



第二章 ASIC 设计流程和方

2.1 法概述

设计过程分

电路设计 --- 前端设计

版图设计 --- 后端设计

设计流程 (方法) 分

自底向上 (Bottom Up)

自顶向下 (Top Down)

数字集成电路设计

行为方面

结构方面

物理方面



第二章 ASIC 设计流程和方

法 2.1 概述

设计策略

设计描述

自动化设计的综合方法

设计验证方法

深亚微米设计方法和 EAD 工具的发展



第二章 ASIC 设计流程和方

2.2 设计描述

描述方面

行为描述

结构描述

物理描述

设计抽象的层次

系统算法级

寄存器传输级 (RTL 级)

逻辑级和电路级

最低层的晶体管级电路



第二章 ASIC 设计流程和方

法 2.2.1 硬件描述语言 HDL (Hardware Description Language)

VHDL

VHDL 描述能力强，覆盖面广，可用于多种层次的电路描述，

VHDL 的硬件描述与工艺技术无关，不会因工艺变化而使描述无效。

VHDL 支持设计再利用 (Reuse) 方法，支持超大规模集成电路设计的分解和组合。

可读性好，易于理解，国际标准，具备通用性。



第二章 ASIC 设计流程和方 法

VHDL 设计描述由五种基本设计单元组成

设计实体说明 (Entity declaration)

结构体 (Architecture body)

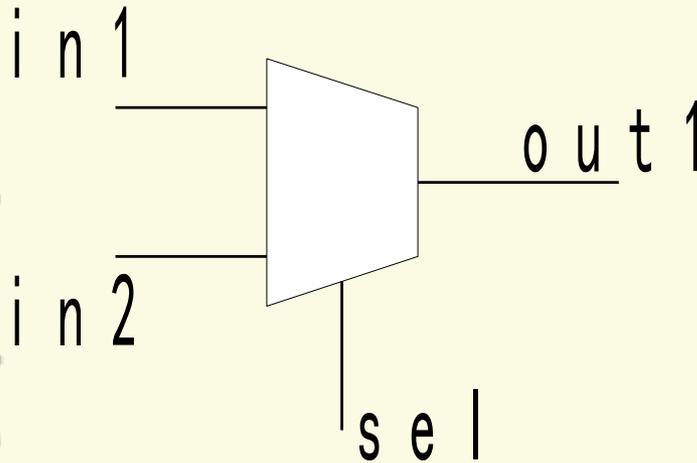
配置说明 (Configuration declaration)

集合元说明 (Package declaration)

集合元 (Package body)



第二章 ASIC 设计流程和方 法



```
ENTITY mux IS
```

```
GENERIC (m:TIME:=2ns);
```

```
PORT (in1,in2,sel:IN BIT;
```

```
out1:OUT BIT);
```

```
END mux;
```

```
-- 设计实体说明
```



第二章 ASIC 设计流程和方 法

AECHITECTURE twown1 OF mux IS

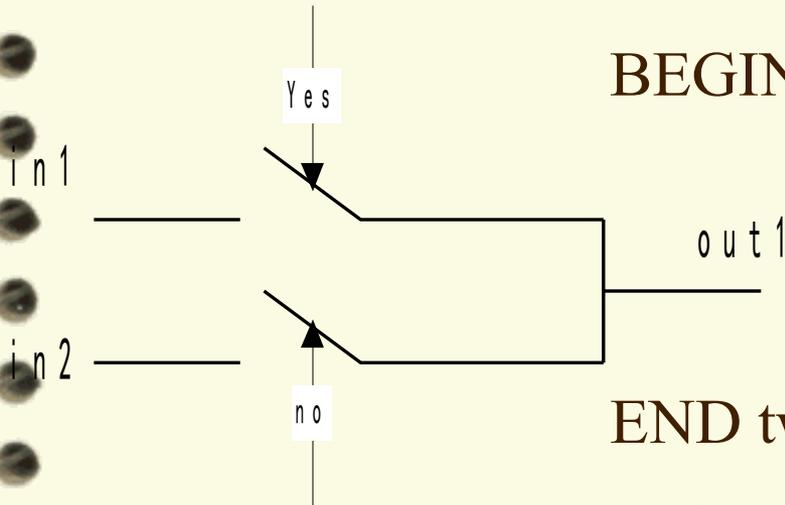
BEGIN

IF sel='1' THEN out1<=1;

ELSE out1<=in2 AFTER m;

END twown1;

-- 行为描述

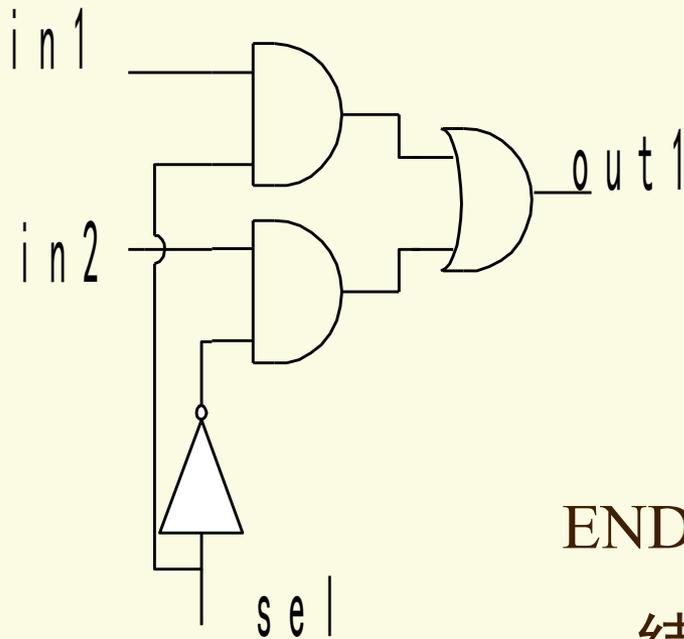




第二章 ASIC 设计流程和方 法

AECHITECTURE twown2 OF mux IS

BEGIN



NOT:Sb=U0(sel);

AND2:S1=U1(sel,in1);

AND2:S2=U2(Sb,in2);

OR:out1=U3(s1,s2);

END twown2;

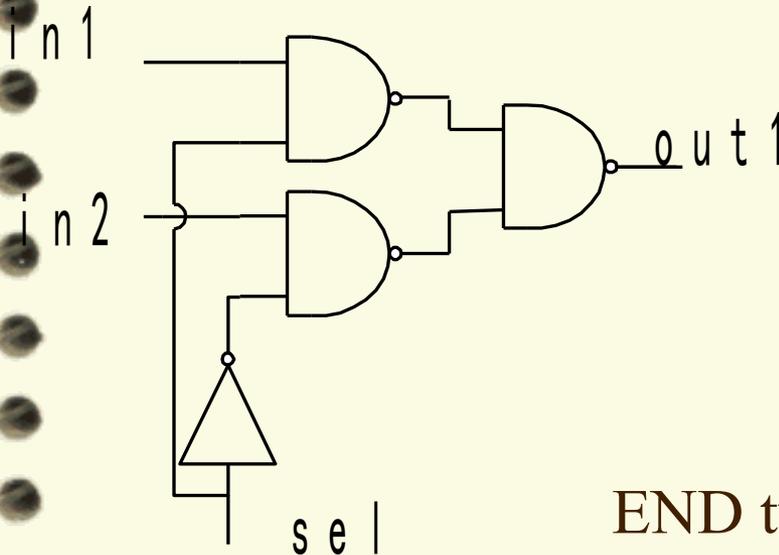
-- 结构描述 1



第二章 ASIC 设计流程和方 法

AECHITECTURE twown3 OF mux IS

BEGIN



NOT:Sb=U0(sel);

NAND2:S1=U1(sel,in1);

NAND2:S2=U2(Sb,in2);

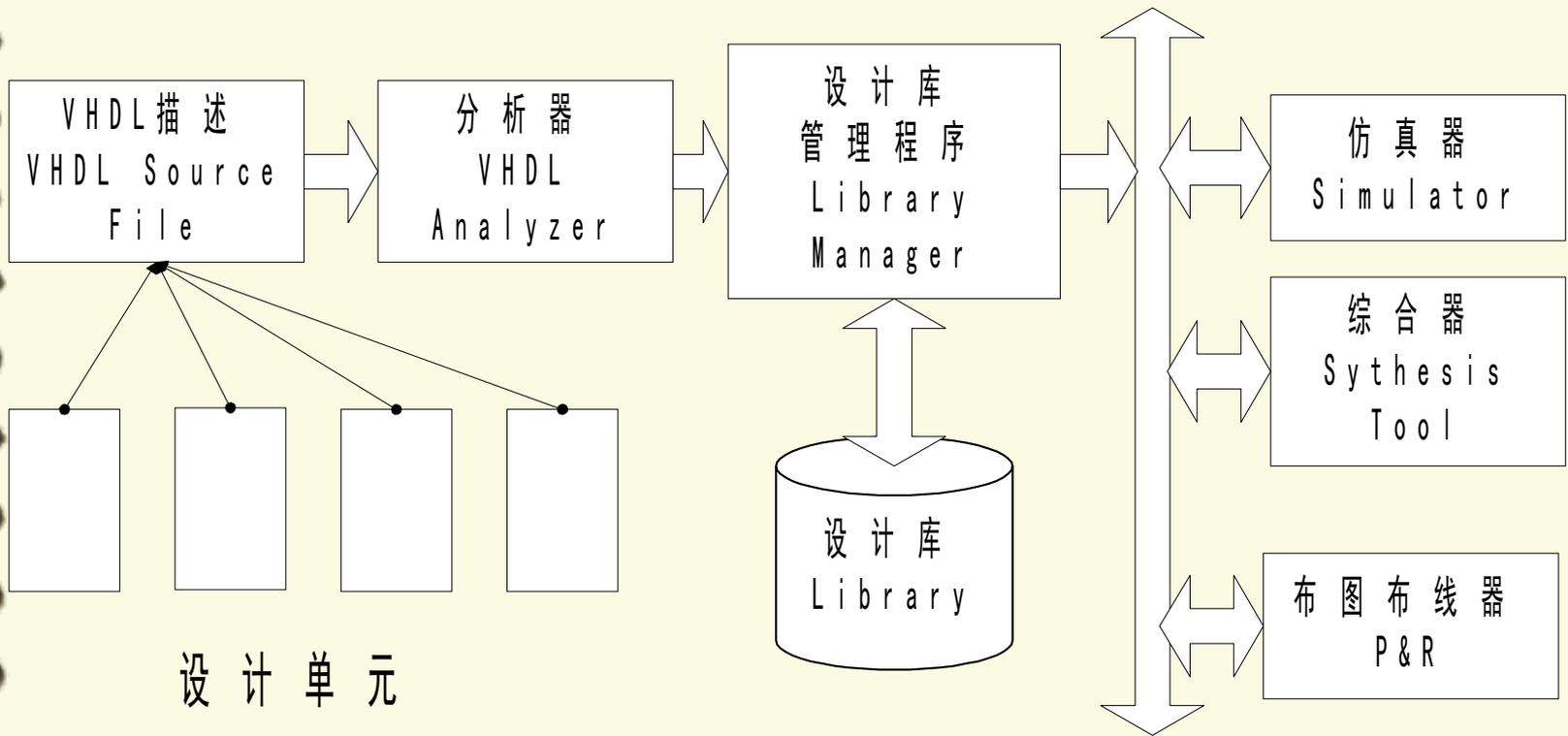
NAND:out1=U3(s1,s2);

END twown3;

-- 结构描述 2



VHDL 设计环境





第二章 ASIC 设计流程和方 法

Verilog HDL

能用于行为描述和结构描述，电路描述同时可以包含不同层次，且能和混合模式的模型一起进行模拟

Verilog 使用四值逻辑，即 0，1，X 和 Z，其中“X”为不定态，Z 为悬空态

使用的基本数据类型是“与”和“寄存器”。

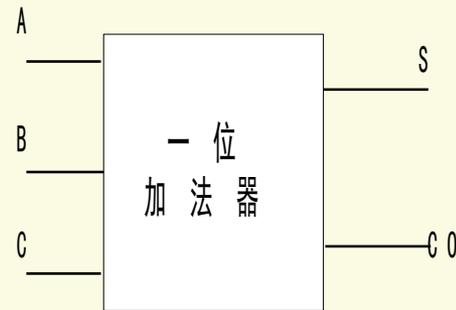


第二章 ASIC 设计流程和方 法.2 行为描述 (算法描述)

举例 一位全加器

布尔表达式 : $S = A\bar{B}\bar{C} + \bar{A}B\bar{C} + \bar{A}\bar{C}B + ABC$
 $CO = AB + AC + BC$

A	B	C	S	CO
0	0	0	0	0
0	0	1	1	0
0	1	0	1	1
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1





