

ASIC 设计

ASIC 作为现在的一个热门技术，受到了很大的关注。本文将对 ASIC 做一个简介。

一、什么是 ASIC

ASIC 是 Application Specific Integrated Circuit 的缩写，中文意为“专用集成电路”。它与批量生产地通用芯片不同，通常是应特定用户要求和特定电子系统的需要而设计、制造的集成电路，一般产量较小、可靠性更高、速度更快。

二、ASIC 的发展历史

谈到 ASIC 的历史就不得不从集成电路（IC）的历史讲起，因为 ASIC 是从用途上分的，其本质还是集成电路。

2.1 集成电路的简史

1947 年贝尔实验室（Bell Laboratory）发明了晶体管，随后肖特基（Schokley）在 1949 年发明了双极性晶体管（Bipolar Transistor）。直到 1956 年才出现了第一个双极性数字逻辑门，这是由 Harris 发明的分立元件构成的。1958 年，美国仪器公司（Texas Instruments）的 Jack Kilby 提出了集成电路的设想，Jack Kilby 因这一突破性的构想而获得了诺贝尔奖。随后 1962 年，仙童半导体（Fairchild Semiconductor）推出了真正成功的 IC 逻辑系列 TTL（Transistor-Transistor Logic），1974 年出现了 ECL（Emitter Coupled Logic）系列。同时，也造就了一大批半导体大公司，如仙童，仪器，国家半导体（National Semiconductor）等。这一逻辑系列直到 20 世纪 80 年代都一直占据着数字半导体市场的主要份额。但 MOS 集成电路最终取代其，在数字 IC 中占据了支配地位。

MOSFET（Metal Oxide Silicon Filed Effect Transistor）原称 IGFET（Isolated Gae Field Effects Transistor），其基本原理是早在 1925 年由 J.Lilienfeld（加拿大人）在一项专利中提出的，1935 年 Q.Heil 也在英格兰独立提出了这一理论。然而由于对材料和门的稳定性问题认识不足，是这个器件的实际使用推迟了很长一段时间。这些问题一经解决，MOS 数字集成电路在 20 世纪 70 年代早期就开始应用了。令人惊奇的是，最初提出的 MOS 逻辑门是 CMOS 类型的，并且这一趋势继续到 20 世纪 60 年代末。制造工艺的复杂性使这些器件的完全使用又推迟了 20 年。而不同的是，第一个使用的 MOS 集成电路是仅用 PMOS 逻辑来实现的。数字集成电路革命的第二个时代无疑是 Intel 公司在 1972 退出的微处理器 4004 和 1974 年推出的微处理器 8008。此后，MOS 集成电路便占据了数字集成电路的支配地位。^[1]

2.2 ASIC 的发展状况

在可编程逻辑器件出现之前，ASIC 占据了专用领域的绝对地位，因为当时的人们无从选择。但在 20 世纪 70 年代，相继出现了初级的可编程逻辑器件 PLA

PAL、GAL等，20世纪80年代FPGA和CPLD相继出现。可编程逻辑控制器件的成本低、开发周期短、风险小，于是专用领域里的应用开始倾向于可编程逻辑器件。

但是早期的可编程逻辑器件速度慢、编程门数少等因素，使人们在中高端应用时有不得不采用ASIC技术。随着ASIC和可编程逻辑器件各自技术的不断发展，人们总是不断地在两者之间摆来摆去。

最近两年FPGA和CPLD的制造工艺有了很大的提高——最先进的FPGA的制造工艺已经达到了40nm——加之ASIC成本高昂、风险较大以及金融危机的影响，于是人们又把目光投向了可编程逻辑器件。

总之，ASIC和可编程逻辑器件是螺旋式发展的。

2.3 ASIC 的发展的重要因素

20世纪80年代以后，半导体制造工艺的不断提高、EDA工具的快速发展，使得ASIC技术有了迅猛发展。

2.3.1 半导体制造工艺的提高

20世纪80年代初，半导体的制造工艺在1 μm 以上，随后制造工艺不断缩小：0.8 μm （1993年）0.25 μm （1999年）0.13 μm （2002年）90nm（2004年）65nm（2006年）45nm（2007年）。随着工艺的不断缩小，芯片的速度越来越快，集成度也越来越高，这些都意味着芯片的功能越来越强大。此外，多阈值工艺的发展也使芯片更加省电。

2.3.2 EDA 工具的发展

1971年，Intel推出了世界上第一块微处理器——Intel 4004，它片集成了2250个晶体管，它几乎完全是人工设计的。但随着IC集成度几何级数的提高，人工设计显然不再可能。于是，出现了早期的计算机辅助设计（CAD）技术，后来逐渐演变为计算机辅助工程（CAE）。到20世纪90年代，终于演变为电子设计自动化（Electronic Design Automation，EDA）。EDA技术贯穿了IC设计的全流程——前端设计、仿真以及后端设计、仿真。EDA技术使IC设计大大简化，并缩短了设计时间，提高了设计可靠性。

三、ASIC 设计

本章将详细介绍ASIC电路的原理及设计。

3.1 ASIC 的类型

设计方法而言，设计集成电路的方法可以分为全定制、半定制两种方式。

3.1.1 全定制设计

全定制ASIC是利用集成电路的最基本设计方法（不使用现有库单元），对集成电路中所有的元器件进行精工细作的设计方法。全定制设计可以实现最小面积，最佳布线布局、最优功耗速度积，得到最好的电特性。该方法尤其适宜于模拟电路，数模混合电路以及对速度、功耗、管芯面积、其它器件特性（如线性度、对称性、电流容量、耐压等）有特殊要求的场合；或者在没有现成元件库的场合。特点：精工细作，设计要求高、周期长，设计成本昂贵。

由于单元库和功能模块电路越加成熟，全定制设计的方法渐渐被半定制方法

所取代。在现在的 IC 设计中，整个电路均采用全定制设计的现象越来越少。全定制设计要求：全定制设计要考虑工艺条件，根据电路的复杂和难度决定器件工艺类型、布线层数、材料参数、工艺方法、极限参数、成品率等因素。需要经验和技巧，掌握各种设计规则和方法，一般由专业微电子 IC 设计人员完成；常规设计可以借鉴以往的设计，部分器件需要根据电特性单独设计；布局、布线、排版组合等均需要反复斟酌调整，按最佳尺寸、最合理布局、最短连线、最便捷引脚等设计原则设计版图。版图设计与工艺相关，要充分了解工艺规，根据工艺参数和工艺要求合理设计版图和工艺。

3.1.2 半定制设计方法

半定制设计方法又分成基于标准单元的设计方法和基于门阵列的设计方法。基于标准单元的设计方法是：将预先设计好的称为标准单元的逻辑单元，如与门，或门，多路开关，触发器等，按照某种特定的规则排列，与预先设计好的大型单元一起组成 ASIC。基于标准单元的 ASIC 又称为 CBIC(CellbasedIC)。基于门阵列的设计方法是在预先制定的具有晶体管阵列的基片或母片上通过掩膜互连的方法完成专用集成电路设计。半定制主要适合于开发周期短，低开发成本、投资、风险小的小批量数字电路设计。

1. 基于标准单元的设计方法

该方法采用预先设计好的称为标准单元的逻辑单元，如门电路、多路开关、触发器、时钟发生器等，将它们按照某种特定的规则排列成阵列，做成半导体门阵列母片或基片，然后根据电路功能和要求用掩膜版将所需的逻辑单元连接成所需的专用集成电路。

单元库中所有的标准单元均采用定制方法预先设计，如同搭积木或砌墙一样拼接起来，通常按照等高不等宽的原则排列，留出宽度可调的布线通道。CBIC 的主要优、缺点：

- a. 用预先设计、预先测试、预定特性的标准单元库，省时、省钱、少风险地完成 ASIC 设计任务。
- b. 设计人员只需确定标准单元的布局以及 CBIC 中的互连。
- c. 标准单元可以置放于芯片的任何位置。
- d. 所有掩膜层是定制的；
- e. 可嵌定制的功能单元；
- f. 制造周期较短，开发成本不是太高。
- g. 需要花钱购买或自己设计标准单元库；
- h. 要花较多的时间进行掩膜层的互连设计。

2. 基于门阵列的 ASIC

门阵列是将晶体管作为最小单元重复排列组成基本阵列，做成半导体门阵列母片或基片，然后根据电路功能和要求用掩膜版将所需的逻辑单元连接成所需的专用集成电路。用门阵列设计的 ASIC 中，只有上面几层用作晶体管互连的金属层由设计人员用全定制掩膜方法确定，这类门阵列称为掩膜式门阵列 MGA (maskedgatearray)。门阵列中的逻辑单元称为宏单元，其中每个逻辑单元的基本单元版图相同，只有单元以及单元之间的互连是定制的。客户设计人员可以从门阵列单元库中选择预先设计和预定特性逻辑单元或宏单元，进行定制的互连设计。门阵列主要适合于开发周期短，低开发成本的小批量数字电路设计。

3.2 ASIC 的设计概述

现代 IC 系统设计包括算法设计、软 / 硬件划分、存储器分配、 RTL设计、仿真、验证、综合、静态时序分析、等价性验证、 Floorplan 、时钟树生成、布局布线、可测性设计、低功耗设计、信号完整性分析等容。为清楚起见，我们将 IC 设计分为三个部分：系统设计、电路 / 逻辑设计、物理设计。如图 3.1 所示。

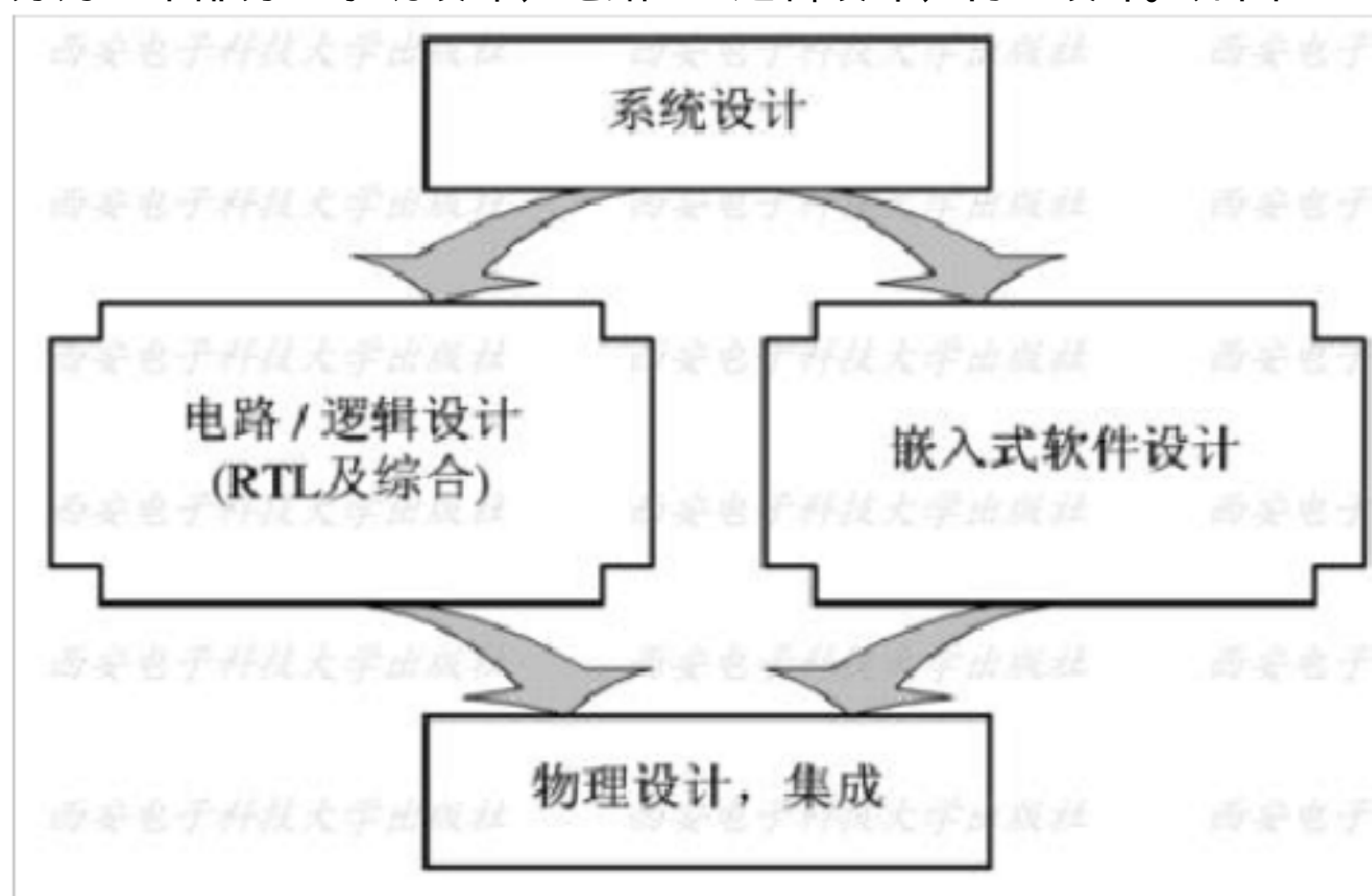


图 3.1 IC 设计的步骤

系统设计主要包括算法设计和架构设计，具体内容如下：

- a. 给出浮点数算法；
- b. 进行软、硬件划分；
- c. 给出 IC 设计中数据的定点表示；
- d. 给出定点算法；
- e. 结构的并行性、流水线设计；
- f. 存储器分配。

