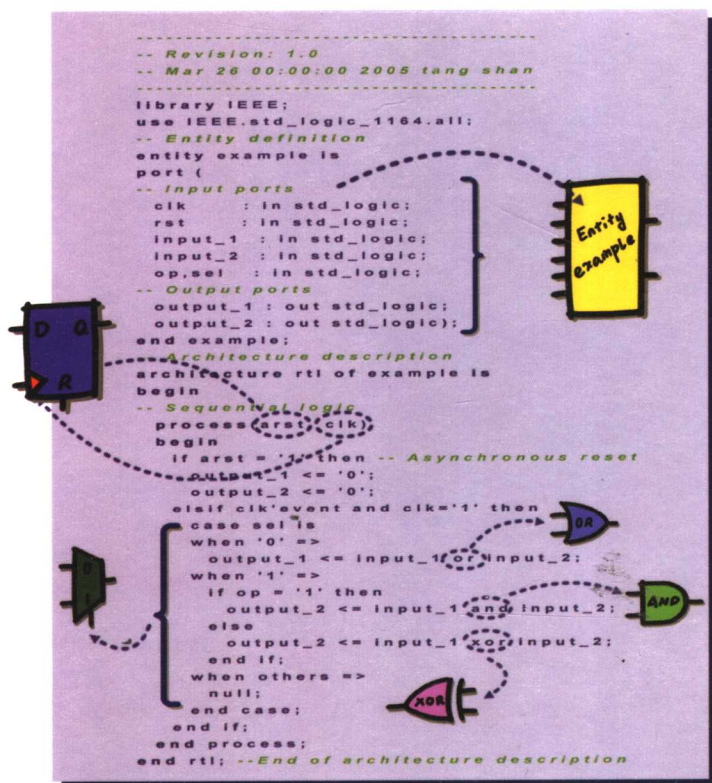


# 数字IC设计

## ——方法、技巧与实践

唐杉 徐强 王莉薇 编著



# 数字 IC 设计——方法、 技巧与实践

唐 杉 徐 强 王莉薇 编著



机械工业出版社

本书内容主要是数字芯片前端设计,不涉及模拟或是混合电路的芯片设计,而前端是指在进行物理设计(布局布线)之前的内容。

本书首先介绍了和芯片设计相关的一些背景知识。然后,使用一章的篇幅介绍芯片设计的流程和各个阶段使用的工具。之后的章节,本书就根据芯片设计的流程逐步介绍前端设计需要的知识。其中第3章为构架设计,比较详尽地介绍了构架设计的任务,一些应当考虑的问题和构架设计的方法。第4章是RTL设计与仿真。首先介绍的是一些RTL的设计规则;之后,讨论了如何在RTL设计中考虑综合和后端设计的问题;然后,给出了一些最常见的设计实例和代码;最后,介绍了仿真的相关知识。第5章为逻辑综合和相关技术。主要介绍了综合工具的功能和基本使用方法,包括基本的综合和优化的方法,以及和综合关系密切的静态时间分析和一致性检查技术。最后一章介绍了芯片设计的项目管理。

本书适于从事通信技术,电子、微电子技术领域内的数字集成电路设计及系统设计的工程师、研究人员以及有关专业师生参考。

## 图书在版编目(CIP)数据

数字 IC 设计——方法、技巧与实践/唐杉,徐强,王莉薇编著.

—北京:机械工业出版社,2006.1

ISBN 7-111-17879-3

I. 数... II. ①唐...②徐...③王... III. 数字集成电路—电路设计  
IV. TN431.2

中国版本图书馆 CIP 数据核字 (2005) 第 133599 号

机械工业出版社(北京市百万庄大街 22 号 邮政编码 100037)

责任编辑:李振标 版式设计:张世琴

责任校对:陈延翔 责任印制:石冉

北京中兴印刷有限公司印刷

2006 年 1 月第 1 版第 1 次印刷

1000mm×1400mm B5·11.375 印张·2 插页·440 千字

0 001—4 000 册

定价:36.00 元

凡购本书,如有缺页、倒页、脱页,由本社发行部调换

本社购书热线电话(010)68326294

封面无防伪标均为盗版

# 前 言

我是从 2000 年开始从事芯片设计工作的。在这之前，我只接触过一点 CPLD 的设计，所以对于芯片设计，可以说我是个门外汉。我参加的第一个芯片设计项目是一个以太网应用的芯片，规模在 30 万门。因为一直以来对芯片设计充满了好奇，所以我首先感到的是兴奋。而另一方面，做好这样一个项目是一个非常大的挑战。

