Topic II

SoC Design Overview

系統晶片驗證
SoC Verification

Sep, 2004

SoC = System on a Chip

What is a System?

- A *<u>self-contained</u>* functional unit that can complete a task or serve some people in certain applications
  - Telecommunications, networking
    - ATM switches, Ethernet switches, bridges, routers
  - Portable consumer products
    - Cell phones, organizers
  - Multimedia
    - Digital cameras, game consoles, digital video
  - Embedded control
    - Automotive, printers, smart cards, disk drives

A system should (may) contain…

- I/O interface

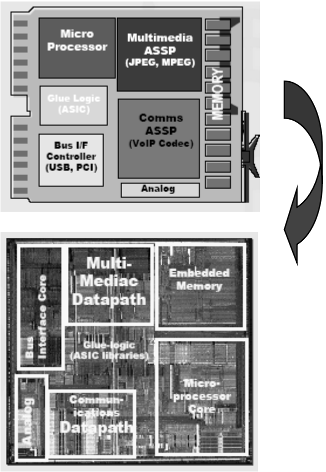
  (Bus, RF, display driver, etc)
- Processor

  (General purposed, DSP, graphic, etc)
- Analog/Mixed signal units

  (ADC, DAC, PLL, etc)
- Network controller
- Embedded memory (RAM)
- Embedded software (ROM)
- Built-in self test (BIST)

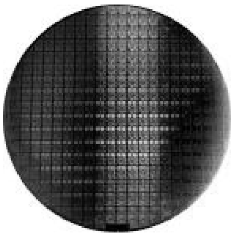
System Migration

- ◆System in a box
- ◆System on a board
- ◆System on a chip



Why System-on-a-Chip?

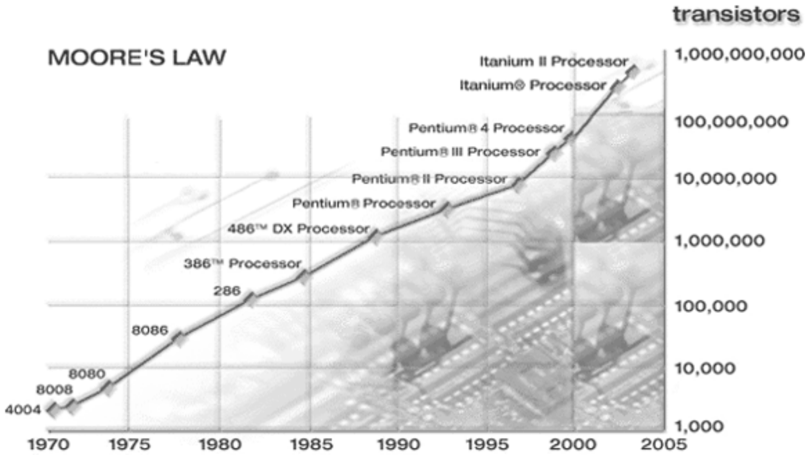
- ◆Cost reduction
  - ● More products per wafer
- ◆Higher performance
  - ● External vs internal bus
- ◆Lower power consumption
  - ● Portable applications (battery life)
- ◆Shorter market cycle
  - ● Design (core) reuse



Sounds great,

but the reality nowadays is…

Famous Moore’s Law



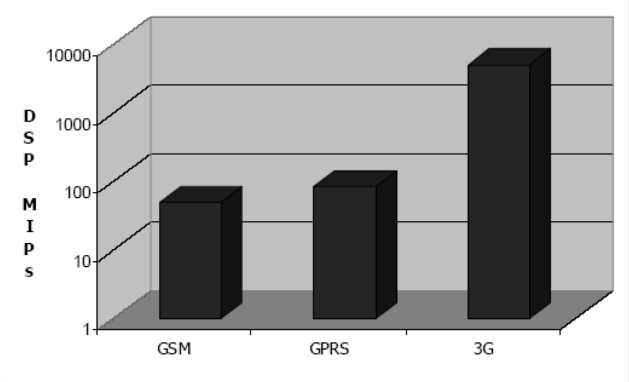
- ◆ #Transistors: double/2 years → double/1.5 years

Source: Intel Corp.

#Transistors in Intel's CPUs

	Year of Introduction	Transistors
4004	1971	2,250
8008	1972	2,500
8080	1974	5,000
8086	1978	29,000
286	1982	120,000
386™ processor	1985	275,000
486™ DX processor	1989	1,180,000
Pentium® processor	1993	3,100,000
Pentium II processor	1997	7,500,000
Pentium III processor	1999	24,000,000
Pentium 4 processor	2000	42,000,000
Itanium® processor	2002	220,000,000
Itanium II processor	2003	410,000,000

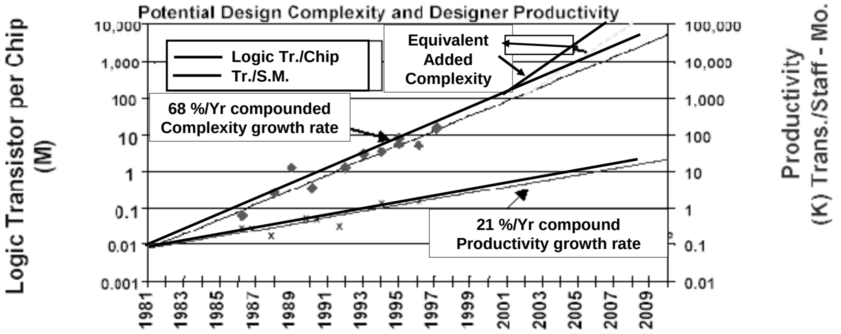
Processing Power Increase



Example: 3G Phones require 100x increase in processing power over 2G

Source: ECIT

Design Productivity Gap

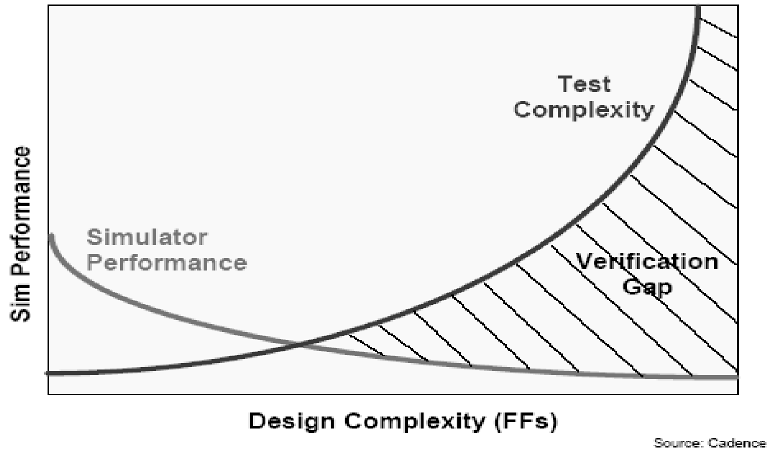


				3 Yr. Design	
Year	**Technology**	**Chip Complexity**	**Frequency**	**Staff**	**Staff Cost***
1997	250 nm	13 M Tr.	400 MHz	210	90 M
1998	250 nm	20 M Tr.	500	270	120 M
1999	180 nm	32 M Tr.	600	360	160 M
2002	130 nm	130 M Tr.	800	800	360 M

