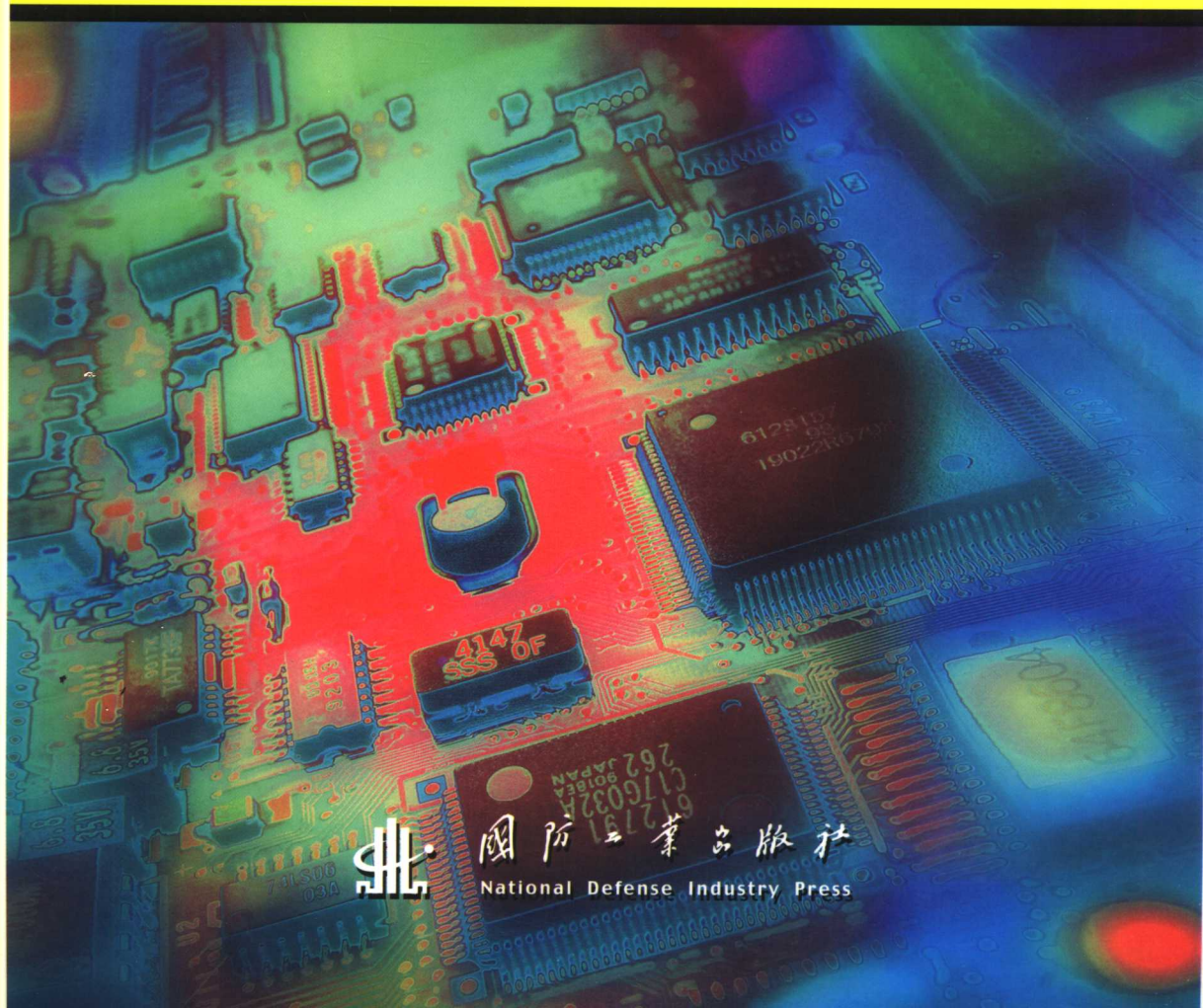


# 数字电路的仿真 与验证

潘中良 编著



国防工业出版社

National Defense Industry Press

# 数字电路的仿真与验证

潘中良 编著

国防工业出版社

·北京·

图书在版编目(CIP)数据

数字电路的仿真与验证/潘中良编著. —北京:国防  
工业出版社, 2006. 10

ISBN 7-118-04719-8

I. 数... II. 潘... III. 数字电路—计算机仿真  
IV. TN79

中国版本图书馆 CIP 数据核字(2006)第 095654 号

※

国防工业出版社出版发行

(北京市海淀区紫竹院南路 23 号 邮政编码 100044)

天利华印刷装订有限公司印刷

新华书店经售

\*

开本 787×1092 1/16 印张 16¼ 字数 369 千字

2006 年 10 月第 1 版第 1 次印刷 印数 1—3000 册 定价 32.00 元

---

(本书如有印装错误,我社负责调换)

国防书店:(010)68428422

发行邮购:(010)68414474

发行传真:(010)68411535

发行业务:(010)68472764

# 前 言

随着集成电路制造工艺的不断发展,数字电路与系统获得了广泛应用。它是现代信息技术的基石,不仅对国民经济起着巨大的推动作用,而且已渗透到了人们的日常生活中。当设计人员设计了一个电路之后,如果在设计中遗留微小的错误将可能造成巨大的经济损失或者灾难性的后果,因此需要验证电路的正确性。随着集成电路规模的不断增大,单个芯片上常常包含数百万个晶体管,使设计过程的复杂性迅速增加,也导致验证工作的难度加大。如何保证设计的正确性无疑是一个具有挑战性的任务。