电路 / 逻辑设计是将系统设计的算法、架构付诸实现的过程。它分为数字设计与模拟设计两种。对于数字设计，多采用 Verilog/VHDL 语言在寄存器传输级完成设计，由综合工具生成网表；对于模拟设计，一般采用电路图与 SPICE 语言来完成设计。物理设计是 IC 设计的最后一步。物理设计得到的 GDSII 文件被送到工厂进行生产。对于模拟电路和有特殊要求的数字电路，采用定制方法完成物理设计；对于一般的数字设计，采用自动布局布线方法完成物理设计。

3.2.1 系统级设计

IC 系统的架构正向“系统级芯片”发展。以往，系统级设计主要是指板级设计。随着硅工艺从 $0.18\ \mu\text{m}$ 向 $0.13\ \mu\text{m}$ 以及 $90\ \text{nm}$ 工艺转变，单个芯片上可以集成大量的处理器、外围器件和大于 $1\ \text{MB}$ 的 SRAM，在单个芯片上能够实现原来多个硅芯片才能实现的功能。在现在的 IC 设计中，系统级设计占的比重也越来越大。

系统级设计主要包括规的制定、算法设计与架构设计几个方面。创建系统设计规是 IC 系统设计的第一步。规的定义必须严格准确。有些对安全等级要求高的场合，需要用形式化语言来描述规。如果可能，在制定规前要多看一些同类产品的数据单 (datasheet)。算法 (或行为模型) 可以看作一个 IC 系统的可执行规。对于通信、视频这类包含大量数字信号处理的系统，算法设计尤其重要。算法工

工程师通常用 C 语言或者 MATLAB 来研究算法。MATLAB 编程比较方便，但对于大型设计，往往更需要 C 语言较快的运行速度。现在，越来越多的设计者开始采用 SystemC 进行架构设计。SystemC 是一种具有 C++ 语法特征的硬件描述语言。SystemC 的真正优势在于：

- a. 它有很强的系统描述能力。它可以在很高的抽象层次上方便地描述大型系统（虽然 VHDL 也具有系统级的描述能力，但相当有限）。用 SystemC 可以很方便地建立系统的可执行规，从而加快开发进度。
- b. 可以用一种语言来描述软、硬件，有利于设计者比较各种软、硬件划分方案，并在设计早期进行协同验证。
- c. 设计者可以将 SystemC 模型逐个替换为 Verilog/VHDL 的 RTL 实现。新的 RTL 模块可以与其他 SystemC 模型一起编译仿真。

架构设计是系统设计的重要一步。通过性能估算及仿真，设计者可以确定设计中所采用的数据通路结构。例如，设计中需要用多大的乘法器，采用何种滤波器的结构，采用并行方式还是串行方式更合理，是否需要流水线 (pipeline)，数据宽度选择多少。这些都属于架构设计的畴。架构设计与算法设计紧密相联。架构设计需要确定各模块如何划分、模块如何相连、存储器如何分配等问题。在架构设计时，要仔细地设计模块之间的接口，把芯片实现的功能和时序局部化。

这样，每个模块都可以在隔离的状态下独立地进行设计和验证。在设计架构时，需要考虑哪些模块由 IP 完成，哪些由自己来实现。在一个片上系统的设计中，要用到许多 IP。IP 有软核、固核和硬核之分：

- a. 硬核是经流片验证过的版图。它的可重复使用性最低，设计者不能进行任何改动，使用起来相当于库单元。
- b. 软核是可综合的 RTL 模型，可重复使用，灵活性高。
- c. 固核是带有布局规划信息的网表。

硬核的优点在于已经通过了物理设计，可以不加修改地直接使用，时序是收敛的；缺点是可能会使芯片难以布局布线。软核避免了硬核的上述弊端，但存在开发周期长的缺点。固核的易用性与灵活性介于硬核和软核之间。在实际的 IC 系统开发过程中，具体采用何种方式，还需要设计者权衡考虑。

算法架构完成后，需要转换为逻辑 / 电路实现。在许多 IC 系统设计中，这种转换要依靠手工来完成。利用现在的系统级设计工具（如 Synopsys 公司的 SystemCCompiler），可以将 SystemC 描述的系统模型翻译成 RTL 级 Verilog/VHDL 代码，也可以将其直接翻译成门级网表。系统设计对设计者提出了非常高的要求。设计者往往要具备嵌入式处理器、数字信号处理器、存储器、系统总线等各方面的知识，能够平衡系统存储器带宽和 DSP 的处理功能两方面的要求，并且能够与软件开发者协同工作。

在系统设计领域已出现了一些 EDA 工具，如 Synopsys 公司的 CoCentric System Studio、Cadence 公司的 SPW (Signal Processing Workstation)，但此类工具还远未达到人们的期望，因此未得到推广，仅在一些特定的领域（如 GSM 有一些应用）。

3.2.2 电路 / 逻辑设计

电路 / 逻辑设计分为两方面：数字电路设计和模拟电路设计。下面首先介绍数字电路设计。数字电路设计一般从 RTL (寄存器传输级) 开始。设计者根据上一阶段得到的各模块的规，用 Verilog/VHDL 语言完成 RTL 代码。RTL 代码完成

后，可以借助 nLint 等语法检查工具改进代码质量。

接下来进行 RTL仿真，以保证设计功能正确。我们将在后面对仿真进行说明。RTL仿真无误后，用逻辑综合工具将 RTL代码转成门级网表，再将门级网表交由后端进行布局布线。图 3.2 给出了这一流程的示意。虚线以上为逻辑设计，虚线以下是物理设计。

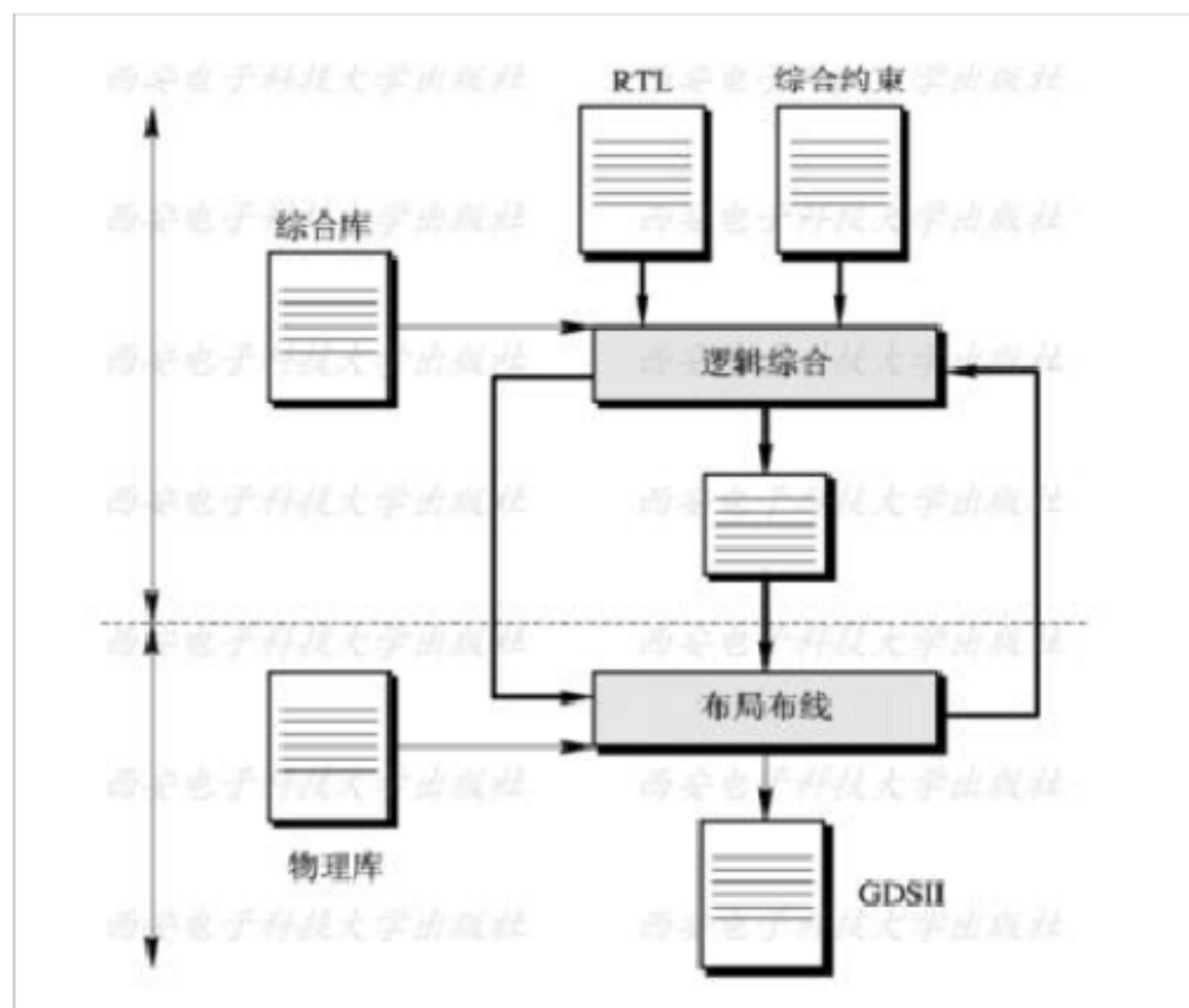


图 3.2 采用逻辑综合的步骤

在一些深亚微米设计中，物理设计与逻辑综合已开始融合，这就是“物理综合”的设计方法。物理综合与逻辑综合的差异在于：在物理综合流程中，可以在芯片规划（Floorplan）之后，采用物理综合工具（如 physical compiler）对网表自动进行反复的综合和布局。由于这时工具能够得到每个单元上较为准确的连线信息，因此可以消除设计不收敛的问题，提高了芯片的性能。图 3.3 给出了物理综合的步骤。

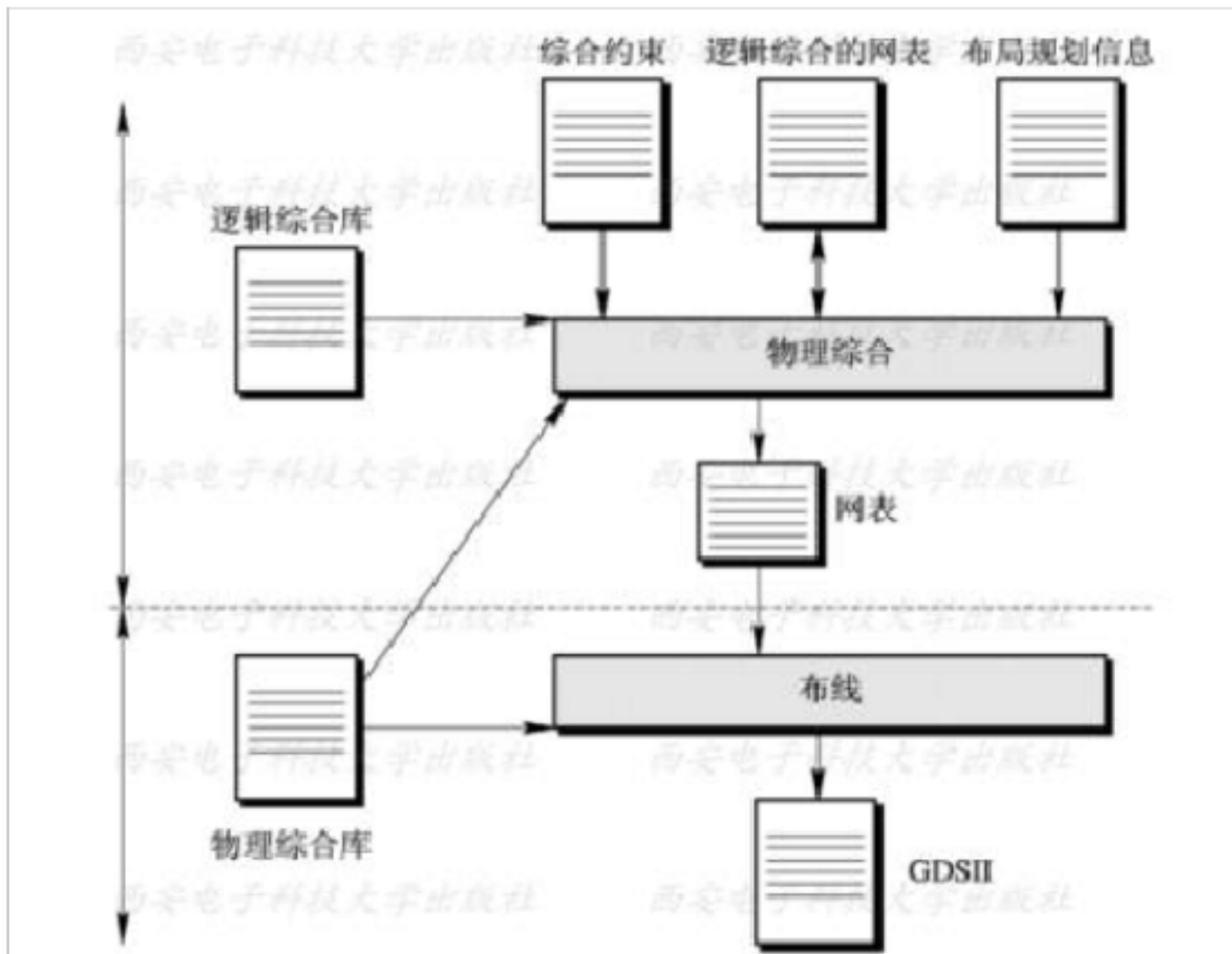


图 3.3 采用物理综合的步骤

在数字电路设计阶段，设计者根据设计规，需要完成如下工作：

- a. 写 RTL代码。
- b. 将 RTL代码综合为门级网表。 如果需要，加入 JTAG扫描链 /clock gating 等容。
- c. 进行时序检查、等价性检查、测试覆盖率估计、功耗分析等。

