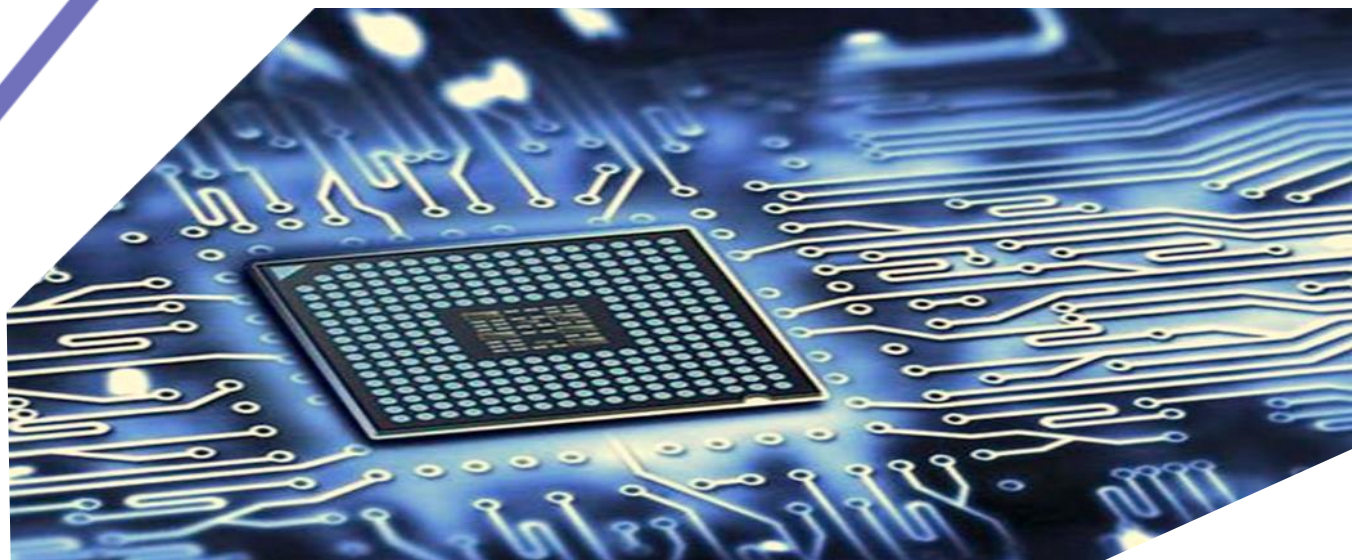




数字EDA理论基础

北京邮电大学集成电路学院

赵康 zhaokang@bupt.edu.cn



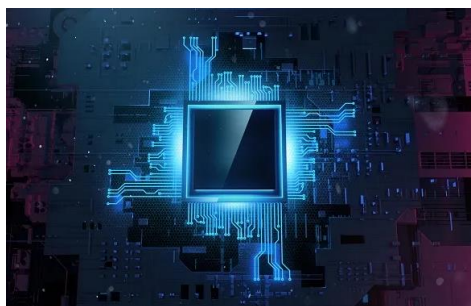
集成电路(芯片) 是国家发展战略的重中之重



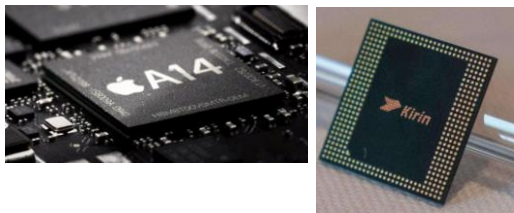
- 2022年全球半导体收入为6332亿美元，同比增长13.4%
- 美国国防部高级研究计划局(DARPA) 提出“电子复兴计划”未来五年投入15亿美元



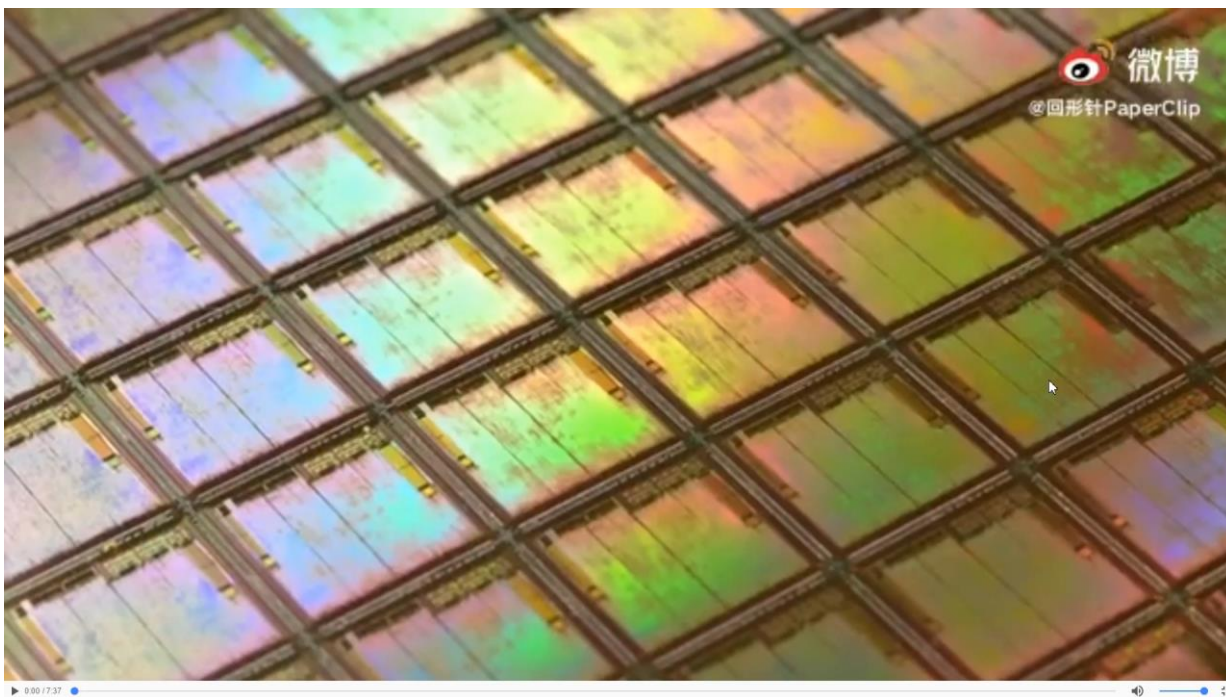
集成电路难点——高复杂度



全新5nm工艺苹果A14晶体管高达150亿



麒麟990处理器的总晶体管数量能达到103亿



在加工 5 纳米制程芯片时，允许的缺陷相当于在北京五环内，最多只能有两个芝麻大小的多余颗粒。



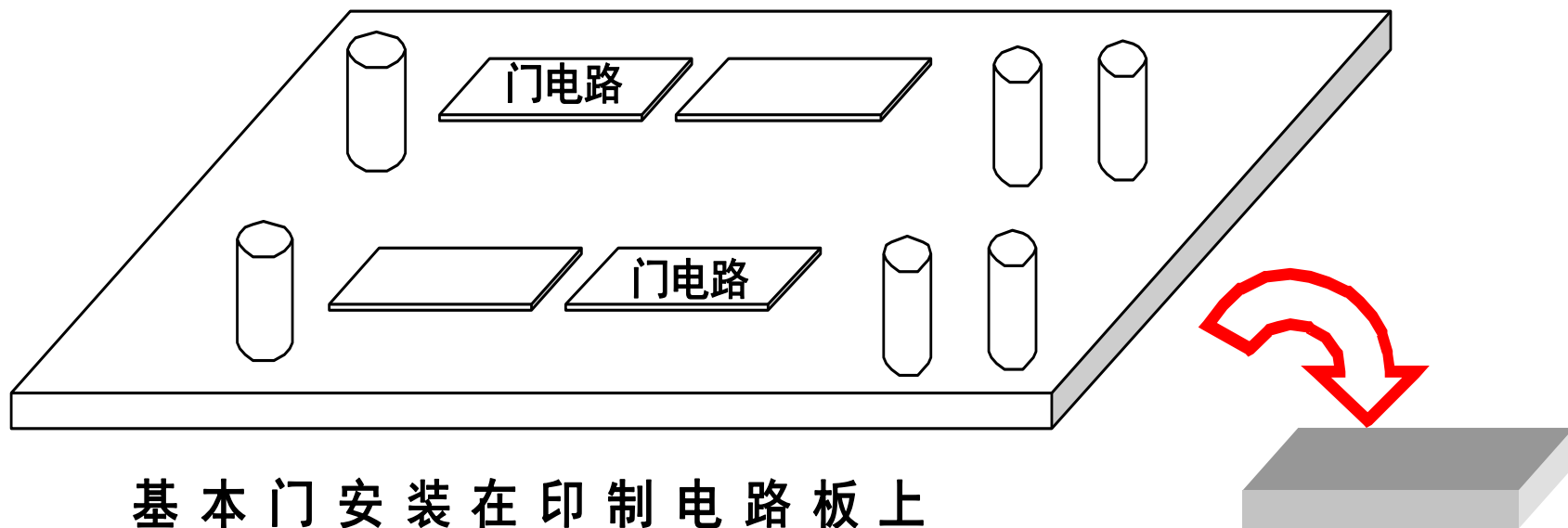
集成电路的复杂度与EDA的需求

- ◆ 集成电路设计复杂度急剧上升
 - 集成了越来越多的晶体管，难以用传统的手工方法来设计和处理
- ◆ 工艺越先进，流片费用越高
 - 最先进的5nm工艺，流片一次的费用高达3亿元人民币！
 - 对芯片设计的成功率，可靠性提出越来越高要求
- ◆ 解决方法：
 - EDA(electronic design automation)工具
 - 设计方法改进

**EDA工具和设计方法
是复杂电路的设计保证！**



集成电路设计的基本思路



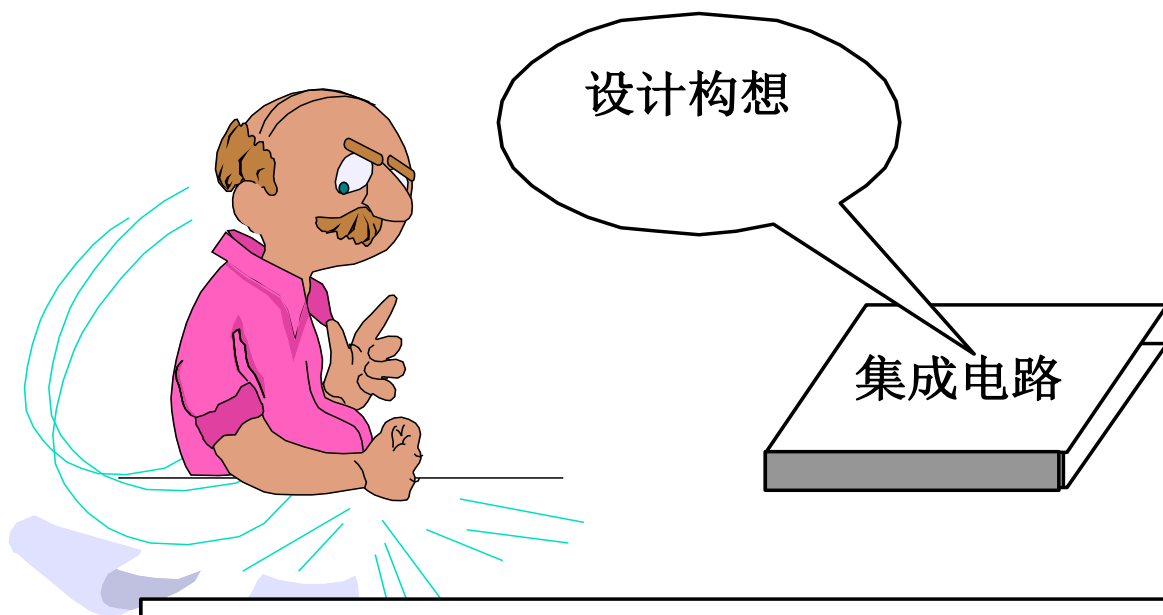
基本门安装在印制电路板上

集成电路

集成电路设计是指将电子元件、电路和系统集成到单个芯片上的过程



设计方法的选择

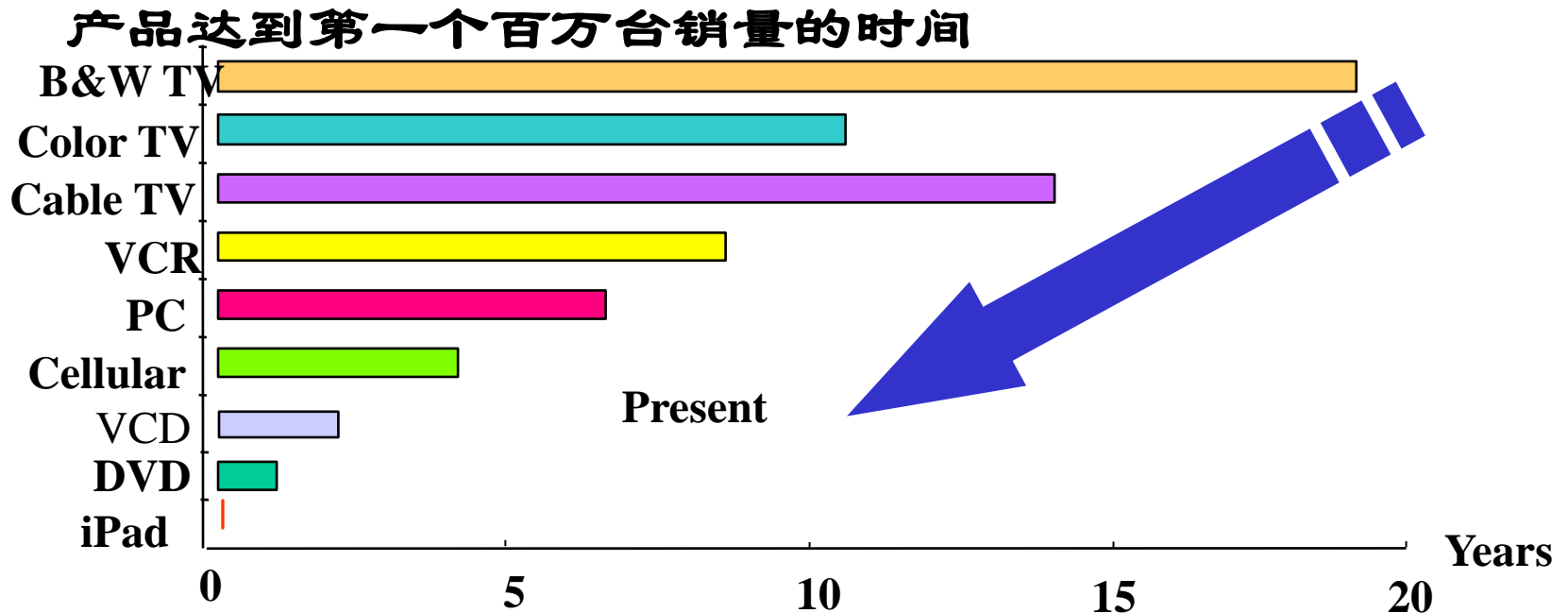


- 手工设计：设计者 + 纸 + 笔
- 自动设计：设计者 + EDA 工具



EDA技术加速了芯片设计

- ◆ 强大的EDA工具，缩短了集成电路的上市时间，是芯片产品竞争力的关键





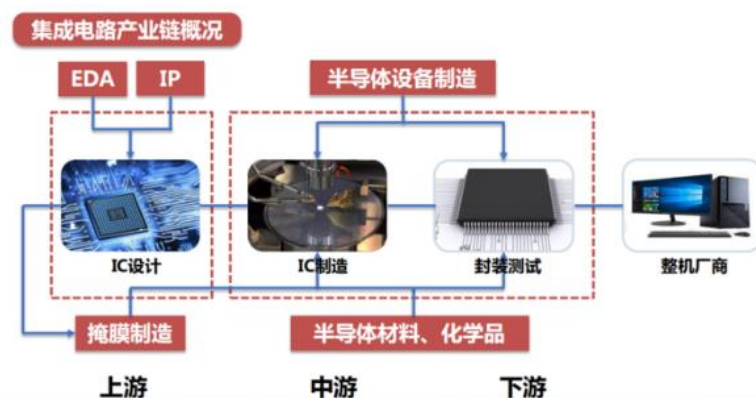
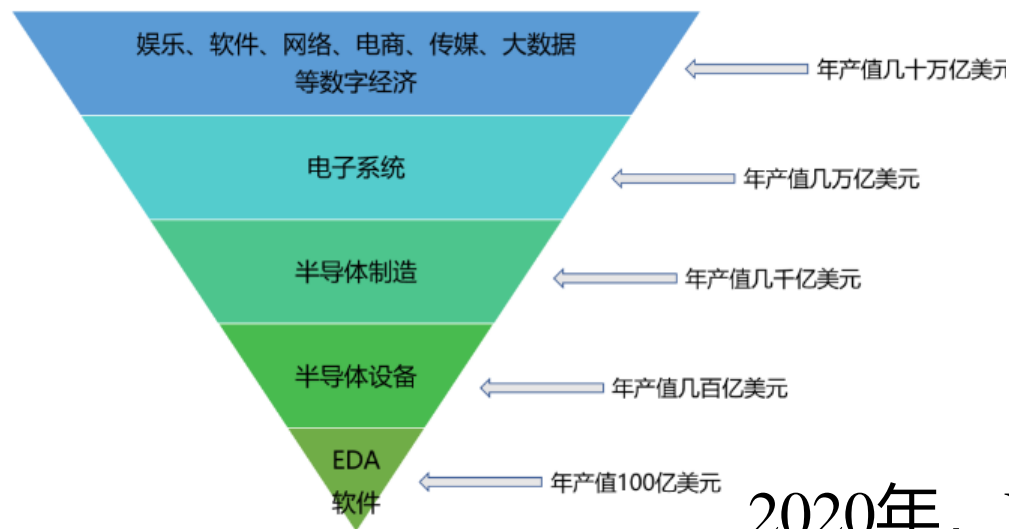
半导体商业模式





EDA是芯片设计的基石

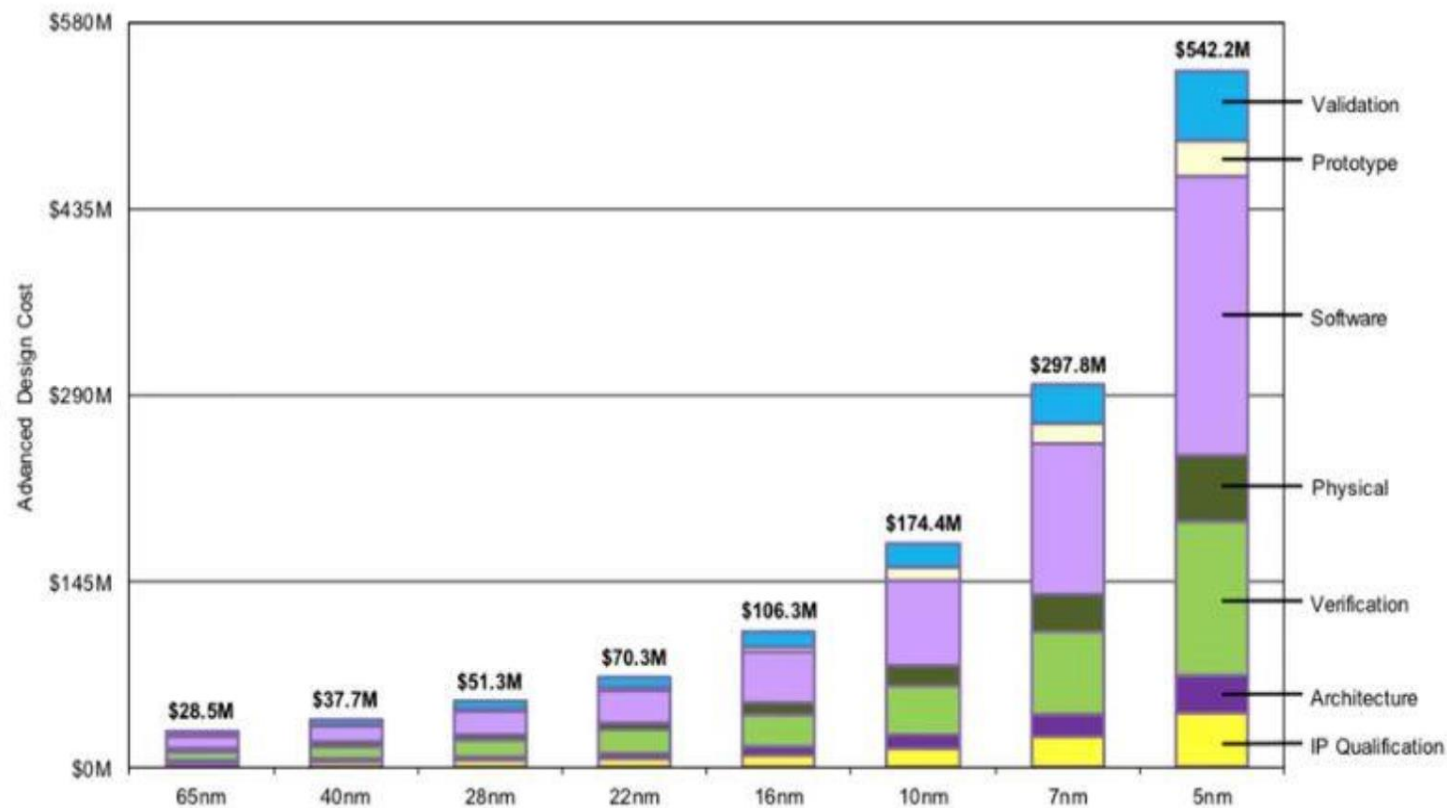
- 如果说光刻机是芯片制造领域的皇冠的话，EDA软件可能就是皇冠上那颗最璀璨的宝石
- 离开了EDA工具，半导体的设计与制造都是不可想象的事情



2020年，EDA行业市场规模约114.67亿美元，规模虽小，但支撑着年产值几十万亿美元的数字经济 → **杠杆效应**



EDA是集成电路产业的基础支柱



随着工艺节点推进，芯片设计成本快速提升

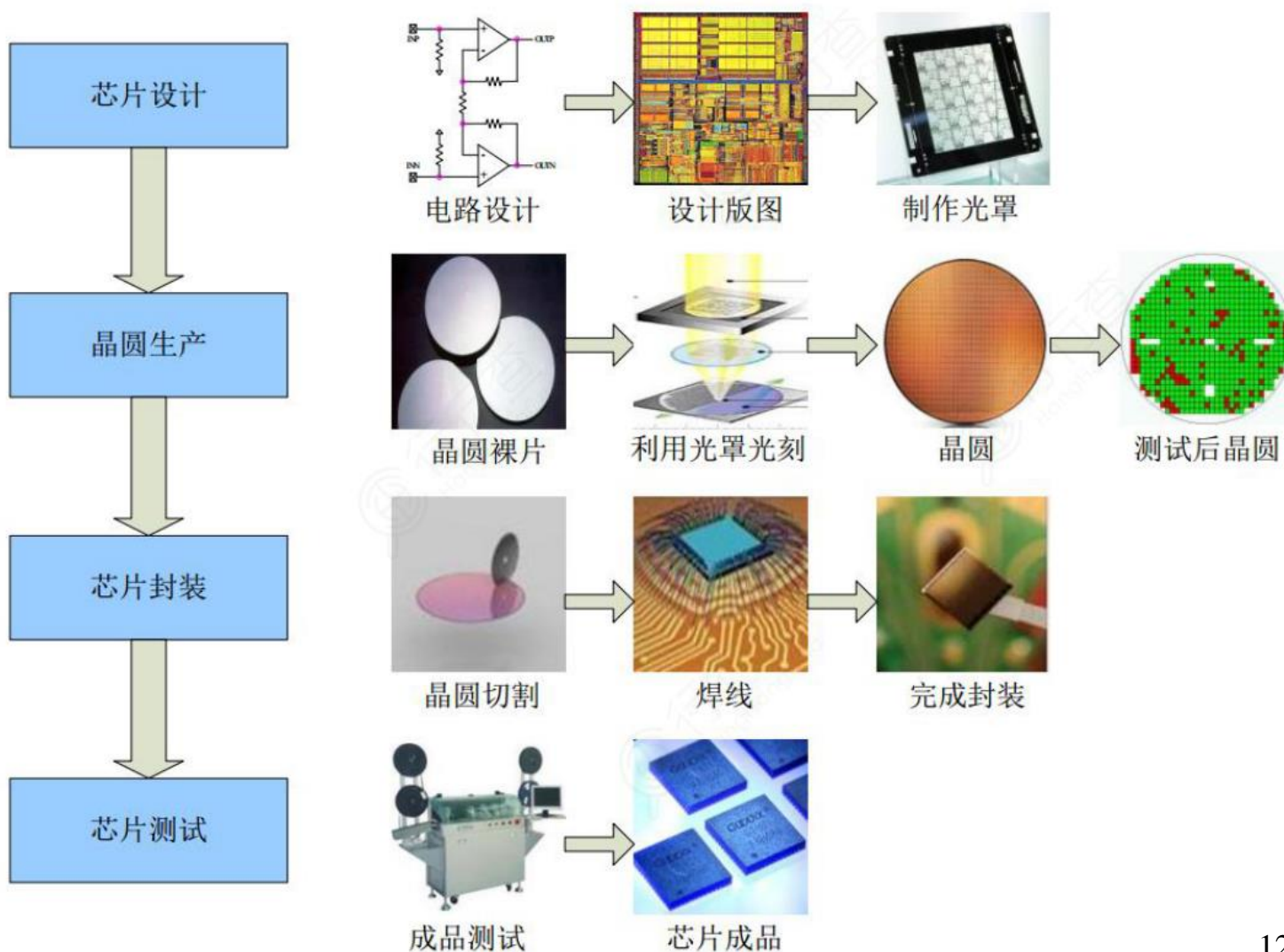


多项国家政策出台支持发展EDA

- 中美科技战中的“卡脖子”关键技术：**光刻机 & EDA**
- 2021年3月，十三届全国人大四次会议表决通过《**中华人民共和国国民经济和社会发展第十四个五年规划和2035年远景目标纲要**》：集成电路位列7大科技前沿领域攻关的第3位，并明确指出，**重点攻关集成电路设计工具（EDA）**
- 2022年1月，国务院发布《**“十四五”数字经济发展规划**》，提出增强关键技术创新能力，瞄准传感器、量子信息、网络通信、**集成电路**、关键软件等战略性前瞻性领域，发挥我国社会主义制度优势、新型举国体制优势、超大规模市场优势，提高数字技术基础研发能力