在电路设计过程中通常是采用层次化设计方法,以解决电路设计的复杂性问题。根据人们对电路的认识能力,对电路从系统级、行为级、寄存器传输级、功能级、门级以及开关级等不同的级别进行描述,在每个级别上还可以细化为许多层次,使得设计对象逐级细化。相应地,电路设计则划分成若干个阶段,每一阶段对应一种从高层描述到低层描述的转换。在设计领域的转换过程中,每个步骤都有可能发生错误。因此,从设计者最初对电路功能、行为的描述到每个层次的设计结果,每一个阶段都要进行正确性验证,以保证设计中的错误早期发现和早期消除。

在 VLSI 设计领域,仿真与验证是整个设计流程中最重要、最复杂与最耗时的步骤。例如,在整个设计流程中,完成了设计输入以及综合、布局与布线,只能说明设计符合一定的语法规则,但其是否满足设计者的功能要求并不能保证,这需要设计者通过仿真对设计进行验证。仿真的目的就是在软件环境下,验证电路的行为与设想中的是否一致。例如,在 EDA 中仿真分为功能仿真和时序仿真。功能仿真是在设计输入之后还没有综合、布局与布线之前的仿真;时序仿真是在综合、布局与布线后,也即电路已经映射到特定的工艺环境后,考虑器件延时的情况下对布局与布线的网表文件进行的一种仿真,其中器件延时信息是通过反标时序延时信息来实现的。功能仿真的目的是设计出能工作的电路,这不是一个孤立的过程。它与综合、时序分析等形成一个反馈工作过程。只有过程收敛,之后的综合、布局与布线等环节才有意义;如果在设计功能上都不能满足,不要说时序仿真,就是综合也谈不上。因此,虽然 FPGA/CPLD 设计灵活、可以反复编程,在一定程度上可以弥补仿真与验证的不足,但是对于大型、高速或复杂的系统设计,仿真与验证仍是整个流程中最重要的一环,目前国内外知名公司验证和逻辑设计人员的配置比例超过 2:1。

本书着重介绍数字系统的仿真与验证的原理与方法,共分 11 章:第 1 章介绍了数字系统的设计流程、设计方式以及发展状况;第 2 章论述了电路的逻辑功能模型、

结构模型、层次模型等的建立方法,这也是电路设计与验证的基础;第3章介绍电路的逻辑仿真;第4章介绍 VHDL 高层次仿真;第5章描述了电路的开关级仿真;第6章论述了在电路验证方面的模型检验,阐述了计算树逻辑、符号模型检验等;第7章介绍了对逻辑布尔函数或数字电路进行有效表示与操作的二元判定图 BDD 的结构以及程序实现方法;第8章论述了二元判定图的性质,并阐述了它在数字电路的综合、验证以及在其他领域的应用;第9章介绍组合电路验证,特别阐述了在逻辑综合环境下的验证方法;第10章介绍时序电路的验证;第11章论述系统芯片(SOC)的验证,说明了芯核和芯片系统的验证方法。

在本书的编写过程中,得到了陈翎老师以及李学时、江泽洁、杨峻山、吴德林、董必立等研究生的支持与帮助,国防工业出版社的刘萍同志为本书的出版做了大量的工作,同时作者在本书中述及的一些研究工作也得到了国家自然科学基金项目(No. 60006002)和广东省高校自然科学研究项目(No. 02019)等的部分资助,在此一并表示衷心感谢。在作者的教学和科研工作中得到了电子科技大学的陈光祚教授、顾亚平教授,中山大学的张光昭教授,清华大学的陈景良教授等的培养、关心、支持与帮助,在此谨向他们致以最诚挚的谢意。

由于作者水平有限,书中难免存在疏漏之处,恳请专家和读者批评指正。

作者

2006年6月于广州

## 内 容 简 介

在数字电路的设计过程中,仿真与验证是一个重要的环节,它能检查所设计的电路是否符合要求。本书系统介绍了数字电路仿真与验证的原理与主要方法。全书共 11 章,内容包括:VLSI 设计的方法与基本流程;电路的逻辑功能模型、结构模型、层次模型等的建立方法;数字电路的逻辑仿真、高层次仿真、开关级仿真;模型检验;组合电路验证;时序电路验证;系统芯片 SOC 的验证;对逻辑布尔函数或逻辑电路进行有效表示与操作的二元判定图 BDD 的结构、性质及其在电路的综合、验证中的应用。

