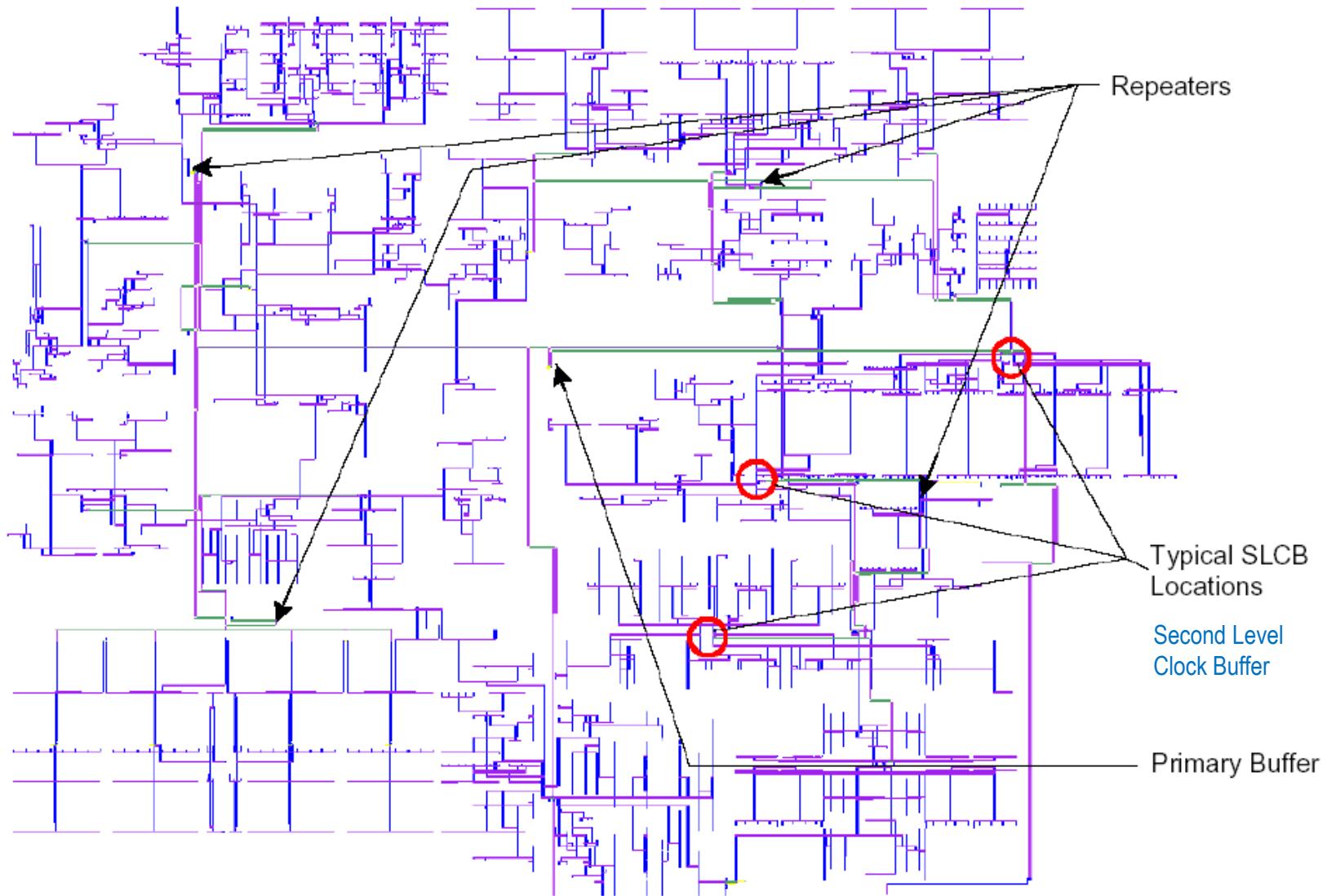
- ◆ On a small chip, the clock distribution network is just a wire
- ◆ On practical chips, the RC delay of the wire resistance and gate load is very long
- ◆ Most chips use repeaters to buffer the clock and equalize the delay



H-Trees



Itanium 2 H-Tree

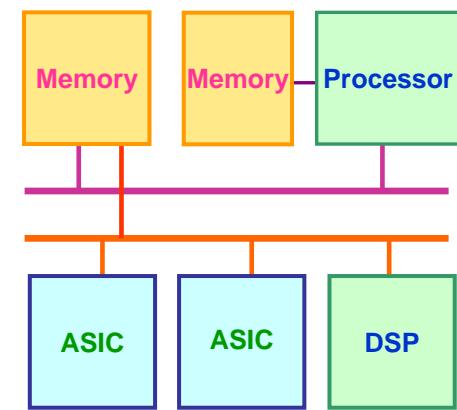
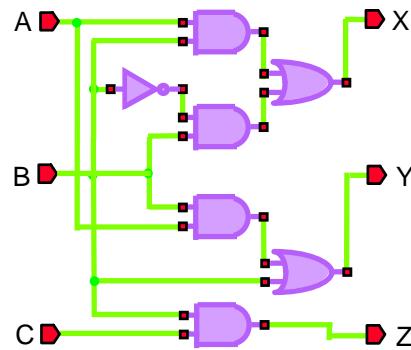
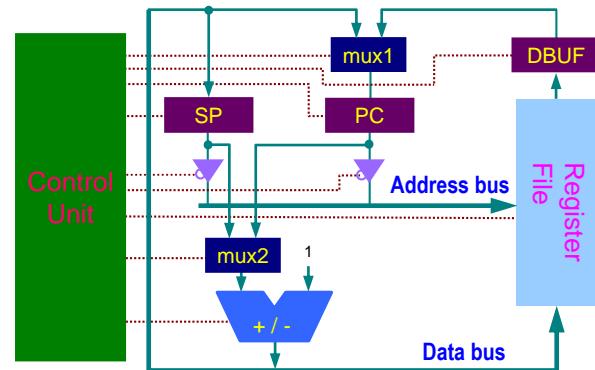
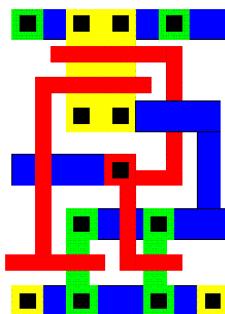
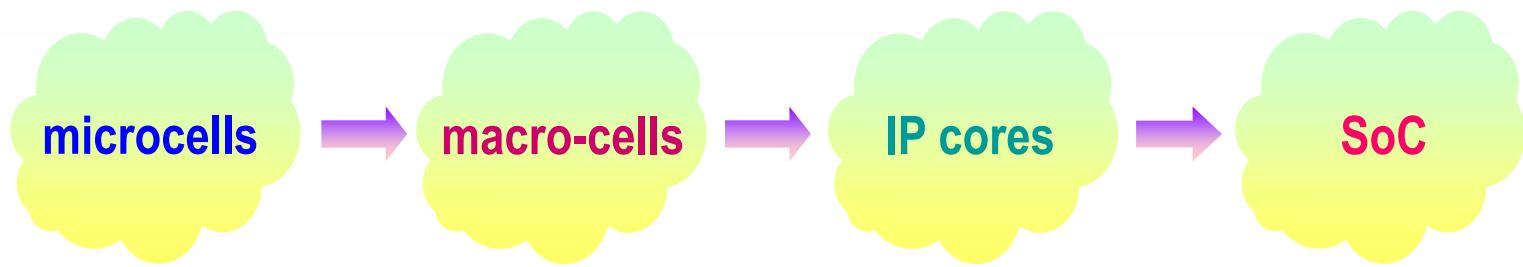


Intellectual Property Cores



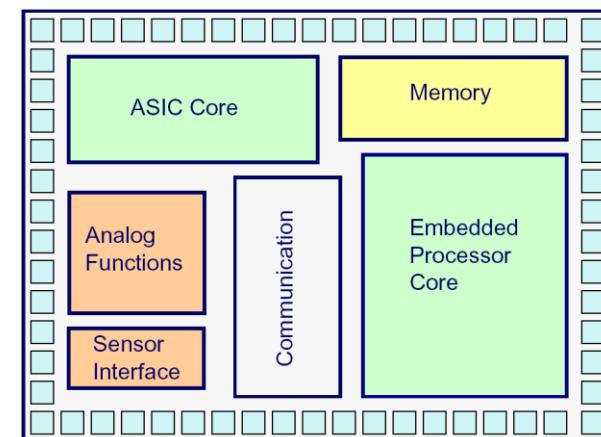
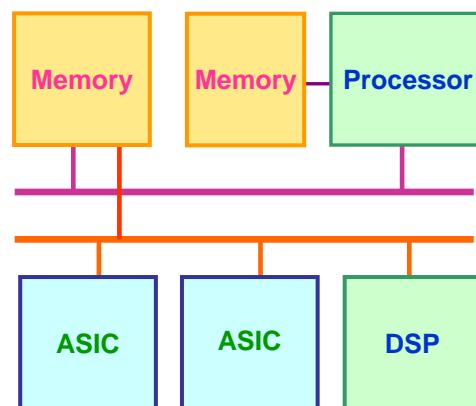
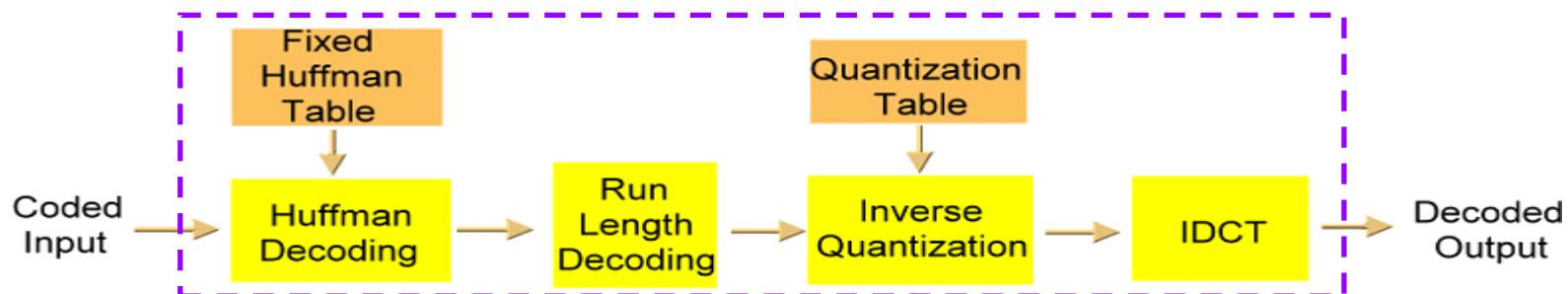
■ Intellectual property cores (IP cores)

- ◆ more complicated building blocks such as microprocessors, digital signal processors, memory modules, etc.
- ◆ can be optimized, verified, and documented to allow efficient reuses
- ◆ **Hard IP core**: the mask information of the circuit, custom built, optimized, and verified for a specific application (better performance)
- ◆ **Soft IP core**: the behavioral description (e.g., Verilog) of a circuit, which can be parameterized and synthesized for different technologies (more flexible)
- ◆ **Firm IP core**: provide a register transfer level (RTL) description of a circuit

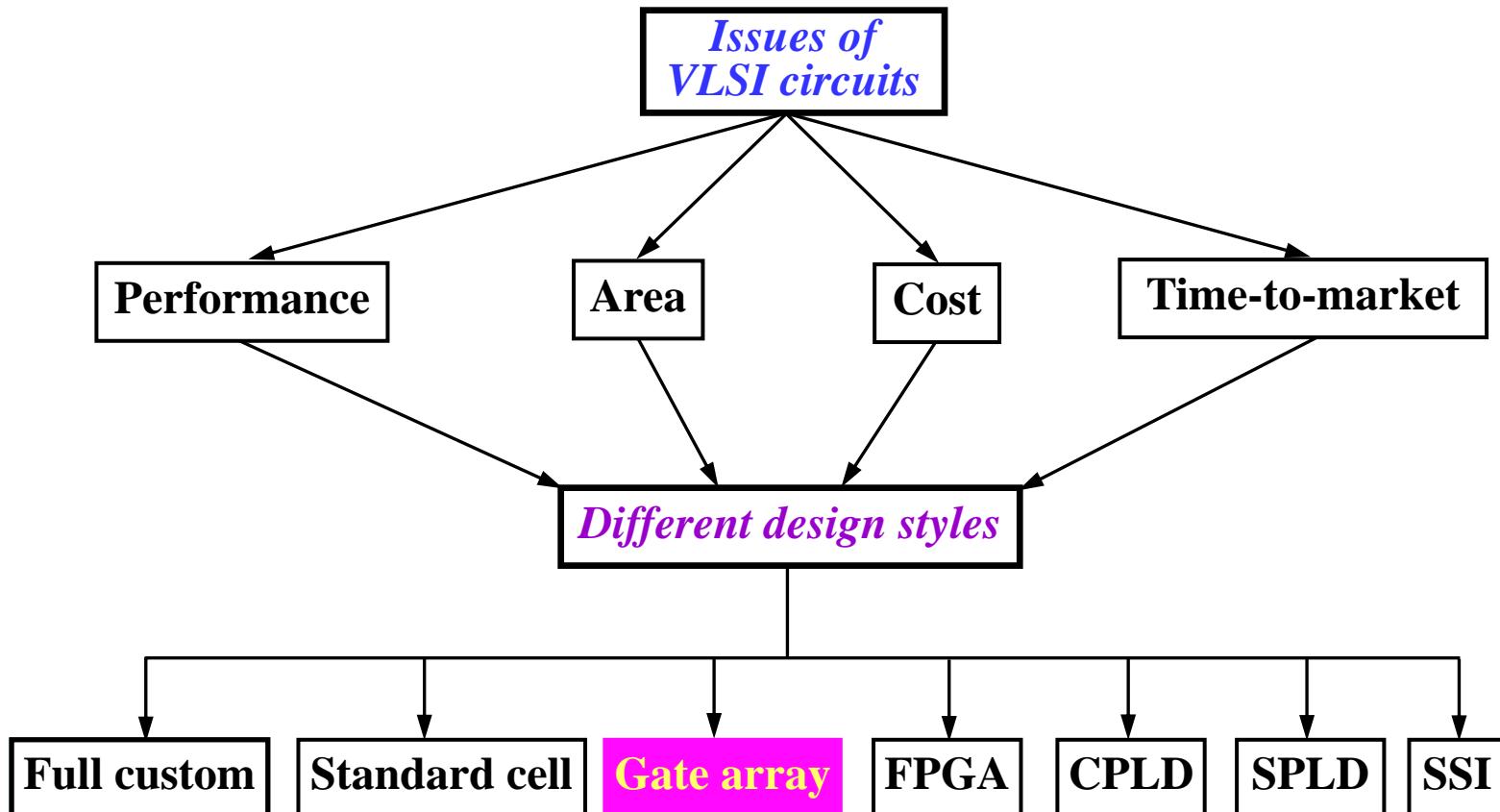


Application-specific Integrated Circuits (ASICs)

- An **ASIC** is a circuit which performs a **specific function** in a particular application
- In the ASIC market, it is important to reduce the manufacturing cost and the **time to market**



VLSI Design Styles

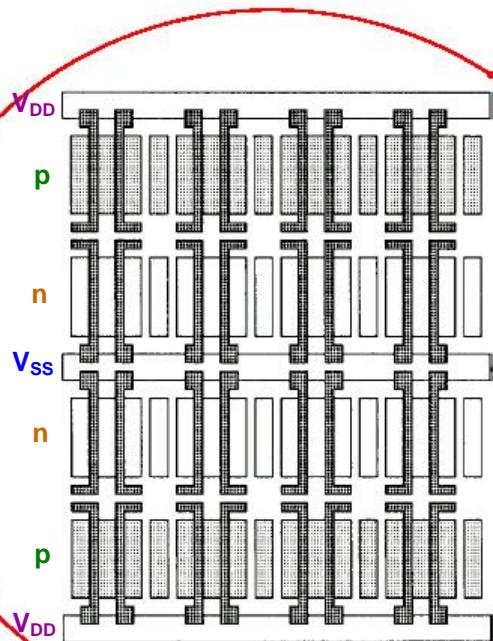


Performance, Area efficiency, Cost, Flexibility

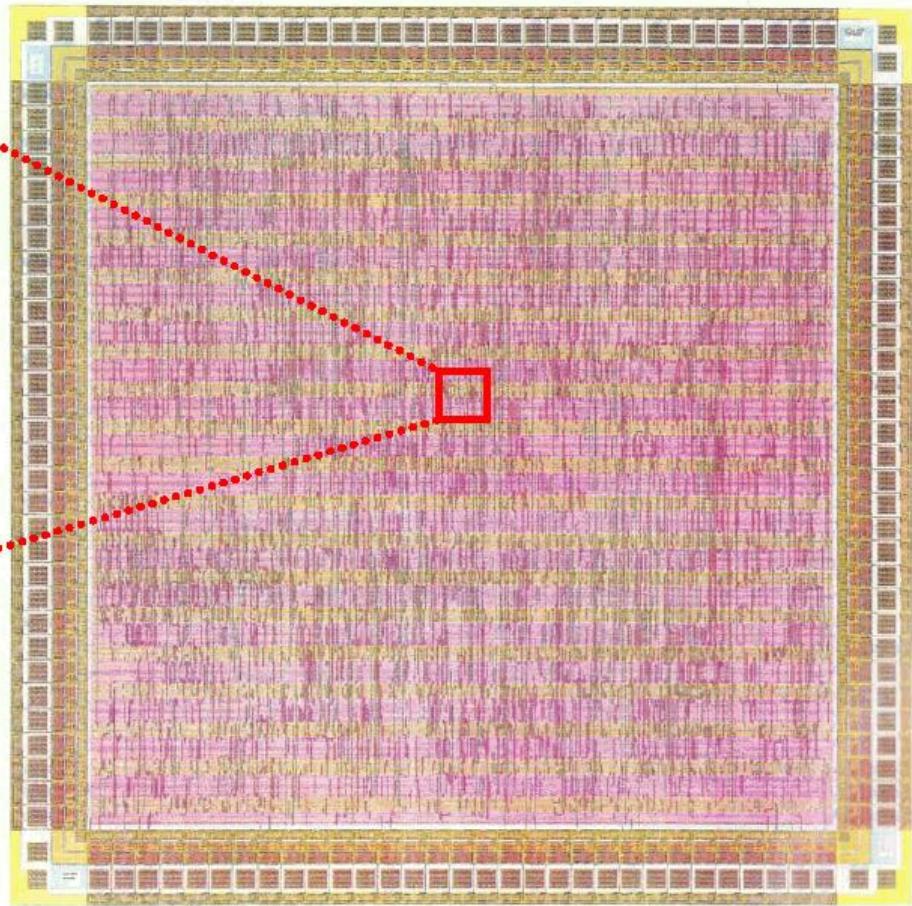
Mask-programmed Gate-array (Gate-array)

- one of special architectures for these **ASIC**
- consist of an array of **unconnected gate cells** prefabricated on a **chip**
- one can personalize a gate array by adding appropriate **metal interconnections** to it
- the **turn-around time** is reduced (the time elapsed between the submission of a design and the receipt of chips)
- cost is **lower** than a **full-custom chip**
 - ◆ since gate-arrays can be mass produced and used in many different designs
- designs are **more restrictive** than **full-custom designs** since all transistors are of fixed sizes

