



中国电子企业协会 职业认证培训教材  
全国IC设计师考试认证委员会

## IC设计中级教程

SPECILLY DESIGNED  
FOR ENGINEERS  
AND  
TECHNICIANS OF  
ELECTRONICS

# Digital IC System Design 数字IC系统设计

王彬 任艳颖 编著

西安电子科技大学出版社  
<http://www.xduph.com>

中国电子企业协会  
全国 IC 设计师考试认证委员会  
职业认证培训教材

# 数字 IC 系统设计

王彬 任艳颖 编著

图例 (CIP) 目録列在并图

数字 IC 系统设计 / 王彬, 任艳颖. — 西安: 西安电子科技大学出版社, 2005

中国电子企业协会职业认证培训教材 / 西安电子科技大学出版社

ISBN 7-5506-1261-1

I. 数… II. 王… III. 数字集成电路—设计—教材 IV. TP369.1

中国版本图书馆 CIP 数据核字 (2005) 第 082947 号

编 著 任 艳 颖

编 审 朱 洪 涛

西安电子科技大学出版社 (西安)

电 话 (029) 8253366

http://www.xjtu.edu.cn

经 销 新华书店

印 刷 西安电子科技大学出版社

开 本 787mm×1092mm

定 价 20.00 元

版 次 2005 年 1 月第 1 版

印 次 2005 年 1 月第 1 次印刷

ISBN 7-5506-1261-1

XDJTP 1828001-1

西安电子科技大学出版社

2005

西安电子科技大学出版社

## 内 容 简 介

IC 设计是一个实践性很强的行业。IC 设计师经常发现，书本上学到的东西，和实践往往有一定的距离。本书结合最先进的工具和设计方法，针对 IC 系统设计中几个重要的专题进行了较为深入的讨论。

全书共分 9 章，第 1 章概述了 IC 设计流程及常用 EDA 工具；第 2 章介绍了算法及架构设计，对数字信号处理算法的开发、AMBA 片上总线、SystemC 设计语言进行了概述；第 3 章对 RTL 设计中的一些重要问题进行了说明，并给出了 HDL 设计指南；第 4 章对逻辑综合库进行了深入讲解，这是理解逻辑综合和静态时序分析的基础；第 5 章介绍了高级综合技术，包括自动芯片综合和物理综合；第 6 章对可测性设计进行了较深入阐述；第 7 章讲解了静态时序分析的一些重要专题；第 8 章介绍了实际中的形式验证技术；第 9 章对低功耗设计技术进行了讨论。

本书主要针对 IC 设计人员，也可作为高等院校相关专业师生的参考书。

## 图书在版编目(CIP)数据

数字 IC 系统设计/王彬等编著. —西安:西安电子科技大学出版社, 2005.9

中国电子企业协会全国 IC 设计师考试认证委员会职业认证培训教材

ISBN 7-5606-1567-8

I. 数… II. 王… III. 数字集成电路-电路设计-技术培训-教材 IV. TN431.2

中国版本图书馆 CIP 数据核字(2005)第 087947 号

策 划 戚文艳

责任编辑 张 友 戚文艳

出版发行 西安电子科技大学出版社(西安市太白南路 2 号)

电 话 (029)88242885 88201467 邮 编 710071

http://www.xduph.com

E-mail: xdupfxb@pub.xaonline.com

经 销 新华书店

印刷单位 西安文化彩印厂

版 次 2005 年 9 月第 1 版 2005 年 9 月第 1 次印刷

开 本 787 毫米×1092 毫米 1/16 印张 14.5

字 数 341 千字

印 数 1~4 000 册

定 价 22.00 元

ISBN 7-5606-1567-8/TN·0314

**XDUP 1858001-1**

\*\*\* 如有印装问题可调换 \*\*\*

本社图书封面为激光防伪覆膜，谨防盗版。

# 前 言

IC设计有两个发展趋势：宏观上，它在向系统级演化，这就是所谓的片上系统（System On a Chip, SOC）；微观上，它已经进入了纳米尺度设计的范畴（某些高端设计已开始采用90纳米及65纳米工艺）。这两种发展趋势大大加剧了IC设计的复杂性，增加了设计风险，并对设计者提出了更高的要求。

为了帮助设计者掌握复杂IC系统的设计方法，掌握最先进的工具和设计流程，应对深亚微米设计中的诸多问题，我们编写了本书，对算法架构设计、RTL设计、综合库、逻辑综合、静态时序分析、可测性设计、形式验证、低功耗设计这些问题进行了深入讨论。如果你已有一定的IC设计经验，渴望进一步提高自己，那么本书将会为你提供直接的帮助。