逻辑设计要求设计者对设计要求、基本电路结构、综合库、 RTL语言、逻辑仿真等非常熟悉，并了解可测性设计、低功耗设计、可复用设计等技术。

模拟电路的设计一般从晶体管级开始， 采用手工设计生成最终网表。 图 3.4 给出了模拟电路的设计流程。

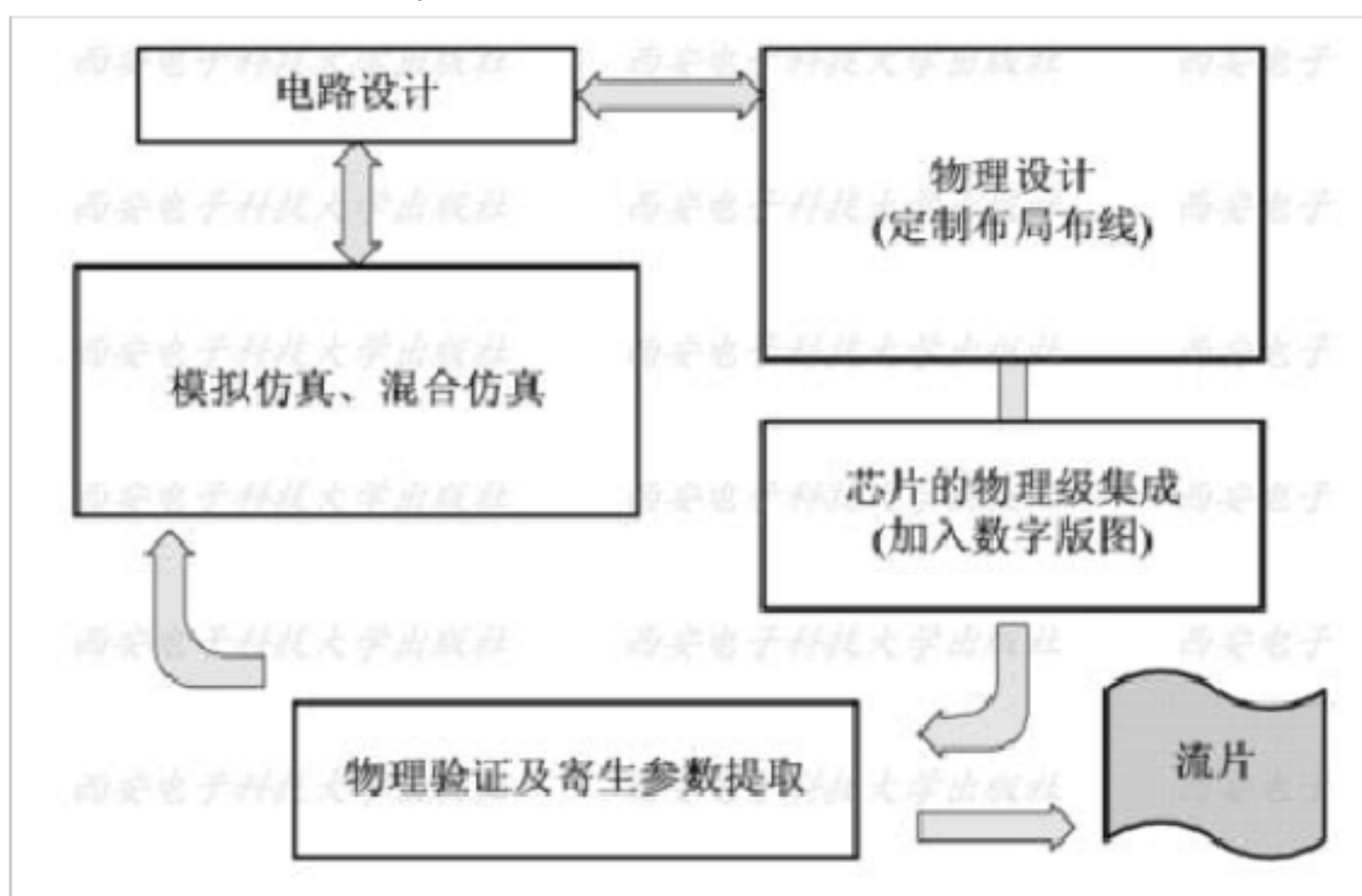


图 3.4 模拟部分的设计流程

实际的 IC 系统往往是数 / 模混合系统。数字 IC 与模拟 IC 单独设计、各自验证的流程导致芯片开发效率降低。目前已出现了数 / 模混合 IC 的设计方法。图 3.5 给出了数 / 模混合仿真的流程。

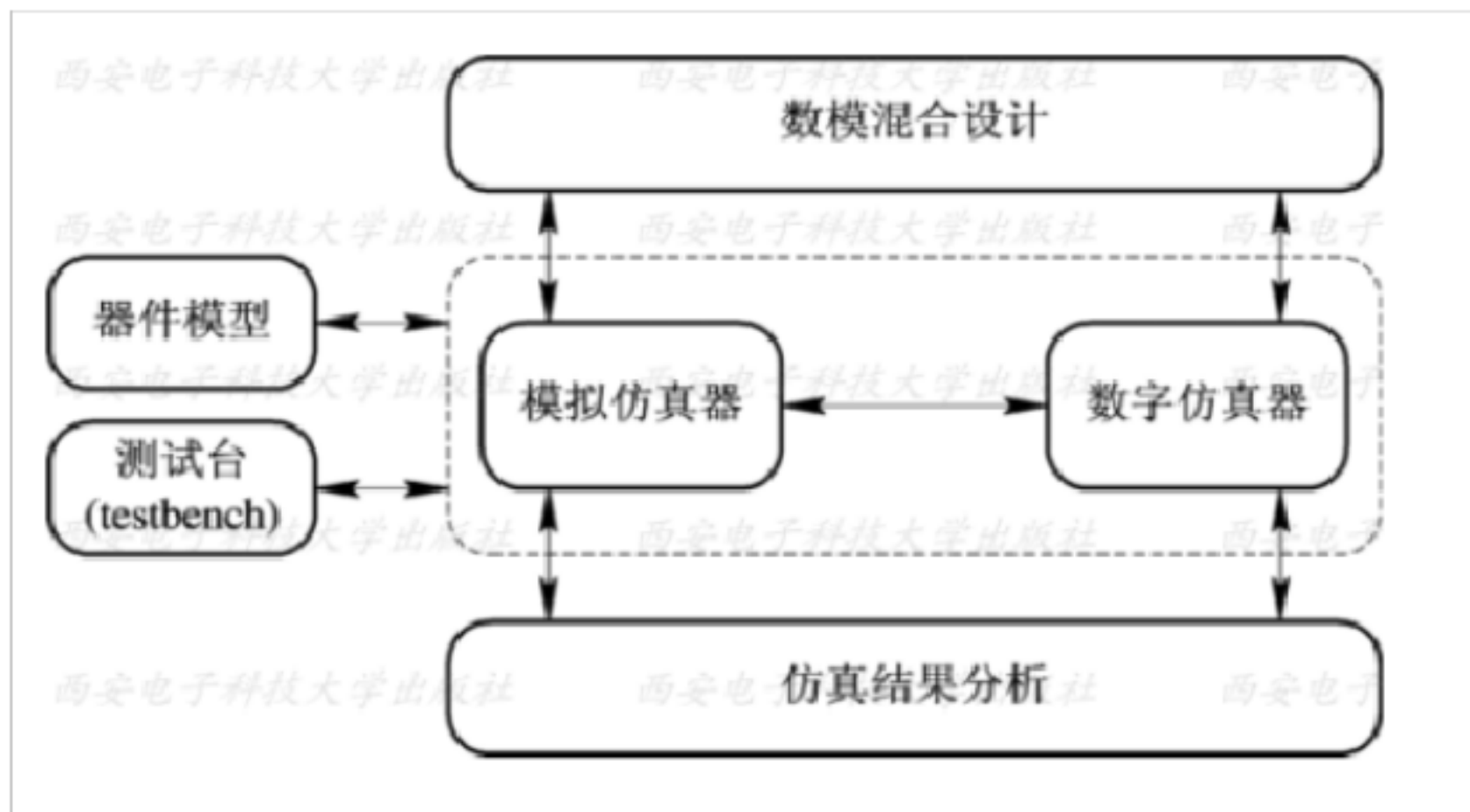


图 3.5 数 / 模混合仿真的流程

3.2.3 物理设计

芯片级的版图设计包括两部分容：全定制电路的版图设计和自动布局布线的版图设计。其中，全定制方法适用于模拟电路和那些对性能面积要求较高的数字部分（例如 CPU 中的关键数据通路），而绝大多数数字设计采用自动布局布线的方法来生成版图。这两部分版图完成后，要在芯片级别整合在一起。图 3.6 给出了自动布局布线（APR）的流程。

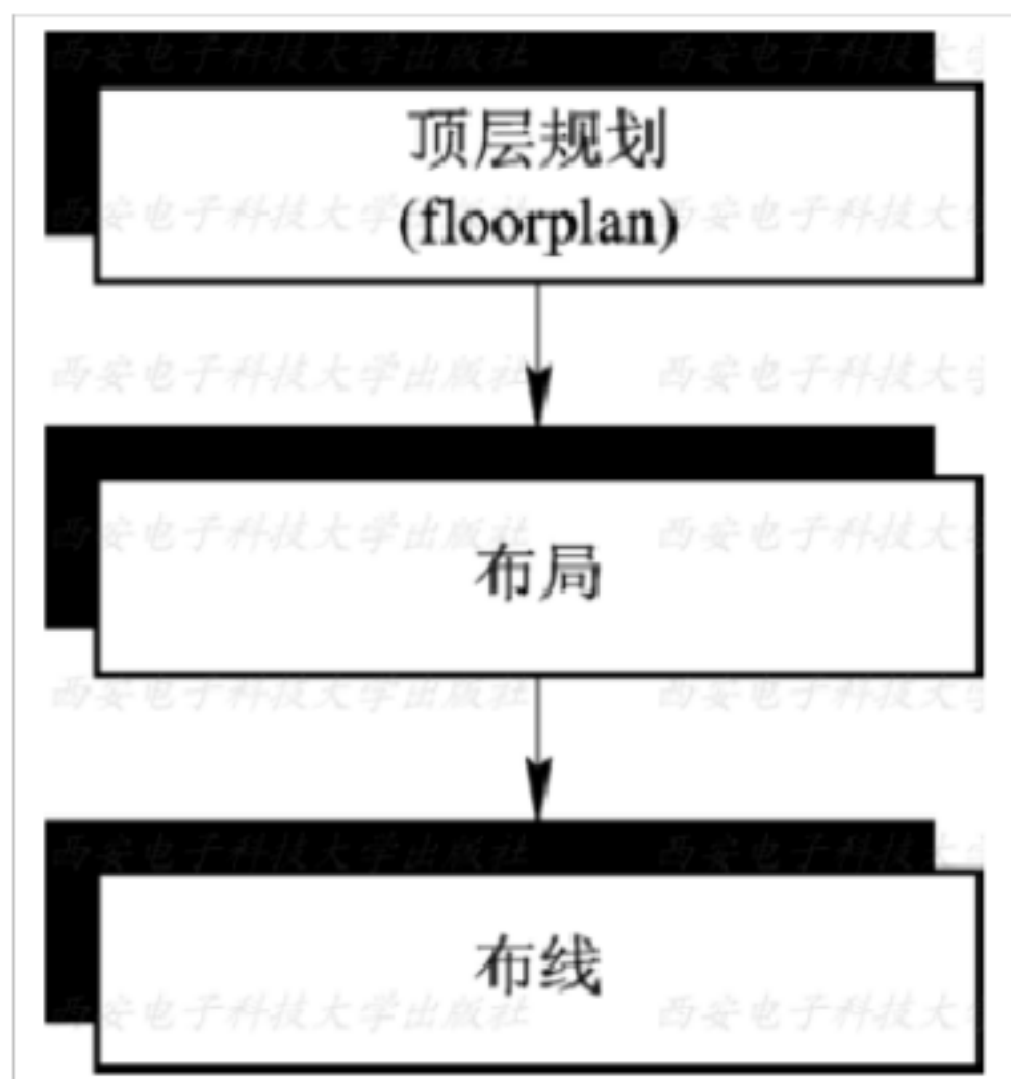


图 3.6 自动布局布线的版图设计流程

1. 顶层规划

在实际的大型芯片设计中，往往很早就确定了芯片中各个块的位置。也就是说，芯片的顶层规划常常在进行电路 / 逻辑设计的时候就开始了。图 3.7 给出了芯片顶层规划的步骤。