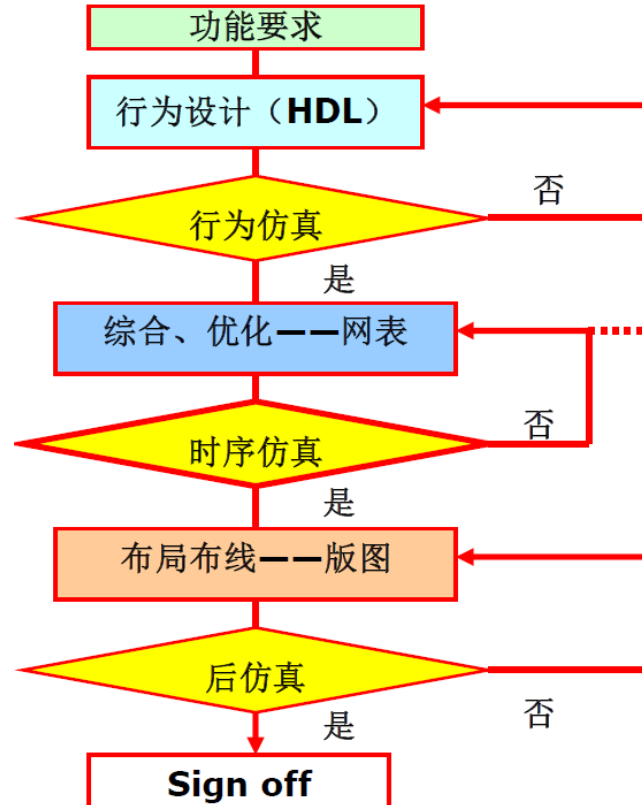
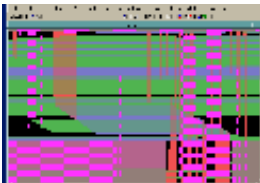
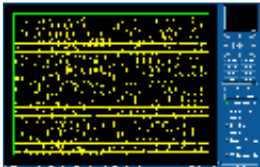
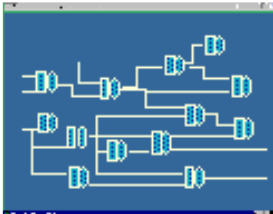
EDA无处不在





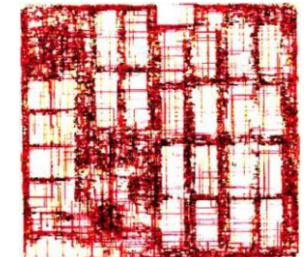
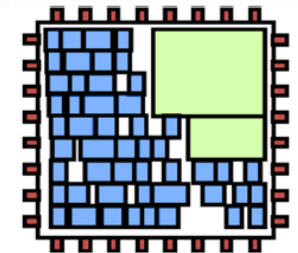
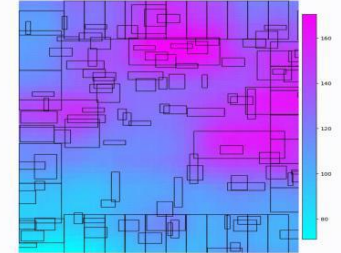
数字系统自动设计的流程

```
module my_adder (a,b,sum);  
  input a, b, c;  
  output sum;  
  // synopsys resource r0  
  // map_to_module = "DW01_add"  
  // implementation = "cla"  
  // ops = "al"  
  assign sum= a+b;  
endmodule
```



逻辑及电路设计

版图设计





课程目的

- ◆ 学习开发EDA工具的方法；
- ◆ 使用EDA工具设计集成电路。



预备知识

- ◆ 硬件设计的基础知识：
 - 数字逻辑；
 - 计算机组成原理。
- ◆ 算法的基础知识：
 - 程序设计语言C/C++；
 - 数据结构。



第一章

概论

1.1 电子设计自动化技术发展的回顾

1.2 从EDA的角度观察VLSI

1.2.1 VLSI的分类

1.2.2 芯片布图模式

1.2.3 可编程逻辑器件

1.3 数字系统自动设计的流程

1.4 EDA的主要领域

1.4.1 硬件描述语言

1.4.2 模拟验证

1.4.3 综合技术

1.4.4 测试诊断

1.4.5 数字系统形式验证

1.4.6 版图设计

1.5 集成电路快速发展对设计自动化的挑战与机遇

1.6 用EDA工具设计集成电路



1.1 电子设计自动化技术发展的回顾

— EDA技术的发展与集成电路的发展 相伴而行 —

- ◆ SSI: 小规模集成电路;
- ◆ MSI: 大规模集成电路;
- ◆ LSI: 中规模集成电路;
- ◆ VLSI: 超大规模集成电路;
- ◆ ASIC (Application Specific IC): 专用集成电路;
- ◆ SOC (System On a Chip): 系统级芯片;



专用名词

- ◆ **设计自动化 (DA) (研究自动设计软件)**
Design Automation
- ◆ **计算机辅助设计(CAD)**
Computer Aided Design
- ◆ **电子设计自动化 (EDA) (模拟电路 + 数字系统)**
- ◆ **数字系统: 计算机, 通信, 控制电路, 集成电路...**
- ◆ **超大规模集成电路 (VLSI)**
Very Large Scale Integrated Circuit
- ◆ **人工设计→CAD→DA**



EDA发展历史与现状

- ◆ 50年代——萌芽
 - 美国Stretch工程
 - 用电子管计算机帮助设计晶体管计算机
 - 逻辑图的保存、检查、修改
- ◆ 60年代 ——简单的单独的CAD软件
 - 印刷电路板PCB、集成电路IC出现
+ 计算机应用的推广
 - PCB布线, 电路模拟, 电路分析, 逻辑模拟



◆ 70年代——第一代：CAD设备

- 小型专用计算机+CAD软件（Daisy系统）
- 逻辑模拟，逻辑优化
- 版图图形编辑+设计规则检查(DRC)

◆ 80年代——第二代：ICCAD软件系统

- 集成电路大量使用，集成电路工艺：5 μ , 3 μ , 1 μ
- EDA迅速发展，覆盖了逻辑图输入
- 集成电路自动设计全过程（逻辑图——产品）
 - 逻辑图输入、逻辑模拟、逻辑综合、
 - 逻辑电路测试码生成、版图和PCB的自动布局布线
- 熊猫系统，Mentor, Cadence, ViewLogic...

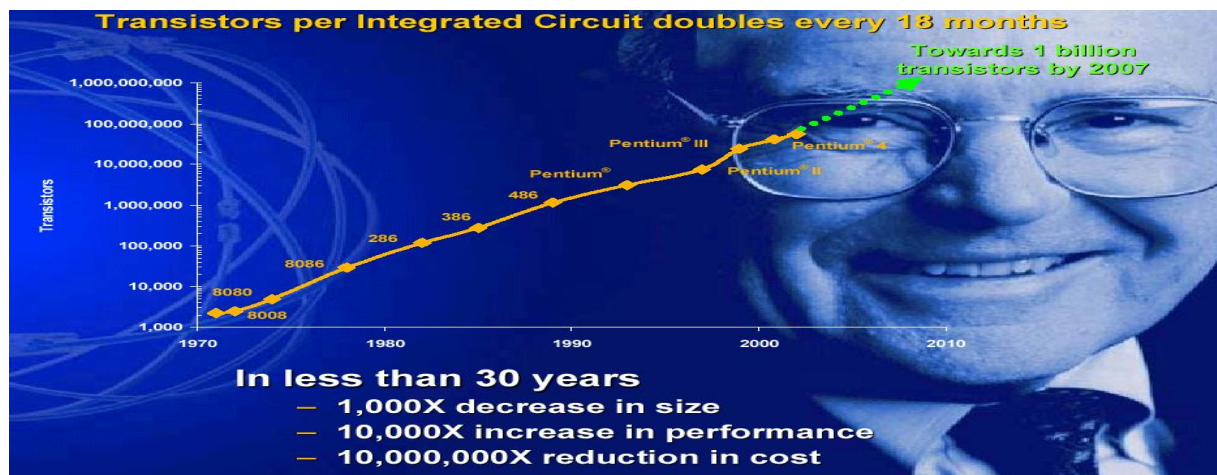


- ◆ 90年代——第三代：覆盖所有级别的EDA开放式集成系统
 - 集成电路工艺：0.8 μ , 0.5 μ , 0.25 μ , 0.18 μ ,
(亚微米, 深亚微米)
 - 高层次设计自动化, 形式验证
 - 标准硬件描述语言VHDL
 - 时延、功耗驱动的高层次综合与版图自动布局布线
 - Synopsys, Mentor, Cadence...
- ◆ 2000以来——以SoC为中心
 - 0.13 μ ,0.08 μ ...



集成电路工艺的发展

- ◆ 集成电路工艺的发展=>设计自动化技术的发展
- ◆ 芯片集成度指数级增长
 - Moore定律: 最小晶体管特征尺寸每三年以0.7倍递减
这个规律至少保持了30年! (first published in 1965)



IC技术: SSI/MSI-> LSI-> VLSI-> ULSI-> GLSI->...

晶圆尺寸: 100mm-> 125mm-> 150mm-> 200mm-> 300mm-> 400mm->...

特征尺寸: 3 μ m-> 2 μ m-> 1.2 μ m-> 0.8 μ m-> 0.5 μ m-> 0.35 μ m-> 0.25 μ m-> 0.18 μ m-> 0.13 μ m-> 90nm-> 28nm-> 7nm-> ...

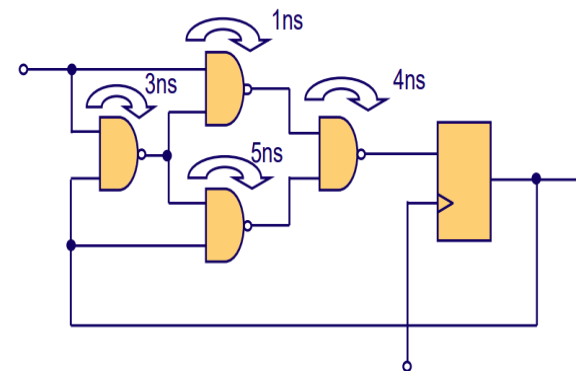
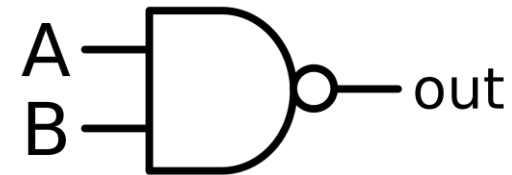


人工设计集成电路

真值表

输 入		输 出
A	B	Y
0	0	1
0	1	1
1	0	1
1	1	0

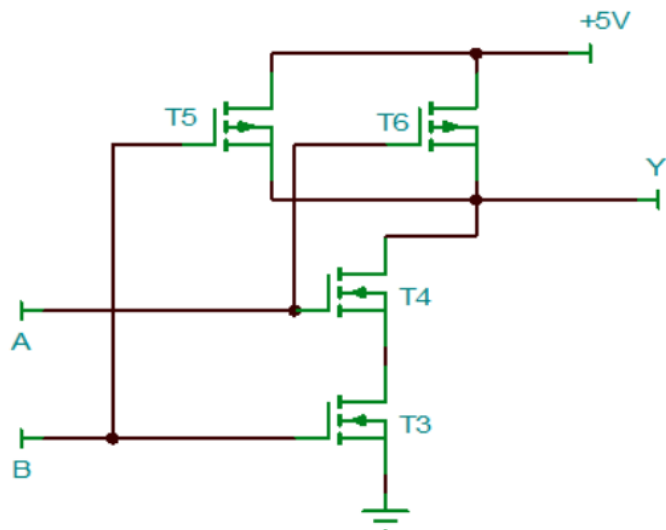
符号表示



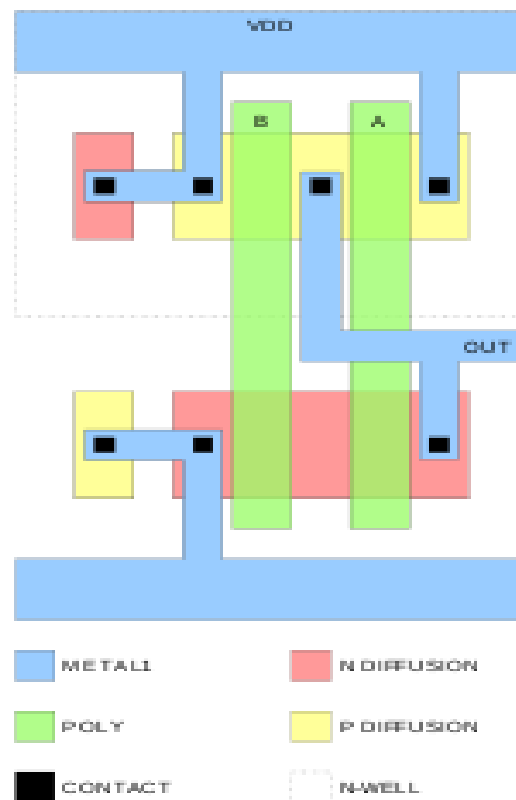


人工设计集成电路

晶体管



版图





自动化设计集成电路

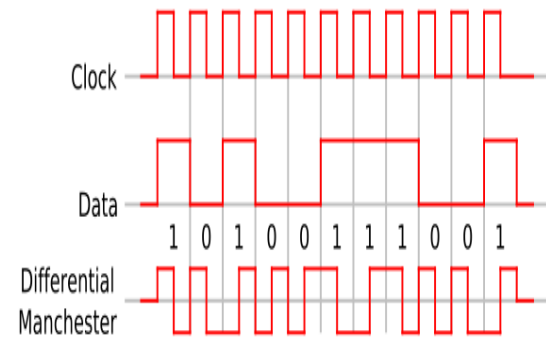
C/C++/
Python

Verilog

功能仿真

```
for(row=0; row<R; row++) {  
  for(col=0; col<C; col++) {  
    for(to=0; to<M; to++) {  
      for(ti=0; ti<N; ti++) {  
        for(i=0; i<K; i++) {  
          for(j=0; j<K; j++) {  
            L: output_fm[to][row][col] +=  
              weights[to][ti][i][j]*  
              input_fm[ti][S*row+i][S*col+j];  
          } } } } }  
}
```

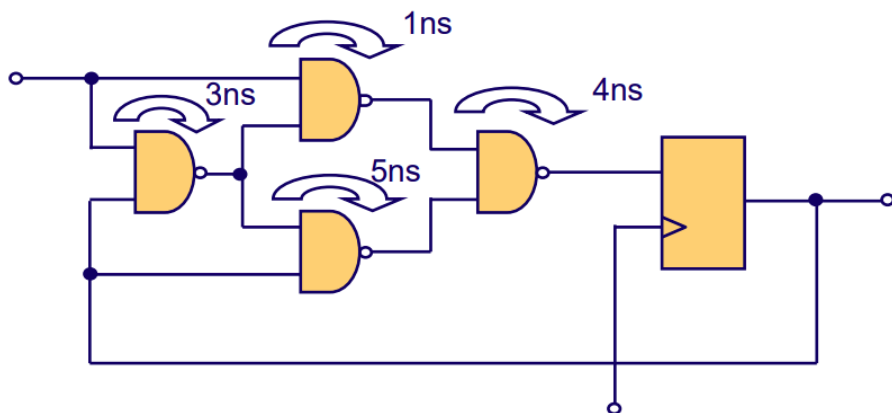
```
module conv;  
  reg [31:0] m[0:8192];  
  reg [12:0] pc;  
  reg [31:0] acc;  
  reg [15:0] ir;  
  
  always  
    begin  
      ir = m[pc];  
      if(ir[15:13] ==  
        3b'000)  
        pc =  
          m[ir[12:0]];  
      else if (ir[15:13]  
        == 3'b010)  
        acc = -  
          m[ir[12:0]];  
      ...  
    end  
endmodule
```



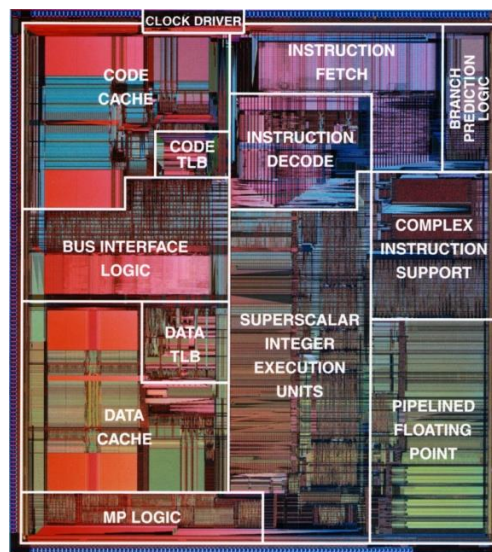


自动化设计集成电路

门级电路



版图实现





EDA是什么

- EDA (Electronic Design Automation), 全称电子设计自动化, 是用来辅助超大规模集成电路设计生产的工业软件, 涵盖**芯片设计、制造、封装、测试**整个流程
- 随着芯片设计的复杂程度不断提升, 基于先进工艺节点的集成电路规模可达到**数千亿个半导体器件**, 不借助EDA已经无法完成芯片设计
- EDA工具是集成电路设计和制造流程的支撑, 是集成电路设计方法学的载体, 也是**连接设计和制造两个环节的纽带和桥梁**





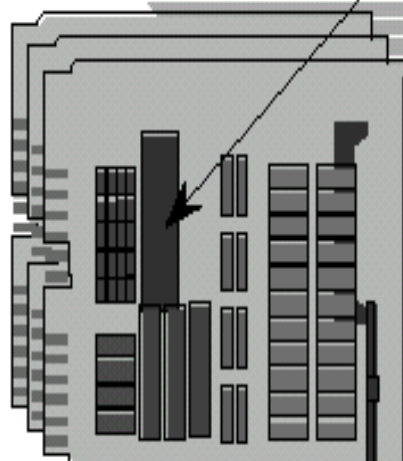
EDA是什么

狭义EDA：专指片级，包括模拟、数字、混合等

广义EDA：片级设计（片上系统SoC）、板级设计

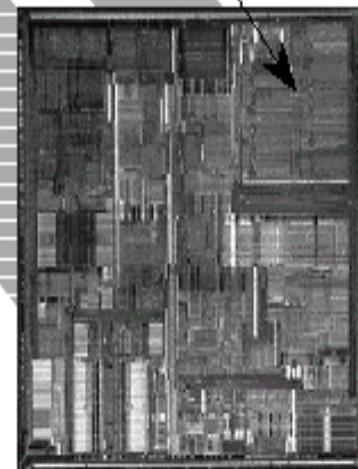


**“Real”
Component**



**System
on Board**

**“Virtual”
Component**



**System
on Silicon**

**Deep
Sub-Micron**

从板上系统到片上系统



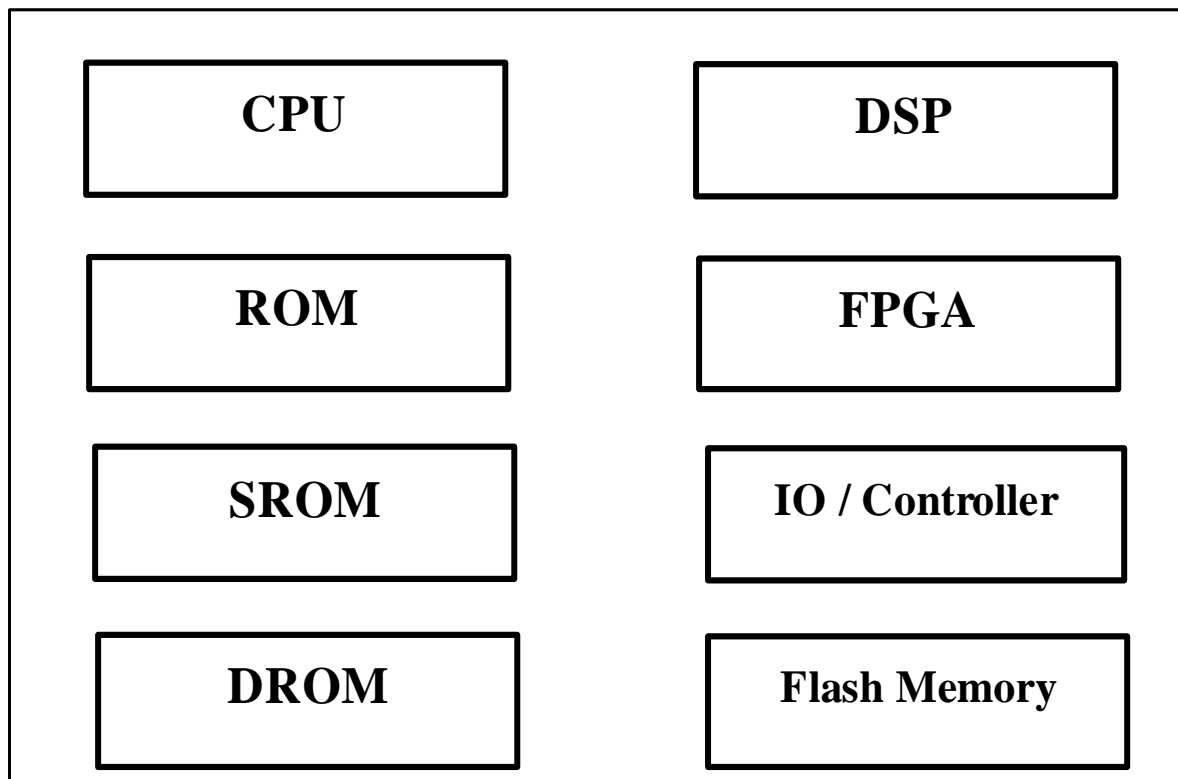
System-on-a-Chip (SoC)

- ◆ **System-in-a-Room** (Vacuum Tubes)
- ◆ **System-in-a-Cabinet** (Transistors)
- ◆ **System-on-a-Board** (Printed Circuit Board)
(Integrated Circuits , Microprocessors)
- ◆ **System-in-a-Package** (Multi-chip Module)
- ◆ **System-on-a-Chip**



