



Embedded Systems technologies

Riferimenti bibliografici

“Embedded System Design: A Unified Hardware/Software Introduction”, Frank Vahid, Tony Givargis, John Wiley & Sons Inc., ISBN:0-471-38678-2, 2002.

“Computers as Components: Principles of Embedded Computer Systems Design”, Wayne Wolf, Morgan Kaufmann Publishers, ISBN: 1-55860-541-X, 2001

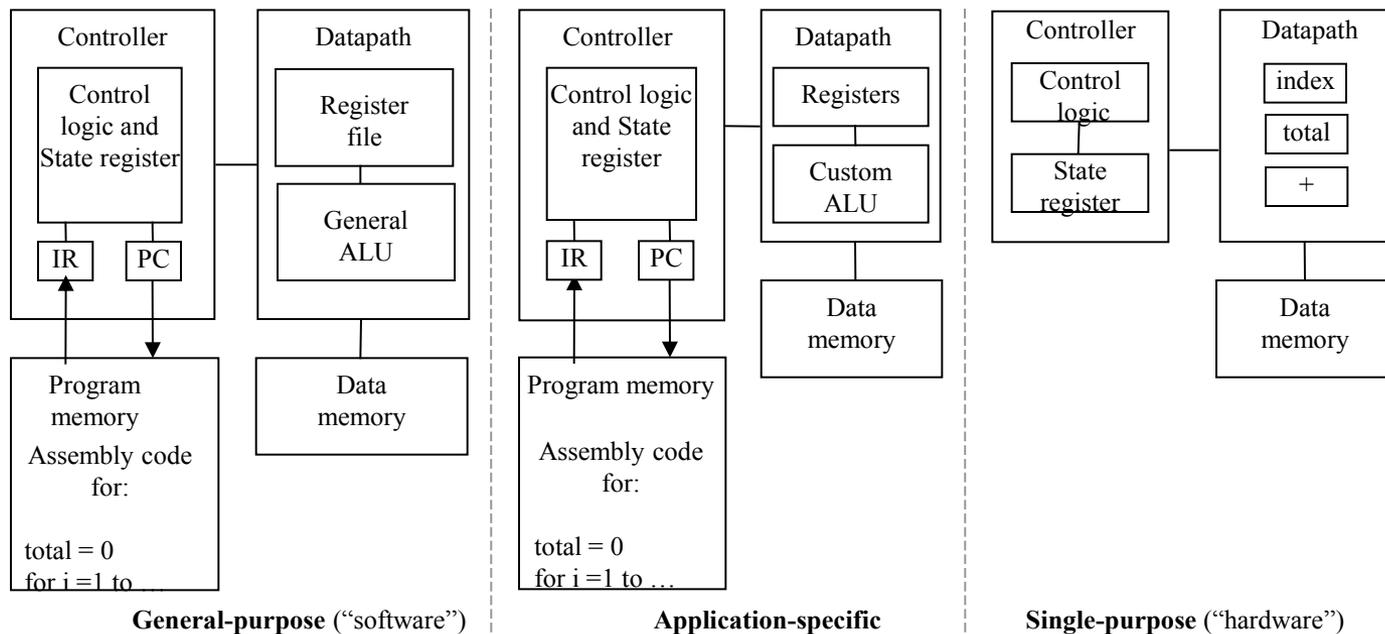
Embedded System Design” by Peter Marwedel, Kluwer Academic Publishers, ISBN: 1-4020-7690-8, October 2003

Three key embedded system technologies

- Technology
 - ▣ A manner of accomplishing a task, especially using technical processes, methods, or knowledge
- Three key technologies for embedded systems
 - ▣ Processor technology
 - ▣ IC technology
 - ▣ Design technology

Processor technology

- The architecture of the computation engine used to implement a system's desired functionality
- Processor does not have to be programmable
 - “Processor” *not* equal to general-purpose processor



Processor technology

- Processors vary in their customization for the problem at hand

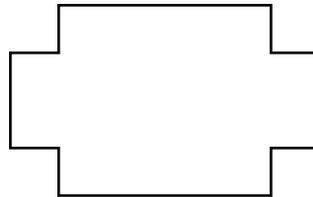


Desired
functionality

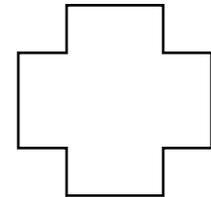
```
total = 0;  
for (i = 0; i < N; i++)  
    total += C[i]*M[i];
```



General-purpose
processor



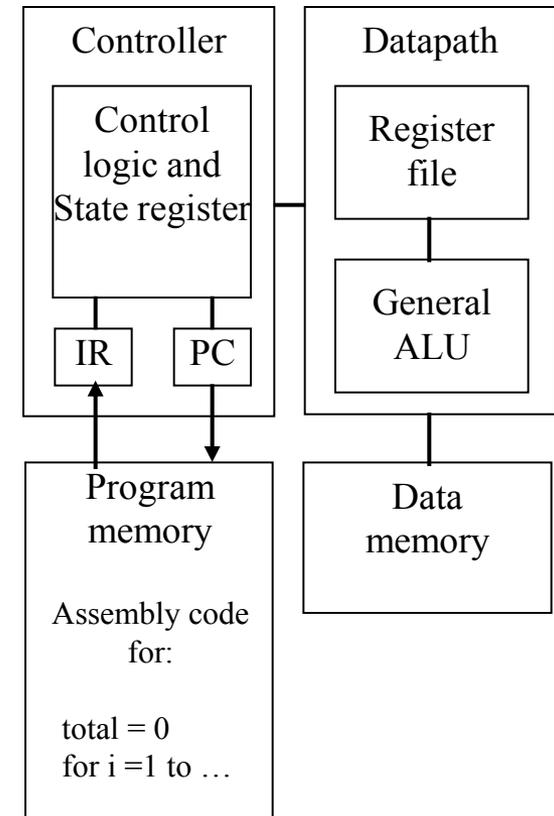
Application-specific
processor



Single-purpose
processor

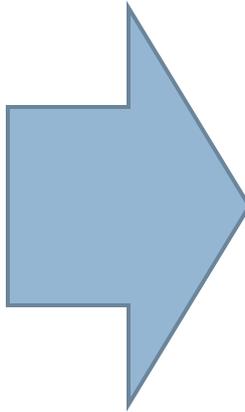
General-purpose processors

- Programmable device used in a variety of applications
 - Also known as “microprocessor”
- Features
 - Program memory
 - General datapath with large register file and general ALU
- User benefits
 - Low time-to-market and NRE costs
 - High flexibility
- Drawbacks
 - High unit cost
 - Low Performance



General-purpose processors

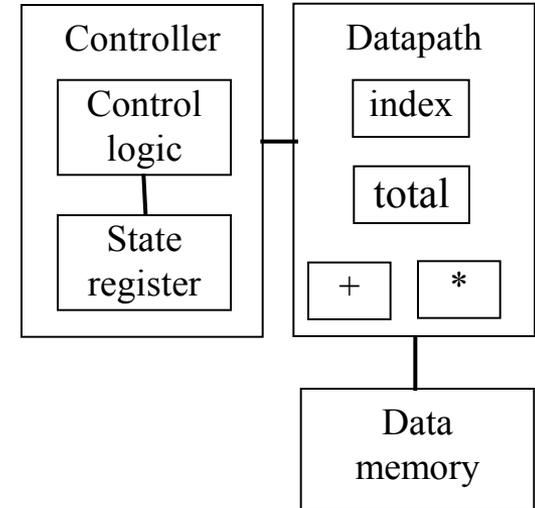
```
total = 0;
for (i = 0; i < N; i++)
    total += C[i]*M[i];
```



```
init:  daddi r8, r0, 0
       daddi r9, r0, 40
       daddi r10, r0, 0
for:   ld r11, M(r8)
       ld r12, C(r8)
       mult r11, r12
       mflo r11
       dadd r10, r10, r11
       daddi r8, r8, 8
       bne r8, r9, for
       sd r10, total(r0)
```

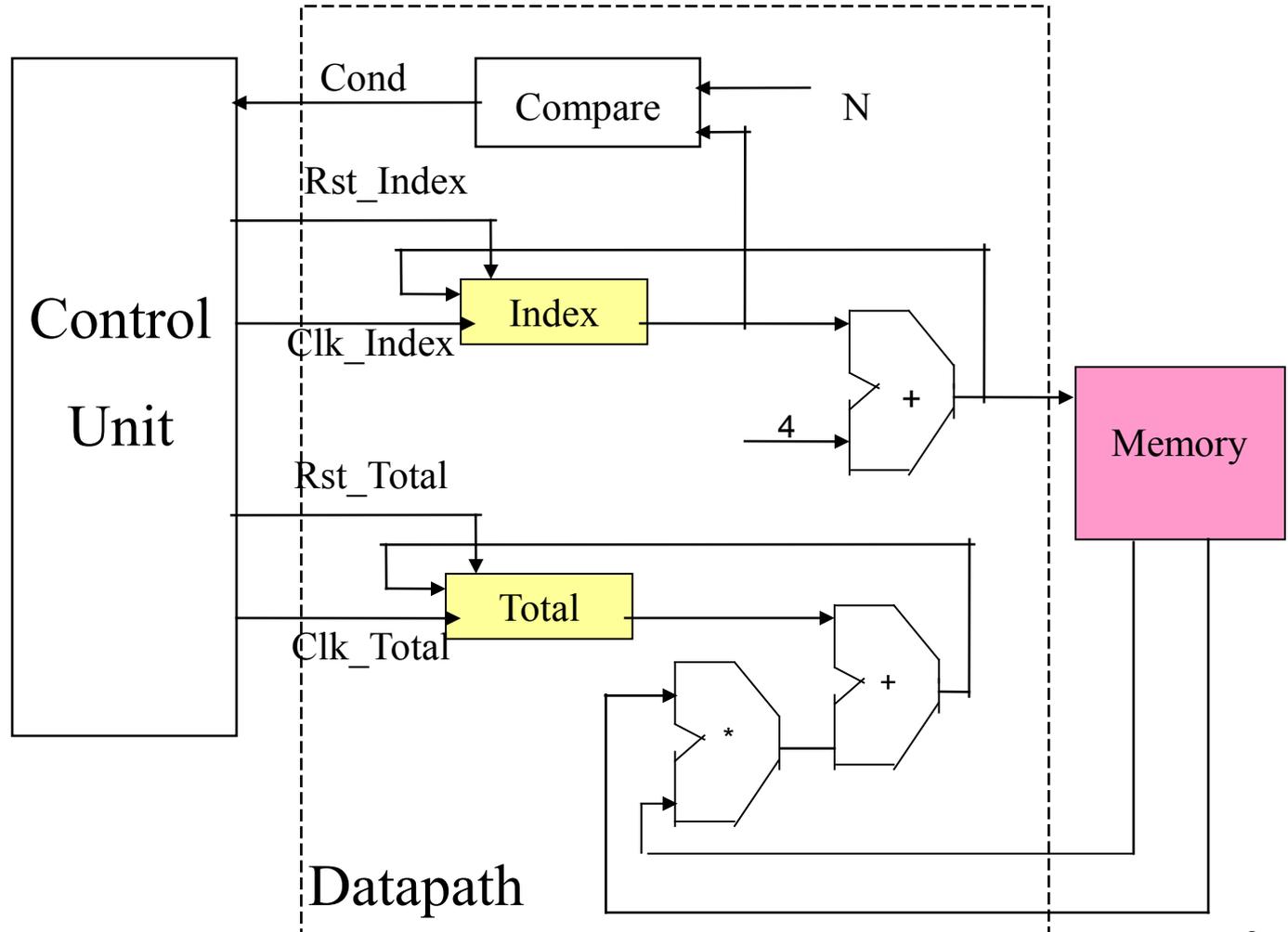
Single-purpose processors

- Digital circuit designed to execute exactly one program
 - a.k.a. coprocessor, accelerator or peripheral
- Features
 - Contains only the components needed to execute a single program
 - No program memory
- Benefits
 - Fast
 - Low power
- Drawbacks
 - No flexibility, high time-to-market, high NRE cost