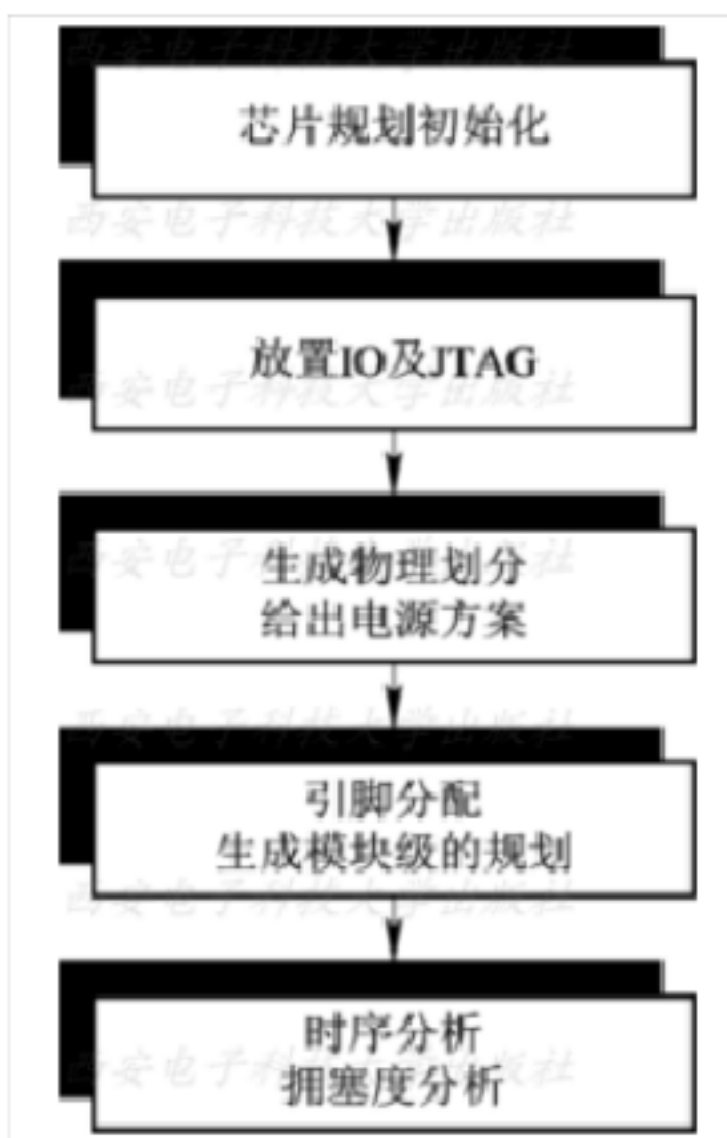


图 3.7 芯片顶层规划的步骤

在进行芯片顶层规划的时候，要考虑如下因素：

- a. 如何使芯片的面积尽可能地小；
- b. 各个宏 (macros) / 块 (block) 如何与 PAD 连接；
- c. 电源方案；
- d. 引脚 (PIN) 方案；
- e. 分析顶层的布线信息，通过合理地放置块，合理地
- f. 定义区域及分组，使得连线的复杂性不会过高。

2. 布局布线

完成芯片顶层规划之后，接下来进行布局布线。通常，这一阶段可分为四个步骤：

- (1) 布局。
- (2) 扫描链的优化。PR 工具重新对扫描链中的单元排序，以便降低布线的复杂性。
- (3) 时钟树的综合及布线。由于时钟是设计中最为重要的信号线，因此在布线时要首先布时钟线。在通常的 PR 工具中，通常将时钟树的生成、时钟的布线作为一个单独的步骤，称为时钟树综合。
- (4) 布线。

图 3.8 给出了布局布线的步骤。

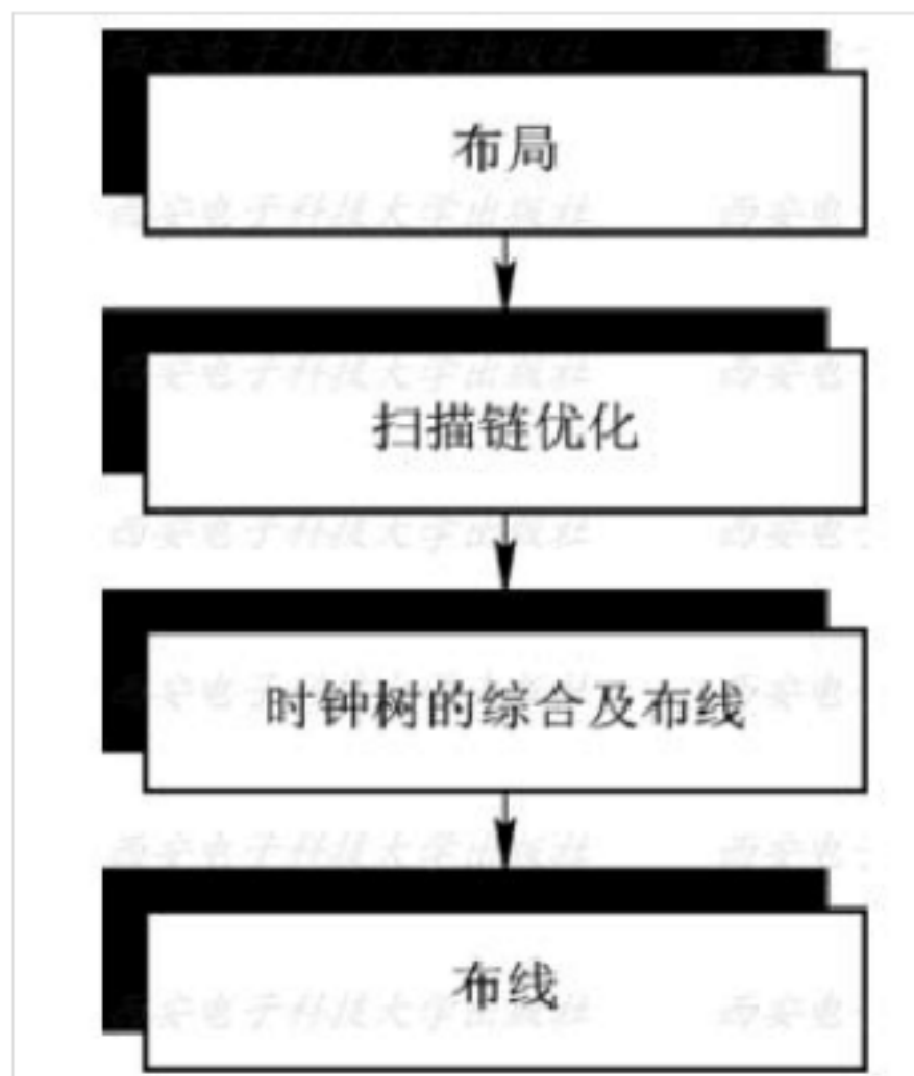


图 3.8 布局布线

3.3 ASIC 系统验证分析概述

究竟要设计什么样的电路？如何保证制造出的电路和原来的设计一致？生产出的产品功耗与速度是否与预期一致？这些都是设计者必须要面对的问题。现在的 IC 系统异常复杂，设计时容易出错，同时应用往往对它的可靠性要求极高。为了验证 IC 系统是否正确，人们提出了种种验证方法，期望借助 EDA 工具的帮助，尽可能地找出错误。

实际上，验证属于设计的畴，但为清楚起见，这里用单独一节来进行介绍。验证贯穿了 IC 系统设计的每个阶段（从算法架构设计直至物理设计）。在系统设计阶段，采用仿真方法进行性能分析和协议分析。在电路 / 逻辑设计阶段，采用软硬件协同验证：用仿真和 FPGA 验证功能；用静态时序分析工具验证时序；用形式验证检查综合过程及 ECO 是否正确。

在物理设计阶段，采用物理验证（LVS、DRC 等）：用静态时序工具验证最终时序；用形式验证检查布局布线过程及 ECO 是否正确。在验证中，错误发现得越早，对开发进度的影响越小，这时验证就越有价值。因此，要尽量在设计的早期阶段（算法架构设计、RTL 设计时）进行充分的验证。常用的验证大致分为如下几类：仿真、静态时序分析、形式验证、物理验证（DRC/LVS）、信号完整性检查、FPGA 验证等，下面分别进行介绍。

3.3.1 仿真

仿真是功能验证的主要手段，其基本原理如图 3.9 所示。

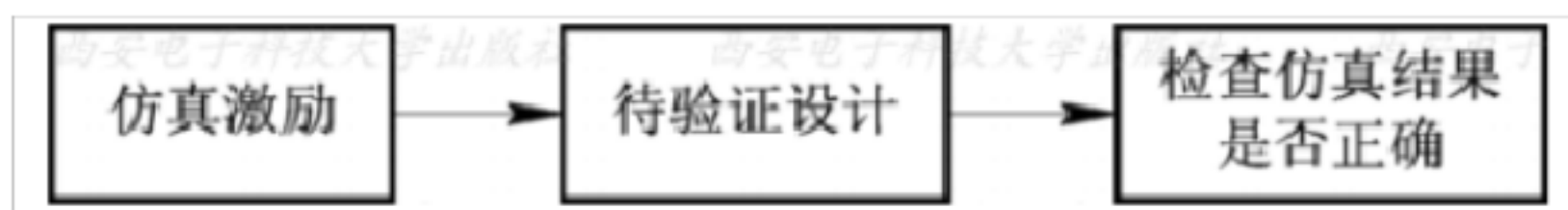


图 3.9 仿真基本原理

除了功能检查，用仿真方法还可以检查系统的时序。与静态时序方法相比，仿真的方法更为直观，能较为真实地模拟出电压发生变化、串扰及毛刺、电容耦合等各种情形下电路的运行情况，其缺点是运行时间较长，且时序检查不完整。仿真贯穿了 IC 系统设计的整个阶段。

1. 系统设计阶段的仿真

在系统级设计阶段，通过行为仿真，可以达到下述目的：

- a. 分析算法是否正确；
- b. 验证性能是否满足要求；
- c. 验证系统划分是否合理。

系统级设计可大致分为浮点数算法设计、定点数算法设计、架构设计等几个步骤。在每一步都可以进行系统仿真，如图 3.10 所示。

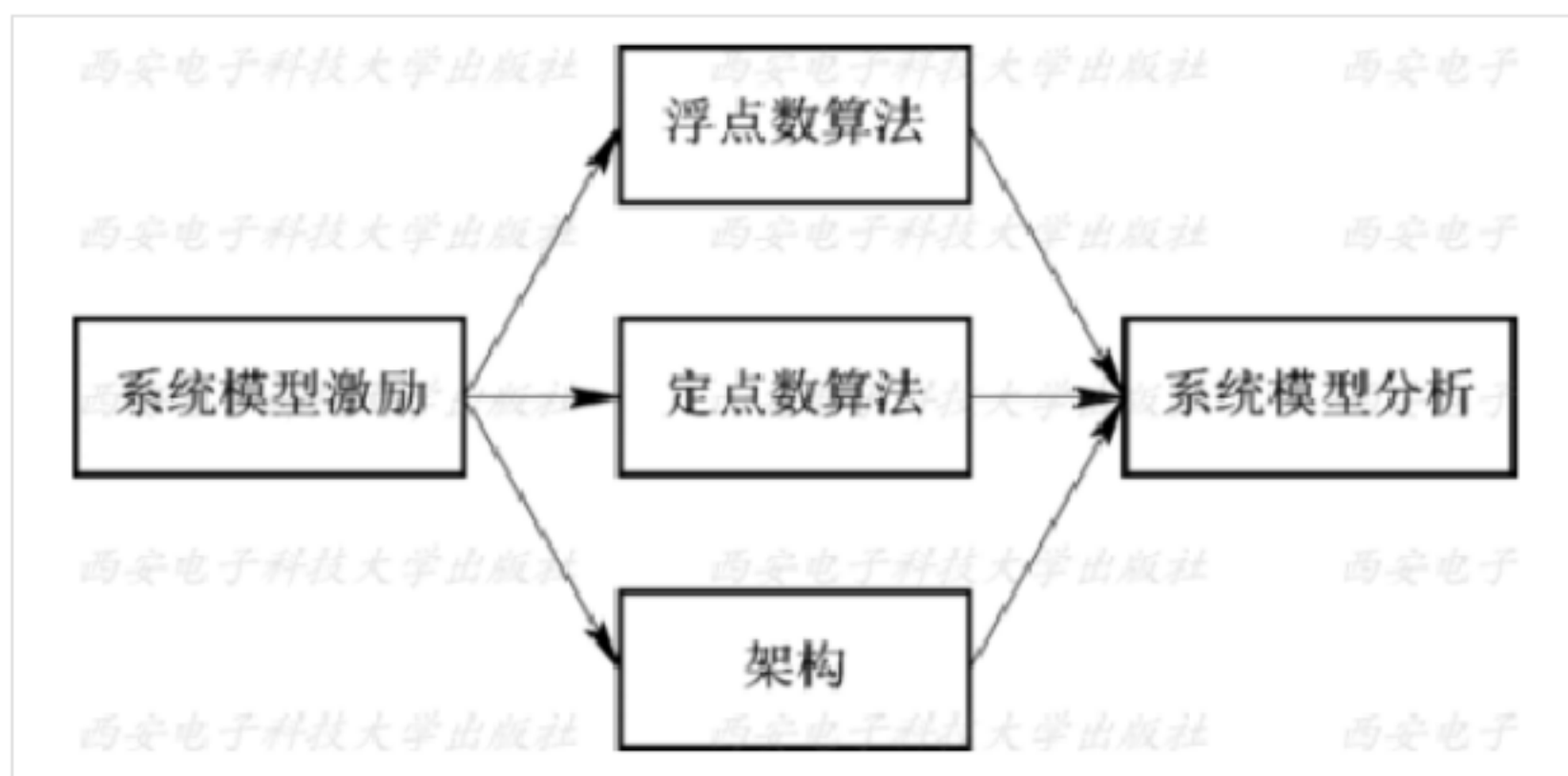


图 3.10 系统级验证

系统级的仿真可以采用数据流仿真方法（不包括时间信息）和基于周期的仿真方法。在系统中，通常用 SystemC/C/MATLAB 来描述系统模型。系统模型可以作为后续阶段验证的参考。例如，在 RTL 设计中，可以利用系统模型的输出结果作为参考基准，检查 RTL 设计得到的结果是否正确。