本书可作为电子、计算机、通信等专业高年级本科生和研究生的教材,也适合于从事数字系统设计、测试和维护的技术人员作为参考书。

# 目 录

|                             |    |
|-----------------------------|----|
| <b>第 1 章 绪论</b> .....       | 1  |
| 1.1 数字集成电路的设计流程 .....       | 1  |
| 1.2 数字集成电路的设计方式 .....       | 2  |
| 1.2.1 自上而下的设计和自下而上的设计 ..... | 3  |
| 1.2.2 基于单元与基于平台的设计 .....    | 3  |
| 1.2.3 芯片的正向设计 .....         | 4  |
| 1.3 系统芯片的设计 .....           | 6  |
| 1.4 电路设计的仿真与验证 .....        | 9  |
| 1.4.1 仿真的层次 .....           | 10 |
| 1.4.2 形式验证 .....            | 11 |
| <b>第 2 章 电路模型</b> .....     | 13 |
| 2.1 电路模型的相关概念 .....         | 13 |
| 2.1.1 数字系统的抽象级 .....        | 13 |
| 2.1.2 功能与结构模型 .....         | 14 |
| 2.2 在逻辑级的功能模型 .....         | 15 |
| 2.2.1 真值表和立方体 .....         | 15 |
| 2.2.2 逻辑函数的二元判定图表示 .....    | 16 |
| 2.2.3 功能模型的程序描述 .....       | 17 |
| 2.3 在寄存器级的功能模型 .....        | 18 |
| 2.3.1 寄存器传输语言的结构 .....      | 18 |
| 2.3.2 RTL 中的时序模型 .....      | 20 |
| 2.3.3 内部 RTL 模型 .....       | 21 |
| 2.4 结构模型 .....              | 21 |
| 2.4.1 结构模型的外部表示 .....       | 21 |
| 2.4.2 结构模型的性质 .....         | 22 |
| 2.4.3 结构模型的内部表示 .....       | 24 |
| 2.5 模型的层次 .....             | 25 |
| <b>第 3 章 电路的逻辑仿真</b> .....  | 28 |
| 3.1 仿真的原理 .....             | 29 |
| 3.2 编译仿真与事件驱动仿真 .....       | 30 |
| 3.2.1 编译仿真 .....            | 31 |
| 3.2.2 事件驱动仿真 .....          | 33 |
| 3.3 元件延迟与冒险检测 .....         | 34 |
| 3.3.1 元件输出值的计算 .....        | 35 |

|            |                          |            |
|------------|--------------------------|------------|
| 3.3.2      | 延迟模型 .....               | 36         |
| 3.3.3      | 冒险检测 .....               | 39         |
| 3.4        | 门级事件驱动仿真 .....           | 43         |
| <b>第4章</b> | <b>高层次仿真及工具软件 .....</b>  | <b>47</b>  |
| 4.1        | 功能仿真 .....               | 47         |
| 4.1.1      | 功能仿真的类型 .....            | 47         |
| 4.1.2      | 功能仿真的途径 .....            | 48         |
| 4.2        | 高层次仿真 .....              | 49         |
| 4.2.1      | VHDL 语言的基本结构 .....       | 49         |
| 4.2.2      | VHDL 仿真系统的结构 .....       | 52         |
| 4.2.3      | VHDL 内部模型的建立 .....       | 54         |
| 4.2.4      | VHDL 仿真算法 .....          | 60         |
| 4.3        | 仿真工具软件 ModelSim .....    | 64         |
| 4.3.1      | ModelSim 仿真软件的特点 .....   | 64         |
| 4.3.2      | 使用 ModelSim 进行设计仿真 ..... | 65         |
| <b>第5章</b> | <b>电路的开关级仿真 .....</b>    | <b>71</b>  |
| 5.1        | 开关级网络 .....              | 72         |
| 5.2        | 稳态响应 .....               | 74         |
| 5.2.1      | 布尔节点和晶体管状态 .....         | 74         |
| 5.2.2      | 三元状态 .....               | 77         |
| 5.3        | 稳态响应的分析 .....            | 79         |
| 5.4        | 稳态响应的代数表示 .....          | 80         |
| 5.4.1      | 有关根路径的一种代数 .....         | 80         |
| 5.4.2      | 从路径到方程 .....             | 81         |
| 5.4.3      | 不动点方程的形式 .....           | 83         |
| 5.4.4      | 不动点方程的例子 .....           | 84         |
| 5.5        | 仿真方程的求解 .....            | 85         |
| 5.5.1      | 求解稀疏方程的不动点 .....         | 85         |
| 5.5.2      | 基于增量的解方法 .....           | 87         |
| 5.6        | 仿真程序 .....               | 89         |
| 5.7        | 基于组的仿真 .....             | 93         |
| 5.7.1      | 估计一个组的稳定状态 .....         | 94         |
| 5.7.2      | 节点的信息量 .....             | 96         |
| 5.7.3      | 节点信息量的迭代估计 .....         | 97         |
| 5.7.4      | 仿真时的电路结构分析 .....         | 100        |
| <b>第6章</b> | <b>模型检验 .....</b>        | <b>103</b> |
| 6.1        | 验证的建模 .....              | 103        |
| 6.2        | 计算树逻辑的语法 .....           | 104        |
| 6.3        | 计算树逻辑的语义 .....           | 106        |
| 6.4        | CTL 公式间的等价性 .....        | 110        |
| 6.5        | CTL 验证的例子——进程互斥 .....    | 111        |