本书第1章对IC系统设计进行了概述，重点是介绍满足复杂系统设计及深亚微米设计的设计方法，帮助读者掌握先进的设计流程。尽管目前我国大多数工程师还在0.25微米以上的工艺上完成设计，系统规模也不算复杂，但我们必须了解90纳米/65纳米的设计流程是怎么样，数百万门的设计是如何完成的，因为明天我们也要完成这样的设计。

第2章对算法与架构设计进行了阐述。这里强调了“系统设计”的概念。这是因为现代IC系统的复杂性越来越高，许多芯片都超过百万门，千万门的设计也已出现；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。在这种背景下，迫切需要设计者具备系统级设计的能力。本章讨论了算法设计的一般步骤，介绍了常见的片上系统总线，讲解了基于SystemC语言的设计流程。需要说明的是，算法设计与应用密切相联，涵盖的内容极其丰富，因而不可能在一章中进行全面论述。我们只是对最基本的内容进行了介绍，而忽略了其余。MATLAB在算法工程师中应用极广，作为DSP算法的首选开发工具，它拥有很大的用户群。但考虑到市场上MATLAB方面的资料已经很多了，所以本书对此并未进行论述。

第3、4、5章对逻辑设计进行了介绍。

第3章对数字IC系统的RTL设计进行了介绍。我们针对的是具有一定RTL设计经验的读者，因此本章不对HDL语言本身进行说明，而是帮助读者解决在RTL设计中经常碰到的、非常重要的一些问题，例如如何设计复杂的状态机，如何处理设计中的多时钟域等。本章还给出HDL设计指南，以帮助读者确立良好的设计风格，提高书写代码的品位。在写RTL代码时需要考虑性能、面积、可复用性、易读性、可测性、功耗等诸多因素。这些要求可能是互相矛盾的，因此需要设计者根据需要进行权衡。“运用之妙，存乎一心”。写出高质量的代码并非易事。

大多数数字IC都采用标准单元来实现。如果将IC设计看作是盖房子，那么标准单元相当于砖瓦。好的建筑师要了解砖瓦的特性，好的IC设计师要了解标准单元的特性。第4章对标准单元进行了较为深入的讨论，对反相器与寄存器的结构进行了说明，并对综合库进行了介绍。这部分内容非常重要，是深入了解数字IC系统的前提，也是进行综合、静态

时序分析、功耗分析等的前提。在本章的最后，还对数据通路的 designware 实现进行了说明，以帮助读者深入综合时所用的 IP 库，以便在实际应用中进行剪裁。

在 RTL 设计完成后，需要借助逻辑综合工具生成门级网表。在第 5 章，针对复杂的系统，介绍了自动芯片综合(ACS)的流程；对逻辑综合中一些重要的命令、变量进行了解释，并对实践中经常要用到的一些问题给出了解决方案。

可测性设计是现代 IC 设计中的重要内容。绝大多数 IC 系统中都包含了可测性设计。第 6 章对这一专题进行了较为深入的讨论，介绍了集成电路的各种测试方法，包括扫描链测试、JTAG 测试、BIST 测试，并分析了常用故障机理和模型。然后针对这几种测试方式，给出了相应的设计流程和脚本。

静态时序分析是 IC 设计师必备的技能之一。通过静态时序分析，可以检查设计的时序是否满足要求。第 7 章对这一重要内容进行了讲解，重点是解答设计者在实际工作中经常遇到的一些问题，例如时钟的建模、多周期路径的设置、测试模式的时序分析、时钟门控逻辑的处理等。相信读者在看完这一章后，对于静态时序分析会有更为深入的理解。

第 8 章介绍了形式验证的流程。在流片前，往往要对网表进行多次修改。这时候，需要依赖形式验证工具来保证 RTL 设计与网表之间的一致性。形式验证是用数学的方法来完备地验证电路功能与设计规范是否一致，现在已成为标准设计流程的一部分。本章对最常见的形式验证工具 verplex 及 formality 的使用进行了介绍。

Intel 声称：功耗问题是决定摩尔定理能否继续适用的惟一因素。由此可见低功耗设计在现代 IC 系统中的重要性。第 9 章对低功耗设计这一专题进行了深入讨论，详述了 IC 系统中各种功耗的成因，阐述了系统级、RTL 级、逻辑级、电路级、物理级的低功耗设计技术，并介绍了功耗分析的流程。相信学完本章后，读者会对低功耗设计有一个清晰的了解。

对于 IC 设计师来说，最宝贵的不是掌握的知识，不是经验，而是永远有好奇心，永远有求知欲。集成电路这个行业真正吸引人的地方在于它的速度。从诞生之初，它就一直遵循着摩尔定理飞速发展，当前最先进的设计技术，可能要不了多久就会落伍。每天都有一些新东西需要被认识，每天都有一些新问题需要被解决，就跟玩过山车一样刺激。愿我们大家都能享受到它的乐趣。