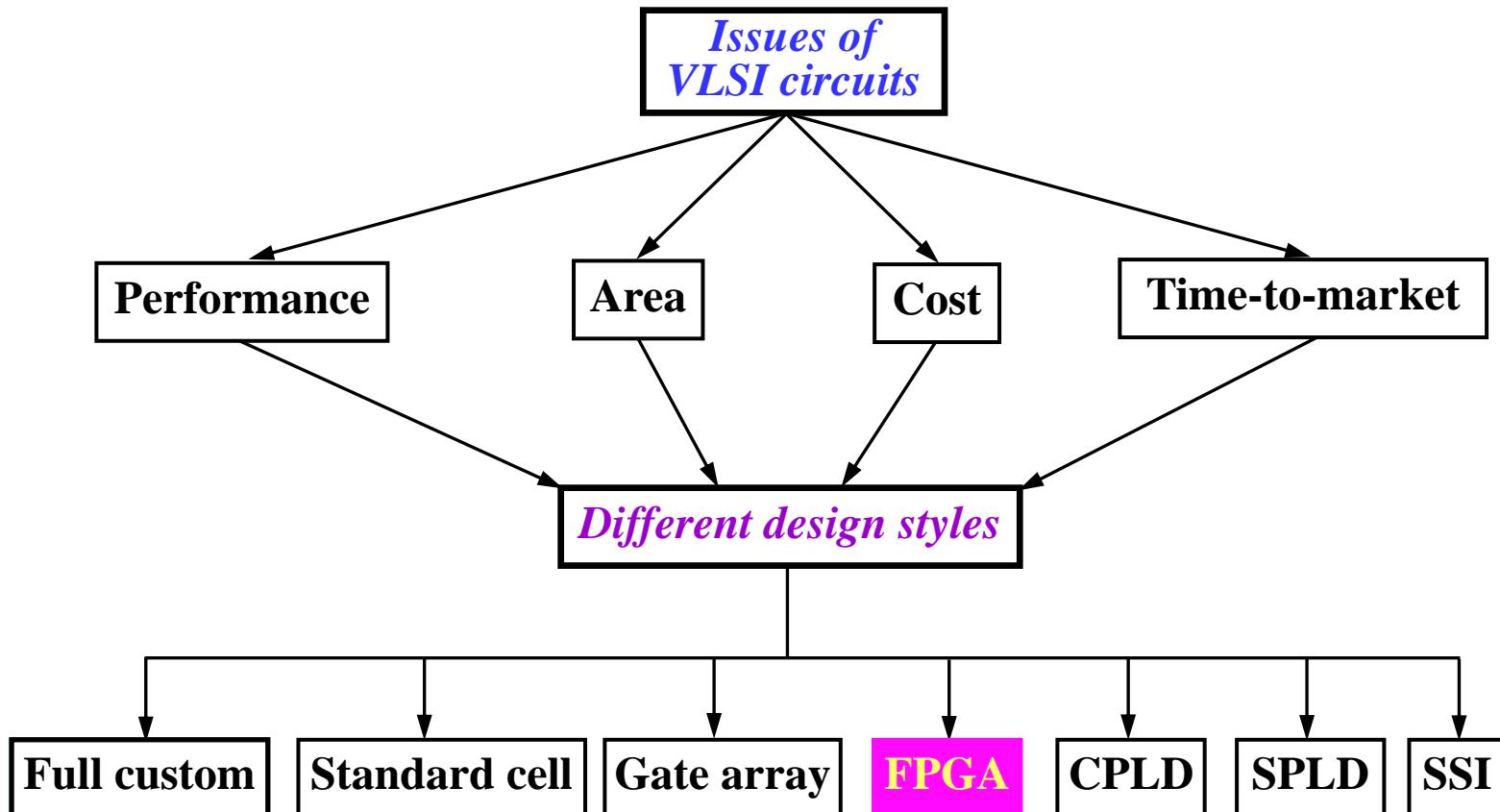
Mask Gate Array



Before customization



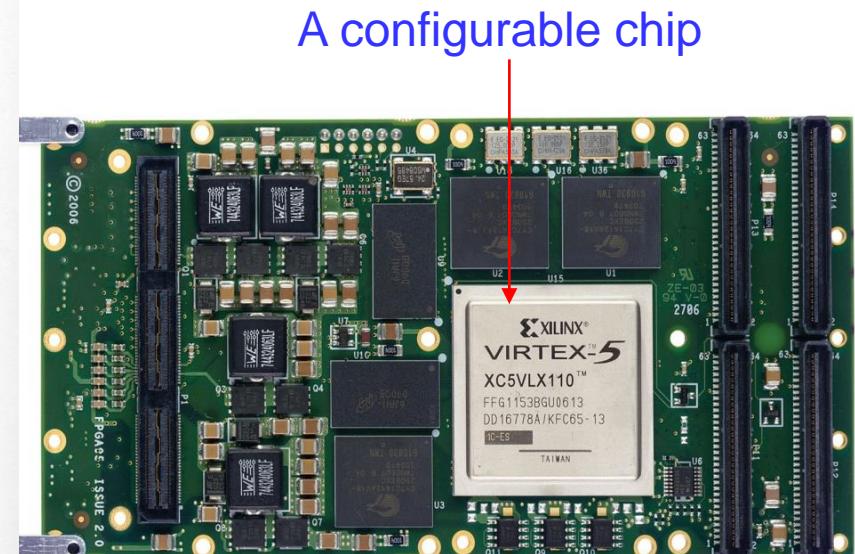
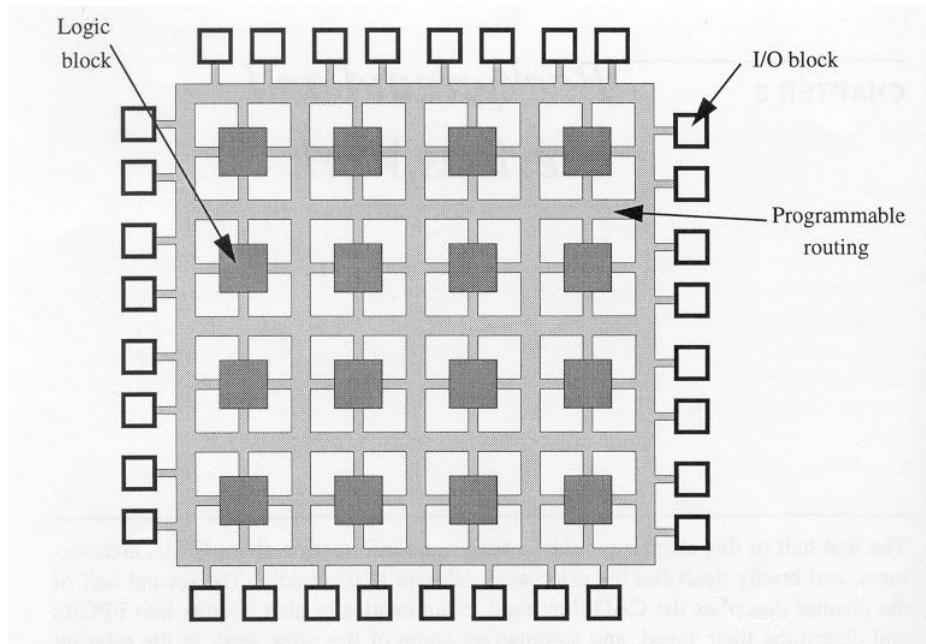
VLSI Design Styles



Performance, Area efficiency, Cost, Flexibility

Field Programmable Gate-Array (FPGA)

- In 1985, Xilinx introduced a **gate-array** structure that is **programmable** by the end users
- Each **configurable logic block (CLB)** can be programmed by the user to implement any logic function of its inputs



Characteristics of FPGAs

- Design on fabricated chips
- Low cost
- Fast time-to-market
- Large area
- Slow circuit speed
- Major venders
 - ◆ Xilinx
 - ◆ Altera



Basic Structure

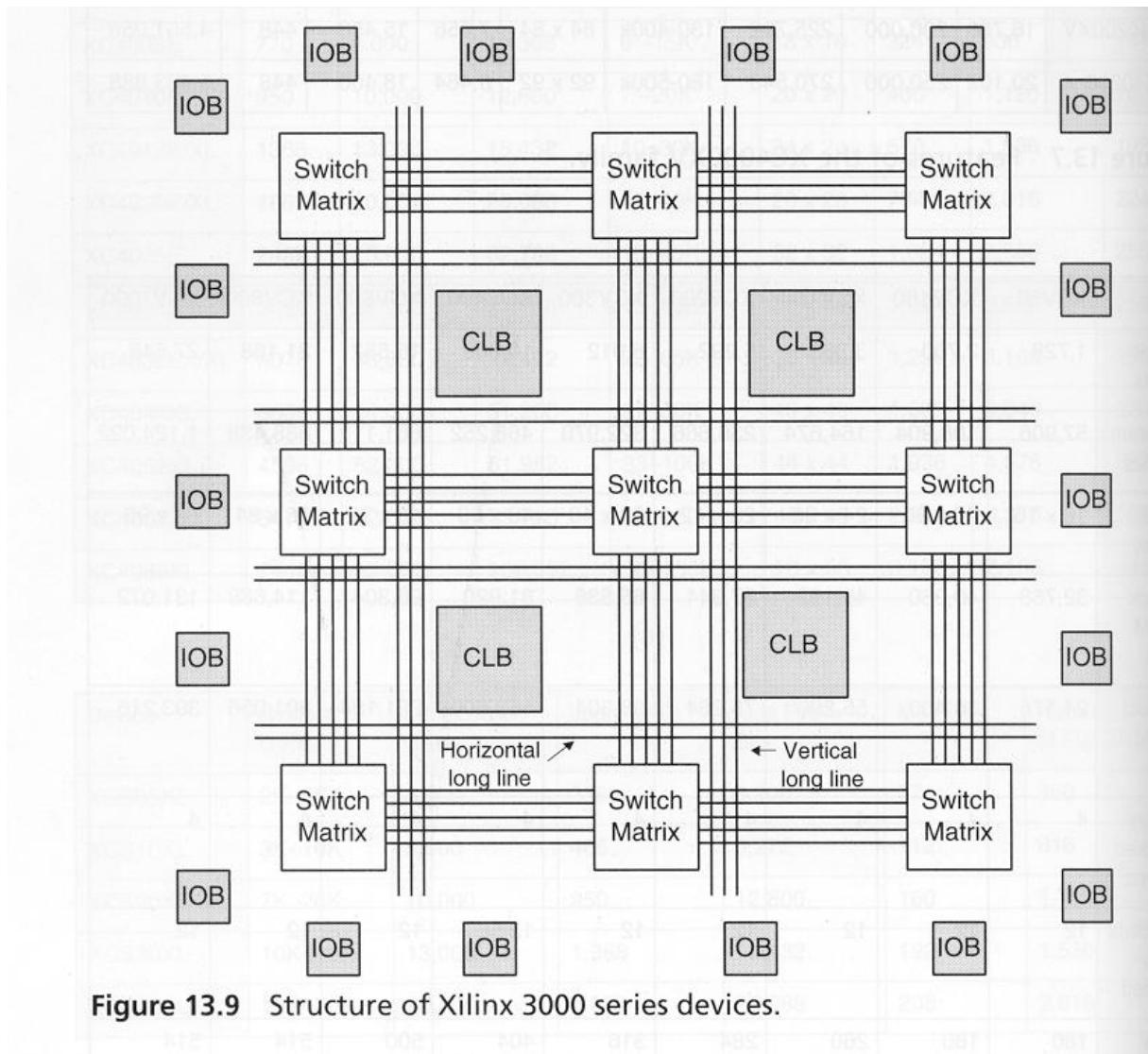


Figure 13.9 Structure of Xilinx 3000 series devices.

Configurable Logic Block (CLB)

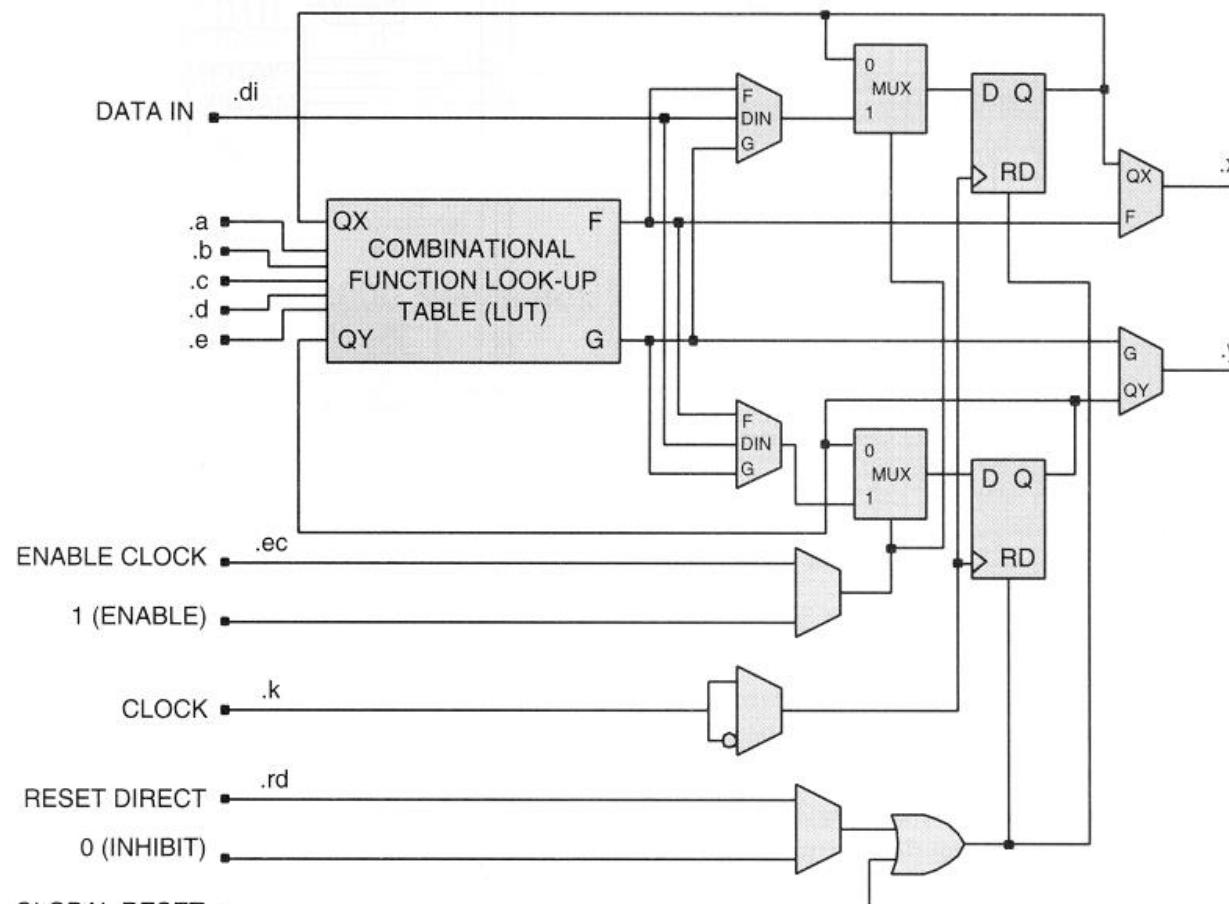


Figure 13.10 Xilinx 3000 series configurable logic block (CLB).

Multiplexer

Example: $F(A, B, C, D) = \Sigma(1, 3, 4, 11, 12, 13, 14, 15)$

A	B	C	D	F
0	0	0	0	0 $F = D$
0	0	0	1	1
0	0	1	0	0 $F = D$
0	0	1	1	1
0	1	0	0	1 $F = D'$
0	1	0	1	0
0	1	1	0	0 $F = 0$
0	1	1	1	0
1	0	0	0	0 $F = 0$
1	0	0	1	0
1	0	1	0	0 $F = D$
1	0	1	1	1
1	1	0	0	1 $F = 1$
1	1	0	1	1
1	1	1	0	1 $F = 1$
1	1	1	1	1

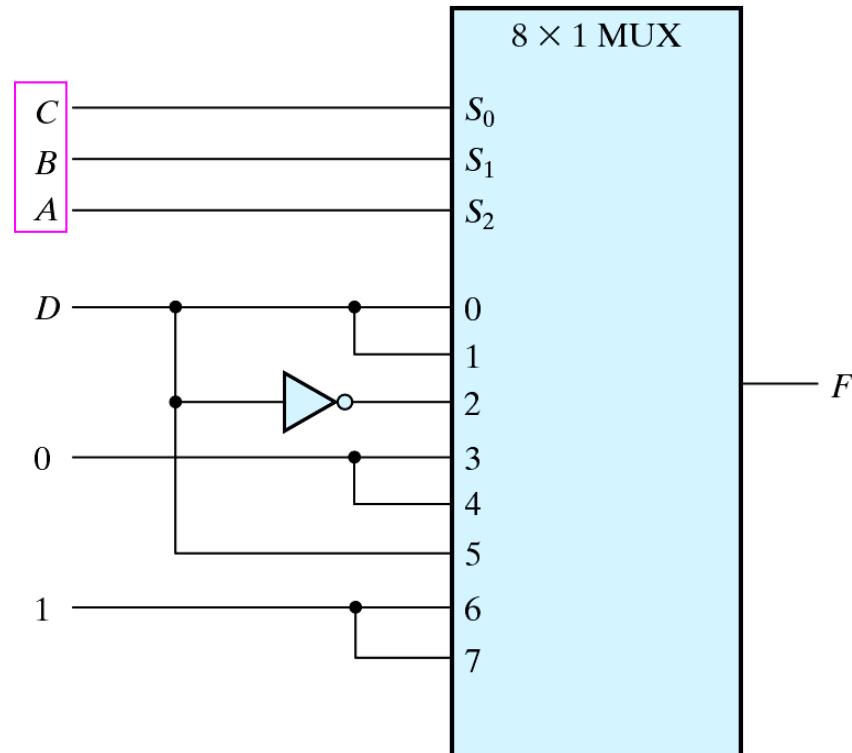
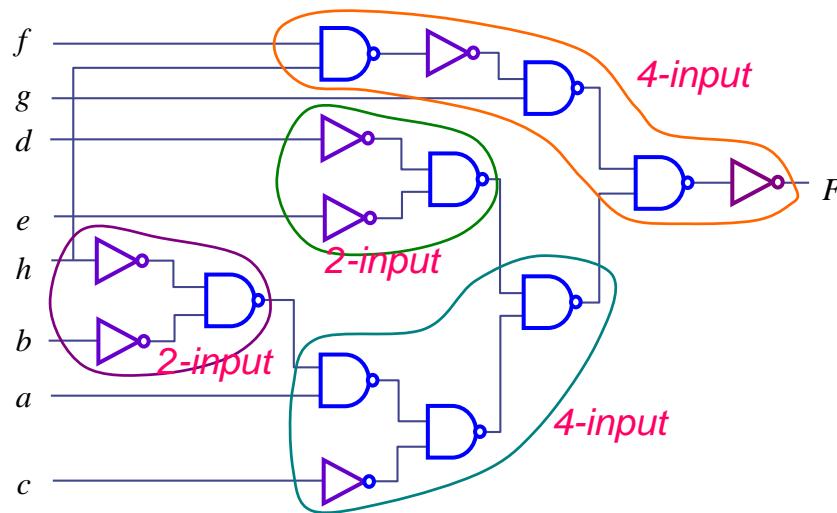


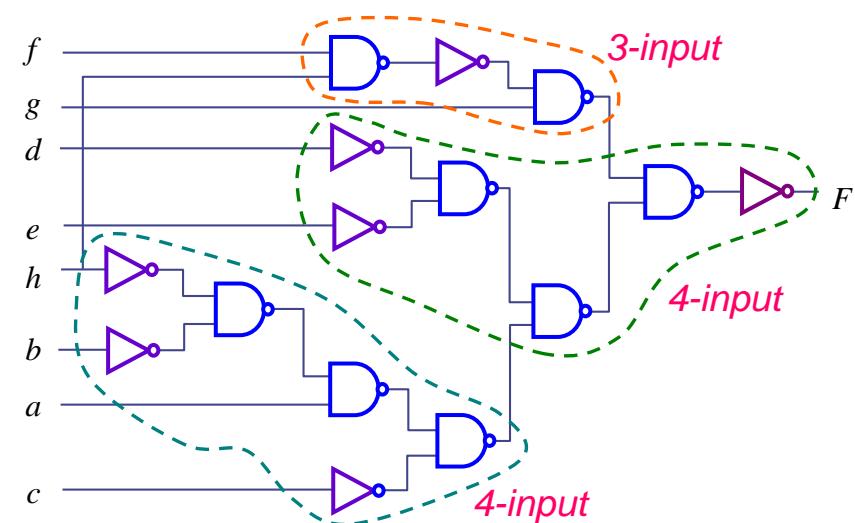
Fig. Implementing a four-input function with a multiplexer

Lookup Table-Based Technology Mapping

- A k -input LUT (k -LUT) can implement any function of up to k inputs



*a mapping with 4
4-input, 1-output LUTs,
delay depth = 3 LUTs*



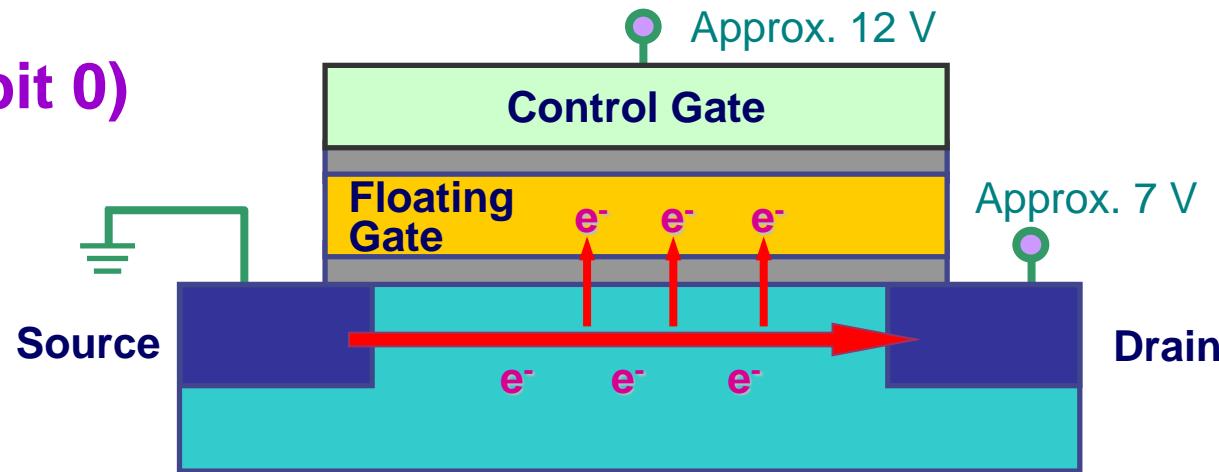
*an optimal mapping with 3
4-input, 1-output LUTs,
delay depth = 2 LUTs*

EEPROM and Flash (1/2)