第二章 ASIC 设计流程和方 法

Verilog-HDL 描述进位算法描述

```
module carry(co,a,b,c);  
    output co;  
    input a,b,c;  
    wire #10 co=(a&b)|(a&c)|(b&c)  
end module
```



第二章 ASIC 设计流程和方 法

2.2.3 结构描述

RTL (register Transfer Level) 级

门级 (Gate Level)

开关级 (Switch Level)

电路级 (Circuit Level)

4 位加法器的结构描述



第二章 ASIC 设计流程和方 法

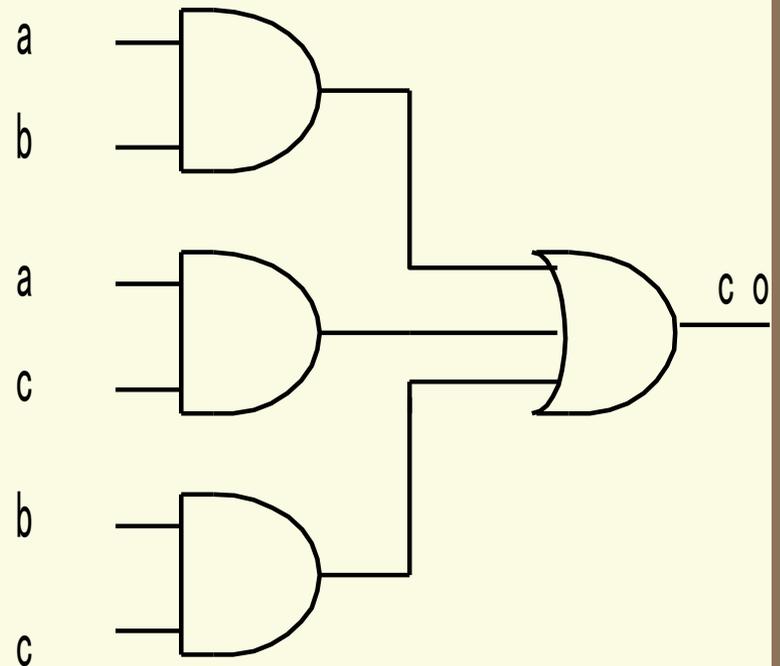
4 位加法器的结构描述

```
module add4(s,c4,ci,a,b);  
    input[3:0] a,b;  
    input ci;  
    output[3:0] s;  
    output c4;  
    wire[2:0] co;  
        add a0 (co[0],s[0],a[0],b[0],ci);  
        add a1 (co[1],s[1],a[1],b[1],c[0]);  
        add a1 (co[2],s[2],a[2],b[2],c[2]);  
        add a1 (co4,s[3],a[3],b[3],co[2]);  
end module
```



```
module add(co,s,a,b,c);  
    input a,b,c;  
    output s,co;  
    sum s1(s,a,b,c);  
    carry c1(co,a,b,c);  
end module
```

```
module carry(co,a,b,c);  
    input a,b,c;  
    output co;  
    wire x,y,z;  
    and g1(x,a,b);  
    and g2(y,a,c);  
    and g3(z,b,c)  
    or3 g4(co,x,y,z)  
end module
```



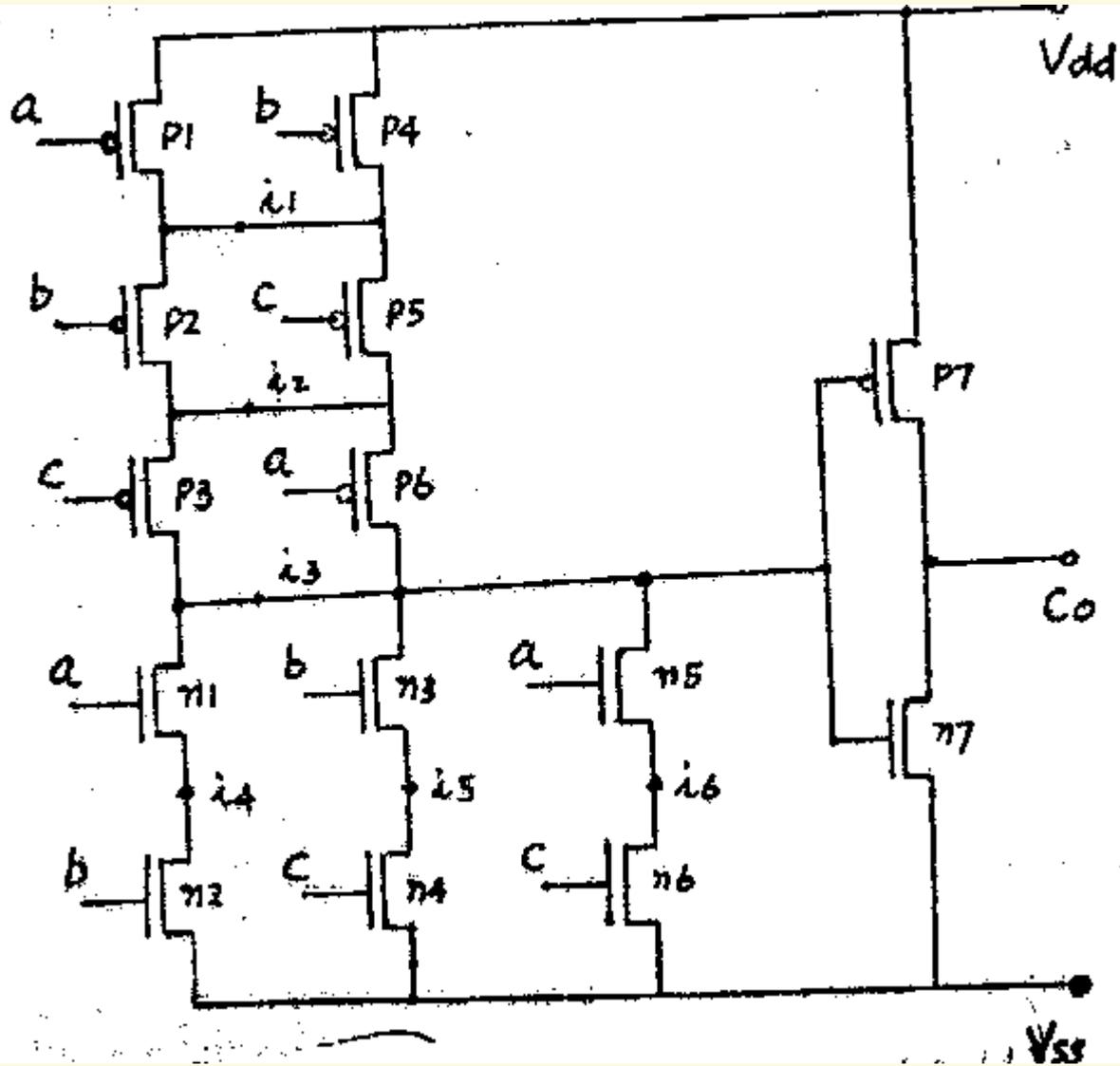


第二章 ASIC 设计流程和方 法

开关级描述 (1)

```
module carry (co, a, b, c);  
  input a, b, c;  
  output co;  
  wire i1, i2, i3, i4, i5, i6;  
  nmos n1 (i3, i4, a);  
  nmos n2 (i4, vss, b);  
  nmos n3 (i3, i5, b);  
  nmos n4 (i5, vss, c);  
  nmos n5 (i3, i6, a);  
  nmos n6 (i6, vss, c);  
  nmos n7 (co, vss, i3);  
  pmos pi (il, vdd, a);
```

```
  pmos p2 (i2, il, b);  
  pmos p3 (i3, i2, c);  
  pmosp4 (il, vdd, b);  
  pmos p5 (i2, il, c);  
  pmos p6 (i3, i2, a);  
  pmos p7 (co, vdd, i3);  
end module
```



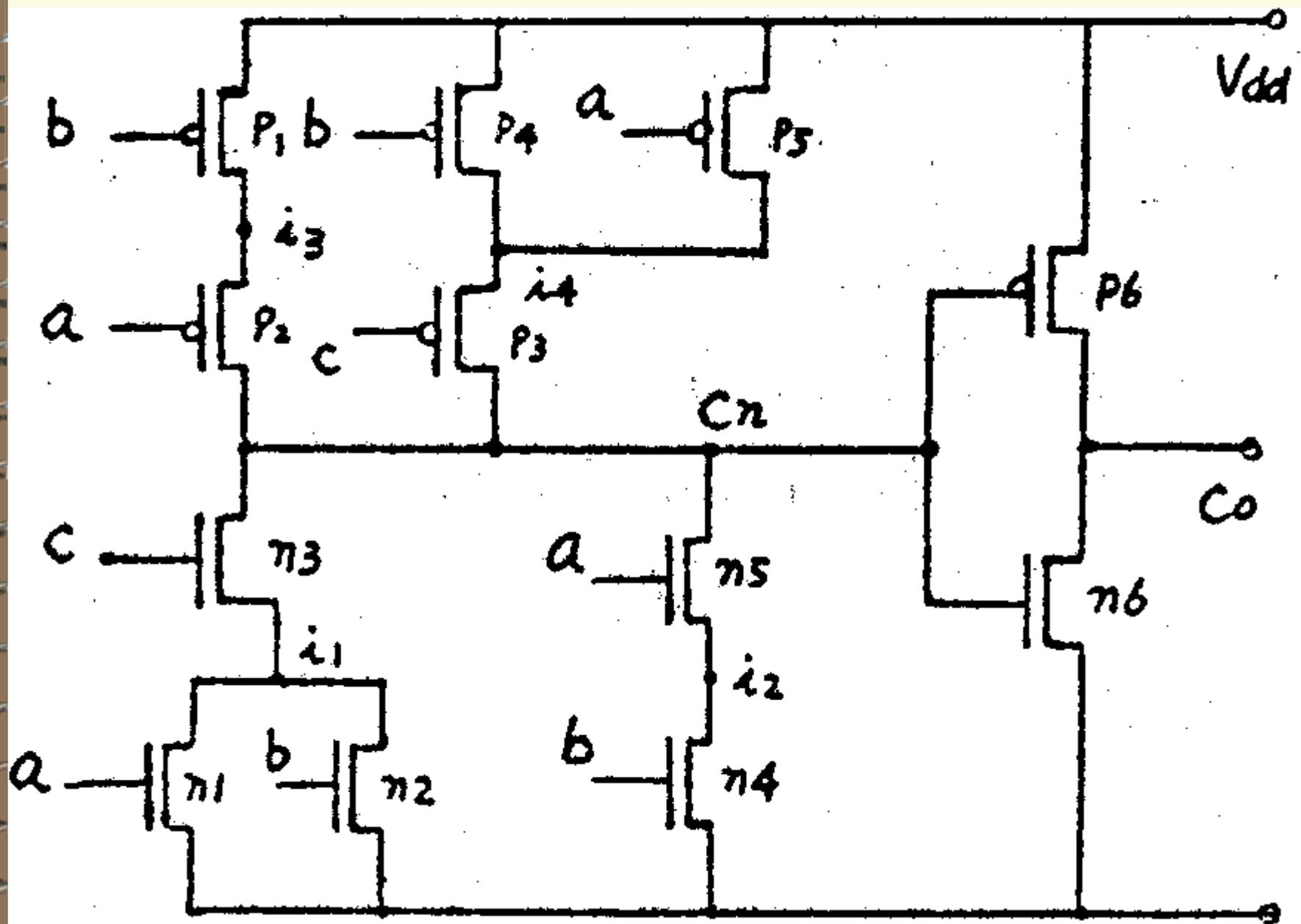


第二章 ASIC 设计流程和方 法

开关级描述 (2)

```
module carry (co, a, b, c);  
input a, b, c;  
output co;  
wire i1, i2, i3, i4, en;  
nmos n1 (i1, vss, a);  
nmosn2 (i1, vss, b);  
nmos n3 (en, i1, c);  
nmos n4 (i2, vss, b);  
nmos ns (en, i2, a);  
pmosp1(i3,vdd,b); .  
pmos p2 (en, i3, a);  
pmos p3(cn, i4, c);
```

```
pmos p4 (i4, vdd, b);  
pmos p5 (i4, vdd, a);  
pmos p6 (co, vdd, en);  
pmos n6 (co, vss, en);  
end module
```





第二章 ASIC 设计流程和方

法

2.2.4 物理描述

```
module add4;  
input a [3:0], b[3:0];  
input ci;  
output s [3:0],  
outpu c4;  
boundary [0, 0, 100, 400];  
port  
port a [0] aluminum width=1  
origin =[0, 25];  
port b [0] aluminum width=1  
origin =[0, 75];  
port ci polysilicon width=1
```

```
origin =[50, 0];  
port a [0] aluminum width=1  
add so origin=[0,0]  
add a1 origin=[0,100]  
end module
```