当时国内的芯片设计能力还很薄弱。我工作的公司之前也只做过较小的设计，使用的还是电路图的设计方法，刚刚开始向 HDL 设计转化。对我来说，最大的困难是工作环境中没有什么经验可以借鉴。我也看了一些相关的书籍，但国内的书籍多是以介绍 VHDL 或是 Verilog 语言为主，很像是教授 C 语言或是 C++ 的方式（这一现象即使现在也没有太大的改变）。于是我开始学习 VHDL 语言，感觉和编程程序差不多，上手挺快。但在逐渐掌握了 VHDL 语言，准备开始实际设计的时候，我才发现芯片设计远不是掌握了 HDL 语言就可以完成的。相反，HDL 语言只是芯片设计者必备的知识中很小的一部分。在这之后的工作中，我遇到过各种困难，在实践中不断地犯错误，然后改正错误，积累经验。4 年多的时间，我又参加了大大小小不少项目，从 FPGA 到 ASIC 到 SoC，从编码到构架设计。当我回头看以前做过的这些项目时，发现真的走了不少弯路，而很多错误都是因为缺乏经验的指导而造成的。一个很典型的例子就是，在进行第一个设计时，我对异步时钟的处理还没有很好的概念。设计中隐藏的一个缺陷在仿真甚至 FPGA 的验证中都没有发现，最后到了产品工作的现场才出了问题，造成了很大的损失。而实际上，在这方面前人已经有了很多经验和教训，如果这些经验可以更好地传递给新的芯片设计者，那么类似的问题就不会重复地发生。

基于这种考虑，我产生了写这本书的构想。希望和我当时一样的

入门者可以从这本书里找到一些问题的答案，得到一些经验。因此，本书的目标就是解答一个进入芯片设计领域的人从开始设计到逐渐深入的过程中会遇到的各种问题，或者至少提供一些解决问题的思路。芯片设计是个非常大的话题，我和其它几位笔者的经验也非常有限。但我们希望也相信这本书可以帮助读者更快地掌握芯片设计需要的知识和技能。

本书由唐杉主编，徐强博士和王莉薇工程师参加了编写。北京邮电大学的程胜博士也给本书提出了很多宝贵的意见。

由于作者水平有限，加之芯片设计领域技术发展和更新的速度很快，书中难免有错误和疏漏之处，敬请广大读者批评指正。

唐 杉

2005 年 5 月

# 目 录

## 前言

<b>第1章 背景知识</b> .....	1
1.1 集成电路工艺、分类和设计方法的演进 .....	1
1.1.1 集成电路工艺介绍 .....	1
1.1.2 集成电路的分类 .....	2
1.1.3 集成电路设计方法的演进 .....	4
1.2 目前面临的问题和发展方向 .....	5
1.2.1 物理综合技术 .....	6
1.2.2 设计重用和 SoC 设计 .....	7
1.2.3 片上网络 (NoC: Network-on-a-Chip) .....	8
1.2.4 FPGA 动态可重构技术 .....	9
1.2.5 提高设计的抽象层次 .....	10
1.3 本书的内容和范围 .....	10
参考文献 .....	11
<b>第2章 芯片设计流程和工具</b> .....	13
2.1 需求分析和需求管理 .....	14
2.2 算法 (Algorithm) 和构架 (Architecture) 设计 .....	15
2.3 模块设计和 RTL 实现 .....	17
2.4 综合 (Synthesis) .....	18
2.5 时序验证 .....	20
2.6 原型验证 (Prototyping) .....	21
2.7 后端设计 (Back-end) .....	22
2.8 生产测试 (Manufacturing Test) .....	24
2.9 工具的作用 .....	25
参考文献 .....	27
<b>第3章 构架 (Architecture) 设计</b> .....	28
3.1 芯片构架选择和设计 .....	28