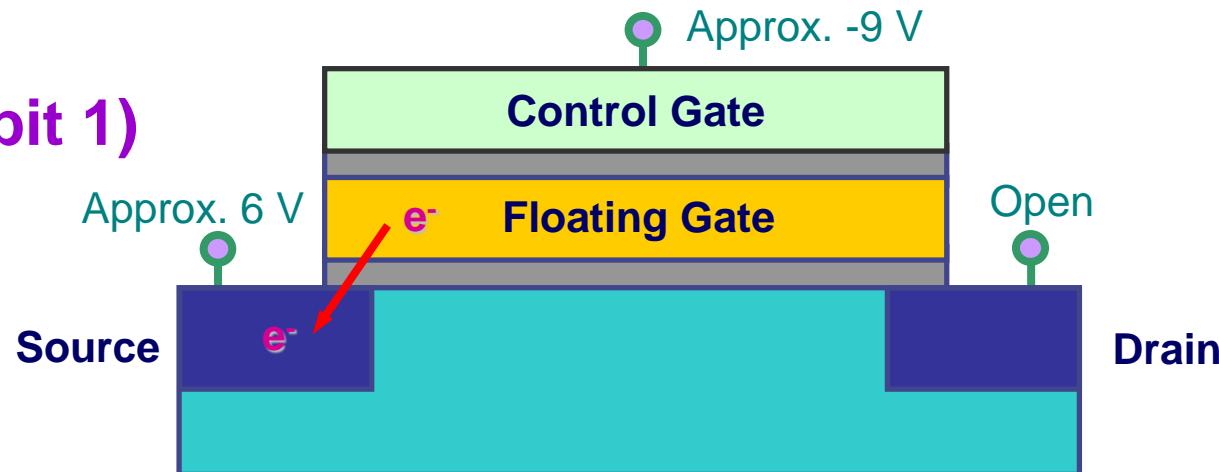


Flash

Write (bit 0)



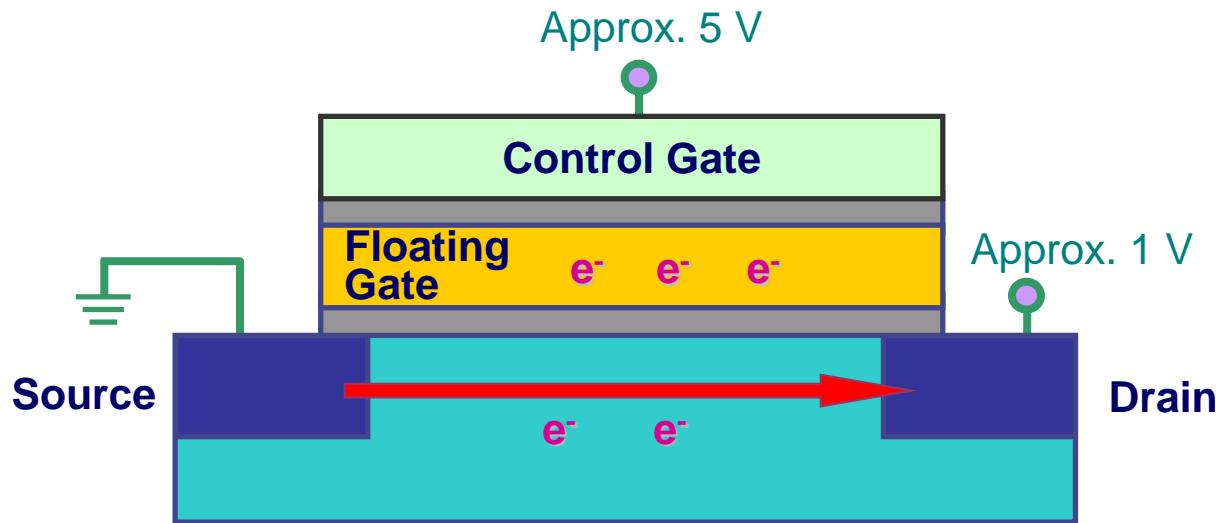
Erase (bit 1)



EEPROM and Flash (2/2)

Flash

Read



Memory Cell 中所存的值由 Drain 與 Source 之間的電流大小來判斷

Design and Technology Styles



■ Custom design

- ◆ Mostly manual design, long design cycle
- ◆ High performance, high volume
- ◆ Microprocessors, analog, leaf cells, IP ...

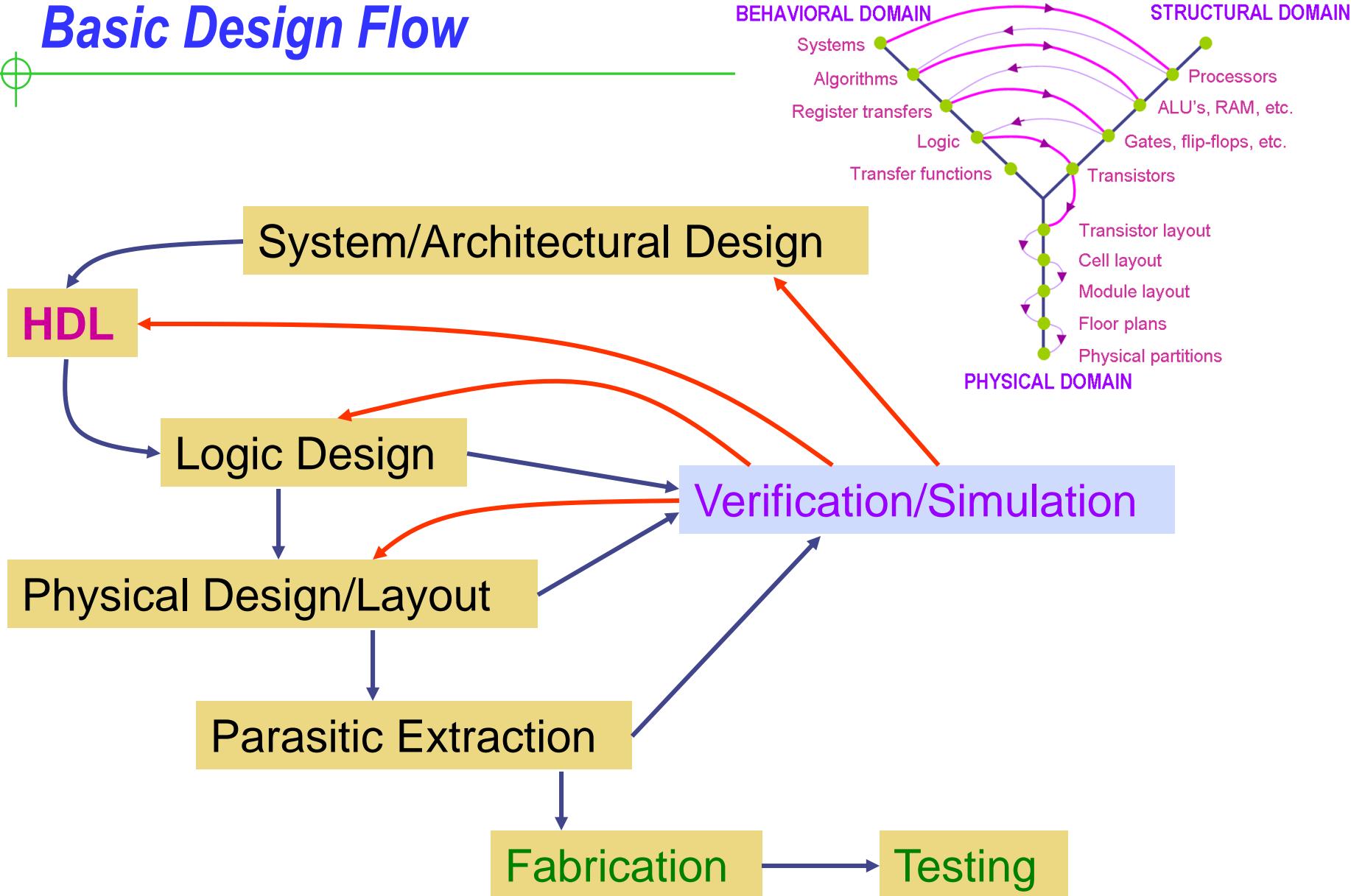
■ Standard cell

- ◆ Pre-designed cells, CAD, short design cycle
- ◆ Medium performance, ASIC

■ FPGA/CPLD

- ◆ Pre-fabricated, fast automated design, low cost
- ◆ Prototyping, reconfigurable computing

Basic Design Flow



HDL-Based Design (1/2)



■ 1980's

- ◆ Hardware Description Languages (HDL) were conceived to facilitate the **information exchange** between design groups

■ 1990's

- ◆ the increasing computation power led to the introduction of **logic synthesizers** that can translate the **description** in HDL into a synthesized **gate-level net-list** of the design

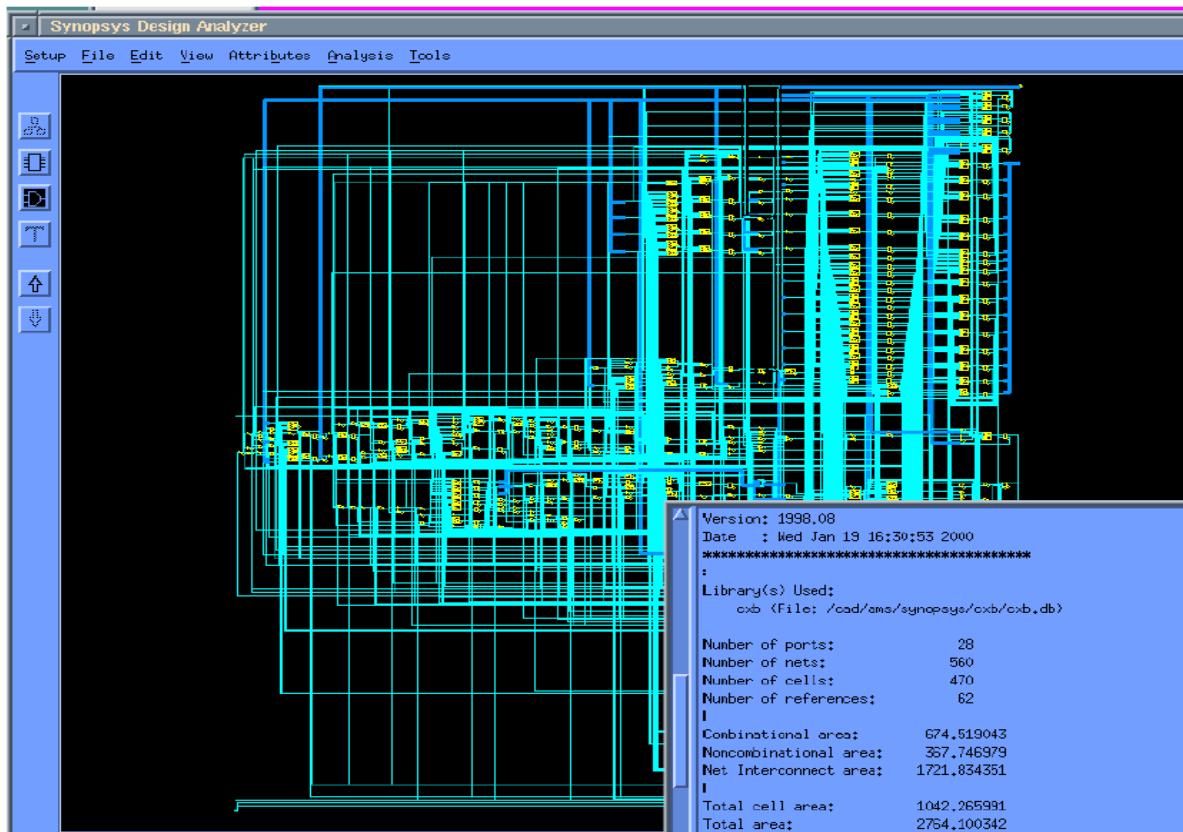
■ 2000's

- ◆ modern **synthesis algorithms** can **optimize** a digital design and **explore different alternatives** to identify the **design** that **best meets** the requirements

HDL-Based Design (2/2)

The design is synthesized and mapped into the target technology

- the logic gates have one-to-one equivalents as standard cells in the target technology



Why do we need HDLs ?

- HDL can describe both circuit **structure** and behavior
 - ◆ Schematics describe only circuit **structure**
 - ◆ C language describes only **behaviors**
- Provide **high level abstraction** to speed up design
- High portability and readability
- Enable rapid prototyping
- Support **different hardware styles**

What do we need from HDLs ?



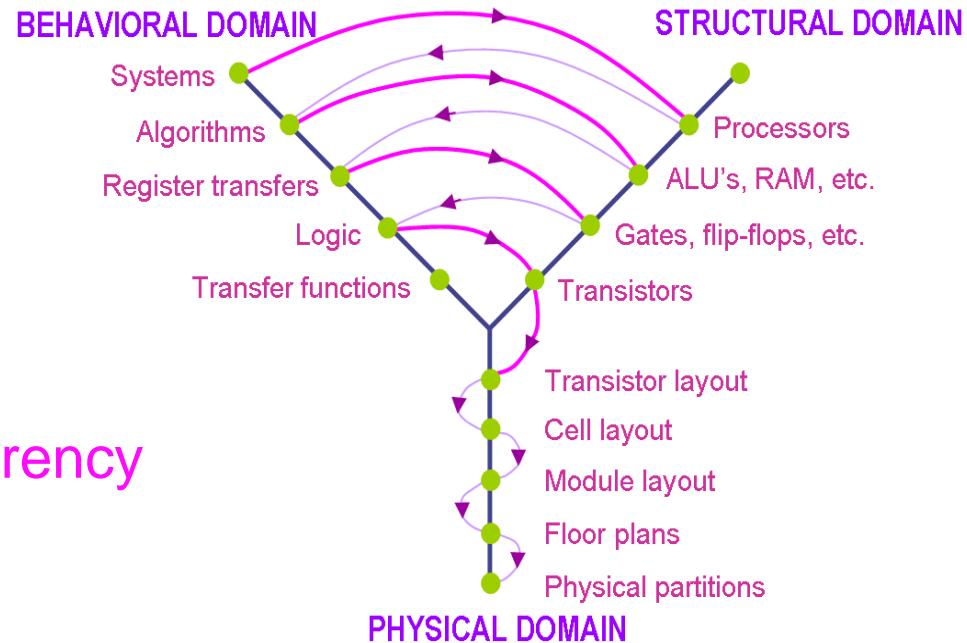
Describe

- ◆ Combinational logic
- ◆ Level sensitive storage devices
- ◆ Edge-triggered storage devices

Provide different levels of abstraction and support hierarchical design

- ◆ System level
- ◆ RTL level
- ◆ Gate level
- ◆ Transistor level
- ◆ Physical level

Support for hardware concurrency



Two major HDLs



Verilog

- ◆ Slightly better at gate/transistor level
- ◆ Language style close to C/C++
- ◆ Pre-defined data type, easy to use

VHDL

- ◆ Slightly better at system level
- ◆ Language style close to Pascal
- ◆ User-defined data type, more flexible

Equally effective, personal preference

Verification Methods



- Check the correctness without actually fabricating
- Prototyping
 - ◆ Rapid system prototyping using **FPGA** (real time)
- Simulation
 - ◆ Make a **computer model** of all relevant aspects of the circuit
 - ◆ Execute the model for a set of **input signals**
 - ◆ Observing the **output signals**
 - ◆ Is **difficult** to have an **exhaustive test** of a circuit of reasonable size
- Formal verification
 - ◆ The use of **mathematical methods** to prove that a circuit is correct
 - ⊕ Check the equivalence of two descriptions of a circuit
 - ◆ Perform these proofs by **computers**



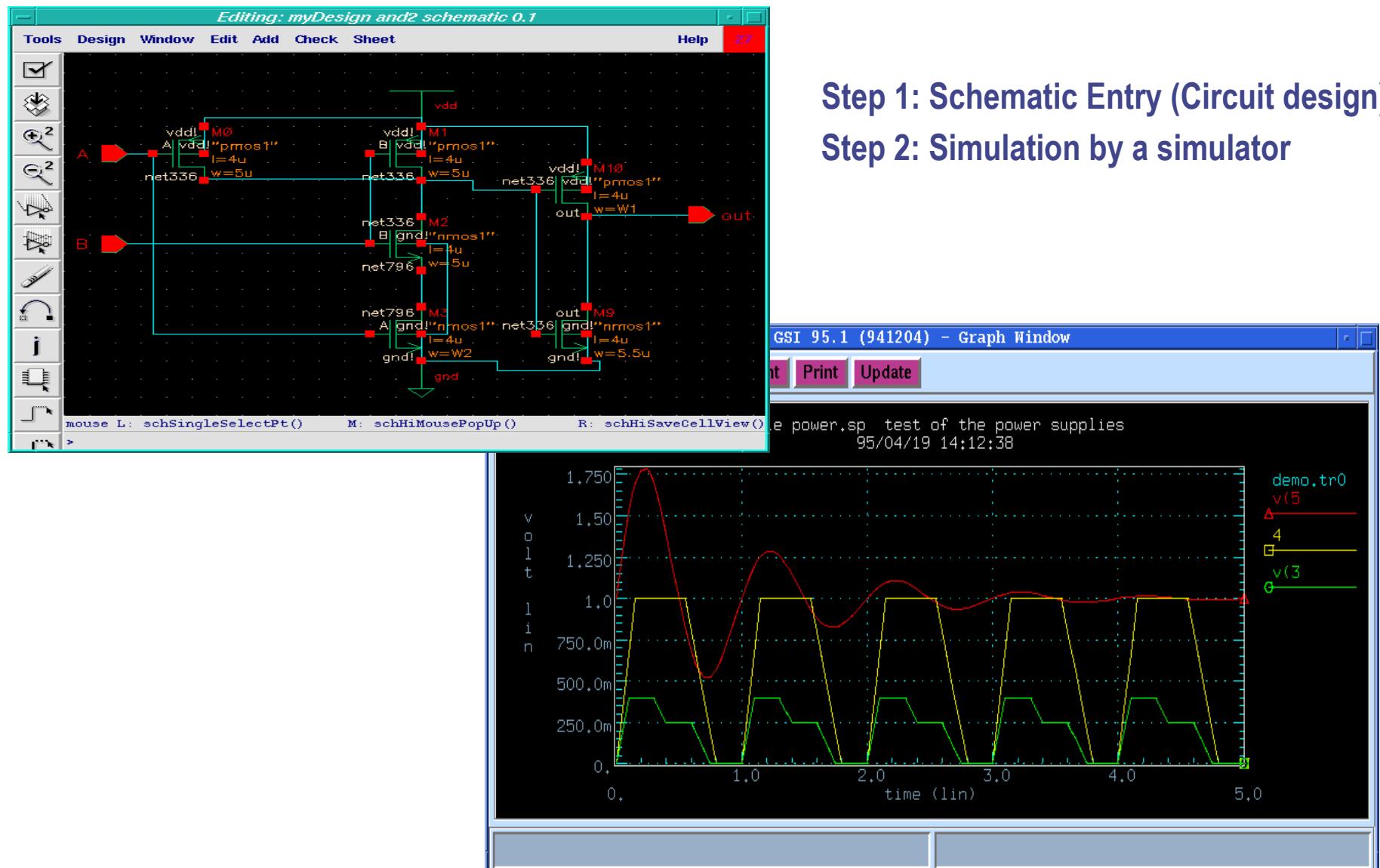
Simulation (1/2)

- A chip design must be **simulated** many times **at different levels** of abstraction to ensure its correctness
- The objective of simulation is to **verify** the functionality of a circuit and to **predict** its performance
- Circuit simulation can be done at **multiple levels** ranging from **transistor level** to **behavioral level**
- **Transistor level** simulation
 - ◆ basic elements: transistors, resistors, capacitors, and inductors
 - ◆ the **most accurate result** at the cost of **long simulation times**
- **Switch level** simulation
 - ◆ **transistors** is modeled as **switches** with propagation delays
 - ◆ significantly reduce the simulation time

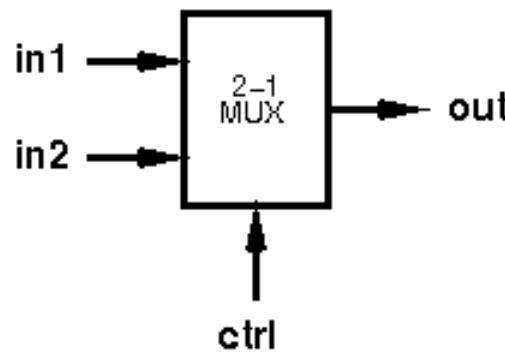
Simulation (2/2)

- Gate level simulation
- Behavioral level simulation
 - ◆ is used if there is **no** need to determine the **internal condition** of a building block
- Mixed-level simulation
 - ◆ different parts of a system are represented differently
 - ◆ is commonly used to analyze a **complex system**
 - ◆ the **critical parts** of the system can be modeled at **transistor or switch** level while the **rest** of the circuit can be modeled at behavioral level