|              |                          |            |
|--------------|--------------------------|------------|
| 6.6          | 模型检验算法 .....             | 114        |
| 6.6.1        | 标签算法 .....               | 114        |
| 6.6.2        | 模型检验算法的程序实现 .....        | 116        |
| 6.7          | 符号模型验证系统 .....           | 118        |
| 6.7.1        | 验证系统的结构 .....            | 119        |
| 6.7.2        | 验证系统中的模块 .....           | 120        |
| 6.7.3        | 验证系统在互斥访问中的应用 .....      | 121        |
| 6.8          | 具有公正性的模型检验 .....         | 123        |
| 6.9          | CTL 的不动点特性 .....         | 125        |
| 6.9.1        | 单调函数 .....               | 127        |
| 6.9.2        | $SAT_{EG}$ 的正确性 .....    | 128        |
| 6.9.3        | $SAT_{EU}$ 的正确性 .....    | 129        |
| 6.10         | 符号模型检验 .....             | 131        |
| 6.10.1       | 状态集的表示 .....             | 131        |
| 6.10.2       | 迁移关系的表示 .....            | 133        |
| <b>第 7 章</b> | <b>二元判定图的结构</b> .....    | <b>136</b> |
| 7.1          | 二元判定图的概念 .....           | 136        |
| 7.2          | 二元判定图的约简 .....           | 139        |
| 7.3          | 二元判定图的程序实现 .....         | 144        |
| 7.3.1        | BDD 的遍历 .....            | 144        |
| 7.3.2        | BDD 约简的程序实现 .....        | 145        |
| 7.3.3        | 由布尔表达式经运算生成 BDD .....    | 146        |
| 7.3.4        | 生成 BDD 的 ITE 算法 .....    | 149        |
| 7.4          | 二元判定图的变量编序 .....         | 151        |
| 7.4.1        | 变量编序对二元判定图的影响 .....      | 151        |
| 7.4.2        | 静态变量编序 .....             | 152        |
| 7.4.3        | 动态变量编序 .....             | 155        |
| <b>第 8 章</b> | <b>二元判定图的性质与应用</b> ..... | <b>160</b> |
| 8.1          | 变量编序中变量交换的效果分析 .....     | 160        |
| 8.1.1        | 局部变量交换的效果 .....          | 160        |
| 8.1.2        | 使用低界值的过滤法 .....          | 164        |
| 8.2          | 最优编序 .....               | 165        |
| 8.3          | ZBDD 的定义与操作 .....        | 170        |
| 8.3.1        | ZBDD 的定义与性质 .....        | 170        |
| 8.3.2        | ZBDD 的操作 .....           | 171        |
| 8.3.3        | ZBDD 在布尔立方集代数中的应用 .....  | 173        |
| 8.3.4        | 用 ZBDD 进行多项式的表示与操作 ..... | 177        |
| 8.4          | 多级逻辑综合与 ZBDD .....       | 183        |
| 8.4.1        | 二元立方集的 ZBDD 表示 .....     | 183        |
| 8.4.2        | 立方集表示的分解 .....           | 184        |
| 8.5          | 使用 BDD 进行电路测试 .....      | 187        |

|             |                           |     |
|-------------|---------------------------|-----|
| 8.6         | 小结 .....                  | 189 |
| <b>第9章</b>  | <b>组合电路验证</b> .....       | 190 |
| 9.1         | 组合与时序电路的相关表示 .....        | 190 |
| 9.2         | 组合电路的验证方法 .....           | 194 |
| 9.3         | 引入断点 .....                | 195 |
| 9.4         | 组合电路验证方法的实现 .....         | 197 |
| 9.4.1       | 验证的步骤 .....               | 197 |
| 9.4.2       | 计算潜在的断点 .....             | 198 |
| 9.4.3       | 检测断点 .....                | 199 |
| 9.4.4       | 选取断点 .....                | 201 |
| 9.5         | 组合电路验证的部分实验结果 .....       | 202 |
| 9.6         | 小结 .....                  | 205 |
| <b>第10章</b> | <b>时序电路验证</b> .....       | 206 |
| 10.1        | 时序电路的验证方法 .....           | 206 |
| 10.2        | 时序电路的等价性 .....            | 208 |
| 10.3        | 可重置电路 .....               | 213 |
| 10.4        | 利用功能依赖 .....              | 213 |
| 10.5        | 时序电路验证方法的实现 .....         | 215 |
| 10.6        | 检测功能依赖性 .....             | 217 |
| 10.7        | 时序电路验证的部分实验结果 .....       | 221 |
| 10.8        | 小结 .....                  | 223 |
| <b>第11章</b> | <b>系统芯片 SOC 的验证</b> ..... | 224 |
| 11.1        | 系统芯片的结构 .....             | 224 |
| 11.2        | 系统芯片的设计过程 .....           | 225 |
| 11.2.1      | SOC 的设计流程 .....           | 225 |
| 11.2.2      | 软核的设计流程 .....             | 227 |
| 11.2.3      | 硬核的设计流程 .....             | 228 |
| 11.2.4      | 系统集成 .....                | 229 |
| 11.3        | 芯核的验证 .....               | 230 |
| 11.3.1      | 验证策略与验证目标 .....           | 230 |
| 11.3.2      | 测试平台的建立 .....             | 233 |
| 11.3.3      | 芯核时序验证 .....              | 234 |
| 11.3.4      | 芯核接口的验证 .....             | 235 |
| 11.4        | SOC 的系统级设计及验证 .....       | 236 |
| 11.4.1      | SOC 的系统级设计 .....          | 236 |
| 11.4.2      | SOC 的系统级验证 .....          | 237 |
| <b>附录</b>   | <b>名词缩写表</b> .....        | 244 |
| <b>参考文献</b> | .....                     | 246 |

# 第 1 章 绪 论

集成电路主要用于存储、传送和处理数字与模拟信息,是信息处理技术及其设备赖以存在的重要基石。近年来,电路设计技术得到了飞速发展。早期的集成电路是具有很少门数的小规模电路,随着技术的不断发展,出现了几百门的中规模集成电路和几千门至上百万门的大规模集成电路。从这时起,电路设计过程变得非常复杂,设计者已经感觉到自动化设计的必要性,开始使用计算机辅助设计工具来进行电路设计。目前随着超大规模集成电路技术的出现,设计者可以在单芯片上集成上万个晶体管。