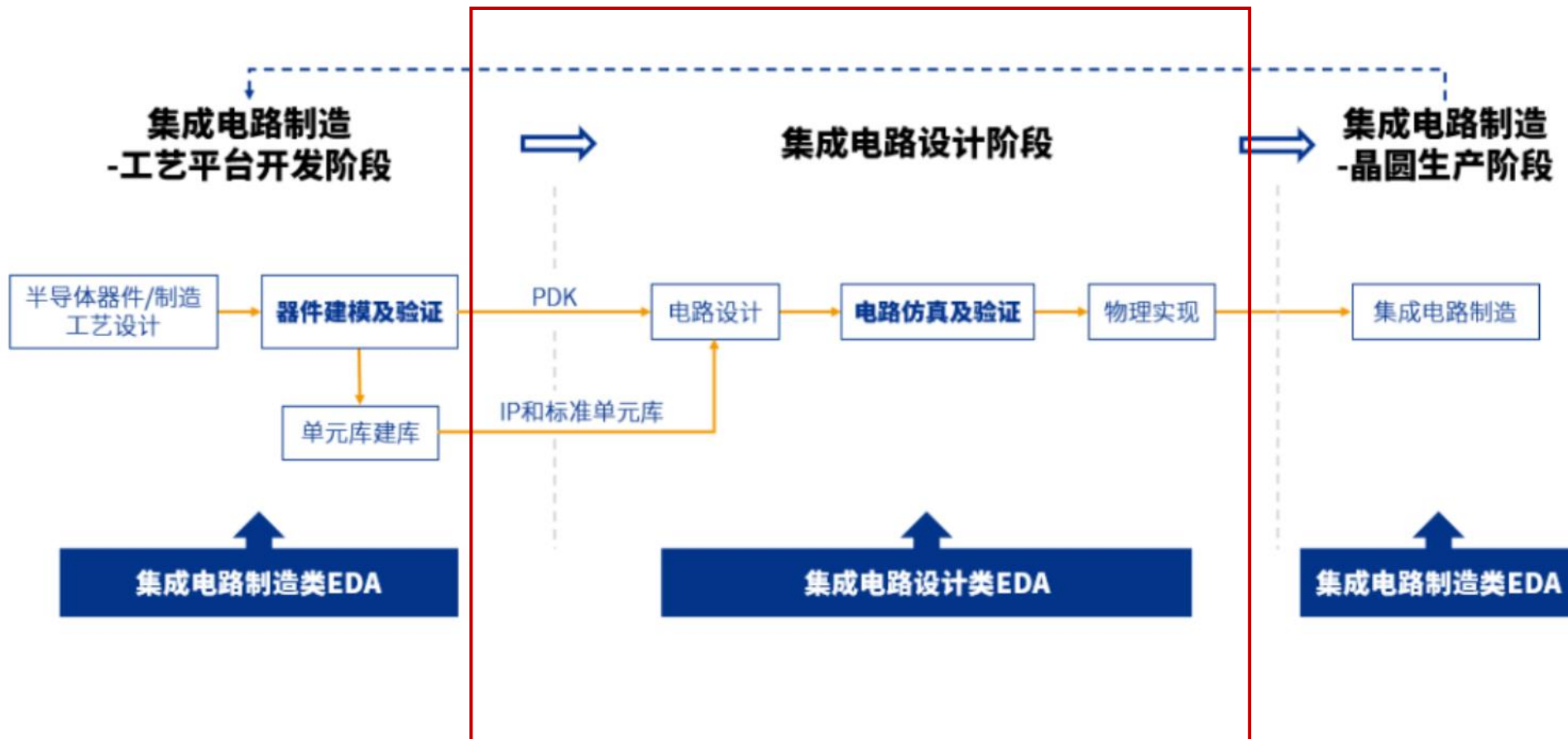
一个SOC结构的示意图

-- 什么是SOC? --





EDA工具的分类





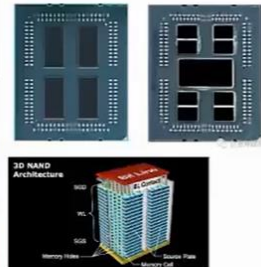
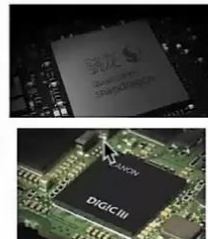
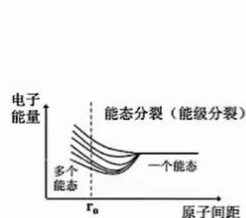
1.2 从EDA的角度观察VLSI

- ◆ EDA工具是为VLSI的设计、生产服务的，因而它必须适应VLSI技术的要求。

- ◆ 1.2.1 VLSI的分类
- ◆ 1.2.2 芯片布图模式
- ◆ 1.2.3 可编程逻辑器件



EDA的发展伴随集成电路的发展



能带理论

晶体管

集成电路

大规模集成

SRAM

处理器

DRAM

FPGA

Flash

EDA

处理器大发展

ARM

SoC

专用集成电路

3DIC

SiP

Chiplet

WSI

量子力学

1930

1940

1950

1960

1970

1980

1990

2000

2010

2020



1.2.1 VLSI的分类

- ◆ 按工艺分类，最主要的有：
 - 金属氧化物半导体工艺
(Metal Oxide Semiconductor, MOS) :
 - pMOS — p沟道MOS;
 - nMOS — n沟道MOS;
 - CMOS — 互补型MOS。
- ◆ 晶体管—晶体管逻辑
(Transistor-Transistor Logic, TTL) ;
- ◆ 发射极耦合逻辑 (Emitter Coupled Logic, ECL) 。



1.2.1 VLSI的分类 (续)

- ◆ 按生产目的分类：
 - 通用集成电路；
 - 专用集成电路（Application Specific Integrated Circuit, ASIC）。
- ◆ 按实现方式（设计风格）分类：
 - 全定制（Full -Custom）方式；
 - 半定制（Semi-Custom）方式；
 - 可编程逻辑器件（ Programmable Logic Device, PLD）方式。



集成电路分类

全定制：专用集成电路（ASIC）

半定制：可编程逻辑门阵列（FPGA）

非定制（通用）：处理器（CPU）、
图形处理器（GPU）





通用集成电路

- 微处理器芯片，存储器芯片，计算机外围电路芯片等
- 特点：规模大，批量大，性能精细，设计成本高
- VLSI规模不断扩大，100万~1000万晶体管，
- 整个系统的集成（芯片上系统）

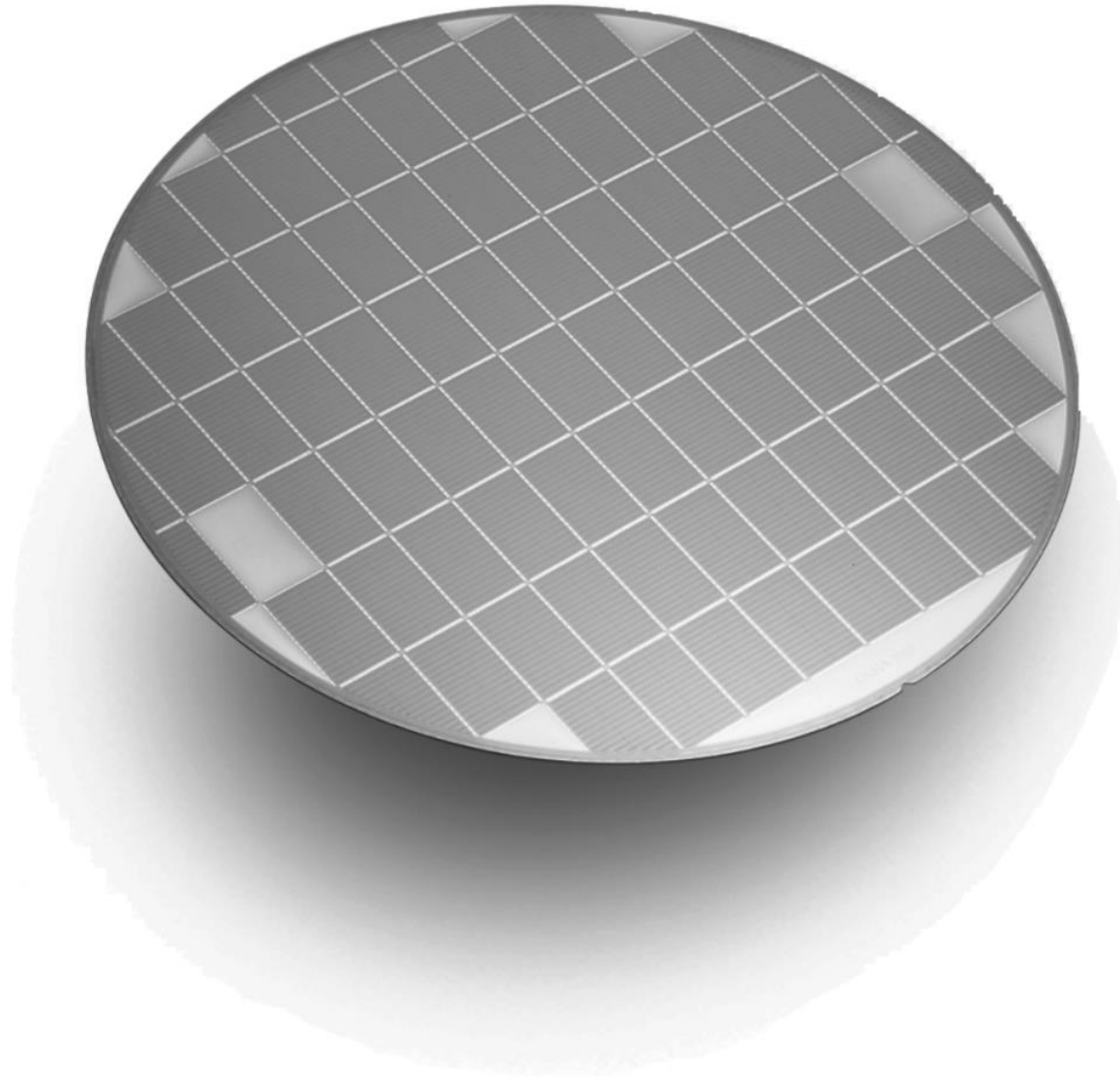


专用集成电路

◆ ASIC

Application Specific Integrated Circuit

- 专门用途的芯片，如控制电路等
- 特点：规模小，批量小，要求设计成本低，周期短
- 分类：
 - 客户全定制芯片
 - 客户半定制芯片：标准单元，门阵列，宏单元



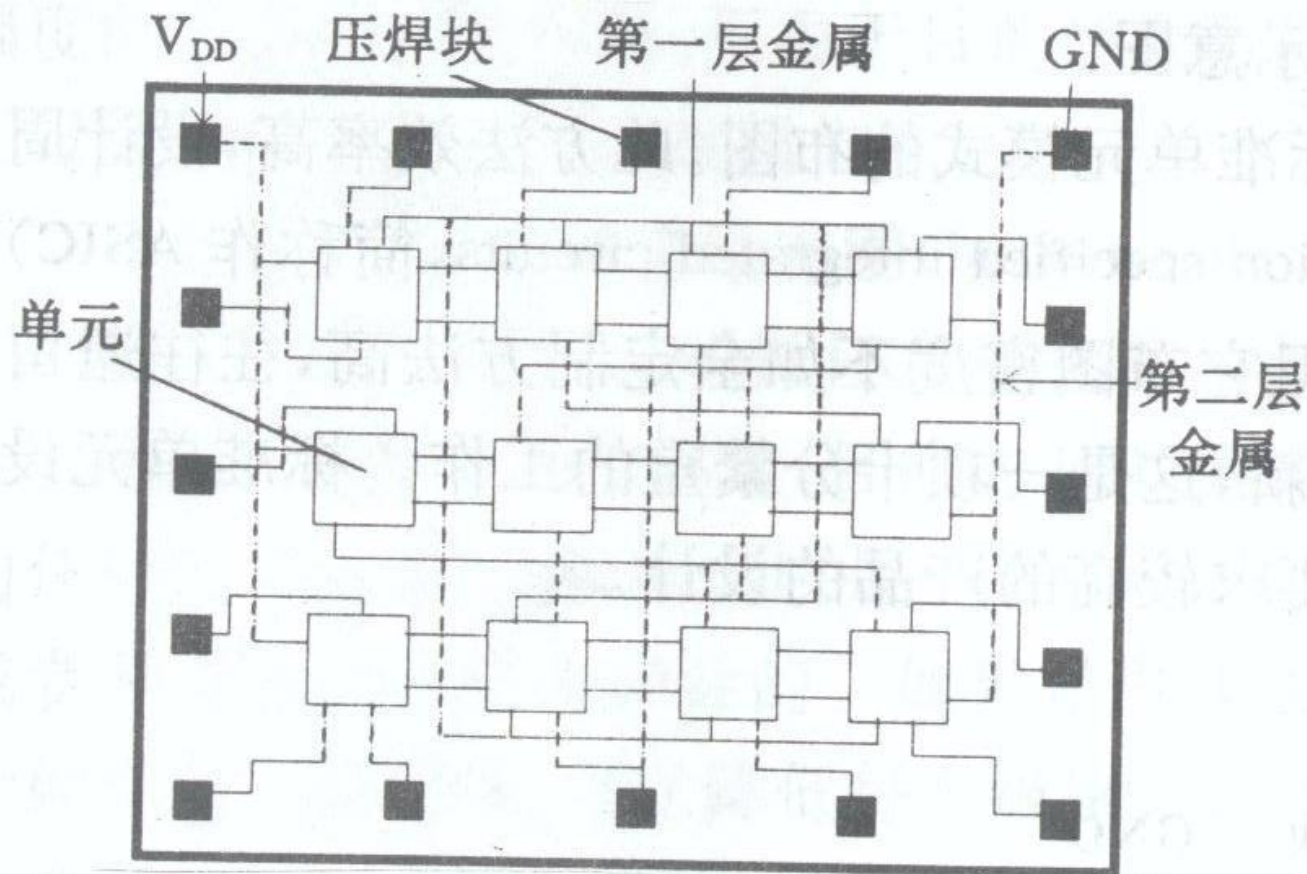


1.2.2 芯片布图模式

- ◆ 门阵列（Gate Array）模式；
- ◆ 标准单元（Standard Cell）模式；
- ◆ 积木块（Building Block）模式；
- ◆ 可编程逻辑器件（Programmable Logic Device）模式。