对 IC 设计者来说，实际经验尤其宝贵。在书中，我们将自己的一些设计经验与读者共享，希望能够为读者提供直接帮助。如果读者对本书有什么建议，请发送邮件到 [algor\\_ren@sohu.com](mailto:algor_ren@sohu.com)，以利于我们改进自己的工作。

编者  
2005 年 8 月

# 目 录

<b>第 1 章 IC 系统设计概述</b> .....	1
1.1 IC 系统组成概述 .....	1
1.2 IC 系统设计概述 .....	6
1.2.1 系统级设计 .....	7
1.2.2 电路/逻辑设计 .....	9
1.2.3 物理设计 .....	11
1.3 IC 系统验证分析概述 .....	13
1.3.1 仿真 .....	13
1.3.2 静态时序分析 .....	15
1.3.3 功耗分析 .....	15
1.3.4 形式验证 .....	16
1.3.5 物理验证(DRC/LVS) .....	17
1.3.6 信号完整性分析 .....	18
1.3.7 基于 FPGA 的验证 .....	19
1.3.8 测试 .....	20
1.4 IC 系统设计的现状 .....	21
<b>第 2 章 系统设计——算法与架构</b> .....	22
2.1 算法设计 .....	23
2.1.1 算法设计基础 .....	23
2.1.2 数字信号处理(DSP)算法综述 .....	27
2.1.3 DSP 算法的描述 .....	31
2.2 IC 系统架构设计 .....	31
2.2.1 SOC 架构 .....	33
2.2.2 AHB 总线 .....	36
2.3 基于 SystemC 的 IC 系统设计 .....	40
2.3.1 基于 SystemC 的设计流程 .....	41
2.3.2 SystemC 的数据类型 .....	43
2.3.3 SystemC 建模基础 .....	44
2.3.4 利用 SystemC 进行系统建模的流程 .....	48
2.4 系统设计工具 SPW 简介 .....	54
<b>第 3 章 数字 IC 系统的逻辑设计——RTL 实现</b> .....	56
3.1 RTL 设计基础 .....	56
3.1.1 同步电路设计要求 .....	56
3.1.2 RTL 设计步骤 .....	58

3.1.3	复位策略 .....	60
3.1.4	状态机的设计 .....	62
3.1.5	多时钟域的处理 .....	67
3.1.6	时钟切换电路 .....	77
3.2	RTL 设计指南(Verilog) .....	78
3.2.1	命名规则 .....	78
3.2.2	设计风格 .....	79
<b>第 4 章</b>	<b>数字 IC 系统逻辑设计基础</b> .....	<b>90</b>
4.1	数字 IC 系统基础: 晶体管、反相器、寄存器 .....	90
4.1.1	MOS 晶体管 .....	90
4.1.2	反相器 .....	92
4.1.3	寄存器 .....	96
4.2	标准单元 .....	98
4.2.1	标准单元的仿真模型基础 .....	98
4.2.2	标准单元的综合模型基础 .....	101
4.2.3	反相器的综合模型 .....	107
4.2.4	寄存器的综合库模型 .....	110
4.3	数据通路的 designware 实现 .....	120
4.3.1	加法器 Dw01_add .....	122
4.3.2	乘法器 Sw02_mult .....	126
<b>第 5 章</b>	<b>IC 设计中的综合技术</b> .....	<b>131</b>
5.1	逻辑综合 .....	131
5.1.1	综合策略 .....	131
5.1.2	DC 中的重要变量及命令 .....	144
5.2	物理综合 .....	146
<b>第 6 章</b>	<b>可测性设计</b> .....	<b>149</b>
6.1	可测性设计综述 .....	149
6.1.1	测试仪 .....	150
6.1.2	故障模型 .....	151
6.1.3	可测试设计方法概述 .....	153
6.1.4	自动测试向量生成 .....	154
6.2	基于扫描路径的可测性设计 .....	155
6.2.1	扫描链基本原理 .....	155
6.2.2	面向扫描测试的 RTL 设计 .....	159
6.2.3	扫描链的综合 .....	162
6.3	基于 JTAG 的可测性设计 .....	168
6.3.1	JTAG 基础 .....	168
6.3.2	边界扫描的实现 .....	173
6.4	基于 BIST 的可测性设计 .....	175