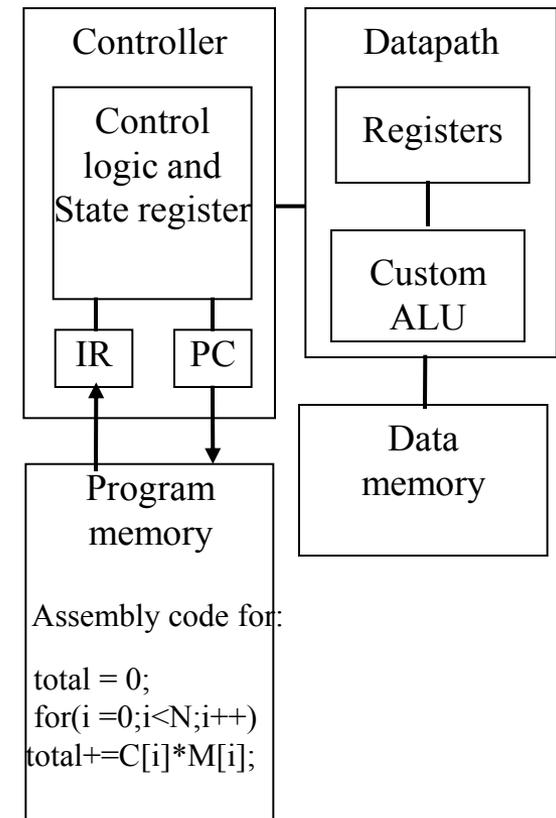
Single-purpose processors

```
total = 0;  
for (i=0;i<N;i++)  
    total +=  
    C[i]*M[i];
```



Application-specific processors

- Programmable processor optimized for a particular class of applications having common characteristics
 - ❑ Compromise between general-purpose and single-purpose processors
- Features
 - ❑ Program memory
 - ❑ Optimized datapath
 - ❑ Special functional units
- Benefits
 - ❑ Some flexibility, good performance, size and power
- Drawbacks
 - ❑ High NRE cost (processor and compiler)
- Examples: Microcontroller, DSP

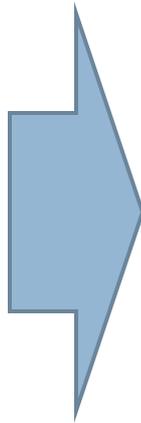


Application-specific processors

```
Macc Ra, Rb, Rc          ; Ra=Ra+Rb*Rb
```

```
Bltid Ra, Rb, target    ; if (Ra<Rb) {Ra+=8; j target}
```

```
total = 0;  
for (i = 0; i < N; i++)  
    total += C[i]*M[i];
```



```
init:  daddi r8, r0, 0  
       daddi r9, r0, 32  
       daddi r10, r0, 0  
for:   ld r11, M(r8)  
       ld r12, C(r8)  
       macc r10, r11, r12  
       bltid r8, r9, for  
       sd r10, total(r0)
```

Digital Signal Processors (DSP)

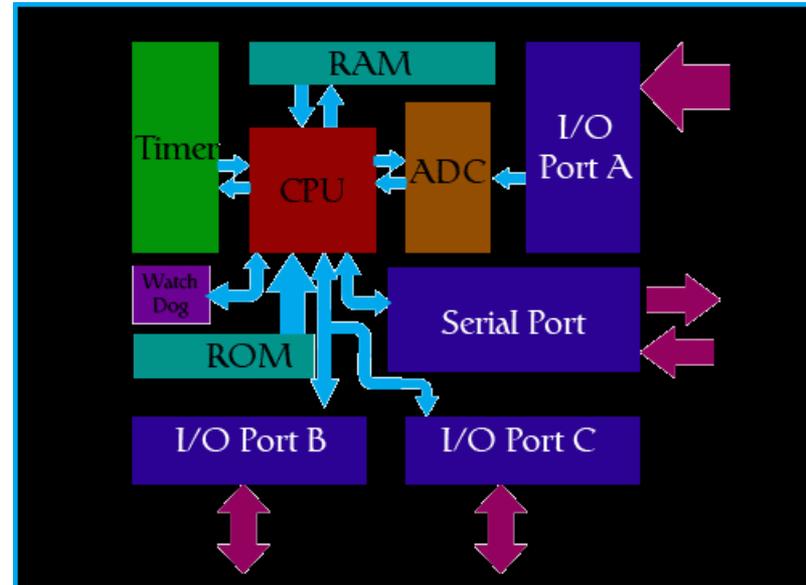
- For signal processing applications
 - Large amounts of digitized data, often streaming
 - Data transformations must be applied fast
 - e.g., cell-phone voice filter, digital TV, music synthesizer
- DSP features
 - Several instruction execution units
 - Multiple-accumulate single-cycle instruction, other instrs.
 - Efficient vector operations – e.g., add two arrays
 - Vector ALUs, loop buffers, etc.

A Common ASIP: Microcontroller

- For embedded control applications
 - Reading sensors, setting actuators
 - Mostly dealing with events (bits): data is present, but not in huge amounts
 - e.g. disk drive, digital camera (assuming SPP for image compression), washing machine, microwave oven

•Microcontroller features

- On-chip peripherals
 - Timers, analog-digital converters, serial communication, etc.
 - Tightly integrated for programmer, typically part of register space
- On-chip program and data memory
- Direct programmer access to many of the chip's pins
- Specialized instructions for bit-manipulation and other low-level

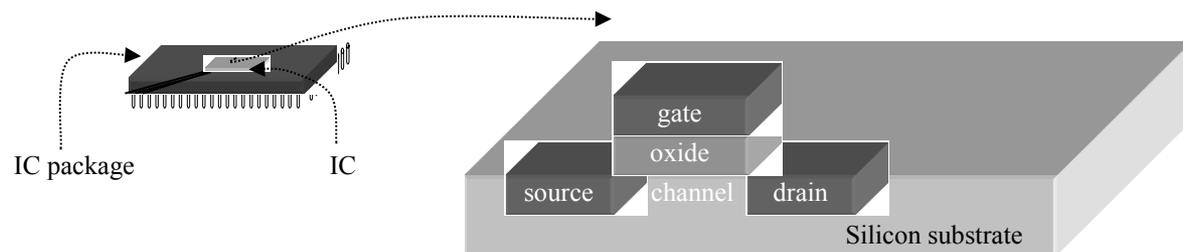




Integrated Circuit Technology

Integrated circuit (IC) technology

- The manner in which a digital (gate-level) implementation is mapped onto an IC
 - ▣ IC: Integrated circuit, or “chip”
 - ▣ IC technologies differ in their customization to a design
 - ▣ IC’s consist of numerous layers (perhaps 10 or more)
 - IC technologies differ with respect to who builds each layer and when



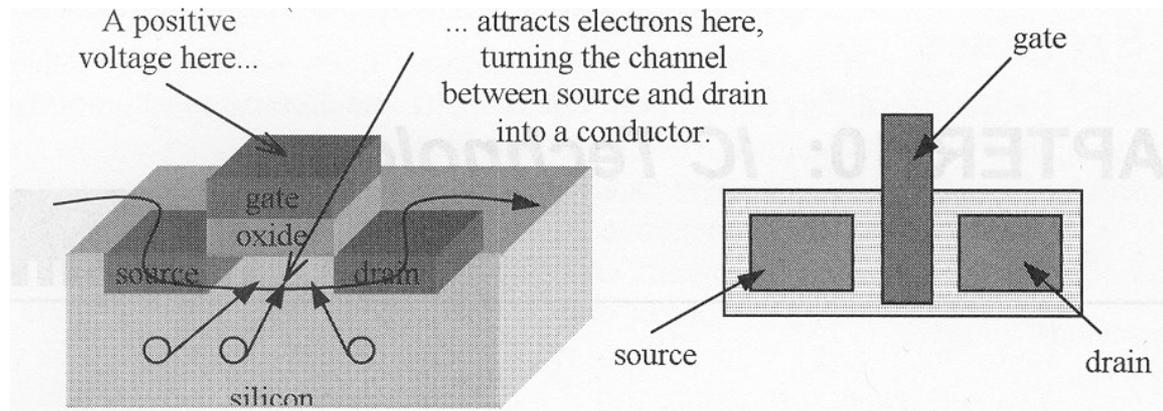
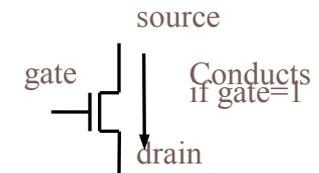
CMOS transistor

The basic electrical component in digital systems

Acts as an on/off switch

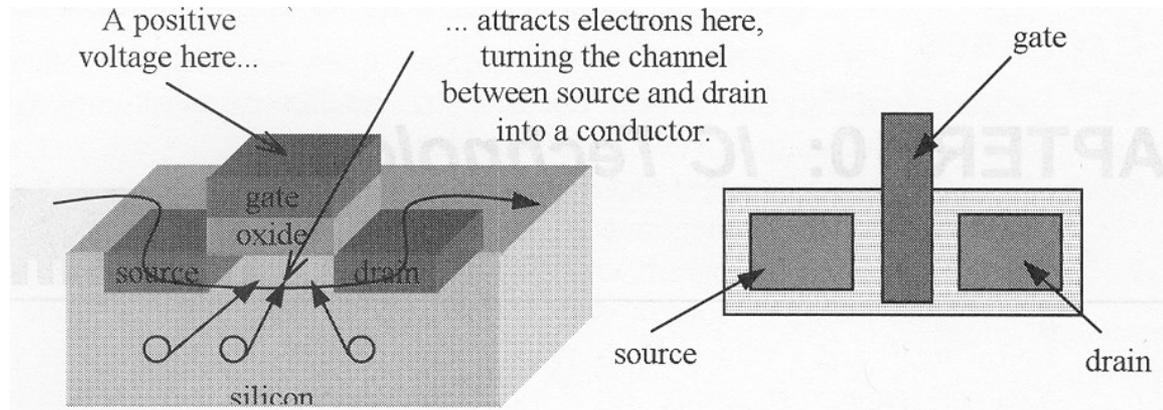
Voltage at “gate” controls whether current flows from source to drain

Don't confuse this “gate” with a logic gate



CMOS transistor

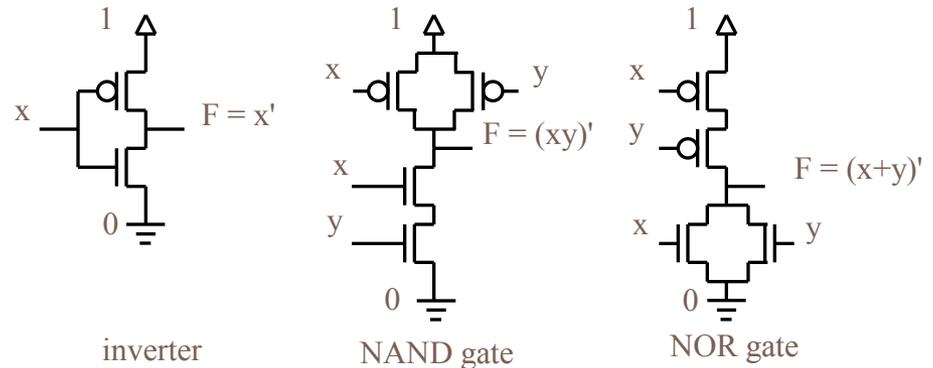
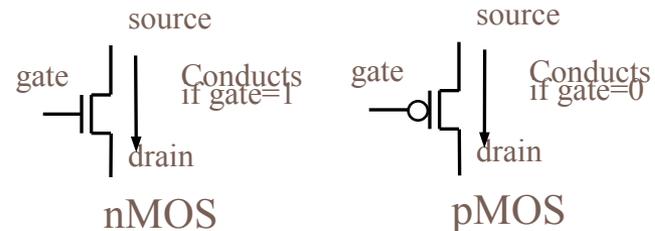
- Source, Drain
 - ▣ Diffusion area where electrons can flow
 - ▣ Can be connected to metal contacts (via's)
- Gate
 - ▣ Polysilicon area where control voltage is applied
- Oxide
 - ▣ Si O₂ Insulator so the gate voltage can't leak