本章简单介绍大规模集成电路的设计方法与发展趋势、电路仿真与验证的概念以及为后续各章提供的一些预备知识。

## 1.1 数字集成电路的设计流程

集成电路自问世以来,得到了迅速发展。1959 年,世界上第一块集成电路在美国德州仪器公司(TI)和美国仙童公司(Fairchild)诞生,这块集成电路上只集成了 4 个晶体管。1962 年世界上出现了第一块集成电路商品,1970 年 1Kb 的存储器问世,接着微处理器于 1971 年投入市场,从此宣告集成电路生产进入大规模集成电路时代。集成电路大大地促进了电子设备的小型化,与采用单个电子管和晶体管相比,采用集成电路可以大大降低电子设备的功耗与故障发生率,使对复杂功能的电子系统的设计制造成为可能。自从集成电路发明以来,经历了小规模(SSSI)、中规模(MSI)、大规模(LSI)、超大规模(VLSI)和甚大规模(ULSI)等发展过程。目前已进入系统芯片(SOC)时代,即可以将各种各样的功能模块(例如存储器、处理器、输入/输出接口电路、数字信号处理器等)集成在一个单一的芯片上<sup>[1,2]</sup>。

图 1.1 是使用硬件描述语言时超大规模数字集成电路设计的典型流程。

在电路的设计过程中,根据所设计电路要完成的功能,需要先写出“设计规范”。设计规范抽象地描述了所设计电路的功能、整体结构和接口。这时,没有必要考虑如何用具体电路来实现系统结构。通过分析电路的功能、性能、所要满足的接口标准以及其他的问题之后,就可以进行电路的“行为级描述”。

完成了行为级描述以后,通常需要进行向寄存器传输级(RTL)描述的转换,获得“RTL 描述”。这里的主要原因在于现有的自动化电子设计工具只能接受 RTL 级描述的 HDL 文件进行自动逻辑综合。转换后的 RTL 描述同样需要进行仿真与验证。在 RTL 描述中,设计者需要描述在所设计电路中的数据流。从这步之后,设计过程是在计算机辅助设计(CAD)工具的帮助下完成的。

“逻辑综合”的目标是将 RTL 的 HDL 代码映射到具体的工艺上加以实现,因此从这

一步开始,电路的设计过程与实现工艺相关联。一个电子系统的设计,都有一些设计指标,如时钟频率、芯片面积、端口驱动能力、可测性等,自动综合工具将这些设计指标作为综合过程的约束条件,在给定的包含工艺参数的综合库中选取最佳单元,实现综合过程。逻辑综合工具把 RTL 描述转换成“门级网表”,门级网表是使用门电路以及门电路之间的连接来描述电路结构的。称逻辑综合后的仿真为“门级仿真”。门级仿真与行为级仿真和 RTL 仿真不同的是,门级仿真包含了门单元的延时信息,因此门级仿真需要相应工艺仿真库的支持。门级网表是产生物理版图的自动布局与布线工具的输入。对于 IC 芯片设计者来说,这一步通常称为“后端设计”。在版图的布局与布线都已确定后,可以从版图中进一步提取出连线电阻、电容等参数。

在物理版图生成之后,把从中提取出的参数反标到门级网表中,进行包含门延时、连线延时的门级仿真,称为“后仿真”。这一步主要是进行时序模拟。如果时序不能满足设计要求,通常需要修改物理版图的布局与布线、逻辑综合的约束条件,有时也有可能回到 RTL 描述、行为级描述甚至在电路设计规范或算法实现上加以调整。物理版图得到验证后就可以做到硅片上。

由上可知,使用 CAD 工具和 VHDL 可以自动完成设计过程中的大部分工作。设计者如何控制和使用这些工具是非常重要的;如果使用不正确,设计效率将非常低。同时从图 1.1 可知,在电路设计的多个阶段,都必须进行仿真与验证。因此,对电路的设计者而言,掌握电路仿真与验证的常用方法的原理是必要的,有助于深入理解相关过程的实现并进行高效的电子系统设计。

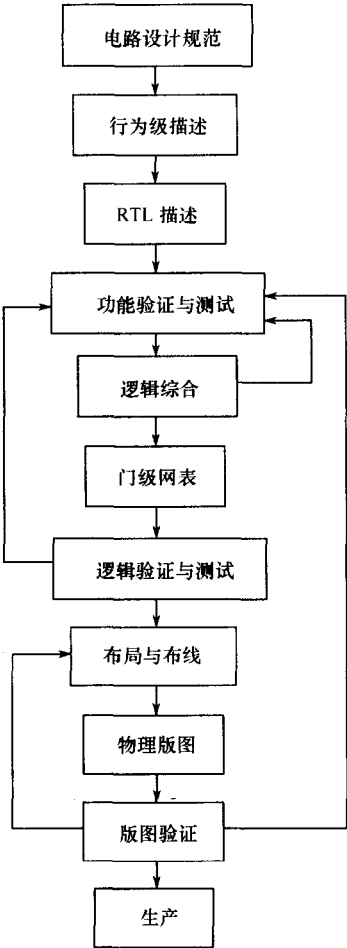


图 1.1 超大规模数字集成电路设计的典型流程

## 1.2 数字集成电路的设计方式

半导体和计算机技术的发展,使数字电子系统的设计理念和设计方法都发生了深刻的变化。从电子 CAD、电子 CAE 到电子设计自动化(EDA),设计的自动化程度越来越高,设计的复杂性也越来越高。以前,数字电路系统大多是采用搭积木式的方式进行设计的,即由一些固定功能的器件加上一定的外围电路构成模块,再由这些模块进一步形成各种功能电路。PLD(可编程逻辑电路)器件和 EDA 技术的出现改变了传统的设计思路,使人们可以通过设计芯片来实现各种不同的功能。新的设计方法能够由设计者自己定义

器件的内部逻辑和管脚,把原来由电路板设计完成的大部分工作放在芯片的设计中进行。这样不仅可以通过芯片设计实现多种数字逻辑功能,而且由于管脚定义的灵活性,减轻了原理图和印制板设计的工作量和难度,增加了设计的自由度和灵活性,提高了效率。