* @ $ 150 k / Staff Yr. (In 1997 Dollars)

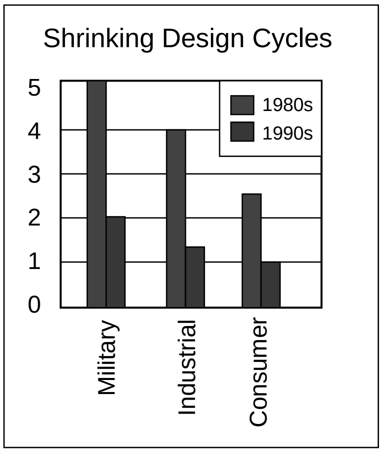
Source: GSRC

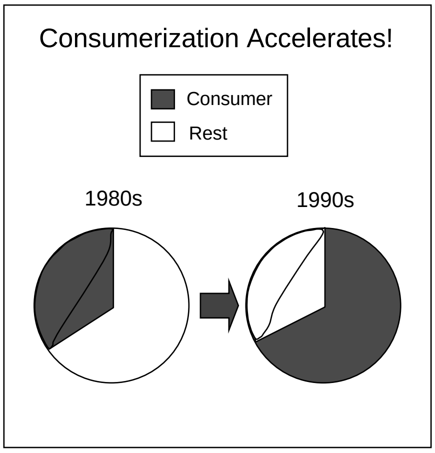
Verification Gap



Source: Cadence

Time to Market Trends

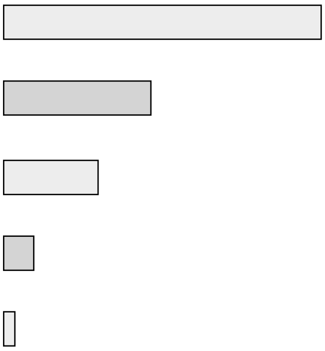

Shrinking Design Cycles


Consumerization Accelerates!

Source: “System-on-a-Chip Verification”, P. Rashinkar, et al

Period from Plan to Mass-Production

- ◆Color TV: 12 Years
- ◆Computer: 5 Years
- ◆Mobile Phone: 3 Years
- ◆PS-1: 10 Months
- ◆PS-2: 3 Months



- ◆Throwing in more engineers? ☹

Satoshi Goto, VLSI/CAD 2004

Change!!

Something gotta be changed…

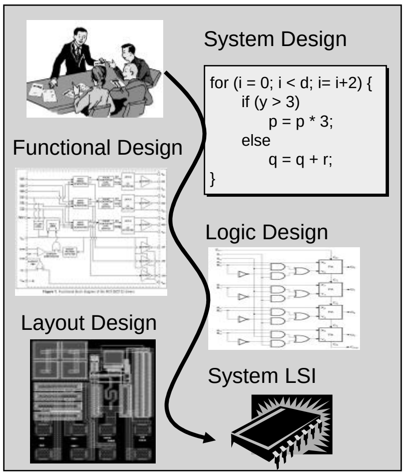
Paradigm Shifts of SoC Design

- ◆What paradigm shifts have happened in SoC?
  - ● Paradigm shift #1 (1980 ~)
  - ● Paradigm shift #2 (1990 ~)
  - ● Paradigm shift #3 (2000 ~)

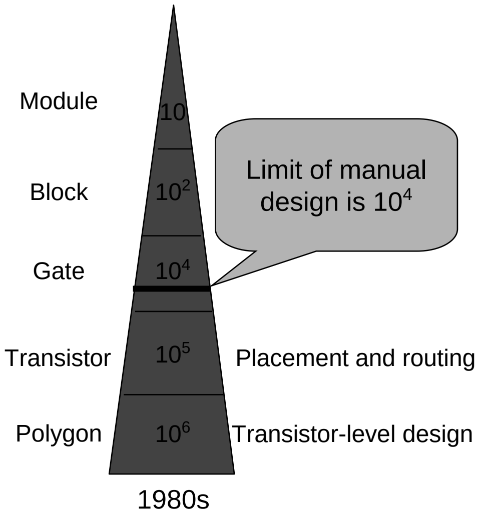
Satoshi Goto, VLSI/CAD 2004

Design Automation and System LSI Size

Design Flow



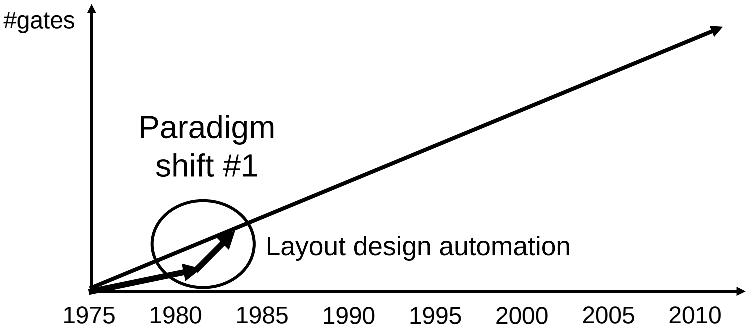
```
for (i = 0; i < d; i= i+2) {
      if (y > 3)
            p = p * 3;
      else
            q = q + r;
}
```



Satoshi Goto, VLSI/CAD 2004

Design Automation Paradigm Shift #1 (1980 ~)

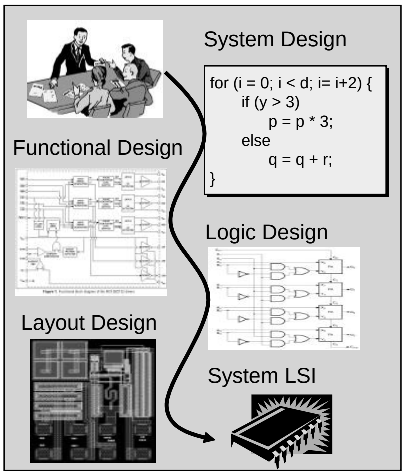
- ◆Layout design automation
  - Placement and routing of gate level circuits on a chip



Satoshi Goto, VLSI/CAD 2004

Design Automation and System LSI Size

Design Flow



System Design

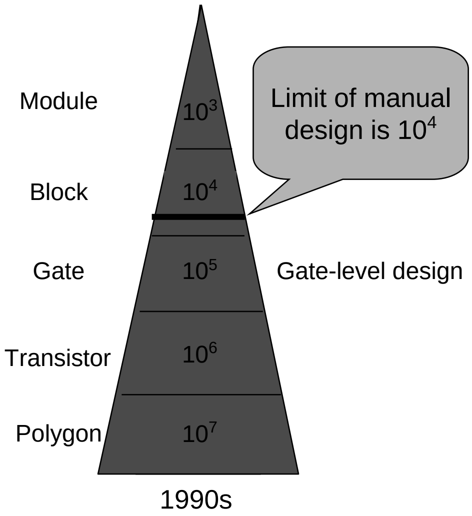
```
for (i = 0; i < d; i= i+2) {
     if (y > 3)
          p = p * 3;
     else
          q = q + r;
}
```