第二章 ASIC 设计流程和方 法

2.3 设计流程

2.3.1 bottom-Up

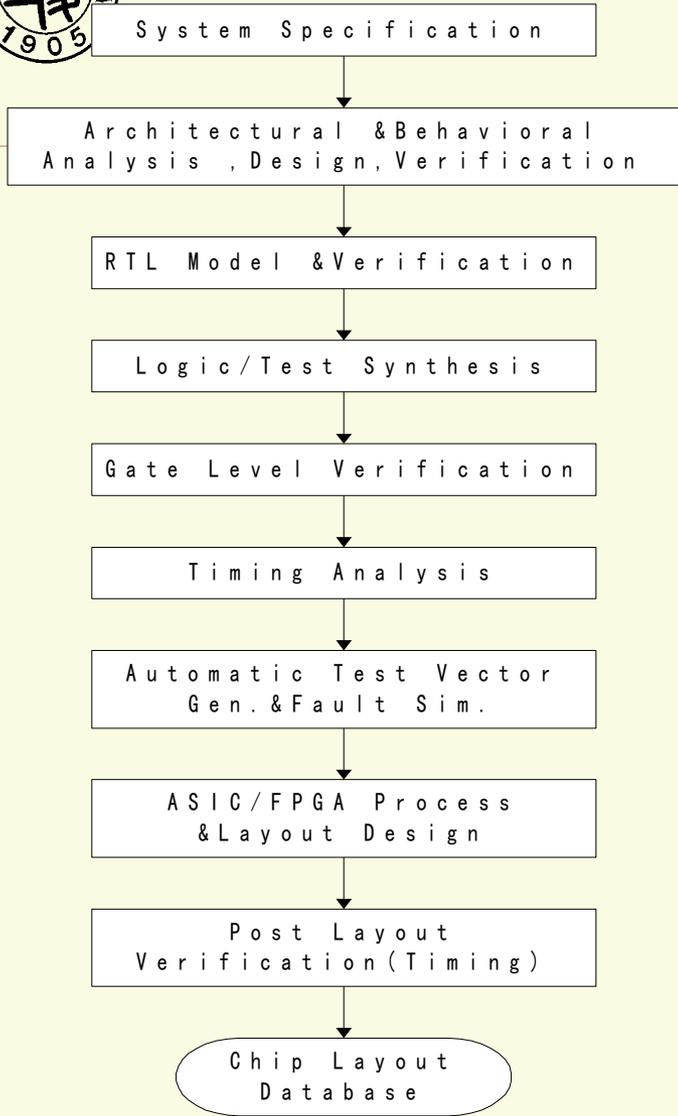
自底向上 (Bottom-Up) 设计是集成电路和 PCB 板的传统设计方法，该方法盛行于七、八十年设计从逻辑级开始，采用逻辑单元和少数行为级模块构成层次式模型进行层次设计，从门级开始逐级向上组成 RTL 级模块，再由若干 RTL 模块构成电路系统

对于集成度在一万门以内的 ASIC 设计是行之有效的，无法完成十万门以上的设计

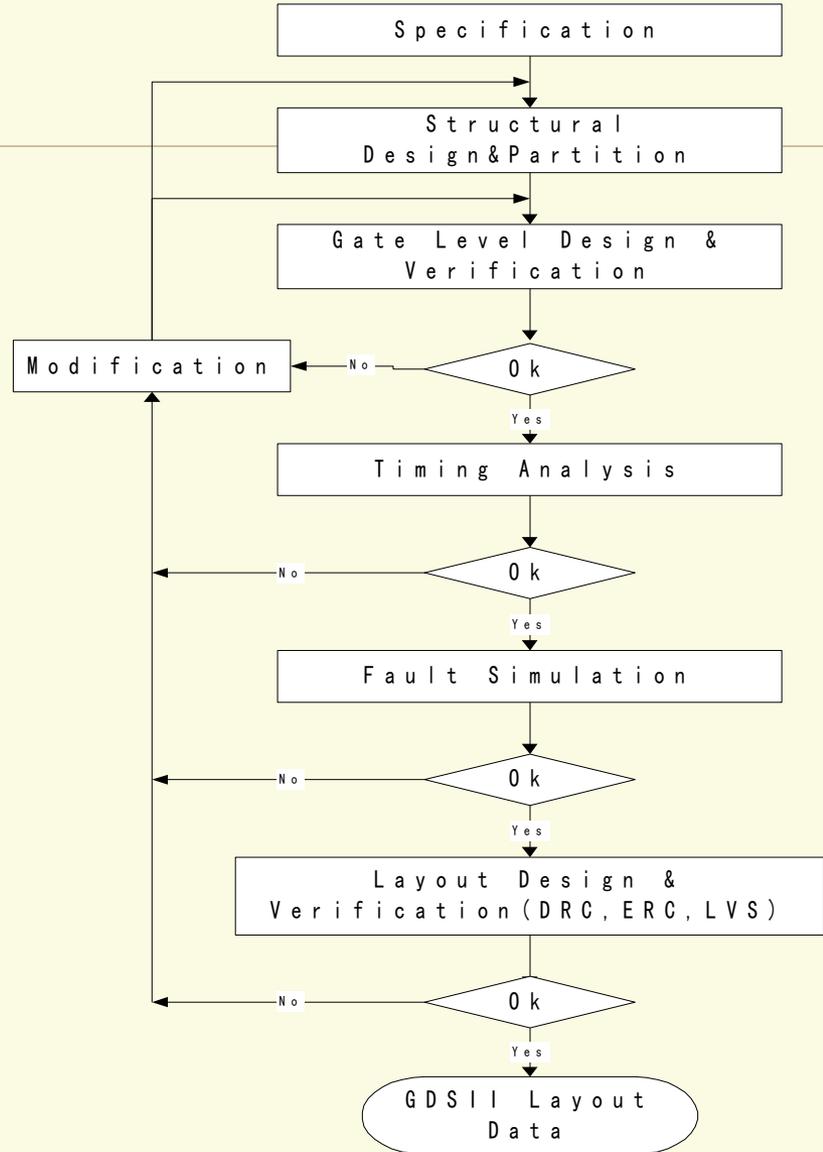
设计效率低、周期长，一次设计成功率低



复旦大学专用集成电路与系统实验室



Top-Down Design Flow



Bottom-Up Design Flow



第二章 ASIC 设计流程和方 法

2.3 设计流程

2.3.2 Top-Down 设计

Top-Down 流程在 EDA 工具支持下逐步成为 IC 主要的设计方法

从确定电路系统的性能指标开始，自系统级、寄存器传输级、逻辑级直到物理级逐级细化并逐级验证其功能和性能



第二章 ASIC 设计流程和方 法

关键技术

首先是需要开发系统级模型及建立模型库，这些行为模型与实现工艺无关，仅用于系统级和 RTL 级模拟。

系统级功能验证技术。验证系统功能时不必考虑电路的实现结构和实现方法，这是对付设计复杂性日益增加的重要技术，目前系统级 DSP 模拟商品化软件有 Comdisco，Cossap 等，它们的通讯库、滤波器库等都是系统级模型库成功的例子。

逻辑综合 -- 是行为设计自动转换到逻辑结构设计的重要步骤



第二章 ASIC 设计流程和方 法

Top-Down 设计与 Bottom-Up 设计相比，具有以下优点：

设计从行为到结构再到物理级，每一步部进都进行验证，提高了一次设计的成功率。

提高了设计效率，缩短了 ASIC 的开发周期，降低了产品的开发成本

设计成功的电路或其中的模块可以放入以后的设计中提高了设计的再使用率 (Reuse)。



第二章 ASIC 设计流程和方 法

2.4 设计策略

2.4.1 概述

设计参数

电路性能，包括功能、速度，功耗和应用特性

芯片尺寸

电路的可测性及测试码生成的难易性；

设计周期

成功率 (Time to Market)

经济性 (Profit)

设计效率 (Efficiency)



第二章 ASIC 设计流程和方 法

2.4 设计策略

2.4.1 概述

设计参数

电路性能，包括功能、速度，功耗和应用特性

芯片尺寸

电路的可测性及测试码生成的难易性；

设计周期

成功率 (Time to Market)

经济性 (Profit)

设计效率 (Efficiency)



第二章 ASIC 设计流程和方

2.4.2 结构设计

层次设计

从高层到低层

从抽象到具体

利于多人同时设计

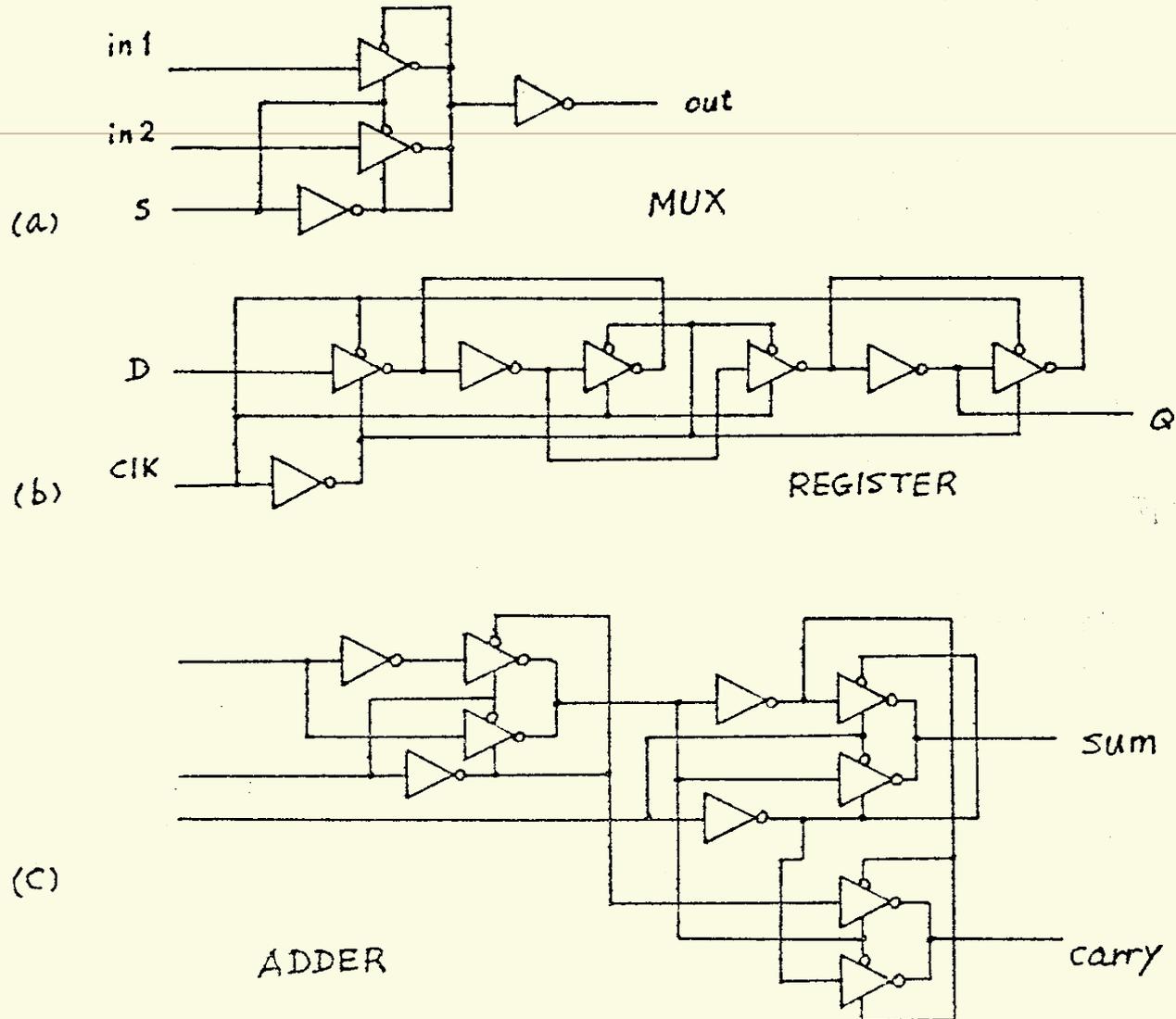
使设计思想清晰，设计工作简化

规则设计

使一个电路系统变成大量不同的子模块

尽可能地将电路划分成一组相同或相似的模块，尽可能采用规划性结构的设计，达到简化设计的目的。

适用于设计的各个阶段和层次



规则性在电路级的体现；用倒相器和三态缓冲器构成的单元电路



第二章 ASIC 设计流程和方 法

2.4.3 设计协调

模块信号的标准化

信号输入的驱动

输入信号的寄存

输出信号的寄存

模块间的连接关系

串接结构

迭代结构

条件选择



第二章 ASIC 设计流程和方 法

2.4.4 模块定时

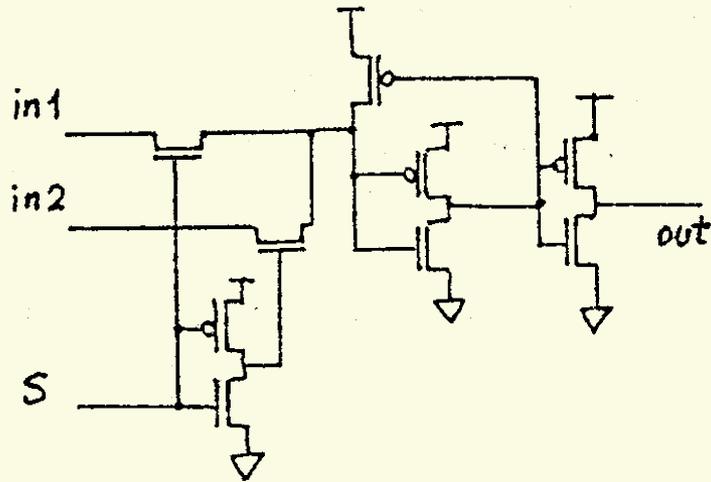
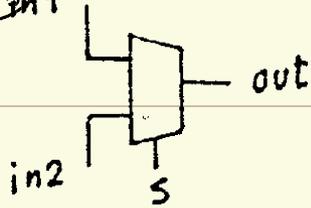
采用公共时钟 (同步时序)