2. 逻辑/电路设计阶段的仿真

逻辑设计阶段的仿真分 RTL 仿真与门级仿真两种。RTL 仿真是验证系统功能是否正确的重要手段。在 RTL 仿真中，如何合理构造 testbench、如何进行“corner case”的验证，是困扰设计者的难题，也是目前验证的热点。一个测试平台应包括如下几个部分：待测试设计（Design under Verification）、Stimulus（用于生成激励向量）、Monitors（用于监控接口上的活动）、响应比较电路（用于检查设计的输出是否与预期的一致）。

门级仿真速度较慢，通常只是作为一种辅助手段来检查时序是否满足要求，所以通常只需运行很少的几种激励即可。在逻辑设计阶段，目前有许多成熟的仿真工具，如 NC Simulator 及 VCS，它们属于事件驱动（event driven）的仿真工具，既支持 RTL 仿真也支持门级仿真。仿真方法存在测试覆盖率低的缺点。一些仿真工具开发商利用形式验证中“断言（assertion）”的概念，提出了“基于断言的仿真（assertion based simulation）”方法。此外，也有人尝试采用符号仿真（symbol simulation）的方法。不过目前这种方法应用很少。

对于模拟电路，通常采用 Hspice 进行仿真。对于特定应用的电路，有一些

特定的仿真工具，例如在分析射频（RF）电路时，Cadence的 Spectre 比较实用。在数/模混合设计中，往往需要协同验证数字电路和模拟电路。这时候，需要用到混合仿真软件。目前，混合仿真软件是将逻辑仿真器与电路仿真器结合起来协同工作。模拟部分通常用 verilog A 来建模。图 3.11 所示为 Synopsys 的混合仿真环境。其中，数字部分采用数字仿真器 VCS,模拟部分采用模拟仿真器 Nanosim；数字部分用 Verilog 描述，模拟部分用 SPICE及 Verilog A 描述。

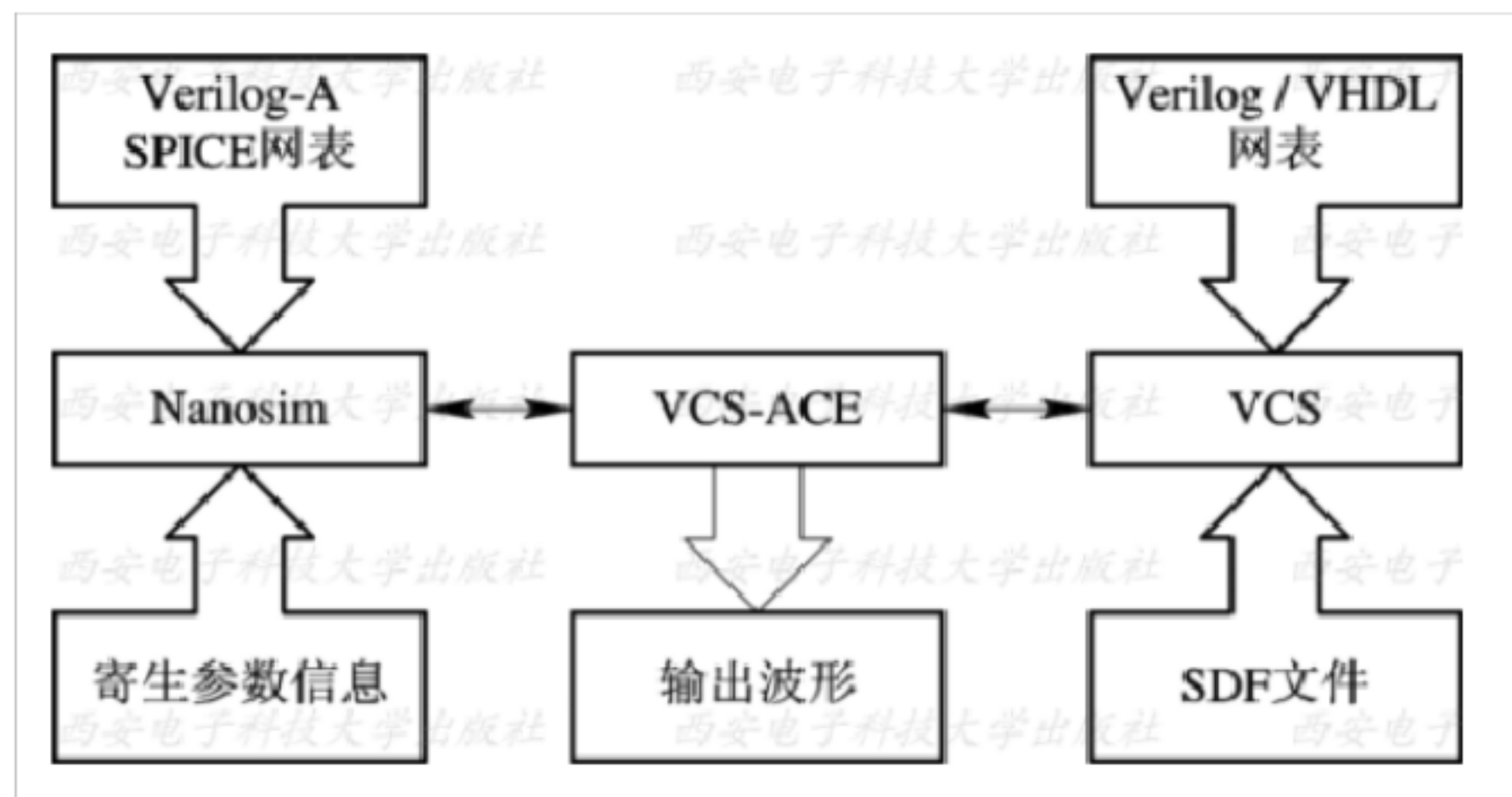


图 3.11 Synopsys 的混合仿真环境

3.3.2 静态时序分析

静态时序分析是检查 IC 系统时序是否满足要求的主要手段。静态时序分析工具根据网表中各节点的负载，利用综合库中各单元的延时查找表，计算出单元延时及连线延时，从而可以判断在该设计中寄存器之间最长路径上的延时有多少，是否满足最大时钟约束。如果不满足，则静态时序分析工具会给出 setup 违例。

静态时序工具还会分析 hold 违例。Hold 违例跟时钟树有关。在深亚微米设计中，串扰（crosstalk）、IR drop 等都会影响芯片的时序乃至功能。这些属于信号完整性的问题。OCV（OnChip Variation，片上偏差）效应也越来越显著。在现在的静态时序工具中，支持对这些现象的分析。静态时序分析可分为两类：逻辑级分析和电路级分析。前者针对用标准单元构成的网表，后者主要针对模拟电路和混合电路。

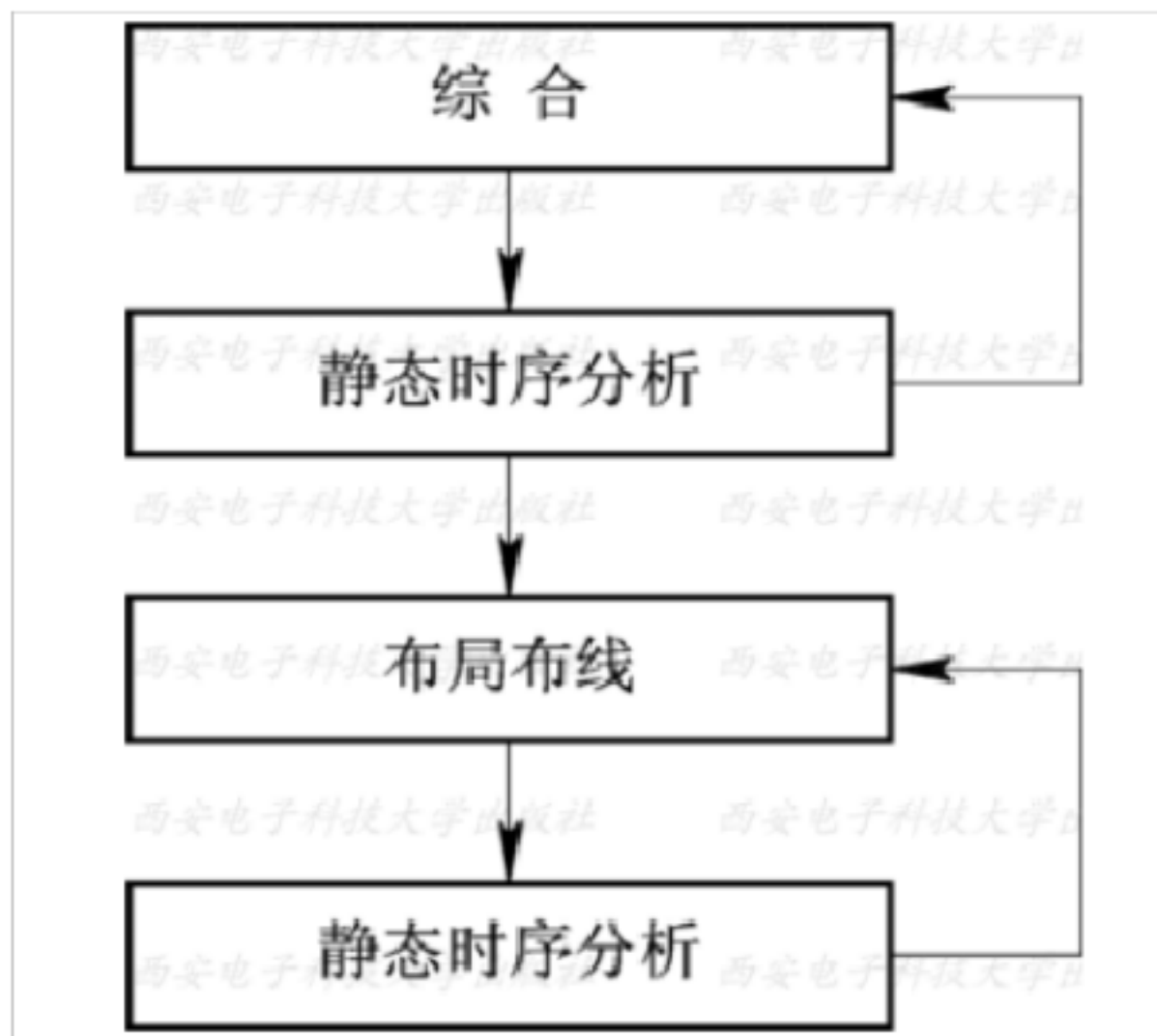


图 3.12 逻辑级静态时序分析

在进行逻辑级静态时序分析时，需要输入如下内容：网表、综合库、时序约束（时钟、输入延时、输出延时等）。如果是对布局布线之后的网表进行静态时序分析，还需要输入寄生参数文件（该文件包含了由版图提取的网表中各节点的寄生电容和寄生电阻等信息）。

图 3.12 给出了逻辑级静态时序分析的步骤。通常来说，在网表综合完成后，需进行静态时序分析，在布局布线完成后，再进行一次静态时序分析。当然，在布局布线尚未最终完成时，也可以进行静态时序分析。电路级的静态时序分析工具（如 PathMill）可以分析一个芯片的 I/O 是否满足时序要求。

