VLSI Design Methodology

电子科学与技术系：陈 松

2015年9月17日

Course Webpage:

http://staff.ustc.edu.cn/~songch/da-ug.htm

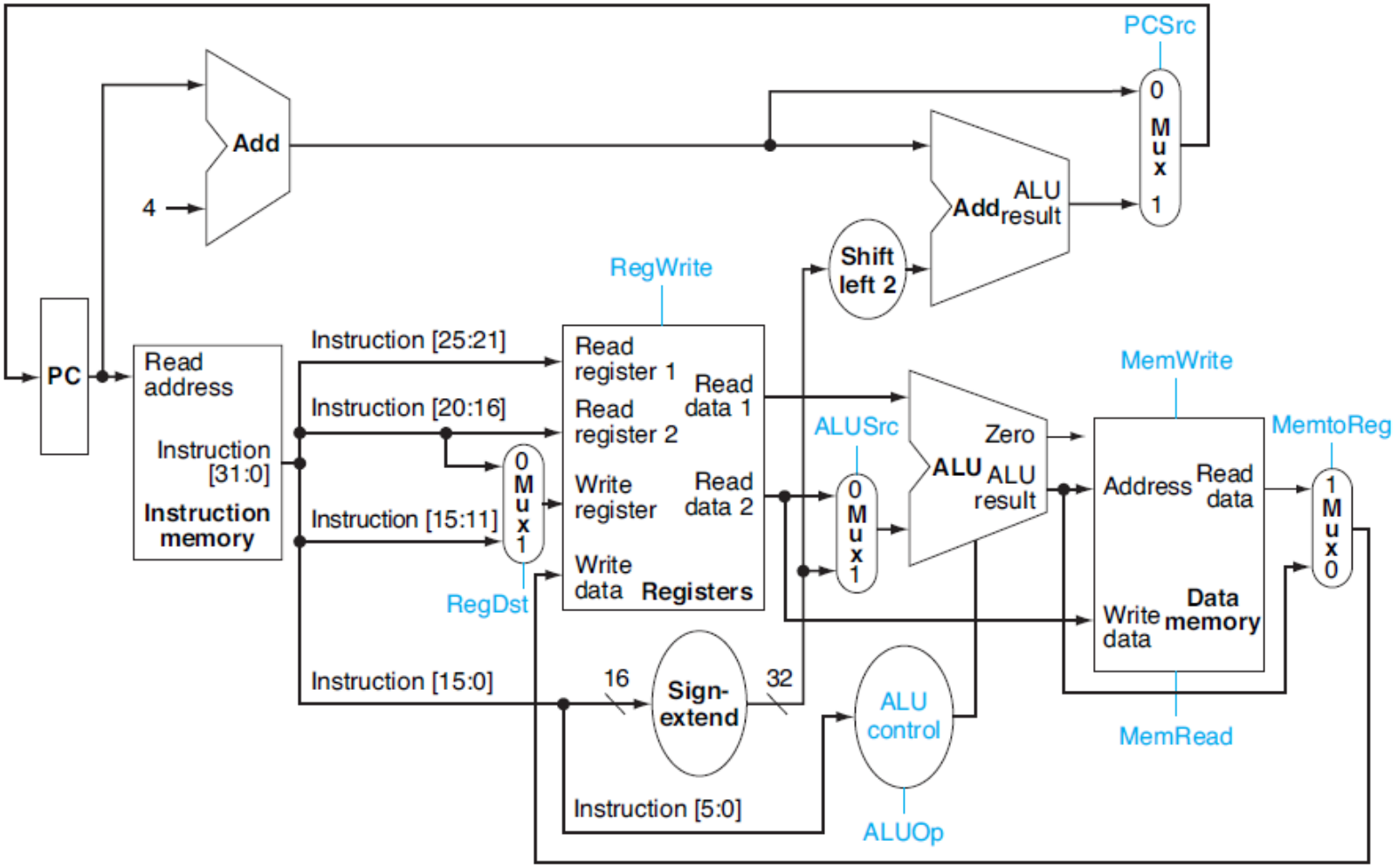
Outline

- Structured Design
  - Hierarchy
  - Regularity
  - Modularity
  - Locality
- Design Partitioning
- Design Methods/Styles
- Design Flow (Cell-based Design)

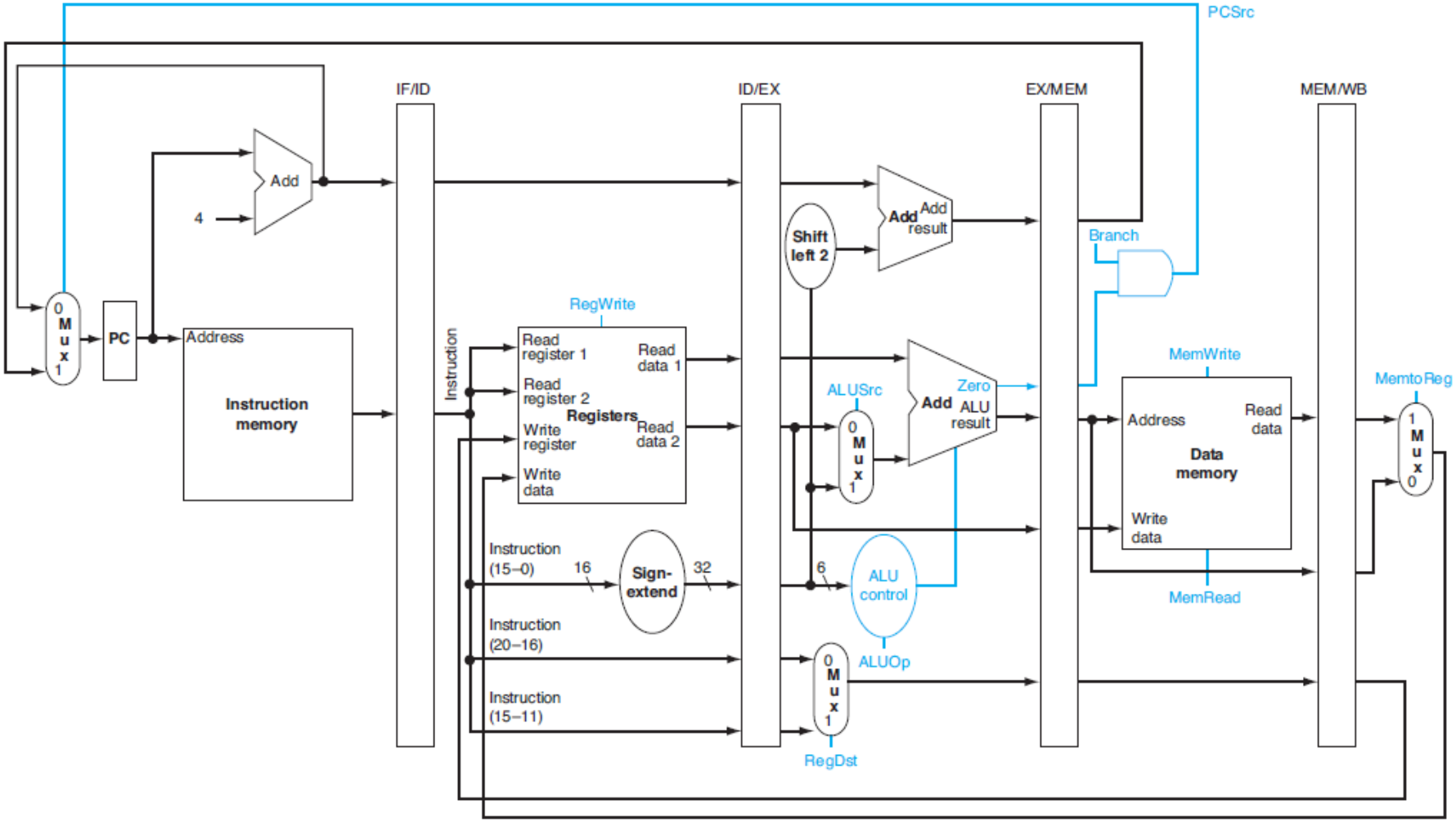
Modern VLSI Design(SoC)