CMOS transistor implementations

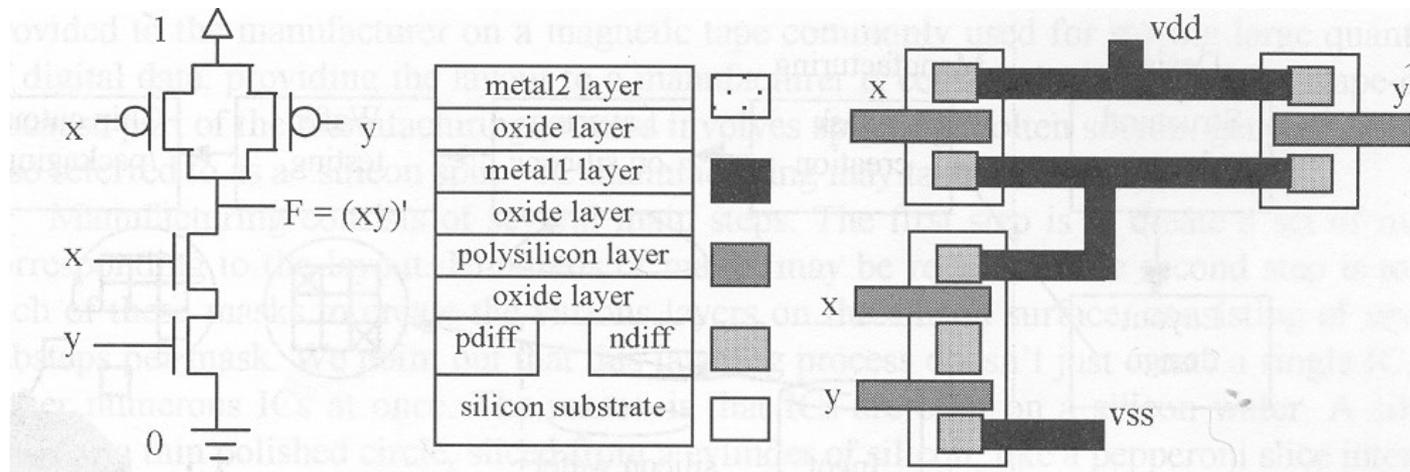
Complementary Metal Oxide Semiconductor

- We refer to logic levels
 - Typically 0 is 0V, 1 is V_{dd}
- Two basic CMOS types
 - nMOS conducts if gate=1
 - pMOS conducts if gate=0
 - Hence “complementary”
- Basic gates
 - Inverter, NAND, NOR



IC technology

NAND



IC Technologies

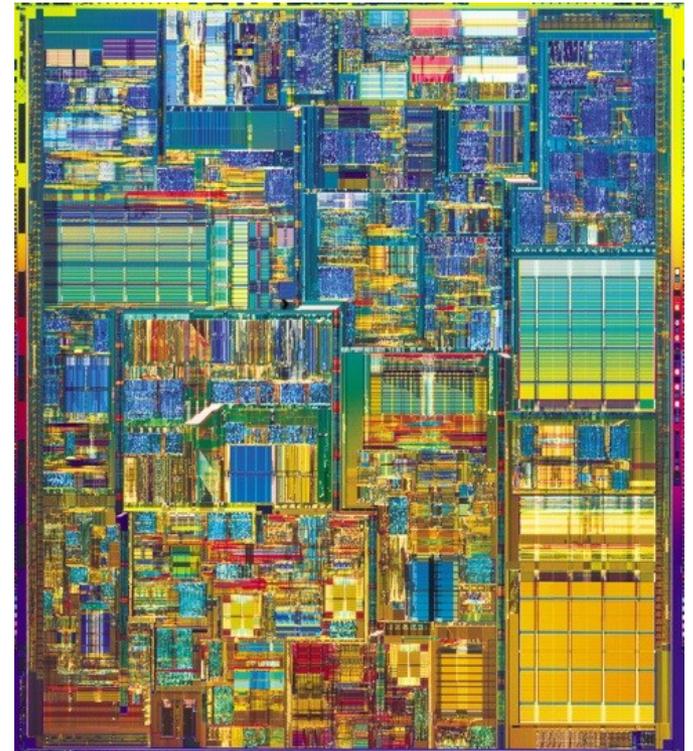
- Three types of IC technologies
 - Full-custom/VLSI
 - Semi-custom ASIC (gate array and standard cell)
 - PLD (Programmable Logic Device)

Full-custom

- Very Large Scale Integration (VLSI)
- All layers are optimized for an embedded system's particular digital implementation
- Placement
 - ▣ Place and orient transistors
- Routing
 - ▣ Connect transistors
- Sizing
 - ▣ Make fat, fast wires or thin, slow wires
 - ▣ May also need to size buffer
- Benefits
 - ▣ Excellent performance, small size, low power

Full-custom/VLSI

- Hand design
 - ▣ Horrible time-to-market/flexibility/NRE cost...
 - ▣ Reserve for the most important units in a processor
 - ALU, Instruction fetch...
- Physical design tools
 - ▣ Less optimal, but faster...
- Drawbacks
 - ▣ High NRE cost (e.g., \$300k), long time-to-market



Semi-custom

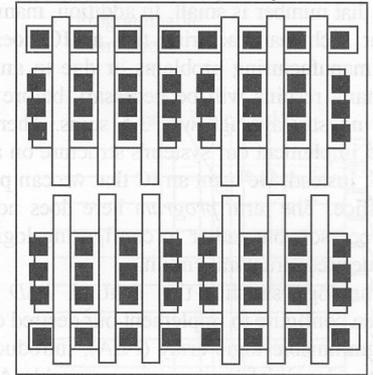
- Lower layers are fully or partially built
 - ▣ Designers are left with routing of wires and maybe placing some blocks
- Benefits
 - ▣ Good performance, good size, less NRE cost than a full-custom implementation (perhaps \$10k to \$100k)
- Drawbacks
 - ▣ Still require weeks to months to develop

Semi-custom

□ Gate Array

- Array of prefabricated gates
- “place” and route
- Higher density, faster time-to-market
- Does not integrate as well with full-custom

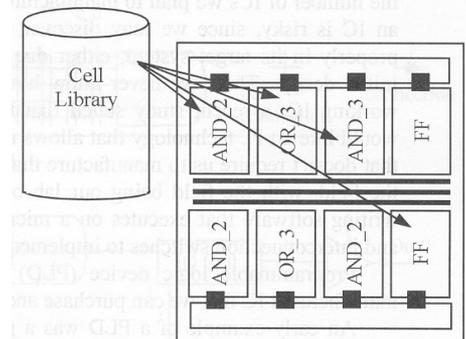
Gate array



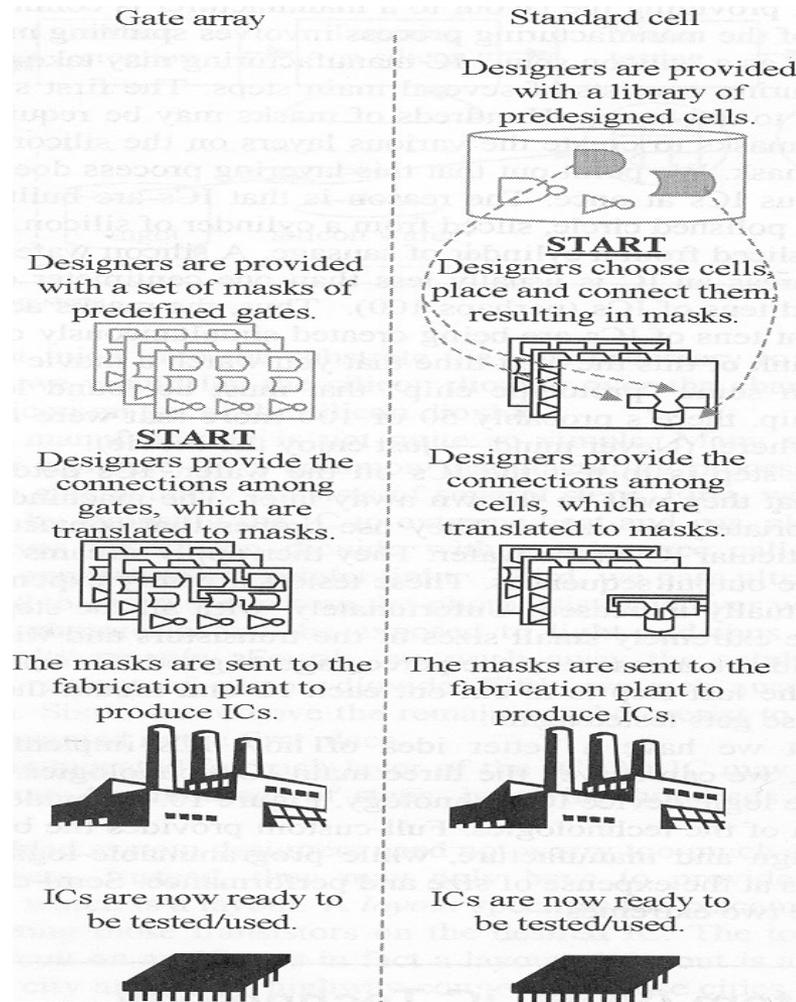
□ Standard Cell

- A library of pre-designed cell
- Place and route
- Lower density, higher complexity
- Integrate great with full-custom

Standard Cell



Semi-custom



PLD (Programmable Logic Device)

- Programmable Logic Device
 - ▣ Programmable Logic Array, Programmable Array Logic, Field Programmable Gate Array
- The layout is composed of an array of elementary programmable modules implementing a generic logic function and the interconnection among modules.
- The layout and fabrication process of each device is completed in advance and independently of the application. The device customization is obtained by programming on-site the device (after production)

PLD (Programmable Logic Device)

- There are different degrees of programmability :
 - ▣ one-time programmable (OTP): the configuration of the chip is irreversible and is obtained by applying electric voltages higher than those of normal power
 - ▣ reprogrammable : the configuration can be done several times offline; interconnections are driven by the bits of a circuit of the volatile memory (static RAM) or persistent (EEPROM, Flash)
 - ▣ reconfigurable: the configuration can be performed several times while the circuit is running and selectively

