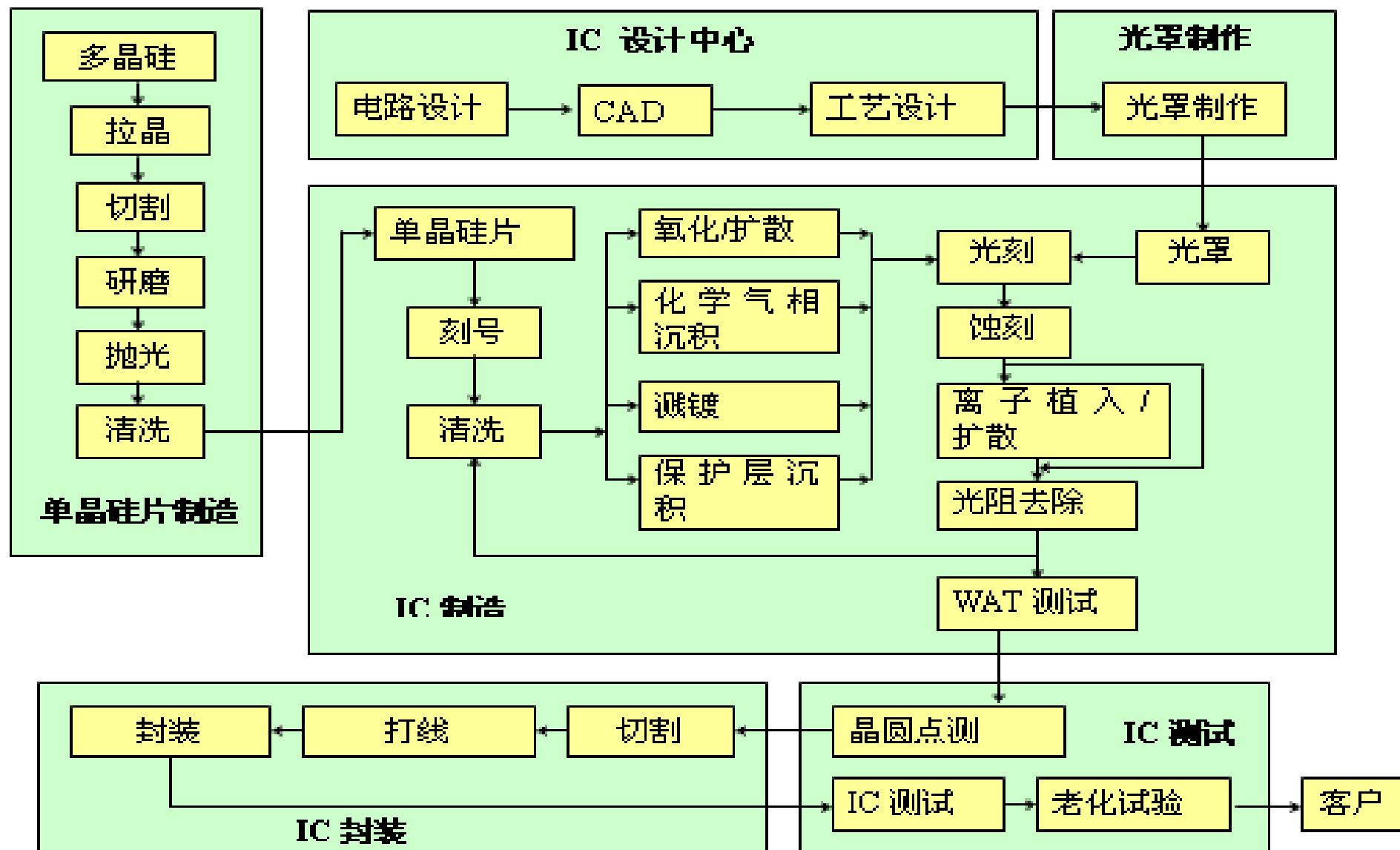


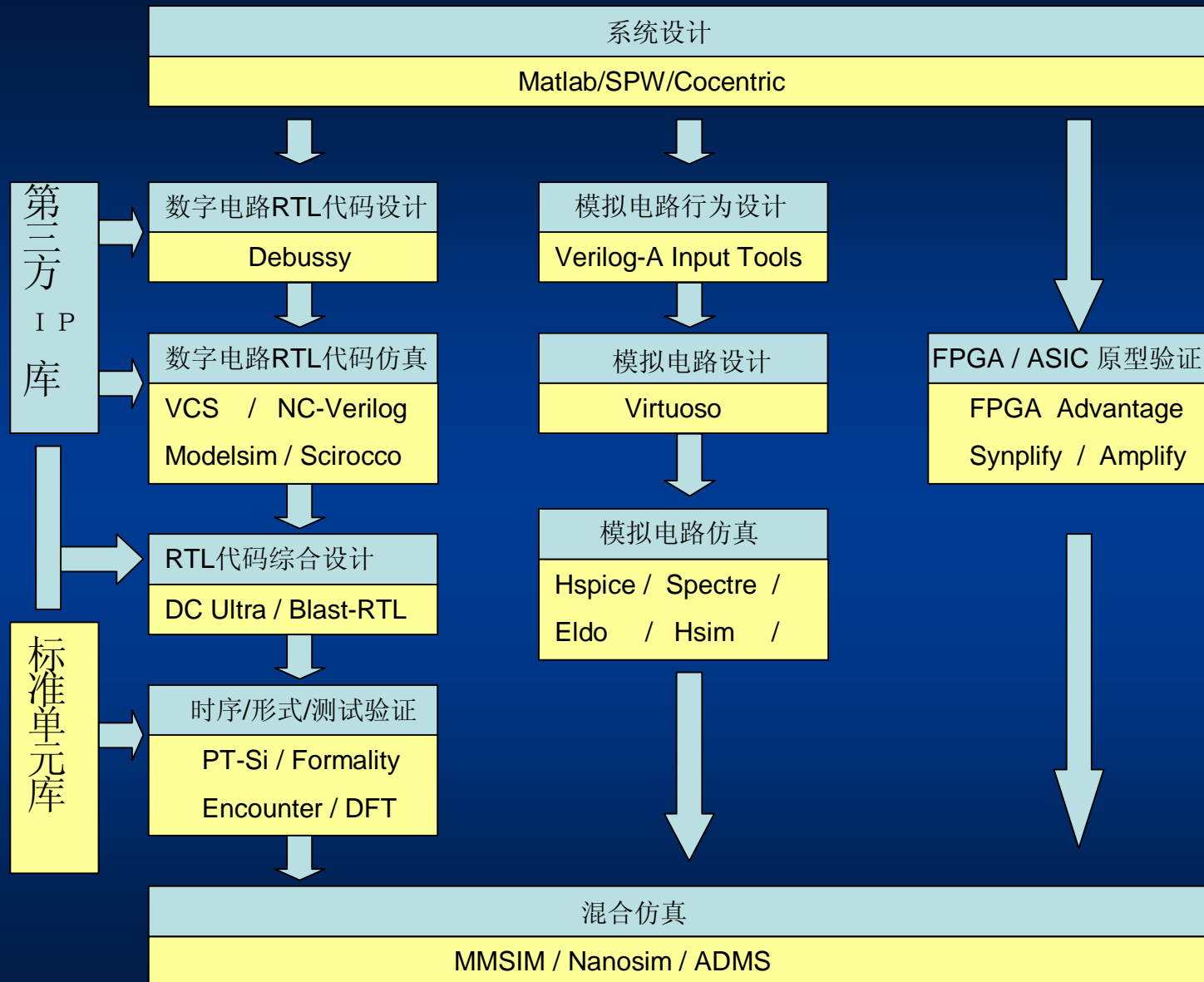
目录

- 1、 Linux培训课程
- 2、 **IC基础知识介绍**
- 3、 DC工具培训
- 4、 ICC工具培训

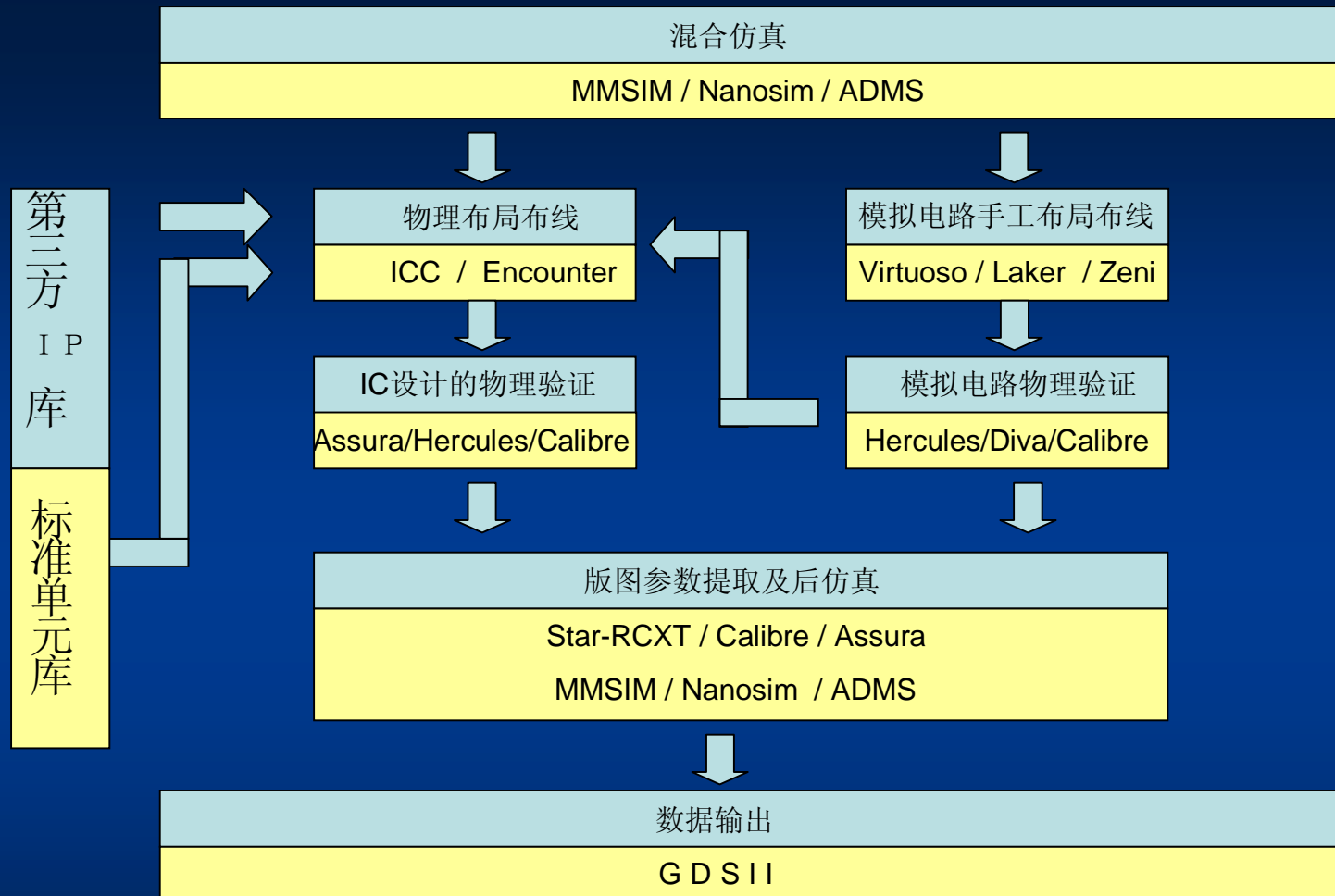
IC制造流程介绍



IC设计流程介绍



IC设计流程介绍



□ IC分类

(1)按其功能、结构的不同分成数字IC和模拟IC (2)按导电类型不同可分成双极型集成