- Coping with complexity
  - How to design System-on-Chips?
    - Many millions (even billions!) of transistors
    - Tens to hundreds of engineers
  - Structured Design
  - Design Partitioning
- Not designing an individual transistor

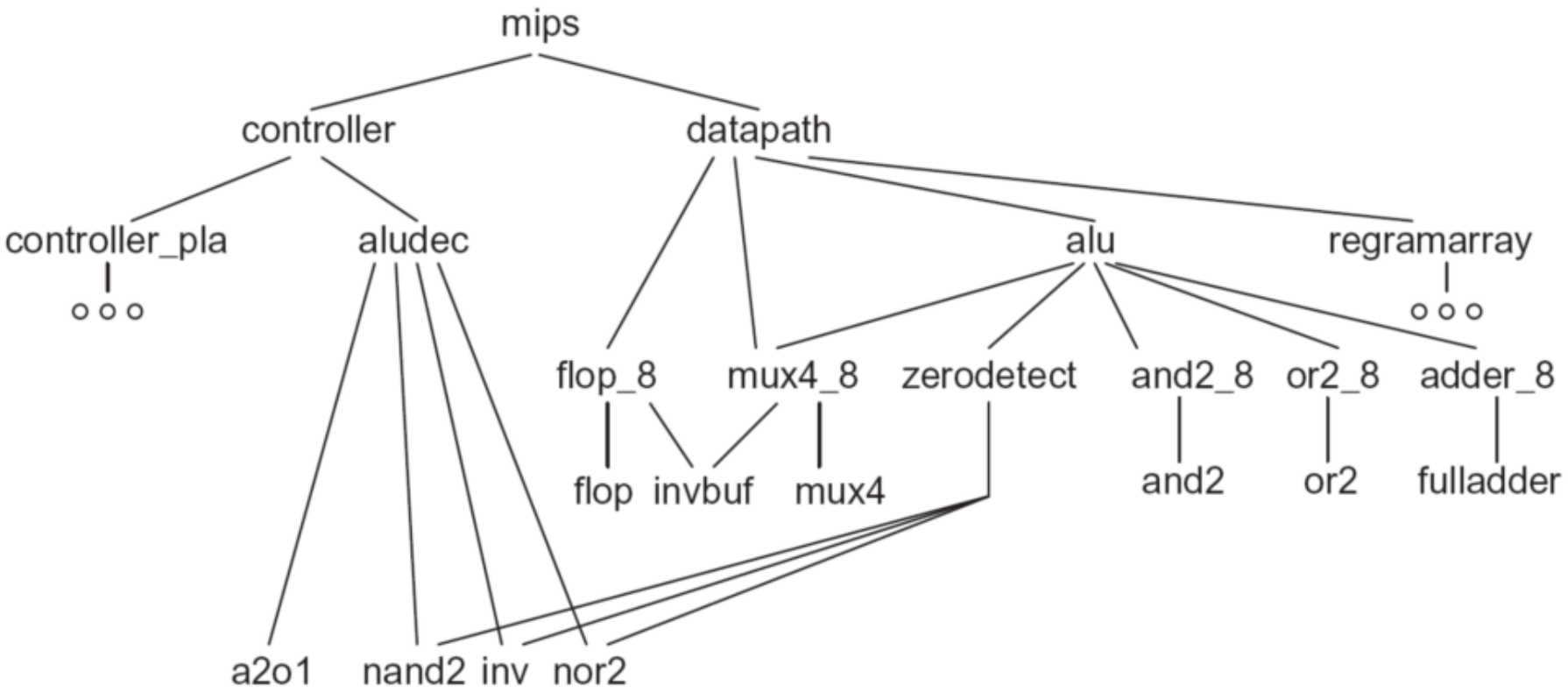
A Simple MIPS Architecture



Pipelined MIPS Architecture

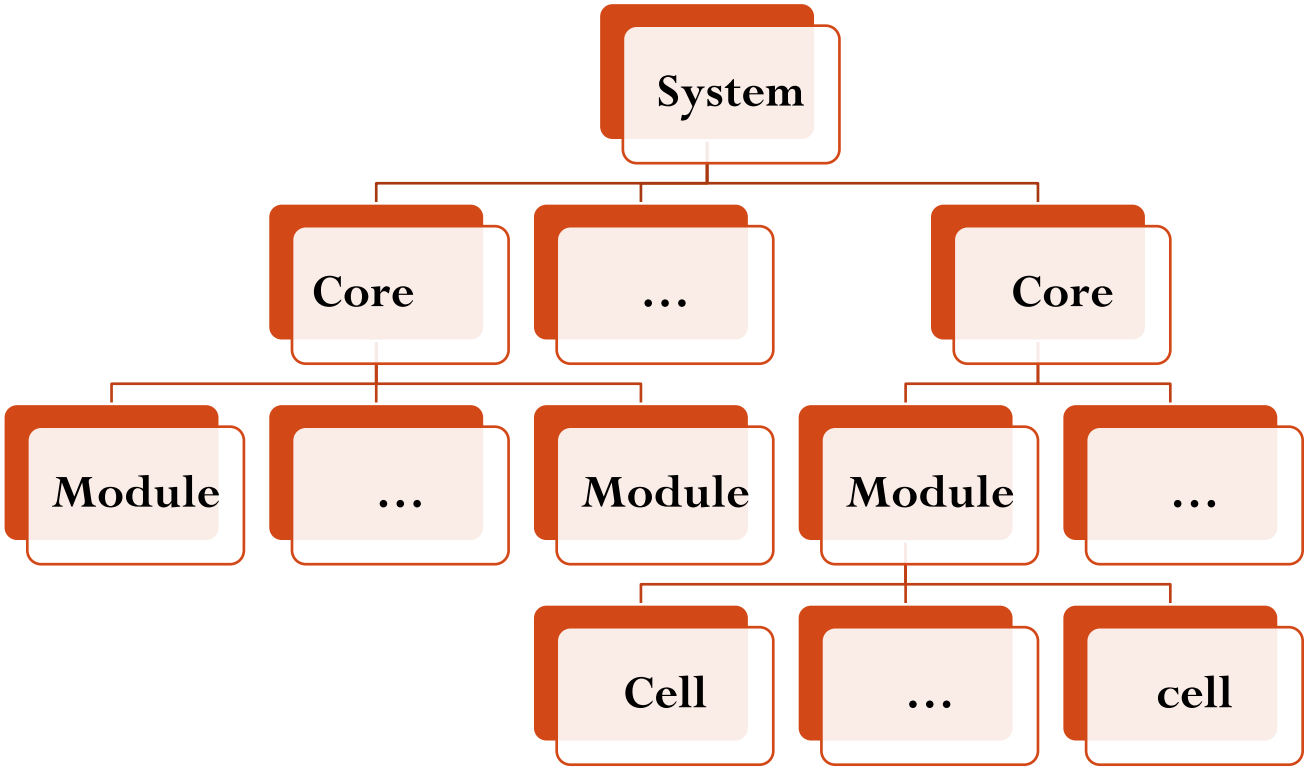


A MIPS CPU Design Hierarchy



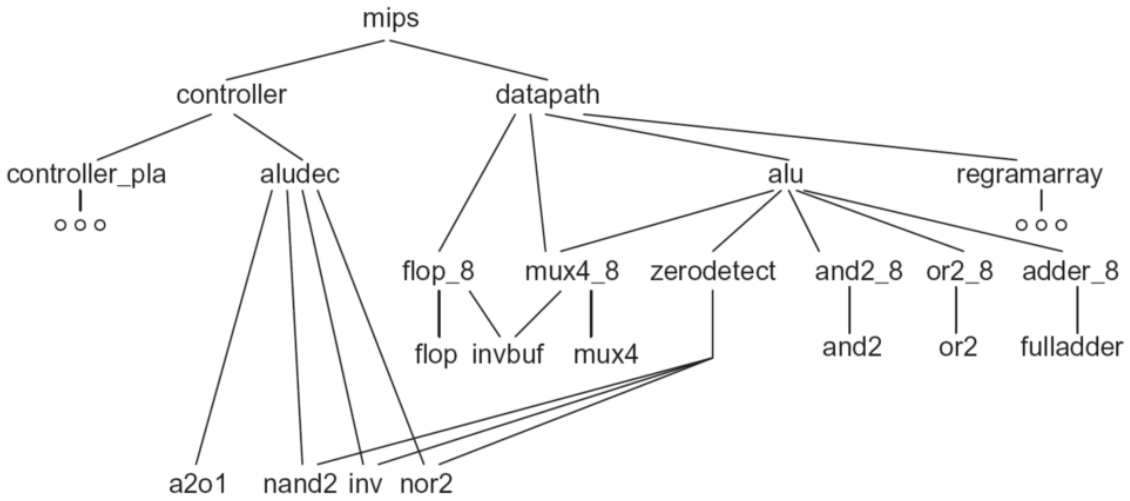
Structured design

- Managing system complexity
  - Hierarchy



Structured design (Cont')

- Regularity : few modules
  - Circuit level: uniformly sized transistors