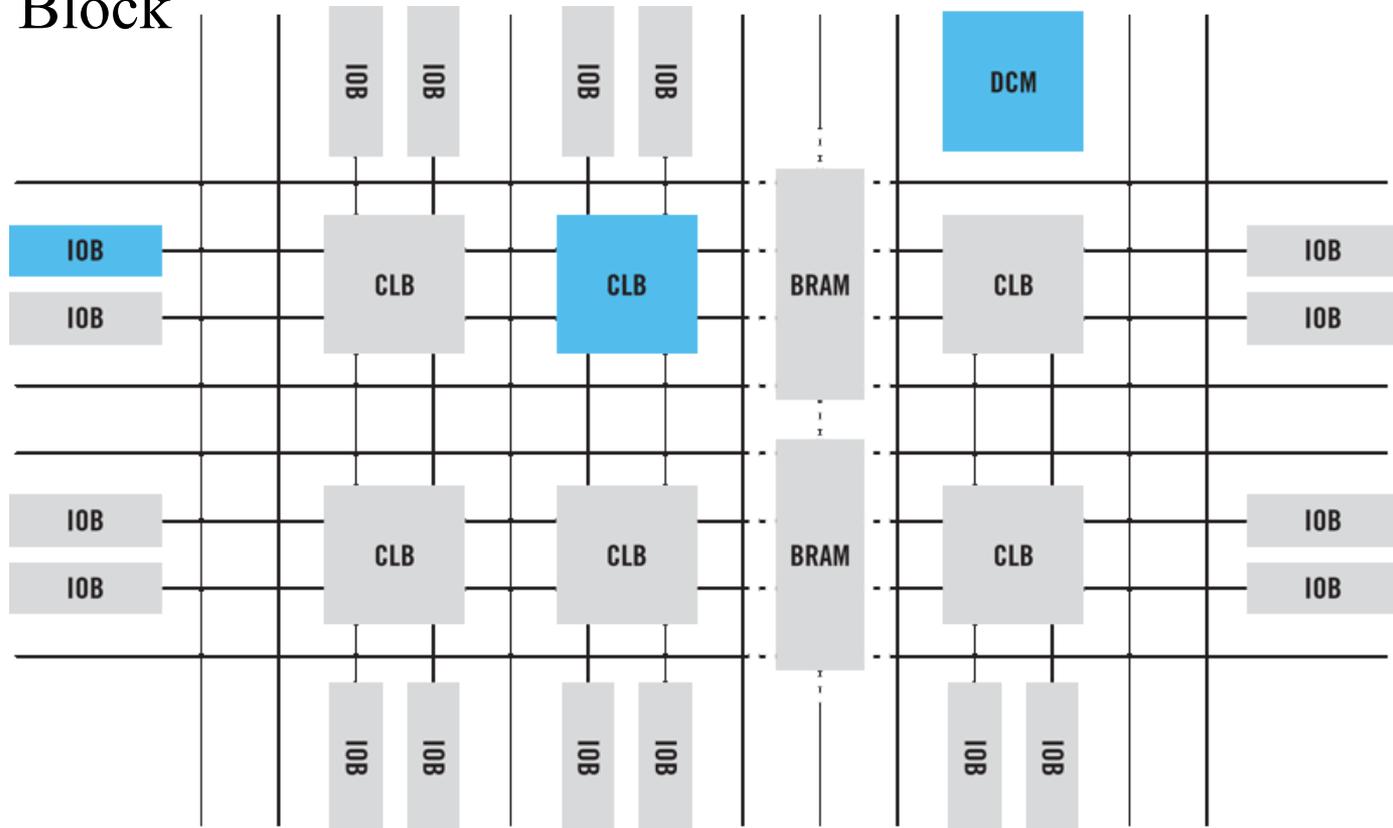
PLD (Programmable Logic Device)

- Benefits
 - Very low NRE costs
 - Immediate turn-around-time
- Drawback
 - High unit cost, bad for large volume
 - Power
 - Except special PLA
 - Low performance and integration density with respect to other design styles
- Suitable for low volumes and for prototyping phases.

FPGA

CLB: Configurable Logic Block

IOB: I/O Block



Configurable Logic Block (CLB)

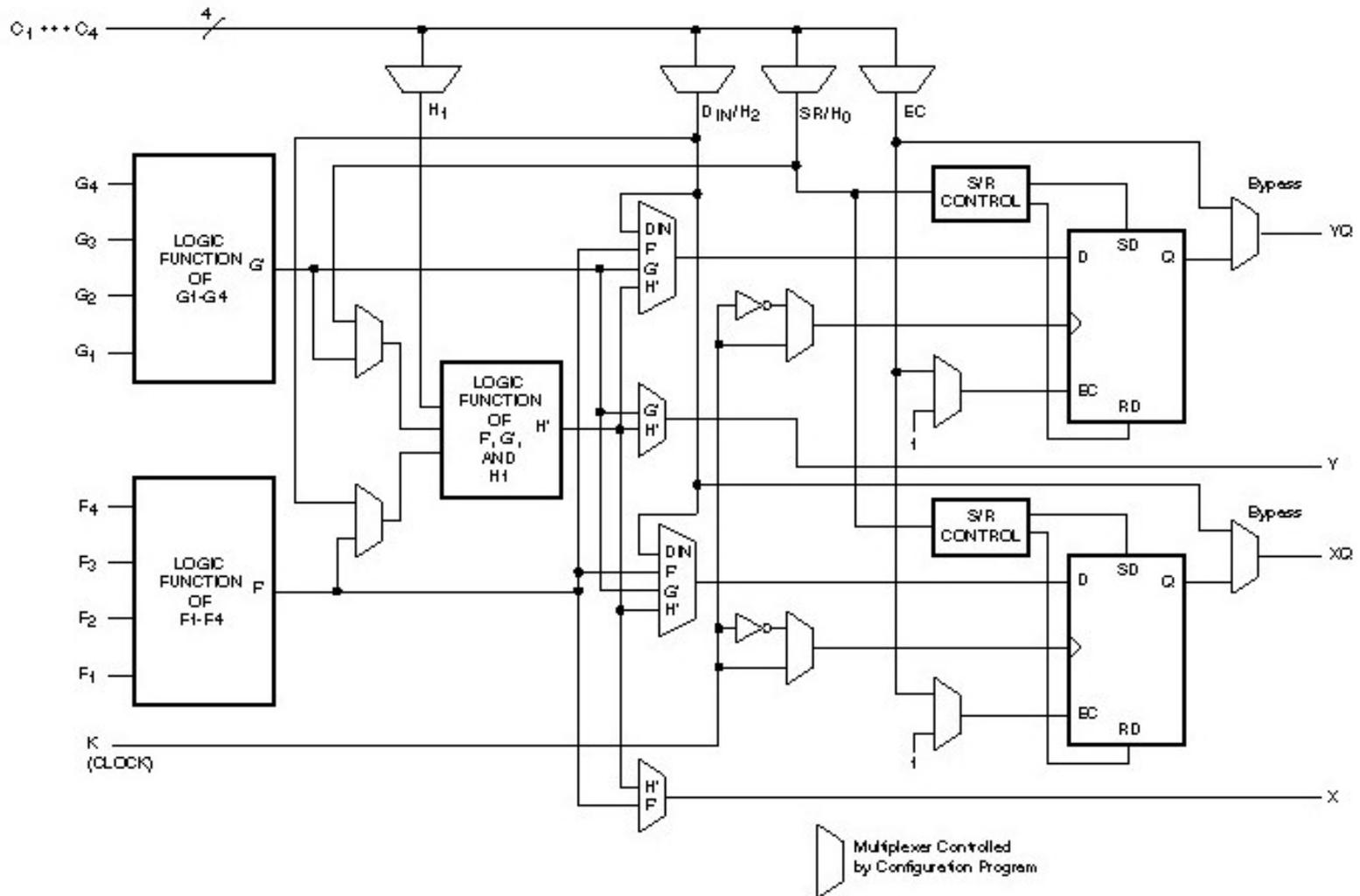
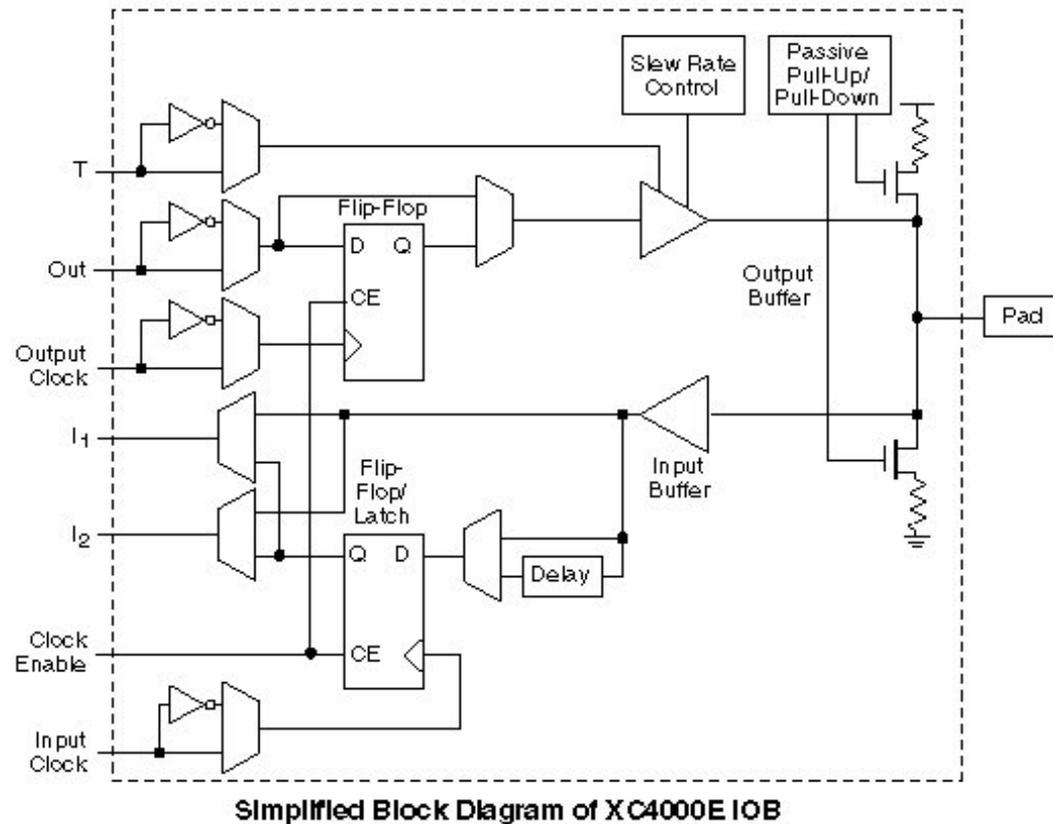
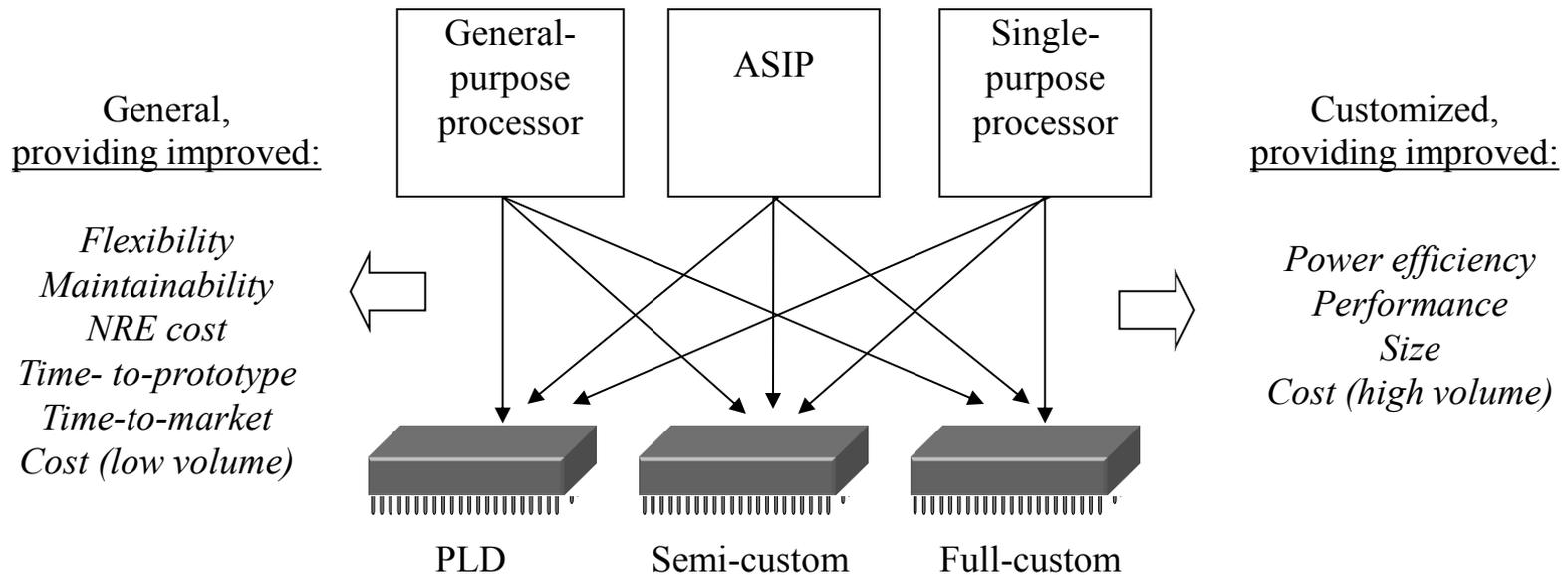