结构清晰

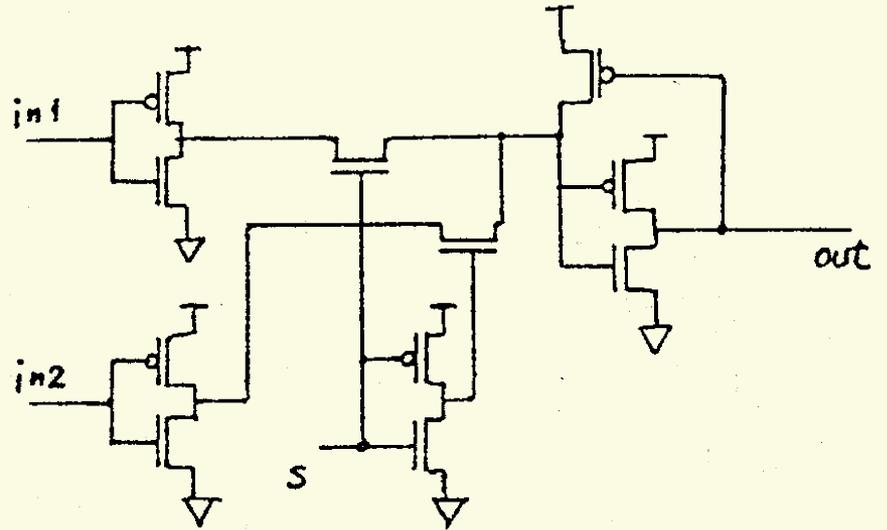
较易验证

可测性好

关键路径



(a)



(b)

图 2.13 (a)二选一多路开关图 1. 2.13 (b)二选一多路开关图 2



第二章 ASIC 设计流程和方 法

2.5 综合方法

2.5.1 概述

对芯片高性能，高密度，高可靠性，设计周期的要求

包含三个层次

行为综合

逻辑综合

版图综合



行为特性

行为综合

结构特性

系统级

RTL级

逻辑级

版图级

逻辑综合

物理综合

物理特性

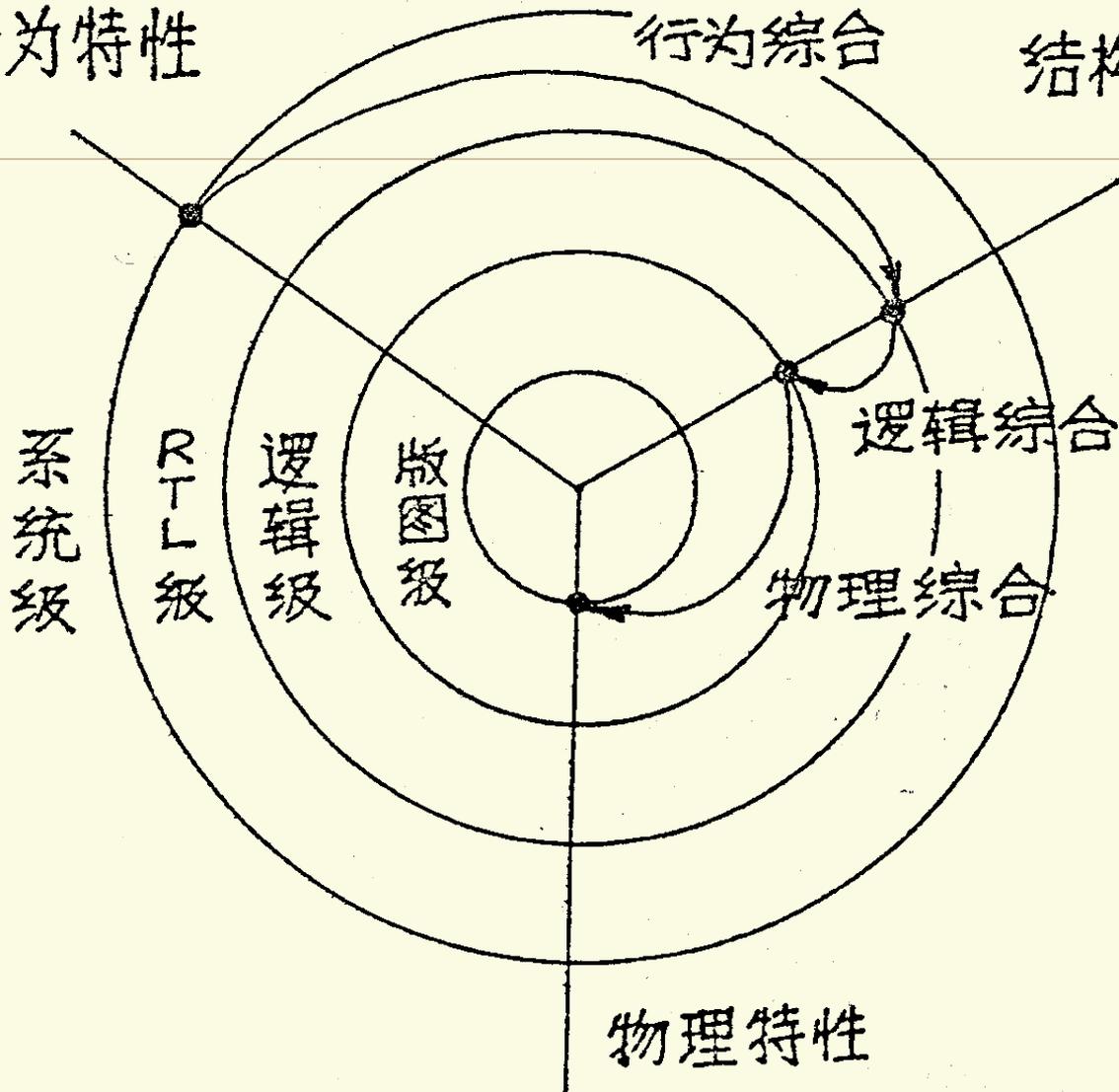
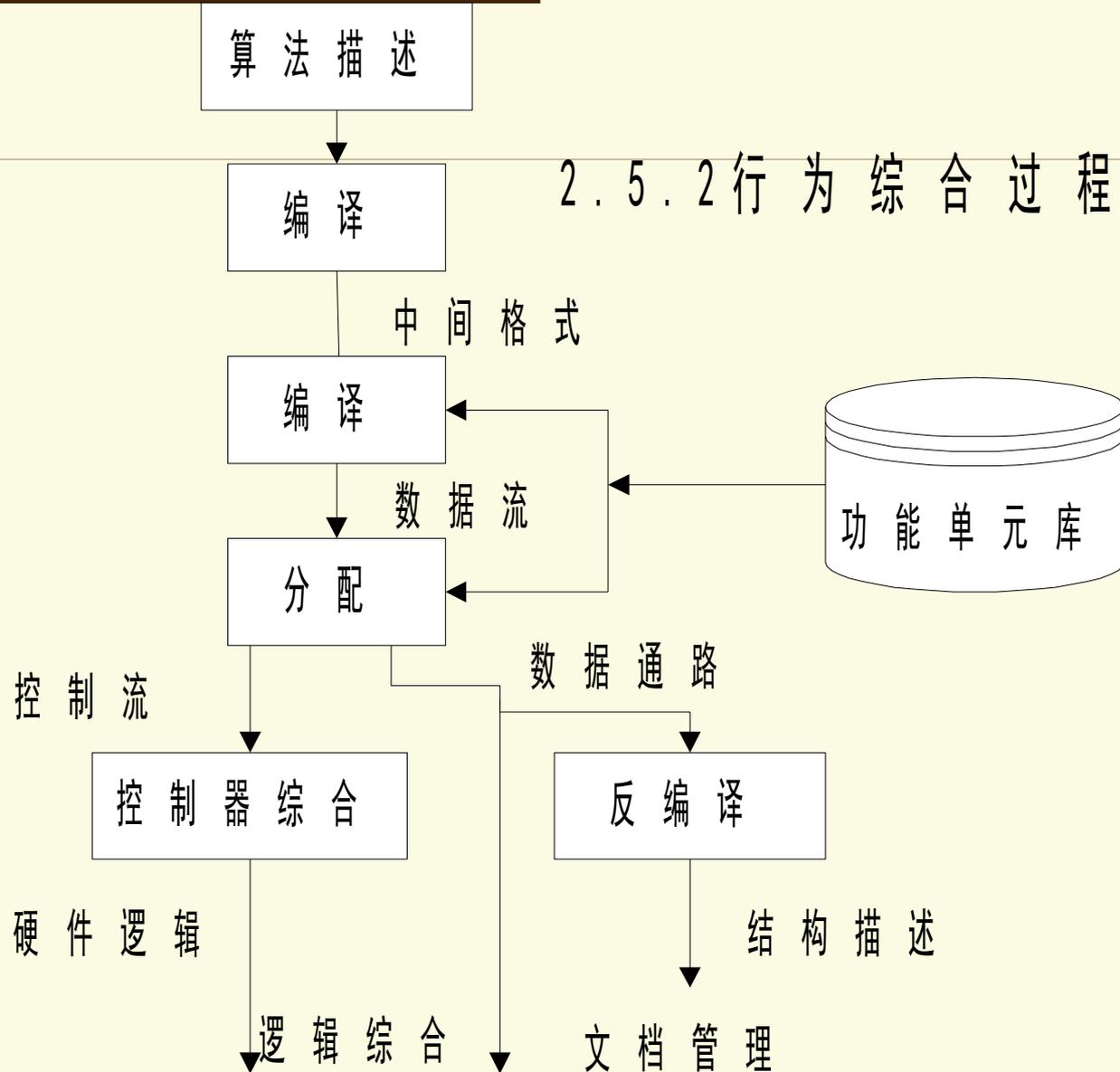
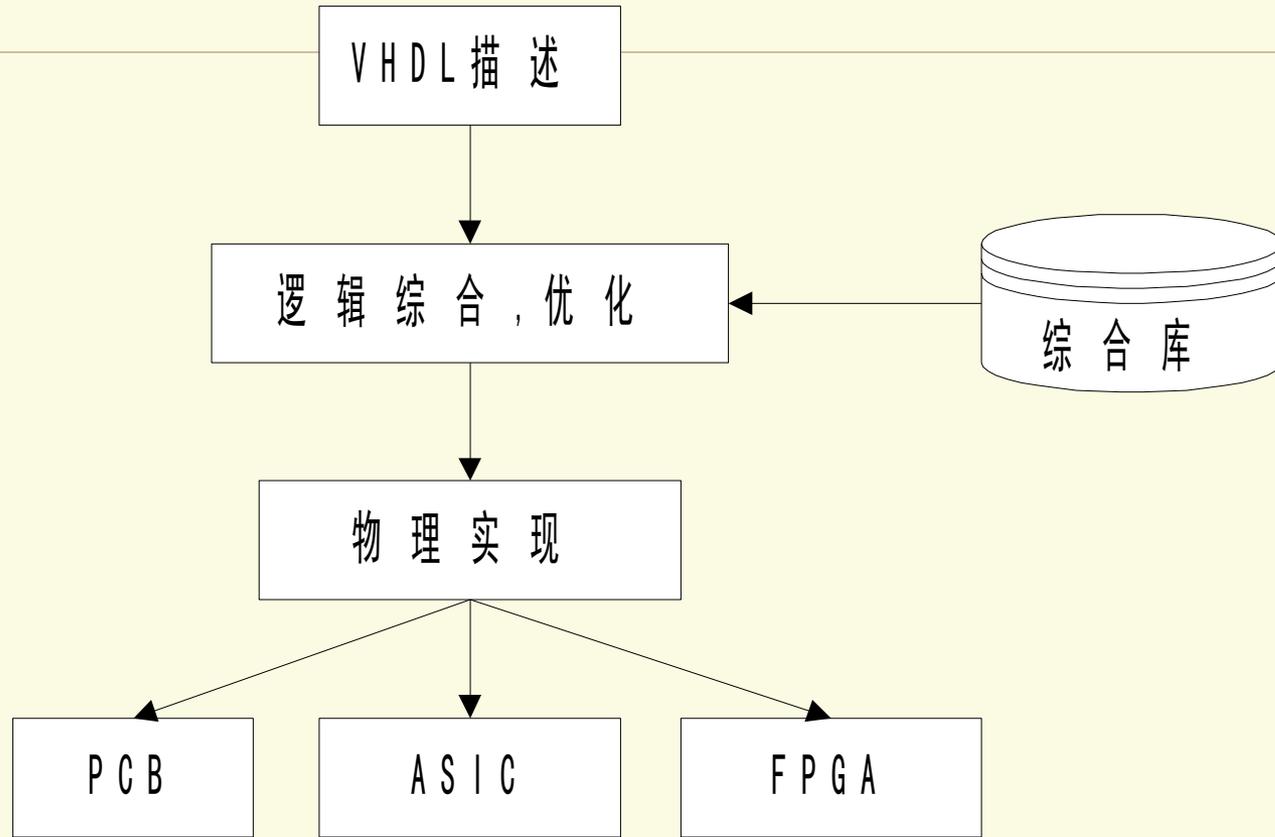