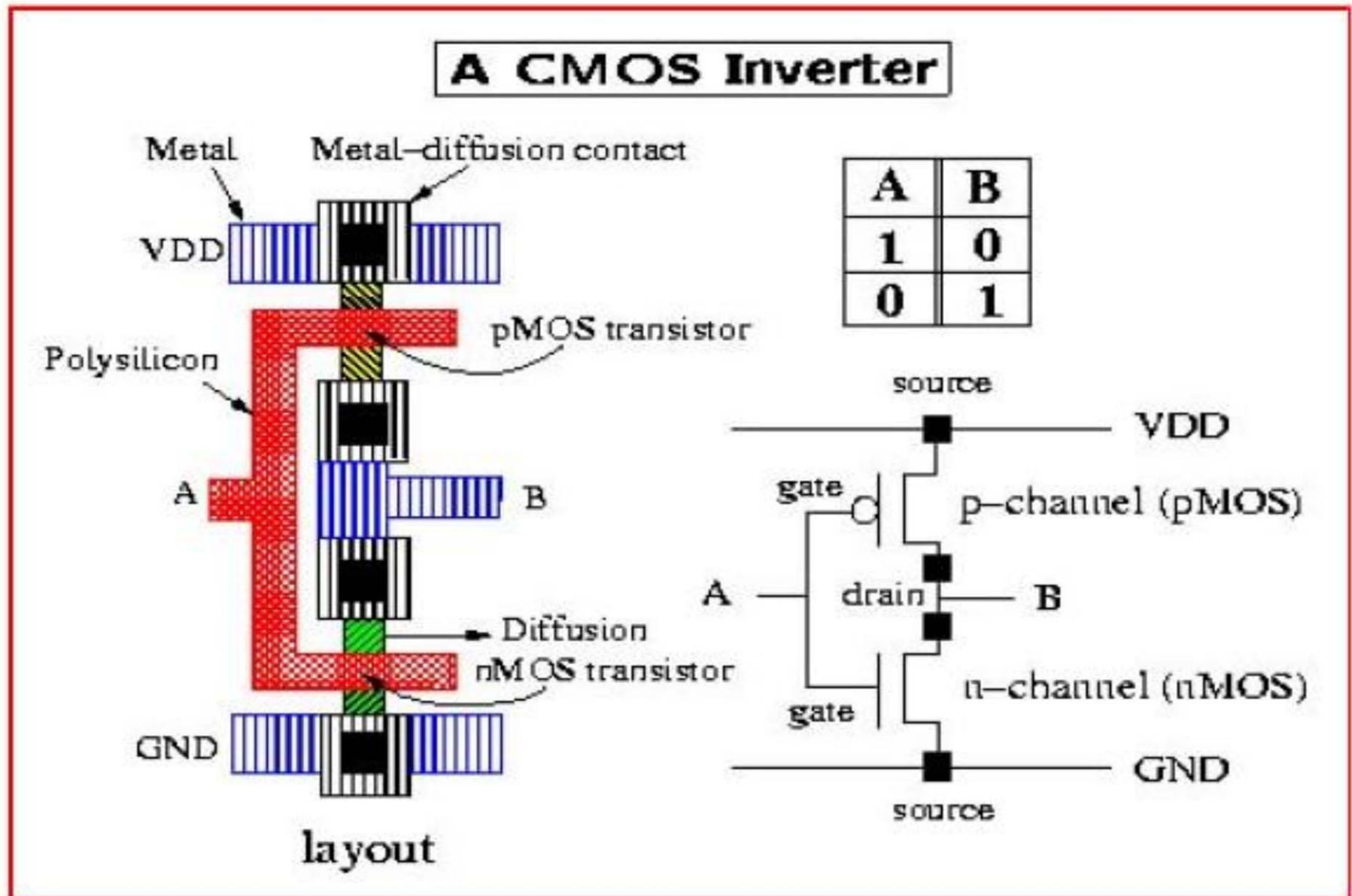
电路和

单极型集成电路

(3)按集成度高低分类小规模集成电路、中规模集成电路、大规模集成电路和超大规模集成电路、SOC片上系统。

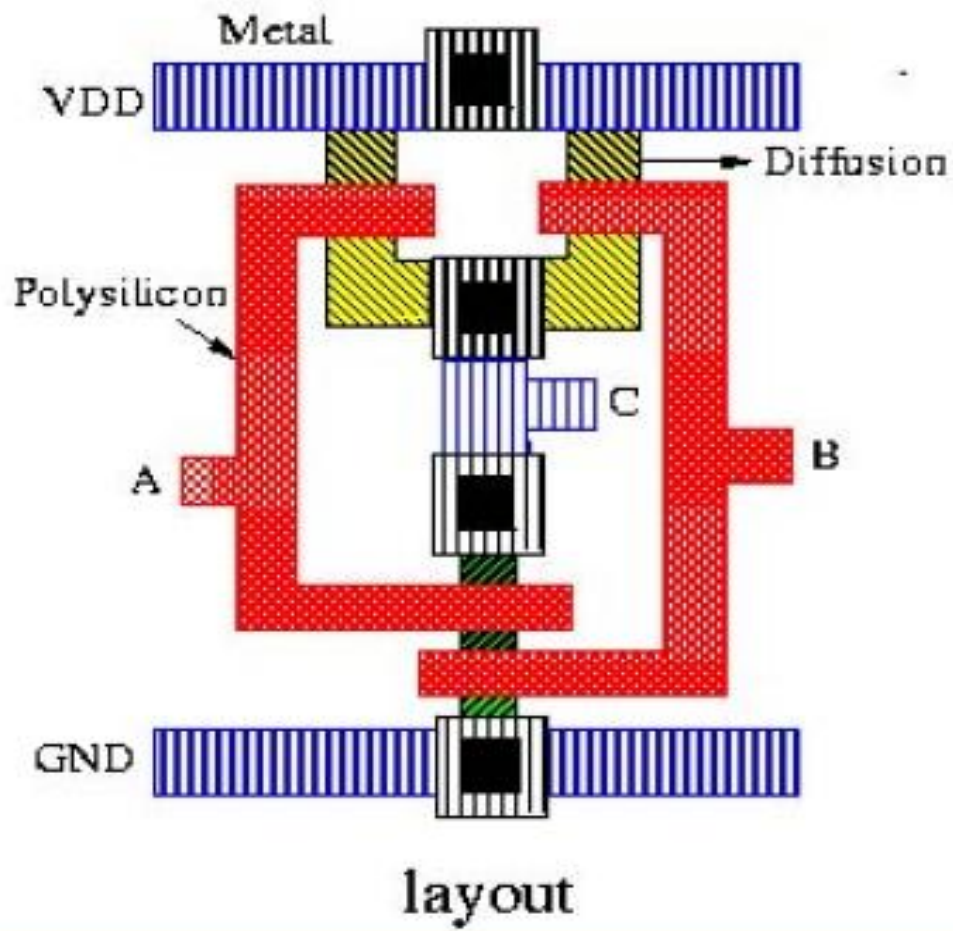
(4)按制作工艺可分为半导体集成电路和膜集成电路。

IC版图介绍



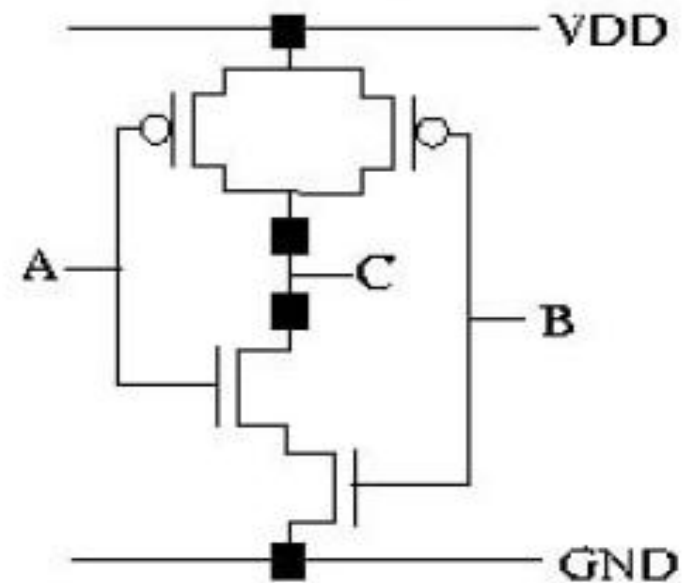
IC版图介绍

A CMOS NAND Gate



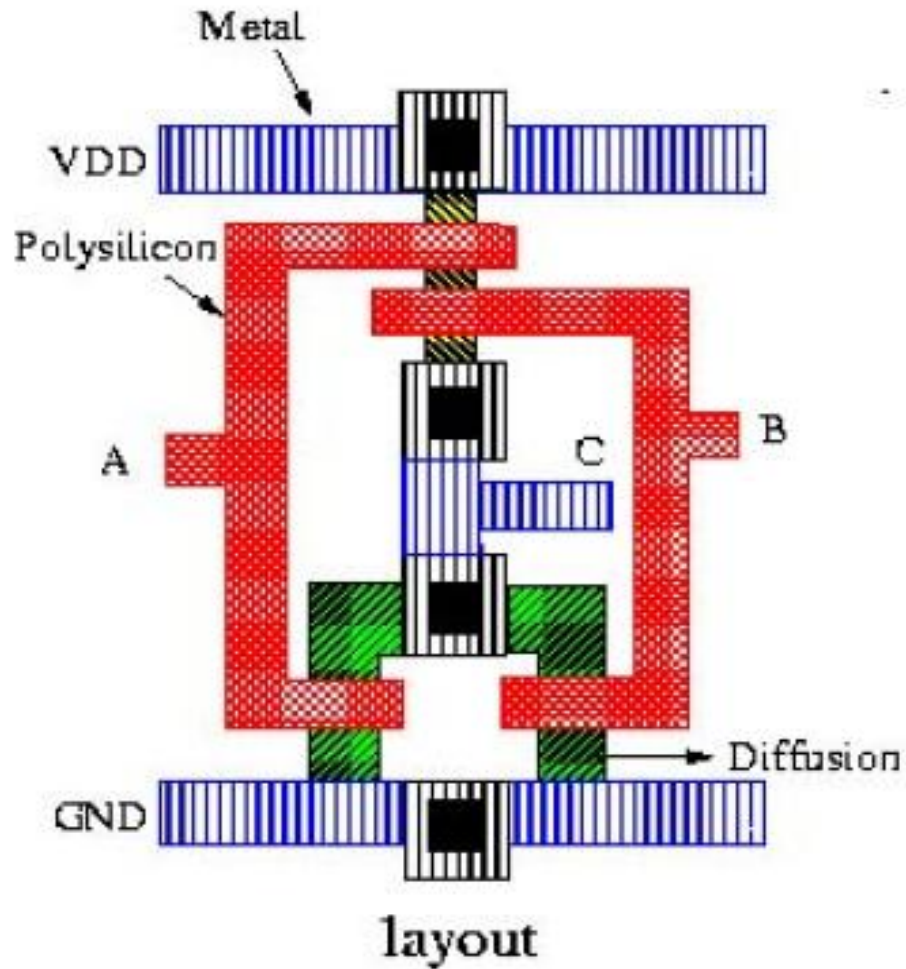