  - Gate level: a finite library of fixed-height, variable-length logic gates
  - Logic level: parameterized RAMs and ROMs
  - Architecture level: multiple identical processors



Structured design (Cont')

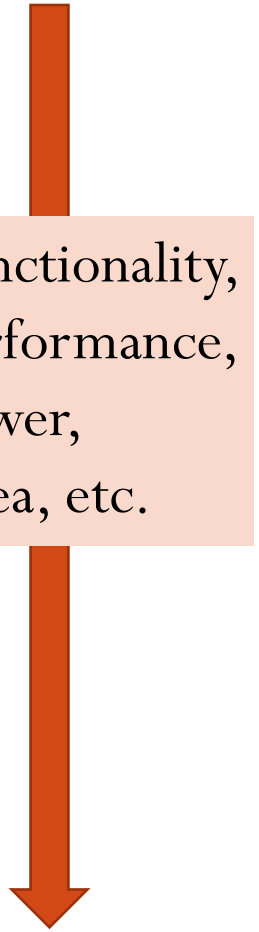
- Modularity: well defined modules, IP sources
  - Clear defined behavioral, structural, physical interface
    - Function
    - Name
    - Signal type
    - Electrical and Timing constraints of ports
    - Position, connection layer, wire width
- Locality
  - Local scoping: reducing the global variables
  - Temporal, Physical