Figure 1: Simplified Block Diagram of XC4000-Series CLB (RAM and Carry Logic functions not shown)

I/O block



Independence of processor and IC technologies





Design Technology

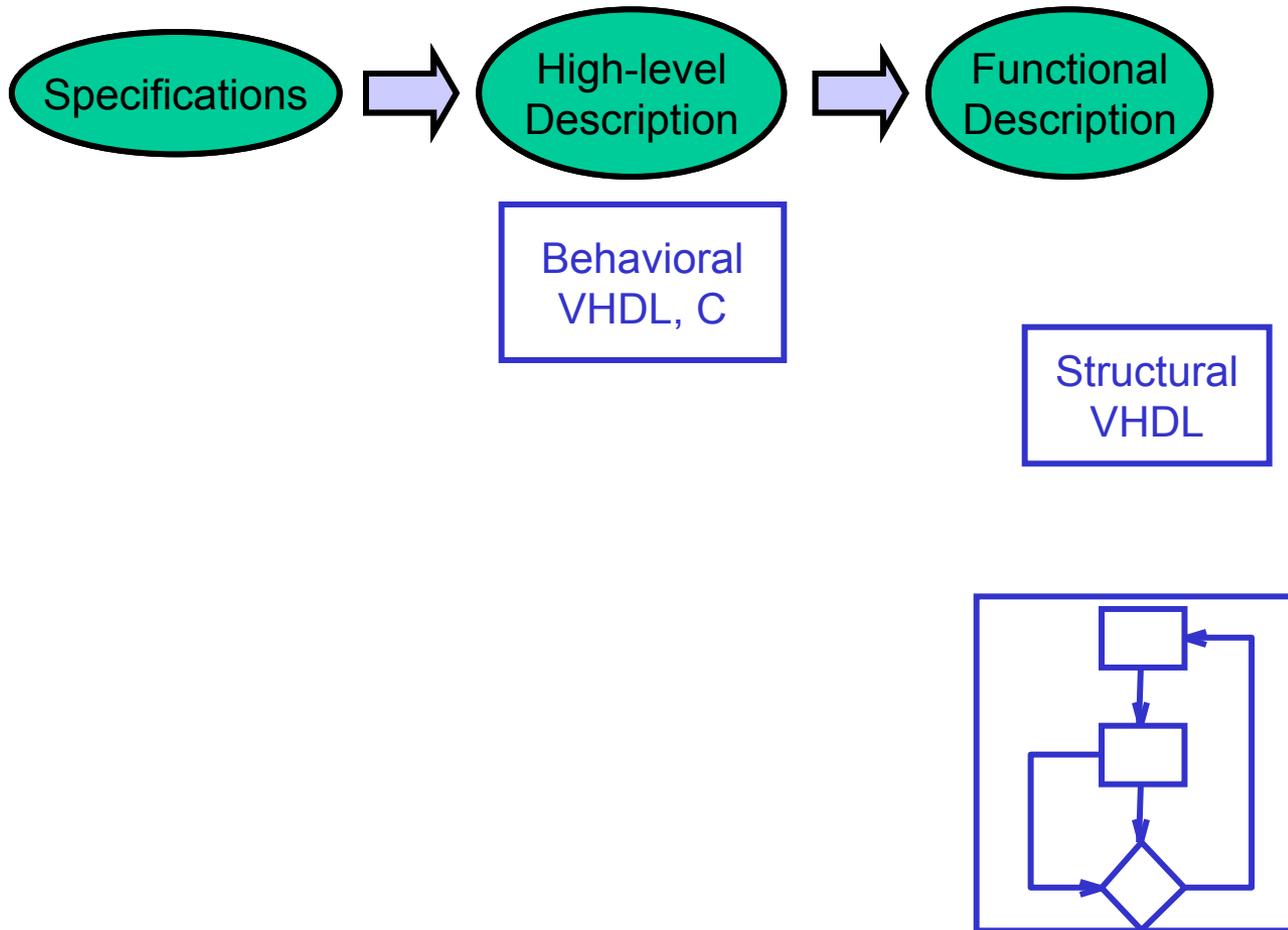
Design Technology

- A procedure for designing a system
- Many systems are complex and pose many design challenges: Large specifications, short time-to-market, high performance, multiple designers, interface to manufacturing.
- Proper design methodology helps to manage the design process and improves quality, performance and design costs

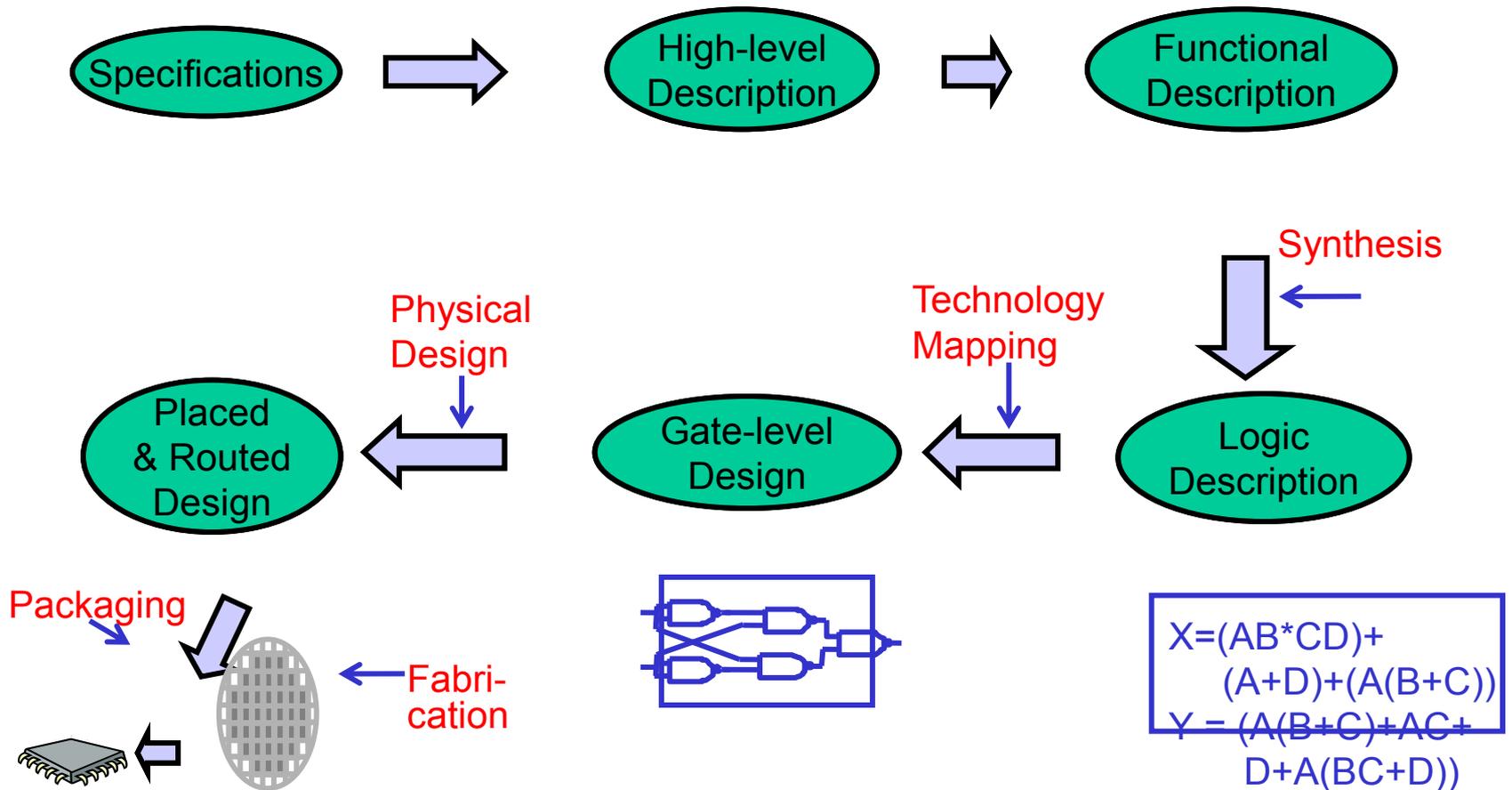
Design flow

- A sequence of design steps in a design methodology
- The design flow can be partially or fully automated
- A set or tools can be used to automate the methodology steps:
 - Software engineering tools,
 - Compilers,
 - Computer-Aided Design tools,
 - etc