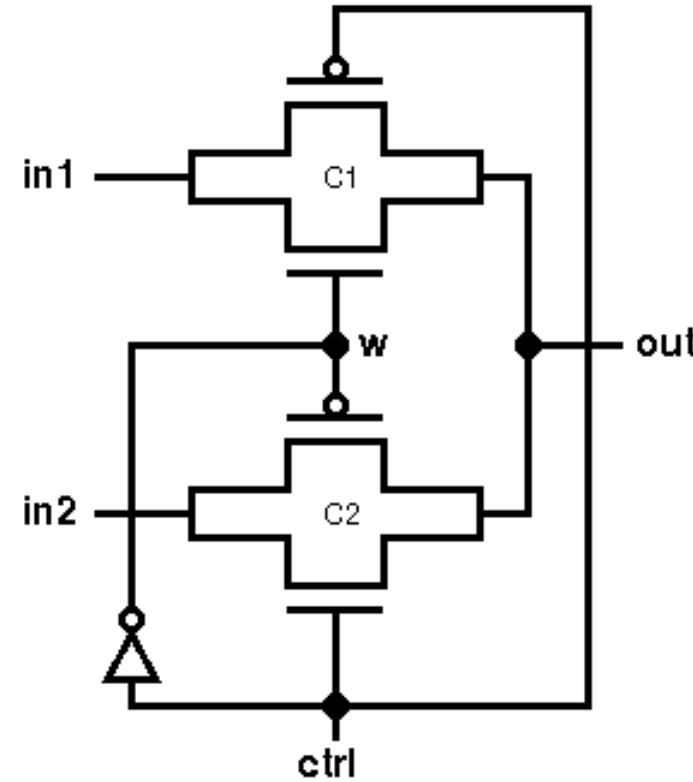
Circuit Level Simulation



Switch Level Simulation



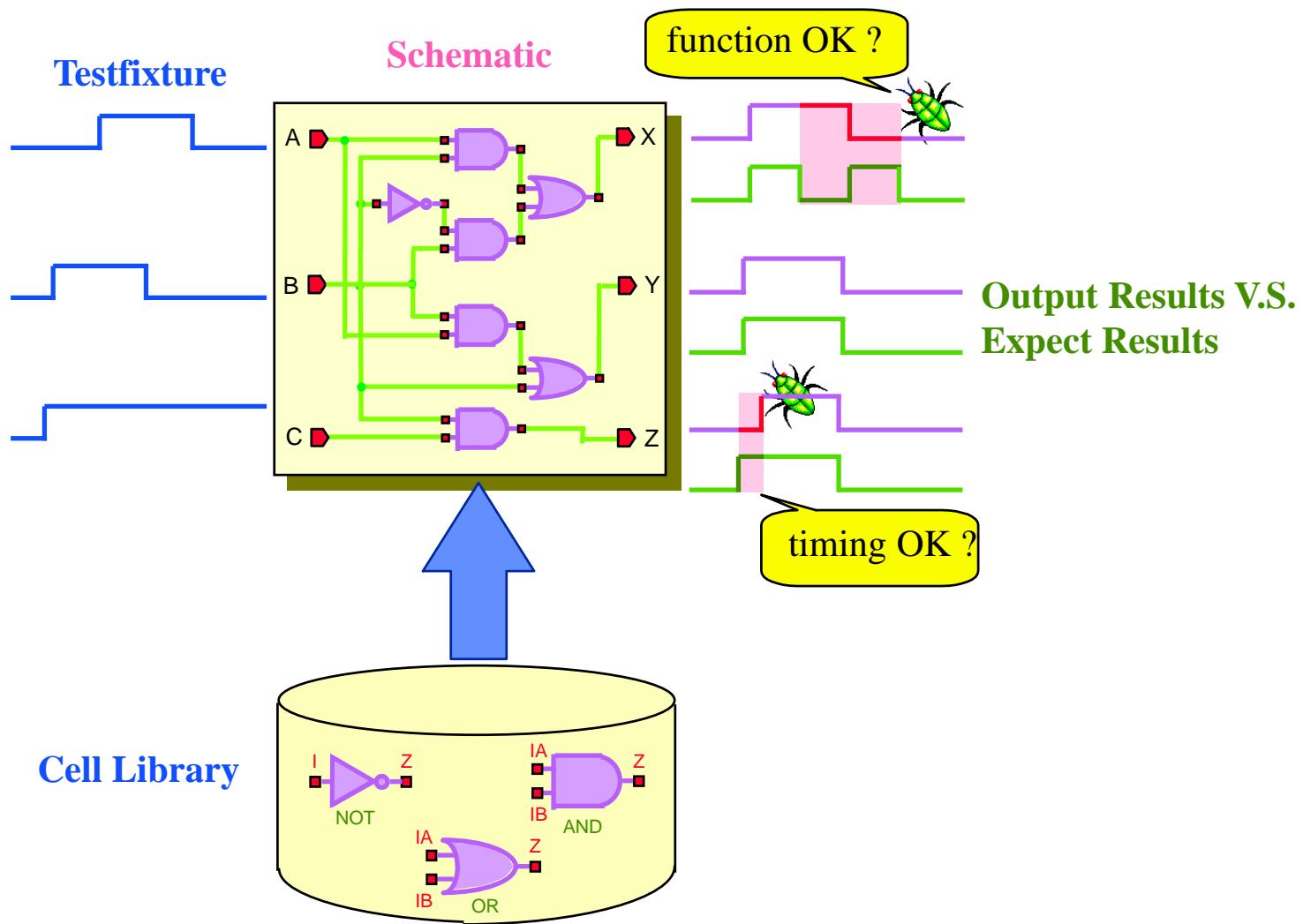
Gate representation



Switch-level model

All the elements are regarded as switch !

Gate Level Simulation

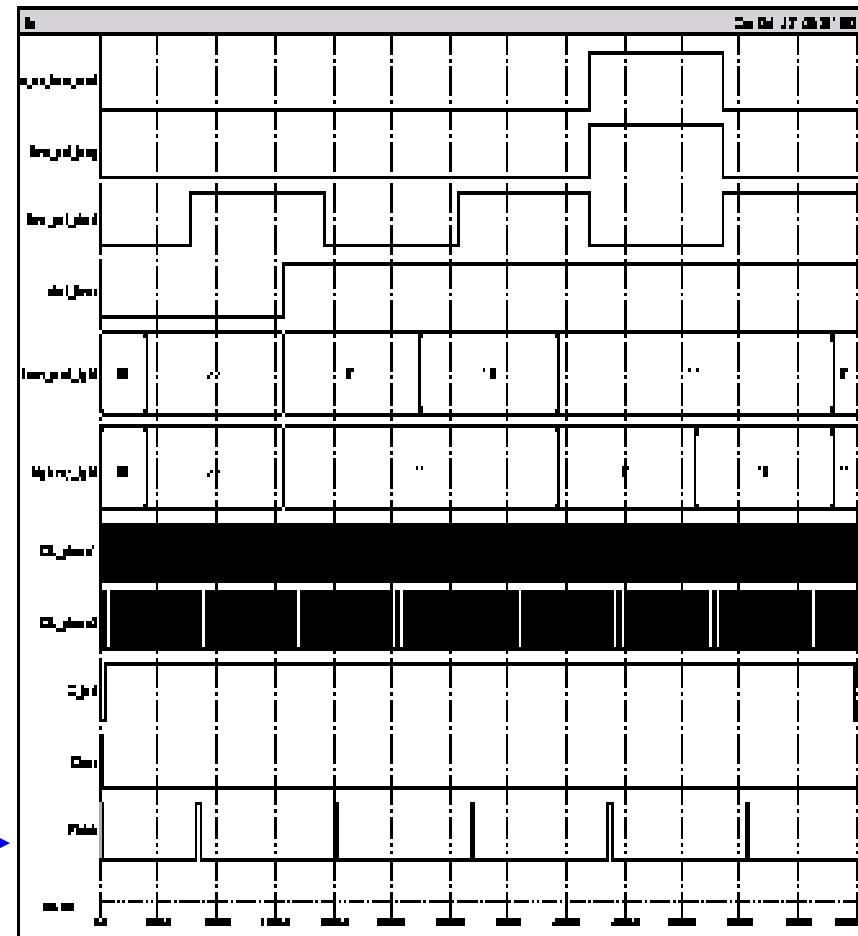


Behavior Simulation

```
ARCHITECTURE behavioral OF full_adder IS
-- carry() and sum()
FUNCTION carry(a, b, cin: BIT) RETURN BIT IS
BEGIN
    RETURN (a AND b) OR (b AND cin) OR (a AND cin);
END carry;
FUNCTION sum(a, b, cin: BIT) RETURN BIT IS
BEGIN
    RETURN a XOR b XOR cin;
END sum;
-- Relationship between the outputs and inputs
-- with propagation delay information
BEGIN
    s <= sum(a, b, cin) AFTER 1 ns;
    cout <= carry(a, b, cin) AFTER 0.8 ns;
END behavioral;
```

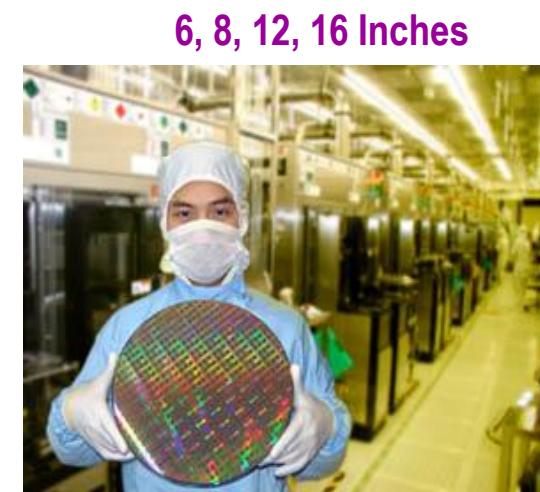
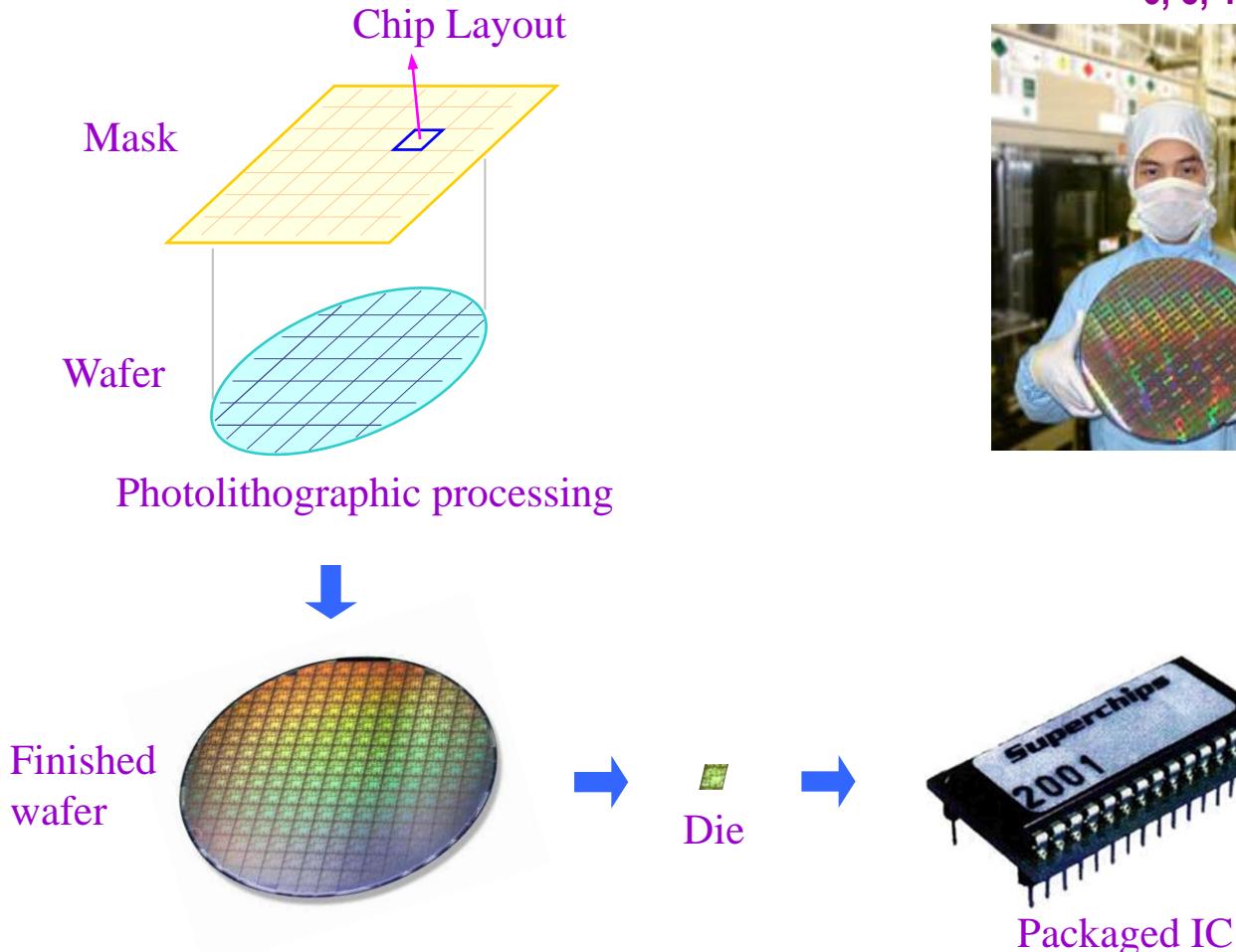
HDL Coding

HDL Simulation



Fabrication

Simplified fabrication process of an IC



Material Classification



p-type n-type

B 硼 5	C 碳 6	N 氮 7
10.811	14.007	14.007

Al 鋁 13	Si 矽 14	P 磷 15
26.982	28.086	30.974

Ga 鎵 31	Ge 鍺 32	As 砷 33
69.723	72.61	74.992

In 銦 49	Sn 錫 50	Sb 銻 51
114.82	118.71	121.76

Tl 鉛 81	Pb 鉛 82	Bi 銻 83
204.38	207.2	208.98

Insulators

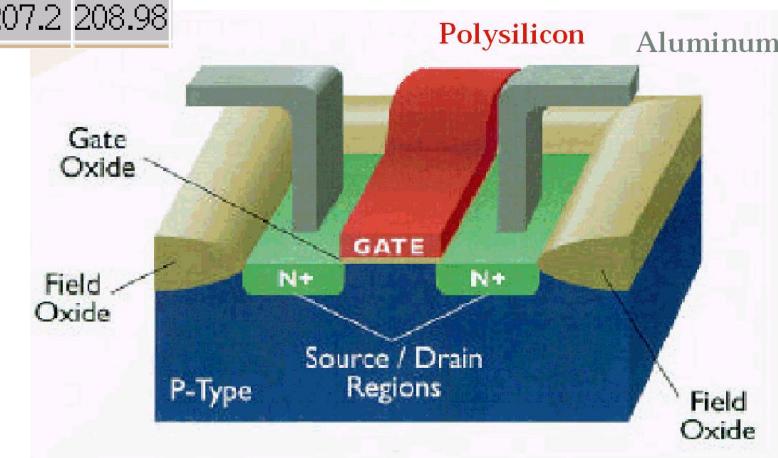
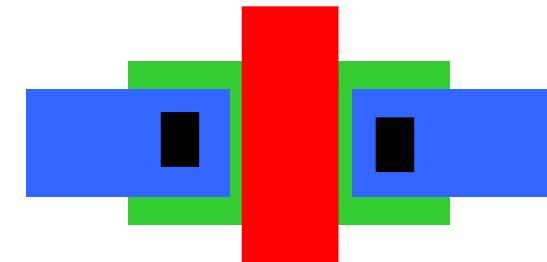
- ◆ Glass, diamond, silicon oxide

Semiconductors

- ◆ Germanium (Ge), Silicon (Si),
Gallium Arsenide (GaAs)

Conductors

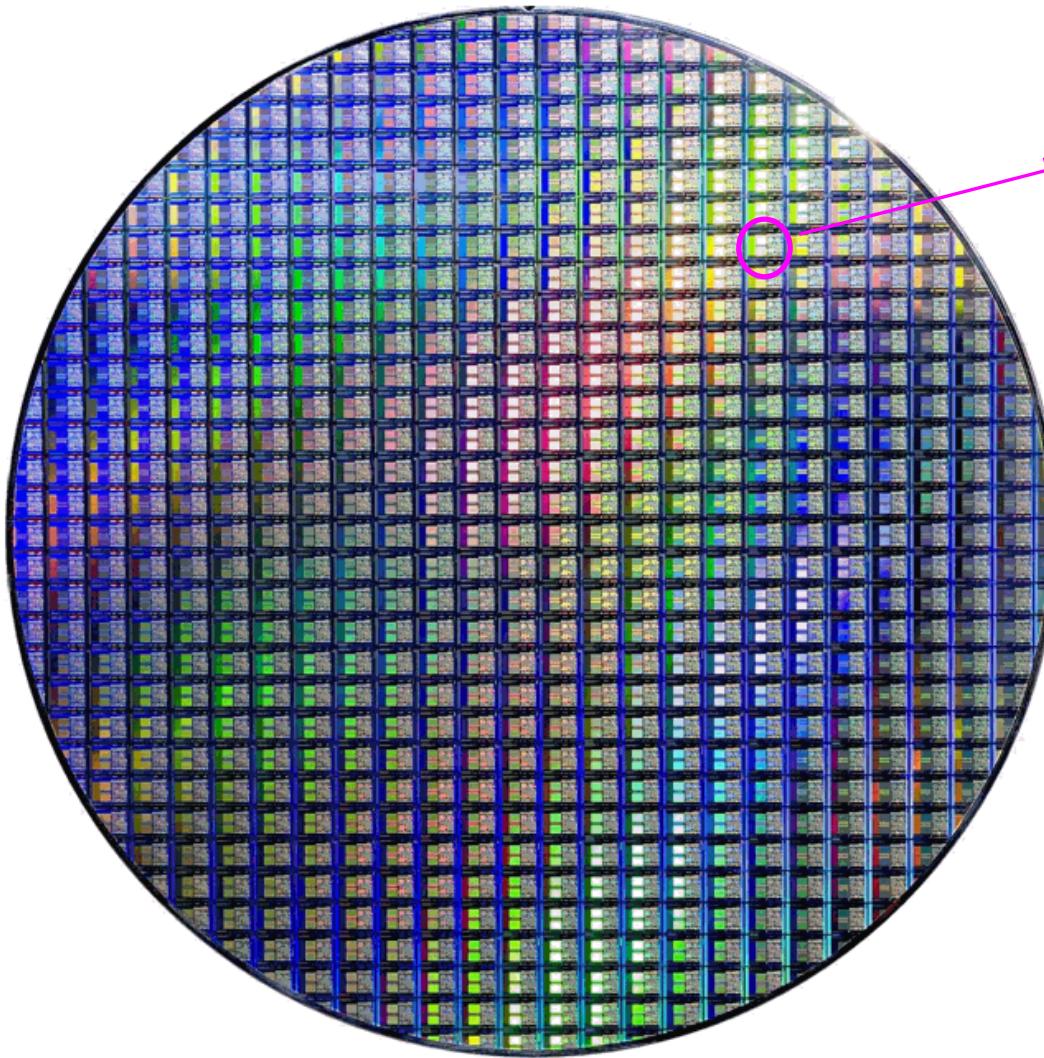
- ◆ Aluminum, copper, gold



Increasing conductivity



Wafer

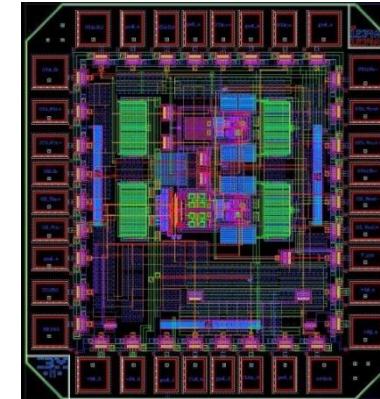
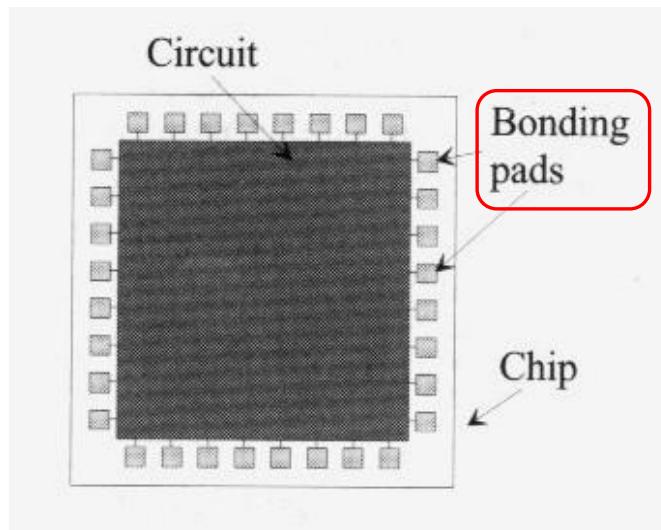


A single die

Packaging (1/2)

■ Chip layout including a circuit and a ring of bounding pads (pad frame)

- ◆ Signal and power connections of the circuit are connected to the bounding pads
- ◆ A signal buffer and a protection circuit are provided for each connection between a bounding pad and the circuit
- ◆ The protection circuit protects the circuit against potential damages (e.g., static electricity)



Packaging (2/2)

Chip mounted in an IC package

- **Bounding wires** connect bounding pads to the pins of an IC package
- To fully utilize the processing power of a chip, **enough I/O pins** must be provided
- **Dual-in-line package (DIP)**: a severe limitation on the number of pins available