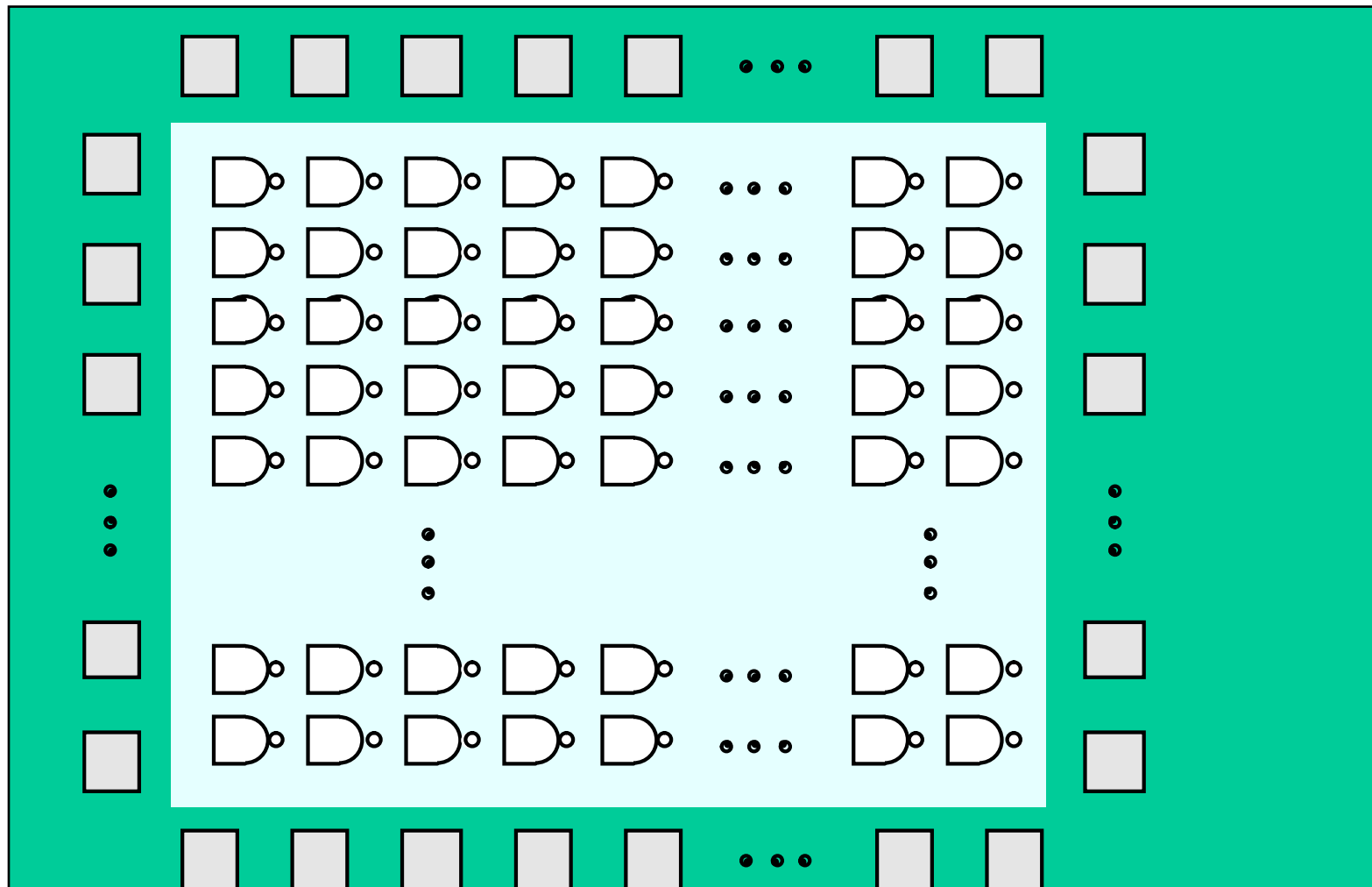


门阵列



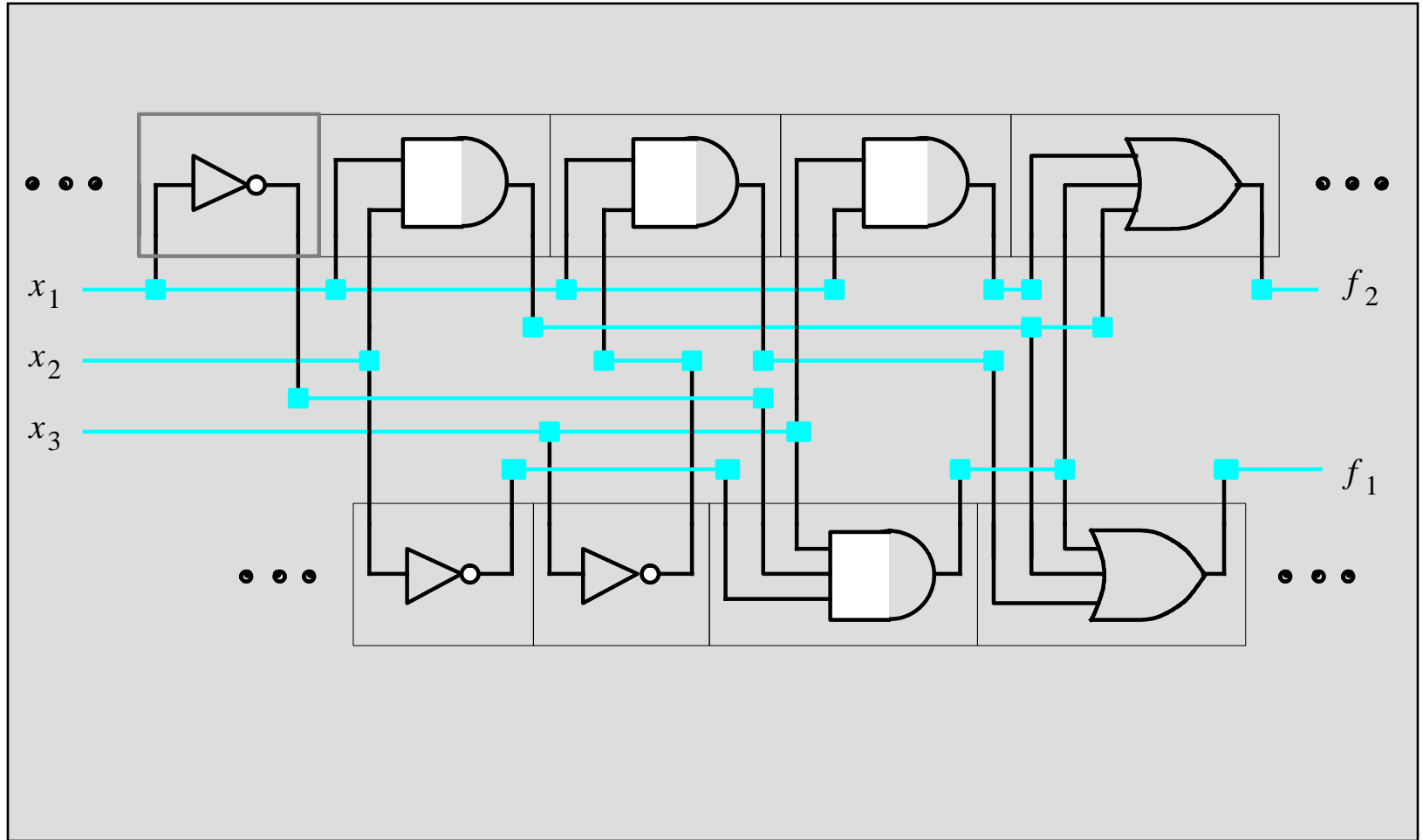


ASIC--门海门阵列结构示意图



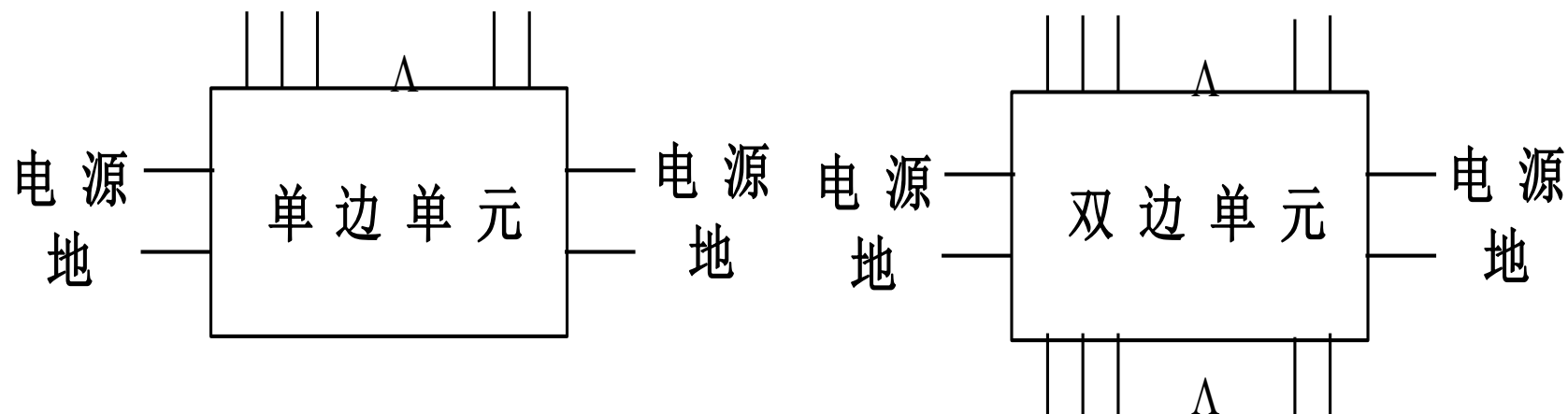


ASIC--标准单元芯片示意图





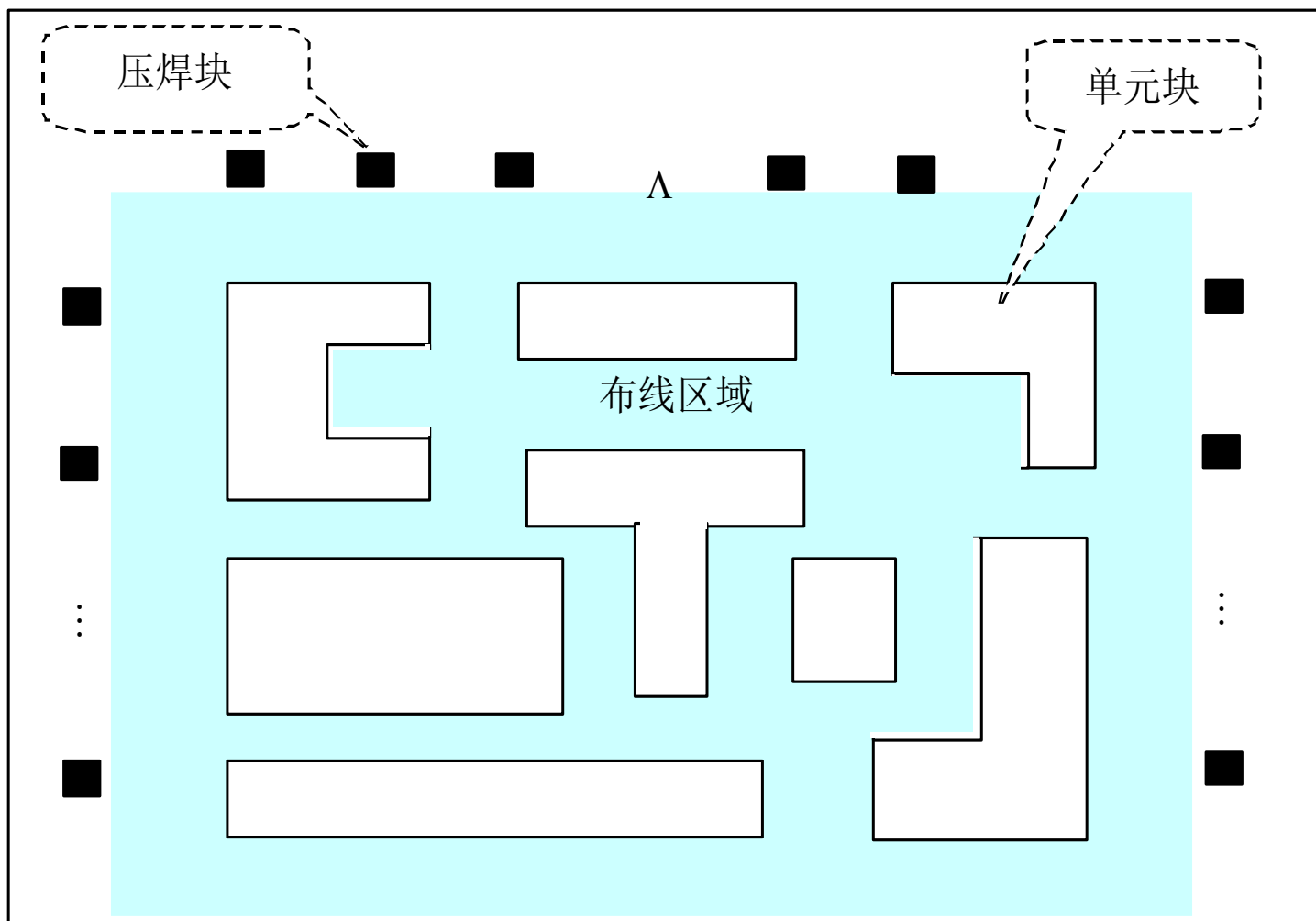
标准单元引出接点示意图



标准单元引出接点示意图



积木块模式示意图





1.3.3 可编程逻辑器件PLD

- ◆ **2.3 现场可编程逻辑器件(PLD)**
- ◆ **Programmable Logic Device**
 - 可写入所需要的电路的“空白”电路
 - 特点：规模小, 批量大, 设计周期很短, 有的具有可重写功能



PLD分类

◆ 简单PLD

- PLA (Programmable Logic Array)
- PAL (Programmable Array Logic)
- GAL (Gate Array Logic)

◆ 复杂PLD

- CPLD (Complex Programmable Logic Device)
- FPGA (Field Programmable Gate Array)



可编程逻辑器件PLD (续)

现场可编程技术可归类如下：

- ◆ 熔丝技术——平时连通，加电可使熔断；
- ◆ 反熔丝技术——平时不连通，加电可使连通；
 - 以上属于一次性写入
- ◆ 电可写技术（可多次写入）
 - 紫外线擦除、电可写；
 - 电可擦、可写。



FPGA和CPLD的比较

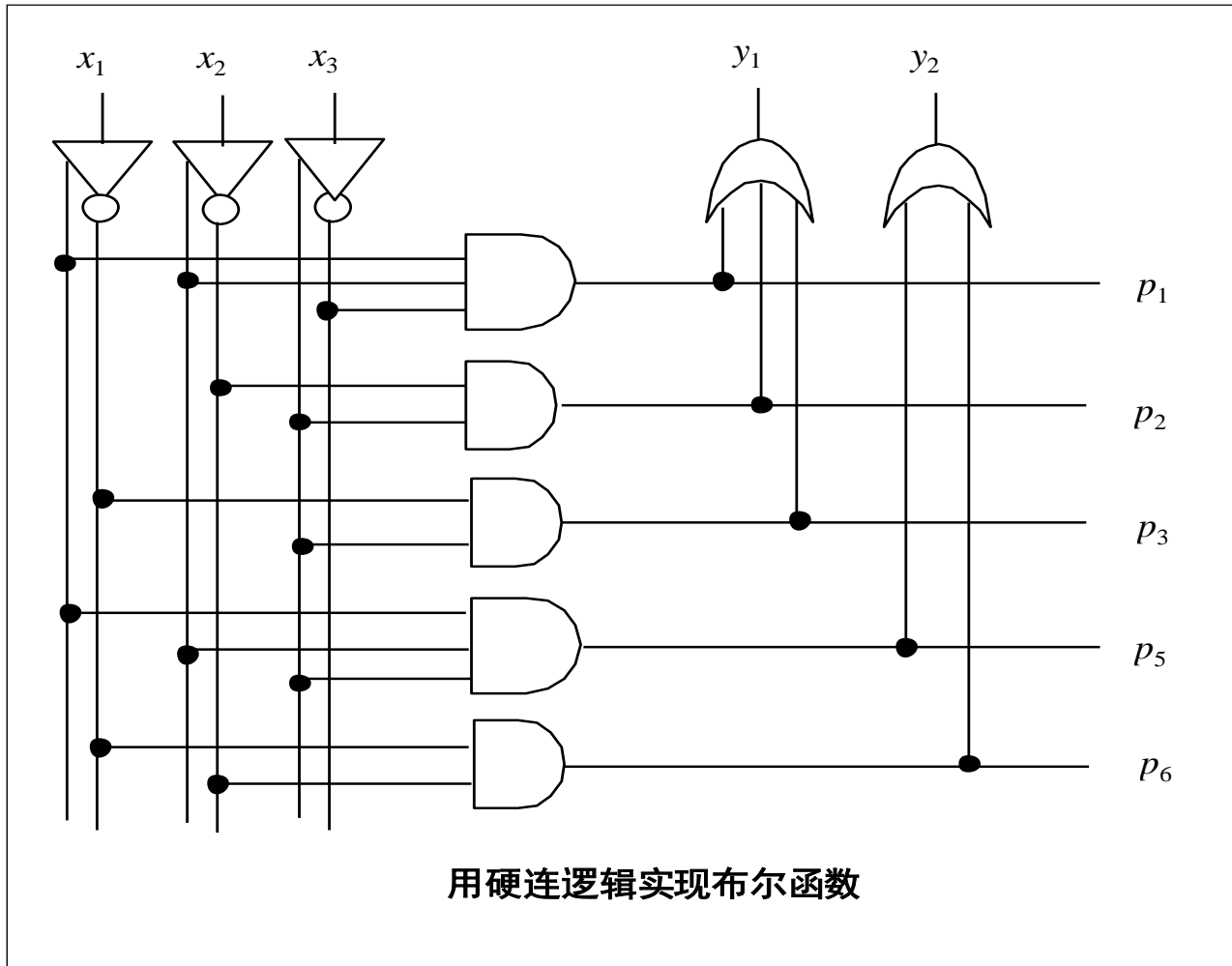
- ◆ 宏单元的规模(输入变量个数):
 - CPLD: 8 ~ 10 属于粗粒度;
 - FPGA: 3 ~ 5 属于细粒度;
- ◆ 宏单元中组合逻辑的实现方法
 - CPLD: 乘积项之和(用与-或阵列实现);
 - FPGA: 查找表(用SRAM/ROM实现);
- ◆ 宏单元中寄存器资源与组合逻辑资源的比例:
 - CPLD: 寄存器资源相对较少;
 - FPGA: 寄存器资源相对较多;



以实例说明PLD怎样实现一个布尔函数

$$y_1 = x_1 \cdot x_2 \cdot \overline{x_3} + \overline{x_2} \cdot x_3 + \overline{x_1} \cdot x_3$$

$$y_2 = x_1 \cdot x_2 \cdot x_3 + \overline{x_1} \cdot \overline{x_2}$$

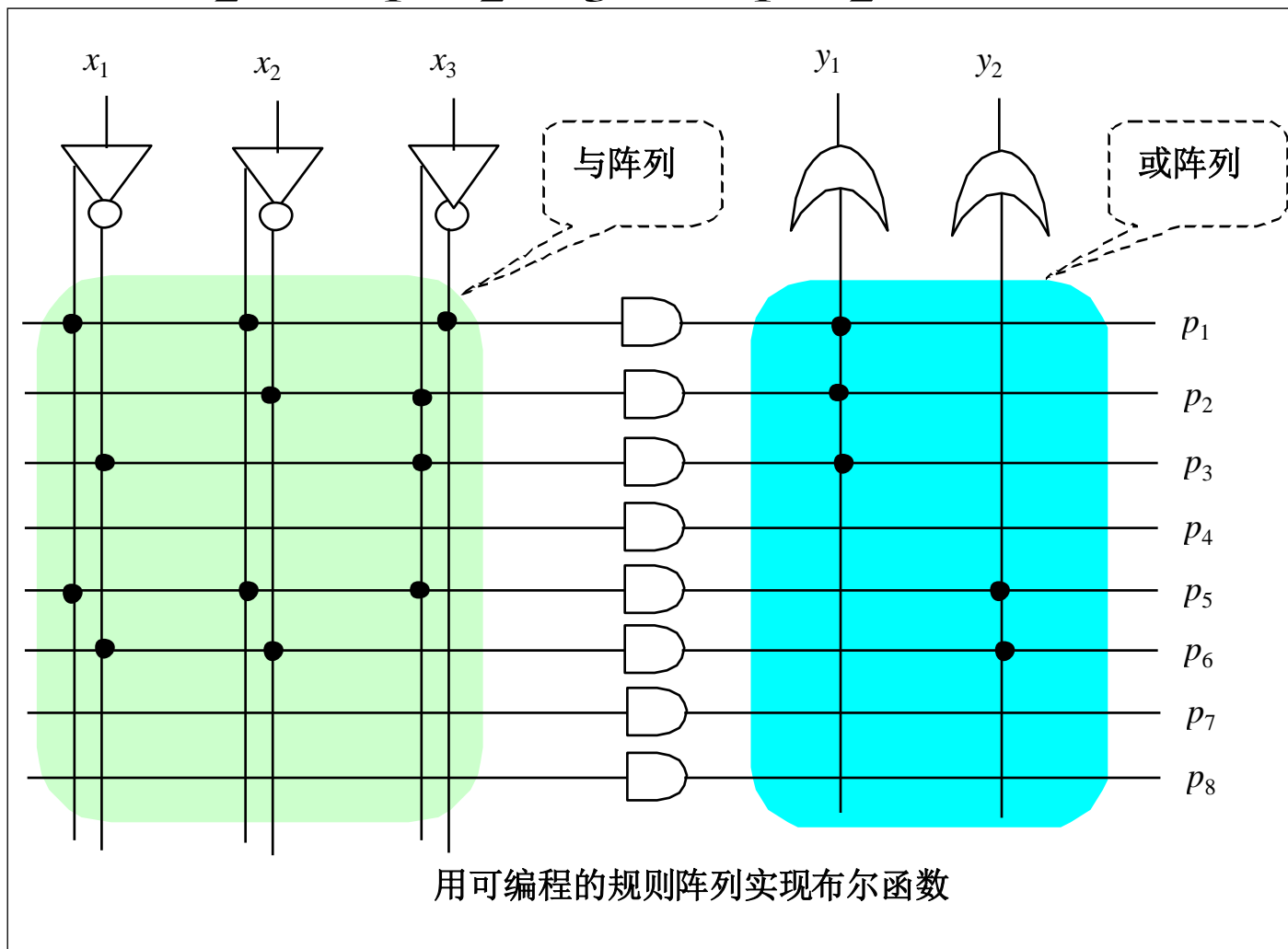




以实例说明PLD怎样实现一个布尔函数（续）

$$y_1 = x_1 \cdot x_2 \cdot \overline{x_3} + \overline{x_2} \cdot x_3 + \overline{x_1} \cdot x_3$$

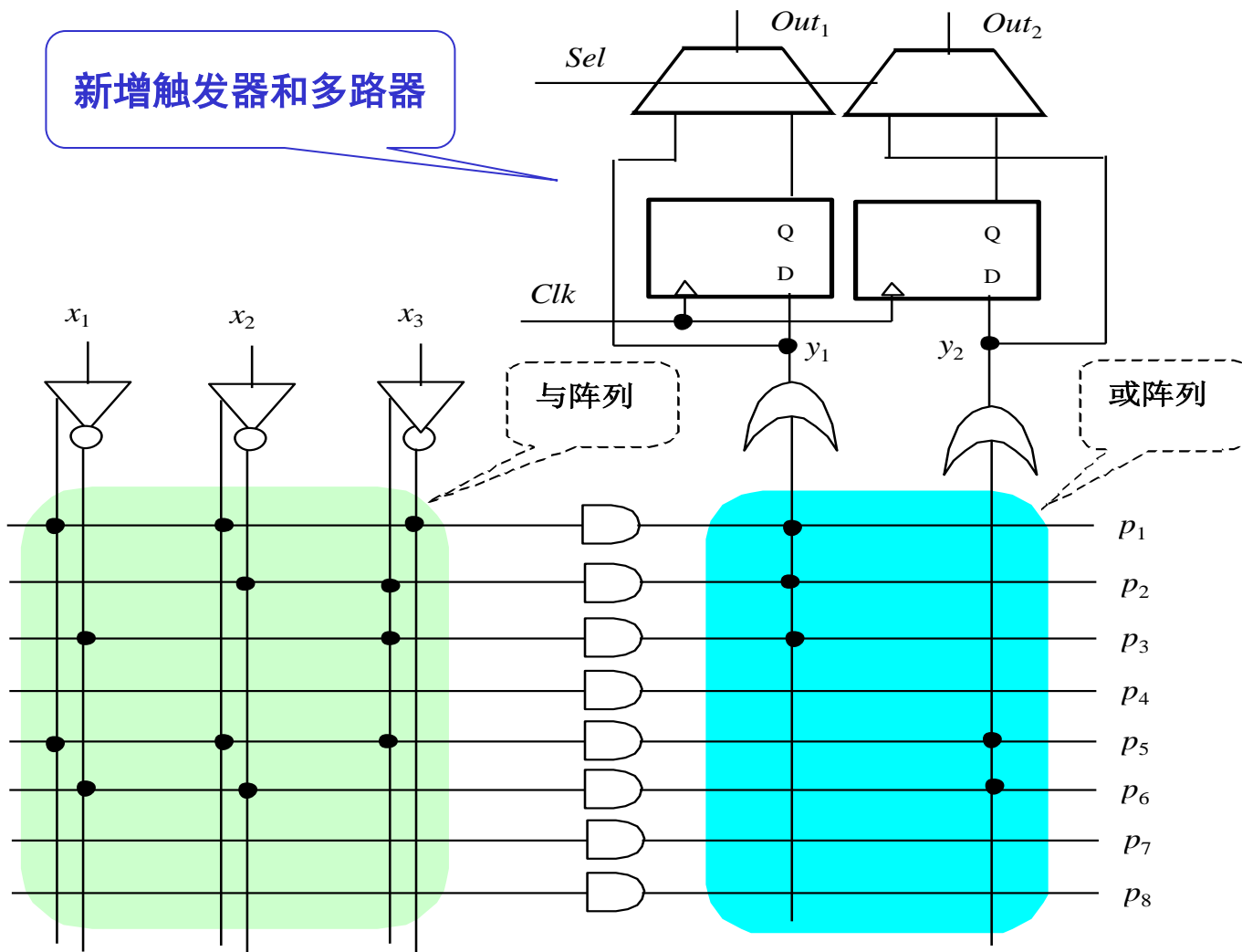
$$y_2 = x_1 \cdot x_2 \cdot x_3 + \overline{x_1} \cdot \overline{x_2}$$