3.3.3 功耗分析

一个芯片的功耗过大，会导致如下问题：

- a. 使芯片性能变差。
- b. 使温度升高，降低芯片可靠性。
- c. 限制便携式产品的使用时间及电池寿命。

因此，对于高性能设计或便携式应用，必须采用低功耗芯片，如奔腾芯片、数字信号处理器、手机芯片等。

为了降低芯片的功耗，必须采用低功耗设计技术。低功耗设计可以从系统级、逻辑电路级、物理级着手。在系统级，进行软、硬件划分时，要考虑哪种划分方案会得到更少的功耗；要选择功耗最小的算法；可以考虑采用并行运算、流水线等手段，降低功耗；可以采用多电压设计方案来降低功耗；可以设置省电模式，以便在系统不工作时降低功耗。

在 RTL 设计级，可以考虑采用时钟门控、操作数隔离（operation isolation）等技术来降低功耗。在逻辑综合时，可以考虑采用插缓冲、相位分配（phase assignment）等技术来降低功耗。在进行布局布线时，也要考虑到功耗的影响。可以将翻转率高的节点用寄生电容较小的金属层来布线，以减少整体功耗。此外，还可以考虑采用新的材料、新的封装技术来降低功耗，以消除功耗的影响。

芯片速度的提高，工艺水平的进步，便携式应用的增加，封装技术发展的相对迟缓，这些因素使低功耗设计在 IC 设计中越来越重要。低功耗设计技术可能

会导致速度变慢、芯片面积增加、设计周期延长，但这总比需要依靠风扇、封装甚至液氮来降温要好。在进行低功耗设计前，首先要进行功耗分析，了解设计中哪一部分消耗功耗最大，何种功耗所占比例最大，接下来才能有的放矢，研究降低功耗的设计方法。

在实际应用中，通常在门级和晶体管级进行功耗分析。有些工具也支持 RTL 级的功耗分析，以帮助设计者尽早了解设计中功耗的情况。而系统级的低功耗分析工具还很不成熟。利用功耗分析工具可以得到峰值功耗与平均功耗。影响温度的是平均功耗。下面列出 Synopsys 与功耗分析有关的工具。

- a. Power compiler：对动态功耗和泄漏功耗进行优化；
- b. Primepower：对峰值功耗和平均功耗进行分析；
- c. JupiterXT：生成电源网格；电源网络分析；
- d. Astro Rail：电压降及电子迁移分析。

3.3.4 形式验证

用仿真的方法验证系统功能有两个缺点：一是很难对一些隐蔽错误进行定位，二是要耗费大量的仿真时间。随着系统规模的增大，这些缺陷越来越令人无法容忍。形式验证是最有希望解决此问题的途径。形式验证大致分为模型检查（Model Checking）、定理证明（Theory Prover）和等价性检查三类。

模型检查用时态逻辑来描述规（Specification），通过有效的搜索方法来检查给定的系统是否规。模型检查是目前研究的热点，但其验证的电路规模受限制这一问题还没有得到很好的解决。定理证明把系统与规都表示成数学逻辑公式，从公理出发寻求描述。定理证明验证的电路模型不受限制，但需要使用者的人工干预及较多的背景知识。等价性检查的验证用于验证 RTL 设计与门级网表、门级网表与门级网表是否一致。在进行扫描链重排、时钟树综合等过程中，都可以用等价性检查保证网表的一致性。等价性检查已融入 IC 标准设计流程中，如图 3.13 所示。

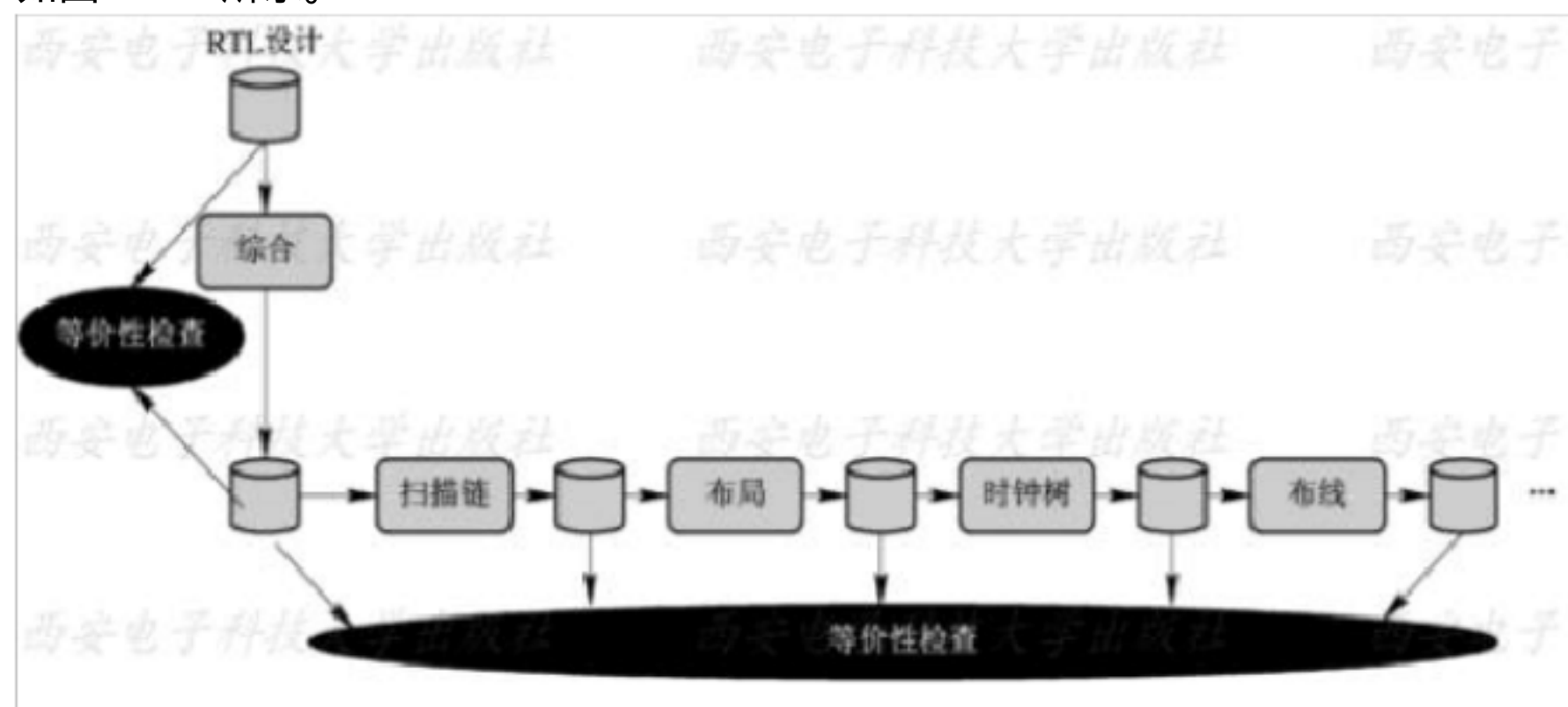


图 3.13 IC 设计中的等价性检查

等价性检查在检查 ECO 时非常有用。例如，设计者在修改门级网表时，由于手误，错将一个或门写成或非门，等价性工具通过比较 RTL 设计与门级网表，可以很容易发现这种错误。

与前两种形式方法相比，等价性检查功能最弱，自动化程度最高。目前商业化的形式验证工具都属于这一类，例如 Cadence 公司的 Verplex 与 Synopsys 公司的 Formality 等。

3.3.5 物理验证 (DRC/LVS)

在物理阶段，要完成的检查包括：串扰分析、电源网格分析、DRC LVS ANT，如图 3.14 所示。

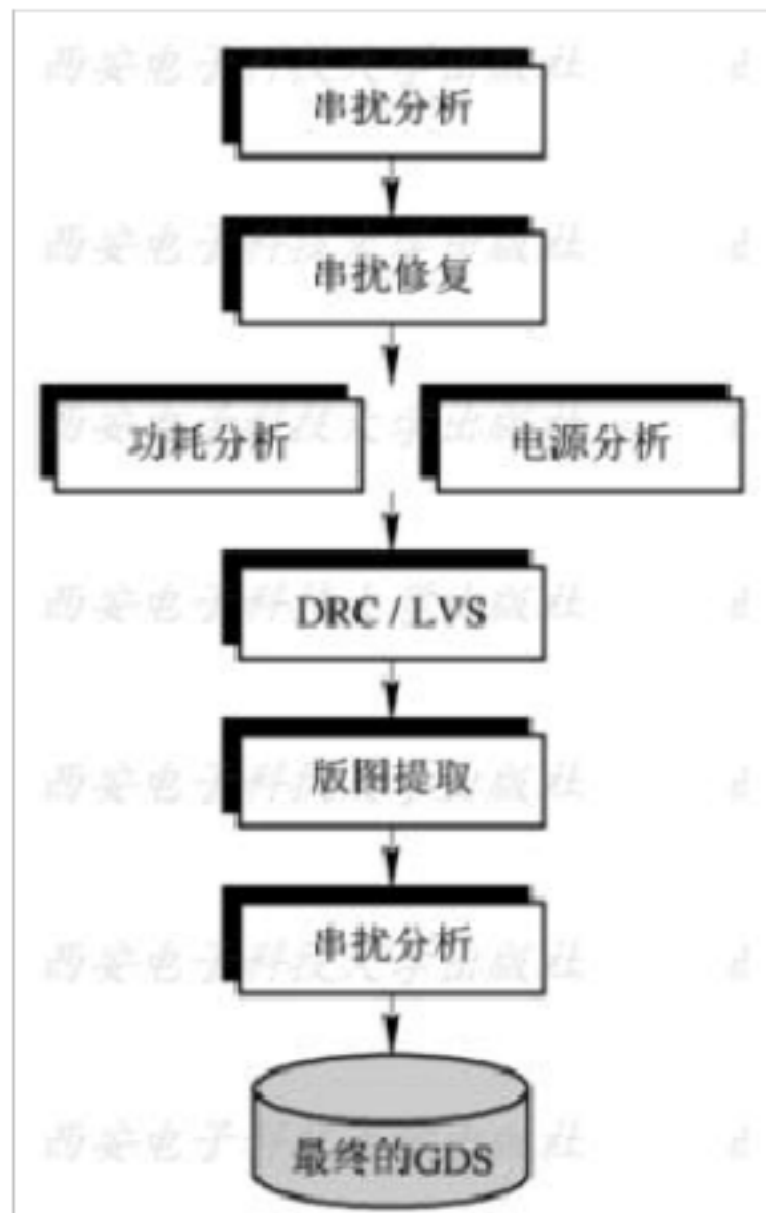


图 3.14 物理设计阶段的验证

串扰分析与电源网格分析属于信号完整性的容，我们将在下节进行介绍。这里对 DRC LVS 进行概述。DRC 检查所设计的版图是否符合工厂给出的版图设计规则。版图级的验证工作 (LVS) 检查版图级的实现是否与门级网表一致。通过 LVS 可以发现版图工具的错误或者是手工对版图进行修改导致的错误。

3.3.6 信号完整性分析

以前信号完整性问题仅出现在高速电路板设计中，现在也开始出现在 IC 系统中。随着器件的特征尺寸越来越小，器件的供电电压、噪声容限均开始下降，而耦合电容增加。这些导致了在 IC 系统中会出现信号完整性问题，例如图 3.15 所示。