EDA 技术的出现使得数字系统的设计思想和设计方式发生了变化。一般说来,电子系统的设计有两种思路:自上而下的设计和自下而上的设计。

### 1.2.1 自上而下的设计和自下而上的设计

自上而下的设计也称为自顶向下的设计。它是首先把设计分成多个不同的层次(例如系统级、功能级和门级等);然后按照自上而下的顺序,在不同的层次上对系统进行设计与仿真,如图 1.2 所示。

首先在系统级对系统进行设计,并进行功能模块的划分与定义;然后在功能级对各个模块进行描述,并做功能仿真,以预测设计的正确性。如果仿真通过,则把功能描述转换成门级描述,或者转换成某一具体结构芯片的网表文件,将网表文件适配到具体芯片中进行布局与布线。在这之后还要进行时序仿真,以检查布局与布线的线延时和门延时对设计带来的影响。在整个设计过程中须经过“设计—验证—修改设计—再验证”的过程,不断反复进行,直到能够完全实现所要求的逻辑功能。

自下而上的设计则是一种传统的方法。设计者首先将各种基本电路单元或电路模块(例如各种基本门电路以及加法器、乘法器、存储器等)做成基本单元库;然后在设计时调用这些基本单元,逐级向上组合,直到设计出满足自己需要的系统为止。

目前,自上而下的设计占据了电路系统设计的主流地位,这是由于这种设计思想更符合人们逻辑思维的习惯,也容易使设计者对复杂的系统进行合理的划分与不断的优化。而自下而上的设计却往往使设计者关注了细节,对整个系统缺乏规划。当设计出现问题要修改时就非常麻烦。目前很多电子设计工具,也越来越多地支持自上而下的设计,许多设计软件都支持高层的设计和仿真。值得指出的是,自上而下的设计也并非绝对的,有时也需要用到自下而上的方法,这种方法是在系统划分和分解的基础上先进行底层单元设计,然后再逐步向上进行功能块和子系统的设计。

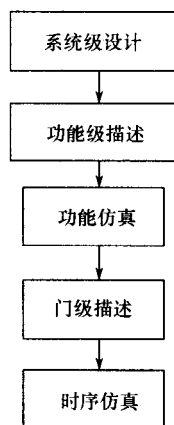


图 1.2 自上而下的设计

### 1.2.2 基于单元与基于平台的设计

使用基于单元的设计方法是结构化设计思想和设计复用思想的具体体现。按照电路结构化设计思想,电子系统将会被划分为较低层次的电路系统。如果在划分电路系统时遵循一定的结构规则和标准化规则,则可以在一定的层次把电路系统划分得更加规整,使相同或相似子系统可以被标准化,通过结构的重用来实现系统功能。因此人们可以事先设计小一些的标准子电路系统,构成单元库,按照一定的系统构建方法,用单元库中的单元来建立更高一级的电路系统<sup>[3]</sup>。

近年来随着系统芯片和芯核(IP 核)的出现,基于单元的设计方法也向 IP 复用技术方向发展。IP 核(Intellectual Property Core)在集成电路设计领域是指一种具有知识产

权的某种特定功能的电路设计与实现。随着人们对 IP 知识产权认识的提高,越来越多的 IP 核被开发出来。尤其是在通信领域和多媒体信息处理领域,各种 IP 核在被开发和转让。典型的 IP 核有各种微处理器、数字信号处理器、存储器、各种媒体标准处理器以及通信标准协议接口等。由于 IP 核一般设计工作量大,设计者在进行设计、仿真、优化、逻辑综合、测试等方面花费大量的时间,供应商在提供 IP 模块时,已经排除了语言描述的冗余性,并且经过了验证,因此电子系统设计者采用 IP 功能块进行设计时,可以集中精力去解决系统中的重点课题,并可将优化的 IP 功能块合并到其核心电路中进行逻辑合成。有了 IP 核,用户在进行电子产品设计时,既可以选择自己从底层进行产品开发,也可以购买第三方提供的标准模块和 IP 核,从而降低开发成本,缩短开发周期。

随着电路达到深亚微米制造阶段,系统芯片架构设计的工作量将超过物理设计,嵌入式软件开发的工作量将超过硬件设计,因此只有大幅缩减架构开发时间,更早地进行软件开发并完成系统芯片验证,才能保证系统芯片的上市时间并控制设计成本。目前,解决系统芯片技术瓶颈的一种方法就是提高设计的抽象级,并采用基于平台的设计方法。

基于平台的设计方法的演变如图 1.3 所示。早期的数字系统设计是基于晶体管的,它要求设计者必须使用版图编辑工具从晶体管的版图尺寸、位置及连线开始进行设计。在这一阶段,设计者最关心的是怎样减小芯片的面积。随着设计方法的改进,出现了以门级模块为基础的设计方法。门级模块为基础的设计是在 IC 厂家提供的电路单元基础上进行的设计。在这一阶段,设计者在考虑芯片面积的同时,更需要关注门级模块之间的延时问题。随着芯片集成度的进一步提高、系统芯片的出现、基于 IP 模块的设计的流行、IP 模块的复用等使设计周期大大缩短;但它也存在一个问题,就是在开发完一个产品后,如何能尽快开发出其系列产品。为此,出现了基于平台的设计方法。

基于平台的设计方法不同于现在集成电路通常的设计方法,即硬件开始于寄存器传输级,软件开始于编码级。基于平台的设计方法开始于一个纯粹的功能模型或者系统结构抽象模型,它是一种基于 IP、面向特定应用领域的系统芯片、系统 SOC 设计环境。不同的用户可以根据自己的应用领域来建立自己的平台,并调整平台所包含的模块。这样基于同一平台,可以迅速开发出一系列相关的项目。