6.5 自动测试向量的生成——ATPG .....	176
<b>第7章 静态时序分析 .....</b>	<b>180</b>
7.1 静态时序分析原理 .....	180
7.2 静态时序分析中时钟的建模 .....	185
7.3 静态时序分析中的常见问题 .....	187
7.3.1 多周期路径的设置 .....	187
7.3.2 DFT 模式 .....	189
7.3.3 时钟门控信号的时序分析 .....	191
<b>第8章 形式验证 .....</b>	<b>193</b>
8.1 等价性验证基础 .....	193
8.2 formality 脚本 .....	195
8.3 verplex 脚本 .....	196
<b>第9章 低功耗设计与功耗分析 .....</b>	<b>198</b>
9.1 IC 系统中的功耗 .....	198
9.2 低功耗设计技术综述 .....	200
9.2.1 系统级的低功耗设计 .....	203
9.2.2 RTL 级的低功耗设计 .....	207
9.2.3 逻辑级的功耗优化技术 .....	214
9.2.4 电路级的低功耗设计 .....	217
9.2.5 物理级的低功耗设计 .....	217
9.3 功耗分析 .....	217
9.3.1 基于 Power Compiler 的功耗分析技术 .....	219
9.3.2 基于 PrimePower 的功耗分析技术 .....	220
<b>附录 版本管理软件 CVS .....</b>	<b>222</b>
<b>参考文献 .....</b>	<b>224</b>



# 第1章 IC系统设计概述

IC设计有两个发展趋势：宏观上，它在向系统级演化，这就是所谓的片上系统(System On a Chip, SOC)；微观上，它已经进入了纳米尺度设计的范畴(某些高端设计已开始采用90纳米及65纳米工艺进行设计)。

这两种发展趋势大大加剧了设计复杂性，增加了设计风险，并对Foundry、EDA工具开发商、设计者提出了更高的要求。

片上系统的一大优势是可以大大降低产品的造价。因此，片上系统在通信、多媒体等领域的应用越来越普遍。通常来说，片上系统包括至少一个嵌入式处理器，使用片上总线连接嵌入式处理器与其他模块。

开发片上系统必须采用基于IP的开发模式，以应对系统日益复杂的功能需求。片上系统设计中要解决如下难点：

- 如何进行设计复用。
- 如何验证别人设计的IP。
- 如何顺利地将不同的IP集成在一起，使它们能协调工作。
- 如何进行系统验证。
- 如何进行软、硬件协同设计和验证。

深亚微米设计中，比较棘手的问题包括：

- 连线延时的估计。由于连线延时与单元延时相当，会导致设计难以收敛的问题。
- 串扰的分析与处理。在深亚微米设计中，相邻连线会出现耦合现象，称为串扰。串扰会导致延时不规律地变化、性能恶化甚至功能出现错误。
- 电源线上的欧姆电压降的处理(包括静态电压降与动态电压降)。电源/地线上的电阻产生的欧姆电压降会引起时序恶化及功能错误。据统计，在第一次流片失败的设计中(0.18  $\mu\text{m}$  以下)，有20%是由于IR drop引起的。
- 在深亚微米设计中，功耗已成为一个严重的问题。在纳米尺度设计中，漏电流引起的功耗开始占据主要地位。

本章将对上述IC系统设计问题进行讨论。

## 1.1 IC系统组成概述

IC系统是什么？对这个问题，算法设计工程师、架构设计工程师、电路设计工程师、版图设计工程师会给出不同的答案。

算法设计工程师说，IC 系统是完成特定功能的硬件。

架构设计工程师说，IC 包括控制、运算、存储部分。

电路设计工程师说，这是加法器、乘法器、与非门、运算放大器、开关电容等的搭配。

版图设计工程师说，它是多边形组成的集合。

这些答案都对。如果把它们组合起来，就是一个较为完备的答案。

图 1.1 给出了一个常见 IC 系统组成的示例。在这个系统中，包括如下内容：

- 数字部分(可能包括微处理器、控制电路、数据通路等)；
- 模拟部分(可能包括 PLL、A/D、RF 等)；
- 连线；
- I/O PAD；
- 存储器。

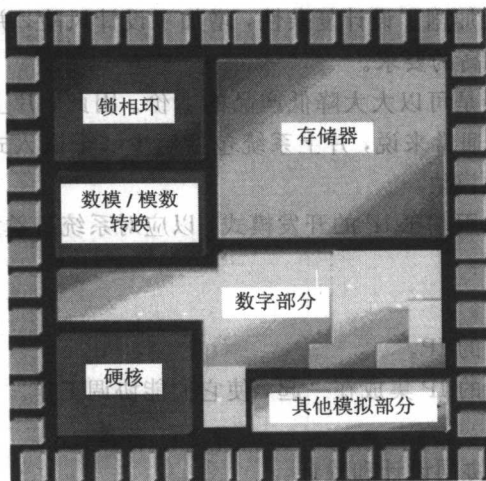


图 1.1 常见 IC 系统组成示例

下面对各个部分进行简要说明。

### 1. 数字部分

数字部分是构成 IC 系统的主要部分，也是本书讨论的重点。绝大多数数字设计采用同步设计方法，即采用时钟来统一协调系统各部分的运行。同步设计是数字设计的主流，因此本书只讨论同步设计，不涉及异步设计。