in1 in2 out

A	B	C
0	0	1
0	1	1
1	0	1
1	1	0

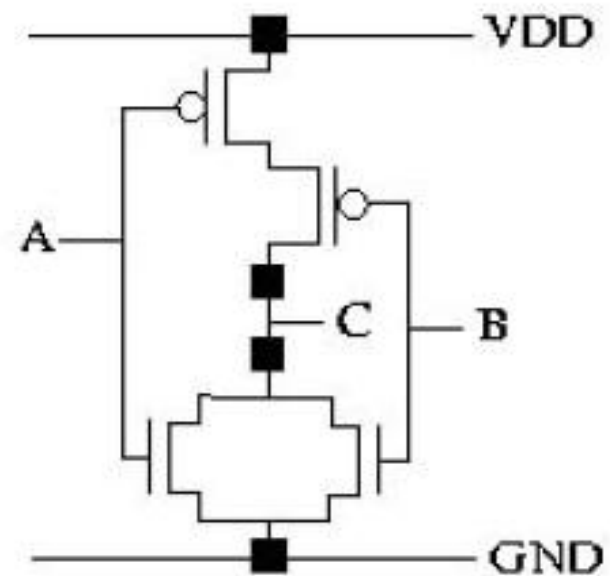


IC版图介绍

A CMOS NOR Gate



A	B	C
0	0	1
0	1	0
1	0	0
1	1	0



Cadence工具介绍

Cadence四个平台工具介绍

Incisive功能验证平台

Incisive Unified Simulator ; NC VHDL

Encounter数字IC设计平台

SOC Encounter ; Nano Encounter;
Encounter RTL Compiler

Virtuoso用户定制平台

Virtuoso Layout Editor; Assure ;

Virtuoso Analog Design Environment

Allegro 系统互连和设计平台

Allegro PCB Editor; Allegro AMS Simulator

Synopsys工具介绍

Galaxy设计平台和Discovery验证平台

VCS 编译型Verilog模拟器

DC 逻辑综合工具

DFT Compiler 扫描式可测性设计分析、综合和验证技术

Power Compiler 功耗优化能力

PrimeTime 静态时序分析的工具。

Formality 形式验证工具

Astro 或ICC 自动布局、布线的工具

Star-RCXT 寄生参数提取工具

HSPICE 模拟电路仿真工具

目录

- 1、 Linux培训课程
- 2、 IC基础知识介绍
- 3、 **DC工具培训**

DC工具培训

1. 综合简介
2. Setting Up and Saving Designs
3. Area and Timing Constraints
4. Partitioning
5. Timing Analysis