Structured design -Summary

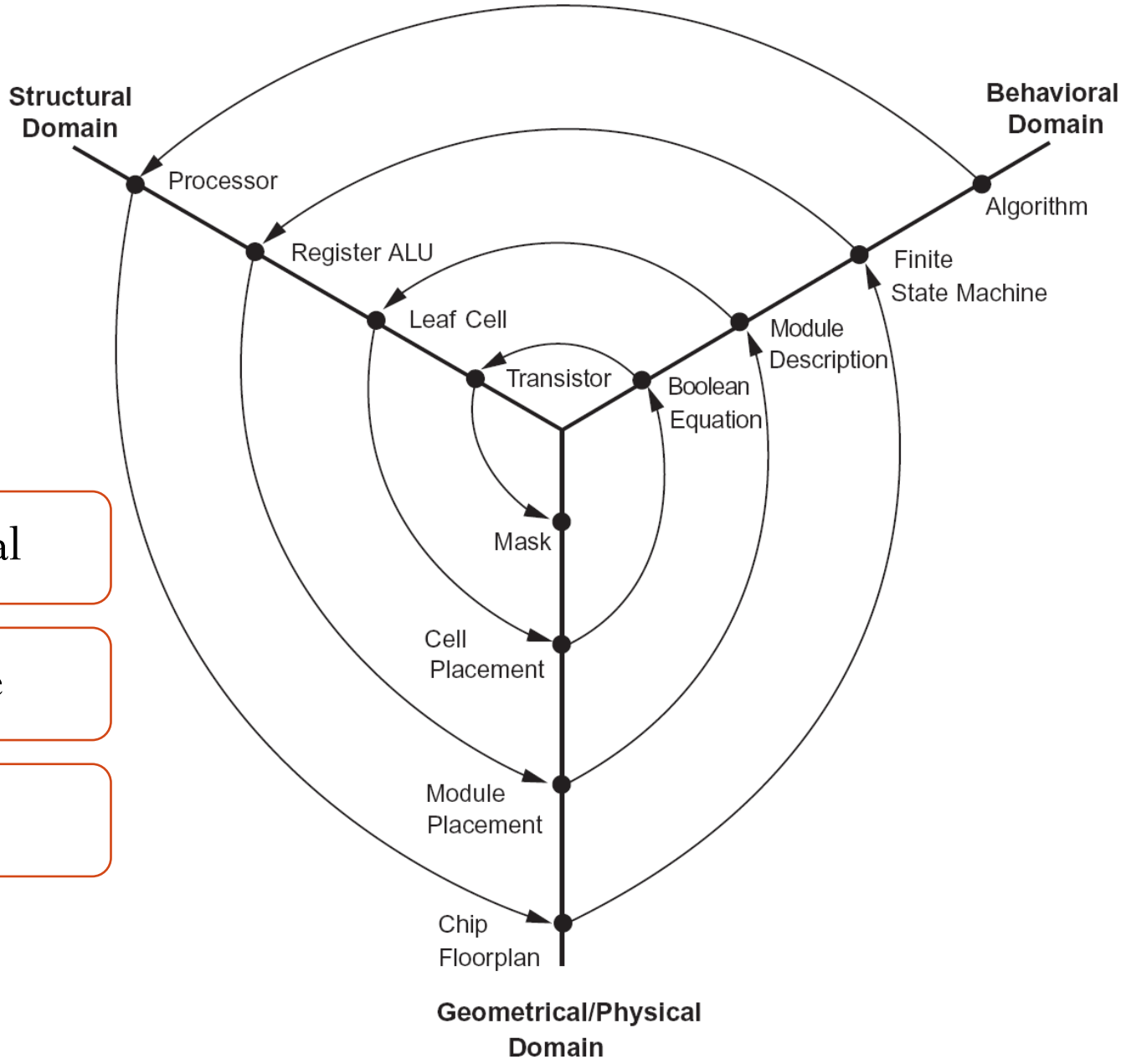
Design Principle	Software	Hardware
Hierarchy	Subroutines, libraries	Modules
Regularity	Iteration, code sharing, object-oriented procedures	Datapaths, module reuse, regular arrays, gate arrays, standard cells
Modularity	Well-defined subroutine interface	Well-defined module interaces, timing and loading data for modules, registered inputs and outputs
Locality	Local scoping, no global variables	Local connections through floorplanning

Design Partitioning - Design Abstractions

- **Architecture**: User's perspective, what does it do?
  - Instruction set, registers, memory model
  - MIPS, x86, Alpha, PIC, ARM, …
- **Microarchitecture**
  - Single cycle, multicycle, pipelined, superscalar?
  - 80386, 80486, Pentium, Pentium II, Pentium III, AMD Athlon, …
- **Logic**: how are functional blocks constructed
  - Ripple carry, carry lookahead, carry select adders
- **Circuit**: how are transistors used
  - **Complementary CMOS (CMOS)**, pass transistors, domino
- **Physical**: chip layout
  - Datapaths, memories, random logic



Design Partitioning



Design Flow:

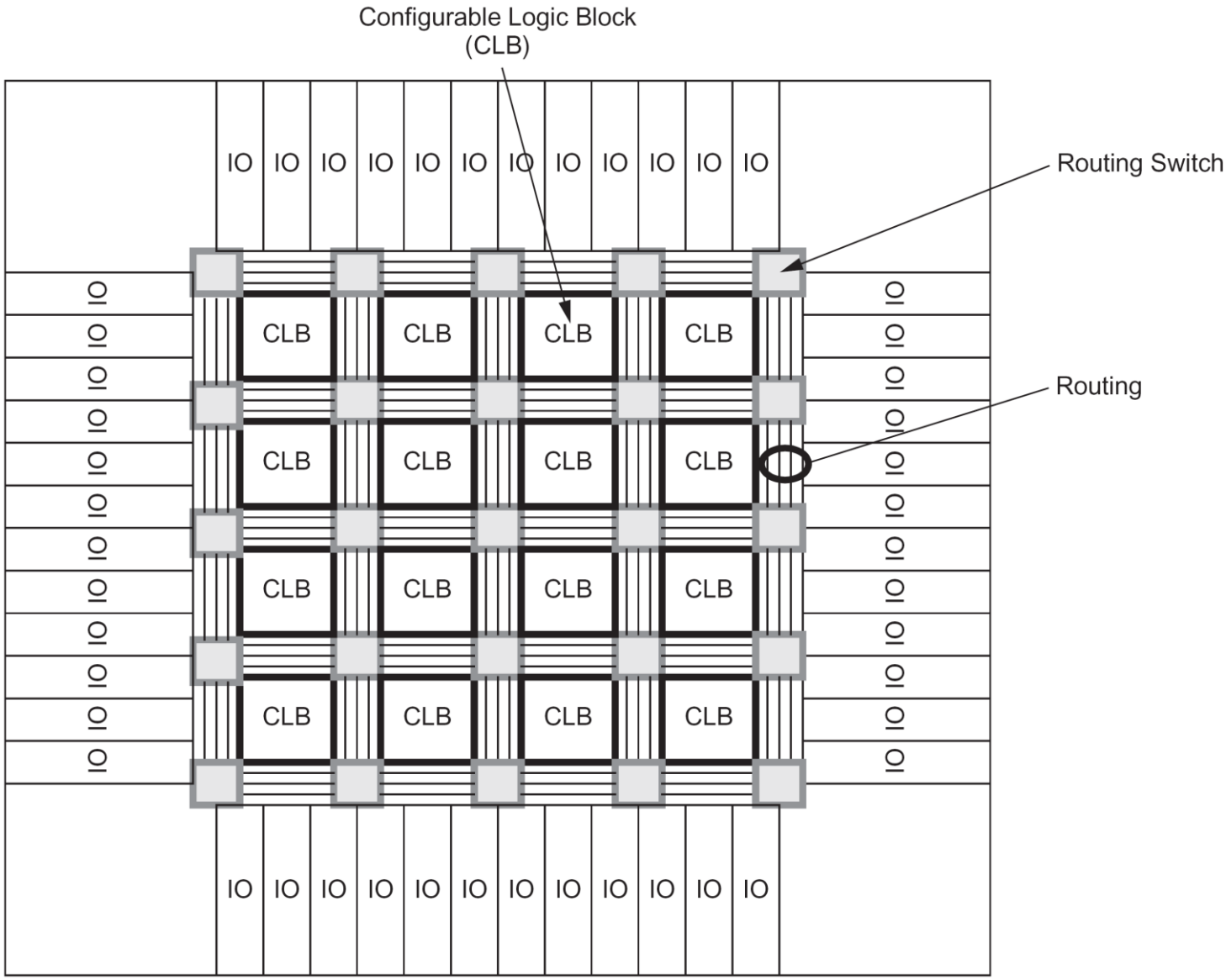
- Behavioral
- Structure
- Physical

Design Methods

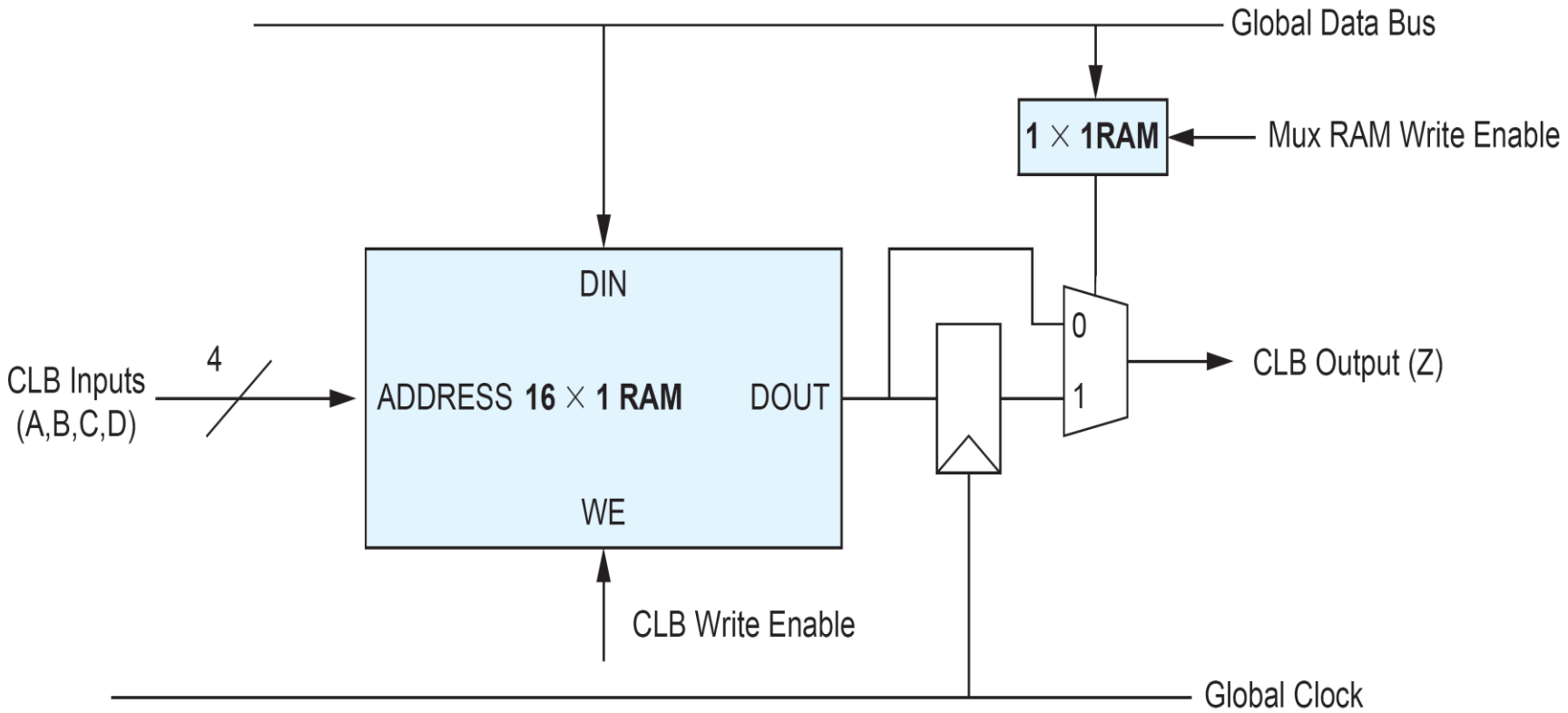
- Microprocessor/DSP
- Programmable Logic
  - **Programmable Logic Devices**
  - **Field-Programmable Gate Arrays (FPGAs)**
- Gate Array and Sea of Gates Design
- **Cell-Based Design**
- Full Custom Design
- *Platform-Based Design-System on a Chip*

Select the right solution: Costs, Capabilities, Limitations,

FPGA


Configurable Logic Block
(CLB)
Routing Switch
Routing
IO
CLB

Simple FPGA Logic Cell



RAM CLB Functions

Address	*ABCD*	$A \cdot B \cdot C \cdot D$	$\sim A$	SUM(*A*,*B*,*C*)
0	0000	0	1	0
1	1000	0	0	1
2	0100	0	1	1
3	1100	0	0	0
4	0010	0	1	1
5	1010	0	0	0
6	0110	0	1	0
7	1110	0	0	1
8	0001	0	1	0
9	1001	0	0	1
10	0101	0	1	1
11	1101	0	0	0
12	0011	0	1	1
13	1011	0	0	0
14	0111	0	1	0
15	1111	1	0	1

FPGA – A Summary

- Chip is still there
  - Millions of logic gate
  - >10GHz I/Os
- Embedded microprocessor cores and DSP accelerator hardware
- Low up-front cost and ease of correcting design errors
  - Best choice now for many low- to medium- volume custom logic applications
- Implement FPGA blocks on any CMOS chip
  - to provide programmability at the gate level