同步数字设计可以看作是由组合逻辑和时序单元(寄存器和锁存器)组成的，如图 1.2 所示，数据在时序单元—组合逻辑—时序单元这种结构间一级级地传递下去。

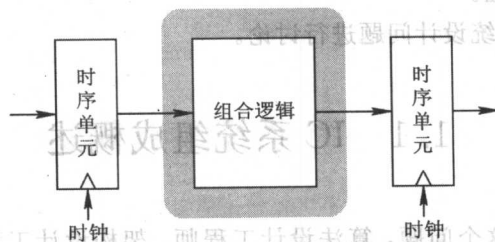


图 1.2 同步设计的组成

与模拟电路不同，不同 IC 系统，其中的数字部分千差万别，难以进行归类。通常可以将数字部分大致分为数据通路部分与控制部分。数据通路是指乘法器、加法器、有限响应冲激滤波器(FIR)等规则的运算结构，控制部分则是控制管理数据流通的逻辑。

## 2. 模拟部分

真实的世界是一个模拟的世界。因此，在一个 IC 系统中，要与外部模拟世界交互，模拟电路必不可少。

一般说来，在 IC 系统中模拟电路包括如下部分：

- 模/数转换器(ADC)：将模拟信号转换为数字信号；
- 数/模转换器(DAC)：将数字信号转换为模拟信号；
- 可编程增益放大器(PGA)：通过数字部分来控制模拟增益；
- 锁相环(PLL)：用于产生高频的时钟和进行时钟恢复(即从数据中恢复出时钟)；
- 其他。

图 1.3 给出了一个 IC 系统的模拟部分。这是一个收发器(transceiver)的示例，它主要由 A/D 转换器、D/A 转换器、时钟恢复电路、时钟生成电路(晶振、锁相环等)等构成。

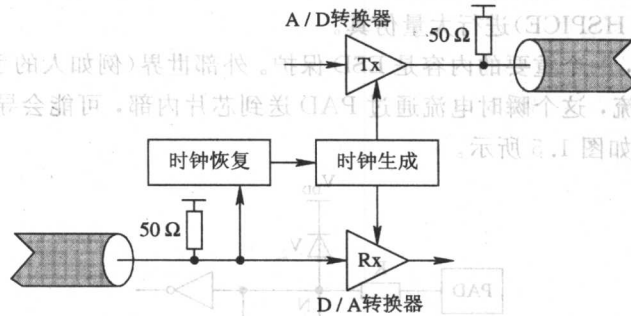


图 1.3 IC 系统中模拟部分示意图

该收发器从接收的数据中恢复时钟。接收到的模拟信号由 A/D 转换器变为数字信号。要发送的数据由 D/A 转换器转变成模拟信号，然后发送出去。

在许多通信芯片、视频处理芯片、音频处理芯片中，模拟电路部分直接决定了芯片的性能、造价等。

说明：I/O 及 RAM 的设计也属于模拟电路设计的内容。这里为清楚起见，将 I/O 和 RAM 单独进行说明。

## 3. I/O 部分

I/O 是 IC 系统与外部环境的接口。I/O 可以非常简单，也可以非常复杂。

Foundry 提供的综合库中，通常包括三类 PAD 单元：输入 PAD、输出 PAD 和双向 PAD。在设计中，可以将这些 PAD 看作是标准单元，直接例化就可以了。

I/O 也可以非常复杂。例如，USB、1394(fireware)、PCI-E 等都可以看作是复杂的 I/O。这些复杂的 I/O 可以作为 IP 集成到整个芯片系统中。

I/O PAD 直接与外部世界相联系，因此必须考虑到外部的寄生参数影响、静电保护(ESD)、封装要求、电压转换、过电压保护、信号完整性等。设计 I/O PAD 比设计标准单元要复杂许多，特别对于有高性能与低功耗要求的场合更是如此。

在板级的世界，有许多的 I/O 标准，例如 LVDS(Low Voltage Differential Signal, 低电压差分信号)、TTL 等。

在高速板级设计中，为了减小噪声、反射等的影响，通常要将信号以差分的形式进行传送，这就需要能够传送差分信号的 PAD。图 1.4 给出了一个带差分引脚的 I/O PAD 的示意图。