GAL的雏形

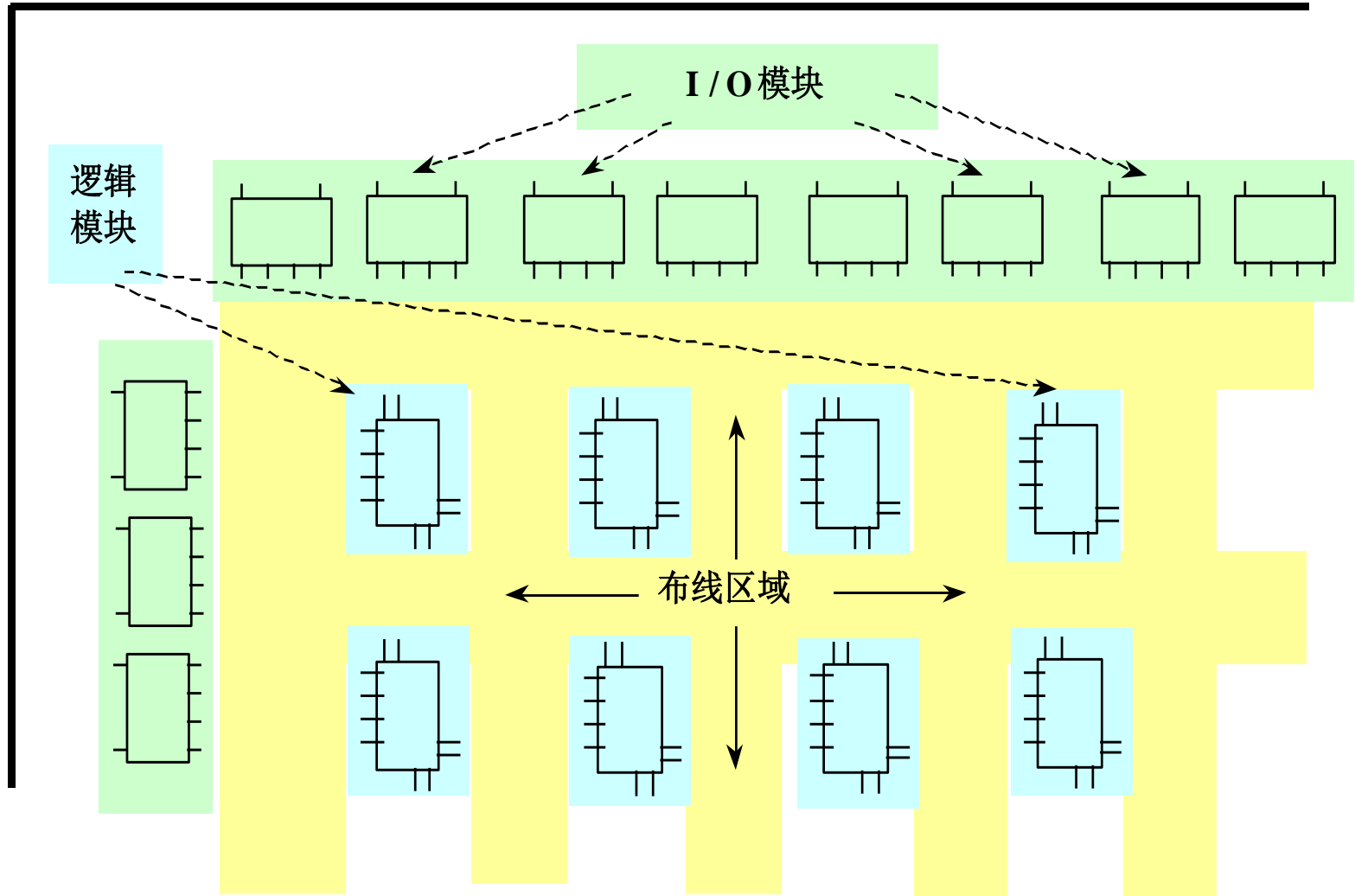
新增触发器和多路器



GAL 的原理示意图

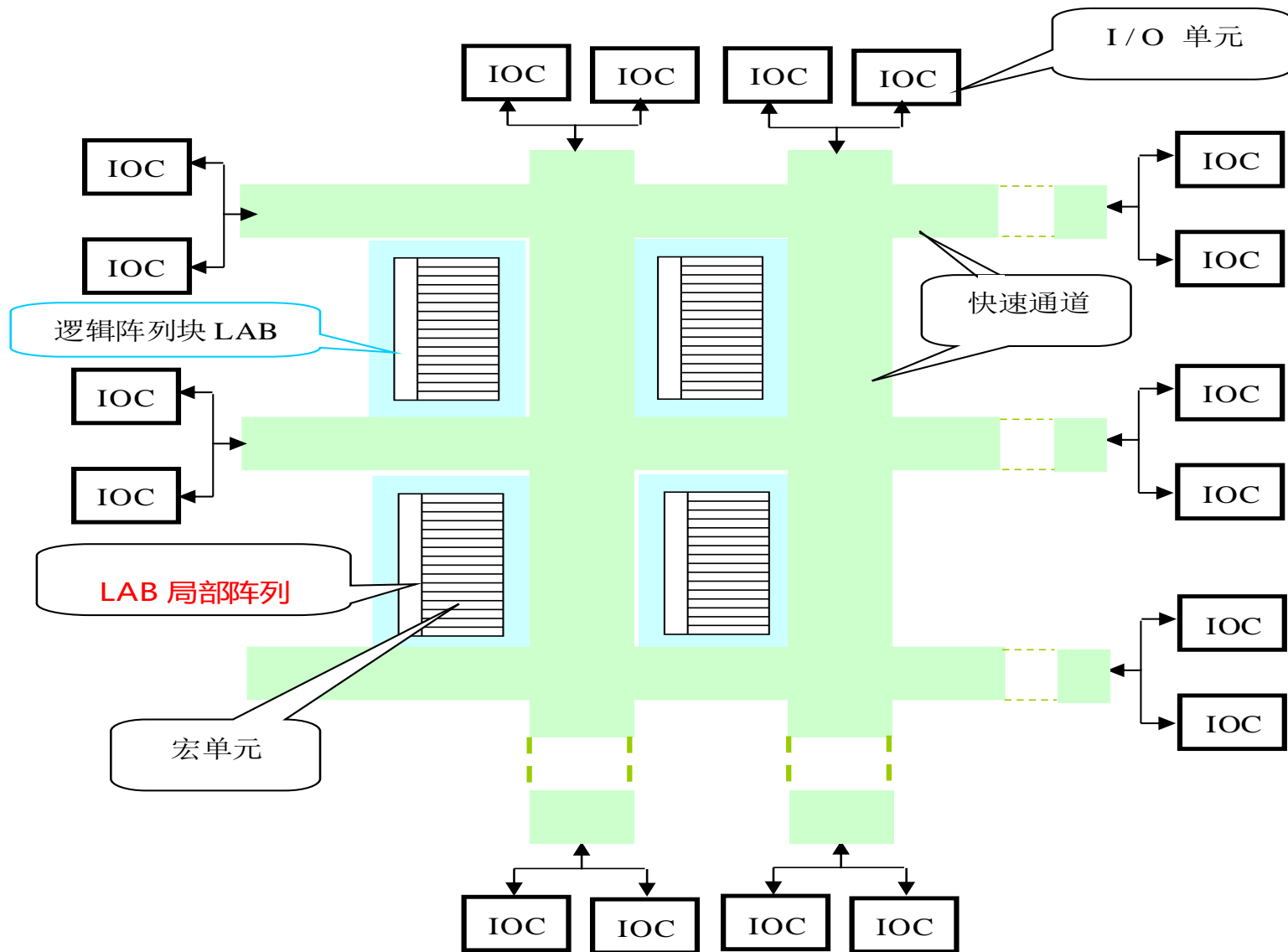


FPGA的结构示意图



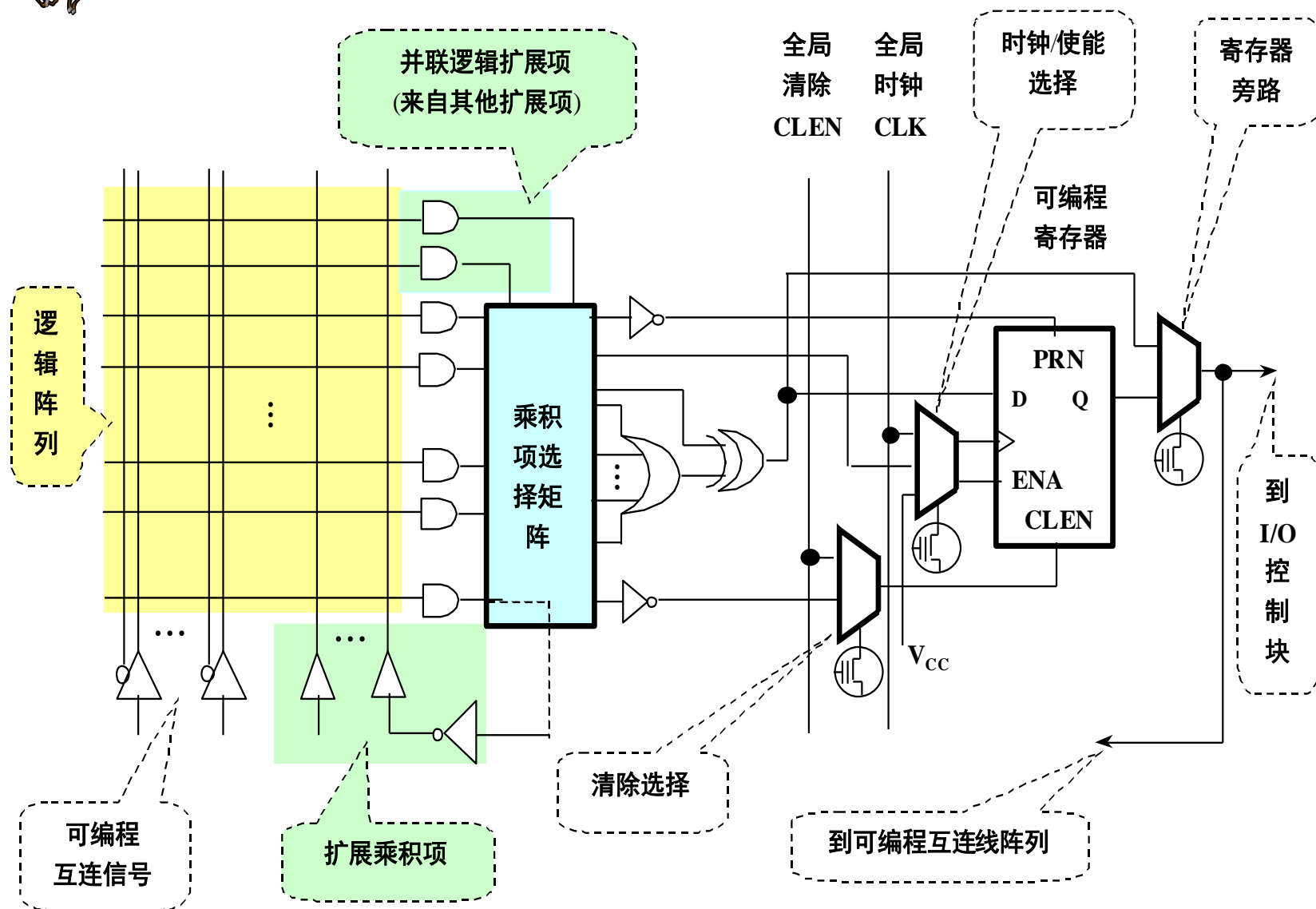


CPLD内部的结构示例



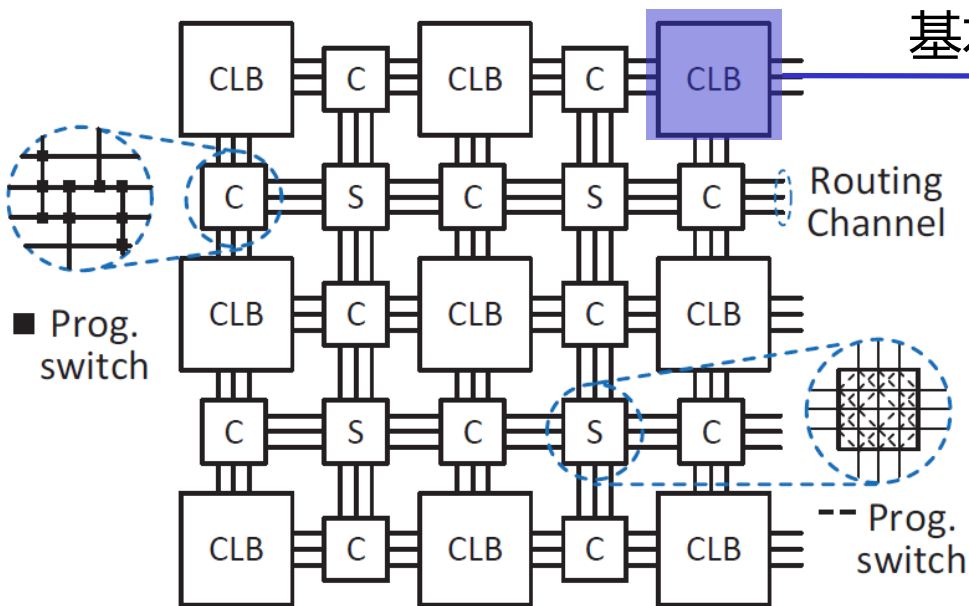


CPLD宏单元的内部结构示意图



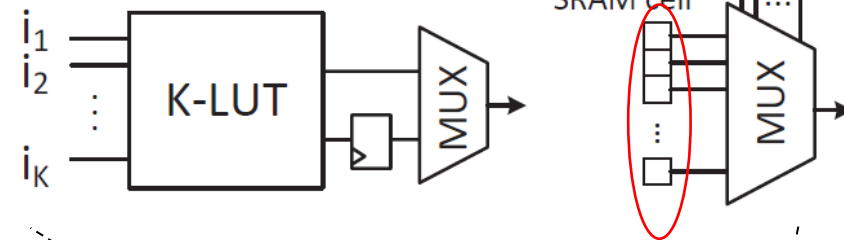


FPGA: Field-Programmable Gate Array

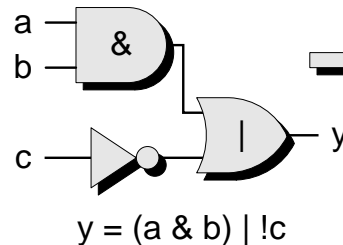


K-LUT: K输入查找表

函数结果



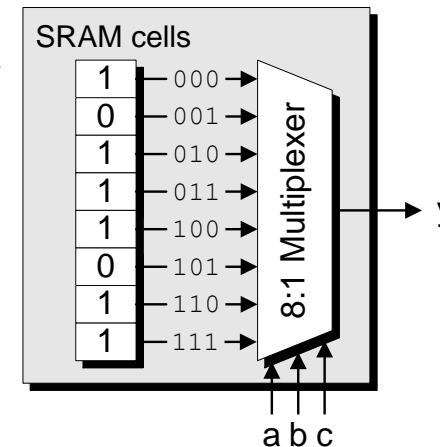
Required function



Truth table

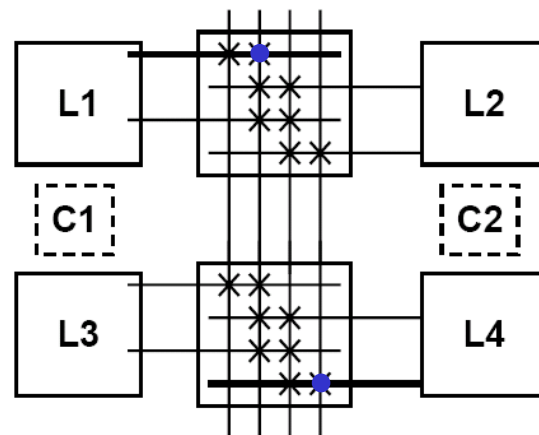
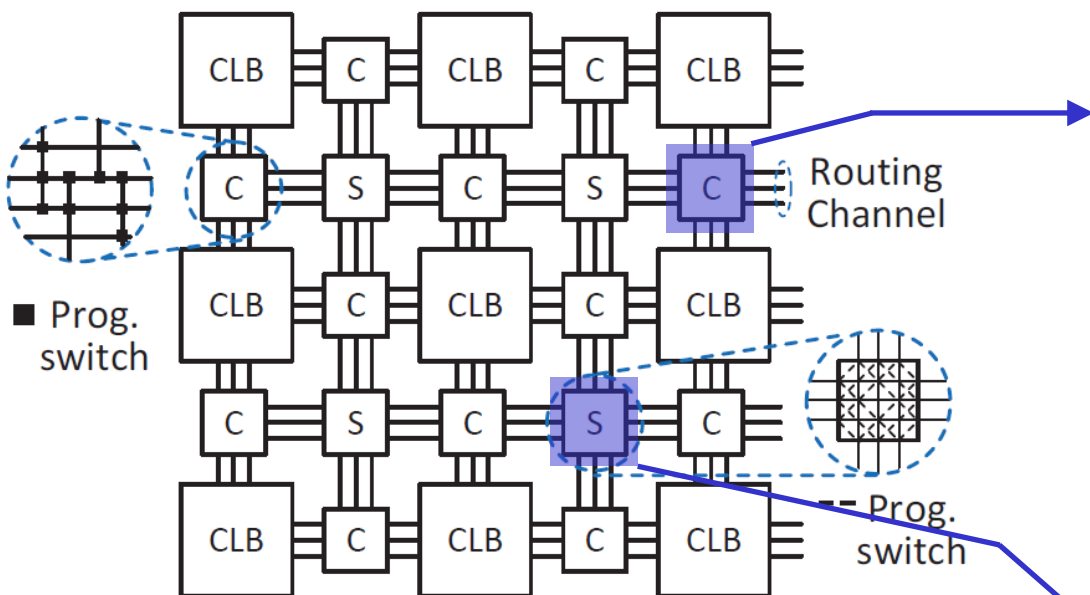
a	b	c	y
0	0	0	1
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	1

Programmed LUT

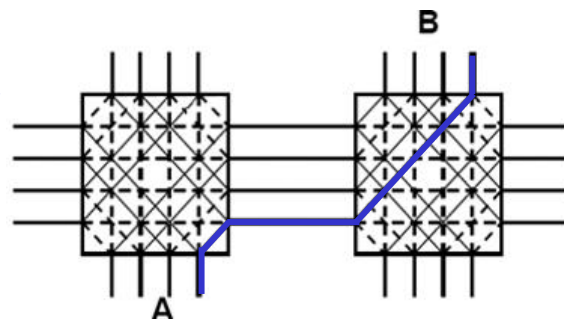




FPGA: Field-Programmable Gate Array



连接块：选通LUT输入输出、确定连接位点



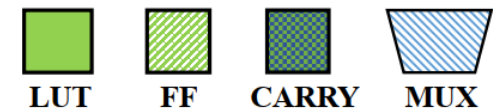
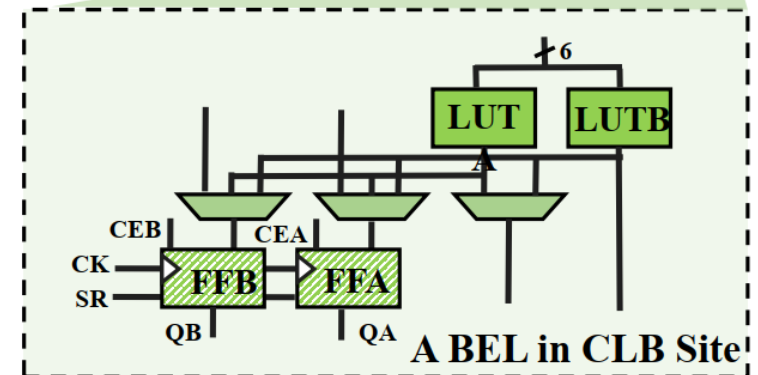
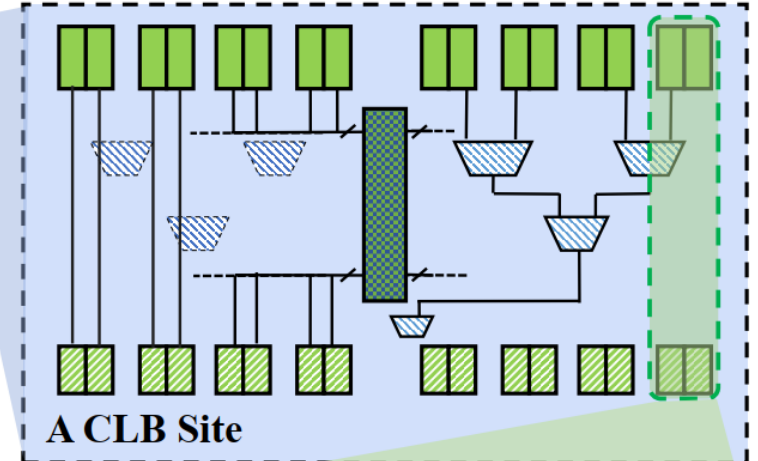
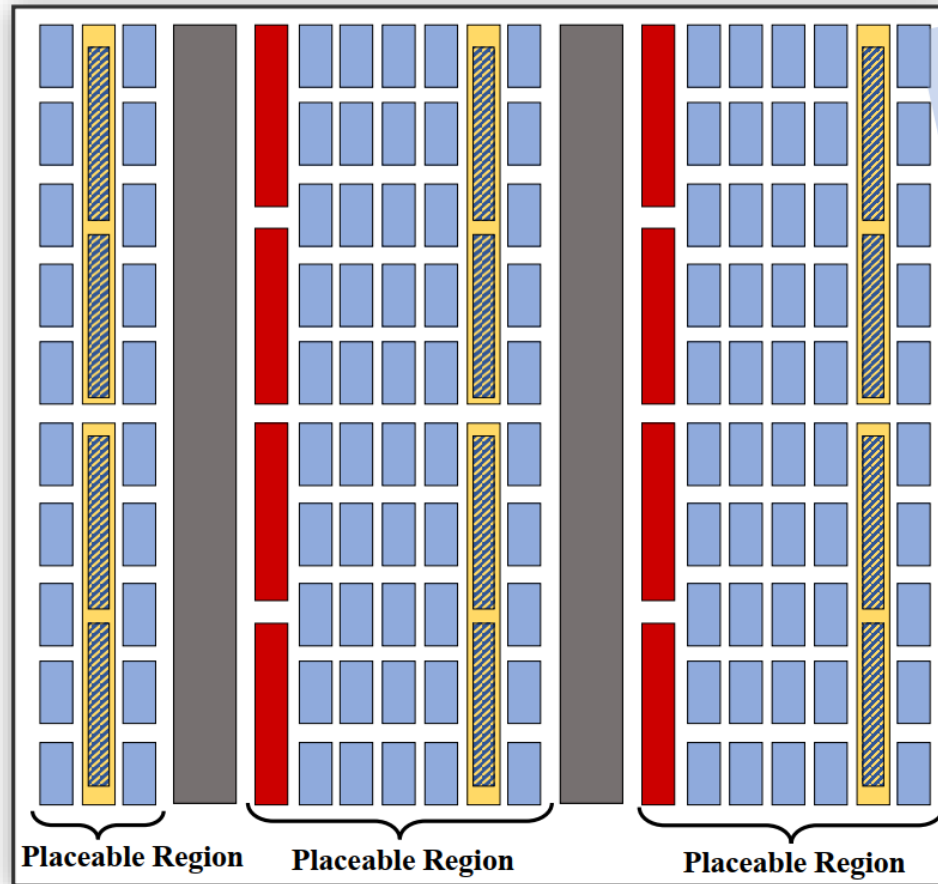
转换块：实现连接位点之间的纵横物理互联

FPGA硬件结构

- 可重构逻辑 + 可重构互联



FPGA: Field-Programmable Gate Array



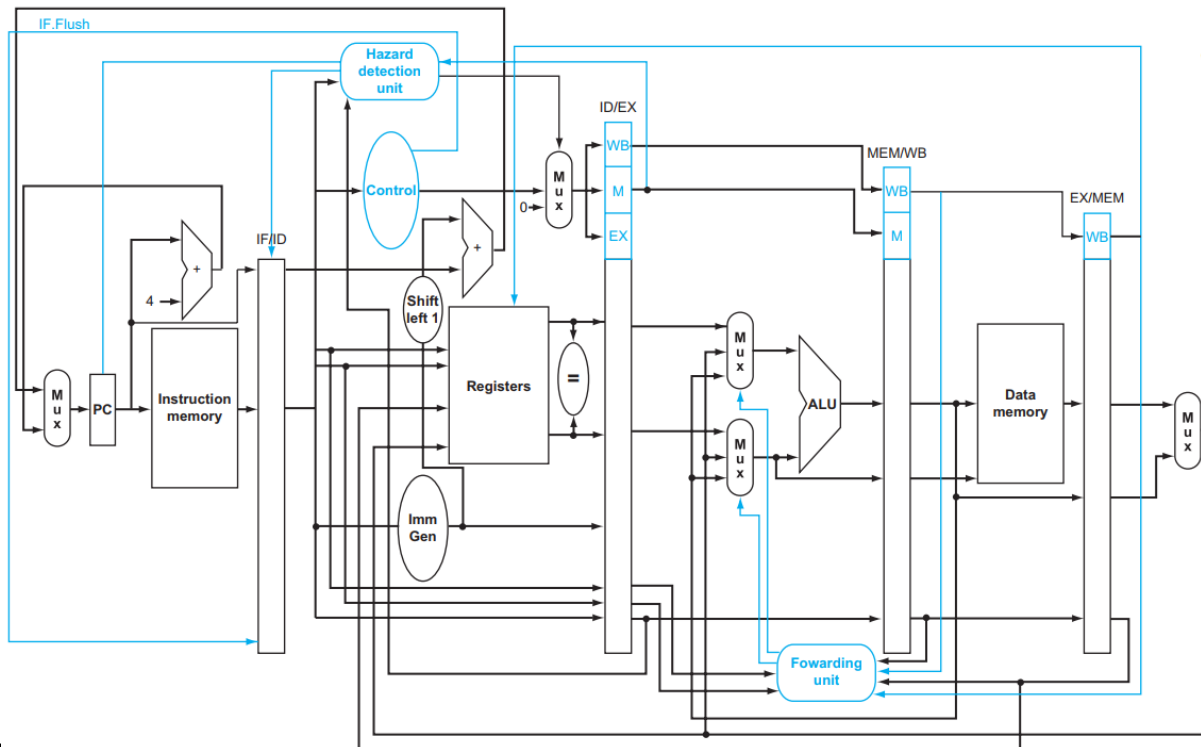


CPU: Central Processing Unit

存储程序控制

```
36  sub  x10, x4, x8
40  beq  x1,  x3, 16
44  and  x12, x2, x5
48  or   x13, x2, x6
52  add  x14, x4, x2
56  sub  x15, x6, x7

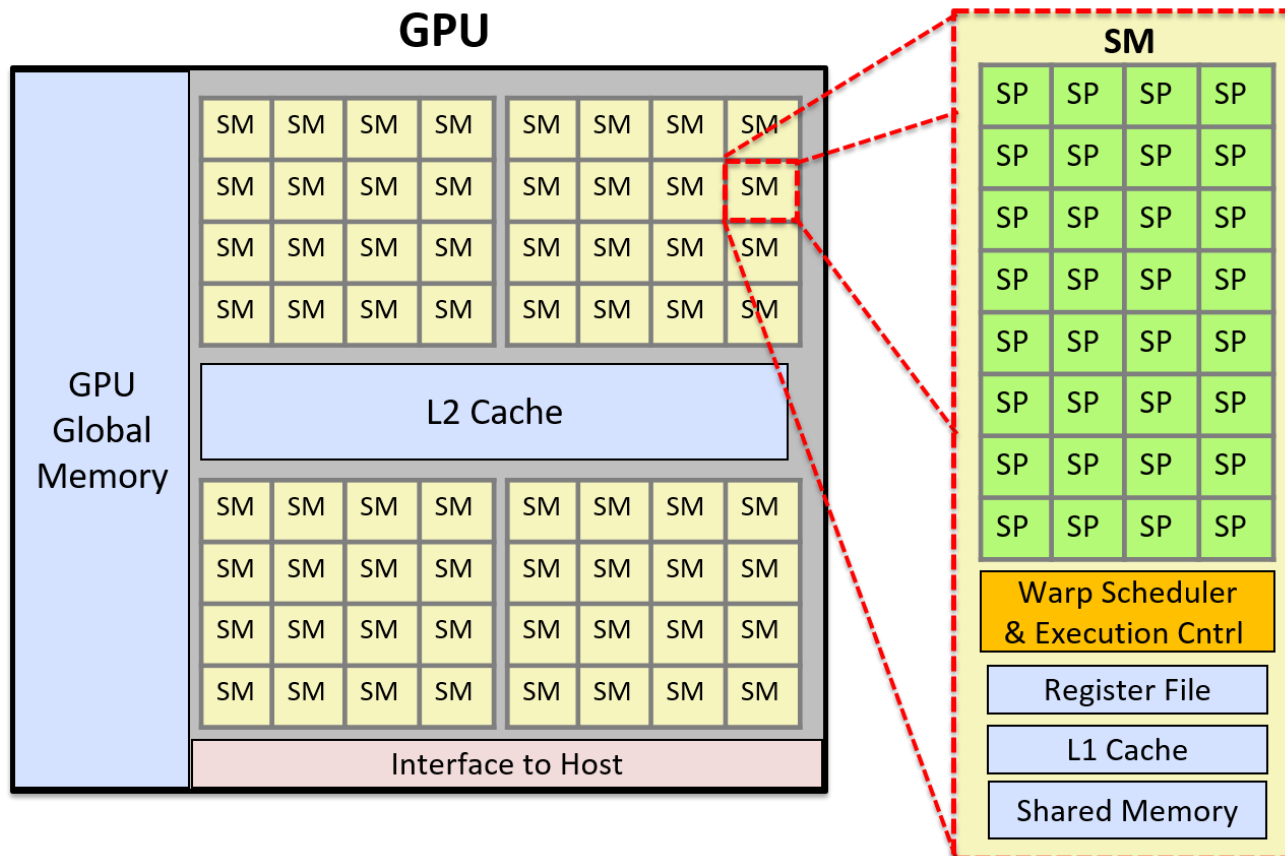
    . . .
72  ld   x4, 50(x7)
```





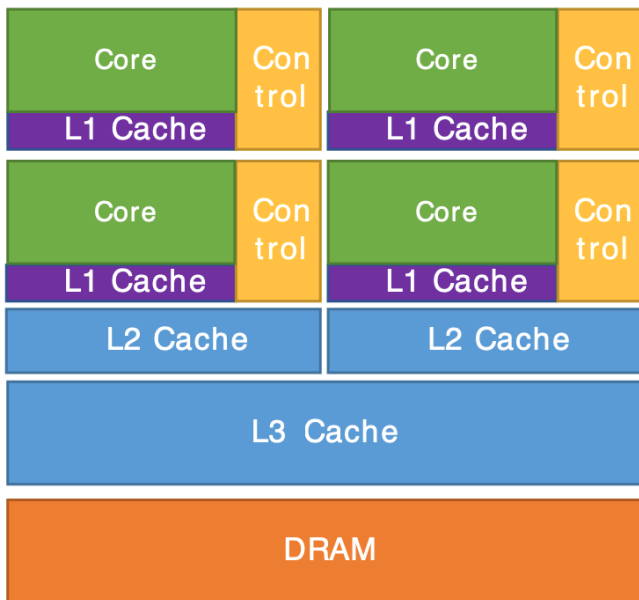
GPU: Graph Processing Unit

- **SM**: streaming multiprocessors
- **SP**: streaming processor
- 图像算法: 中值滤波、插值等