IC Design Steps

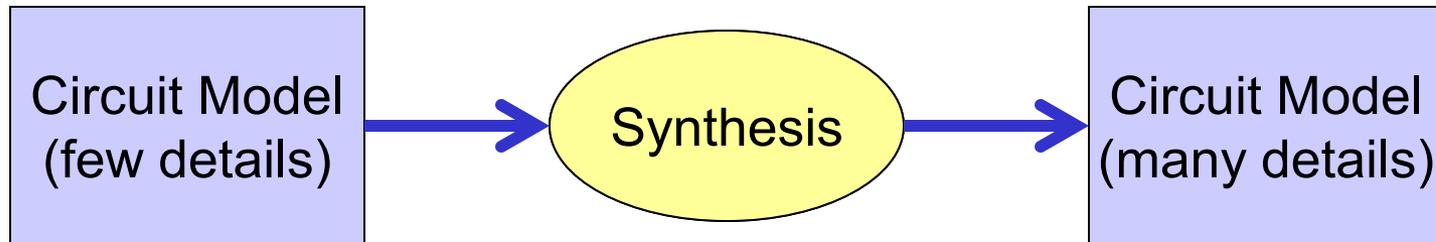


IC Design Steps

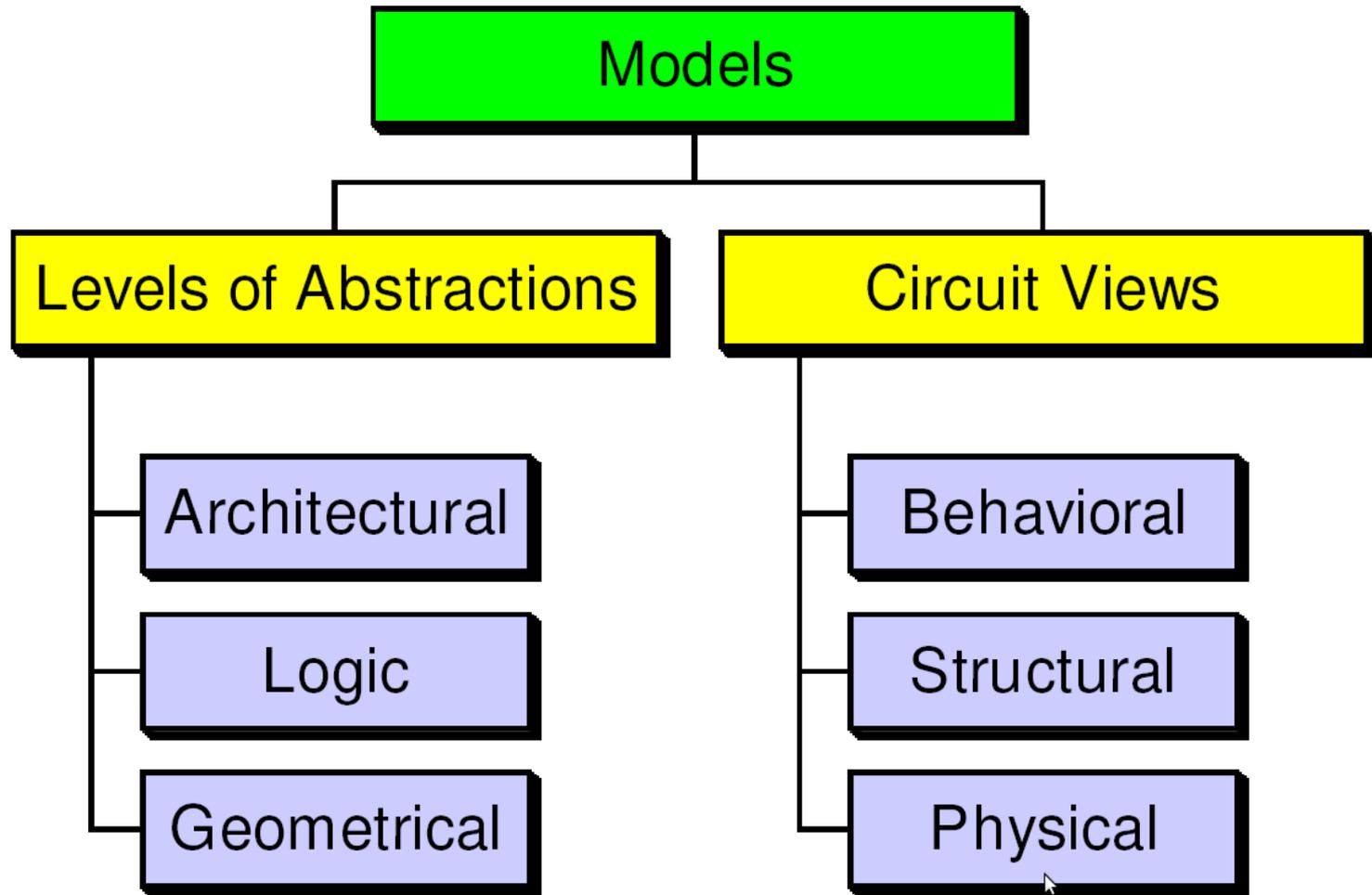


Circuit Models

- A *model of a circuit* is an *abstraction*
 - A representation that shows relevant features without associated details



Model Classification



Views of a Model

■ Behavioral

→ Describe the function of a circuit *regardless* of its implementation

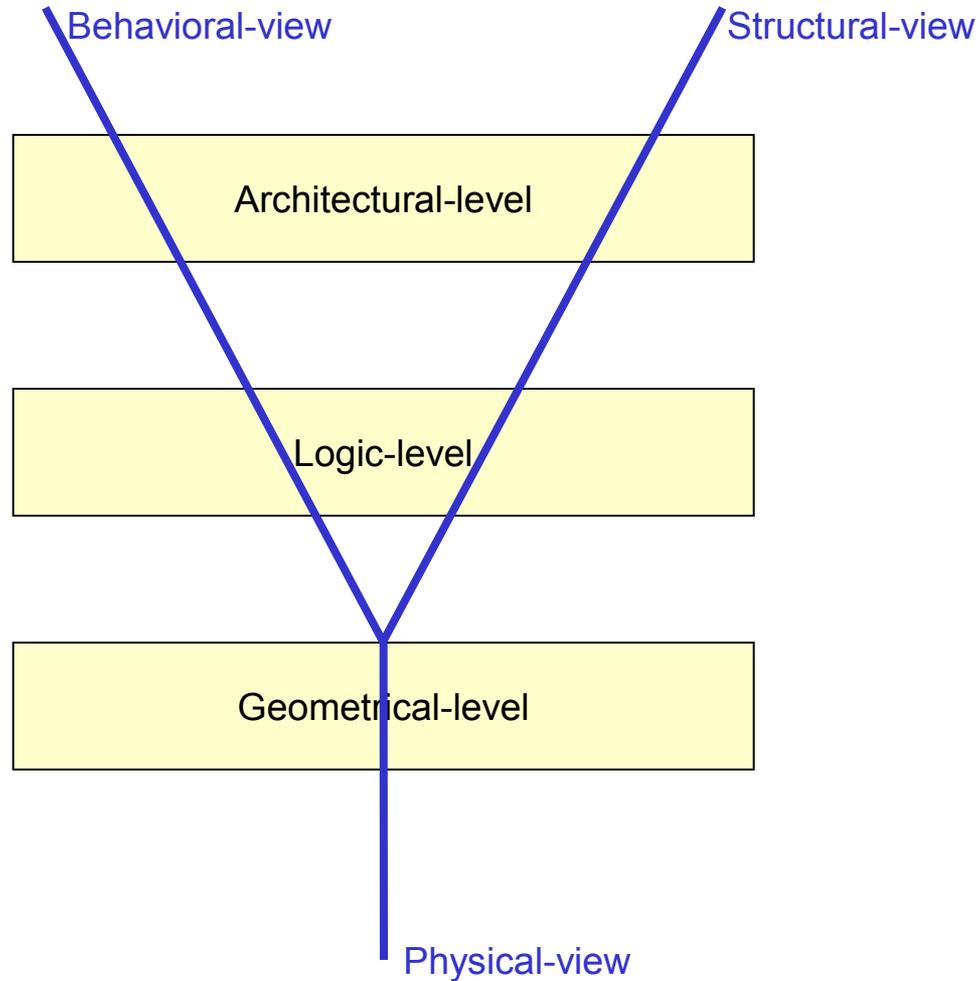
■ Structural

→ Describe a model as an *interconnection* of components

■ Physical

→ Relate to the *physical object* (e.g., transistors) of a design

The Y-chart



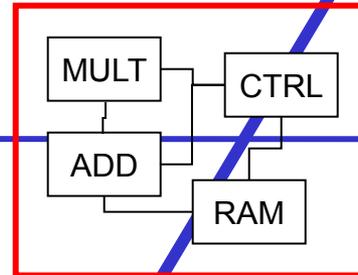
Gajski and Kuhn's Y-chart
(Silicon Compilers, Addison-Wesley, 1987)

The Y-chart

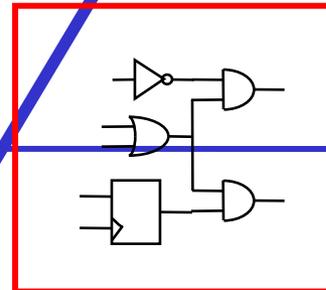
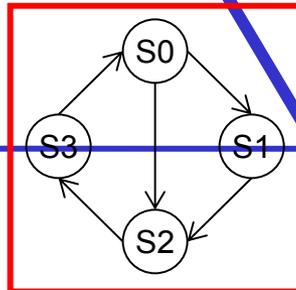
Behavioral-view

```
...  
PC = PC + 1;  
Fetch(PC);  
Decode(Inst);  
...
```