Functional Design

Logic Design

Layout Design

System LSI



Limit of manual design is 10^4

Satoshi Goto, VLSI/CAD 2004

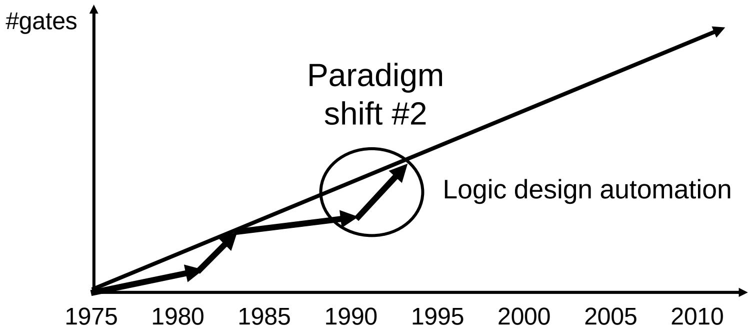
Design Automation Paradigm Shift #2 (1990 ~)

- Logic Synthesis
  - Generate gate level circuits from RT level description by circuit structure optimization
- Formal Logic Verification
  - Verify a circuit validaty by mathematical way, but not by simulation
  - 500 times faster and guarantee 100% validity
- Integration of Logic Synthesis and Layout Design
  - Signal net delay or clock timing is considered at logic synthesis

Satoshi Goto, VLSI/CAD 2004

Design Automation Paradigm Shift #2 (1990 ~)

- ◆ Logic Synthesis
- ◆ Formal Logic Verification
- ◆ Integration of Logic Synthesis and Layout Design



Satoshi Goto, VLSI/CAD 2004

Design Automation Paradigm Shift #3 (2000 ~)

Ultra Large Scale and

Ultra Fine Process System LSI

1. Scalability: Ultra Large Scale Circuits
   ➔ Almost impossible to design within a reasonable time
2. Reliability: High Complex Circuits
   ➔ Almost impossible to design with error free
3. Performance: Signal Integrity
   ➔ Almost impossibility to achieve high TAT

Satoshi Goto, VLSI/CAD 2004

Design Automation Paradigm Shift #3 (2000 ~)

1. Scalability: Ultra Large Scale Circuits
   - Reduce the circuit (design) size
     - High level design: Functional synthesis
     - Hierarchical design: Partitioning & floor planning
   - Design-less
     - Exploitation of design results

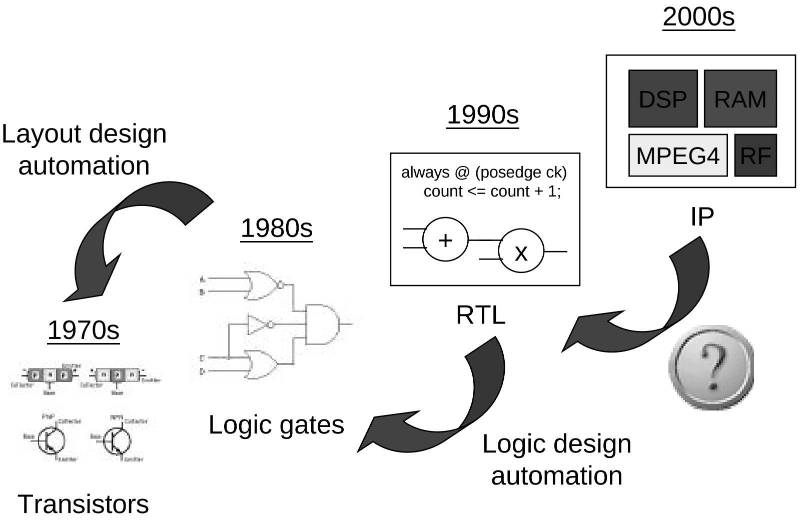
       ➔ IP (Intellectual Property)

Satoshi Goto, VLSI/CAD 2004

IP --- The SoC Design Enabler

- IP = Intellectual Property
  - Also called embedded core, virtual component, megacell, system-level macro
- A well defined, pre-designed, pre-verified, and pre-characterized functional block that can be used across different system chips
  - e.g. ARM core, MPEG decoder, PCI, DMA controller, etc.

Design Component Paradigm Shifts



2000s

DSP RAM

MPEG4 RF

IP

1990s

```
always @ (posedge ck)
  count <= count + 1;
```

RTL

Layout design automation

1980s

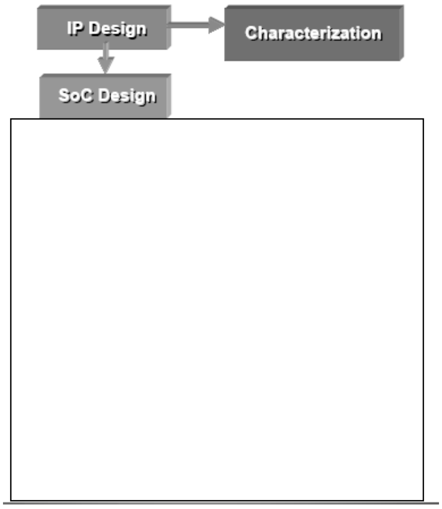
Logic gates

1970s

Transistors

Logic design automation

(IP-based) IC Realization Flow



IP Design

Characterization

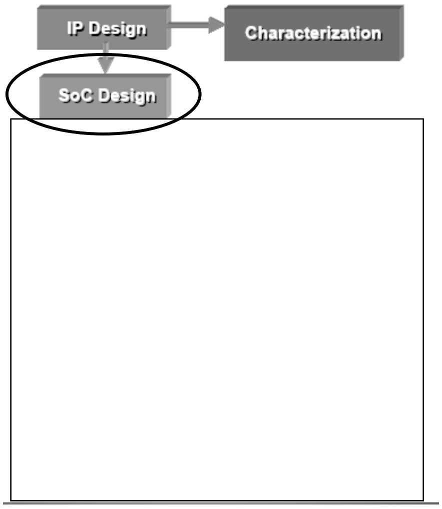
SoC Design

Yervant Zorian, DAC 2002

Types of IPs

- Soft IP
  - A verified synthesizable HDL (usually RTL) description
- Firm IP
  - A synthesized gate-level netlist that meets some basic assessments (e.g. timing)
- Hard IP
  - A complete design that contains layout and detailed timing, power, etc information