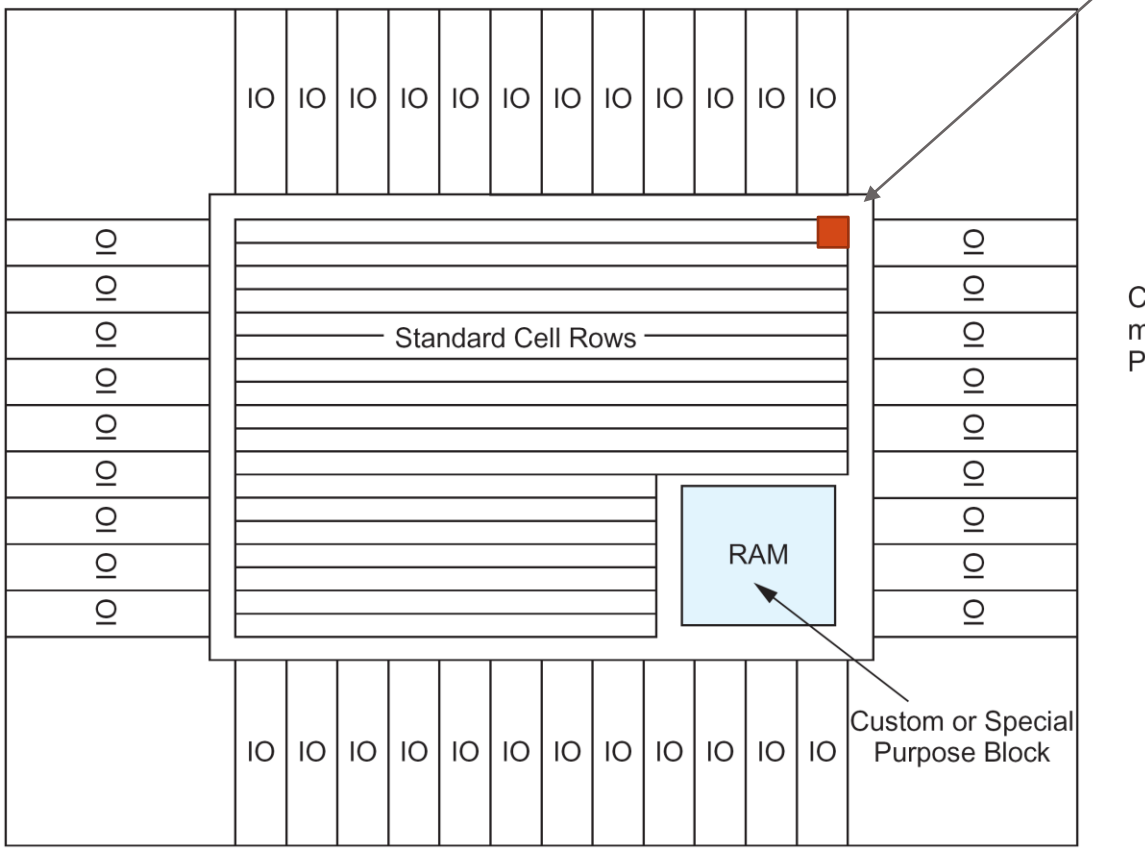
Cell-based Design

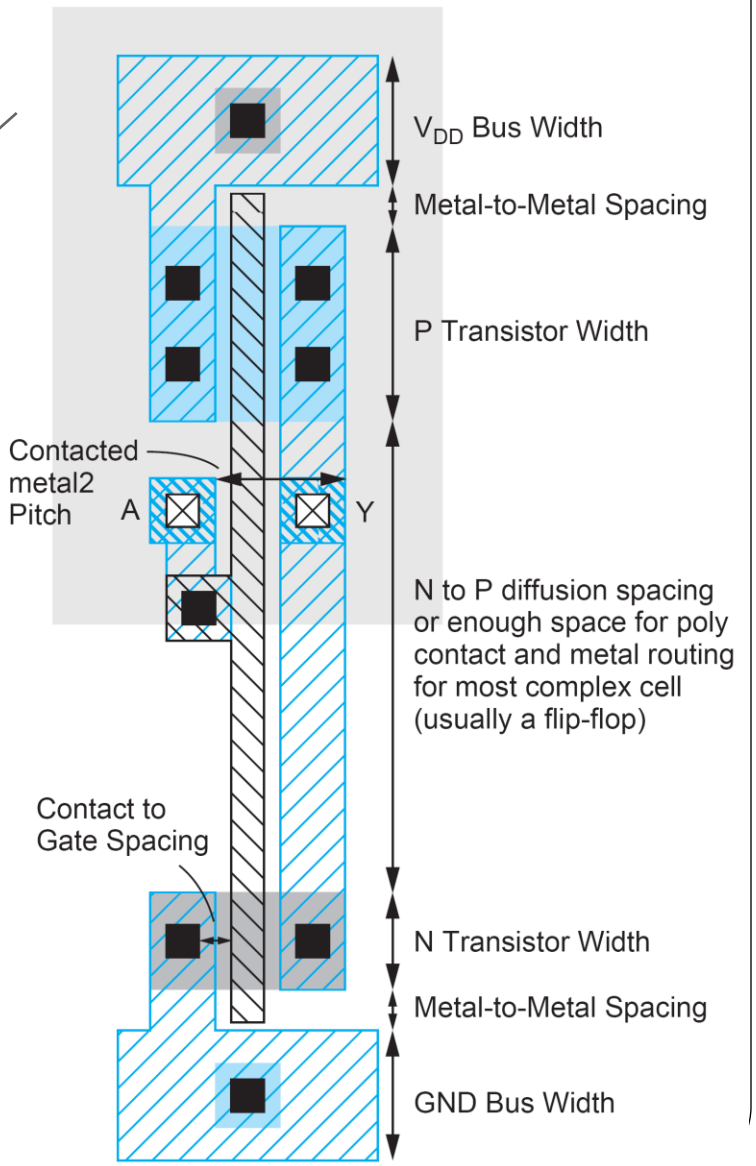
- Use a standard cell library
  - Small-scale integration logic
    - NAND, NOR, XOR, AOI, OAI, inverters, buffers, registers
    - Memories (RAM, ROM, CAM, register files)
    - System-level modules such as processors, protocol processors, serial interfaces, and bus interfaces
    - Possibility of mixed-signal and RF modules
  - Minimum Scale Integration: adders, multipliers, parity blocks

Typical Standard Cell Library

Gate Type	Variations	Options
Inverter/Buffer /Tristate Buffer		Wide range of power options, $2^{\wedge 0\text{-}6}$x minimum size inverter
NAND/AND	2-8 inputs	High, normal, low power
NOR/OR	2-8 inputs	High, normal, low power
XOR/XNOR		High, normal, low power
AOI/OAI	21,22	High, normal, low power
Multiplexers	Inverting/noninverting	High, normal, low power
Latches		High, normal, low power
Flip-Flops	D, with and without synch/asynch set and reset, scan	High, normal, low power
I/O Pads	Input, output, tristate, bidirectional, boundary scan, slew rate liminted, crystal oscillator	Various drive levels (1-16mA) and logic levels