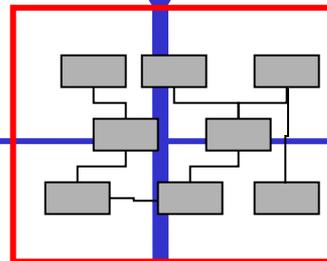
Structural-view



Architectural level



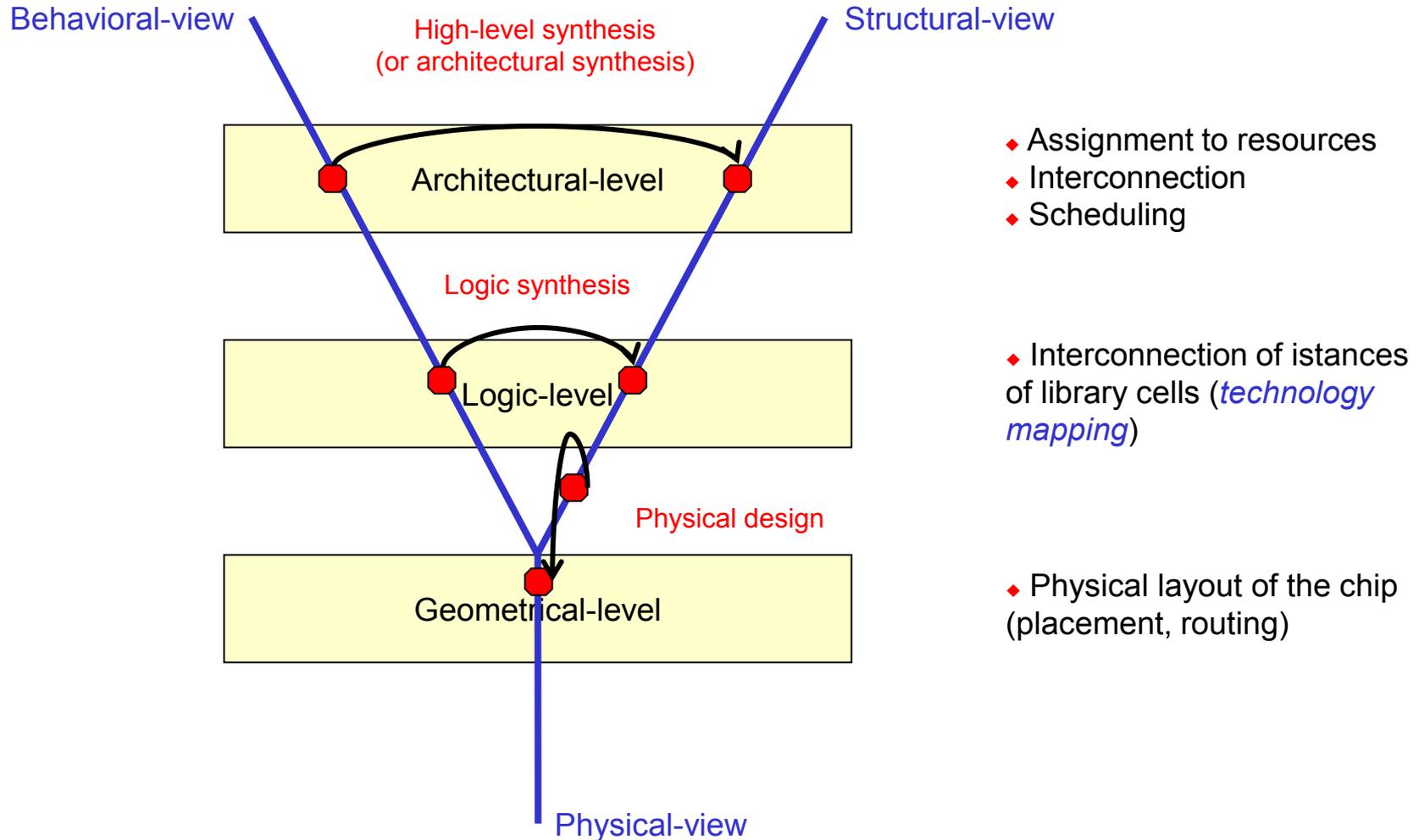
Logic level



Geometrical level

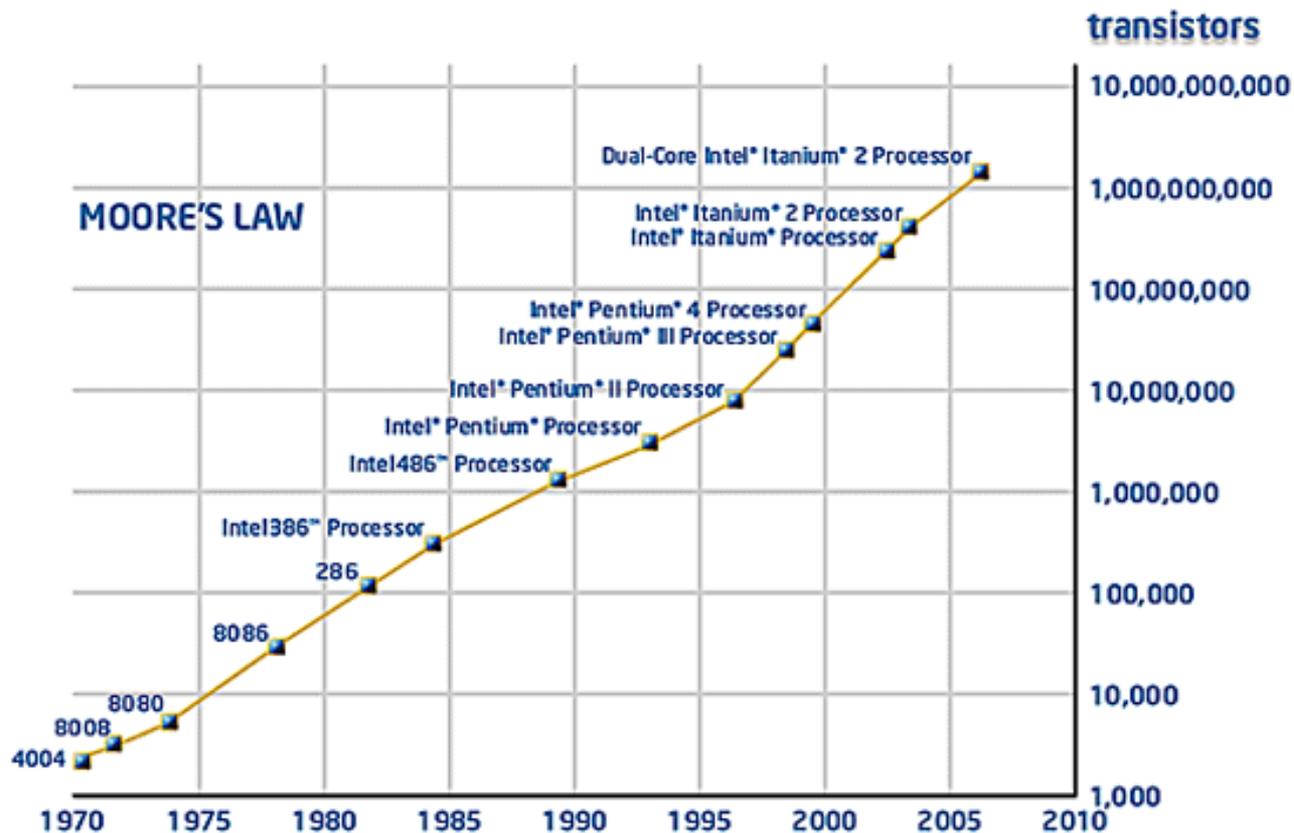
Physical-view

Synthesis



Moore's Law

■ *Gordon Moore* predicted in 1965 that the number of transistors that can be integrated on a die would **double every 18 months**.



Device Complexity

- Exponential increase in device complexity

- Increasing with Moore's law (or faster)!

- Require exponential increases in design productivity

We have exponentially more transistors!

Heterogeneity on Chip

- Greater diversity of on chip elements

- Processors

- Software

- Memory

- Analog

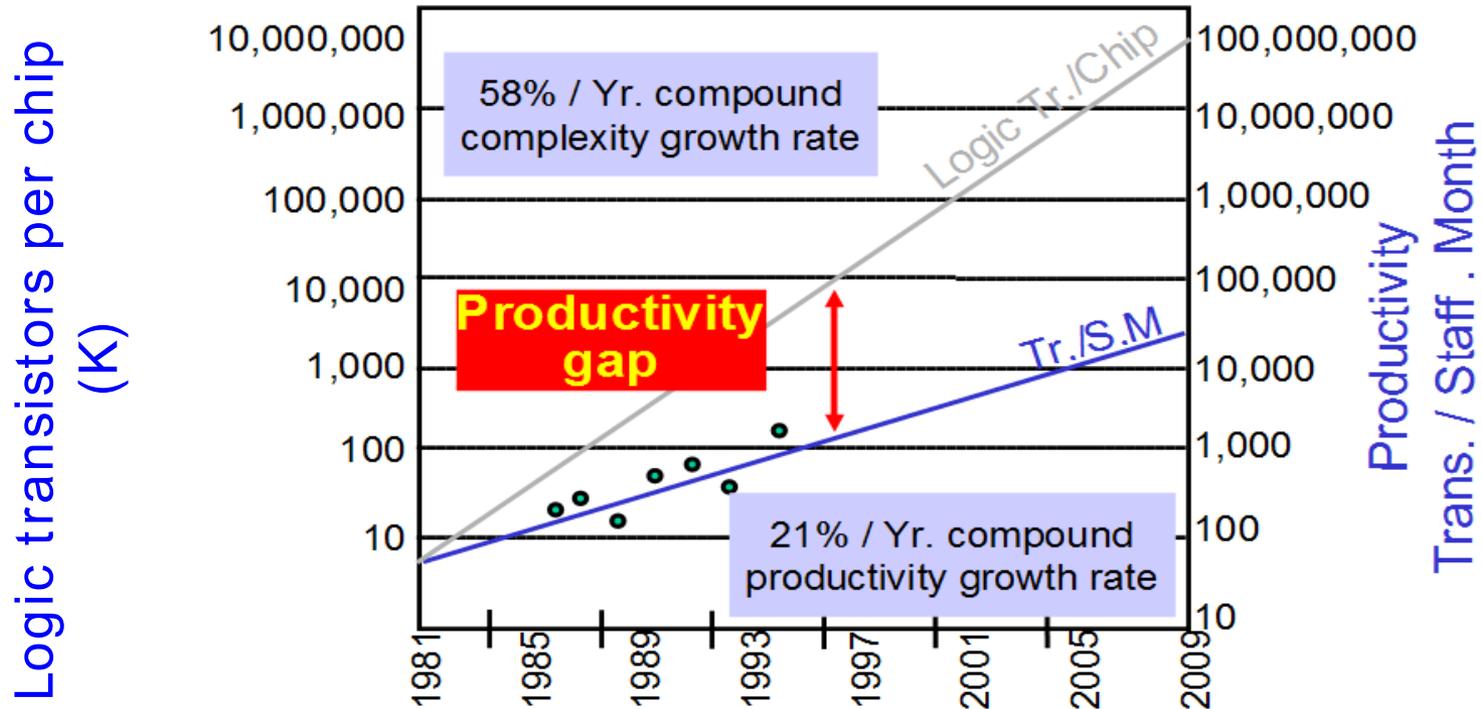
More transistors doing different things!

Stronger Market Pressures



- Time-to-market
 - Decreasing design window
 - Less tolerance for design revisions

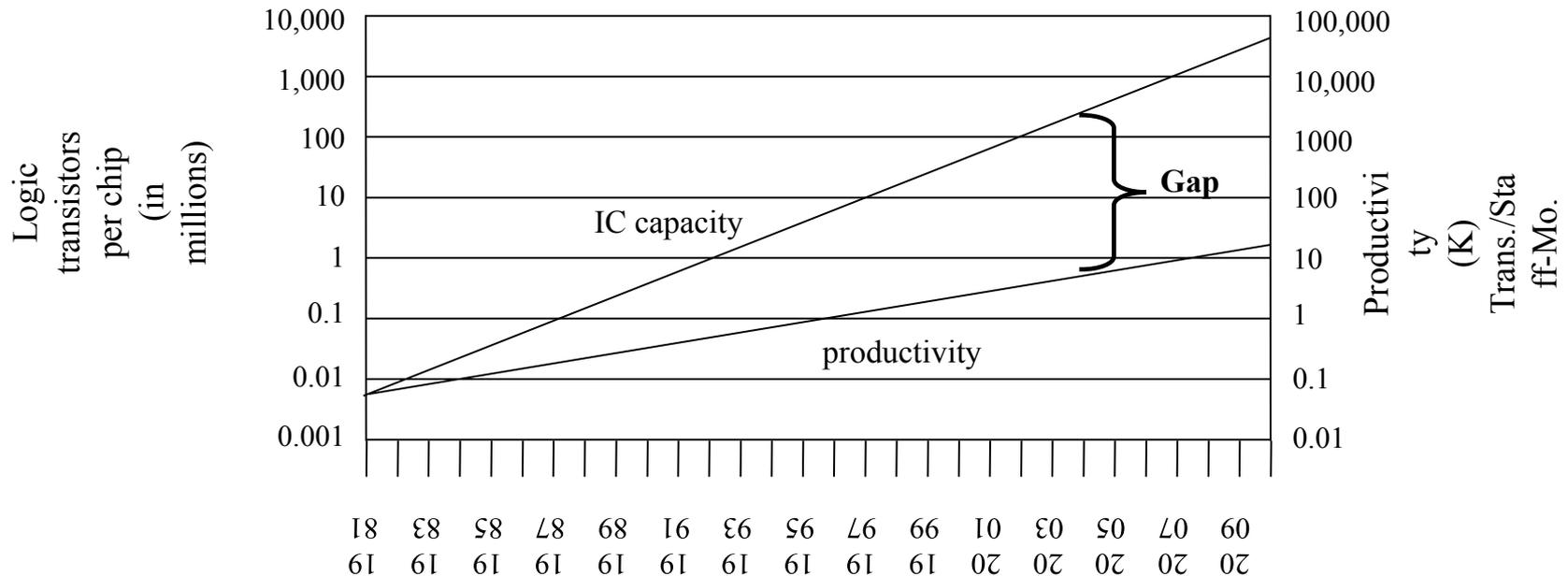
Design productivity gap



Role of EDA: close the productivity gap

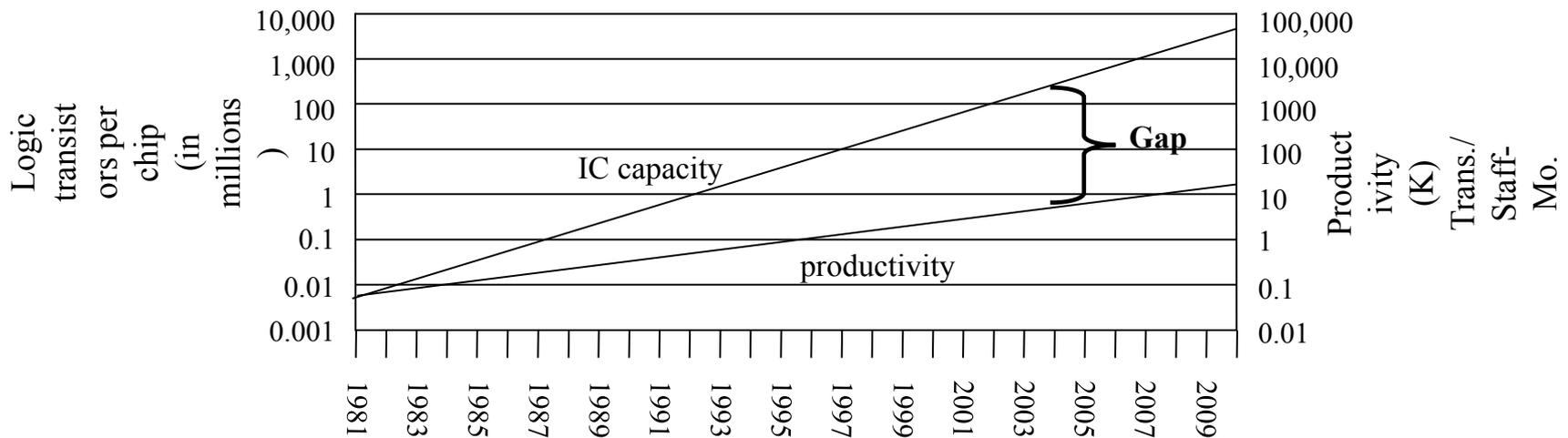
Design productivity gap

- While designer productivity has grown at an impressive rate over the past decades, the rate of improvement has not kept pace with chip capacity