(IP-based) IC Realization Flow



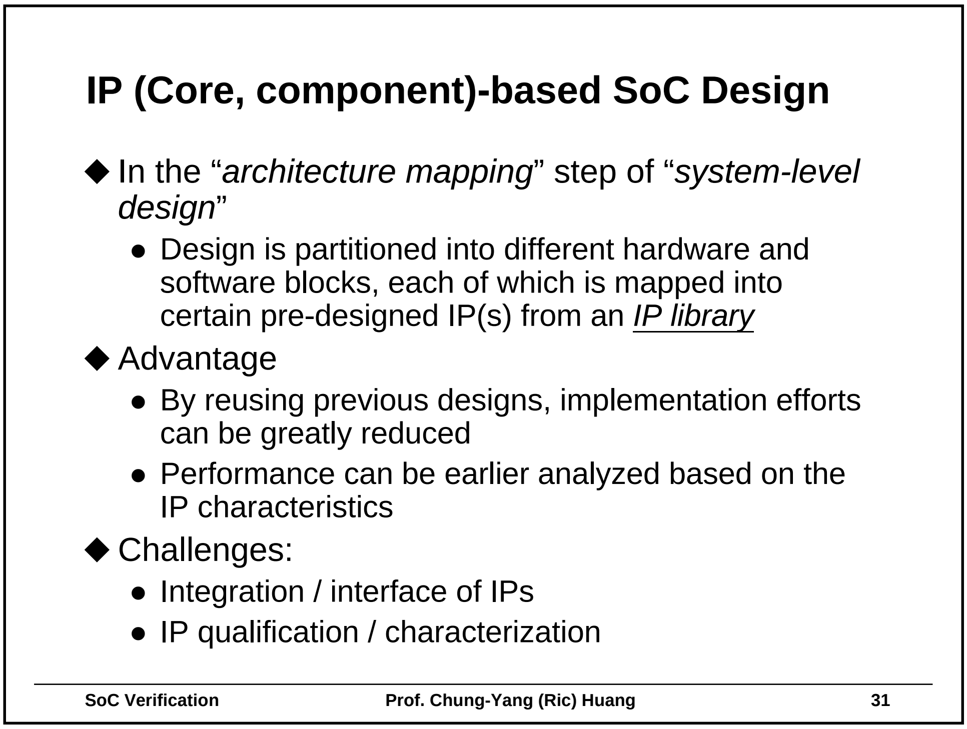
Yervant Zorian, DAC 2002

IP (Core, component)-based SoC Design

- In the “*architecture mapping*” step of “*system-level design*”
  - Design is partitioned into different hardware and software blocks, each of which is mapped into certain pre-designed IP(s) from an *IP library*
- Advantage
  - By reusing previous designs, implementation efforts can be greatly reduced
  - Performance can be earlier analyzed based on the IP characteristics

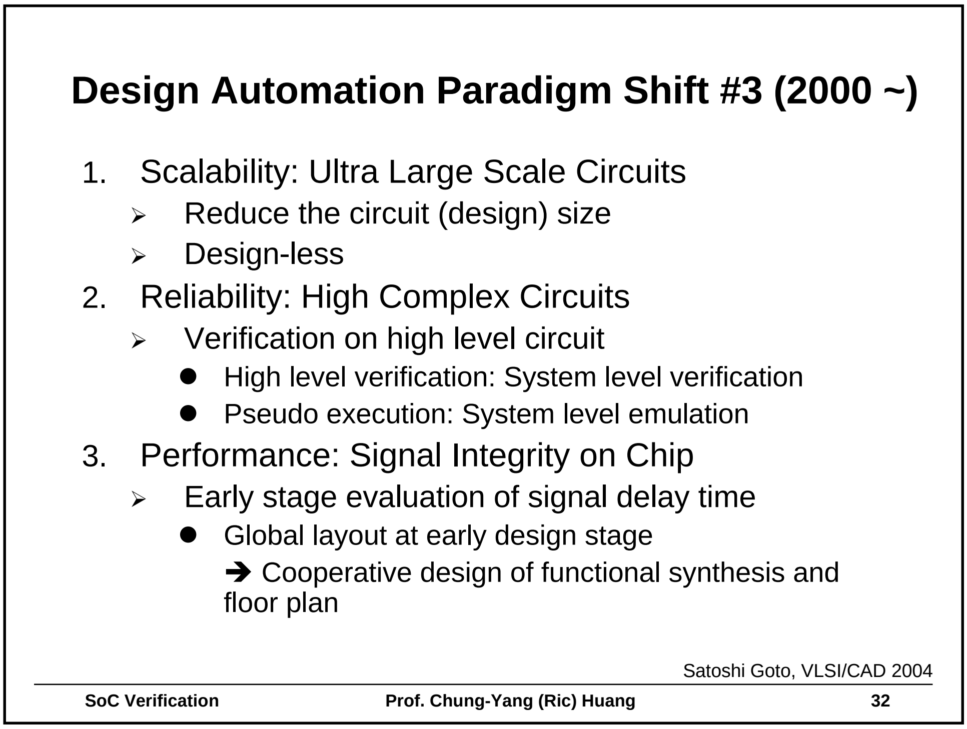
But remember, a system should (may) contain…

- I/O interface
  (Bus, RF, display driver, etc)
- Processor
  (General purposed, DSP, graphic, etc)
- Analog/Mixed signal units
  (ADC, DAC, PLL, etc)
- Network controller
- Embedded memory (RAM)
- Embedded software (ROM)
- Built-in self test (BIST)



IP (Core, component)-based SoC Design

- In the "*architecture mapping*" step of "*system-level design*"
  - Design is partitioned into different hardware and software blocks, each of which is mapped into certain pre-designed IP(s) from an *IP library*
- Advantage
  - By reusing previous designs, implementation efforts can be greatly reduced
  - Performance can be earlier analyzed based on the IP characteristics
- Challenges:
  - Integration / interface of IPs
  - IP qualification / characterization



Design Automation Paradigm Shift #3 (2000 ~)

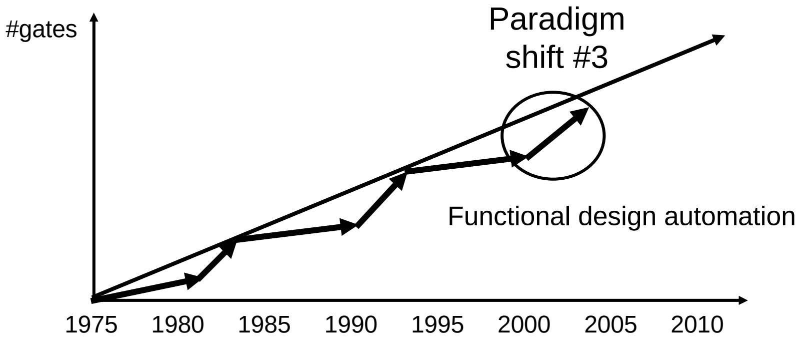
1. Scalability: Ultra Large Scale Circuits
   - Reduce the circuit (design) size
   - Design-less
2. Reliability: High Complex Circuits
   - Verification on high level circuit
     - High level verification: System level verification
     - Pseudo execution: System level emulation
3. Performance: Signal Integrity on Chip
   - Early stage evaluation of signal delay time
     - Global layout at early design stage