图 2.14 数字系统三个层次的综合方法



2.5.2 行为综合过程





逻辑综合和优化过程

Logic Synthesis and Logic Optimization



第二章 ASIC 设计流程和方

2.5 法逻辑综合和逻辑优化

综合过程是将 VHDL 描述转换成非优化约布尔等式的描述，也就是门级描述，读转换过程是综合软件自动完成的，其过程不受用户控制。

RTL 级描述 --- 一般使用 HDL 硬件描述语言，从描述语句和结构特征来分析可归纳为以下几种情况：

- 使用 if then-else 和 case 语句来控制流程；

- 反复迭代

- 层次

- 字宽、位向量和位场

- 串行和并行操作

- 算术、逻辑运算和比较操作

- 寄存器的规定和分配。



六位约翰逊计数器 VHDL 描述

ENTITY counter IS

PORT (

clk: IN STD_LOGIC;

rs: IN STD_LOGIC;

count_out: OUT STD_LOGIC_VECTOR(0 TO 2)

END counter;

ARCHITECTURE behav OF counter IS

signal next_count: STD_LOGIC_VECTOR(2 DOWNTO 0)

BEGIN

IF rs= '0' THEN

count_out<= "000";

ELSEIF rs= '1' AND prisig (clk) THEN

CASE count_out (0 TO 2) IS

WHEN "000" => next_count <= "001";

WHEN "001" => next_count <= "011";

WHEN "011" => next_count <= "111";

WHEN "111" => next_count <= "110";

WHEN "110" => next_count <= "100";

WHEN "100" => next_count <= "000";

END CASE;

count_out <= next_count AFTER 10ns;

END IF;

END PROCESS;

END behav;



第二章 ASIC 设计流程和方

法 约束条件 (Constrain)--- 芯片面积 (Area), 延时 (Delay), 功耗 (Power Consumption) 和可测性 (Testbility) 等

属性描述 --- 规定电路的负载数或驱动能力 (Load), 输入信号定时 (Timing), 实际上也是 Constrain.

综合库 (Synthesis Library)-- 包含可综合单元的全部信息 -- 逻辑功能 (Function), 定时关系 (Timing), 输入的负载数 (Capacitance), 输出扇出数 (Load), 单元的面积 (Area)



两输入端与非门的综合库描述

```
LIBRARY ( ABC ) {  
  CELL ( and2 ) {  
    area: 5  
    pin (a1, a2){  
      direction: input;  
      capacitance: 1;  
    }  
    pin (o1){  
      direction: " a1*a2";  
      timing ( ){  
        intrinsic_rise: 0.37;  
        intrinsic_fall: 0.56;  
        rise_resistance: 0.1234;  
        fall_resistance: 0.4567;  
        ralated_pin: " a1, a2";  
      }  
    }  
  }  
}
```



第二章 ASIC 设计流程和方

法 逻辑优化 --- 在给定综合库的情况下，对于逻辑描述所形成的门电路网络进行优化，优化的目标是根据电路速度和面积等约束条件进行协调，简化和改善电路的逻辑设计优化过程分两个阶段进行，它们是：

与工艺无关的逻辑优化阶段：运用代数和布尔代数技术对电路进行优化（运用两级极小化过程）

结合综合库，与目标工艺对照阶段：根据制造工艺的要求，将已简化的逻辑描述转换成综合库要求的表达形式，也就是用相应的单元符号，包括标准单元或 FPGA 元件符号以及其它物理实现的逻辑符号替代已简化的描述



第二章 ASIC 设计流程和方

法级映射 (Mapping) 及门级网表 --- 利用综合库单元的逻辑功能及定时信息, 进行门级映射并产生门级网表。该网表是逻辑综合和优化的结果, 是电路面积和速度目标的体现。门级映射过程是根据优化的布尔描述, 综合本库以及用户的约束条件, 得到一个以综合库单元为基础的优化网表, 该综合库单元是与物理实现的工艺参数紧密结合的



第二章 ASIC 设计流程和方

法市场上著名的逻辑综合软件有 Synopsys 软件和 Ambit 软件，作为逻辑综合软件应具
具有以下功能：

支持 RTL 级 VHDL 或 Verilog HDL, 这是最低限度的要求
能支持多种目标工艺，例如多种 CMOS 工艺、FPGA 或其它
实现工艺

具有自动扫描插入能力

满足 ATPG(测试码自动生成) 集成的要求

支持对超大规模 ASIC 进行自动布局布线的集成要求

除了逻辑综合以外还有测试综合软件，这是运用编译的方法，
使逻辑门设计产生自动测试链，以提高电路测试覆盖率的方法，
上述第 3 条功能就是为了测试综合而设置的。



第二章 ASIC 设计流程和方

法

2.5.4 版图综合 -- 从电路的逻辑结构到集成电路版图的转换是物理综合的过程，也称为版图综合，实际上就是自动布局布线的过程。按照设计流程，逻辑设计验证完毕接着就可以进行自动的版图设计。

布局算法 -- 布局是放置版图模块的工作，考虑到以后的布线通常是把连接紧密的模块依次放置，目的是使整个版图的面积和电路的工作周期最小，这就是所谓基于 Timing 的布局。



第二章 ASIC 设计流程和方

布法算法 -- 布局是放置版图模块的工作，考虑到以后的布线通常是把连接紧密的模块依次放置，目的是使整个版图的面积和电路的工作周期最小，这就是所谓基于 Timing 的布局。

MinCut 算法和“热退火”算法。

布线算法 -- 布线是根据连接网表对布局后的模块进行连线，布线器的类型分成通道布线、开关箱布线和迷宫 (Maze) 布线等几种。



第二章 ASIC 设计流程和方

2.5 **法** 硅编译器 --- 可以将高级的行为描述转换成低一级的结构性的功能块，也可变为供模拟和生成版图用的数据格式，一旦确定了电路体系结构，硅编译器就可以提供制造这个电路所需的版图。

适用于版图结构规则的

RAM、ROM、PLA 和通用的数据流等



第二章 ASIC 设计流程和方

2.6 设计验证 (Design Verification)

2.6.1 概述

功能验证 (Function)

时序验证 (Timing)

参数验证 (Parameter)

2.6.2 模拟 (仿真) Simulation

模拟软件是用来验证和预测电路的特性。模拟软件有许多种，主要的特性是适用范围、模拟的精度和速度。



第二章 ASIC 设计流程和方

法 电路级模拟 (Circuit Simulation)--- 电路级分析也就是晶体管级分析，可以看作是最详尽和最精确的模仿真技术电路分析软件典型的代表是 SPICE 和 ASTAP, 目前作为软件商品的实用版本是 AVANTI 公司的 HSPICE 程序。

分析电路的直流 (DC)、交流 (AC) 和瞬态特性 (Transient)，例如：静态工作点分析、直流传输特性分析、交流小信号分析、瞬态分析以及失真、噪声和直流灵敏度分析，其特点为：



第二章 ASIC 设计流程和方

法 根据电路拓扑结构，以节点电压法为基础，构成描述电路特性的代数—微分方程组

采用数值积分方式，将描述电路特性的代数—微分方程转化为非线性代数方程

以牛顿—莱夫森方法为基础，将非线性代数方程转化为线性代数方程

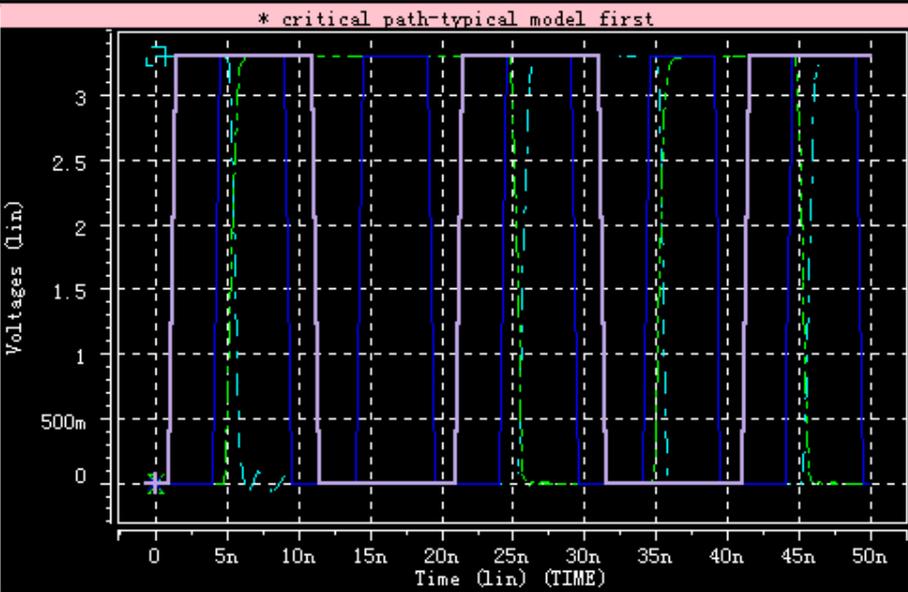
采用稀疏矩阵技术和高斯消元法、LV 分解法等方法求解线性代数方程。

仿真时间通常与 N^m 成比例，这里 N 是电路中器件的个数， m 数值在 1-2 之间

Model 的精度直接影响仿真结果（目前较多使用的 MOSFET 型 Model---Level28,Level48,BSIM3 等）



- Wave List
- DO: AO: v (q)
 - DO: AO: v (qn)
 - DO: AO: v (clk)
 - DO: AO: v (data)



Results Browser

Design: DO: G:\AVANTI\99.4\DEMO\AWAVES\TUTORIAL\TRAN\CPATH

AO: Transient: * critical path-typical model first

Hierarchy: Top

Types: Time, Voltages, Current 1, Current 2

Curves: v (clk), v (data), v (q1), v (q2), v (q3), v (q4), v (q5), v (q6), v (q), v (qn)

Current X-Axis: TIME

Filter: *

Buttons: Apply, Default, Close, Help

```

Cpath.sp - 记事本
文件(F) 编辑(E) 搜索(S) 帮助(H)

* Critical Path-Typical Model First

.option parhier = local

*****
* CadenceLink Version 1.2
* Hspice Netlist
* May 22 13:12:16 1995
*****

.Global vdd gnd sub

.subckt dff  clk d q qn
mp12 ms2 ms1 vdd vdd pch l=1en w=5
mp9 s11 phn ms2 vdd pch l=1en w=5
mp8 s12 s11 vdd vdd pch l=1en w=3.3
mp7 s11 s12 vdd vdd pch l=1en w=0.5
mp6 q s12 vdd vdd pch l=1en w=6
mp5 qn s11 vdd vdd pch l=1en w=6
mp4 ms1 ms2 vdd vdd pch l=1en w=0.5
mp3 ms1 ph n1 vdd pch l=1en w=3.3
mp2 n1 d vdd vdd pch l=1en w=3.3
mp1 ph phn vdd vdd pch l=1en w=2.3
mp0 phn clk vdd vdd pch l=1en w=2.3
m10 ms2 ms1 gnd sub nch l=1en w=2.5
m9 q s12 gnd sub nch l=1en w=3
m8 ms2 ph s11 sub nch l=1en w=2.5
m7 qn s11 gnd sub nch l=1en w=3