综合简介

综合是使用软件的方法来设计硬件，然后将门级电路实现与优化的工作留给综合工具的一种设计方法。它是根据一个系统逻辑功能与性能的要求，在一个包含众多结构、功能、性能均已知的逻辑元件的单元库的支持下，寻找出一个逻辑网络结构的最佳实现方案。即实现在满足设计电路的功能、速度及面积等限制条件下，将行为级描述转化为指定的技术库中单元电路的连接。

综合主要包括三个阶段：转换(translation)、优化(optimization)与映射(mapping)。

1、自顶向下的综合方案(top-down)

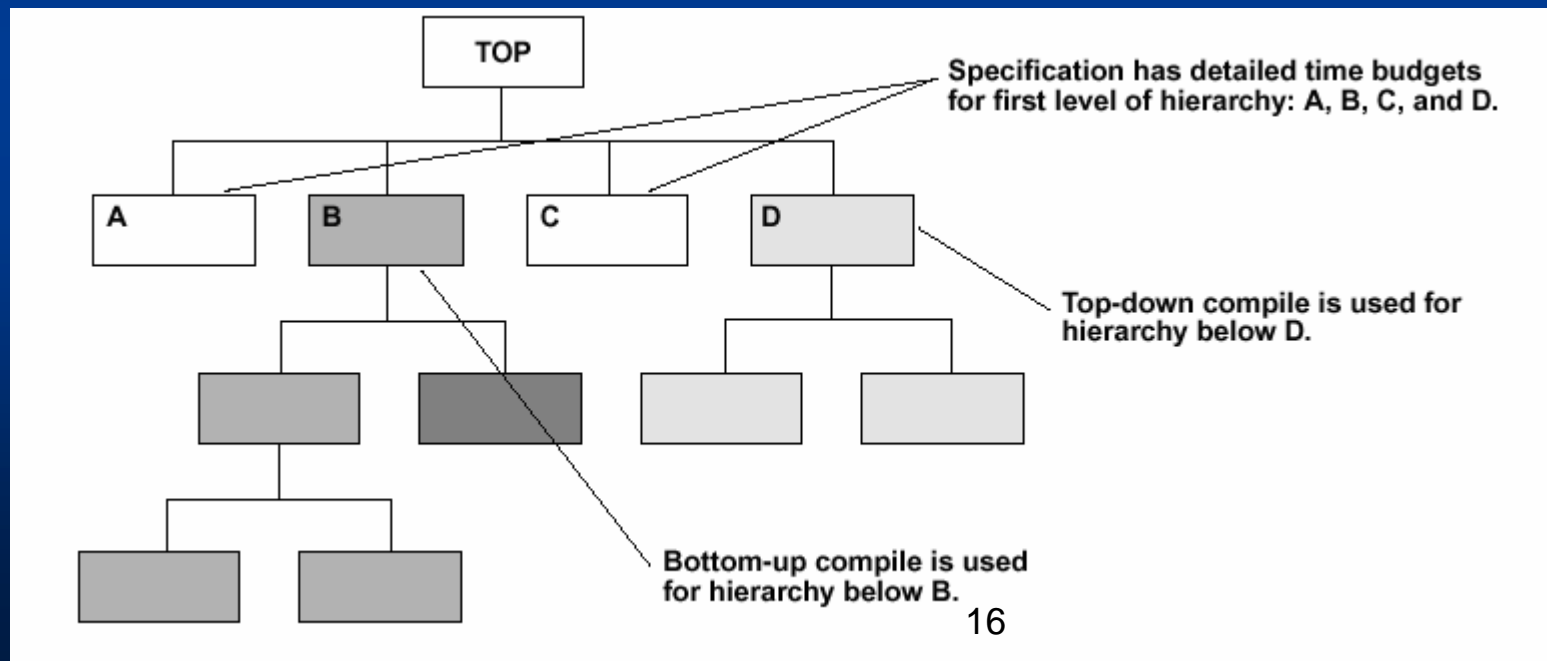
在**top-down**的综合方案中，顶层模块和它的所有子模块放在一起优化，所有的限制条件和电路工作环境一般也都是针对顶层模块设置的，因此，这种综合方案能够自动将模块之间的连接和依赖关系考虑到综合中去，从而优化综合结果。但是，这种方法对于一个规模比较大的电路显然不适合，因为在综合的过程中，所有的模块必须同时存在于存储区中，而且运行时间也会比较长。

2、自底向上的综合方案(bottom-up)