3.1.1	软硬件划分 .....	29
3.1.2	功能模块的划分和接口定义 .....	31
3.1.3	IP 选择和设计 .....	32
3.1.4	模块互连机制的选择和设计 .....	37
3.1.5	构架建模和仿真 .....	39
3.2	芯片设计的特殊考虑 .....	42
3.2.1	芯片制造商和工艺的选择 .....	42
3.2.2	设计的层次化 .....	43
3.2.3	时序闭合性设计 (Design for Timing Closure) .....	43
3.2.4	可调试性设计 (Design for Debug) .....	49
3.2.5	可测性设计 (Design for Test) .....	49
3.2.6	可验证性设计 (Design for Verification) .....	58
3.2.7	低功耗设计 .....	59
3.2.8	封装和引脚 .....	64
3.3	制定构架 (或功能) 规范 .....	64
3.4	制定功能验证计划 .....	66
3.4.1	功能验证的基本概念和方法 .....	67
3.4.2	随机测试 (Random Test) .....	70
3.4.3	衡量功能验证的质量 .....	70
	参考文献 .....	71
<b>第 4 章 RTL 级设计和仿真 .....</b>		<b>73</b>
4.1	概念 .....	73
4.2	RTL 代码编写的规则 .....	75
4.2.1	通用规则 .....	75
4.2.2	VHDL 设计规则 .....	82
4.2.3	Verilog 设计规则 .....	82
4.2.4	使用 HDL 检查工具对 RTL 设计规则进行检查 .....	83
4.3	RTL 级设计与综合及后端设计的关系 .....	86
4.3.1	RTL 级设计的综合结果 .....	86
4.3.2	在 RTL 编码中考虑时延 .....	93
4.3.3	在 RTL 编码中考虑面积问题 .....	112
4.3.4	在 RTL 编码中考虑功耗问题 .....	115
4.3.5	在 RTL 编码中考虑可测性问题 .....	118
4.3.6	在 RTL 编码中考虑布线问题 .....	119
4.3.7	根据综合结果改进 RTL 级设计 .....	130
4.4	典型设计应用 .....	130

---

4.4.1 时钟和复位	130
4.4.2 状态机	142
4.4.3 存储单元 (RAM, Gray 码 FIFO, CAM 设计, 堆栈)	154
4.4.4 寄存器空间和 CPU 接口的设计	170
4.4.5 双向信号和内部总线	173
4.4.6 增强代码的可移植性	175
4.4.7 基本逻辑单元设计	180
4.5 RTL 级设计的仿真验证	196
4.5.1 与仿真相关的概念	196
4.5.2 仿真竞争 (Simulation Race)	209
4.5.3 仿真中的常见问题和解决	211
参考文献	218
<b>第 5 章 逻辑综合和相关技术</b>	<b>220</b>
5.1 综合 (Synthesis) 的概念和流程	220
5.1.1 逻辑综合	221
5.1.2 等效性检查 (Equivalence Check)	223
5.1.3 静态时间分析	223
5.1.4 布局布线和验证	224
5.2 使用 DC 进行综合	224
5.2.1 预备知识	226
5.2.2 准备 HDL 文件	239
5.2.3 确定设计库	244
5.2.4 DC 对设计的一些操作	246
5.2.5 定义设计的环境	250
5.2.6 定义设计约束 (Constraints)	254
5.2.7 设计优化	267
5.2.8 分析和解决设计中存在的问题	281
5.3 扫描综合	286
5.3.1 扫描替换和扫描链组装	286
5.3.2 自底向上和自顶向下的扫描插入	290
5.3.3 如何获得最好的测试结果	291
5.3.4 边界扫描 (Boundary Scan)	292
5.4 静态时间分析	293
5.4.1 PT 基础	294
5.4.2 PT 基本操作	296
5.4.3 Pre-layout 静态时间分析	298



5.4.4	Post-layout 静态时间分析 .....	300
5.4.5	静态时间分析报告 .....	302
5.5	等效性检查 .....	304
5.5.1	基本概念 .....	304
5.5.2	Formality 基础 .....	306
5.5.3	Formality 的一些关键概念 .....	307
	参考文献 .....	315
<b>第 6 章 芯片设计的项目管理 .....</b>		<b>316</b>
6.1	项目计划 .....	316
6.1.1	功能、性能、成本以及设计周期的权衡 .....	317
6.1.2	项目策划的原则 .....	317
6.1.3	项目策划的流程 .....	318
6.1.4	项目计划 (project plan) 的内容 .....	319
6.1.5	挑选项目成员 .....	322
6.2	项目控制与度量 .....	323
6.2.1	项目跟踪与控制 .....	323
6.2.2	芯片设计生产率的度量 .....	324
6.2.3	缺陷分析 .....	328
6.3	风险管理 .....	329
6.3.1	风险评估 .....	330
6.3.2	风险最小化 .....	332
6.4	数据管理 .....	334
6.4.1	数据管理规则 .....	334
6.4.2	芯片设计文档 .....	342
6.4.3	配置管理 .....	343
6.5	芯片设计的质量保证 .....	347
6.5.1	质量保证的主要功能 .....	347
6.5.2	质量保证活动的管理 .....	348
6.5.3	项目中的评审 .....	349
	参考文献 .....	353