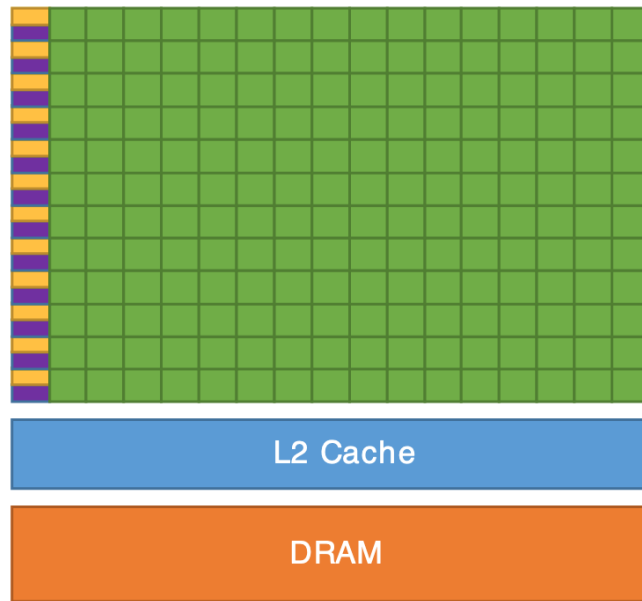


CPU



- 串行处理能力强
- 核心数量有限
- 通用性高
- 擅长处理复杂任务

GPU



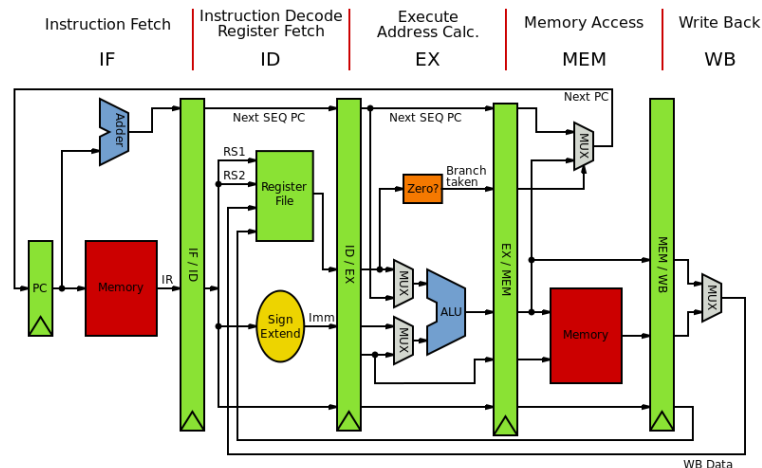
- 并行处理能力强
- 核心数量较多
- 视频处理、图形渲染、AI应用
- 同时处理多个小型任务



向量相加

```
void main(){  
    while(a[i]<b[i]){  
        c[i] = a[i] + b[i];  
        i++;  
    }  
}
```

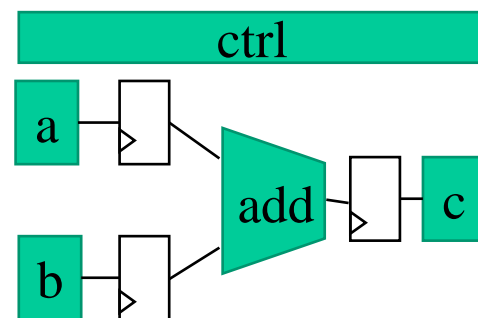
CPU



向量相加

```
void main(){  
    while(a[i]<b[i]){  
        c[i] = a[i] + b[i];  
        i++;  
    }  
}
```

ASIC/
FPGA



灵活性、通用性

性能、能效比

ASIC/FPGA v.s. CPU/GPU: 通过**硬件定制流水线/并行计算**，无需冗余控制/计算逻辑，实现**高性能和能效比**



集成电路分类

■ ASIC、FPGA上的设计是**硬件架构设计**

- 生成新的硬件结构
- 依赖于**EDA工具**
- ASIC的EDA与FPGA的EDA具有较大区别 → **why?**



■ CPU、GPU上的设计是**程序设计**

- 不生成新的硬件结构
- 编译器：C/C++等高级编程语言到汇编语言
- 汇编器：汇编语言到机器指令



ASIC EDA 三巨头



主攻领域：

- 逻辑综合工具DC
- 时序分析工具PT
-

Cadence

主攻领域：

- 版图设计工具
Virtuso
- 仿真验证 NC-
Verilog

SIEMENS

主攻领域：

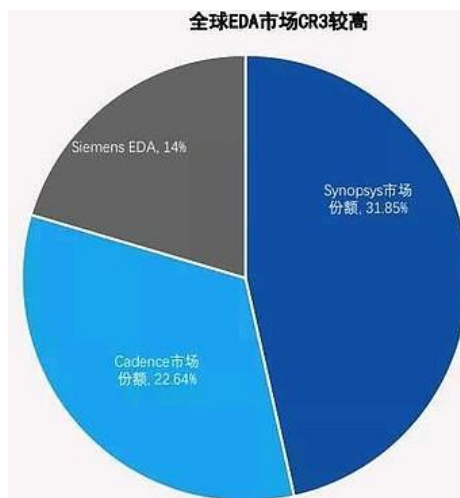
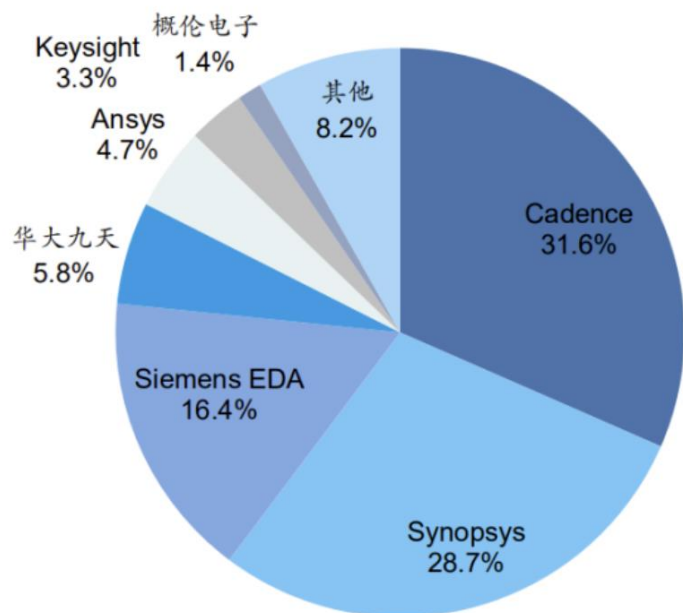
- 后端验证
- 可测试性设计
- 光学临近修正
-



ASIC EDA 三巨头

三巨头占据中国市场份额的~90%

占据全球市场份额的~75%





FPGA EDA 二巨头



2022年AMD收购Xilinx

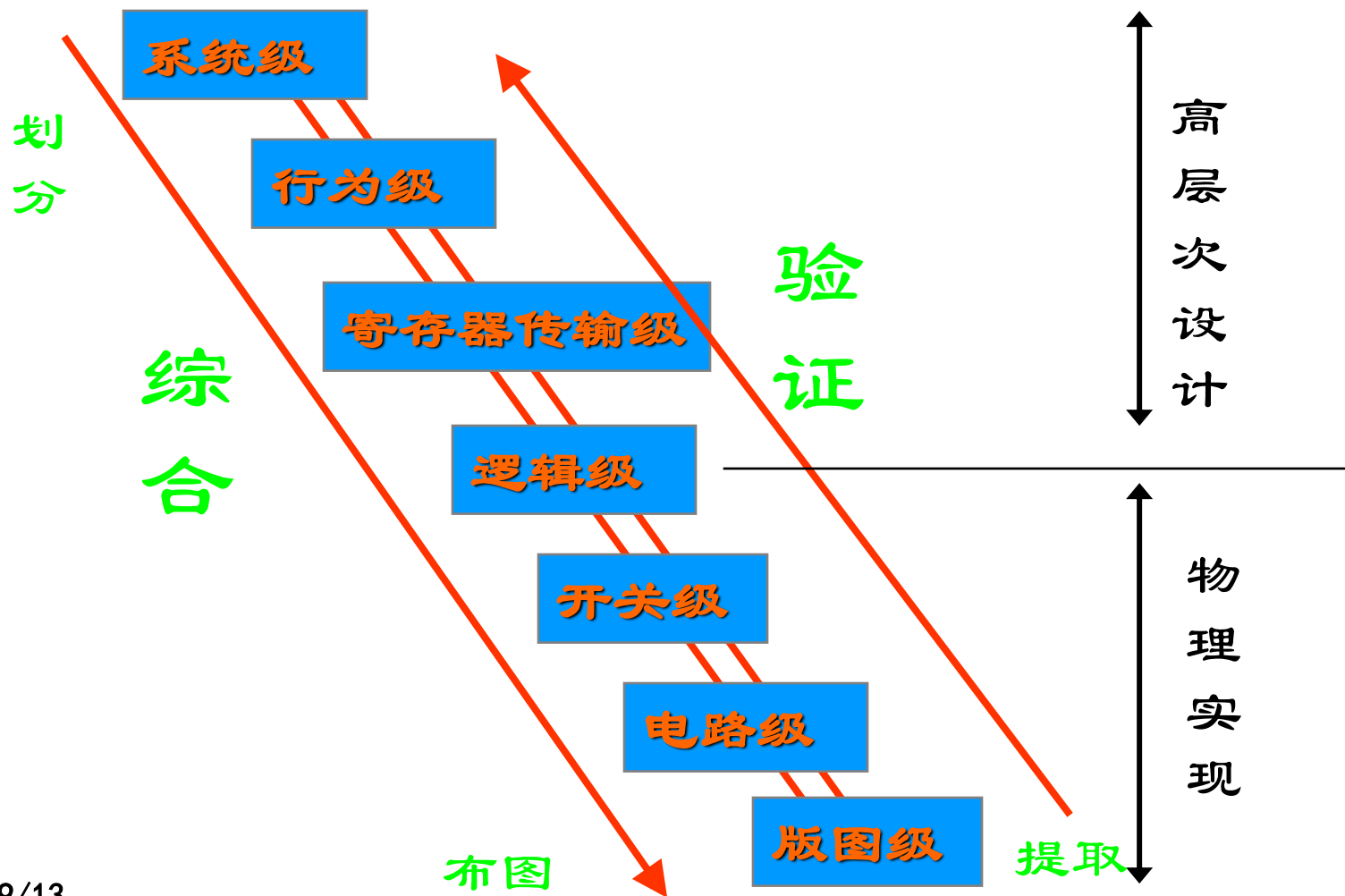


2015年Intel收购Altera

- 数据中心有两个主要趋势：第一，**数据流量**的增加；第二，**计算密集型应用程序**的增加。数据中心不仅必须能够处理大量数据，而且需要兼具**低功耗和高性能**。
- 数据中心对FPGA寄予厚望：FPGA的**可重构性和并行处理能力**可以无缝集成到现有的HPC基础设施中，补充传统的基于CPU的集群和基于GPU的系统。通过将特定任务卸载到FPGA，HPC系统可以实现更高的性能、更低的功耗并提高效率。
- 在数据中心，FPGA相比GPU的核心优势在于**延迟**。



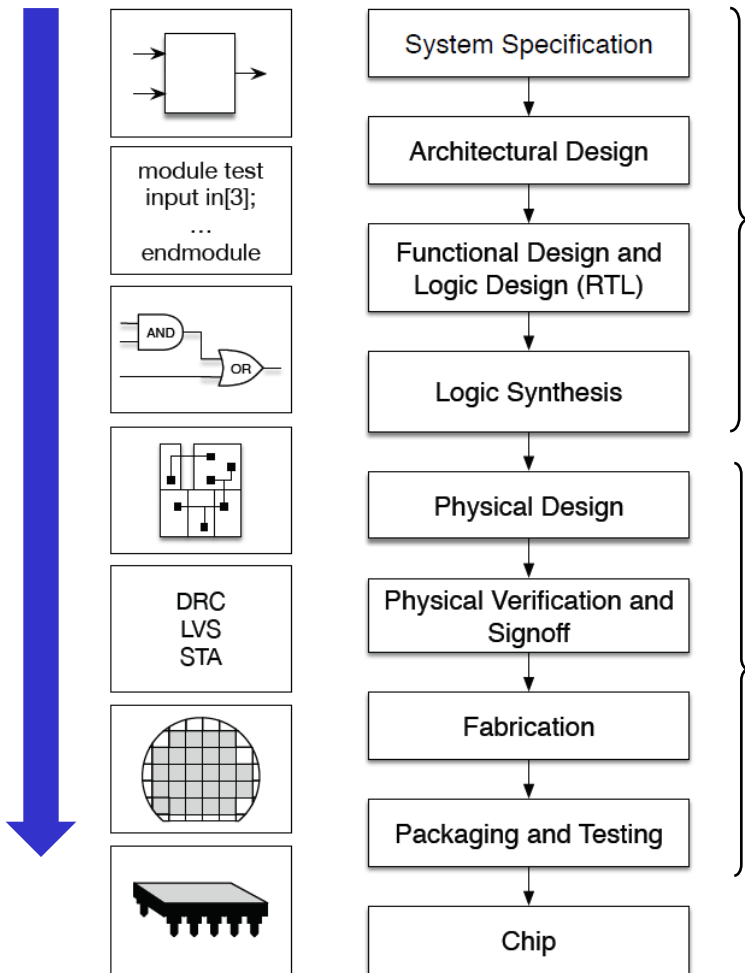
1.3 数字系统自动设计的流程





数字系统自动设计的流程

E
D
A



前端设计

- 架构设计
- 高层次综合
- 逻辑综合
- 功能仿真

.....

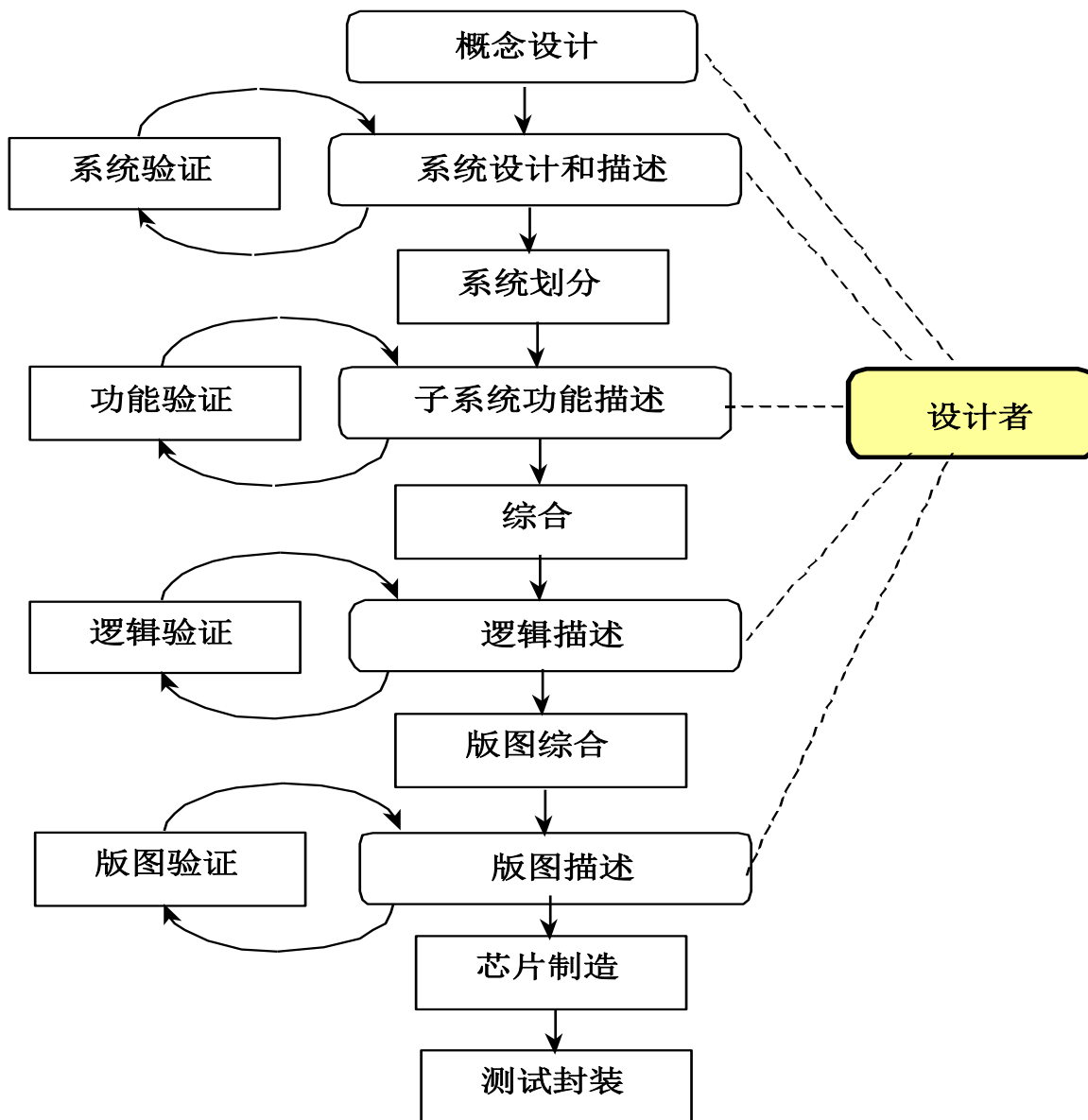
后端设计

- 布图规划
- 布局布线
- 时序分析
- 形式验证

.....

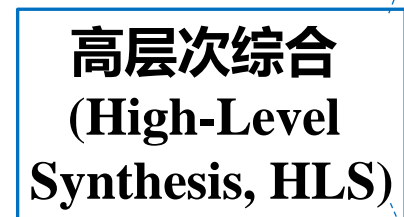
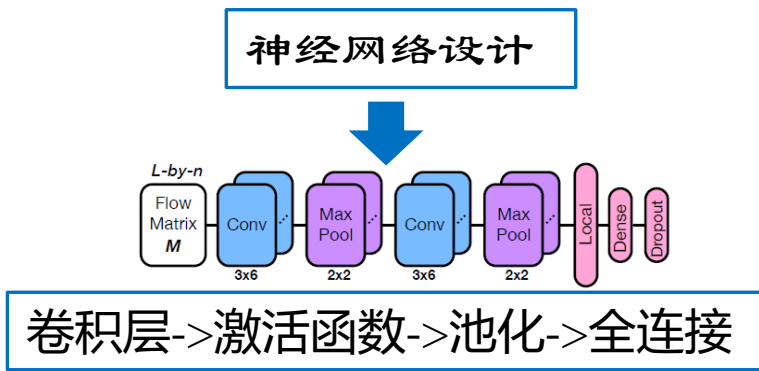
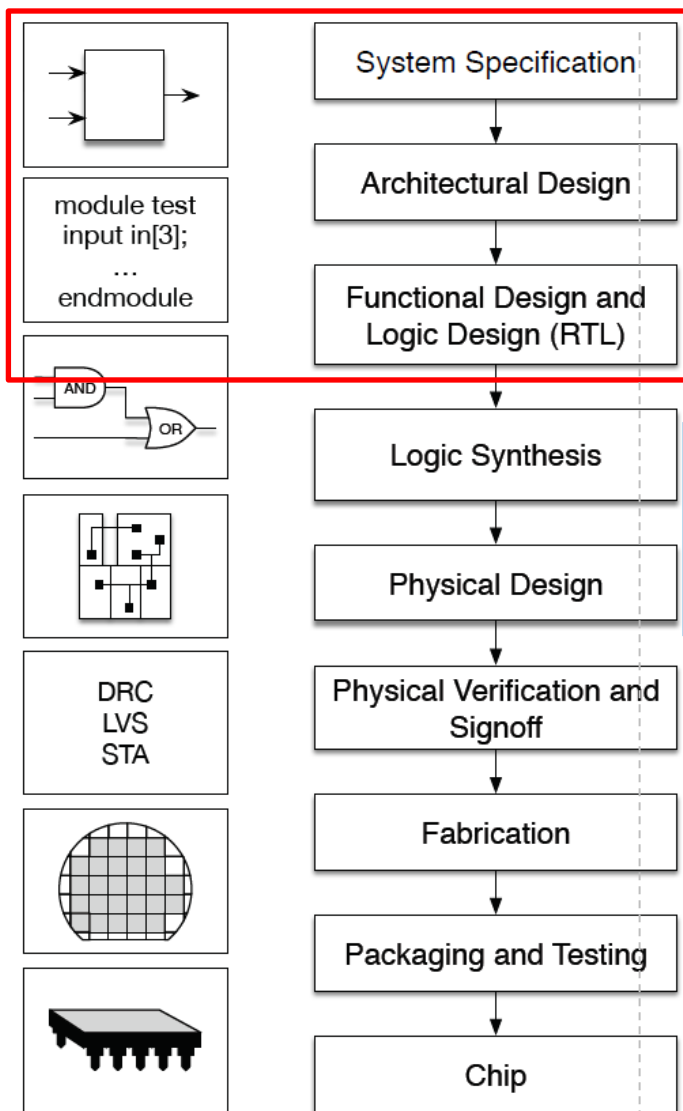


数字系统自动设计的流程



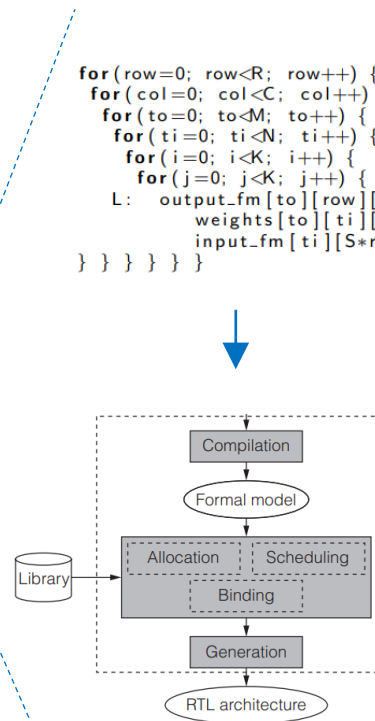
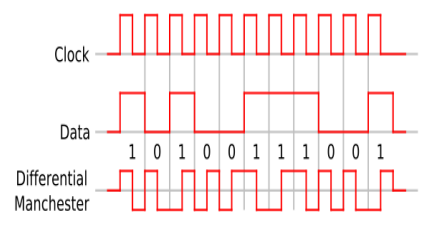


数字EDA：系统级综合



```
module conv;
reg [31:0] m[0:8192];
reg [12:0] pc;
reg [31:0] acc;
reg[15:0] ir;

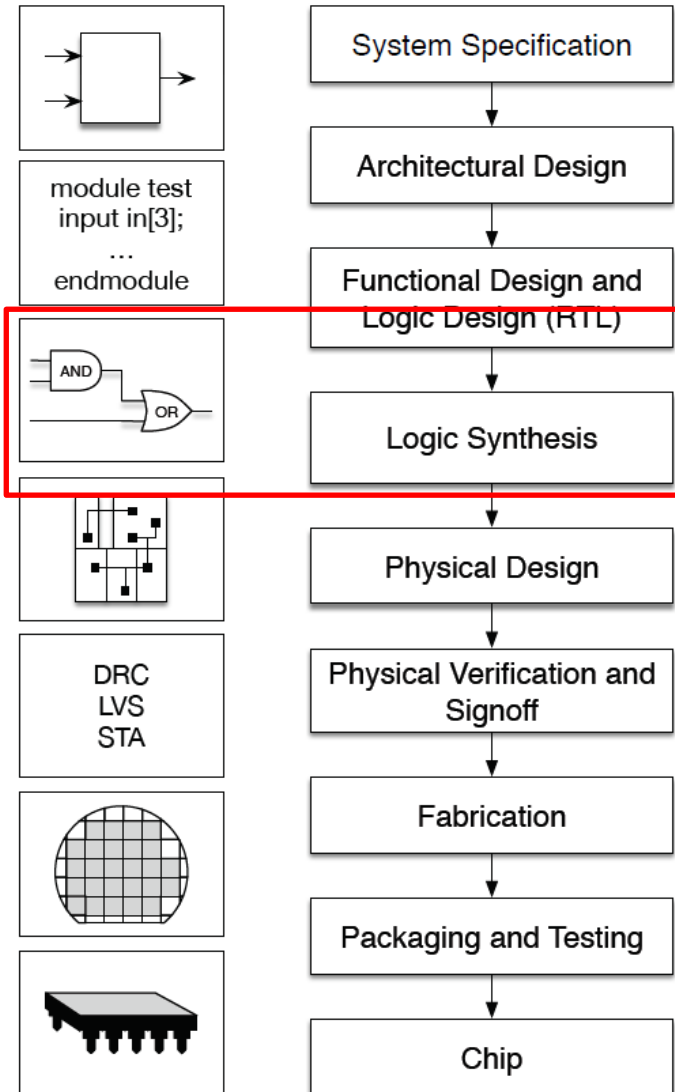
always
begin
ir = m[pc];
if(ir[15:13] == 3b'000)
pc = m[ir[12:0]];
else if (ir[15:13] == 3'b010)
acc = -m[ir[12:0]];
...
```



```
for(row=0; row<R; row++) {
for(col=0; col<C; col++) {
for(to=0; to<M; to++) {
for(ti=0; ti<N; ti++) {
for(i=0; i<K; i++) {
for(j=0; j<K; j++) {
L: output_fm[to][row]
weights[to][ti]
input_fm[ti][S*r
}}}}}
```