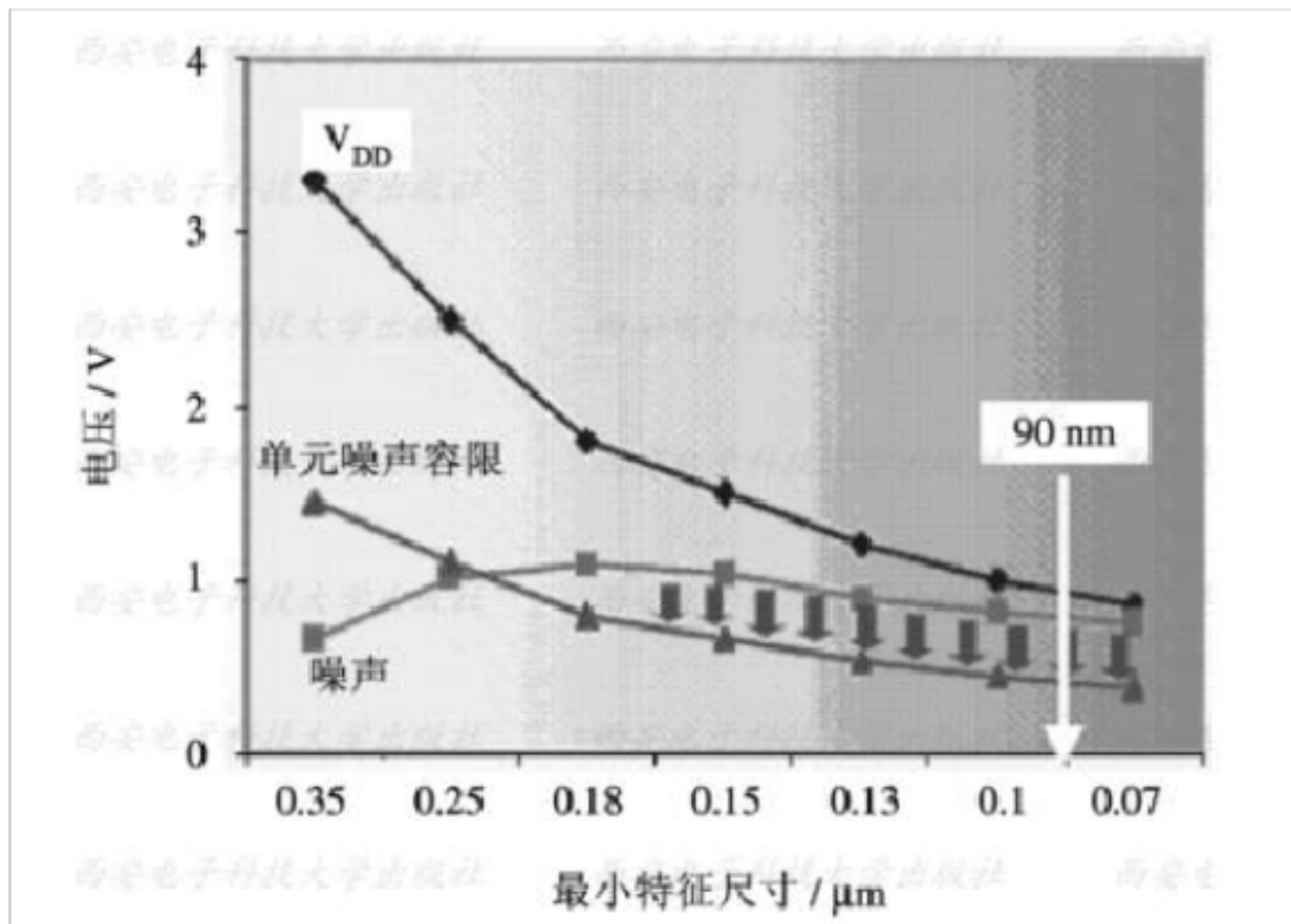


图 3.15 最新工艺对信号完整性的影响

影响信号完整性的因素主要有串扰 (cross talk) 和电源上的电压降 (IR drop)。

其中串扰主要是由连线的耦合电容引起的，电源欧姆电压降主要是由连线的电阻效应引起的。串扰会影响芯片的时序和功能。图 3.16 给出了串扰的示例。

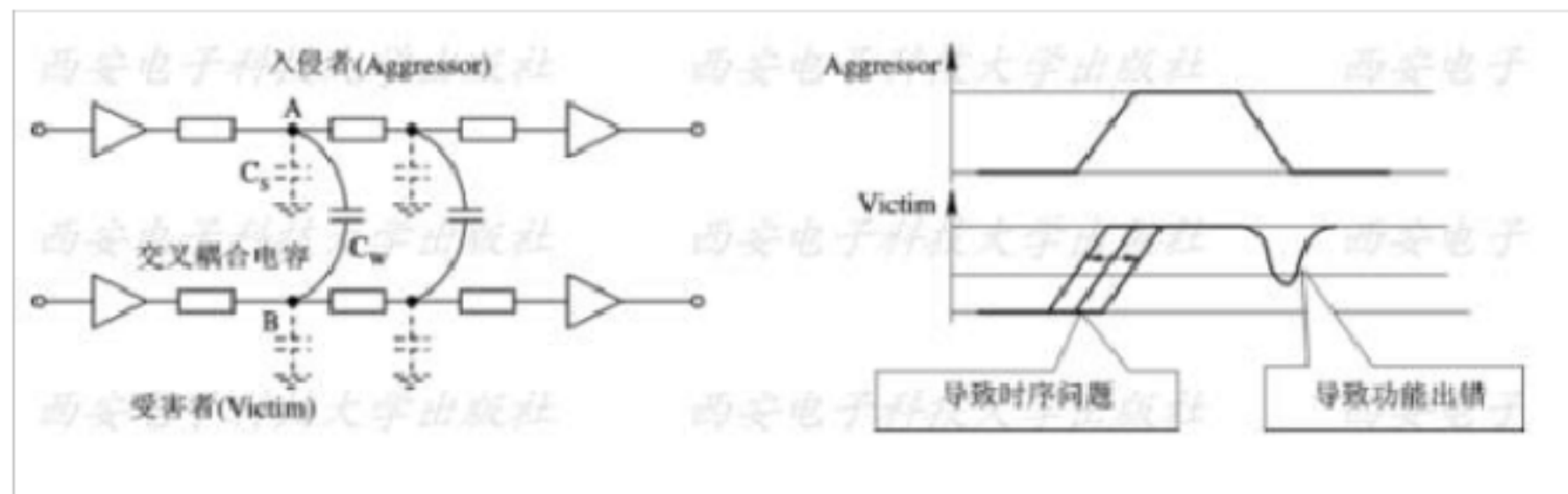


图 3.16 串扰对时序和功能的影响

在图 3.16 中，A 节点与 B 节点之间有耦合电容 C_w 。当 A 节点电压由低电平变到高电平时，或者从高电平变到低电平时，这种变化会通过耦合电容影响到 B 节点。这时候，称 A 为 Aggressor (入侵者)，称 B 为 Victim (受害者)。串扰会导致 Victim 的延时增加或降低，从而引起建立时间违例或保持时间违例。

Aggressor 上的跳变还会引起 Victim 上电压的跳变，导致功能出错。串扰的分析与纠正必须借助串扰分析工具、寄生参数提取工具与物理设计工具，如图 3.17 所示。分析串扰时，必须要有噪声库 (noise library)。

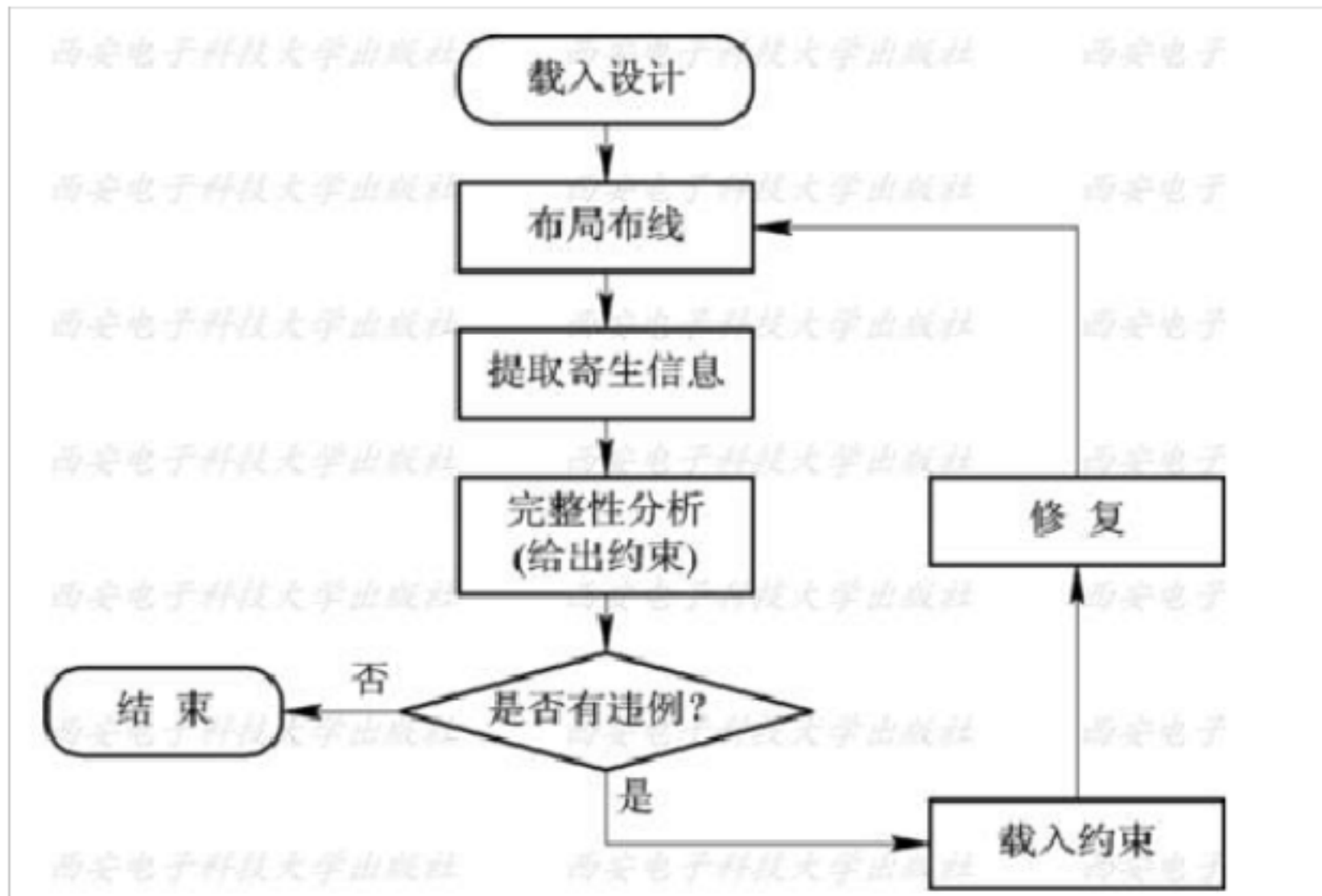


图 3.17 串扰分析与纠正

众所周知，电流流经一个有电阻的导线时会导致欧姆电压降。电源网格（VDD 与 GND）上的电压降会降低芯片单元的供电电压，会影响芯片时序甚至导致功能出错。图 3.18 所示为 IR drop 效应示例。

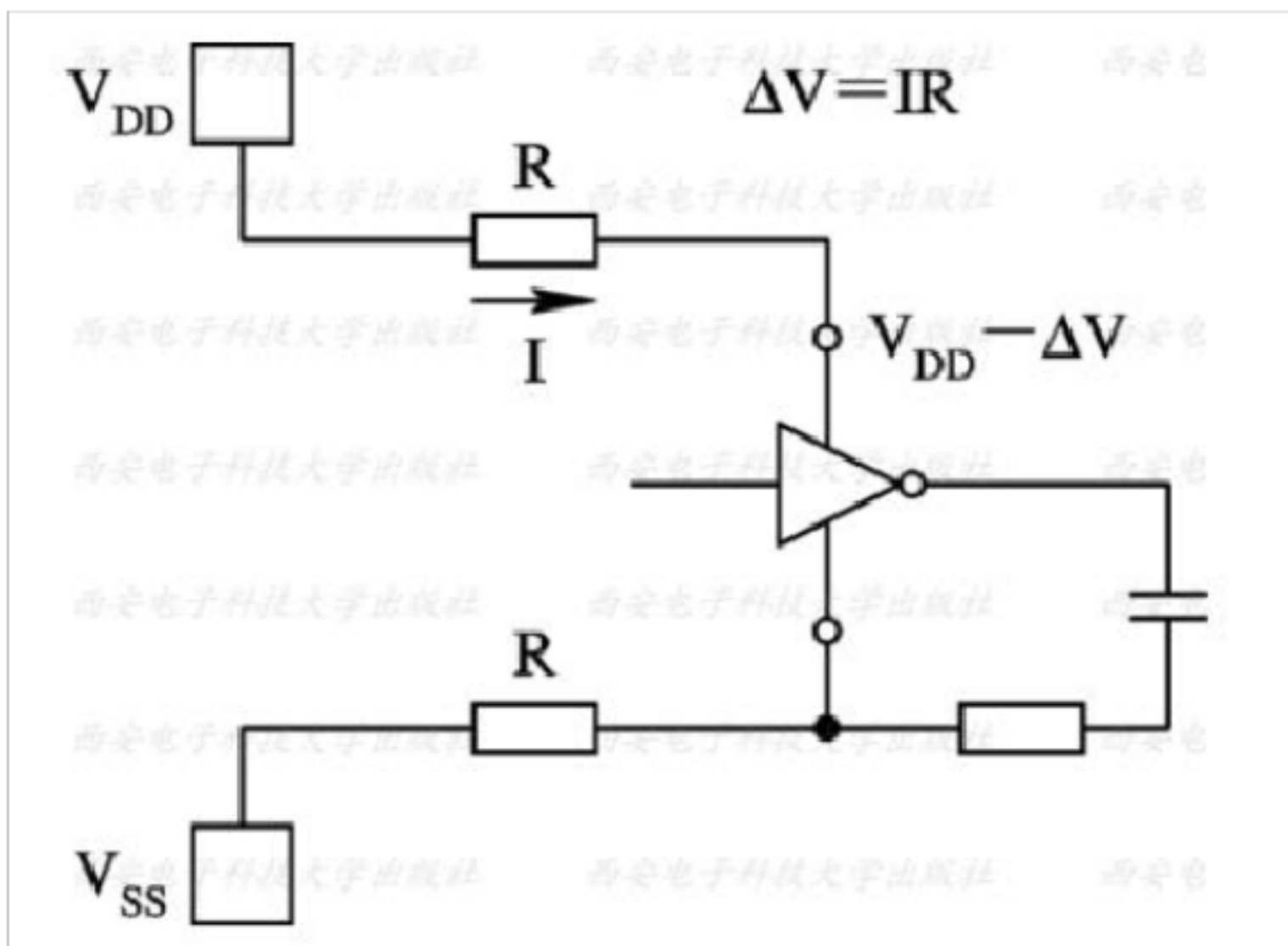


图 3.18 IR drop 效应示例

分析 IR drop 时，可以将电源网格看作是由电阻（导线）、电流源（逻辑）构成的网格。由于在芯片中各单元的峰值电流是随时间变化的，因此 IR drop 是一种动态现象。影响 IR drop 的主要因素是一些切换事件，如时钟上的切换事件，总线驱动器上的切换事件等。目前已有一些电源网格分析工具，例如 Cadence Power 和 RailMill。这些工具可以对芯片中的电流进行动态分析，并