       ➔ Cooperative design of functional synthesis and floor plan

Satoshi Goto, VLSI/CAD 2004

Design Automation Paradigm Shift #3 (2000 ~)

1. Scalability: Ultra Large Scale Circuits
2. Reliability: High Complex Circuits
3. Performance: Signal Integrity on Chip

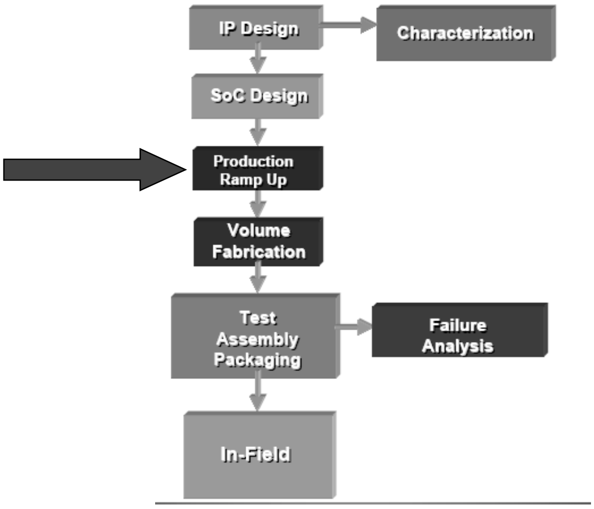


Satoshi Goto, VLSI/CAD 2004

SoC Design Challenge Summary

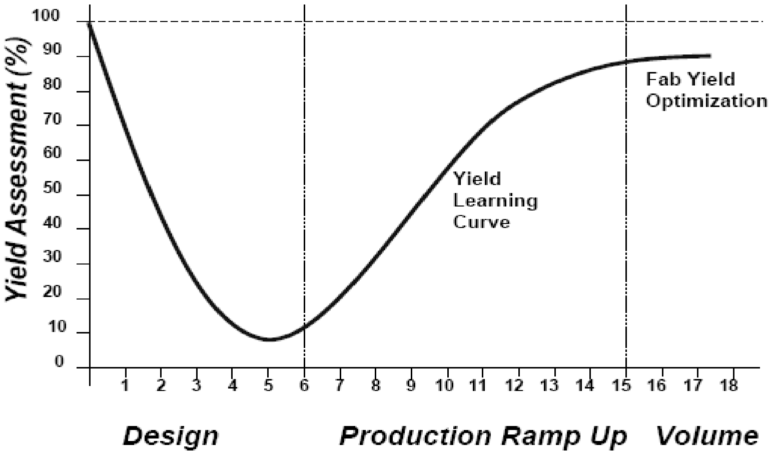
1. Design complexity
   - Human intelligence limitations
   - EDA tool limitations
   - New verification/testing methodologies
     ➔ Heterogeneous architectures, mixed signals, etc.
2. Yield & reliability
   - Manufacturability: Deep Sub-Micron (DSM) physical effects
   - Testability (test volume and time)
   - Yield learning (Design for Manufacturability) (DfM)
   - Yield optimization (Infrastructure IP; I^2P)

(IP-based) IC Realization Flow



Yervant Zorian, DAC 2002

Yield Life Cycle Curve

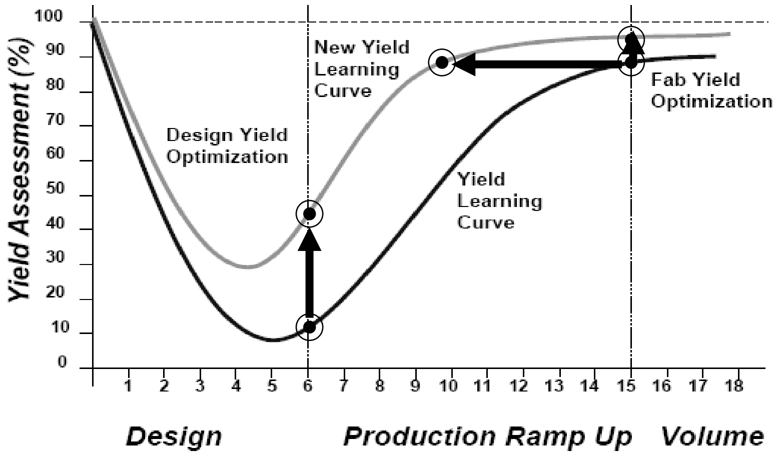


Yervant Zorian, DAC2002

Traditional Testing Infrastructure

- ◆External (of the Chip)
- ◆As the device geometry becomes smaller, and design gets more complex,
  - ● Testing
  - ● Manufacture monitoring
  - ● Timing parameter measurement
  - ● Diagnosis
  - ● Defect repair
  - ● Fault tolerance
  - ➔ becomes much more difficult and expensive

Yield Life Cycle Curve



Yervant Zorian, DAC2002

Functional vs. Infrastructure IP

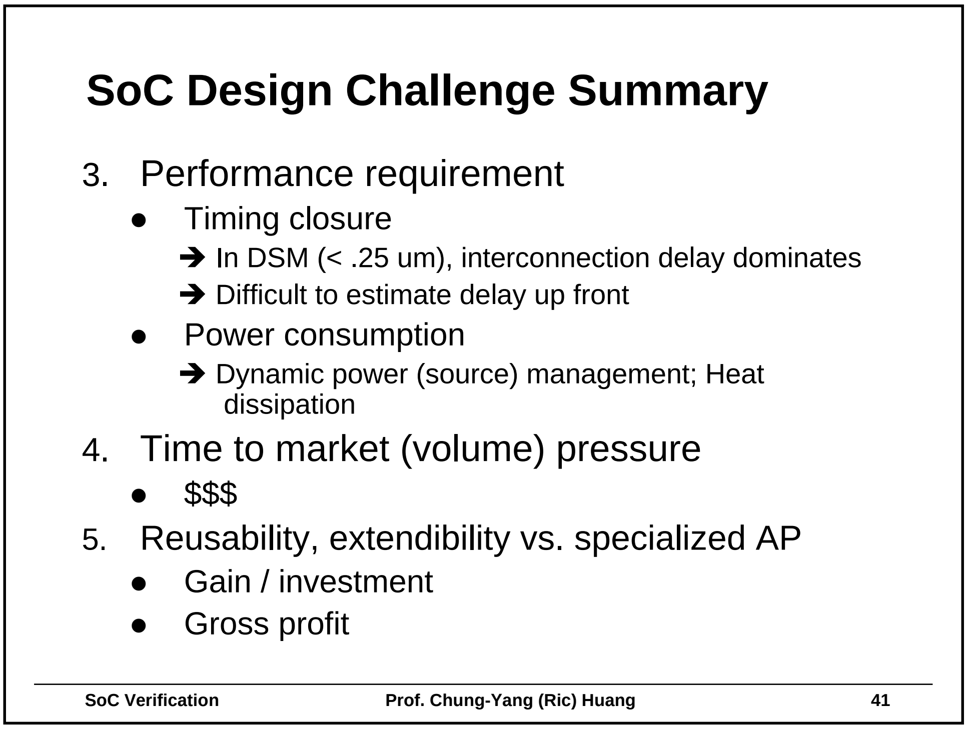
- Functional IP: the IPs we mentioned earlier
  - e.g. ARM core, MPEG decoder, PCI, DMA controller, etc.
- Infrastructure IP (I^2P):
  - Similar to that of a building's infrastructure elements, such as wiring networks or plumbing, which are independent from the actual function of the building