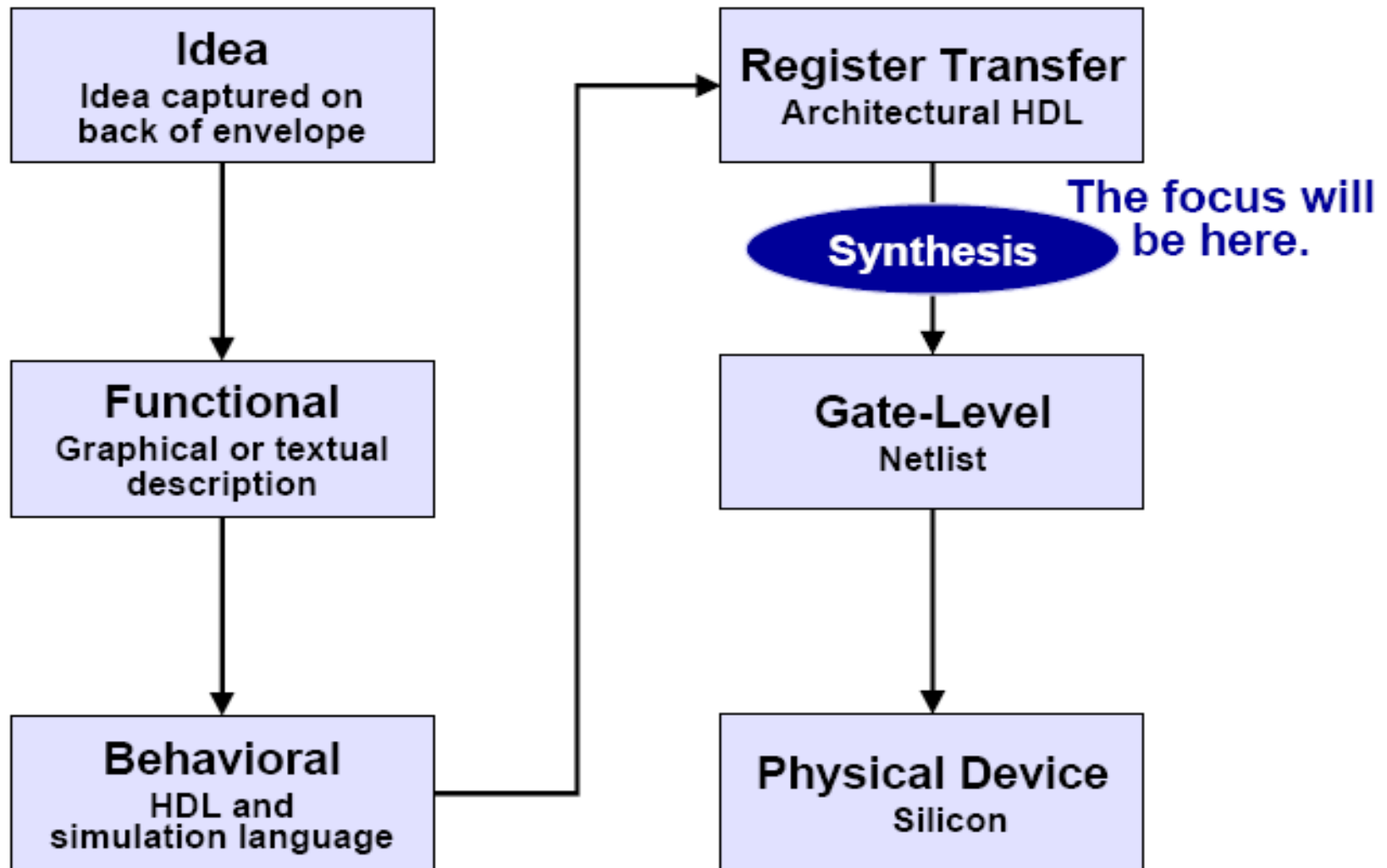
即分而治之的解决方法。从最低层模块开始综合优化，必须对所有底子模块施加限制条件并且需要单独优化，这个过程一直延续到顶层模块。其中每一个子模块综合完成之后，直接将该模块集成到它的上一级中和上一级的其他模块一起优化，而且一般使用 `set_dont_touch` 命令让DC不再改变该子模块的结构。使用这种方法进行优化的优点是当工作站的处理能力不足时，可以分别进行子模块的优化，而不需要将所有的模块都放到存储区中，这种做法的缺点是只能在子模块内部进行优化，无法考虑到模块周围的环境而将子模块和其他的模块一起优化。

综合简介

3、综合采用自底向上和自顶向下的综合方案 考虑到以上两种综合方案的有缺点，在做综合的时候一般同时采用以上两种方案。即对于层次较多或者比较复杂的模块采用自底向上的综合，而对于层次结构简单底模块则使用自顶向下底综合。如下图**D**子模块采用自顶向下底综合，**B**子模块采用自底向上底综合。



Levels of Abstraction



Synthesis In this Workshop

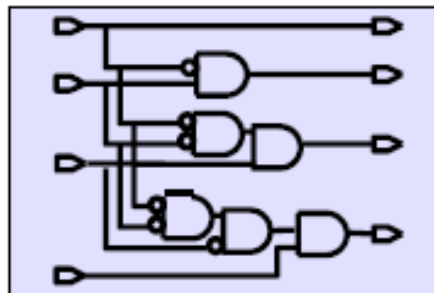
Synthesis = Translation + Logic Optimization + Mapping

```
residue = 16'h0000;  
if (high_bits == 2'b10)  
    residue = state_table[index];  
else  
    state_table[index] = 16'h0000;
```

HDL Source

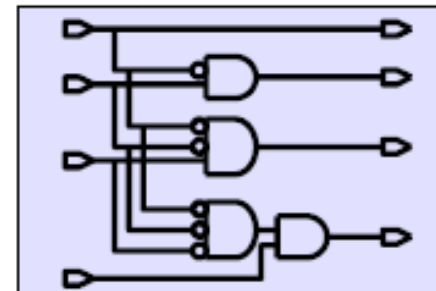
Constraints
Set_max_area...
Create_clock...