能够对电源网格进行模拟。

3.3.7 基于 FPGA 的验证

FPGA 已成为设计者验证设计的得力助手。与仿真相比，用 FPGA 检查设计的效率要高出许多。在实际芯片开发中，经常将 FPGA 原型机验证与 RTL 仿真结合起来。今天 FPGA 的性能已能达到 500 MHz，而大多数 ASIC 的性能都小于这个数字。也就是说，大多数产品都可以用 FPGA 进行验证。特别是对于 100~200 MHz 这个范围的 IC，用 FPGA 验证非常合适。

在验证时，可以将待测信号引到 FPGA 的引脚，采用示波器实时观测信号波形。此外，现在的 FPGA 实现工具一般都提供了通过计算机观察 FPGA 内部信号的功能，其原理是：通过 JTAG 口，将感兴趣的信号的波形实时传送到计算机上。例如 Xilinx 的 chipscope 就具有此类功能。在引脚比较缺乏的场合，这种方法是非常有用的。用 FPGA 进行验证的大致步骤如下：

- (1) 新建工程文件，读入设计文件。
- (2) 建立引脚定义文件及时钟定义文件。
- (3) 给出综合约束和布局布线约束。
- (4) 综合，布局布线。
- (5) 生成下载文件，并将其下载到 FPGA 中。
- (6) 进行调试。

常用的 FPGA 综合工具是 Synplicity 公司的 Synplify。此外，Xilinx 与 Altera 也提供了自己的综合和布局布线工具。

3.3.8 测试

测试包括如下几个方面的容：

(1) 在芯片量产之前，验证设计是否正确和符合规范，主要进行功能测试和电气特性测试。功能测试主要是测试输入和响应的一致性。电气特性测试可分为直流特性测试与交流特性测试两种。直流特性测试包括短路、开路、最大电流、漏电流、输出驱动电流、开启电平等测试。交流特性测试主要包括传输延时、建立和保持时间、速度、访问时间等测试。

(2) 量产测试。在量产时，每个芯片均需经此测试，以剔除存在制造缺陷的产品。在进行量产测试时，要考虑到成本因素，因为每个芯片都要经过测试，所以要求测试时间短，只做 go/no go 测试，不做错误诊断。进行量产测试一般采用基于扫描链的方式，也可以采用建自测试 (BIST, build in self test) 的方法。进行量产测试，要求在设计中包含相应的结构，即通常所说的可测性设计。图 3.19 给出了可测性设计的示例。

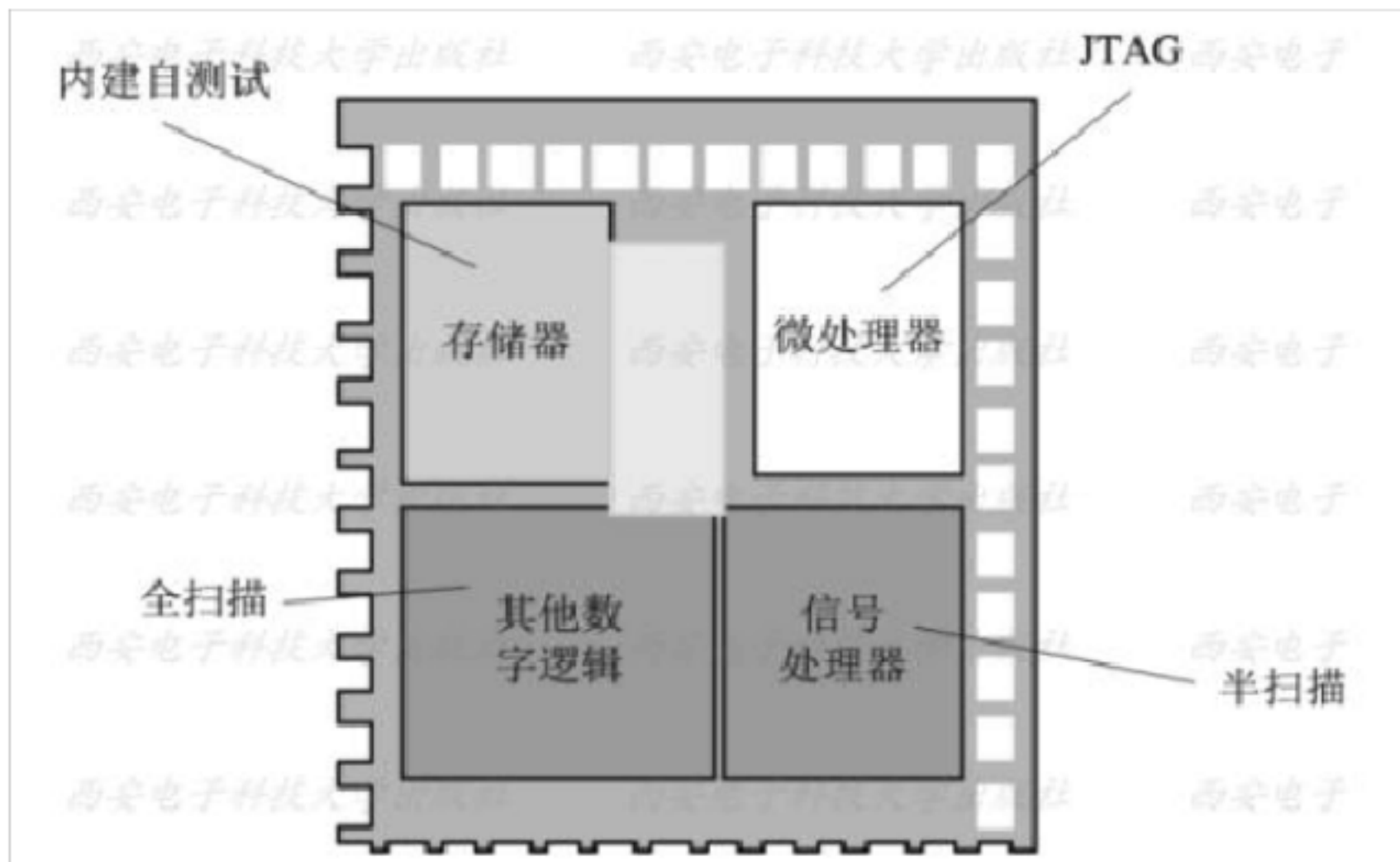


图 3.19 可测性设计示例

在这个例子中，包括存储器、微处理器、信号处理器和其他数字逻辑。其中，存储器采用建自测试方式；微处理器可采用 JTAG 方式进行测试；数字信号处理器可以采用半扫描方式进行测试；其他数字部分可以采用全扫描方式进行测试。

(3) 老化 (Burn in) 测试。老化测试用于测试可靠性 (reliability)，采用各种加速因子来模拟器件长期的失效模型。常用的有加高温，加高出其额定电压的电压等。

四、ASIC 设计的现状及国的研究水平

本章介绍 ASIC 设计的现状及其或的研究水平。

4.1 ASIC 设计现状

ASIC 系统设计现状可以归纳如下：

- a. 频率向 GHz 发展。
- b. 设计的复杂性越来越高，许多芯片都超过百万门，千万门的设计也已出现。
- c. 大都采用同步设计方法，异步设计很少。
- d. 基于 IP 的设计技术越来越成熟，系统级的芯片设计越来越倾向于采用 IP 集成这种方式。
- e. 片上系统往往采用 ARM 或 MIPS 嵌入式处理器，采用标准的片上总线。
- f. 片上存储器所占比例越来越大。
- g. 系统级设计逐渐受到重视，但这方面工具还不完善；行为级的综合并未获得商业上的成功。
- h. 验证在开发中占的比重越来越大，形式验证受到重视。
- i. 连线已成为影响 IC 系统时序的重要因素。
- j. 功耗已成为摩尔定律能否继续适用的主要障碍，低功耗设计越来越重要，漏电流引起的功耗占的比重越来越大。在一个设计中采用多种电压和多种阈值器件，可极大地减少功耗，目前这一方法是低功耗综合的研究热点。

- k. 信号完整性效应开始出现。在静态时序分析中，必须加入串扰的影响。在较长的总线之间，电容耦合现象会进一步增加连线延时。
- l. OCV效应的出现，导致在同一个晶圆上制造的不同芯片性能有差异。
- m. 数字部分在整个 IC 系统中所占比重越来越大。

4.2 国内 ASIC 设计的研究水平

改革开放以来我国经过多年的发展，ASIC 设计方面有了很大的进步，但与世界领先水平相比，差距仍然很大。总体上，目前的研究水平如下：

1、IC 设计水平接近实际领先水平

从 2000 年开始，我国涌现出了大量的 IC 设计单位，到 2007 年已达到 491 家，其中不乏优秀的设计单位，如中星微、炬力、展讯、大唐微电子、海思半导体等。这些优秀的 IC 设计公司的部分成果不仅达到了世界的领先水平，还占据了相应领域产品市场的绝大份额。尤其是计算所龙芯的设计成功，不仅鼓舞了中国人，尤其是 IC 设计人员的志气，还说明了我国的 IC 设计水平已经接近世界领先水平。

2、EDA 设计工具落后

虽然我国的设计水平已经相当出色，但我国 EDA 工具的发展却不尽人意。国内最优秀的 EDA 工具——华大公司的熊猫 EDA 系统还不完善，目前还没有完整的验证工具。此外，它和国外的 EDA 大公司 Synopsys、Cadence、Mentor 的产品比起来，还有一定得差距。因此国产的 EDA 工具在国内的市场份额很小，国内的设计单为几乎都采用国外的 EDA 工具。

五、ASIC 设计的发展趋势

近年来随着半导体制造工艺的不断提高，ASIC 设计的难度原来越大，主要表现在：

1. 元件模型变化：能精确地描述深亚微米工艺的物理特性和电学特性的短沟道器件模型，充分考虑工艺、电压和温度；
2. 电路元件延迟减小，互连线延迟增大 (50~70%, 0.35um)，必须考虑互连线的分布电阻和分布电容，建立精确的互连线延迟模型；
3. 串扰和噪声：互连线越来越细长，间距越来越小，连线的分布电阻和分布电容变大，当工作频率较高时易产生串扰和噪声。版图设计时避免平行线，缩短连线；
4. 时钟线和电源线的影响：互连线延迟会造成时序问题；电源线因连线阻抗造成压降，使得达到各节点的电压变化，需要合理地分布时钟线和电源线；
5. 功耗和散热问题：SOC 可包含一亿个晶体管，散热成为一个问题。不仅移动通讯、便携电脑，而且所有电子设备低功耗永远是一个努力的目标；
6. 铝线的电迁移造成连线断裂：连线变细使得通过连线的电流增大，时间过长会使铝线断裂，造成失效。对流过大大电流的连线要加宽；
7. 热载流子对 ASIC 可靠性的影响：器件尺寸减小，也会造成纵向电流强度增大。会引起热载流子效应，造成集成电路失效；
8. 逻辑与物理的反复设计问题 (0.8um—1 次，0.5um—5 次，0.35um—>10 次)。

同时 EDA 设计工具与设计方法也面临很大的挑战：

1. 相对独立于工艺的高层次综合技术将不再适用，前端设计与后端设计必须更密切地配合；
2. 综合、布局、布线等功能将集成为一体；同时，高层抽象描述语言越来越重要，“更高的抽象级语言”是较 HDL语言更高层次的设计描述。而在高级语言的发展中，C/C++以及 VHDL成为 EDA业界关心的新话题；
3. Single Pass, Physical Synthesis, Floorplanning, 模拟技术, 性能驱动的布局布线技术, 基于 IP 核的 SOQ设计可重用方法；

因此，EDA设计必然朝着以下几个方面发展：

1. 更有效的设计方法、设计流程和 EDA工具；
2. 因设计而异，找出合适的实现方式；
3. 更高层次的设计综合和验证手段；
4. 多方面考虑低功耗设计；
5. 前端、后端更多交流与沟通，减少迭代次数；
6. 可测性设计需要重新考虑；

参考资料

- [1] Jan M. RaBaey, Anantha Chandrakasan and Borivoje Nikolic 著, 周润德等译, 数字集成电路——电路、系统与设计 (第二版), 北京: 电子工业, 2004.
- [2] James R. Armstrong, F. Gail Gray 著, VHDL Design—Representation and Synthesis (第二版), 北京: 机械工业, 2003年3月.
- [3] 曙光等编著, 可编程逻辑器件原理、开发与应用. : 电子科技大学, 2000年8月.
- [4] 潘松、黄继业编著, EDA技术使用教程 (第二版). 北京: 科学, 2005年2月.
- [5] 百度百科, <http://baike.baidu/>
- [6] 维基百科, zh.wikipedia.org