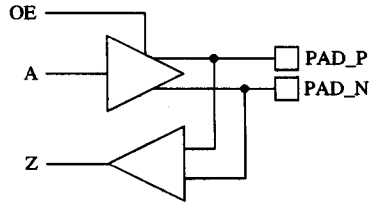


图 1.4 带差分引脚的 I/O PAD 示意图

I/O PAD 上的功耗可能占到整个 IC 系统功耗的 10%。所以，如果芯片用于功耗要求苛刻的场合，例如用于手机等便携式设备，那么 I/O 的设计必须关注功耗。此外，I/O PAD 的性能对整个芯片的速度影响也很大。为了分析 I/O 的速度、功耗等性能指标，需要借助 EDA 工具(如 HSPICE)进行大量仿真。

在 I/O 设计中，一个重要的内容是 ESD 保护。外部世界(例如人的手指)携带的静电会产生很大的瞬时电流，这个瞬时电流通过 PAD 送到芯片内部，可能会导致芯片失效。

ESD 保护电路如图 1.5 所示。

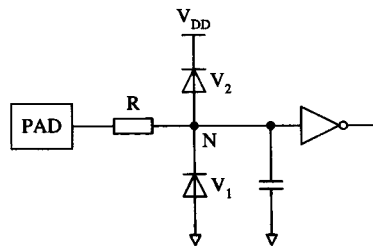


图 1.5 ESD 保护电路示例

为了降低板上噪声的影响，在输入 PAD 上，一般包含施密特触发器(Schmitt Trigger)单元。施密特触发器的功能如图 1.6 所示。可以看到，由于干扰等影响，输入信号上有振荡；输入信号经过施密特触发器之后，这些干扰就被消除了。

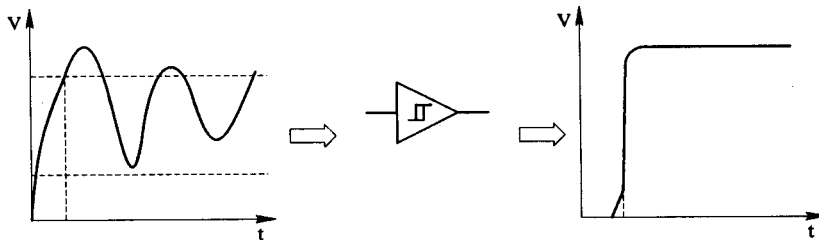


图 1.6 施密特触发器的功能

#### 4. 存储器

随着 SOC 这种架构的流行，嵌入式存储器在芯片中的应用越来越多。如图 1.7 所示，到 2005 年，存储器差不多占据了整个芯片 70% 的面积。随着时间的推移，这一比例还会继续增长。