Translate (read)



**Generic Boolean
(GTECH)**

**Optimize + Map
(compile)**

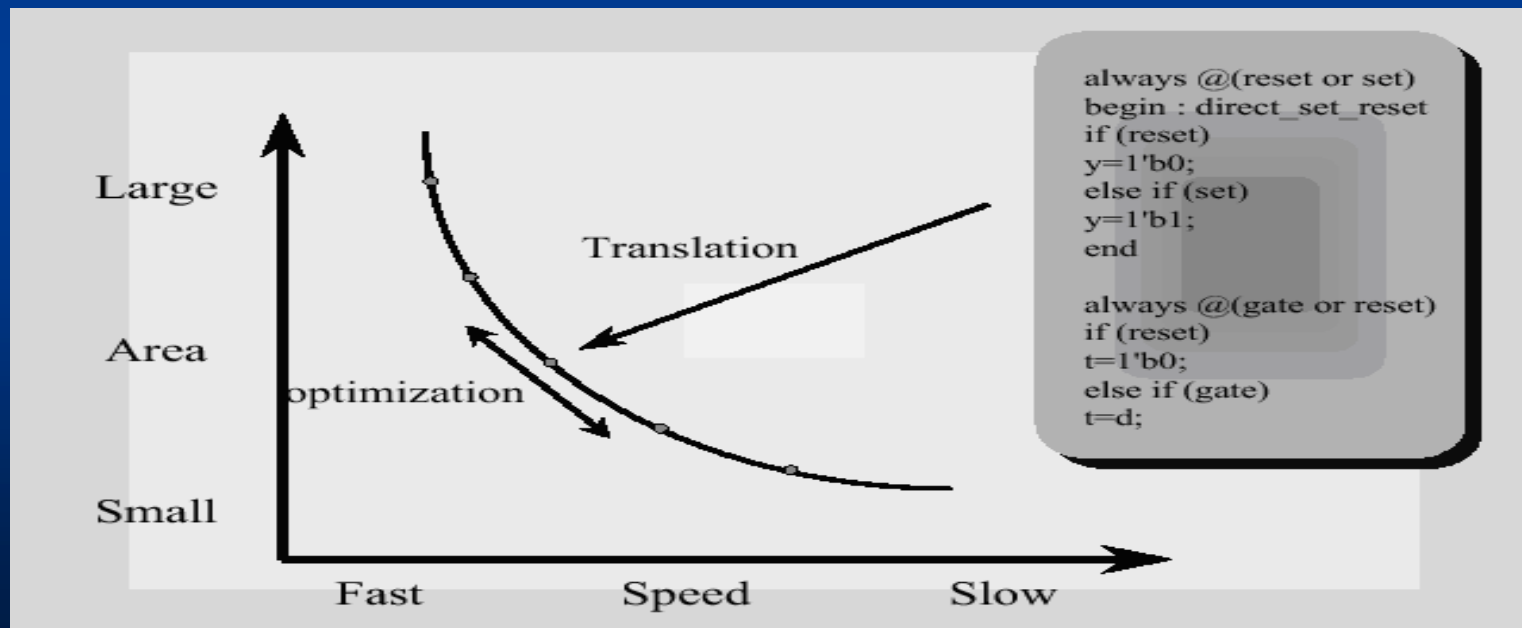


Target Technology

综合简介

综合是综合工具对电路一些的性能进行折中的结果，对于数字电路来说就是在电路的面积与功耗，面积与时序上的性能进行折中。

下图是综合曲线的示意图，可以看出，当路径的延迟减小到一定程度，面积上的代价就不能明显的改善时序上的性能；同样，当面积减小到一定程度，路径延迟的增大也不能明显改上面积上的性能。



Setting Up and Saving Designs

Setting Up and Saving Designs

1、target_1library变量： 是DC中保留的变量，这个变量指定的库是DC用来构建一个电路的。DC进行映射的几个步骤：设置它让它指向你的生产商提供的库文件。使用厂家提供的工艺库中的时序数据计算这些门的时序。从target_1library指定的库中选择功能正确的门。设定目标库使用如下语句：

```
set target _1library "sc_max.db"
```

Setting Up and Saving Designs

2、link_library 变量： 是用来解析设计参照的。

set link_library “*sc_max.db”

*号代表DC 内存，后面接着的是指定的目标库（如果这个变量没有加入目标库的名字则可能会存在单元不能解析的问题）。link_library 中指定的工艺库的名字只与用做参照的单元库有关，在link_library 指定的工艺库不是为了给DC 做优化判断的。它主要用在综合的转换阶段用以生成GTECH 中间文件。

Setting Up and Saving Designs

3、search_path 变量:

可以给DC 指定寻找文件时的工作路径。解析时DC 首先，寻找内存，然后寻找 link_1library 变量指定的库文件。接着，DC 寻找所有的search_path 变量中指定的路径。

../ref/db

./scripts

Setting Up and Saving Designs

4、symbol_library: 这个系统变量指定的库包括了工艺库中单元的图形sc.sdb,当使用图形化的前端工具时,它可以用来表示门级的逻辑图,如果不指定则工具使用默认的generic.sdb。