```

HSP UI hspui

File Configuration Help

Design: g:\avanti\99.4\demo\awaves\tutorial\tran\cpath.sp Open

Title: * Critical Path-Typical Model First Edit Netlist

Listing: g:\avanti\99.4\demo\awaves\tutorial\tran\cpath.lis Edit Listing

Version: g:\avanti\99.4\pc\hspice.exe

MT Option: 1 Simulate Avanwaves Multi-Jobs Close



第二章 ASIC 设计流程和方

法八十年代后期以松弛法为基础的第三代电路分析程序，是采用迭代法求解电路方程，其最大的优点是能够有效地利用电路中的休眠特性，大大提高了分析的速度和规模。商业化软件代表 Star-sim

逻辑模拟 -- 逻辑模拟通常是指门级 (Gate Level) 模拟，目的是要验证 ASIC 门级逻辑设计的正确性

功能模拟 -- 验证 ASIC 逻辑功能的正确性 (单位延时或 0 延时)

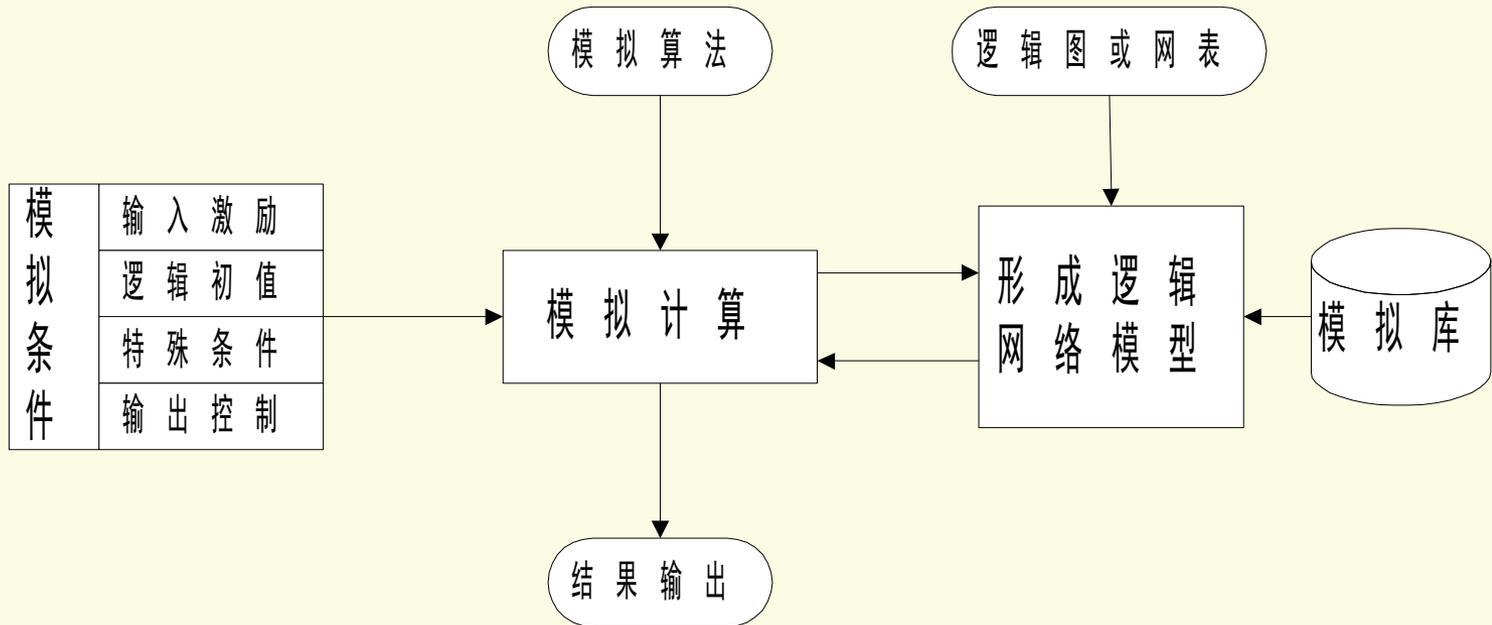
时序模拟 -- 同时考虑器件延迟和连线延迟的功能模拟

故障模拟 -- 故障模拟是为了检验测试向量的有效性，为以后的芯片测试作准备



第二章 ASIC 设计流程和方 法

逻辑模拟系统





第二章 ASIC 设计流程和方

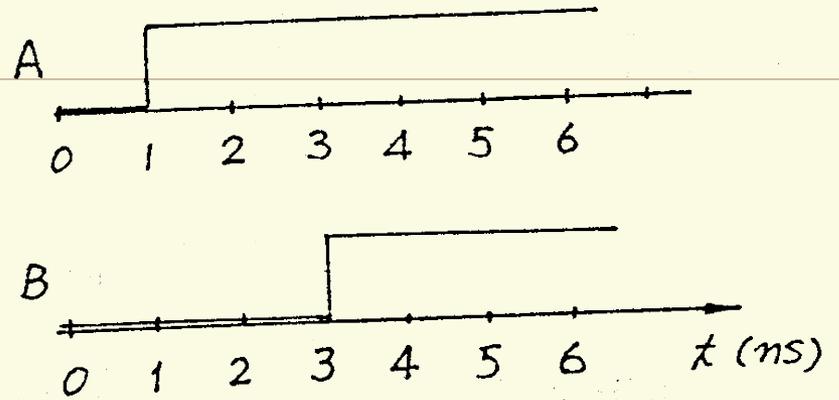
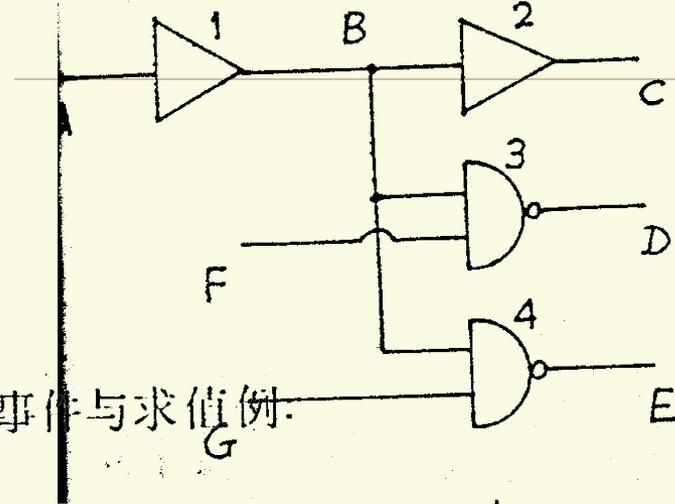
法

逻辑模拟算法 -- 逻辑模拟算法是根据输入激励向量和逻辑模型求解电路响应的方法，算法还应包括在计算前对电路各元件进行排序以及对各元件输出值计算的方法。模拟算算法应遵循的原则 --- 尽量减少计算的次数，同时又保证一定的计算精度，和出发点。

时间驱动 - 首先设定了时间步长，每隔一个时间步长就对电路中的每个元件的输出值计算一遍。

事件驱动 -- 算法能够跟踪电路的信号活动，仅对输入信号有变化的元件求值，也就是将模拟时间分割成离散的时间间隔，在给定的时间里仅对可能引起电路状态变化的那些元件进行模拟

因此它以做到精确定时，且具有较高的性能和



事件与求值例

时刻	事件	求值	处理新事件
$t=0ns$	$A=0$ (初始条件)		将与 A 相连的 Buffer 1 放入队列 $t+1$ 中;
$t=1ns$	$A=0 \rightarrow 1$ (事件)	查询对应的队列 $(t+1)$, 找到 Buffer 1 求值, 由于 $Rise\ Delay=2ns$, 确定新事件 (即 B 的上跳变) 将在 $2ns$ 后发生;	将与 B 节点相连的 Buffer 2 和与非门、或非门放入 $t+3$ 的队列中 (该队列在 $2ns$ 后到达当前时刻)
$t=3ns$	$B=0 \rightarrow 1$ (事件)	查询对应 $t+3$ 队列, 找出三个元件, 分别求值, 若门 2、3、4 的延迟时间分别为 $2ns$, $3ns$ 和 $3ns$;	将新的事件放入以后相应的队列, 即门 2 放入 $t+5$ 的队列, 门 3、门 4 放入 $t+6$ 的队列中;



第二章 ASIC 设计流程和方

法 模拟结果分析

2 进制,8 进制或 16 进制的数据文件

c r i o

l s n u

k t t

0 0 0 L

1 0 0 H

1 0 0 H

波形图

商业化软件：

Cadence Verilog-XL

Aldec

test - ViewSim for Windows Console - G:\yu\

File Edit View Signals Simulate Help

Simulation stopped at 10.0ns.
 SIM> h a
 SIM> s 100
 Simulation stopped at 20.0ns.
 SIM> l a
 SIM> h b
 SIM> s 100
 Simulation stopped at 30.0ns.
 SIM> h a
 SIM> s 100
 Simulation stopped at 40.0ns.
 SIM>

Commands / Compilation /

Ready 40.0ns

File Edit View Waveform Marker Window Help

Ctrl	Name
1	A
2	B
3	B1
4	C

Time: 0.0m TickMark: 10n Ticksiz: 100

ViewDraw - [sch\A]

File Edit View Add Project Tools Analog Window Help

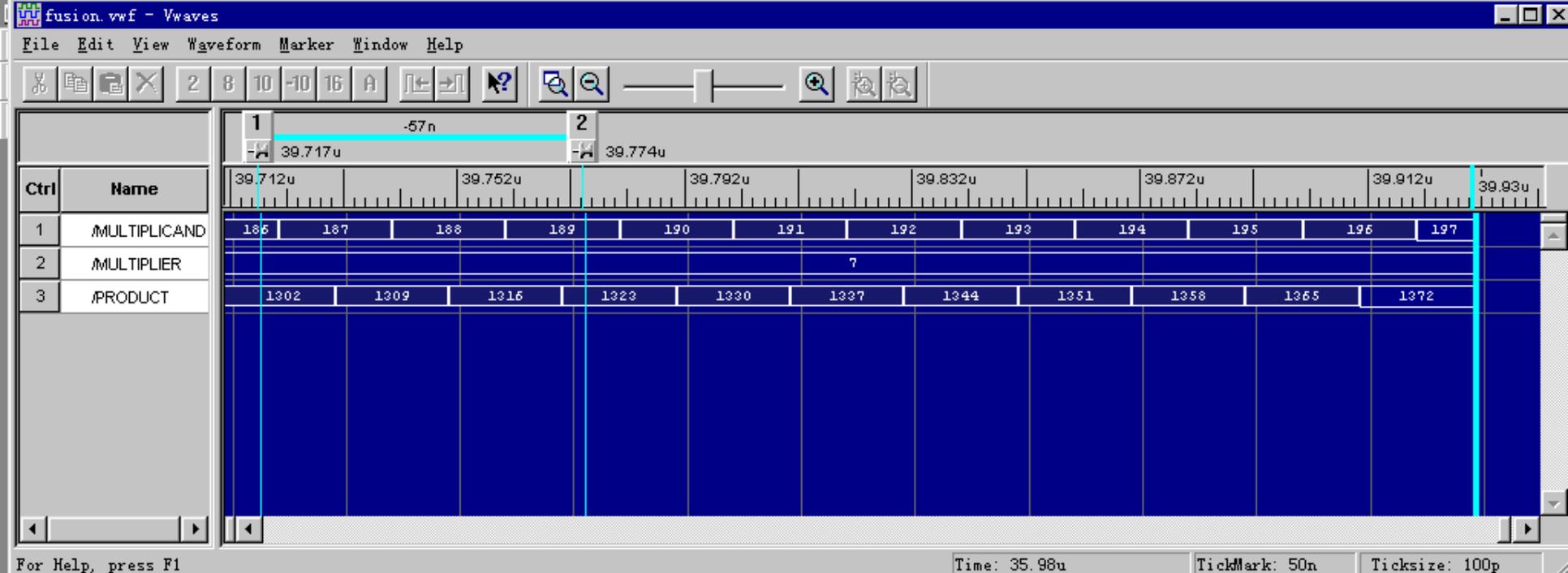
0 objects selected Select Sheet 1 NUT

fusion.vwf - Vwaves Signal Browser

fusion.vwf

Ctrl	NAME: *(4 of 4)
1	A
2	B
3	B1
4	C

Search... Filter... Add Add All Close Help



vhdlut - Speedwave for Windows Console - g:\vhdlut\vhdlut.vpj

File Edit View Signals Simulate Help

```

SIM> net -vhdl -eac top -libfile G:\VHDLTUT\vsslib.ini
VHDL Simulator, v6.102.
Copyright (c) 1987-1997, Vantage Analysis Systems, Inc.
Design processing complete.
SIM> wave fusion.vwf "/MULTIPLICAND" "/MULTIPLIER" "/PRODUCT"
[I] Starting Vwaves process: ....
SIM> sim 1000.0ns
Simulation stopped at 1000.0ns.
SIM> sim 100000.0ns
[Interrupted at time=39930.0ns]
SIM>