基于平台的设计开发费用低廉,能在更短的时间内设计出满足需要的电路,特别适合于专用集成电路(ASIC)的应用需求,具有更大的灵活性。特别是,基于平台的系统芯片的设计可以带来一系列新的变化。例如,硬件描述语言倾向于采用 C++/System C 来进行电路设计和嵌入式软件的仿真,可比传统的寄存器传输级方法快很多倍;把电路的设计的自顶向下方法改为层次性设计方法;使用模型构建可复用设计平台,可以快速完成新的派生设计。

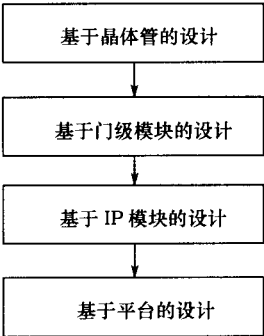


图 1.3 基于平台的设计方法的演变

### 1.2.3 芯片的正向设计

电子系统的设计一般采用自上而下的方式,但是如果是设计芯片,则通常采用正向设

计方式,它也是一种自上而下的方式,其流程如图 1.4 所示。

这里“系统描述”就是在最高层对芯片进行规划,包括芯片的功能、成本、功耗甚至尺寸大小等一系列指标,并确定选择什么样的工艺。“功能设计”是对系统的功能行为特性进行设计,常用的方法是时序图、子模块关系图和状态机等。通过“逻辑设计”这一步将得到系统的逻辑结构,并且要反复模拟以验证其正确性。之后,需要对设计进行综合和优化。经过“电路设计”这一步,就可以把设计转化为晶体管级或电路级,通常用详细的电路图来表示。

“版图设计”也称为“物理设计”,这是芯片设计中最费时的一步,它要把每个元件的电路表示转换成几何表示,同时,元件间的网表也被转换为几何连线图形。这种电路的几何表示即为版图。版图设计要符合与制造工艺有关的设计规则要求,通常要进行物理设计规则检查、版图网表提取、电学规则检查、版图和原理图一致性比较等一系列检查,以确保版图设计的正确性。“芯片制造”也称为流片,是指把经过验证的版图送到半导体厂家去做芯片,一般要经过硅片准备、注入、扩散和光刻等工艺。“芯片的封装与测试”,芯片的封装形式有多种,可以根据需要封装为贴片或其他形式,同时对芯片进行功能、故障的测试,以确保芯片的性能满足要求。

在如上的芯片设计过程中,需要不断进行仿真与验证,依次为功能仿真、时序仿真和版图验证等,只有这样,才能保证设计的正确性。

上面介绍了数字系统的设计与芯片的设计过程。在设计过程中,必然涉及到器件。选择什么样的器件关系到数字系统设计的周期、成本和风险等,因此需要综合考虑。下面首先对数字器件的设计进行说明。数字器件的设计按其实现方式可分为全定制和半定制两种。半定制设计又包括门阵列法、标准单元法和基于可编程逻辑器件的设计方法等,如图 1.5 所示。



图 1.4 芯片的正向设计

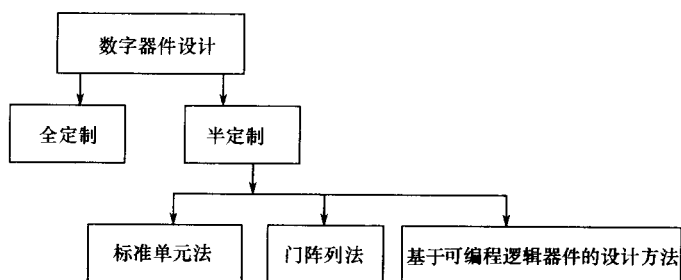


图 1.5 数字器件的设计

全定制设计是一种基于版图级的设计方法,即是在物理版图级实现设计的方法。它要求设计者必须使用版图编辑工具从晶体管的版图尺寸、位置及连线开始进行设计。运用全定制法设计芯片,当芯片的功能、性能、面积和成本确定后,设计人员要对芯片结构、逻辑和电路进行精心的设计,对不同的方案反复比较,对单元电路的结构、晶体管的参数反复优化。在版图设计时,设计人员要手工设计版图并精心地布局与布线,以获得最佳的

性能和最小的面积。由此可见,采用全定制法可以设计出高速度、低功耗、小面积的芯片。但这种设计的周期长,设计成本高,只适用于对性能要求很高或批量很大的芯片。

半定制设计是一种约束性设计方法,它是在厂家预先设计的基本单元基础上进行的二次设计。在一些产品的产量不大或者不允许设计时间太长时,采用半定制法进行设计就比较适宜。在有些情况下,为了尽快地把产品推向市场,可先采用半定制法。半定制法主要有标准单元法、门阵列法、可编程逻辑器件法等。

在标准单元法中,一些基本的电路单元例如各种基本门、D 型触发器、全加法器等版图是预先设计好的,放在 CAD 工具的版图库中。设计者先利用 CAD 工具绘制电路方框图,然后 CAD 工具能够利用方框图中单元逻辑的电路符号与单元电路版图的对应关系,自动布局与布线,生成版图。设计者也可以利用标准单元的版图,人工布局与布线。一般来说,人工布局与布线的硅片面积利用率高,但费时较多。标准单元法不要求设计者必须具有专门的半导体工艺知识。