  ➔ Promote the testability, manufacturability, diagnosability, and error correctability

Examples of IIPs

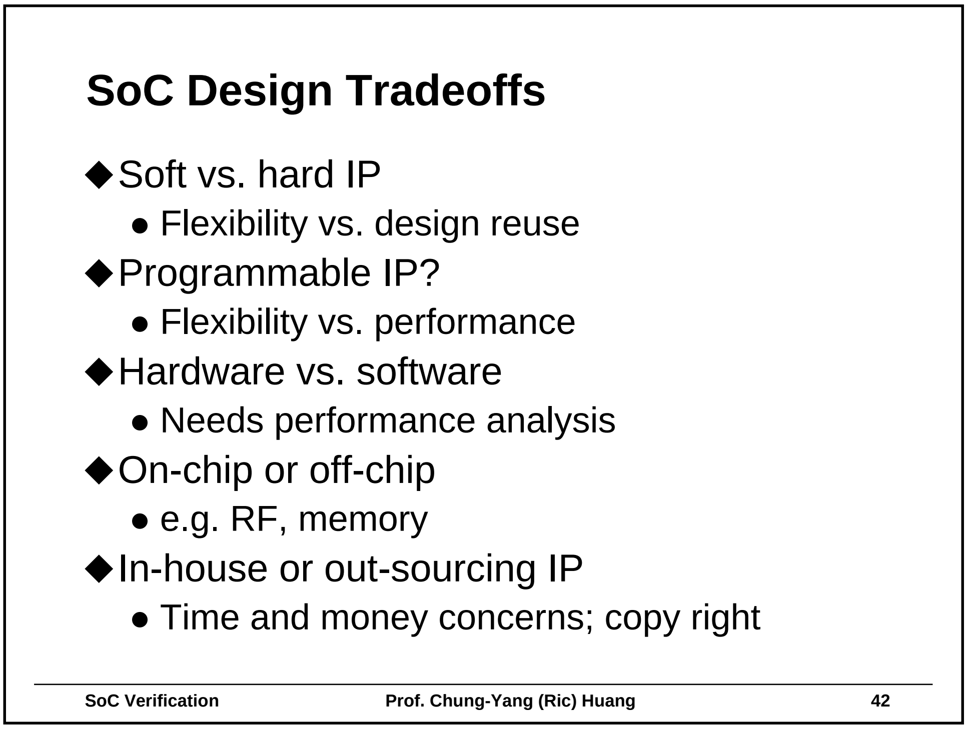
- Built-In Self-Test (BIST) for logic and memories
- Built-In Repair Analysis (BIRA)
- Built-In Self-Repair (BISR)
- Error-Correcting Codes (ECC) for embedded memories
- Embedded core test logic for SOCs
- Embedded timing analyzers to measure timing specifications
- Embedded fuse technology for on-chip repair

Yervant Zorian et al, Evaluation Engineering 2004



SoC Design Challenge Summary

3. Performance requirement
   - Timing closure
     - ➔ In DSM (< .25 um), interconnection delay dominates
     - ➔ Difficult to estimate delay up front
   - Power consumption
     - ➔ Dynamic power (source) management; Heat dissipation
4. Time to market (volume) pressure
   - $$$
5. Reusability, extendibility vs. specialized AP
   - Gain / investment
   - Gross profit



SoC Design Tradeoffs

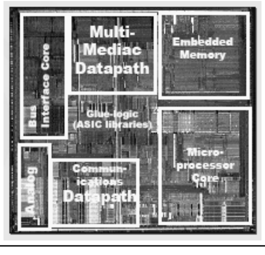
- ◆Soft vs. hard IP
  - Flexibility vs. design reuse
- ◆Programmable IP?
  - Flexibility vs. performance
- ◆Hardware vs. software
  - Needs performance analysis
- ◆On-chip or off-chip
  - e.g. RF, memory
- ◆In-house or out-sourcing IP
  - Time and money concerns; copy right

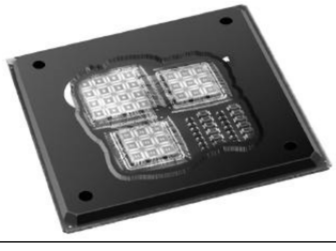
So, is SoC a viable solution?