```

Commands / Compilation

Ready 39930.0ns delta:2

G:\VHDLTUT\mult.vhd - Source Viewer

File Edit View Signals Simulate Help

Scope History: /

```

120:     end loop first_while_loop;
121:   end process;
122: end stimarch;
123:
124: entity top is
125: end top;
126:
127: architecture toparch of top is
128:   signal multiplier, multiplicand, product : integer:=0;
129:   signal eq : bit;
130:
131: component mult
132:   port (multiplier, multiplicand : in integer;
133:         product : out integer;
134:         eq : in bit);

```

mult.vhd

Ready Ln 133, Col 32



第二章 ASIC 设计流程和方 法

模拟结果

		工 作	不 工作
实际结果	工 作	一 致 A	"保守"情况, 必须减少 B
	不 工作	"危险"情况 必须杜绝 C	一 致 D

表 2.3 模拟结果分析



第二章 ASIC 设计流程和方 法

RTL 级模拟

RTL 级功能模拟是对于 RTL 级描述进行的，电路采用行为描述，激励文件也比较简洁，而且这些与实现的逻辑结构无关，因此模拟速度快，效率高。

硬件模拟 (Quickturn, Aptix)

采用软件模拟的优点是灵活，方便，且费用少，但不足之处是速度慢、验证不充分（受激励文件限制），书写激励文件颇费功夫。

采用硬件模拟的特点如下：

- 处理速度比软件方式要快得多

- 可将实现的“硬件模型”放入实际电路系统中去进行实时验证，验证应充分得多

- 在“硬件模型”实现和模拟控制方面采用软、硬件结合的方式；

- 可以省却编写激励文件的工作。

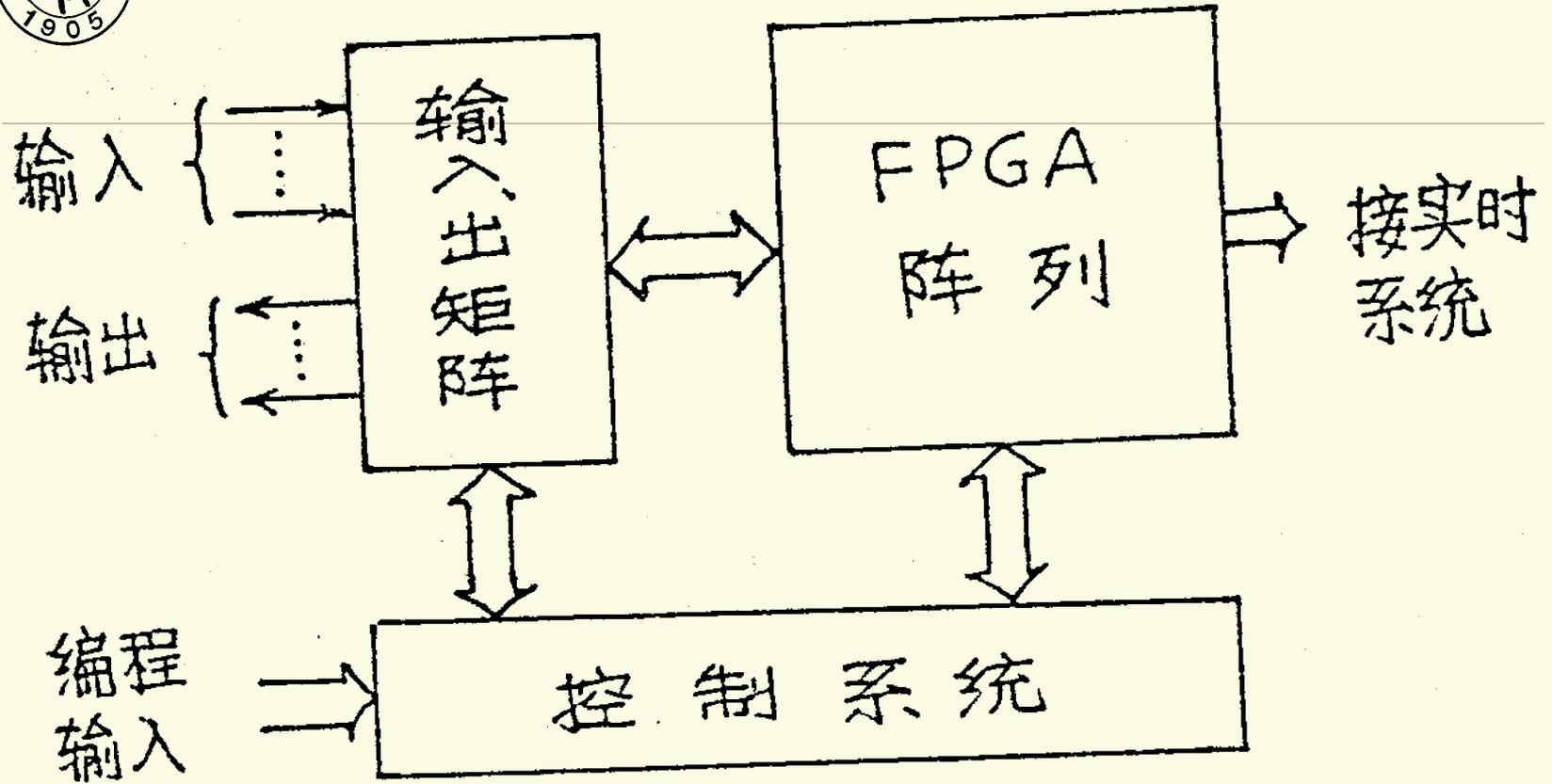


图 2.20 Quickturn 模拟器结构图



第二章 ASIC 设计流程和方 法

2.6.3 时序验证 (Timing Verification)-- 时序验证过程是为了检验电路的时序状况，确定和分析影响数字系统时序的关键路径 (Critical Path)，找到影响电路速度的因素，进一步改进。

动态时序模拟 -- 动态时序模拟是在逻辑模拟同样的环境下进行的，它与功能模拟的不同之处在于器件模型、模拟算法等方面，动态时序模拟采用精确时延的单元模型

静态时序分析 -- 静态时序分析简称为时序分析，它是一种比较新的时序验证技术。时序分析的基本思想是首先分析、找出影响电路时序的最坏情况 (Worst Case)，然后验证此时电路的性能依然能符合定时要求，这样可以确保在其它任何情况下电路都能正常、可靠地工作。



第二章 ASIC 设计流程和方

法 静态时序分析

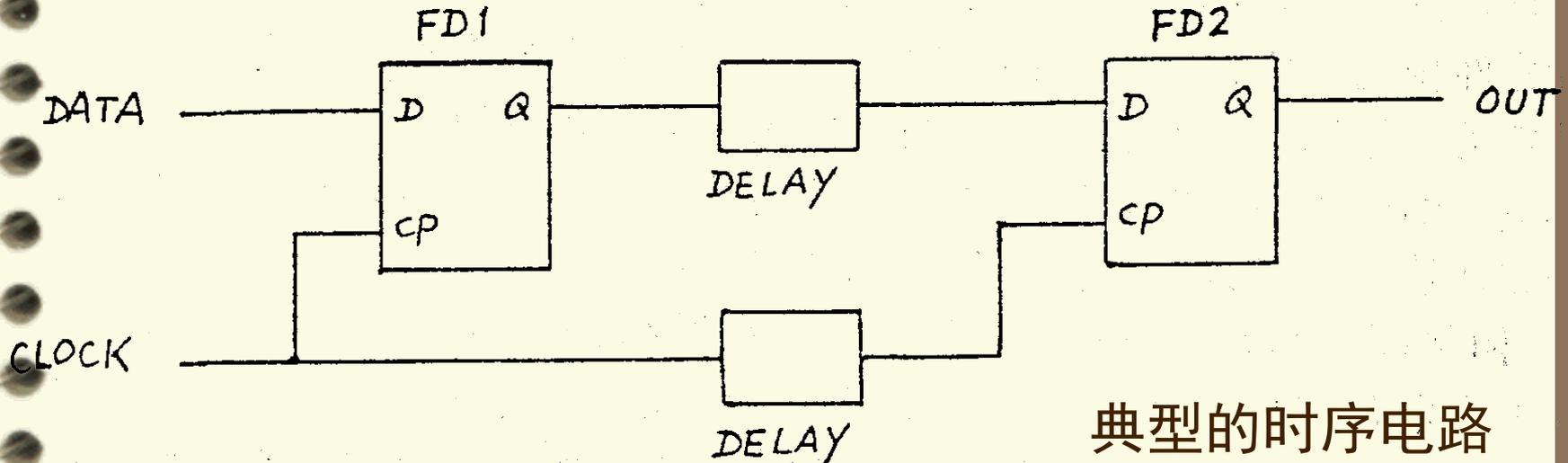
约束条件

max propagation delay (speed)

setup time

hold time

min clock width





第二章 ASIC 设计流程和方 法

动态与静态时序分析比较

动态：速度慢，对于大规模电路很难提供完备的激励，这样会导致验证的不充分，因而不易发现所有的时序错误

静态：不需要任何激励信号，因此速度快、验证充分，它能够找出电路中所有路径的时序错误。但是也正因为没有激励，使静态分析器无从理解电路的功能，以至于会找出实际不存在的“伪路径”



第二章 ASIC 设计流程和方

法
2.6.4 LVS 验证 (Layout versus Schematic)

2.6.5 版图参数提取 (Parameter Extraction, 反标注 (Back Annotation) 和后仿真 (Post Layout Simulation))

2.6.6 设计规则检查 (Design Rule Check) 和电学规则检查 (Electrical Rule Check)

商业化软件

Cadence Dracula

Mentor Graphics Calibra

Avanti Hercules



第二章 ASIC 设计流程和方

法 ASIC 的经济性

ASIC 市场价 = 产品成本 / (1 - 利润比例 %)

ASIC 成本

一次性费用 (Nonrecurring Costs)(NRE) \$10,000-1,000,000

-- 设计人员和资源的费用 + 样品 (Prototype) 生产中的
Mask, testing, packaging 的费用

可变成本 -- ASIC 生产制造的费用

$$C = (W + P) / (N * Y_w * Y_{pa} * Y_{ft})$$

(W+P)-- 流片费 N-- 芯片数 / 每个 Wafer Y_w -- 工艺
流片成品率 Y_{pa} -- 封装成品率 Y_{ft} -- 测试成品率

固定成本 --- 设计环境和测试环境的建立, 人员培训等基础
建设支出



第二章 ASIC 设计流程和方

2.1 ASIC 的经济性 不同 ASIC 类型的成本比较

FPGA > MGA > CBIC > FCC

FPGA 用于量少,上市要求快的产品 (NRE 和固定成本低,可变成本高 (5 倍于 CBIC))

CBIC, FCC 用于产量巨大的产品 (NRE 和固定成本高,但可变成本低,在巨大产量的分摊后,NRE 和固定成本变得不重要)



第二章 ASIC 设计流程和方

法 设计指标 -- 设计指标书的内容如下：

ASIC 芯片总体说明，包括以下细节：

- 芯片及标识符；

- 芯片功能及用途的简要说明；

- 特性说明；

ASIC 的封装及管脚说明：

- 芯片的封装说明及封装图；

- 管脚名及管脚类型；

- 管脚功能的简要说明；

- 管脚信号特性的说明。



第二章 ASIC 设计流程和方

法 ASIC 的使用说明：

直流规格：

包括电源电压、管脚电平和管脚接口特性。其中管脚电平，应指明读管脚采用的是 TTL 或 CMOS 或 ECL 类型的电平，同时应指明电压最小、最大值范围。

交流规格：

通常是指 ASIC 电路的工作频率，包括时钟频率以及输入信号的建立时间和保持时间，输出延迟时间，还包括其它关键信号的定时，例如最小脉冲宽度等。



第二章 ASIC 设计流程和方

法 ASIC 设计的综合因素考虑：

ASIC 设计要求

软硬件的折衷

ASIC 的实现方式

采用的制造工艺及工艺生产线 (Foundry)

测试

封装

开发费用和生产成本

市场



复旦大学专用集成电路与系统实验室

第二章 ASIC 设计流程和方 法



复旦大学专用集成电路与系统实验室





第二章 ASIC 设计流程和方

法 深亚微米设计方法和设计技术

2.10.1 深亚微米工艺给集成电路设计带来的新问题

元件模型变化

电路元件延迟减小，互连线延迟增大
(50~70%, 0.35 μm)