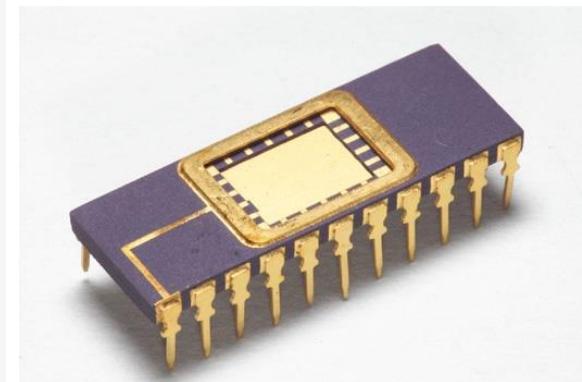
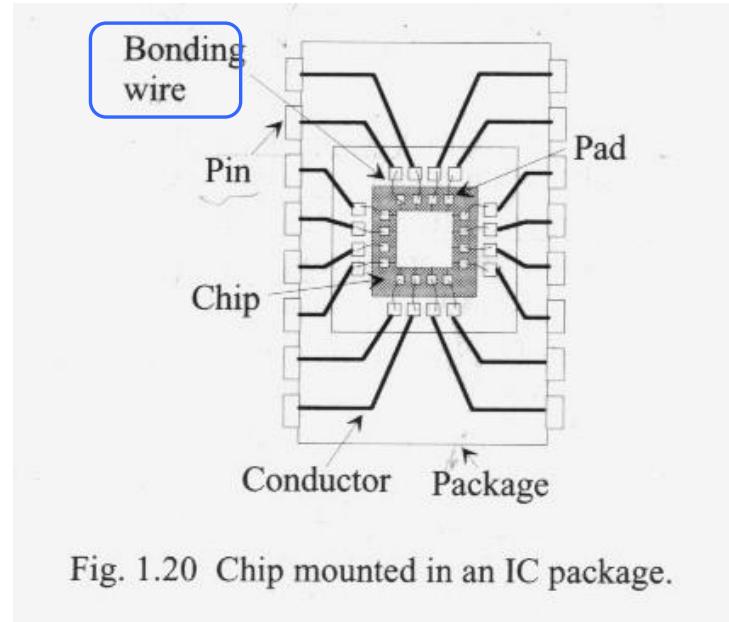
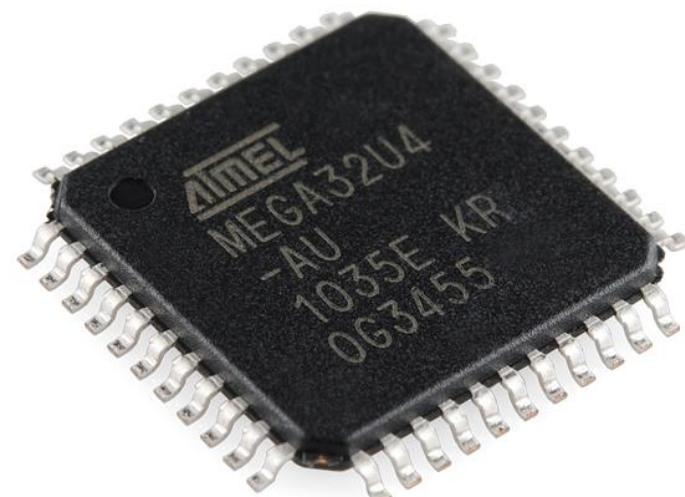
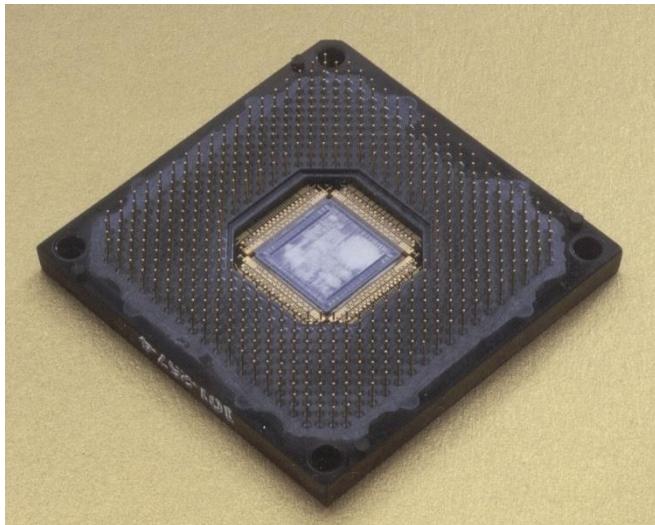


Fig. 1.20 Chip mounted in an IC package.

Package Functions

- Epoxy Molding Compound, EMC: 環氧樹脂封裝材料
 - Electrical connection of signals and power from chip to board
 - Protects chip from mechanical damage
 - Removes heat produced on chip

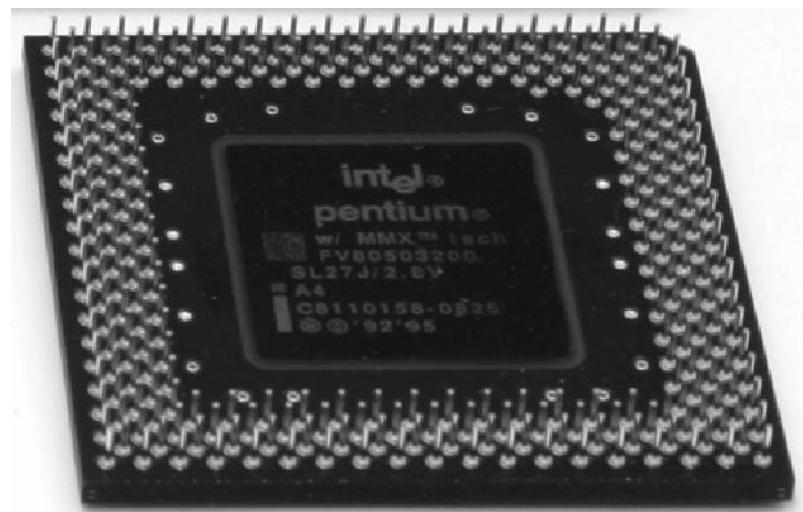
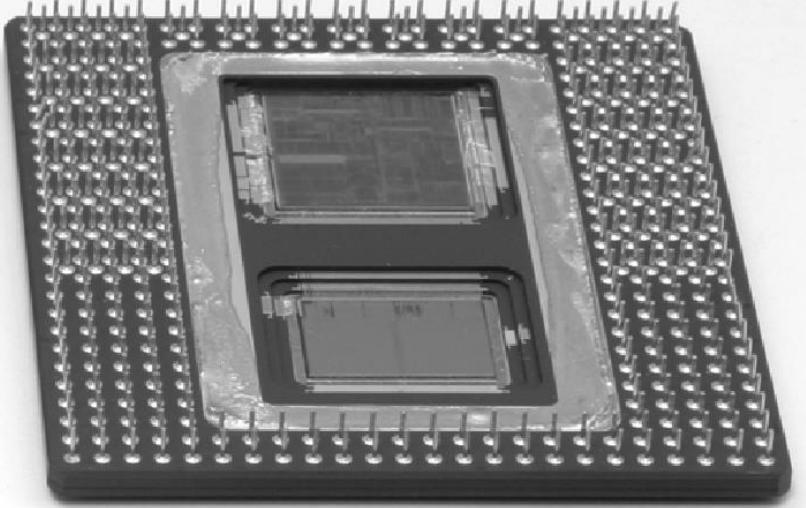


Package Types (1/4)

■ Pin grid array (PGA) package

- ◆ arrange pins at the bottom of the package and can provide 400 or more pins

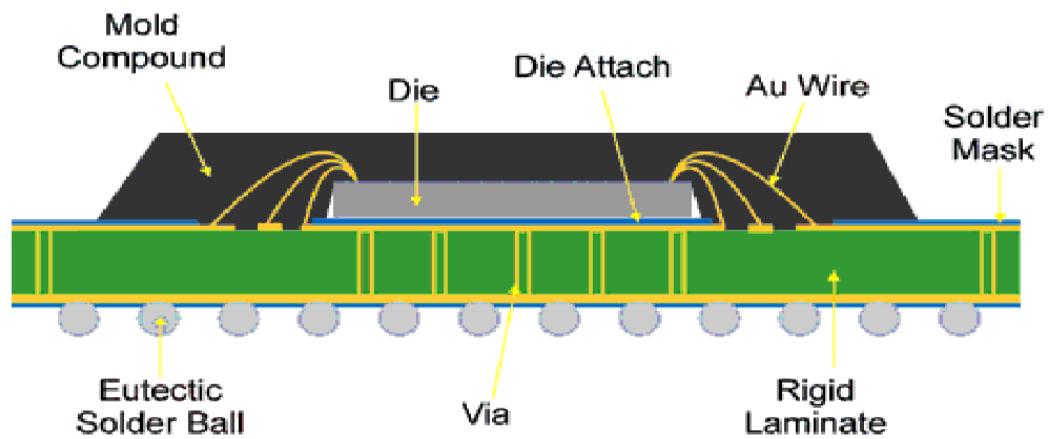
387-pin PGA Multichip Module



Package Types (2/4)

Ball grid array (BGA) package

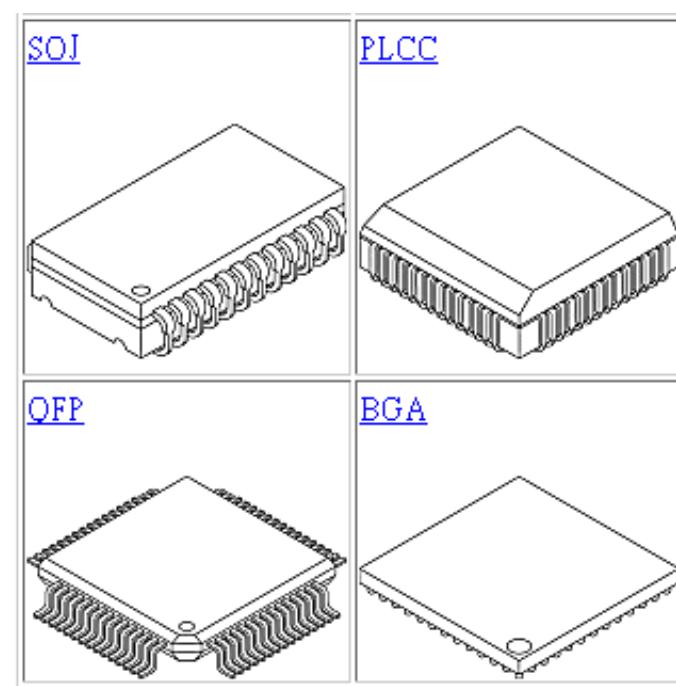
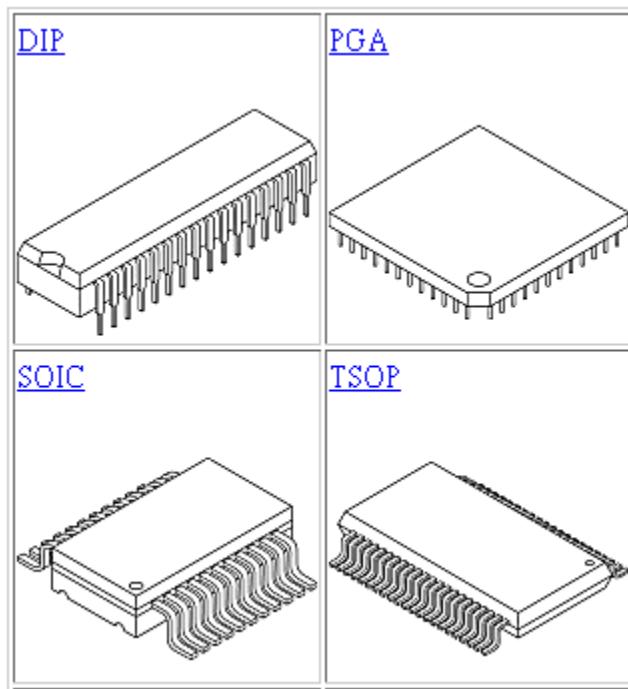
- has a similar array with PGA but has replaced the pins with solder bumps (balls)
- routable laminate substrate (many Layers)
- high pin count (over 2000 pins)
- ASICs, DSPs, PC Chipsets



Package Types (3/4)

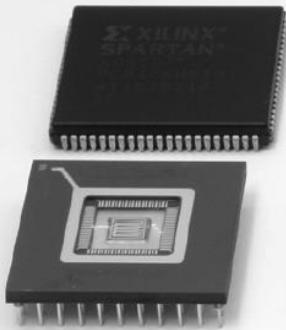
■ Another Important Issue with Packaging

- ◆ Its capability of **heat dissipation**
- ◆ Manufacturer's specification should be carefully evaluated before a package is selected



Package Types (4/4)

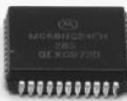
84-pin PLCC



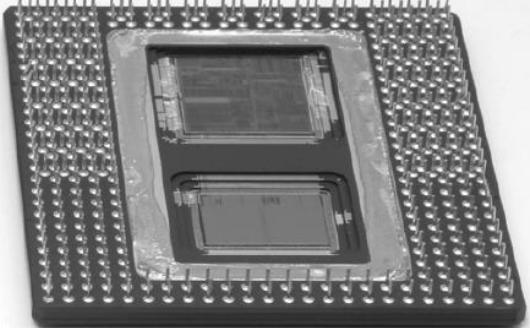
14-pin DIP



44-pin PLCC



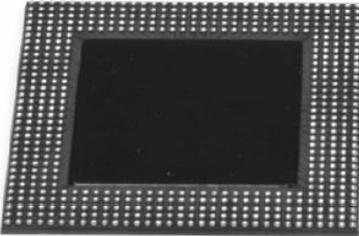
387-pin PGA Multichip Module



84-pin PGA



280-pin QFP



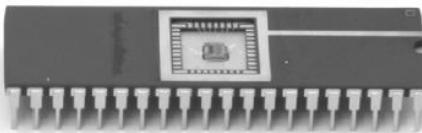
86-pin TSOP



296-pin PGA



40-pin DIP



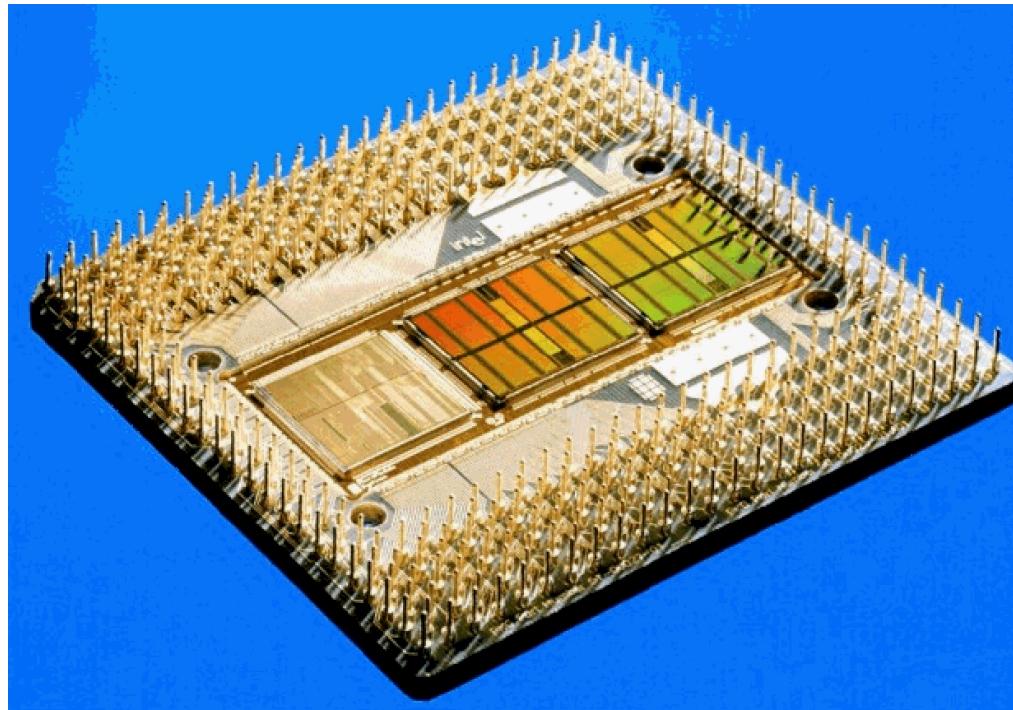
560-pin BGA



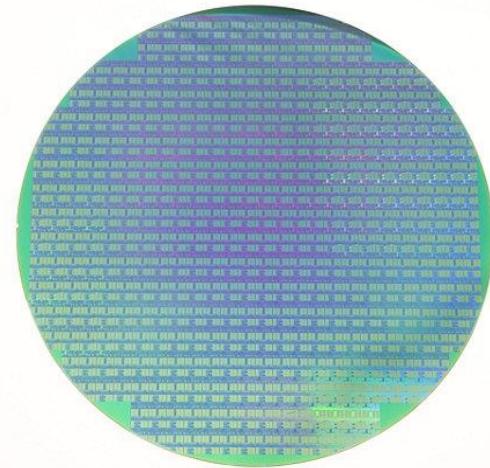
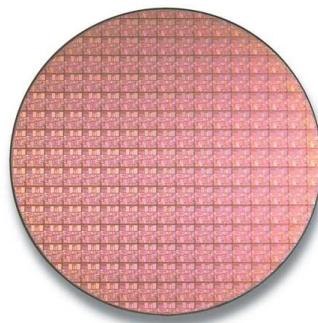
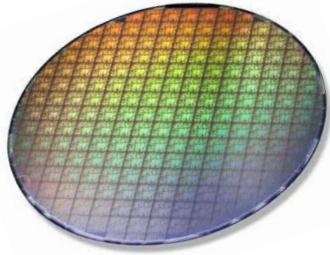
Multichip Modules

Pentium Pro MCM

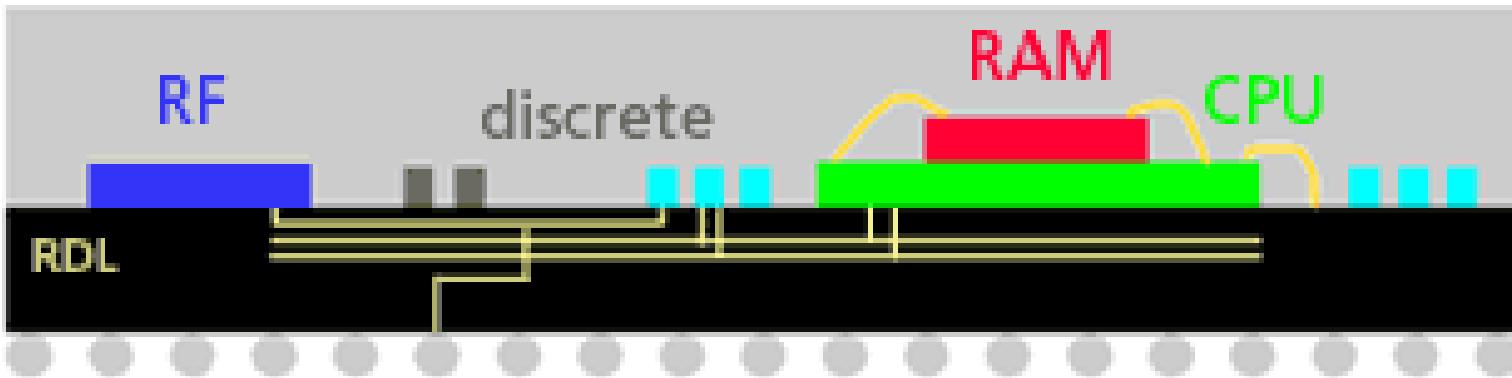
- ◆ Fast connection of **CPU** to one or two external **cache** dices
- ◆ Expensive, requires known good dice (tested bare unpackaged ICs)



System in Package (SiP), System on Package (SoP)