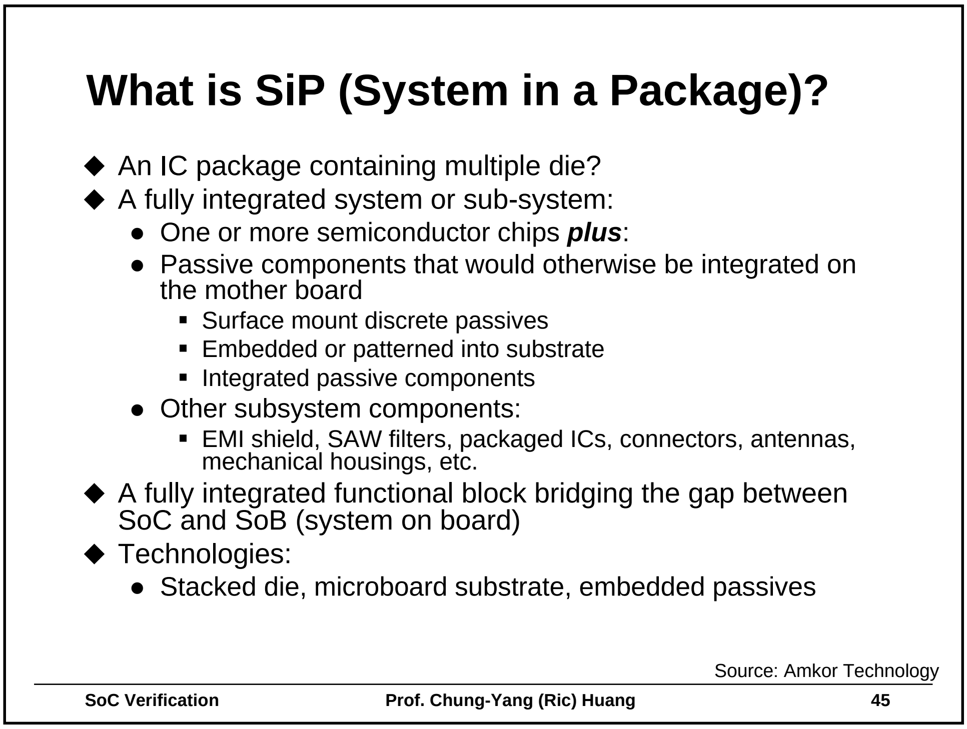
System Migration

- ◆System in a box
- ◆System on a board
- ◆System on a chip

System in a package



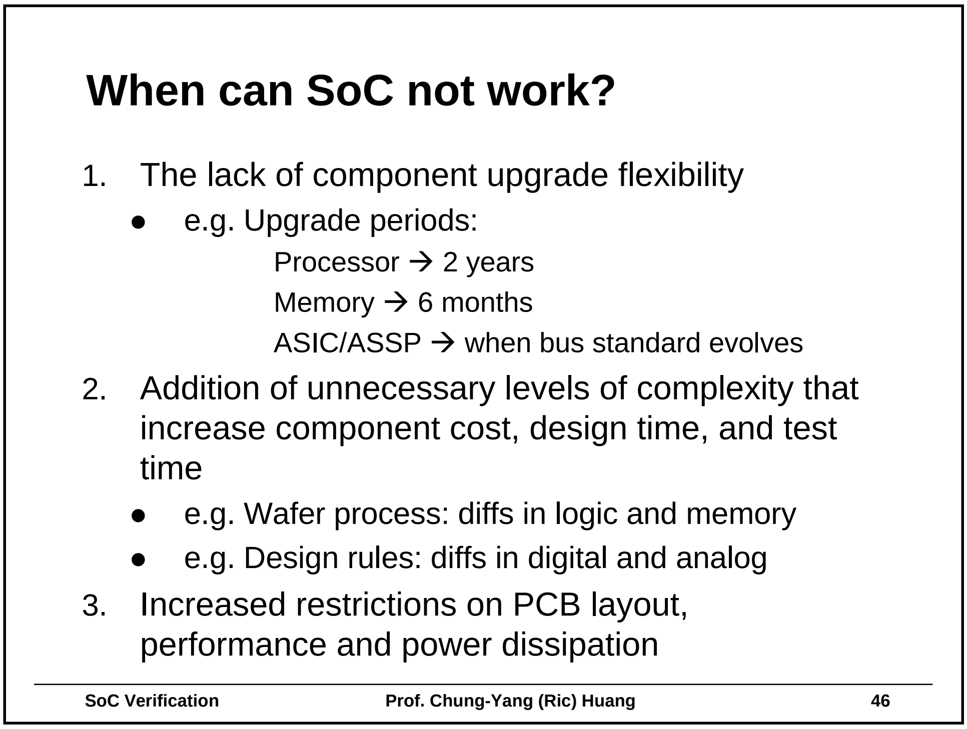




What is SiP (System in a Package)?

- An IC package containing multiple die?
- A fully integrated system or sub-system:
  - One or more semiconductor chips ***plus***:
  - Passive components that would otherwise be integrated on the mother board
    - Surface mount discrete passives
    - Embedded or patterned into substrate
    - Integrated passive components
  - Other subsystem components:
    - EMI shield, SAW filters, packaged ICs, connectors, antennas, mechanical housings, etc.
- A fully integrated functional block bridging the gap between SoC and SoB (system on board)
- Technologies:
  - Stacked die, microboard substrate, embedded passives

Source: Amkor Technology



When can SoC not work?

1. The lack of component upgrade flexibility
   - e.g. Upgrade periods:

     Processor → 2 years

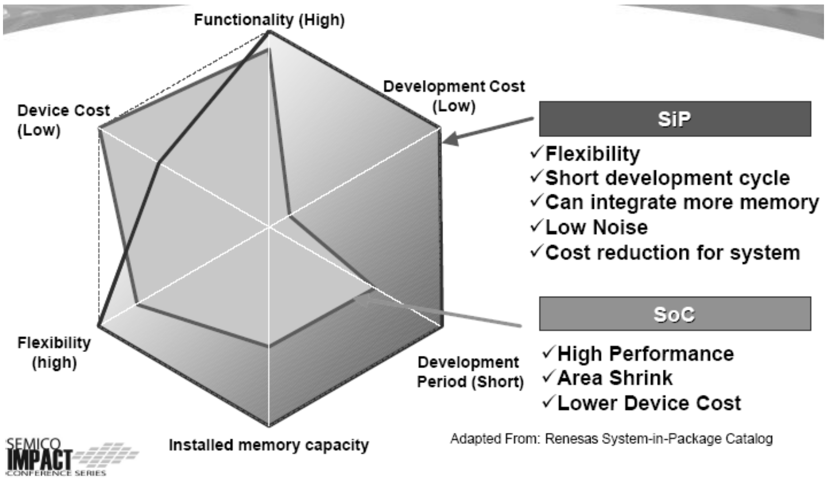
     Memory → 6 months

     ASIC/ASSP → when bus standard evolves
2. Addition of unnecessary levels of complexity that increase component cost, design time, and test time
   - e.g. Wafer process: diffs in logic and memory
   - e.g. Design rules: diffs in digital and analog
3. Increased restrictions on PCB layout, performance and power dissipation

Why SiP?

- ◆Performance of high-speed digital devices
  - ● Limited by inadequate packaging solutions rather than by internal delays
- ◆Power consumption of RF components
  - ● Poor quality factors of too many passive components spread all over the board
- ◆Consumer products
  - ● Packaging is becoming the most important cost factor

SoC vs. SiP



SiP

- ✓Flexibility
- ✓Short development cycle
- ✓Can integrate more memory
- ✓Low Noise
- ✓Cost reduction for system

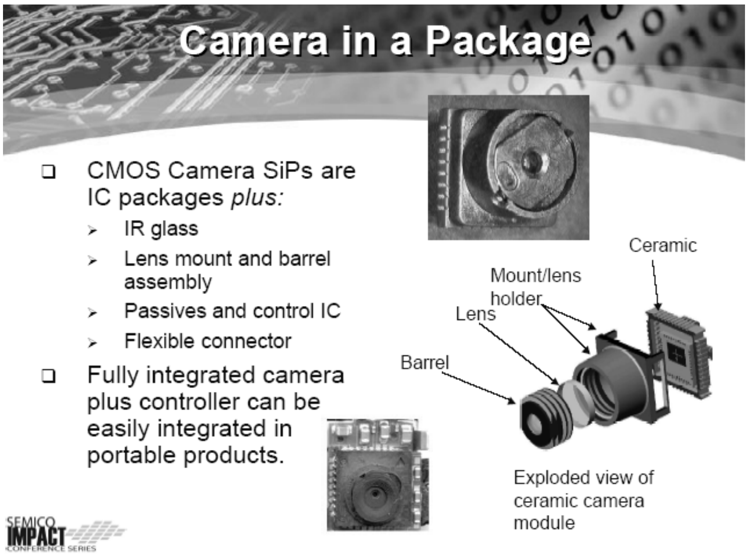
SoC

- ✓High Performance
- ✓Area Shrink
- ✓Lower Device Cost

Adapted From: Renesas System-in-Package Catalog

In comparison with SoC,
SiP has strong competitiveness in
design and test cycles,
production complexity,
and time-to-market
without sacrificing performance