# 第 1 章

## 背景知识

### 1.1 集成电路工艺、分类和设计方法的演进

#### 1.1.1 集成电路工艺介绍

从 1958 年集成电路 (IC: Integrated Circuit) 诞生以来, 集成电路的发展到目前为止一直遵从 Intel 公司创始人 Gordon Moore 1965 年提出的摩尔定律, 即“集成电路的集成度大约每 18 个月翻一番”。这种集成度按指数增长的能力应主要归功于晶体管 (Transistor) 的发明。理论研究人员、开发设计人员不断创造出新的技术, 制造工艺的特征尺寸 (Feature Size) 随之不断减小, 集成电路的集成度也就随之不断提高。这里所说的特征尺寸大致可以认为是不同工艺制造的最小的晶体管的长度的一半, 通常用微米 ( $\mu\text{m}$ ) 来衡量, 所以我们在谈论某一集成电路是用  $0.5\mu\text{m}$  工艺制作, 就是说它内部最小的晶体管的长度是大约  $1\mu\text{m}$ 。此外, 我们通常讲的集成电路内部的门数量一般是指相对于二输入与非门 (包含 4 个晶体管) 的等价数量。

早期的集成电路采用双极技术 (Bipolar Technology), 大多数是晶体管-晶体管逻辑 (TTL: Transistor-Transistor Logic) 或者射极耦合逻辑 (ECL: Emitter-Coupled Logic)。随着阻碍金属氧化物半导体 (MOS: Metal-oxide Semiconductor) 晶体管的大规模生产制造问题的解决, 由于 MOS 管在制造中需要较少的掩膜步骤, 功耗也比同规模的采用其它技术制造的晶体管小, 所以采用 MOS 技术的集成电路很快成为主流。MOS 技术发展过程中的一个重要进步是制作 MOS 管中的铝材料被多晶硅 (Polysilicon) 材料所代替, 这使得在同一个集成电路上面能很容易地同时制造两种类型的晶体管, n 沟道 MOS 管和 p 沟道 MOS 管, 这一技术被称为互补型金属氧化物半导体 (CMOS: Complementary Metal-Oxide Semiconductor) 技术。CMOS 技术最大的优点在于它的静态功耗非常小 (趋于零), 因此它的集成度可以非常的高。现在, 采用特征尺寸为  $0.09\mu\text{m}$  CMOS 工艺的集成电路已进入大规模生产, 利用该工艺可以生产超过 1GB 的 DDR RAM 和 3GHz 的中央处理器 (CPU) 芯片, 每片上可集成上千万个晶体管。在不远的将来,  $0.65\mu\text{m}$  以下的 CMOS 工艺将会成为高性能设计的选择。尽管 CMOS 工艺是当

前毫无争议的主流技术，同时也应注意到，双极技术并未完全消亡，由于它的速度快，匹配性好，耐压能力强，在混合信号 (Mixed-signal) 集成电路、电源器件等领域 Bipolar 和 CMOS 经常被集成到同一芯片上，称为 BiCMOS 工艺。

集成电路的集成度成指数增长的驱动力主要在于：器件越小，集成度越高，芯片成本越低，性能和稳定性也越好。然而随着集成电路的特征尺寸逐渐逼近半导体硅的物理极限，新工艺的制造成本越来越高，成品率 (Yield) 却逐渐下降，许多人怀疑摩尔定律已经走到了尽头。新的非硅材料 (例如锗、镓、砷等等) 逐渐进入了人们的视野，同时，使用 DNA 技术构建集成电路也成为微电子领域的热点。但是，微电子领域的研究人员依然相信，硅材料在集成电路中的应用还远远未达到尽头。