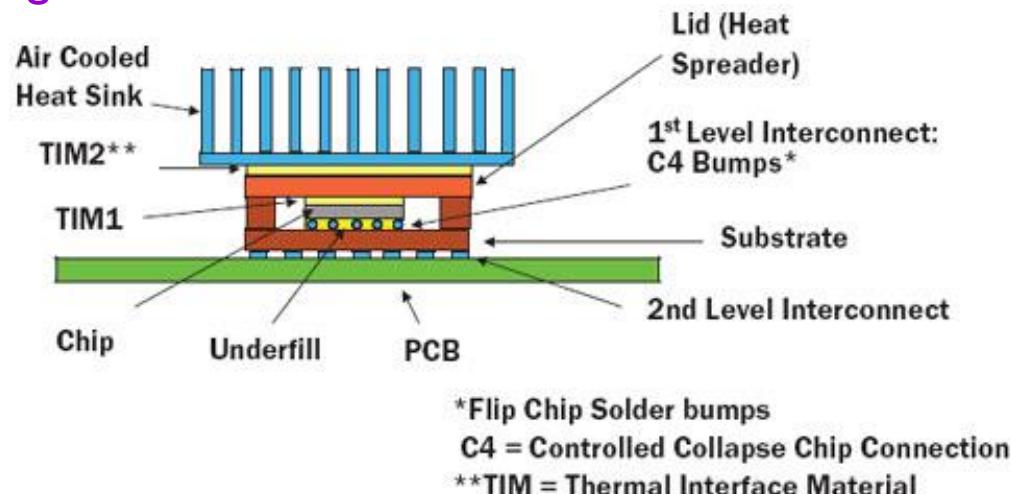


SiP System in Package

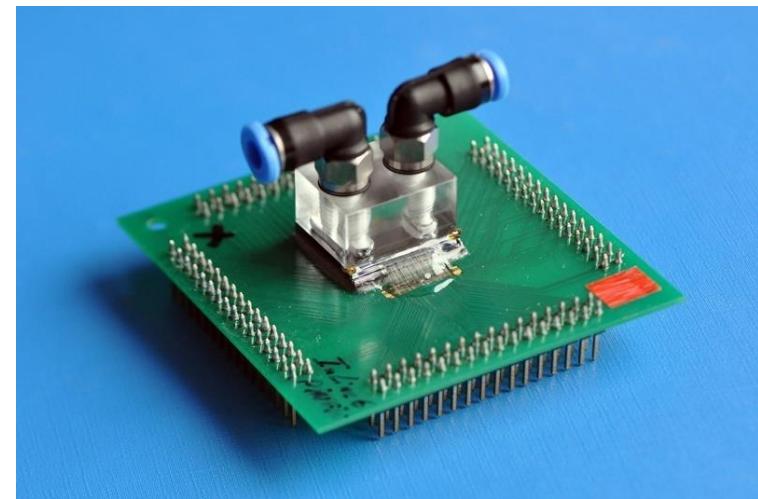
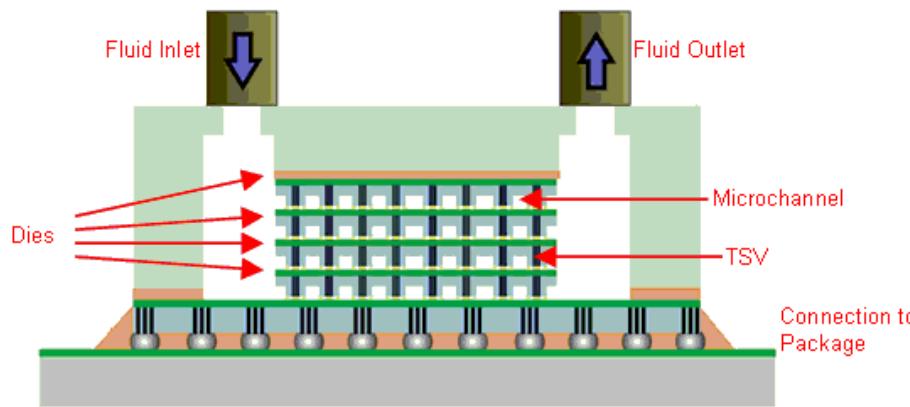
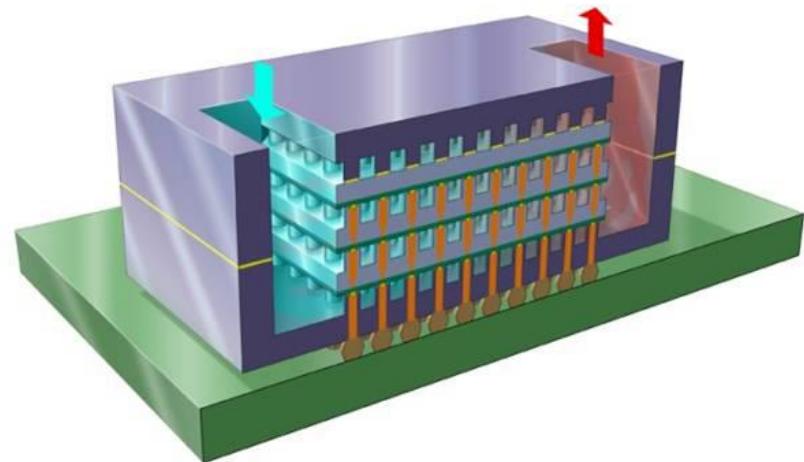
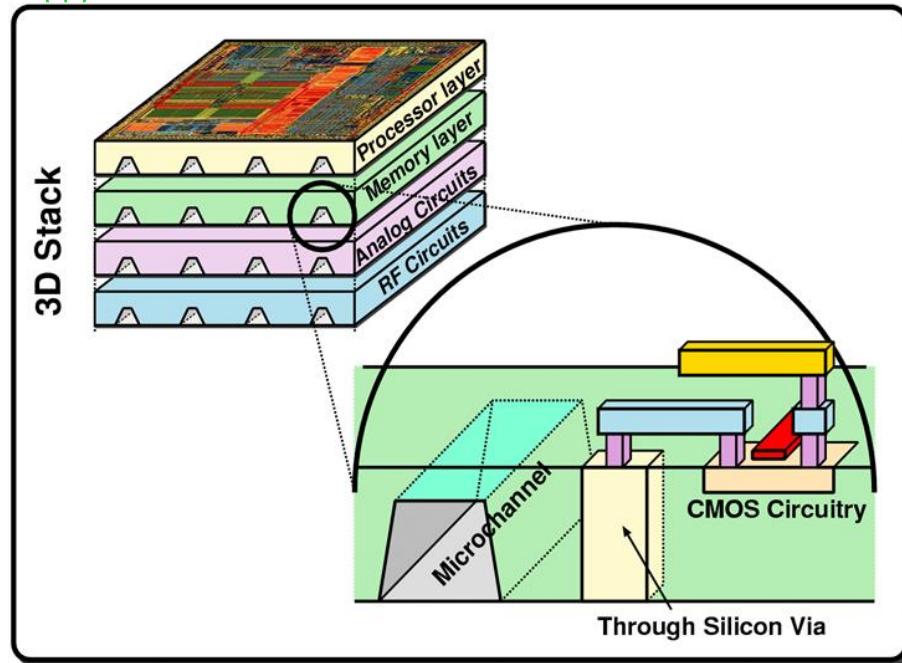


Heat Dissipation

- 60 W light bulb has surface area of 120 cm²
- Itanium 2 die dissipates 130 W over 4 cm²
 - ◆ Chips have enormous power densities
 - ◆ Cooling is a serious challenge
- Package spreads heat to larger surface area
 - ◆ Heat sinks may increase surface area further
 - ◆ Fans increase airflow rate over surface area
 - ◆ Liquid cooling used in extreme cases



3D Stack



I/O Pad Design

■ Output Pads

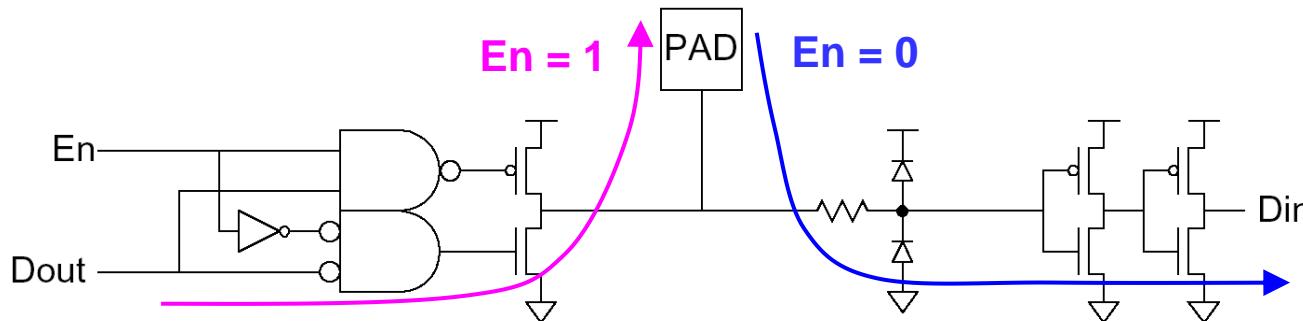
- ◆ Drive large off-chip loads ($2 \sim 50 \text{ pF}$)
- ◆ Guard rings to protect against latchup

■ Input Pads

- ◆ Level conversion, Noise filtering
- ◆ Protection against electrostatic discharge

■ Bidirectional Pads

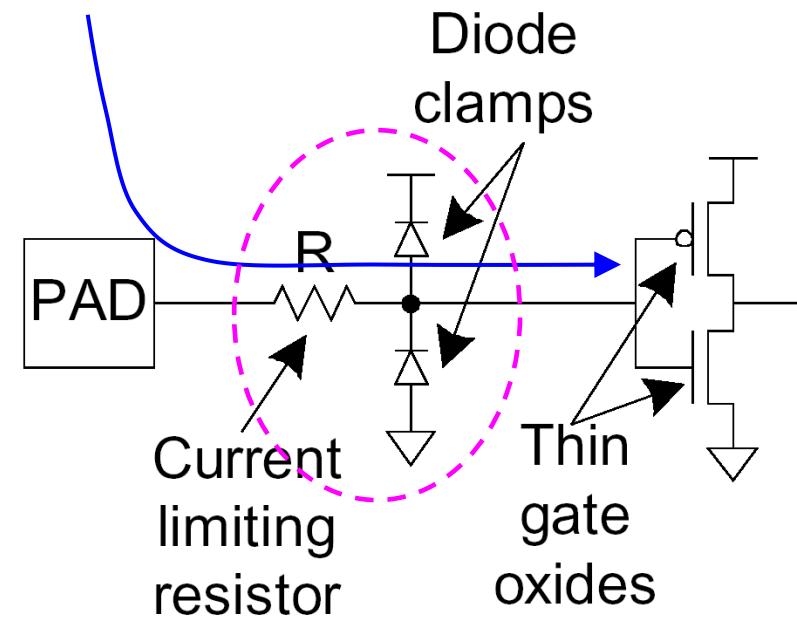
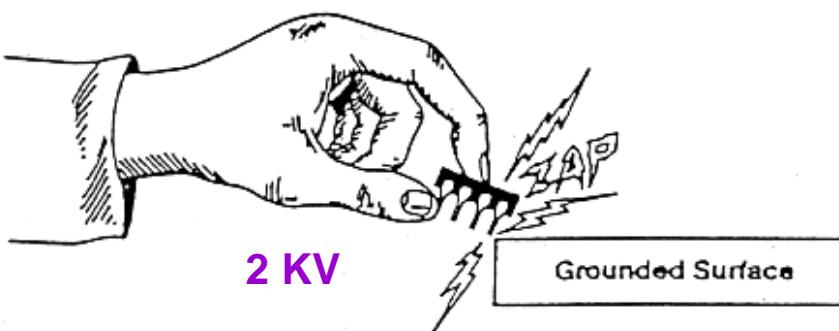
- ◆ Combine input and output pad
- ◆ Need tristate driver on output



ESD Protection

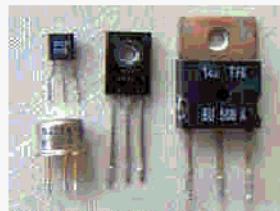


- Static electricity builds up on your body
 - ◆ Shock delivered to a chip can fry thin gates
 - ◆ Must dissipate this energy in protection circuits before it reaches the gates
- ESD (Electrostatic Discharge) protection circuits
 - ◆ Current limiting resistor
 - ◆ Diode clamps



3D IC

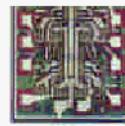
Solid State Discrete Transistors



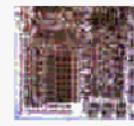
1950s



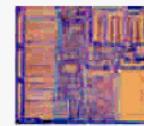
1
Transistor



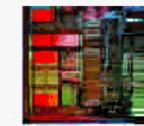
16
Transistors



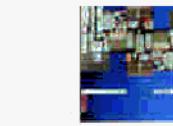
4500
Transistors



275,000
Transistors

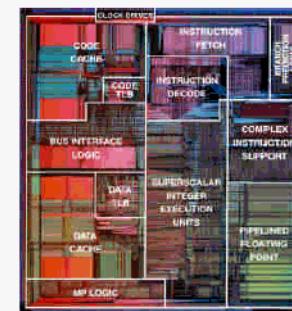


3,100,000
Transistors

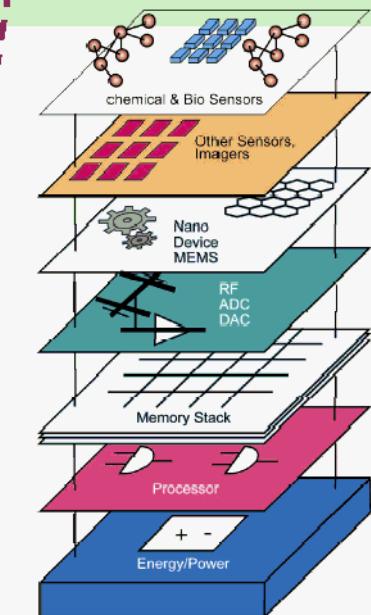


592,000,000
Transistors

Planar (2-D) Integrated Circuits



3-D Integrated Circuits



Source : STC 重製

Heterogeneous Integration

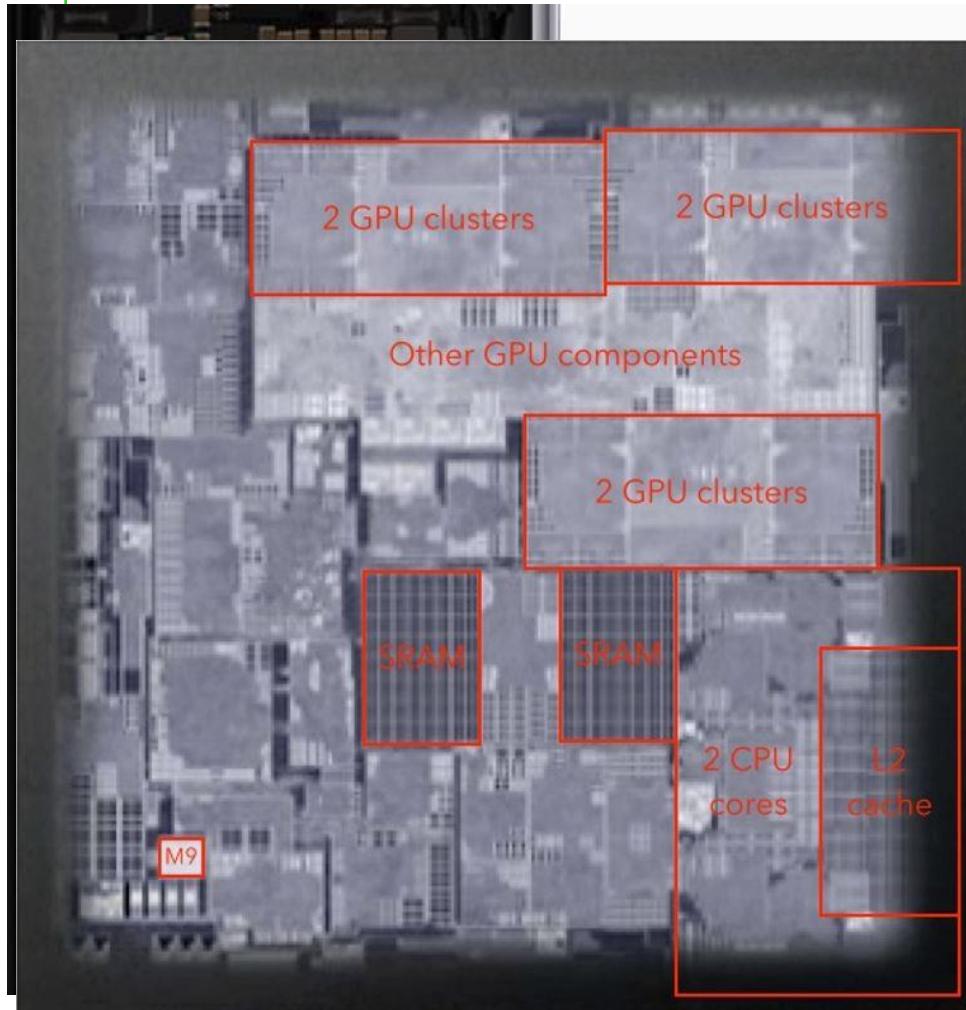
2000s

2010s

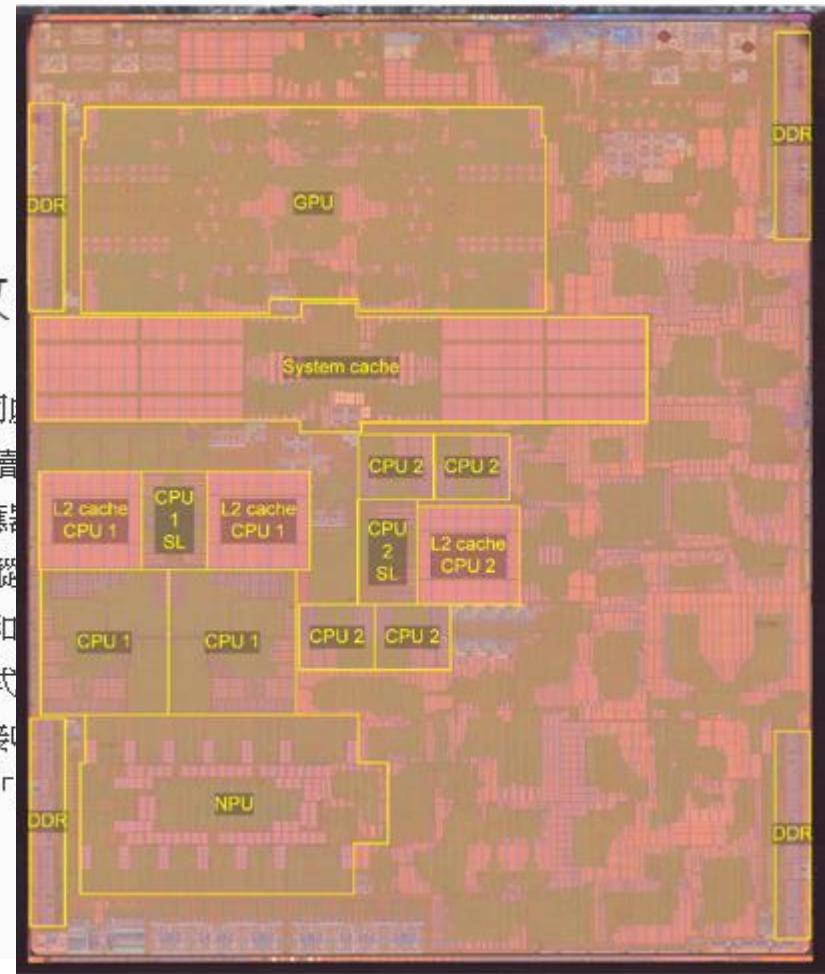
2020s

>>1B
transistors

Apple A9 & A14 SoC

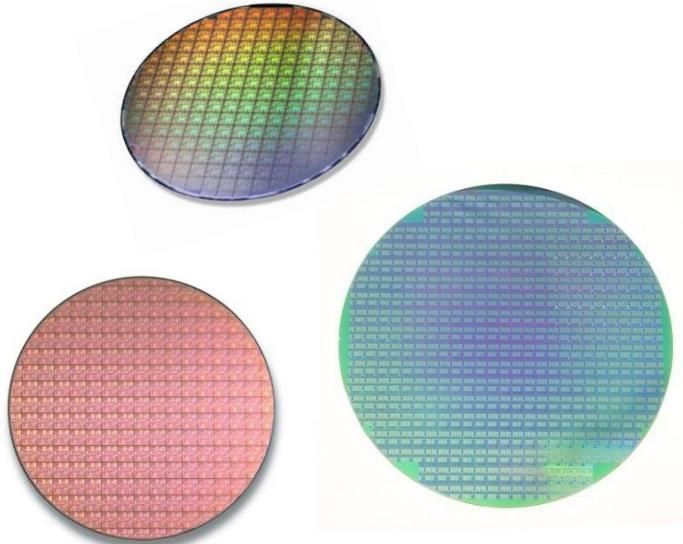
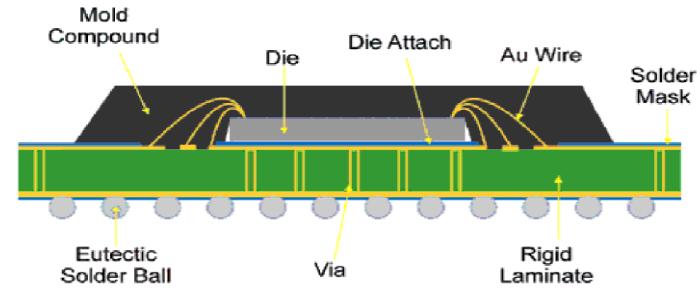
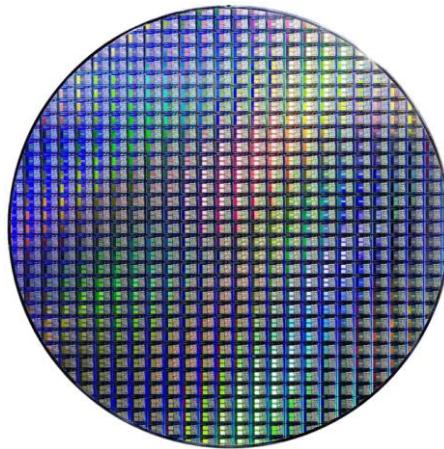
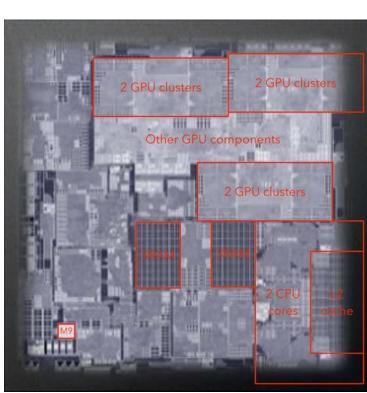


A9 (iPhone 6S, TSMC 16nm)