门阵列法又称母片法。母片是 IC 工厂按照一定规格事先生产的半成品芯片。硅片出厂时已有预先制造好的大量规则排列的晶体管,芯片四周预制好 I/O 焊盘,芯片中只留下一层或两层金属铝连线的掩膜,需根据用户电路的不同而定制。门阵列法的设计是在 IC 厂家提供的电路单元基础上进行的,设计者只需掌握很少的集成电路知识,设计过程比较简单,并且生产周期短、成本低。其缺点是门的利用率不高,芯片面积大,而且母片上制造好的晶体管都是固定尺寸的,不利于设计出高性能的芯片。所以这种方法适用于一些对芯片性能要求不高的场合。

基于可编程逻辑器件的设计方法。可编程逻辑器件是一种已完成了全部工艺制造可直接从市场上购得到的产品,用户只要对它编程就可实现所需要的电路功能。前面的全定制法和门阵列法都必须到 IC 工厂去加工制造才能完成,设计周期长,而且一旦有了错误,就需重新修改设计,大大增加了设计成本和设计时间。采用 PLD 器件,设计者在实验室足不出户就可以设计出满足自己需要的芯片,而且可以反复修改,反复编程,直到完全满足要求为止。PLD 器件从诞生到现在,功能越来越强,规模越来越大,价格不断降低。

以上介绍的几种设计器件的方式,各有优缺点,对电路设计开发者而言,应根据实际情况来选择一种最适合自己的方式。

### 1.3 系统芯片的设计

近年来随着集成电路的深亚微米制造技术的迅速发展,已经可以将一个完整的系统集成到一个单一的芯片中,即系统芯片(SOC)。系统芯片在单一芯片上实现了信号采集、转换、存储、处理和 I/O 等功能,或者说在单一芯片上集成了数字电路、模拟电路、信号采集和转换电路、存储器、微处理器、DSP(数字信号处理)、MPEG(运动图像专家组)等,实现了一个系统的功能。SOC 是在 ASIC 的基础上发展起来的,它与 ASIC 完全不同,具有很多独特的优点:功能强、功耗低;降低了整机的成本与体积,促进了整机系统更新换代的速度。它的这些优点正好顺应了通信、计算机、消费类电子产品向轻、薄和耗电少等发展方向,因此市场对 SOC 产品有强烈的需求。

系统芯片是采用深亚微米级工艺,采用芯核(IP 核)来设计的。这与传统的 IC 设计

技术完全不同,器件的物理特性和电学特性将会发生很大的变化,因此给系统芯片设计带来了新的困难。集成电路设计进入深亚微米阶段后,特征尺寸缩小,其横向和纵向尺寸也都大大缩小,芯片内的互连线长度却急剧增大,互连线与连线间的电阻和电容对信号传输的影响非常显著。由于信号频率高,也会引入串扰影响和噪声影响;由于互连线变细,易于引起电迁徙和热载流子效应,因此在集成电路的设计策略上需做较大的调整<sup>[4~6]</sup>。过去的面向逻辑的设计方法已不适用,而改为面向互连的设计方法。特别是深亚微米级芯片的速度较快,这时对时序的要求更为严格,因此前端的逻辑设计与后端的物理设计间很难保持一致<sup>[7~9]</sup>。对于在逻辑设计中仿真分析后功能和时序都正确的网表,在布线设计后却由于芯片空间和连线的限制,造成互连引线的延迟与逻辑设计中使用的模型不一致,使得时序不再满足约束的要求。这时必须反过头来回到逻辑设计中进行修改,然后再进行仿真分析。因而会使整个设计周期加长。

此外,深亚微米集成电路的设计在设计验证方面同样会遇到很大的挑战,完整的设计验证已变得不可能,接口的验证成为验证工作的主体。软件、硬件协同验证是解决系统验证的有效方法,但这种协调验证必须找到合适的准则和方法。深亚微米集成电路的设计流程如图 1.6 所示。

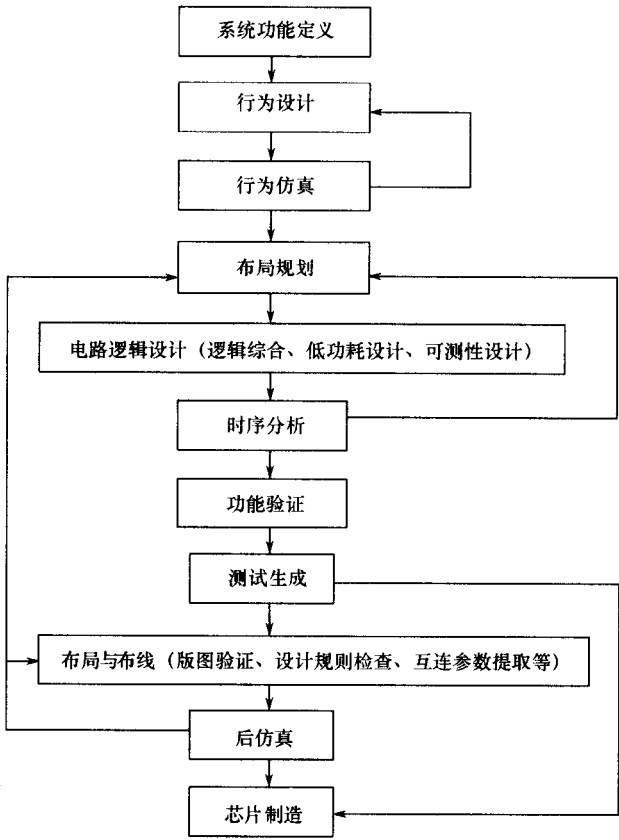


图 1.6 深亚微米集成电路的设计流程

系统芯片设计的一个显著特点是对 IP 核的设计复用或重用。首先,IP 核也称知识