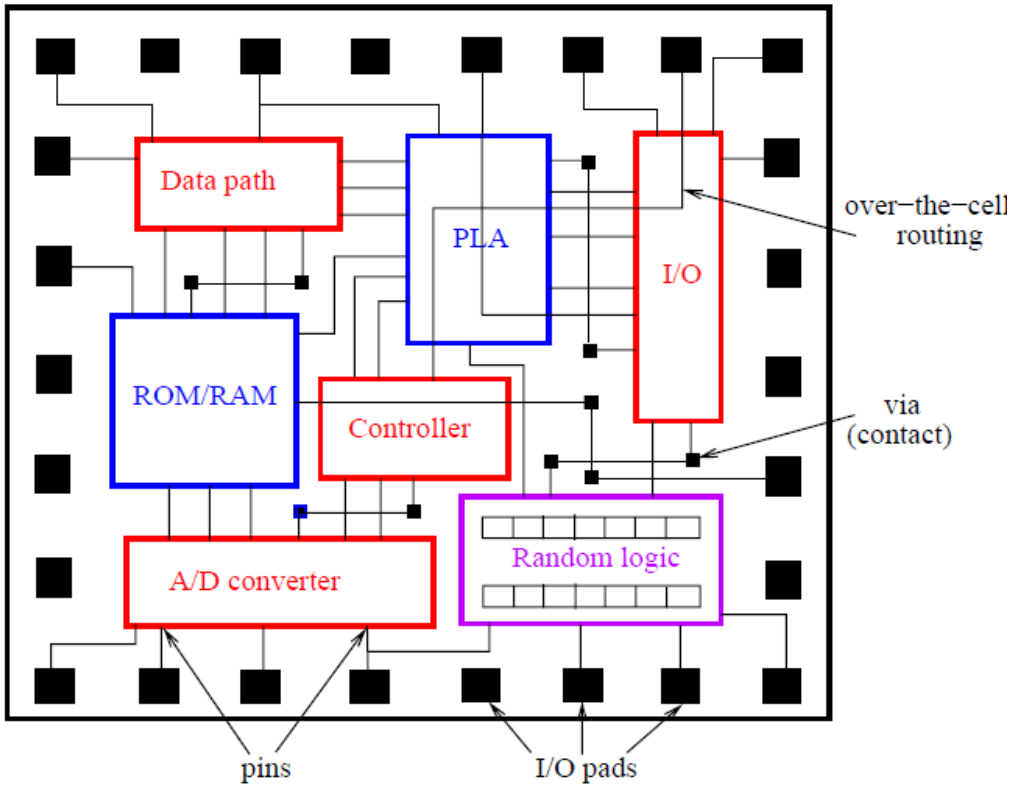
Standard cell layout





Full Custom Design

- Custom-designed microprocessors
  - >2GHz
- Synthesized ASIC
  - 200-350MHz
  - 3-7x more power
  - 2.6x if better tools and cell libraries.



Design Styles

- Custom and Semi-Custom
  - Hand-drawn transistors (+ some standard cells)
  - High volume, best possible performance: used for most advanced microprocessors
- Field-Programmable Gate Arrays
  - Transistors have been there, you configure the functions and connections
  - Prototyping
  - Low volume, low-moderate performance applications
- Standard-Cell-Based ASICs
  - High volume, moderate performance: Graphics chips, network chips, cell-phone chips

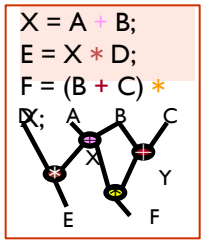
Platform-based Design – System on a Chip

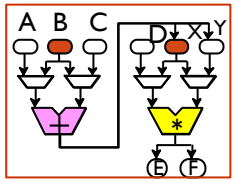
- Use of predefined IP blocks has become commonplace
  - RISC processors, memory, I/O functions, common buses, NoC
  - High-level languages to program
  - Typically, a basic RISC processor, extended with multipliers, floating point units, or specialized DSP units, special hardware assisted instructions, etc.
- What you need do
  - Put the blocks together
  - Design any application-specific blocks
  - Place and route a correctly operational chip

Design Method Summary

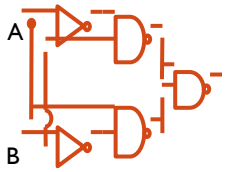
Design Method	Non-Re-curring En-gineering	Unit Cost	Power Dissipation	Complexity of Implementation	Time-Market	Performance	Felxibility
Microprocessor/DSP	Low	Medium	High	Low	Low	Low	High
PLD	Low	Medium	Medium	Low	Low	Medium	Low
FPGA	Low	Medium	Medium		Low	High	High
Cell-based	**High**	**Low**	**Low**	**High**	**High**	**High**	**Low**
Custom Design	High	Low	Low	High	High	Very High	Low
Platform-Based	High	Low	Low	High	High	High	Medium

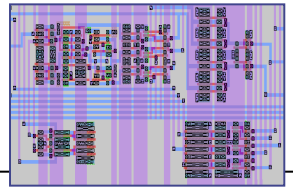
ASIC (专用集成电路)





Adder 1 Multiplier 1
Register 8 Multiplexer 4





System / Behavioral Descriptions

Register Transfer Level Descriptions

Gate Level Netlist

Layout mask

Design Automation

High-level Synthesis
高层次综合
- Scheduling;
- Binding;
- Allocation.

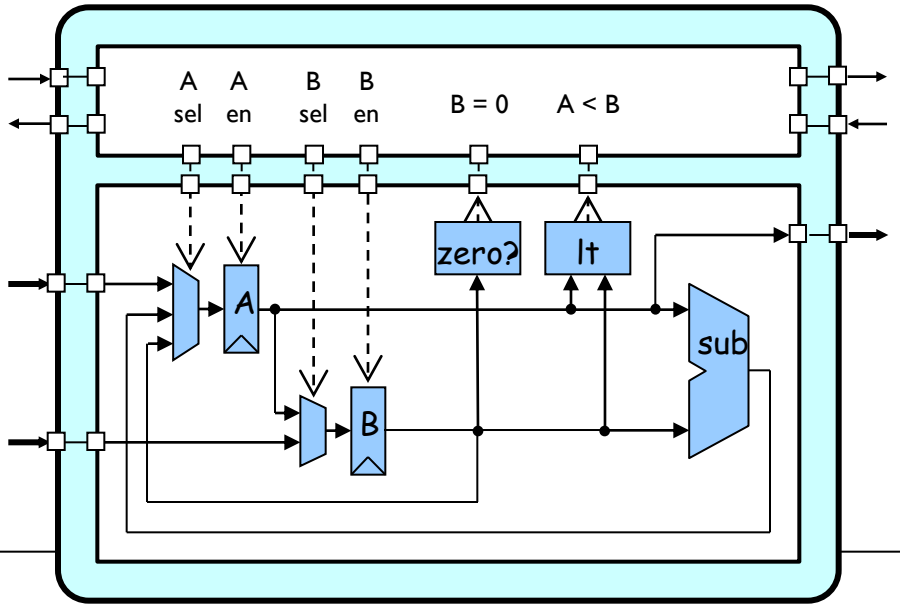
Logic Synthesis
逻辑综合
- Logic Optimization;
- Technology Mapping.