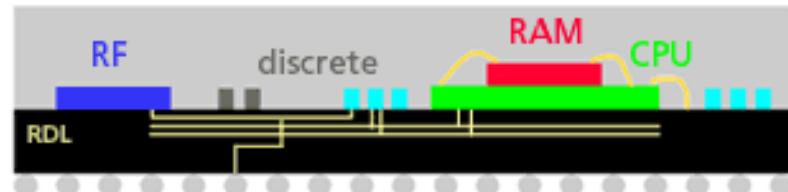


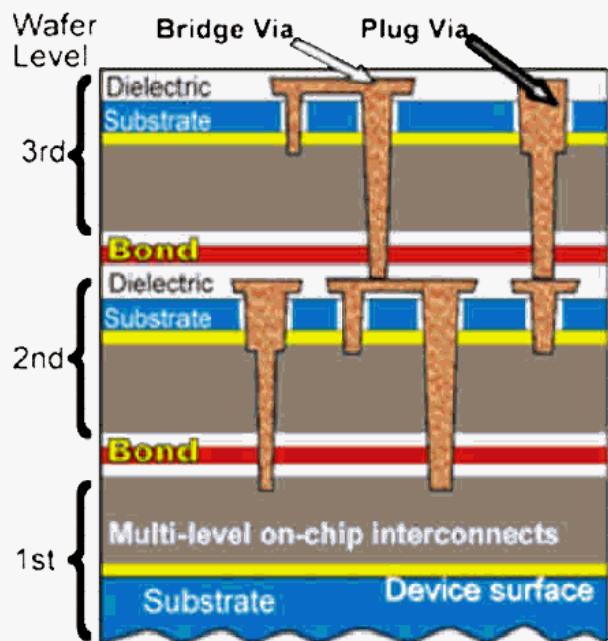
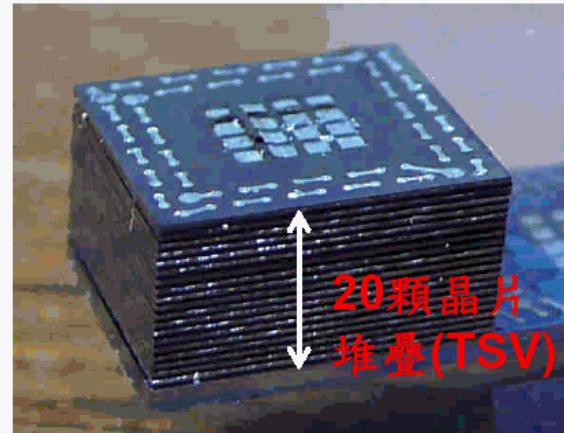
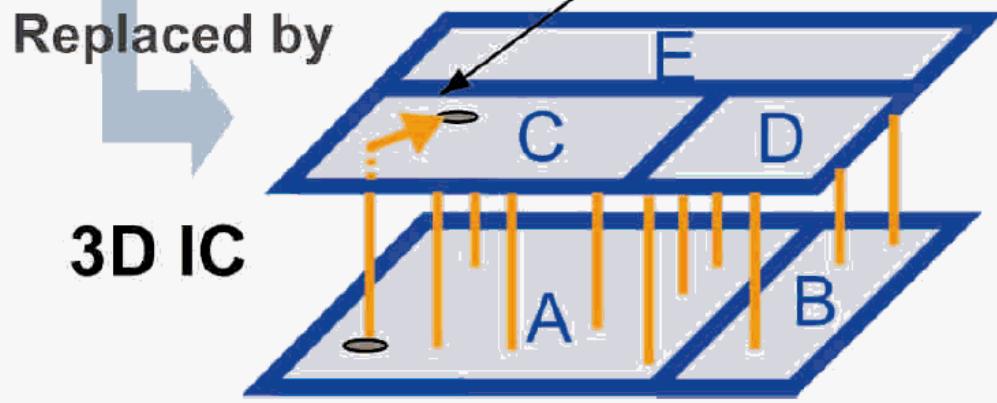
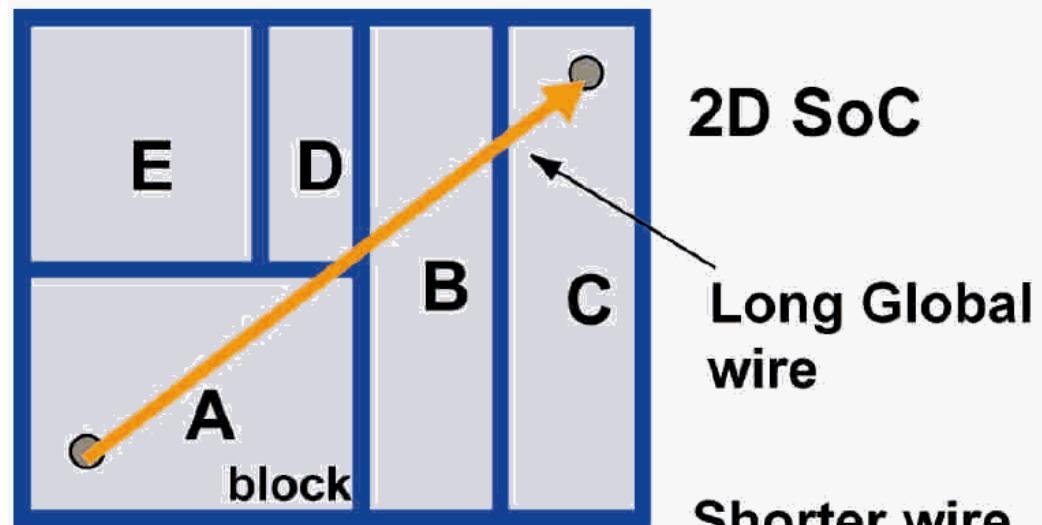
A14 (iPhone 12, TSMC 5nm)

SOC & SiP



SiP System in Package



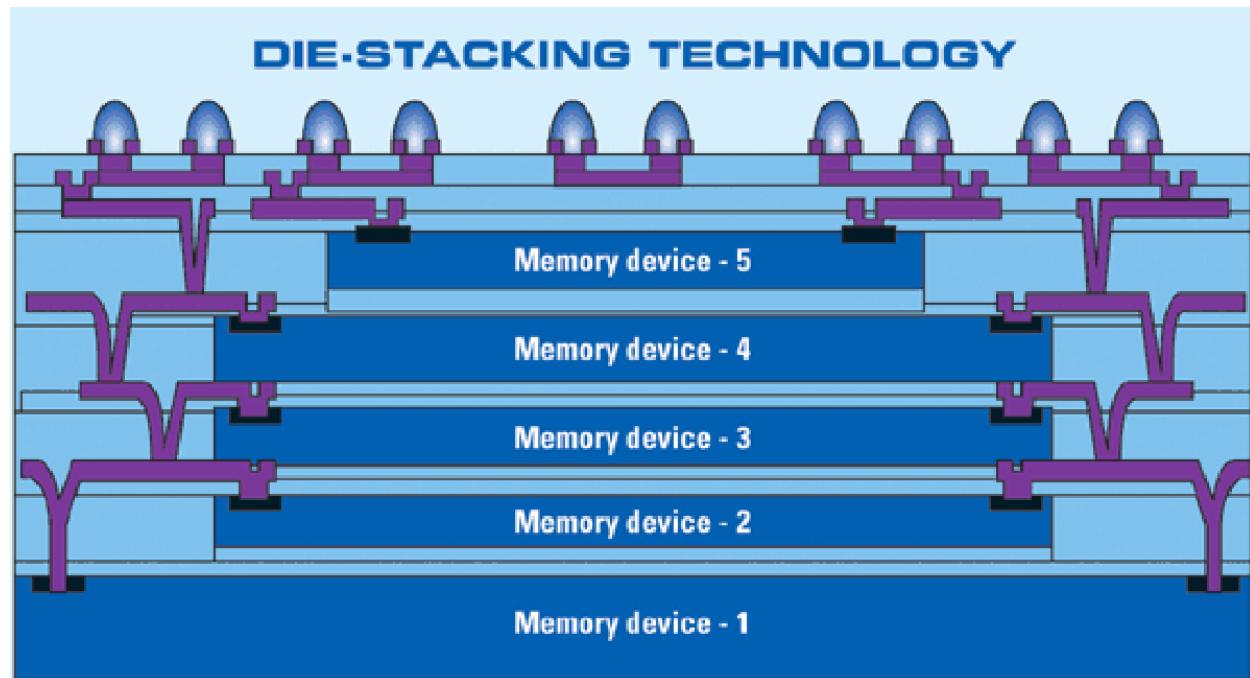




- Improve Interconnect Density
 - ◆ Reduce Form Factor
 - ◆ Reduce Parasitic Capacitance/Inductance
 - ◆ Increase Speed
 - ◆ Reduce Power Consumption
- Reduce Cost
- Provide Heterogeneous Integration
- Improve Reliability
- Reduce ESD Requirement
- Increase Yield
- Improve Data Security
- Scalable/Reconfigurable/Replaceable

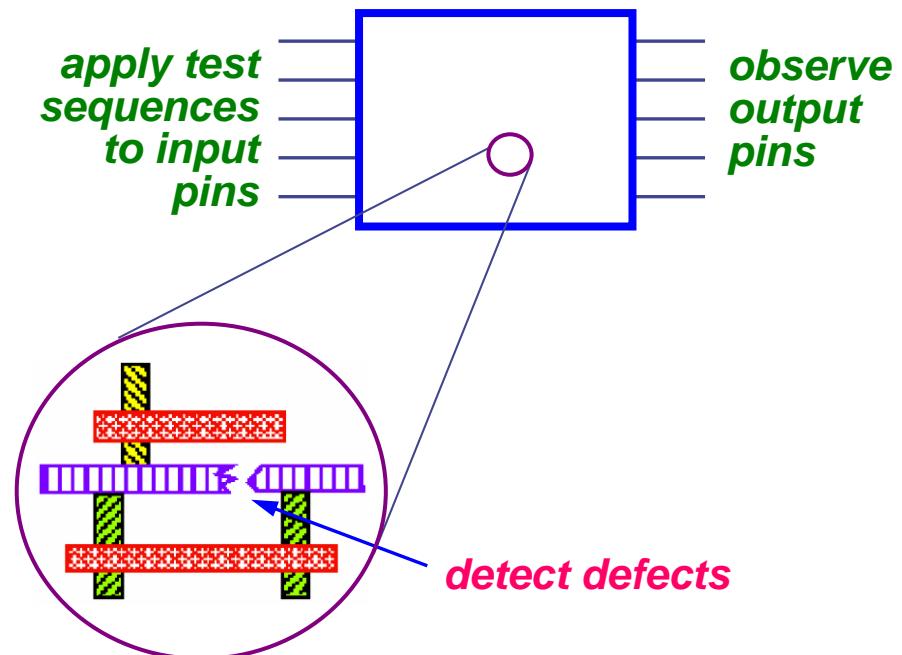
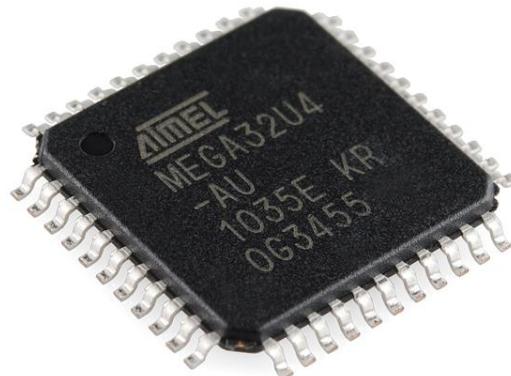
Major Applications of 3D IC

- CMOS Image Sensors
- Processor
- DSP (Sensor and DSP)
- Memories
 - ◆ SRAM/DRAM
 - ◆ NAND
- FPGA
- MEMS



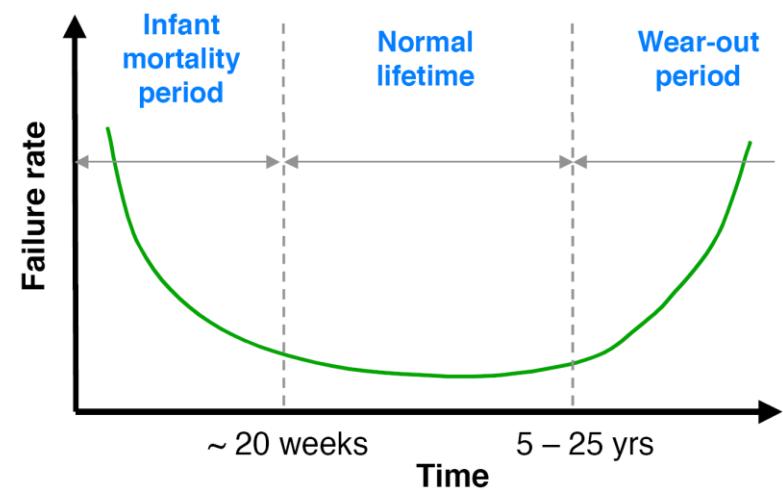
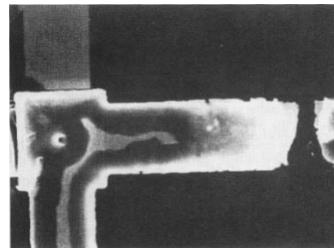
Testing (1/3)

- Two principal causes of failure
 - ◆ **design errors** and **manufacturing defects**
- Design errors
 - ◆ violations of design rules
 - ◆ incorrect mapping between different levels of design
 - ◆ incomplete or inconsistent specification

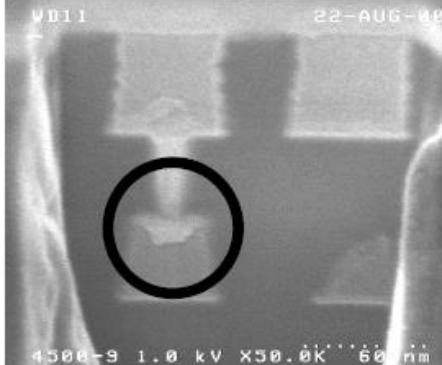


Faulty Chip

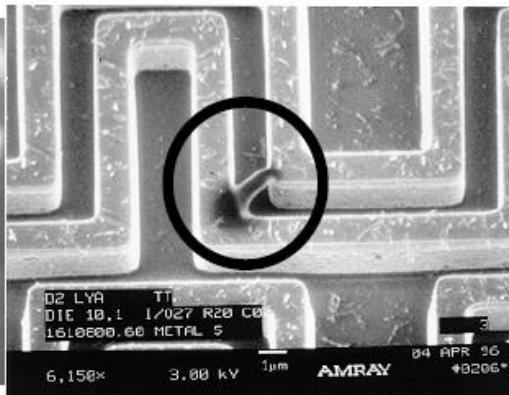
- Fabrication errors (caused by human errors)
 - ◆ Wrong components
 - ◆ Incorrect wiring
 - ◆ Shorts caused by improper soldering
- Fabrication defects (caused by imperfect manufacturing process)
 - ◆ Mask alignment errors
 - ◆ Improper doping profiles
- Physical failures
 - ◆ Electro-migration or corrosion
 - ◆ Aging/Wear-out of components



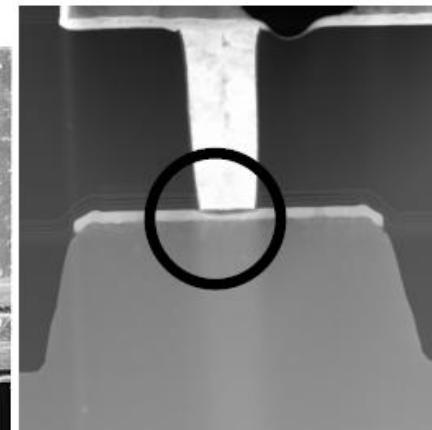
Manufacturing Failures



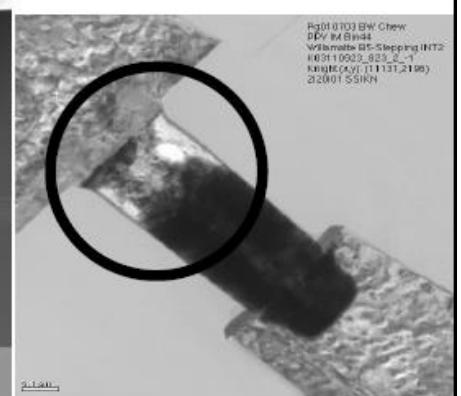
Metal 1 Shelving



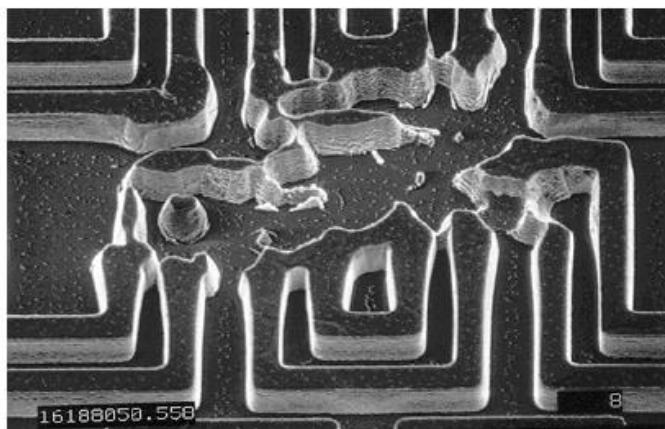
Metal 5 film particle
(bridging defect)



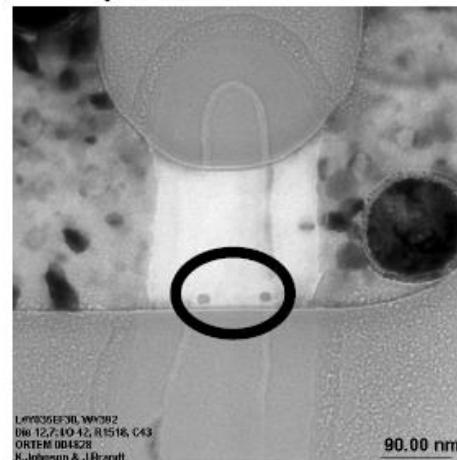
Open defect



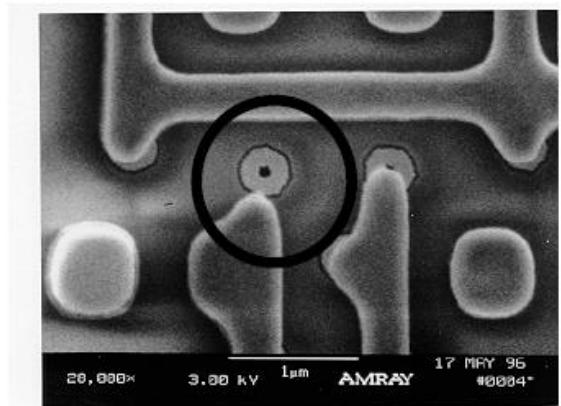
Spongy Via2
(Infant mortality)



Metal 5 blocked etch
(patterning defect)

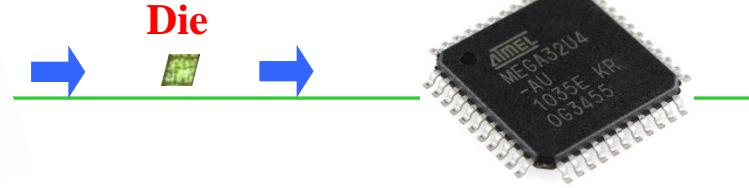
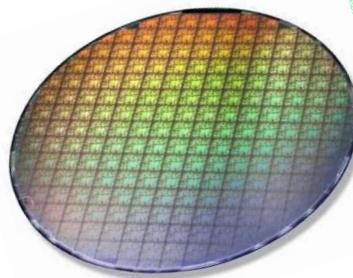


Spot defects
“Co” Defect under Gate



Metal 1 missing pattern
(open at contact)

Testing (2/3)



Wafer test

- is applied to visually check, identify, and mark **faulty chips**
- the wafer is then **sawed into individual circuit chips** and the marked faulty chips are discarded
- each **good chip is mounted** in an IC package

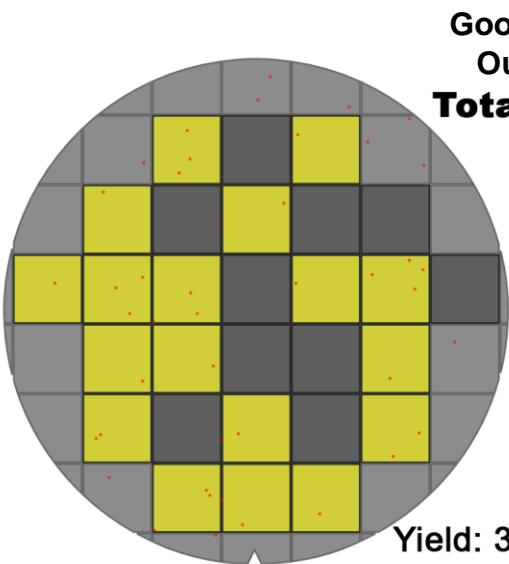
Functional test

- **packaged ICs** then undergo a **functional test** to separate acceptable parts from failed ones
- in the **fabrication of prototype, analysis** must be performed to determine the causes of detected failures
- after **design errors** have been identified and eliminated, the **chip** goes into production

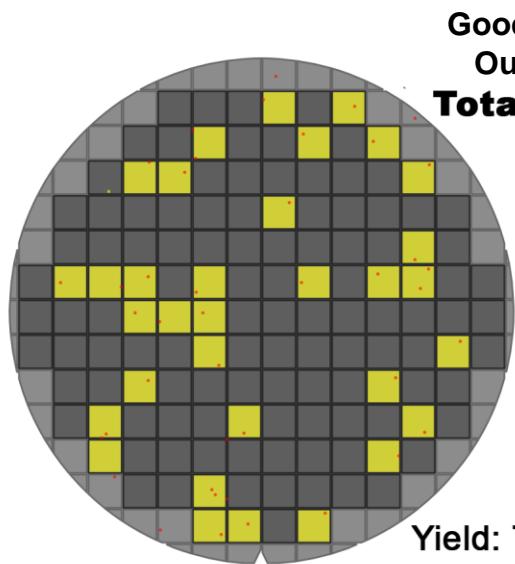
Yield

- during IC production, a certain number of chips can be expected to fail the acceptance test and be discarded along the way
- yield: the percentage of acceptable parts obtained from a wafer