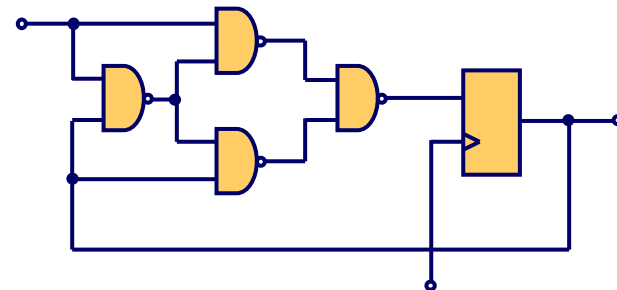

数字EDA：逻辑综合



```
module conv;  
  reg [31:0] m[0:8192];  
  reg [12:0] pc;  
  reg [31:0] acc;  
  reg [15:0] ir;  
  
  always@(posedge clk)  
  begin  
    ir = m[pc];  
    if(ir[15:13] == 3b'000)  
      pc = m[ir[12:0]];  
    else if (ir[15:13] == 3'b010)  
      acc = -m[ir[12:0]];  
    ...  
  end  
endmodule
```

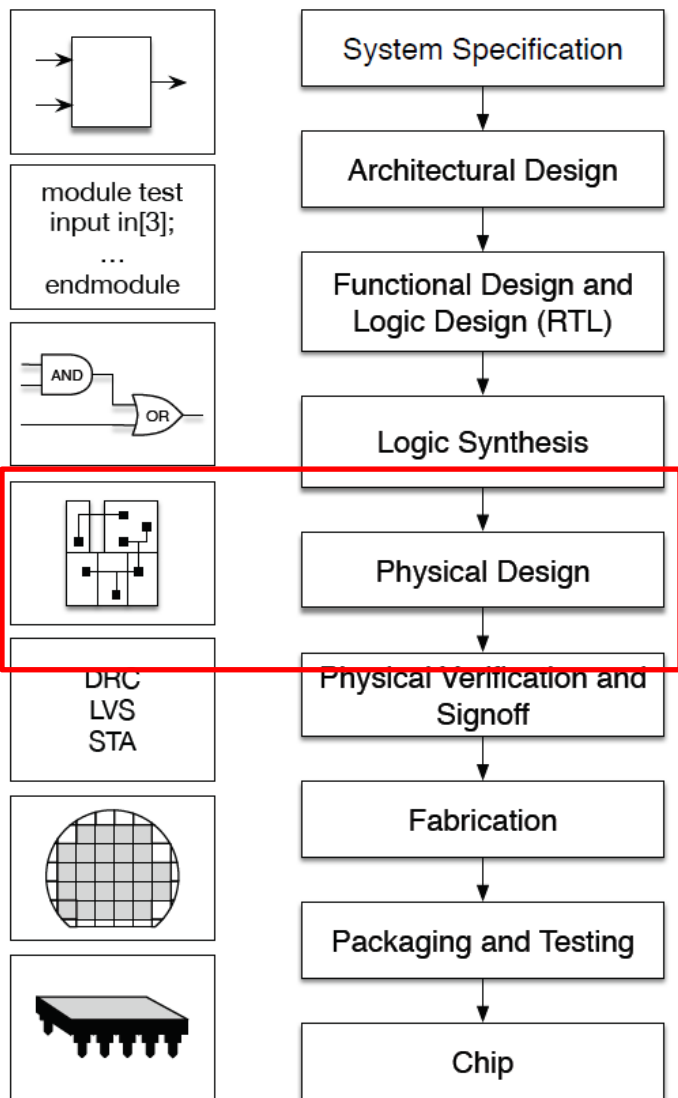


逻辑综合

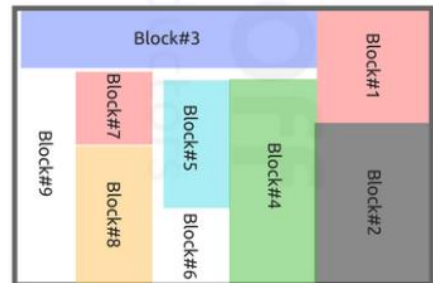




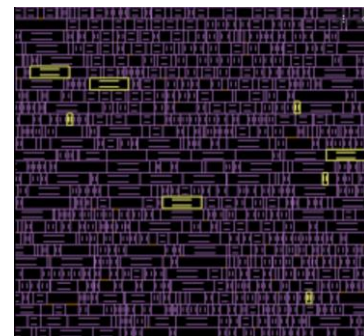
数字EDA：版图自动设计



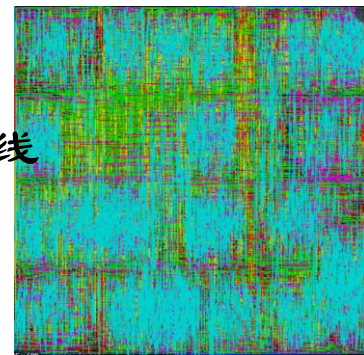
布图规划 (Floorplanning) :
确定各个大模块的位置



布局 (Placement) :
确定模块中不同元器件的位置



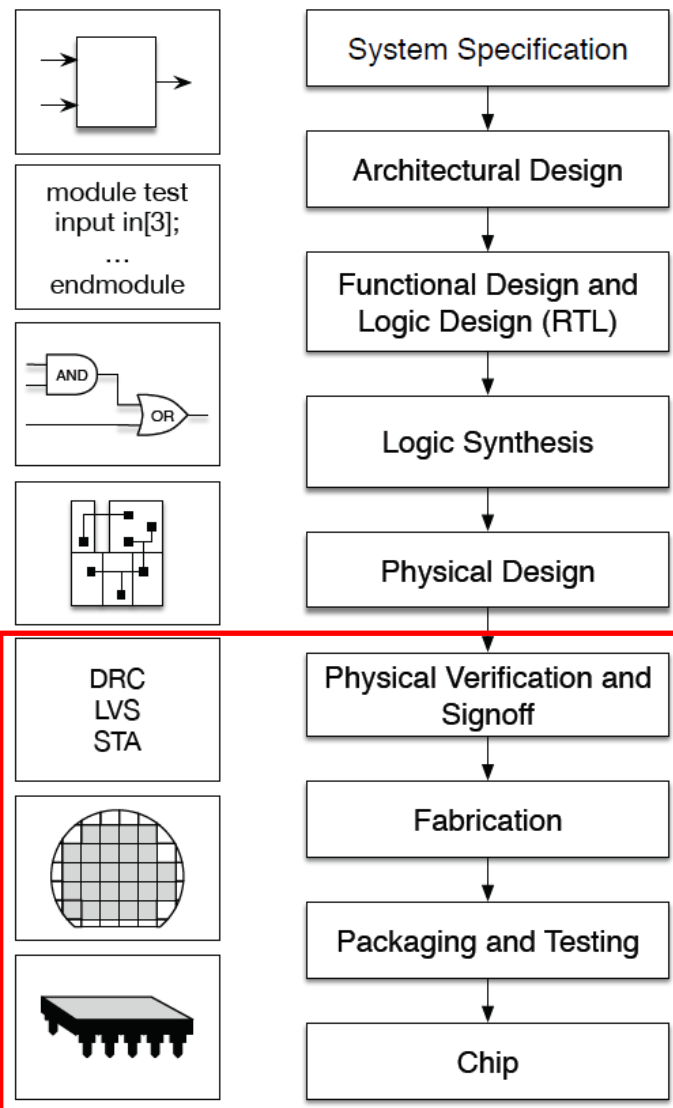
布线 (Routing) :
确定不同元器件之间的物理连线



注：此处列出关键步骤，完整流程还包含时钟树综合 (clock tree synthesis) 等



数字EDA：制造



设计验证、签核

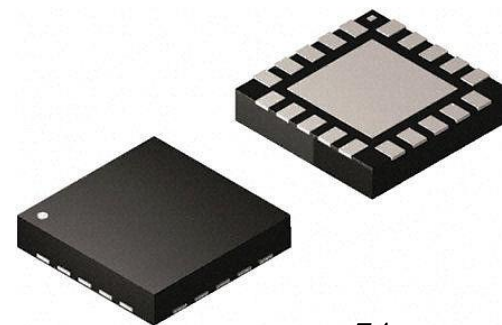
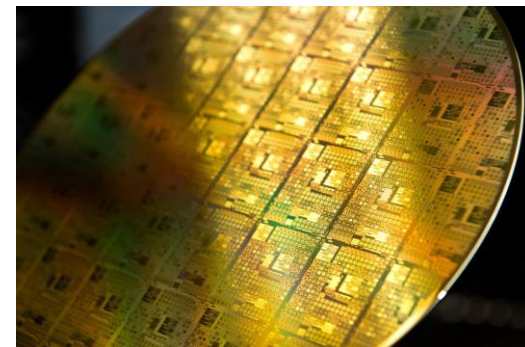
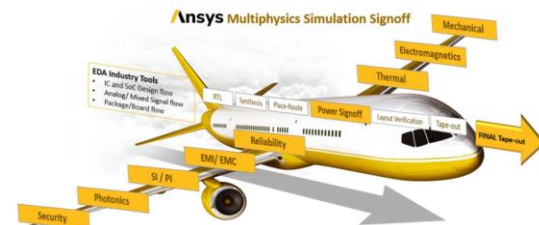
功耗、噪声、散热、静电等确认满足要求



芯片制造



封装测试





1.4 EDA的主要领域

- ◆ 设计输入 (Design Entry)
 - 硬件描述语言
(Hardware Description Language)
 - 交互式图形输入
(Interaction Graphical Design Entry)
- ◆ 综合 (Synthesis)
 - 高层次综合
 - 寄存器传输级综合
 - 逻辑综合 (逻辑最小化, 逻辑优化)



◆ 验证 (Verification)

- 模拟验证 (仿真) (Simulation)
- 形式验证 (Formal Verification)
- 设计规则检查 (Design Rule Checking, DRC)
电路提取, 功能提取

◆ 测试 (Testing)

- 故障诊断
- 测试码生成
- 可测性设计 (Design for Test, DFT)



◆ 布图 Layout

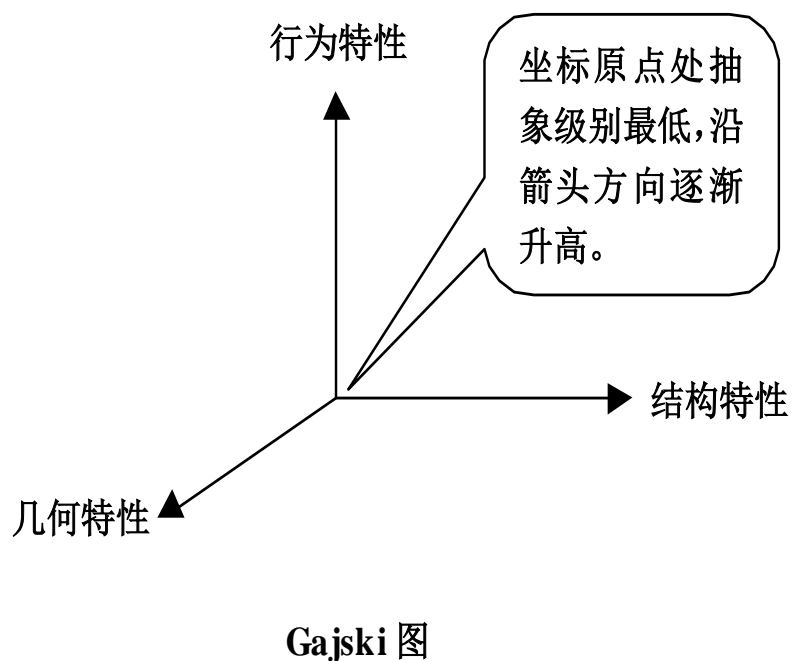
- 布局 (Placement) ;
- 布线 (Routing)

◆ 版图参数提取和验证 (Parameter Extraction)

- 寄生电阻、电容、电感



1.4.1 硬件描述语言



HDL 描述与应用目的相互关系

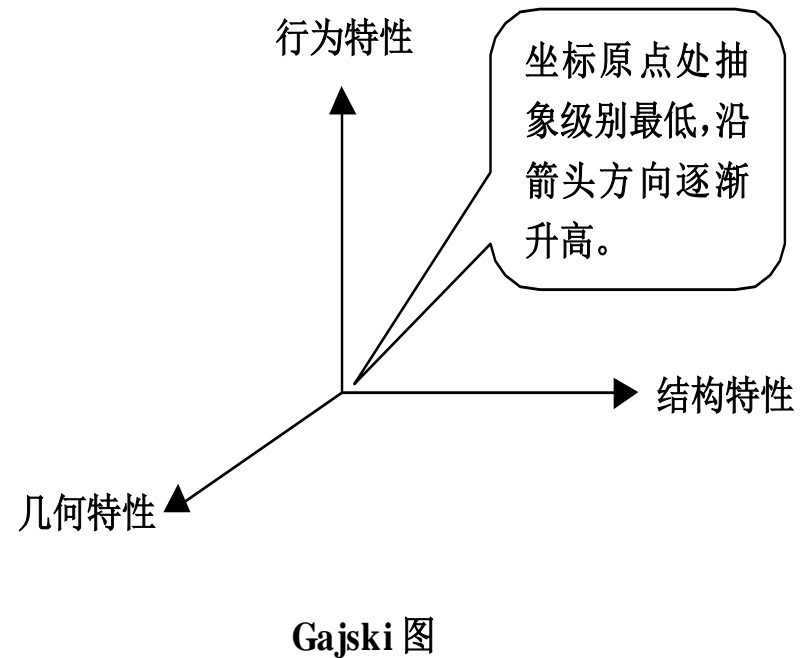
应用 \ 特性	行为	结构	物理
逻辑模拟	○	★	○
逻辑综合	★	○	○
PCB 布线		○	★
芯片布线		○	★

说明: ★表示有此项要求; ○也表示有此项要求, 但重要性次于★。



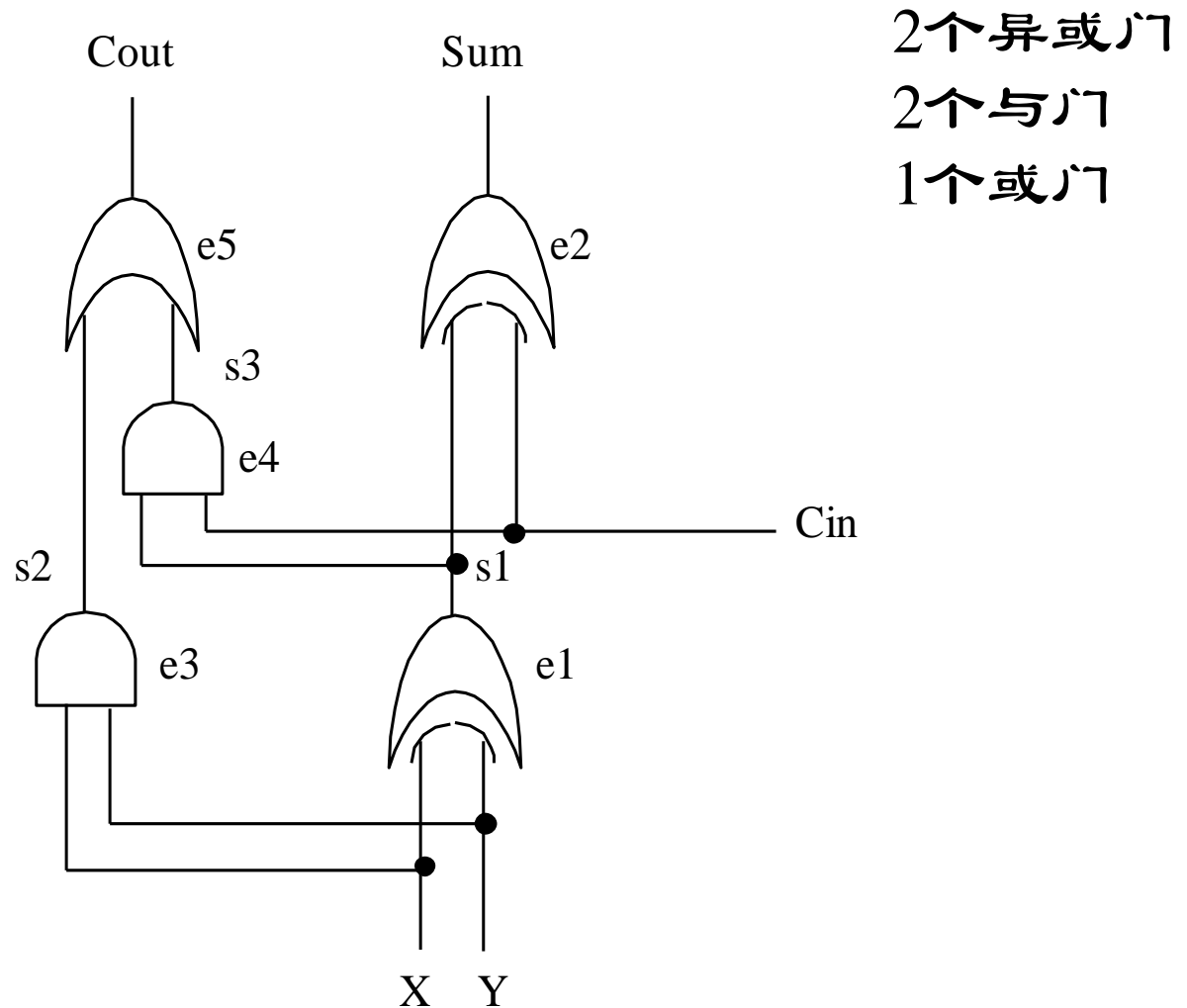
结构描述中的抽象级别:

- ◆ 寄存器传输级 (Register Transfer Level, RTL) ;
- ◆ 功能块级 (Functional Block Level) ;
- ◆ 门级 (Gate Level) ;
- ◆ 开关级 (Switch Level) ;
- ◆ 电路级 (Circuit Level) ;
- ◆ 版图级 (Layout Level) ;





1位加法器的结构描述 (原理图)





1位加法器的结构描述

1位全加器的结构模型（示意，并非严格的VHDL描述）：

architecture STRUCT of ADD is:

block

Component: and2, xor, or2 ;

begin

xor: s1 = e1 (x, y) ;

sum = e2 (s1, cin) ;

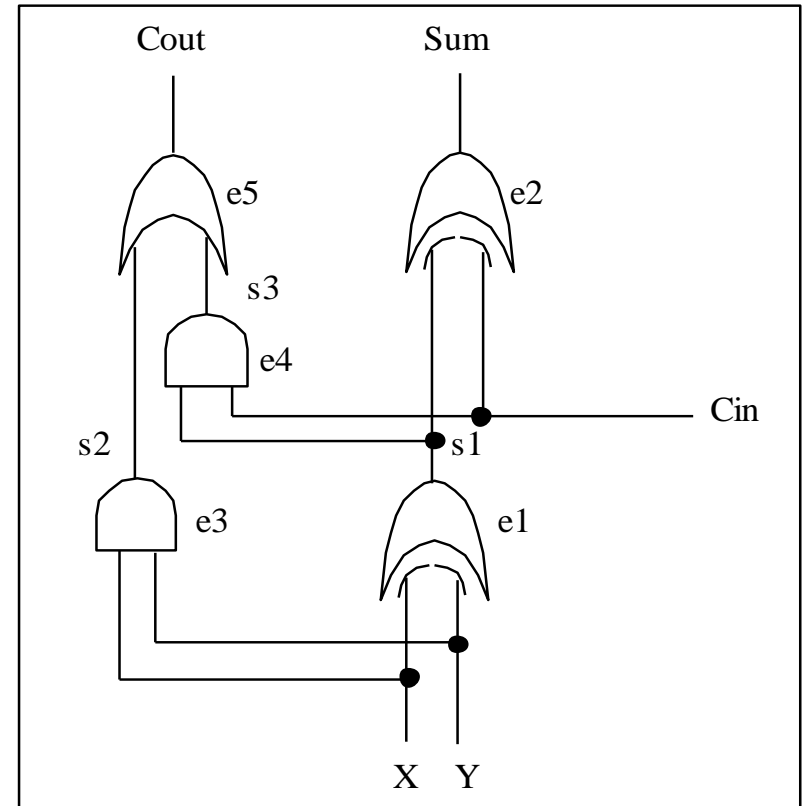
and2: s2 = e3 (x, y) ;

s3 = e4 (s1, cin) ;

or2: cout = e5 (s2, s3) ;

end block ;

end STRUCT ;





1位加法器的行为、结构混合描述

-- Behavioral MODEL with implied STRUCTURE

achitecture MIXED of ADD is:

block

signal s1, s2, s3, cout: bit;

begin

s1 <= X **xor** Y;

sum <= s1 **xor** cin;

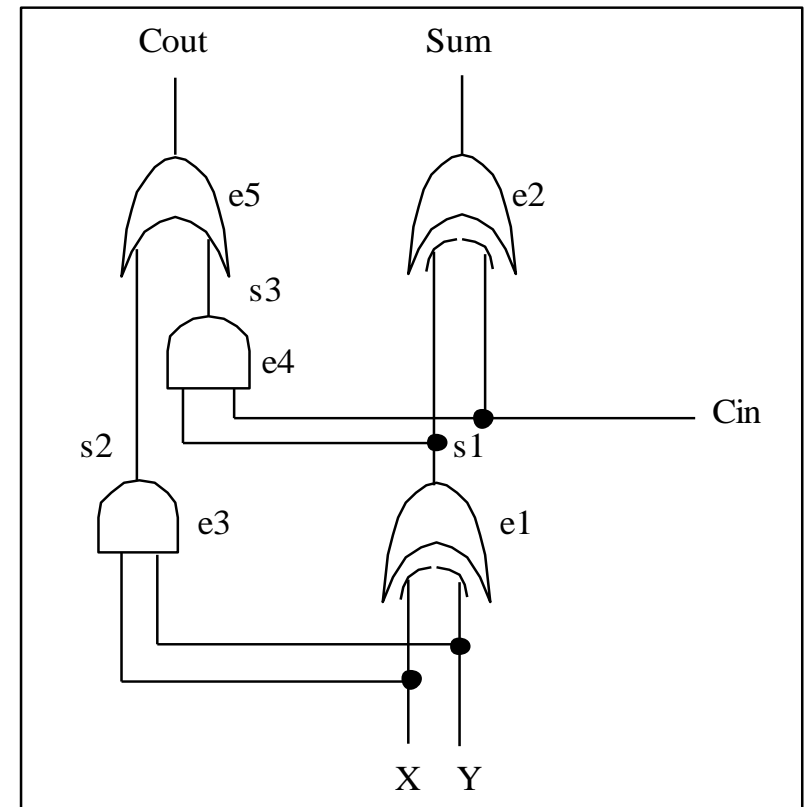
s2 <= X **and** Y;

s3 <= s1 **and** cin;

cout <= s2 **or** s3;

end block;

end MIXED;