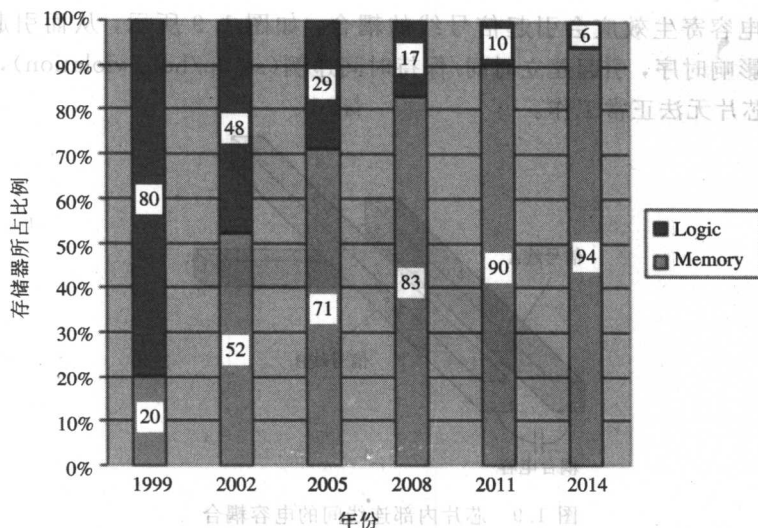


图 1.7 存储器占整个芯片面积的比例

存储器对于整个芯片设计至关重要。这是因为：

- 在许多高速 IC 设计中，存储器的带宽已成为性能的瓶颈；

- 存储器可能占据芯片 70% 以上的面积；

- 存储器消耗了芯片的大部分功耗。

嵌入式存储器的基本结构如图 1.8 所示。它主要包括三部分：位单元阵列；行/列译码器；敏感放大器 (sense amplifier)。

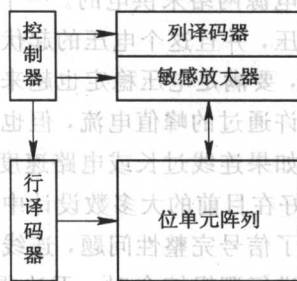


图 1.8 嵌入式存储器的基本结构

位单元阵列由许许多多的位单元组成。这种规

则的位单元阵列可以有效地节省面积。位单元的实现在《IC 设计基础》(任艳颖等编著，西安电子科技大学出版社出版)一书中进行了说明，这里不再详述。

说明：在大部分设计中，我们无需自己设计存储器。只需根据需要(深度、速度、面积、功耗等)使用 memory compiler，将别人提供的存储器单元拼起来即可。

在使用存储器时，要注意接口时序是否满足要求。时序分析可以通过仿真和静态时序分析来进行。更为准确的时序检查需要借助晶体管级的仿真/静态时序工具来完成。

### 5. 连线

连线包括芯片的信号连线、电源线、时钟线。

在一个芯片系统中，连线是最容易被人忽略的部分。实际上，它很重要，特别是在深亚微米的设计领域更是如此。可以说，如何处理连线，将是困扰 EDA 设计者与 IC 设计师的一个长久问题。

连线为什么会如此棘手呢？

根本原因在于，在深亚微米设计中，不能将连线简单地看作是理想的连接。电路理论告诉我们，金属连线具有电容、电阻和电感效应。

这些寄生效应会影响信号的完整性。具体来讲：

· 连线的电容寄生效应会引起信号线的耦合, 如图 1.9 所示, 从而引起串扰(cross talk)。串扰会影响时序, 引起建立时间/保持时间违例(setup/hold violation), 也可能会影响功能, 导致芯片无法正常工作。

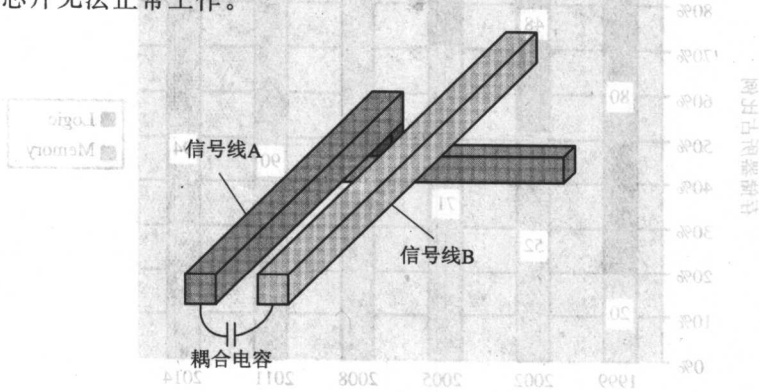


图 1.9 芯片内部连线间的电容耦合

· 连线的电阻寄生效应会导致显著的电压降, 从而影响信号电平。在一个 IC 系统中, 是通过电源网络来供电的。一个数字系统要能正常工作, 必须为它的逻辑单元提供稳定的直流电压, 并且这个电压的起伏要尽可能地小。随着芯片规模的增加, 芯片内部的电流越来越大, 要满足电压稳定也越来越困难。加宽导线可以降低电阻, 从而减小压降, 并且会增加允许通过的峰值电流, 但也会占用过多的面积。

· 如果连线过长或电路速度极高, 则导线的电感效应开始变得显著。电感效应很难处理, 幸好在目前的大多数设计中电感效应都很小, 可以忽略不计。

除了信号完整性问题, 连线还会导致设计时序难以收敛。

在进行逻辑综合时, 无法得到具体的连线信息, 因此常采用连线负载模型(wireload model)来估计连线延时。这种模型是一种统计平均得到的结果, 与实际延时有明显的误差。

众所周知, 在亚微米设计中, 连线延时在整个路径延时中占的比例很小。即使连线负载模型不准确, 对整个路径延时的影响也很小。

而在  $0.13 \mu\text{m}$  以下的工艺中, 连线延时已经与门的延时相当。这样, 如果连线上的延时估计不准确, 则对于整个路径的影响将非常大。这会导致综合的网表能满足时序要求, 但在布线后却无法实现, 这就是时序收敛问题。

在  $90 \text{ nm}$  的工艺设计中, 互连线延时甚至占到总延时的  $75\%$ 。

为了解决这个问题, 人们提出了“物理综合”的概念。这些将在后面的章节进行详述。

需要说明的是, 信号完整性问题与时序收敛问题是互相联系的。通常, 串扰会使时序更为恶化, 更难以收敛。

## 1.2 IC 系统设计概述

IC 设计实际上是在性能、面积、功耗之间进行平衡。随着 IC 系统向 SOC 演化, IC 设计的内容也越来越丰富。

在几年前，我们还可以将 IC 系统设计流程看作是简单的写代码、综合、布局布线，但对于今天的 IC 系统设计，其设计内容已经被大大拓展了。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、信号完整性分析等内容。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 1.10 所示。嵌入式软件设计属于软件设计范畴，这里不作为 IC 设计内容。