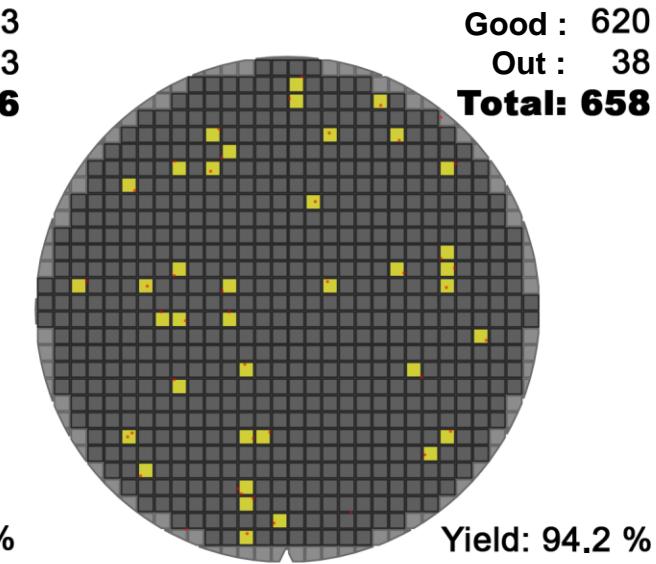
Yield



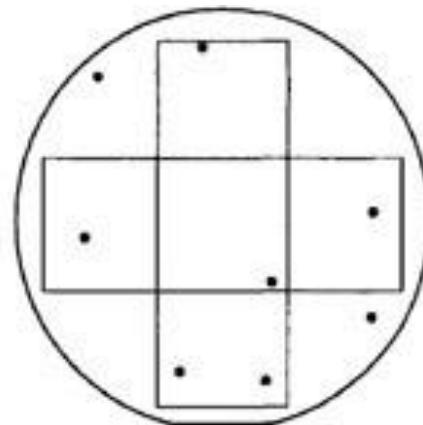
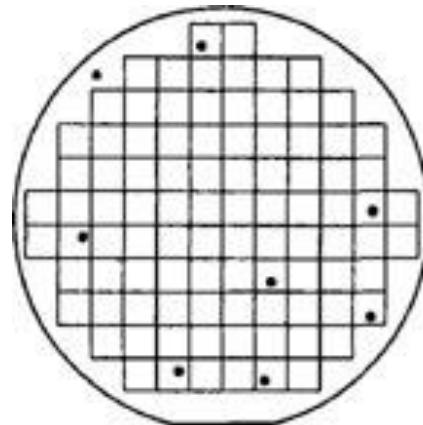
Die size: 40 mm x 40 mm



Die size: 20 mm x 20 mm



Die size: 10 mm x 10 mm



Testing (3/3)



Controllability and Observability

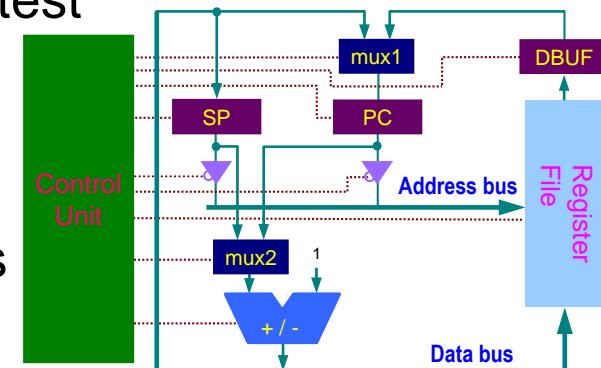
- functional test of a circuit requires its internal nodes to be accessible so that test signals can be injected (controllability), and responses at the internal nodes can be monitored (observability)

Design-for-testability (DFT)

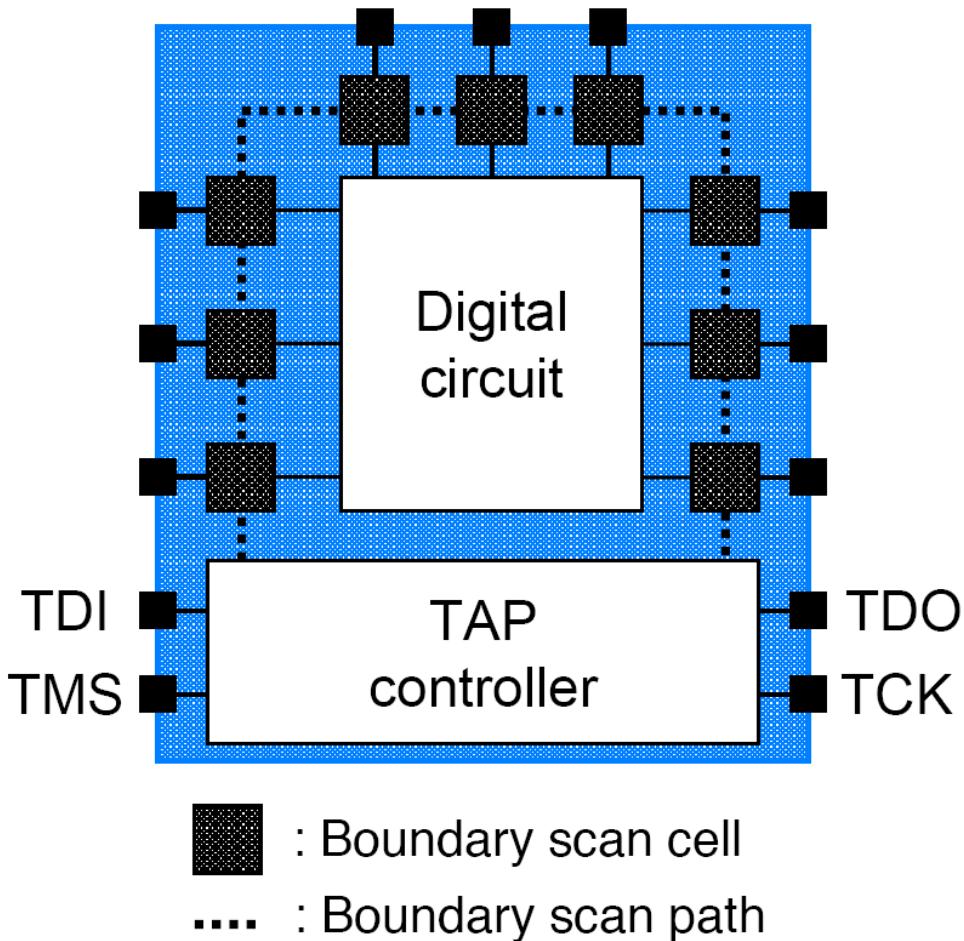
- the need for testing has to be taken into account throughout the design process to improve the testability of circuits
- boundary scan
 - build circuitry into an IC to assist in the test

Complexity of testing

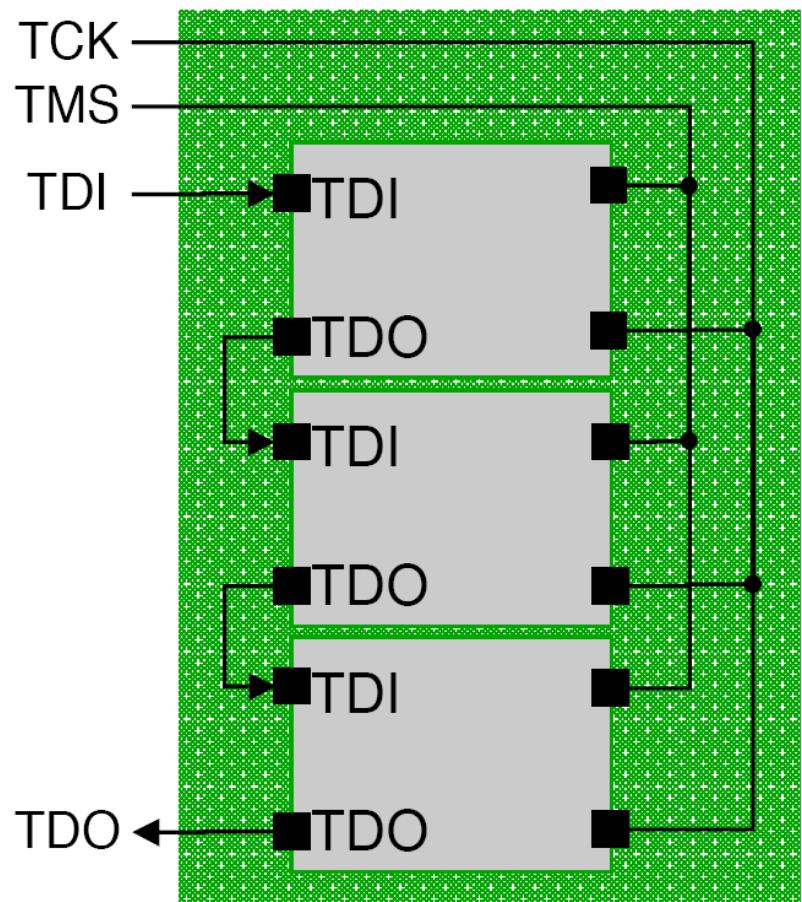
- exhaustive testing
 - n inputs, 2^n possible input combinations
- test scheme
 - identify the smallest set of test vectors necessary to verify the circuit functions



Boundary Scan (1/2)



an IEEE 1149.1 compliant chip



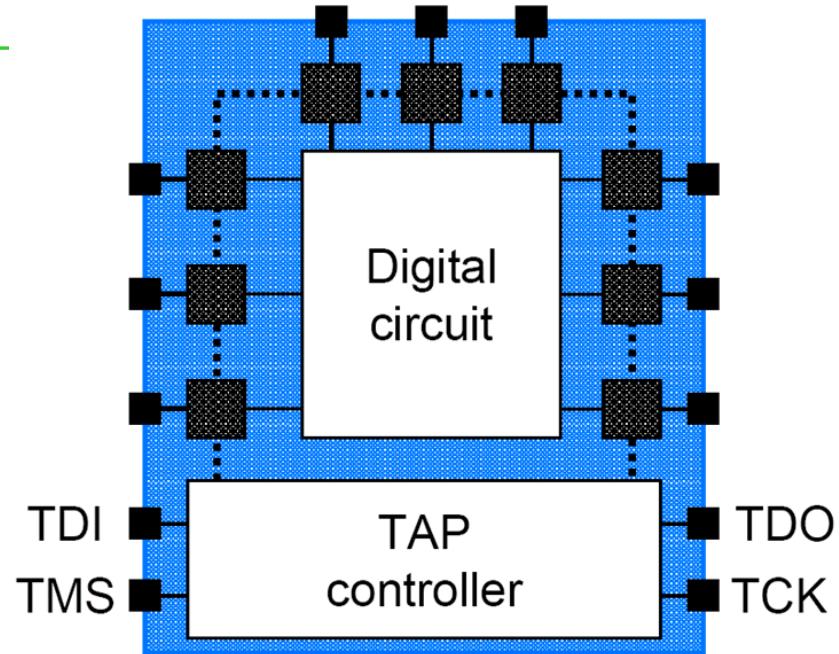
1149.1 board-level view

Boundary Scan (2/2)



Boundary scan

- ◆ IEEE 1149.1
- ◆ TAP (test access port)
- ◆ TDI (test data input)
- ◆ TDO (test data output)
- ◆ TMS (test mode select)
- ◆ TCK (test mode clock)



Advantages

- ◆ existing well-known, well-documented standard
- ◆ reuse IC level implementation

Drawbacks

- ◆ fixed 1-bit TAM (test access mechanism) width
- ◆ complexity of test control and test data wiring grows with the number of cores

Built-in Self-test (BIST)



- BIST is an effective technique for digital logic and memory circuits
- Digital circuits, tests are generated by a pseudorandom vector generation circuit
- BIST are the economy of external test data, reduced test time through parallel test execution, and high-speed testing
- The analog BIST methods lack generality and could be expensive in hardware overhead

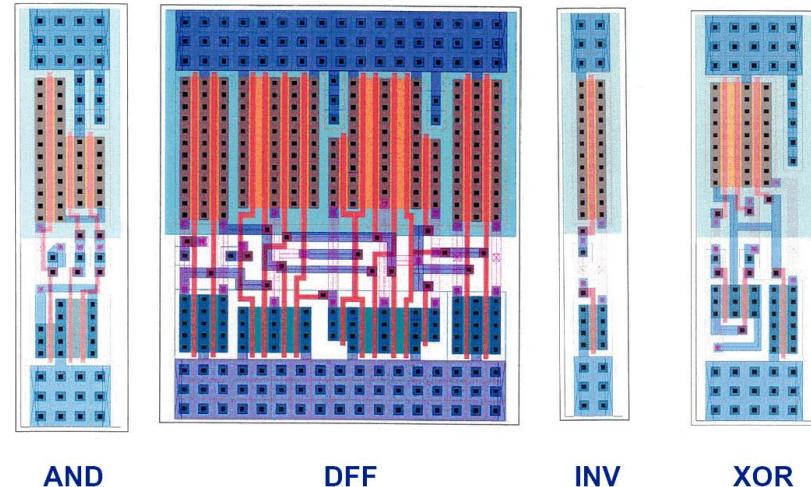
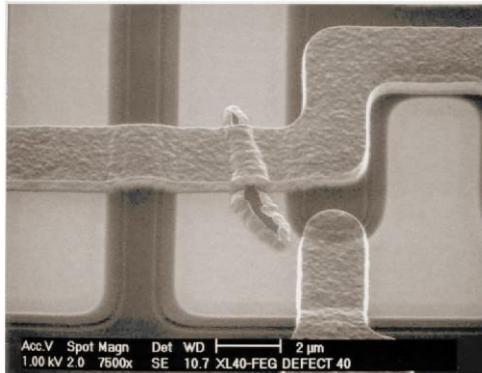
Stuck-At Faults

■ How does a chip fail?

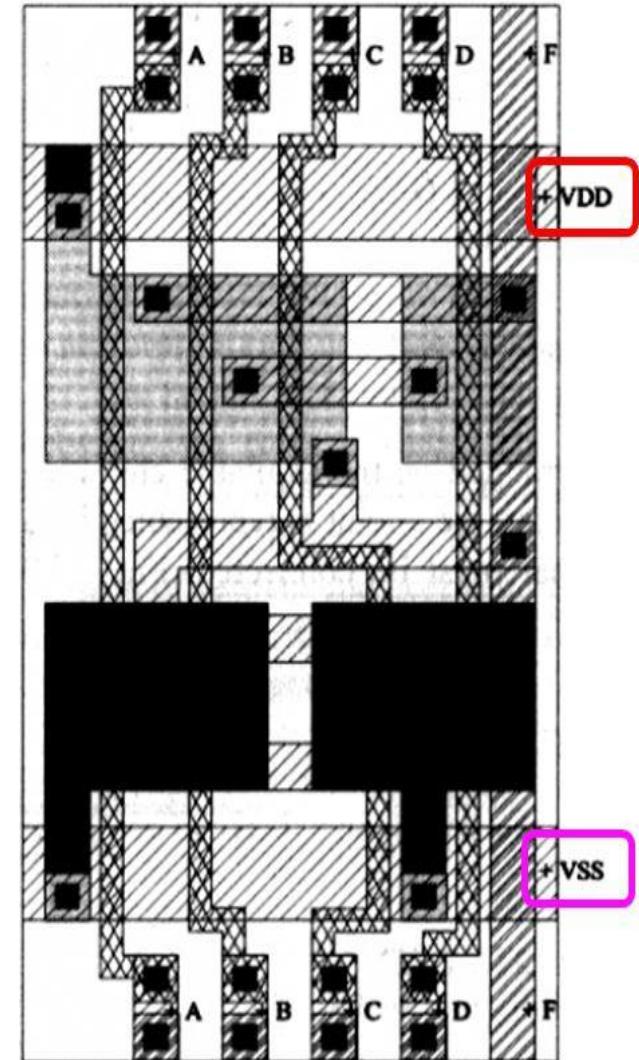
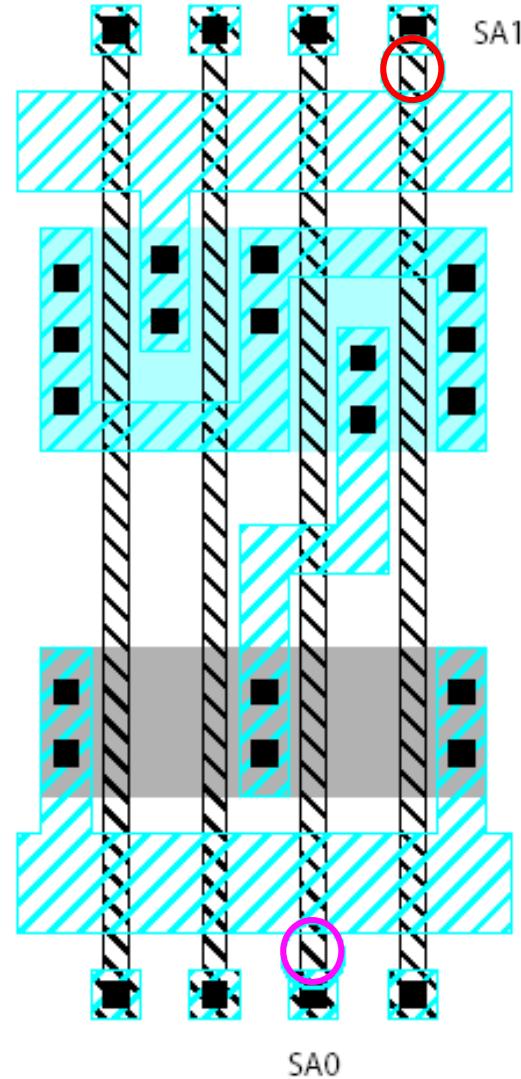
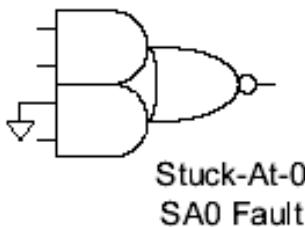
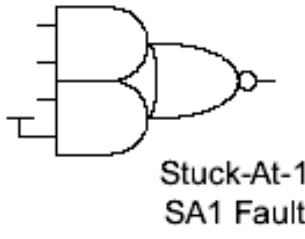
- ◆ Usually failures are **shorts** between two conductors or **opens** in a conductor
- ◆ This can cause very complicated behavior

■ A simpler model: *Stuck-At*

- ◆ Assume all failures cause nodes to be “stuck-at” 0 or 1, i.e. shorted to GND or V_{DD}
- ◆ Not quite true, but works well in practice



Examples



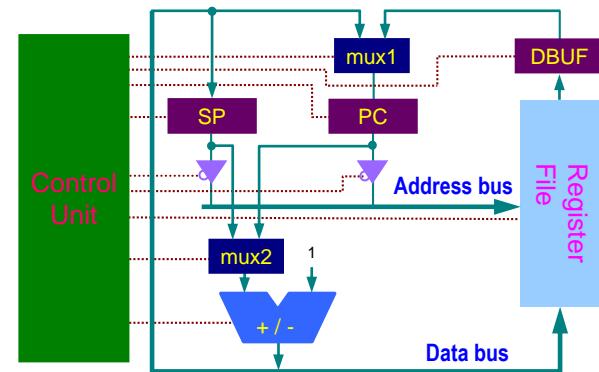
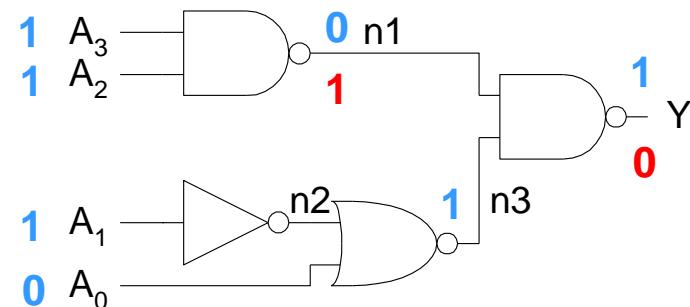
Test Pattern Generation

- Manufacturing test ideally would check every node in the circuit to prove it is not stuck
- Apply the smallest sequence of test vectors necessary to prove each node is not stuck

Test Example



	SA1	SA0
A ₃	{0110}	{1110}
A ₂	{1010}	{1110}
A ₁	{0100}	{0110}
A ₀	{0110}	{0111}
n1	{1110}	{0110}
n2	{0110}	{0100}
n3	{0101}	{0110}
Y	{0110}	{1110}



- Minimum set: {0100, 0101, 0110, 0111, 1010, 1110}

Automatic Test Pattern Generation (ATPG)

- Test pattern generation is tedious
- Automatic Test Pattern Generation (ATPG) tools produce a good set of vectors for each block of **combinational logic**
- **Scan chains** are used to control and observe the blocks
- Complete coverage requires a large number of vectors, raising the cost of test
- Most products settle for covering 90+% of potential stuck-at faults

Summary

Basic Design Flow