4 位加法器的行为描述

- - ABSTRACT BEHAVIORAL MODEL

architecture TOP of ADDER is:

block

begin

process (A, B, SUM)

variable A, B: integer range 0 to 3;

variable SUM: integer range 0 to 4;

begin

SUM := A + B;

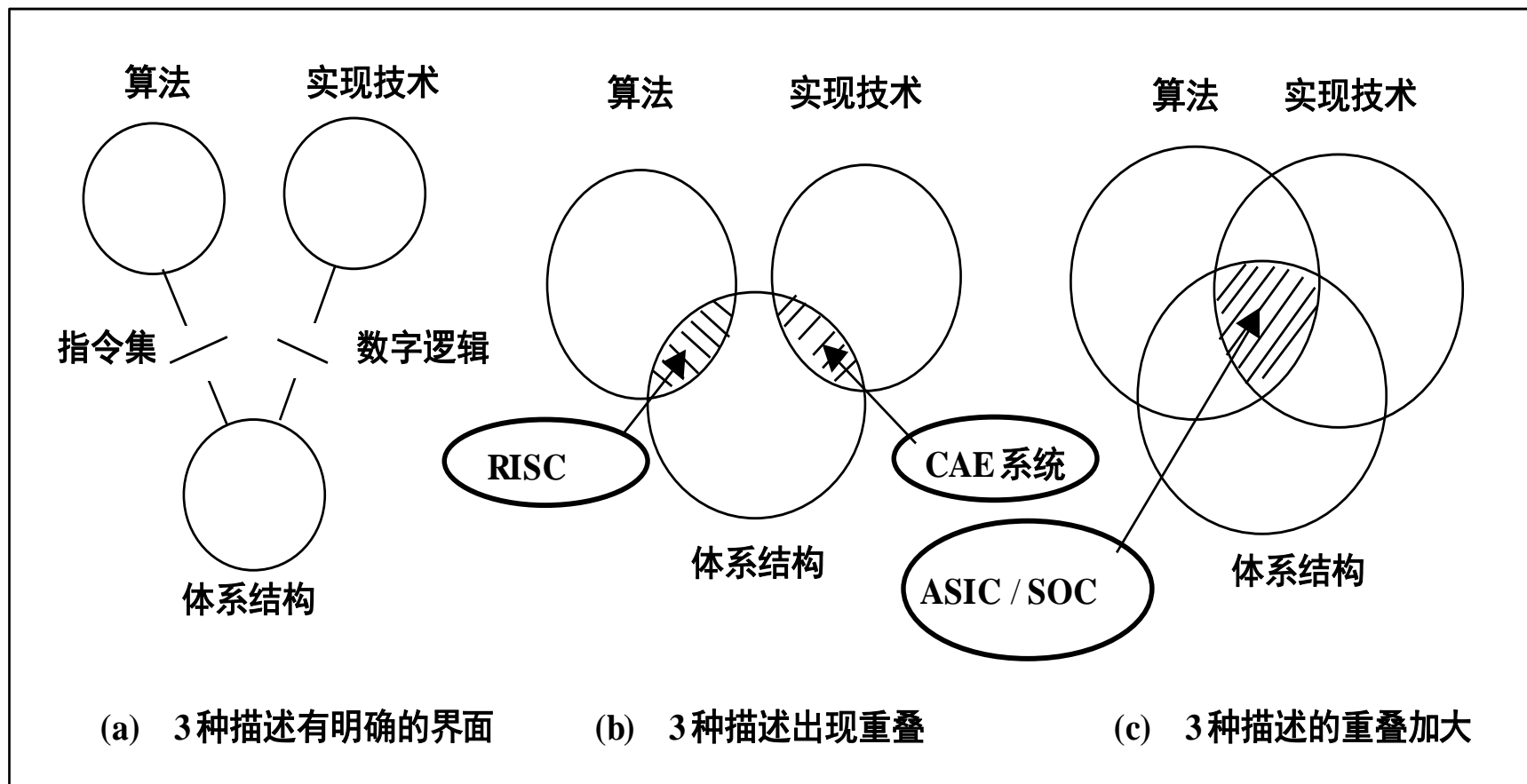
end process ;

end block ;

end TOP;



不同级别的描述逐渐重叠





1.4.2 模拟技术

- ◆ **设计者必须给模拟器提供以下信息：**
 - 数字系统基本元件的功能特性。
 - 基本元件的互连关系或相互作用的关系。
 - 模拟过程所需的信息，
 - 如外部激励信号的名称
 - 波形、观察点的名称以及如何表示被观察信号的命令等。
- ◆ **模拟器提供该电路模型在激励信号作用下的反应（各点波形）。**
- ◆ **设计者分析模拟器提供的波形，自己判断该电路模型的功能是否和预期相符。**
- ◆ **模拟器的优点：**不需要将电路模型实际制造出来，就可以进行调试；
- ◆ **模拟器的缺点：**
 - 设计者的责任还比较多；
 - 模拟器需要的资源（Memory, CPU时间）很大；



1.4.3 综合技术

- ◆ 由**设计**要求到设计**实现**的整个过程，如果是靠人工完成，通常简单地称之为设计；如果**依靠EDA工具自动生成**，通常称之为**综合**。
- ◆ 设计方案必须**满足预期功能和约束条件**的要求。约束条件包括：
 - 成本（Cost）；
 - 性能（Performance）；
 - 功耗（Power）；



1.4.4 测试诊断

◆ 难点所在：

外部激励信号只能从输入引脚引入，观察点只能从输出引脚引出，这就增加了测试的难度到输出端，需要设法在外部激励信号的作用下，把内部元件的故障“驱赶”，以便能在输出引脚上观察到故障的存在。

◆ 主要内容：

- 故障测试；
- 故障诊断（故障定位）；
- 可测性设计



1.4.5 形式验证

◆ 模拟技术存在以下缺点：

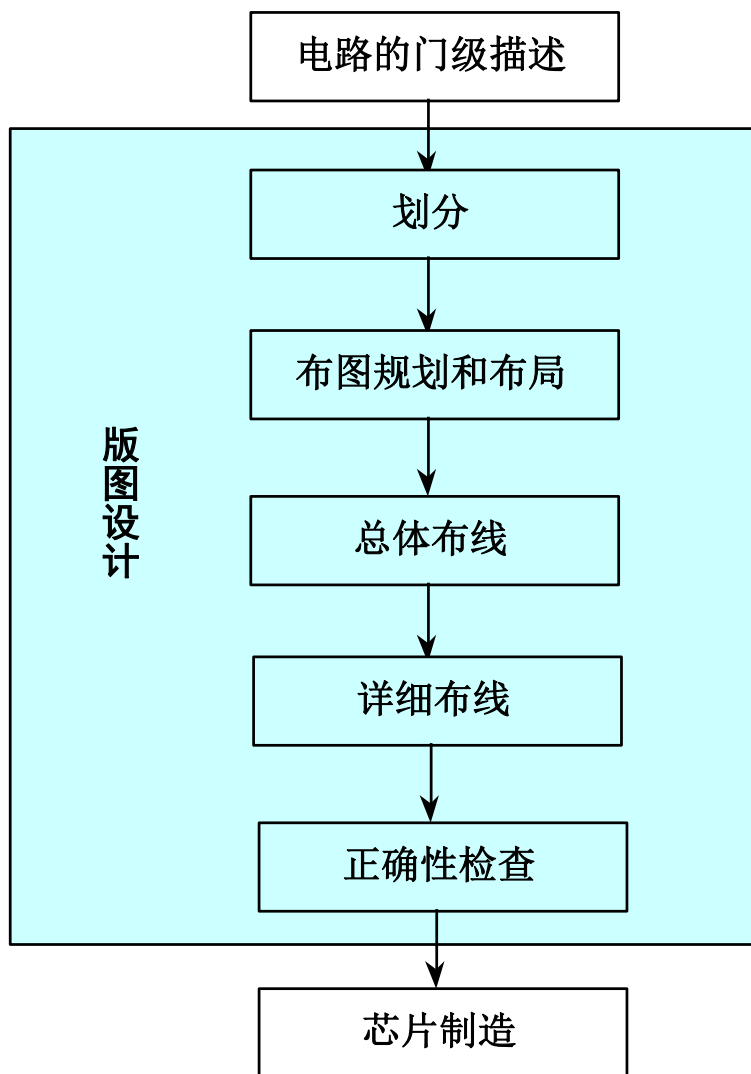
- 外加激励信号决定所能查出问题的多少，而它要由用户给出，
- 输出结果乃是外加激励信号作用下该数字系统的反应，并不表示正确与否。
- 判断该数字系统是否正确的责任落在设计者本人的肩上。
- 受给定的外加激励信号质量的影响及设计者判断能力的影响，很难保证现存错误都能查出。
- 对计算机的资源需求较大。

◆ 形式验证：

- 对数字系统作结构描述，这是验证的对象。
- 对数字系统作功能描述，作为验证的目标。
- 验证的过程是：根据描述和一套公理系统证明该结构描述所实现的功能是否与相应的功能描述等价。



(6) 版图实现





1.5 集成电路快速发展对设计自动化的挑战与机遇

◆ 1.5.1 工艺微细化

- 问题：对性能的影响：

时延：连线延迟 \gg 元件延迟

- 要求：系统设计阶段考虑连线时延的影响

综合与布图相结合

◆ 1.5.2 电路规模膨胀

- 问题：算法复杂性影响严重

- 要求：解决NP难题：

- 启发式方法, 简化模型, 新算法
- 新方法: 形式验证方法

- 对硬件的要求：

- 速度快, 容量大, 多CPU并行计算

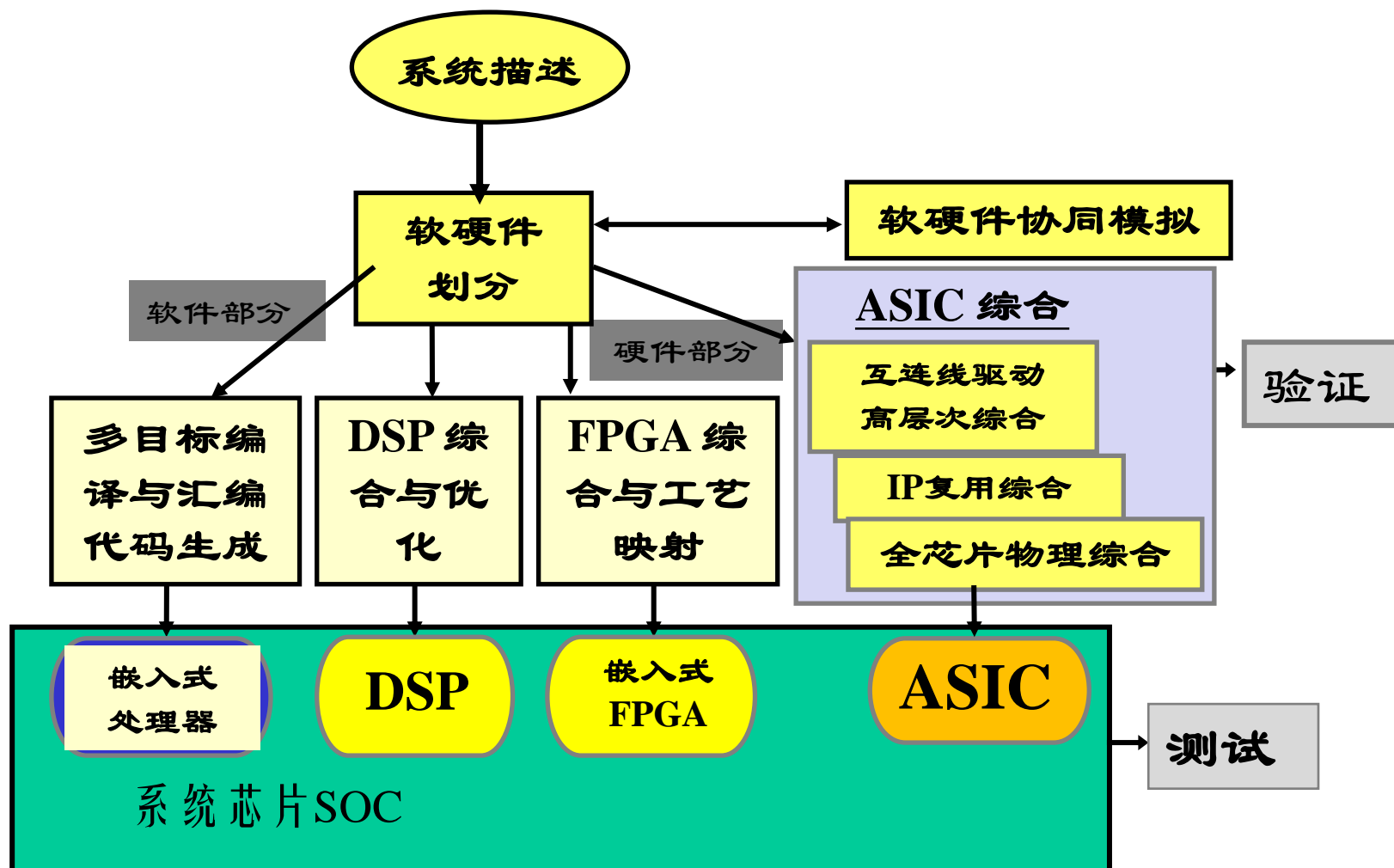


◆ 1.5.3 电路系统化

- **系统芯片SOC (System-On-a-Chip)**
(处理器+存储器+ DSP+FPGA+ASIC+其他)
 - SOC在性能、功耗、体积、重量和总的设计制造费用上都有巨大的优越性。
- **问题：硬件+软件， 数字+模拟**
- **要求：自动化程度高，开发周期短，可复用**
- **硬件/软件协同设计 (HW/SW co-design)**
 - 软硬件划分
 - IP核（IP core）为中心的设计
 - 嵌入式软件
 - 协同验证
- **模拟电路和视频电路与数字电路的混合设计技术**



高层次综合与验证结构图



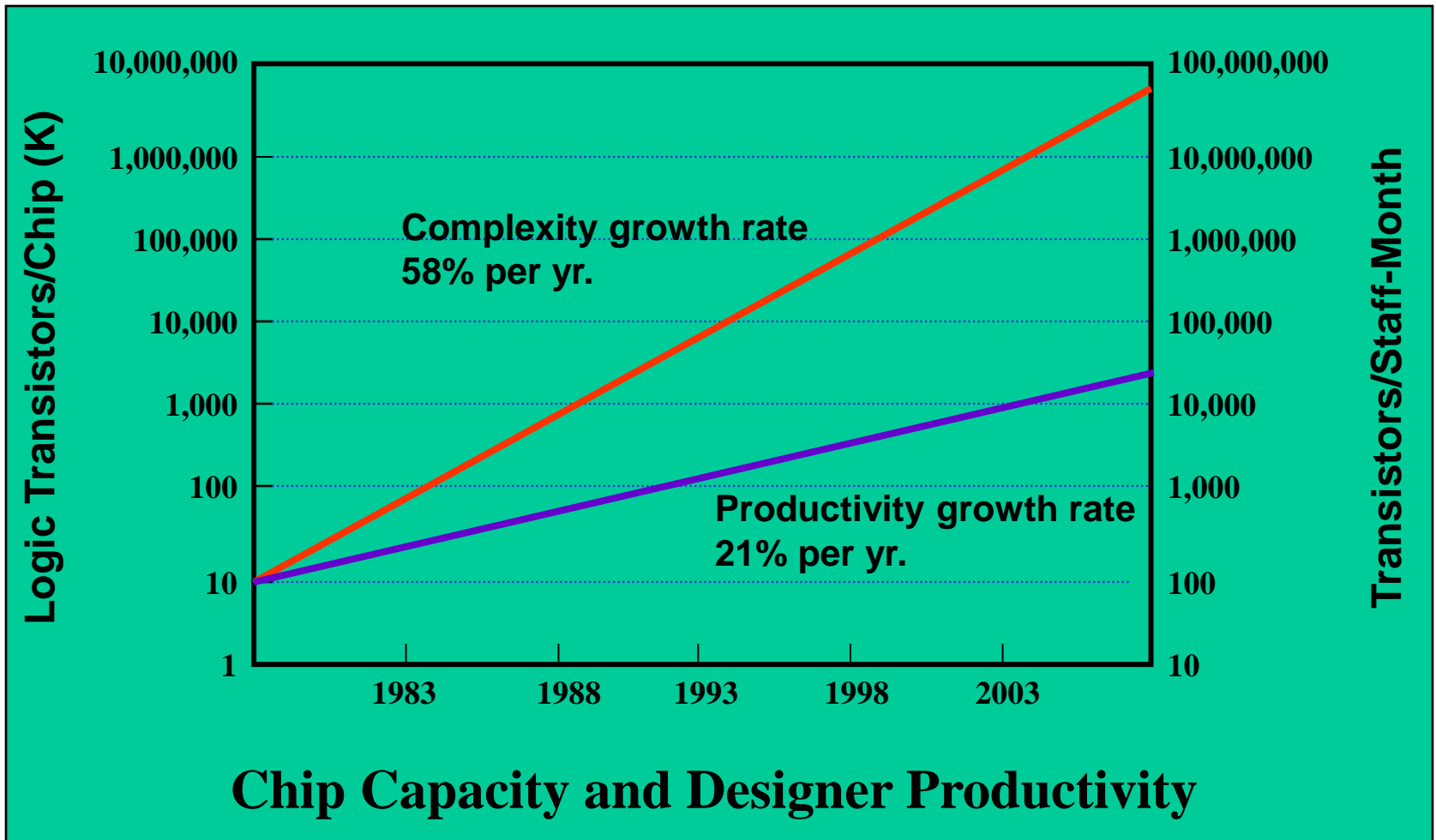


1.5.4 设计复杂性的双倍指数级的增长

- ◆ C1: 芯片集成度指数级增长引起的复杂性
 - 更多的器件
 - 更大的功耗
 - 异种器件、部件或电路的集成
- ◆ C2: 特征尺寸指数级减小引起的复杂性
 - 互连线延迟
 - 耦合噪声
 - 电磁干扰
- ◆ 设计复杂性 $\propto C1 \times C2$



1.5.5 设计能力与芯片集成度增长的差距



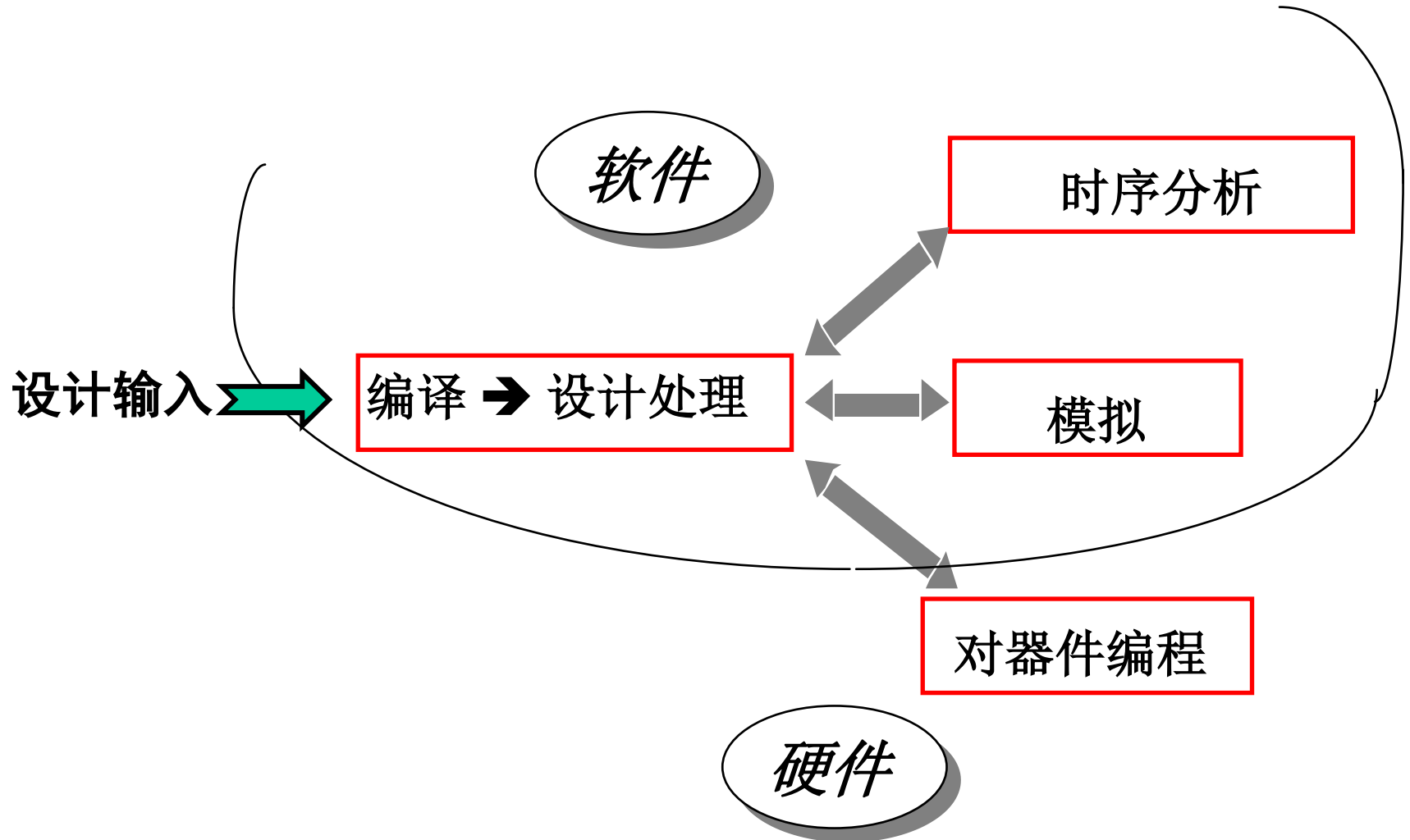


1.6 用 EDA 工具设计集成电路

- ◆ EDA 工具 —— MUXPLUS 2
- ◆ 描述方法 —— VHDL / 图形
- ◆ 集成电路种类 ——
 - 通用型 —— CPU, MEM ...
 - 专用型 —— 厂家制造
 - 可编程逻辑器件 CPLD/FPGA ✓
 - Complex Programmable Logic Device
 - Field Programmable Gate Array

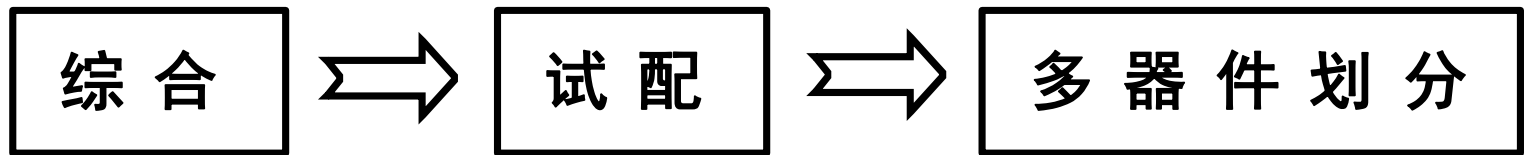


设计流程





设计处理



- ◆ 综合： 逻辑实现和优化；
- ◆ 试配： 综合结果映射到特定器件，生成相应文件；
 - 例如网表文件，供模拟器使用
 - 例如编程文件，供编程使用
 - 例如报告文件，报告芯片利用率、管脚分配...
- ◆ 多器件划分： 如果一片装不下；分装入多片。



模拟：检验电路的功能是否满足要求

- ◆ 编辑激励信号波形；
- ◆ 运行模拟器
（其输入为：网表文件 + 激励信号文件）；
- ◆ 观察、分析各点波形，判断是否满足要求；
 - 如果不满足要求，修改设计；
- ◆ 反复执行以上步骤，直到满足要求为止。



课程理论内容

◆ 主要内容：

- 数字EDA系统的流程、基本技术和工具

◆ 章节（部分内容与教材不同）

- 第2章 硬件描述语言（3课时）
- 第3章 高层次综合（6课时）
- 第4章 逻辑综合（6课时）
- 第5章 划分（3课时）
- 第6章 布图（3课时）
- 第7章 布局（3课时）
- 第8章 布线（3课时）
- 第9章 时序分析（3课时）
- 第10章 模拟（1课时）
- 第11章 形式验证（2课时）