Physical Synthesis
物理综合/芯片布图
- Floorplanning;
- Placement;
- Routing;
- etc.

High-level Synthesis (高层次综合)

- **Transformation from a behavioral specification of a system to its RTL structure specification**
  - E.g., from C/C++ to RTL, Bluespec, Xilinx AutoESL
  - Promote the productivity
  - Essential tasks: scheduling, binding, allocation

```
begin
    done = 0; A = inA; B = inB;
    while ( !done )
    begin
      if ( A < B )
        swap = A;
        A = B; B = swap;
      else if ( B != 0 )
        A = A - B;
      else
        done = 1;
    end
```



Logic Synthesis (逻辑综合)

- Transformation from a Register-Transfer-Level (RTL) description to a gate-level netlists
  - Hardware Description Language
    - Verilog HDL, VHDL
  - Logic Optimization
  - Technology Mapping

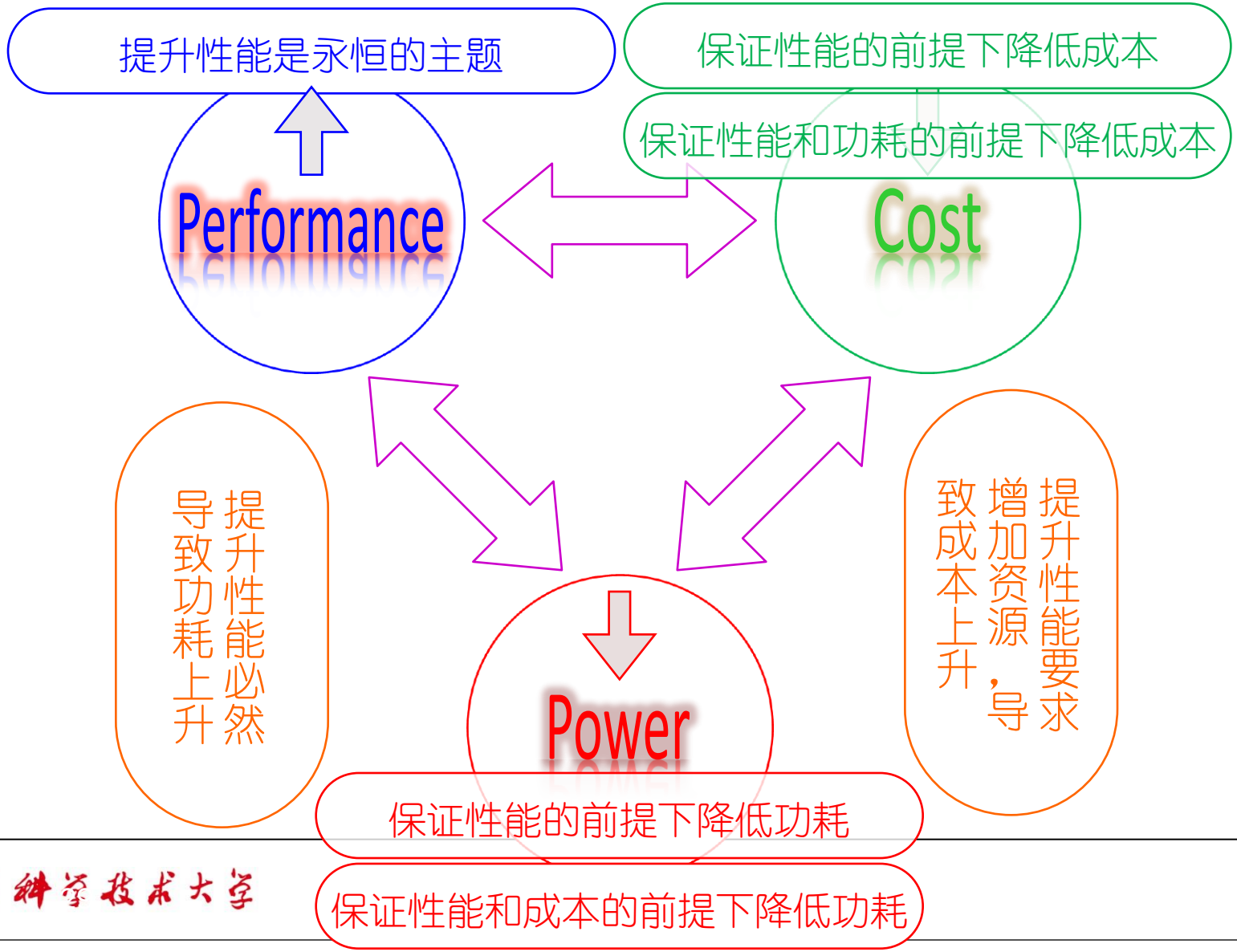
Physical synthesis(物理综合)

- **Transformation from a gate-level netlists to its physical layouts (masks)**
  - Partitioning (划分)
  - Floorplanning （布图规划）
  - Power/Ground Network Synthesis （电源线/地线网络合成）
  - Placement （布局）
  - Routing （布线）： Clock Tree, Signal, Power/Ground
  - Post-routing （后布线优化）
    - Redundant Via Insertion (Design for Yield)

Large Scale Discrete Optimization Problems: Computation-Consuming.

How to find good solutions in acceptable time?

Performance, Power, Cost



Modern Chip (VLSI) Design

- Managing Design Complexity (设计复杂性)
  - System-on-Chip: IP reuse, Platform-based Design Methodology
  - Electronic System Level Synthesis (High-level synthesis)
    - Raising the design abstraction
- Power (功耗)
  - Low Power Technologies: LeakagePower, Dynamic Power
  - New Device: FinFET
- Interconnect （互连）
  - Delay, System-level design & Physical Design
  - Network-on-Chip
  - 3D ICs
  - Optical-/wireless-/CNT-/RF Interconnect
- Reliability （可靠性）
- Design For Manufacturability: EUV, OPC, PSM, …

References

- Meil H. E. Weste and David Money Narris, "CMOS VLSI Design: A circuits and Systems Perspective", fourth edition, Addison-Wesley
  - Chapter 14.
- Further reading: Chapter 14.5 Design Economics