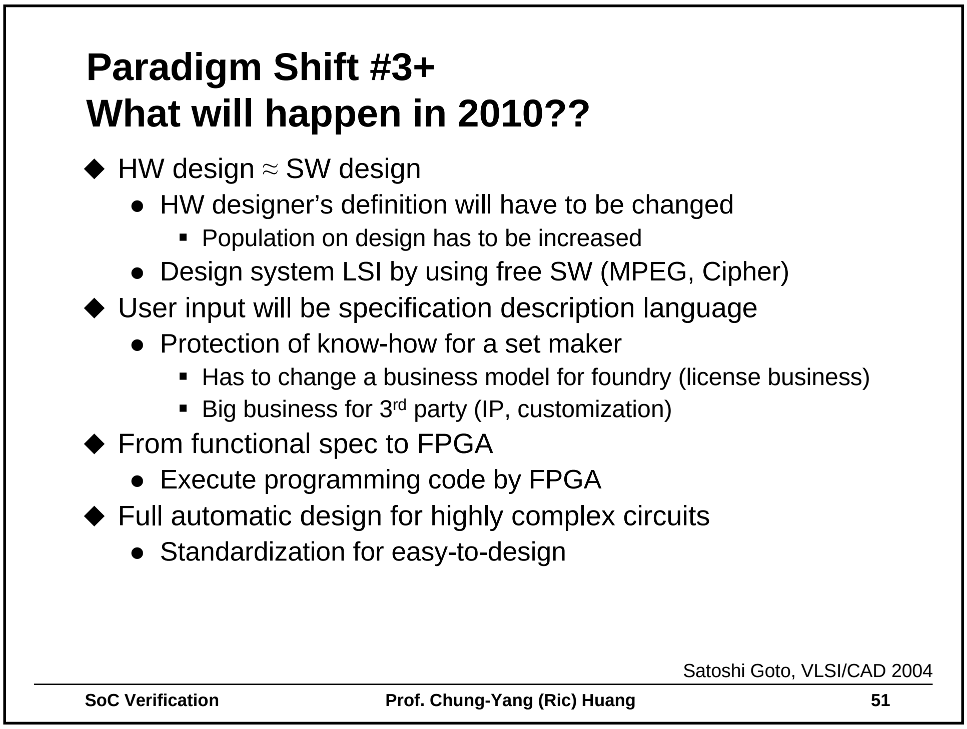
SiP Example



Camera in a Package

- CMOS Camera SiPs are IC packages *plus:*
  - IR glass
  - Lens mount and barrel assembly
  - Passives and control IC
  - Flexible connector
- Fully integrated camera plus controller can be easily integrated in portable products.

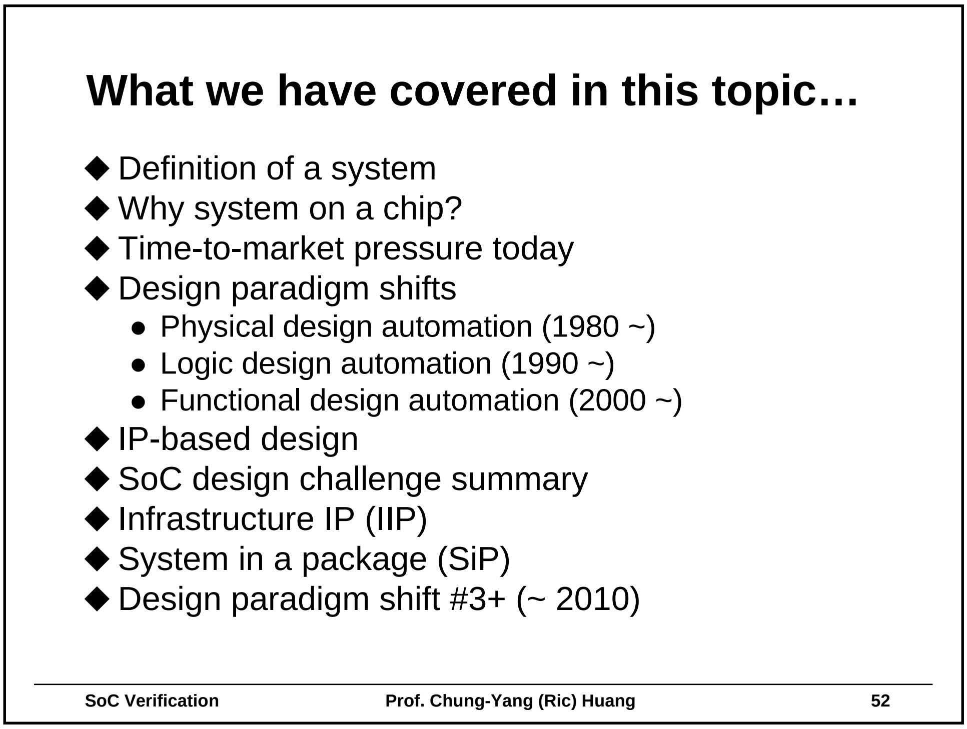
Ceramic
Mount/lens holder
Lens
Barrel
Exploded view of ceramic camera module
SEMICO IMPACT CONFERENCE SERIES



Paradigm Shift #3+ What will happen in 2010??

- ◆ HW design ≈ SW design
  - ● HW designer’s definition will have to be changed
    - ▪ Population on design has to be increased
  - ● Design system LSI by using free SW (MPEG, Cipher)
- ◆ User input will be specification description language
  - ● Protection of know-how for a set maker
    - ▪ Has to change a business model for foundry (license business)
    - ▪ Big business for 3rd party (IP, customization)
- ◆ From functional spec to FPGA
  - ● Execute programming code by FPGA
- ◆ Full automatic design for highly complex circuits
  - ● Standardization for easy-to-design

Satoshi Goto, VLSI/CAD 2004



What we have covered in this topic…

- ◆ Definition of a system
- ◆ Why system on a chip?
- ◆ Time-to-market pressure today
- ◆ Design paradigm shifts
  - ● Physical design automation (1980 ~)
  - ● Logic design automation (1990 ~)
  - ● Functional design automation (2000 ~)
- ◆ IP-based design
- ◆ SoC design challenge summary
- ◆ Infrastructure IP (IIP)
- ◆ System in a package (SiP)
- ◆ Design paradigm shift #3+ (~ 2010)