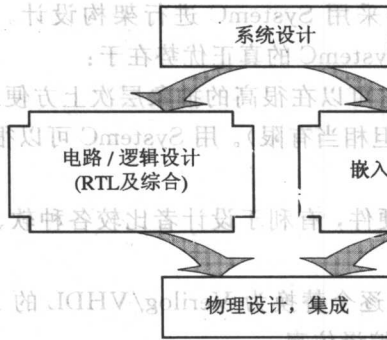


图 1.10 IC 设计的步骤

系统设计主要包括算法设计和架构设计，具体内容如下：

- 给出浮点数算法；
- 进行软、硬件划分；
- 给出 IC 设计中数据的定点表示；
- 给出定点算法；
- 结构的并行性、流水线设计；

• 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。

需要说明的是，对于深亚微米的 SOC 设计，上述开发流程不再是按“自始至终”的单一顺序进行，而是互相迭代。

### 1.2.1 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范的制定、算法设计与架构设计几个方面。

创建系统设计规范是 IC 系统设计的第一步。规范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范。如果可能,在制定规范前要多看一些同类产品的数据单(datasheet)。本书着重于规范制定之后的设计,对这方面的内容不再详述。

算法(或行为模型)可以看作一个 IC 系统的可执行规范。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:

- 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范,从而加快开发进度。

- 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。

- 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。

说明:虽然 SystemC 的语法类似于 C++,但其优势并不在于仿真速度。实际上,如果在 SystemC 设计对象中将系统中所有的时钟跳变都描述出来,则仿真速度并不比 Verilog 或 VHDL 模型快多少。

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范畴。

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。

- 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。

- 软核是可综合的 RTL 模型,可重复使用,灵活性高。

- 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。

说明:为了推动 IP 的标准化,在 1996 年成立了国际虚拟插座接口联盟(VSIA)组织。该组织已推出了一些进行 IP 设计标准化的文档。例如,它选择一些典型的、可公开的虚拟元件(VC),定义了推荐的提供内容,如测试、逻辑综合、物理设计、通信协议、总线接口等。



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 SystemC compiler)，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 Co - Centric 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。

说明：在芯片的验证中，测试平台(testbench)的开发一直是个难题。使用高级验证语言构造 testbench 要容易许多。

在第 2 章，我们将对系统级设计进行更详尽的介绍。

### 1.2.2 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范，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 1.3.1 节对仿真进行说明。

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 1.11 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。

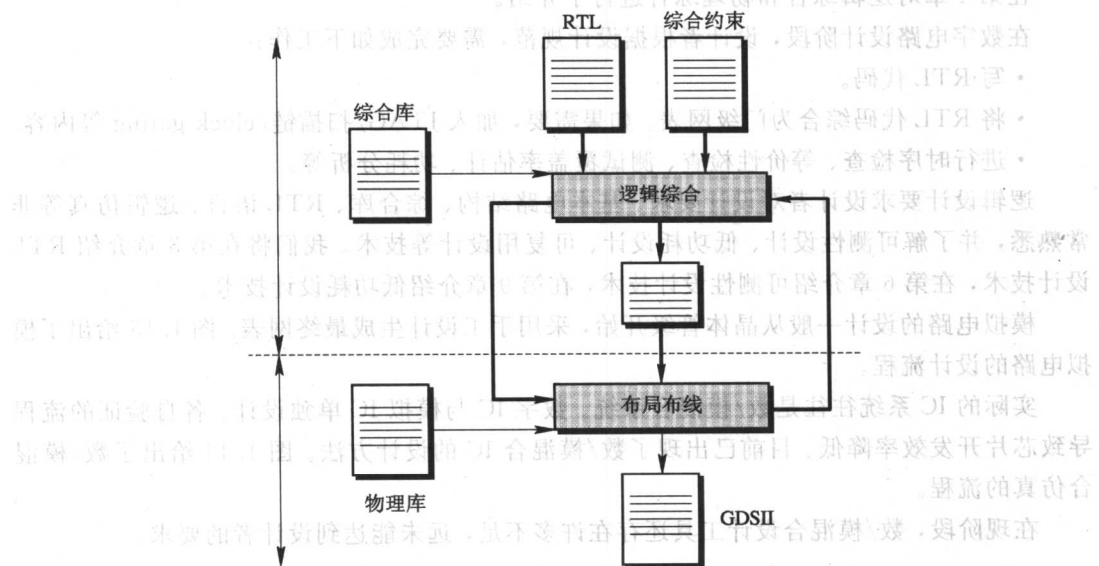


图 1.11 采用逻辑综合的步骤