串扰和噪声

时钟线和电源线的影响

功耗和散热问题

铝线的电迁移造成连线断裂

热载流子对 ASIC 可靠性的影响

逻辑与物理的反复设计问题 (0.8 μm --1 次, 0.5 μm --5 次,
0.35 μm -->10 次)



第二章 ASIC 设计流程和方

法 深亚微米设计方法和设计技术

2.10.2 深亚微米设计方法和设计技术的改进

高层次设计规划 (Floorplanning)

在行为级验证成功，进入寄存器传输级设计中生成 RTL 模块的物理抽象，进行预布局，结合物理特征，得到布局、时序及面积以及互连线信息，由此产生的综合优化的约束条件，便综合生成的门级时序得到较好的控制；

经门级功能及时序验证，并生成门级物理抽象，设计规划进行更精确的布局探索和各模块驱动、延迟的分析计算，并精确地得到关键路径"的延时和电路时序；

在物理级，将门级设计得到的驱动、延迟信息作一规划分析并作为时序驱动布局布线的约束条件



第二章 ASIC 设计流程和方 法

综合优化技术 --- 进入深亚微米设计阶段，由于互连线延迟超过单元延迟，综合技术必须考虑由此引起的时序问题，改进电路时序特性有以下几点措施：

使用预布局得到的互连线模型替代原来的连线负载模型，原来模型是对指定工艺库单元的扇出和 RC 树的统计模型，它没有考虑深亚微米连线的种种影响，而互连模型是使用设计规则工具得到的互连特性模型。它能比较精确地反映互连延迟、分布特性及 RC 特性

使用设计规划工具得到的时序约束和互连线模型去驱动综合优化过程，由此得到满足时序要求的综合结果。在物理级，将门级设计得到的驱动、延迟信息作一规划分析并作为时序驱动布局布线的约束条件



第二章 ASIC 设计流程和方

法 将综合技术与版图设计算法结合起来，产生基于布局的物理综合工具，它将时序约束、逻辑网表和布局拓扑关系一起进行分析、调整。例如，根据电路驱动与负载情况，调整缓冲器和驱动单元的大小和布局；根据时序要求，减少可能存在的长连线及并行走线，减少时钟线的影响，生成较优的时钟树布局。

总之，使布局布线能满足电路的时序要求。

采用行为级综合技术，这种高层次综合的任务是实现从系统级算法描述到底层结果级表示的转换，其核心技术是调度和分配。调度 (scheduling) 是将操作赋给所指定的控制步，在满足约束条件下使得给定的目标函数（例如控制步数、硬件资源、延迟和功耗）最小。分配是将操作和数据赋给相应的功能单元和寄存器，其目标是使所占用的硬件资源最少。



第二章 ASIC 设计流程和方

模拟技术— 模拟是设计的基础，从行为级、RTL 级到门级，从逻辑功能模拟、时序模拟到故障模拟，模拟过程就是验证的过程。对于深亚微米设计，设计的数据巨量增加，电路的时序复杂性等对模拟技术提出更高的要求。总的来说，近年来模拟技术有以下几方面的发展：

传统的线性延迟模型不再适用，需要建立考虑高速、低电压、低功耗以及负载和工艺影响的模型；精碗的模型可以保证电路功能和时序的设计正确，这是 ASIC 设计过程的核心

门级模拟过程变为先进行单位延迟的功能模拟，得到门级网表，然后使用设计规划工具估算由于互连线、负载、输入信号变化速率等影响产生的延迟信息，再将它们和网表及功能模型一起模拟，得到门级功能和时序结果



第二章 ASIC 设计流程和方

法采用基于时钟的节拍式 (Cycle-Based) 模拟技术，它比传统使用的事件驱动技术要快几个量级。这种技术通常适用于同步电路的功能验证，但不能作时序验证。Synopsys 公司近年推出的 CycloneRTL 级迷你软件，是利用高层次节拍模拟技术开发的，适用于 VLSI 的 RTL 级设计及相应的测试程序进行仿真，由于不需要象常规的将 RTL 级描述转换为门级描述长时间的编译步骤，使模拟时间大大缩短。

深亚微米电路的时序分析成为设计的中心问题，静态时序分析是解决电路时序问题行之有效的方法



第二章 ASIC 设计流程和方

布图技术—布图技术是集成电路芯片物理设计的关键技术，虽然早在八十年代；已经实现了布局布线自动化，但由于深亚微米设计中又出现了时序问题和设计数据量巨大的问题，显然采用以前的布图技术是无法解决：

时序驱动 (Timing Driven) 和性能驱动 (Performance Driven) 是近几年布图技术发展的方向。对于 VLSI 芯片设计来说，希望有一种快速的时序驱动，性能驱动的布局布线技术，这样可以在短时间内试探多种布图的可能性，而且可以将快速预布局、预布线得到的结果反馈给综合优化工具，对网表和时序进行优化，这种技术也是物理设计规划工具的基础

进入深亚微米阶段，使用层次化设计和购买知识产权 (IP) 模块的可能性增多，因此布局布线工具应能灵活地处理模块，能够把 IP 模块、第三方厂商提供的模块和设计人员自己开放的模块有机地组合起来



第二章 ASIC 设计流程和方

法 深亚微米的布线设计是一重要课题，金属线的层数已从二、三层上 升到六层左右。因此首先要支持多层布线，可以进行通道式或基于区域的布线，也可以是二者混合型的。通道式布线是一传统方法，它要求单元排列成行，行与行之间留出互连线通道，通道宽度可以调节，以保证 100% 约有通率。另一种区域布线是假设单元布局固定，在确定的区域内完成布线。前一种方法无法预测芯片大小，后一种方法比较死板苛求两者的优化组合是所谓混合型布线方法。布线还要考虑采用宽线条克服电迁移问题，避免高频串扰，以及对平行线分布电容计算，对电源线、时钟线也应合理分布，以保证时序要求

总之提高布图质量和布通率，满足时序要求是布图的目标。



第二章 ASIC 设计流程和方

法

内嵌式系统和软硬件协同开发技术 --- 内嵌式系统是硬件与软件协同设计实现特定要求的系统，在内嵌式系统中，通常包括有微处理器模块、专用电路模块以及存放应用软件代码的 ROM、RAM 等。在设计过程中需要硬件与软件紧密配合，共同完成一定的电路功能，所以也称为硬软件协同设计

设计重用方法 (Design Reuse)-- 片上系统的设计是极其复杂的，采用设计重用方法是行之有效的。设计重用方法是将 ASIC 设计中核心部分的设计可以不用修改或只作少量修改就可用在其它的设计之中。也就是说，对于一些有价值的模块或 IP(Intellectual Property)，设计一次，可以使用多次。

设计重用概念上是简单的，但是实现起来也有一定的难度。首先 设计重用方法需要一定的设计环境和设计工具，常规的 Top-Down 设计方法应作适当扩展。首先要建立设计重用模块的系统级模型及相应的模块库，也就是用 VHDL 或 Verilog 语言编写行为级模型，进行行为级验证然后是综合优化，直到物理设计和工艺制造，经测试和试用证实设计正确无误后，



第二章 ASIC 设计流程和方

法 才能把该模块的行为级模型、RTL 级模型等存入重用模块库。因此设计工具应具有对重用模块的建立, 修改, 调用和管理的功能 也应具有对重用模块和其他方式生成的模块协同设计和界面格式转换的能力。

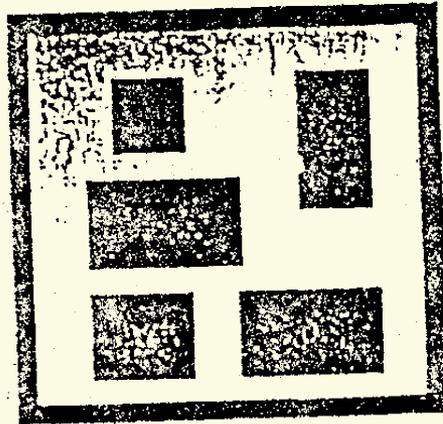
设计重用的应用一般有两种情况。一种情况是重用模块包含了行为级模块和 RTL 模块, 也就是事先已经把模块的行为级描述综合成适合于某一工艺过程的形式, 设计时只需要将 ASIC 行为级模型分配成专用工艺过程的 RTL 级描述, 然后调用所需的 RTL 级模块, 一起进行逻辑综合。第二种情况是针对某一特定工艺过程, 将设计重用模块除了硬件块外还有软件块。软件块是执行特定操作的一般程序。例如微处理机中的微程序, 它们通常以代码形式放置在 ROM 中, 这种软件硬化的 ROM 也称为固件, 因此软件块一般以 ROM 形式出现。



第二章 ASIC 设计流程和方

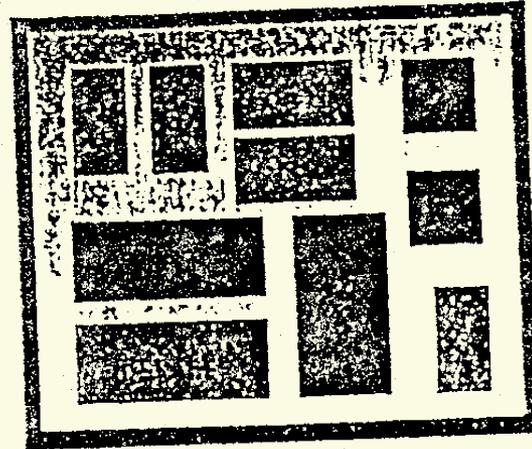
法-*Systems-On-Chip Through Design Reuse*

1994



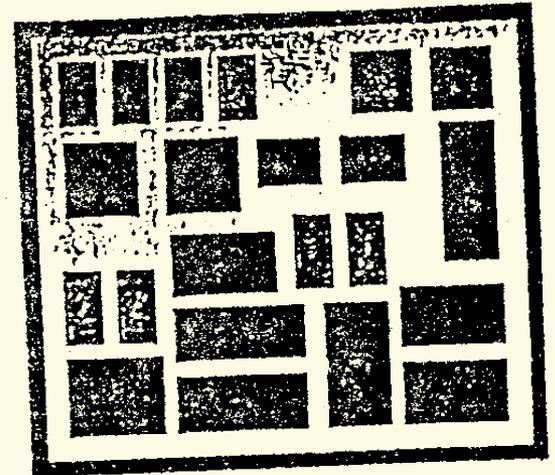
 = 10K gates

1996



 = 20K gates

2000



 = 100K gates

- # of reusable blocks is increasing
- # of gates in the reusable blocks is increasing

图 2.25 设计重用芯片示意图



第二章 ASIC 设计流程和方

法

设计重用也是深亚微米设计的主要方法，设计重用也是对 IP 的再使用，通常可以把设计成功的子模块建成一个核心模块库，以便在以后的 ASIC RTL 级设计中调用。常用的核心模块可以有 MPU、DSP、A/D、D/A、RAM、ROM、输入/输出接口以及加法器、乘法器等。设计重用方法对于超大规模或更大规模的集成电路设计尤其适用。

2.11 集成电路 CAD 技术发展概况

计算机辅助设计 (CAD)

计算机辅助测试 (CAT) 计算机辅助工程 (CAE)

计算机辅助制造 (CAM)

电子设计自动化 --EDA(Electronics Design Automation)



第二章 ASIC 设计流程和方

电子设计自动化 --EDA(Electronics Design Automation)

第一代 -- 绘图及版图图形编辑,七十年代 IC 发展初期

第二代 -- 逻辑模拟和版图设计自动化,八十年代,集成电路已从中规模发展到大规模

第三代 -- 概念驱动设计和 Top-Down 的设计方法 1; 从八十年代后期

第四代 -- 深亚微米设计技术



第二章 ASIC 设计流程和方 法

表 2.6 集成电路设计技术和 CAD 工具发展一览

	设计技术特点	CAD 工具	级别	典型设计系统
第一代 (70 年代)	交互式版图设计及版图设计规则检查	制版语言, 版图图形编辑及 DRC 检查	物理级	Calma, Applicon, Computer Vision.
第二代 (80 年代)	逻辑模拟和半定制版图自动设计及验证	电路图输入、逻辑模拟、门阵和标准单元版图设计及版图验证、通用电路分析.	门级	Mentor, Daisy, Valid, Spice
第三代 (90 年代初)	从 RTL 级 → 门级 → 物理级, 实现 Top-Down 设计自动化	VHDL 和 Verilog 硬件描述语言、行为级模拟、逻辑综合、故障模拟、时序分析、测试生成、可测性设计及模拟电路设计工具.	RTL 级	Cadence, Mentor, Synopsys, Viewlogic.
第四代 (90 年代中期开始)	深亚微米设计技术和片上系统的设计	高层次规划、高层次综合、多层次布图及验证、模拟工具、软硬件协同设计工具及模拟电路设计工具	系统级	Synopsys, Avanti, Cadence, Mentor, Viewlogic.