Design productivity gap

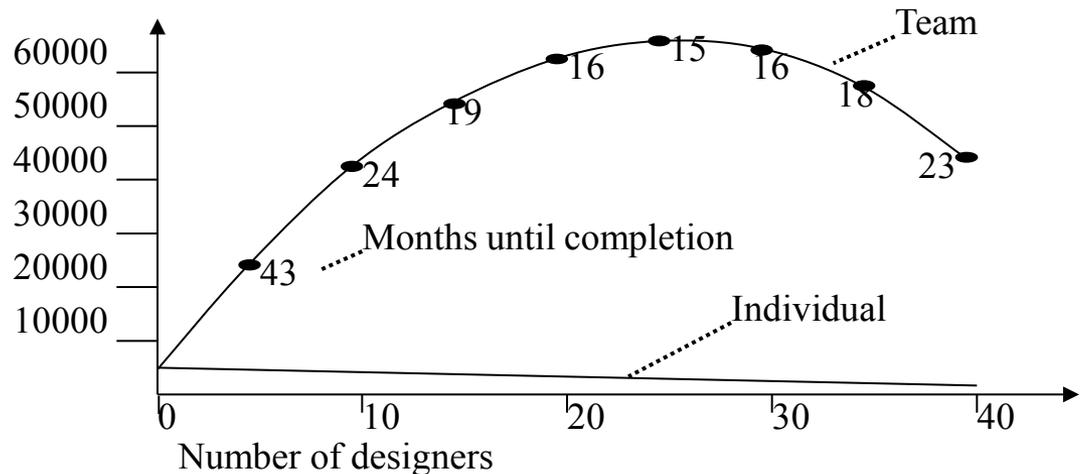
- 1981 leading edge chip required 100 designer months
 - ▣ 10,000 transistors / 100 transistors/month
- 2002 leading edge chip requires 30,000 designer months
 - ▣ 150,000,000 / 5000 transistors/month
- Designer cost increase from \$1M to \$300M



The mythical man-month

- The situation is even worse than the productivity gap indicates
- In theory, adding designers to team reduces project completion time
- In reality, productivity per designer decreases due to complexities of team management and communication
- In the software community, known as “the mythical man-month” (Brooks 1975)
- At some point, can actually lengthen project completion time! (“Too many cooks”)

- 1M transistors, 1 designer=5000 trans/month
- Each additional designer reduces for 100 trans/month
- So 2 designers produce 4900 trans/month each



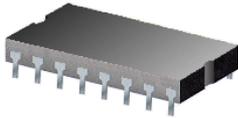
Managing the design productivity crisis

- IP (Intellectual Property) Reuse
 - ❑ Assembly of predesigned Intellectual
 - ❑ Property components, often from external vendors
 - ❑ Soft and Hard IPs
- System-Level Design and verification
 - ❑ Rather than at the RTL or gate-level
 - ❑ Focus on Interface and Communication

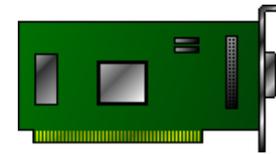
Evolution of Design Methodology

■ We are now entering the era of *block-based design*

ASIC/ASSP
Design

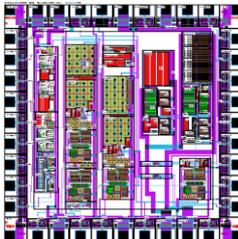


Yesterday
Bus Standards,
Predictable, Preverified

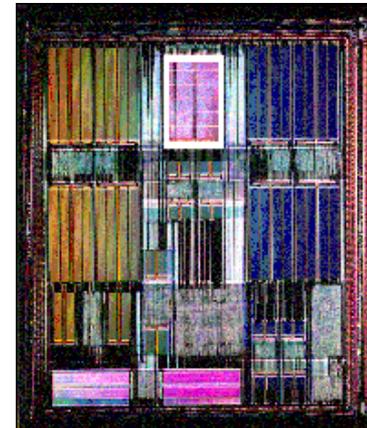


System-Board
Integration

IP/Block
Authoring



Today
VSI Compatible Standards,
Predictable, Preverified



System-Chip
Integration

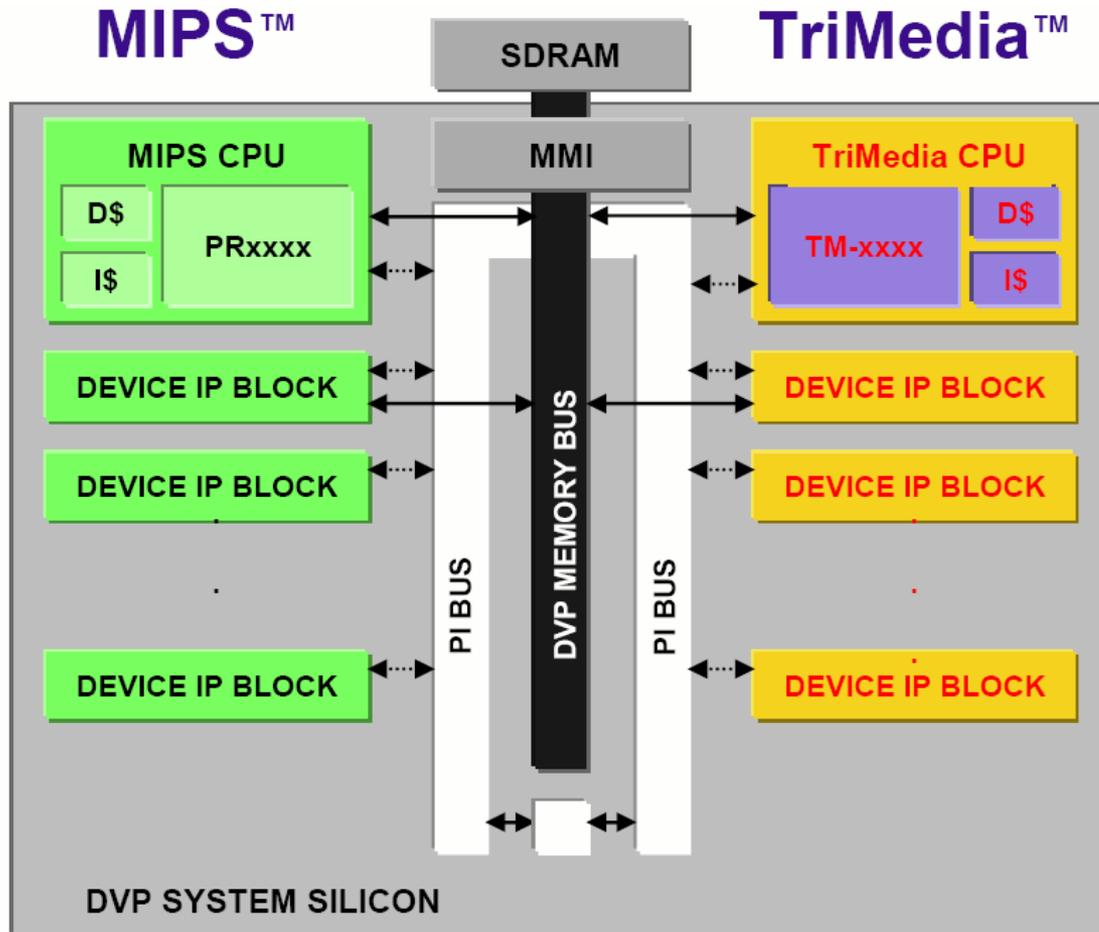
Evolution of SoC Platforms

General-purpose Scalable RISC Processor

- 50 to 300+ MHz
- 32-bit or 64-bit

Library of Device IP Blocks

- Image coprocessors
- DSPs
- UART
- 1394
- USB



- ## Scalable VLIW Media Processor:
- 100 to 300+ MHz
 - 32-bit or 64-bit

- ## Nexperia™ System Buses
- 32-128 bit

2 Cores: Philips' Nexperia PNX8850 SoC platform for High-end digital video (2001)

What's Happening in SoCs?

■ Technology: no slow-down in sight!

Faster and smaller transistors: 90 → 65 → 45 → 32 → 22 nm

→ ... but slower wires, lower voltage, more noise!

✓ 80% or more of the delay of critical paths will be due to interconnects

■ Design complexity: from 2 to 10 to 100 cores!

→ Design reuse is essential

→ ...but differentiation/innovation is key for winning on the market!

■ Performance and power:

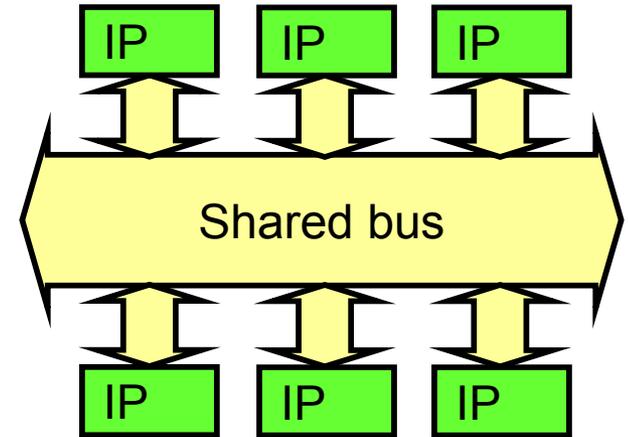
→ Performance requirements keep going up

→ ...but power budgets don't!

Communication Architectures

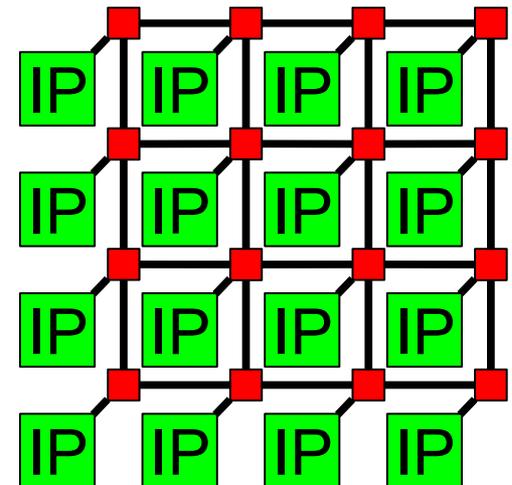
■ Shared bus

- Low area
- Poor scalability
- High energy consumption

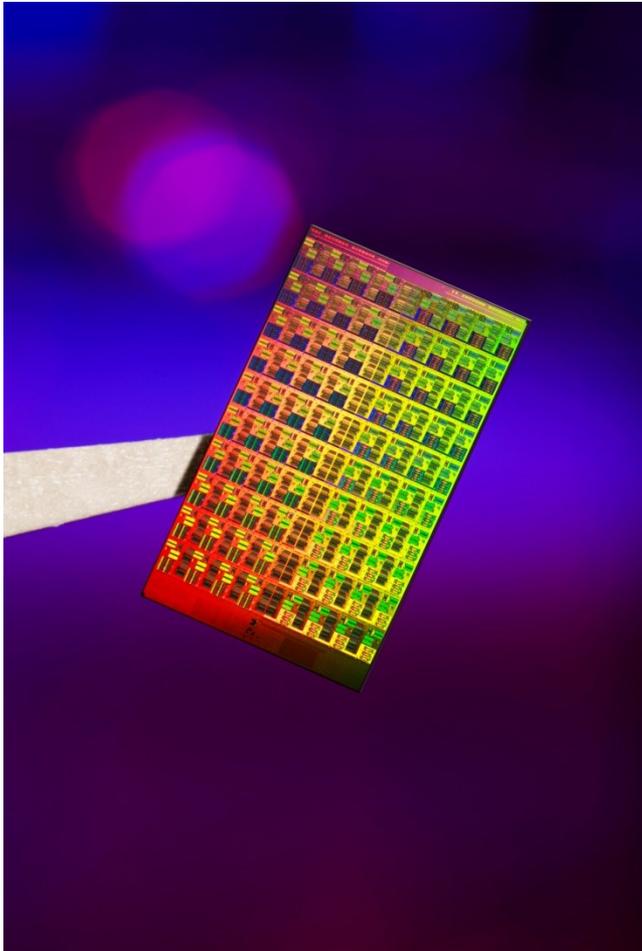


■ Network-on-Chip

- Scalability and modularity
- Low energy consumption
- Increase of design complexity



Intel's Teraflops



- 100 Million transistors
- 80 cores, 160 FP engines
- Teraflops perf. @ 62 Watts
- On-die mesh network
- Power aware design