### 1.1.2 集成电路的分类

集成电路的分类方法很多，从工艺上分，如上节所述，有 CMOS、Bipolar 以及 BiCMOS 等等。从特性上分，有数字的、模拟的和混合信号的芯片。芯片建造在晶片 (Wafer) 上，它的制造是一个非常复杂的过程 (涉及到化学处理、光处理等)，片上电路经过连续的掩膜 (Mask) 处理生成，所谓掩膜处理是指在不同的制造阶段需要处理晶片上的不同位置，这就需要遮住其它的不需要处理的地方，称之为不同的掩膜层。根据集成电路设计中设计者和制造厂商的不同分工和不同的掩膜层设计，芯片可分为以下几类：

■ 全定制芯片 (Full-Custom IC) 的所有版图都是由设计者 (用户) 设计的，制造厂商只需将其印制在晶片上。可以认为，全定制设计开始于晶体管级 (Transistor-Level)，因此这种方法对设计者来说是最灵活的，他可以控制几乎所有的电路参数，唯一的限制是制造厂商的工艺条件。全定制的芯片可以达到最好的性能和最低的功耗，但它的设计成本也是最高的，通常需要花费更多的时间并具有更大的风险。因此，全定制设计只用于可多次重用、产量非常大或对性能/功耗要求非常苛刻的设计。目前来说，只有高性能 CPU 的部分电路、标准单元库和一些特殊应用 (如高电压器件) 是采用全定制设计的。

■ 基于标准单元的芯片 (Standard-Cell-Based IC) 是目前使用较多的一种半定制芯片。这种设计的特点是生产厂家提供给设计者标准单元库，包括各种门电路、触发器、RAM，甚至较大的功能模块，用户可以灵活地将它们组合起来实现自己需要的功能。这些标准单元库是预先使用全定制的方法设计，并经过验证和优化的。由于这些标准单元按照大小排成等高的行，物理设计可以完全自动化，因此设计者可以在相对较短的时间内 (相比较于全定制设计) 实现功能比较强大的电路，但是设计的灵活性取决于生产厂商提供的标准单元库，即使对于相同的工艺、不同的标准单元库的性能可能会有很大差别，因此设计者要在系统设

计阶段认真考虑好选用哪种库。由于不同的基于标准单元的芯片采用不同的标准单元器件实现,因此它仍然需要进行所有的掩膜层设计,它的制造周期较长(大约两个月),制造成本也较高。

■ 基于门阵列的芯片(Gate-Array-Based IC)也属于半定制的芯片。在生产厂商提供的基片上已经完成生产所用的基本阵列(Base Array),通常由一种或多种特定的基本单元(Base Cell)构成,需要定制的掩膜层只有布线层。例如,由于理论上所有的数字电路都可由二输入与非门构成,一种最简单的门阵列就是集成电路上的所有逻辑单元都是二输入与非门(称作“门海”),不同的设计通过综合工具映射到不同的二输入与非门连接关系产生最终电路。由于使用门海实现的电路效率比较低,我们通常使用的是嵌入式门阵列(Embedded Gate-Array),它也是目前比较热门的所谓结构化的专用集成电路(Structured ASIC)的一种,即在生产厂商提供的基片上已经嵌入可能使用的一些逻辑功能块,如处理器、RAM、PLL、DLL、数/模、模/数转换单元,甚至连接好的时钟树等等。需要注意的是,由于这些逻辑块已经制造完成,即使最终设计者不需要这些逻辑单元,它们也只能被浪费了。例如,如果厂家提供的基片上提供32k字节SRAM,而用户只需要16k字节,另外16k字节仍然存在于最终的集成电路上,因此相比较于基于基本单元的设计,基于门阵列的设计不够灵活,门的利用率也较低,它的优势是生产周期短(两周以内),而且由于生产厂商储备了大量的已经制造好的晶片可以用于多个不同设计,制造成本也较低。

■ PLD(Programmable Logic Devices)属于“可编程”的芯片,没有需要定制的掩膜层,用户只需使用EDA软件将设计转换成“程序”烧写或下载到器件中即可工作。目前使用的PLD器件分为FPGA现场可编程门阵列(Field Programmable Garry Array)和CPLD复杂可编程逻辑器件(Complex Programmable Logic Device),它们的区别在于,FPGA基于SRAM技术,结构灵活,但逻辑无法保持,需要在每次上电时写入或者在线写入。组合逻辑通过查找表(LUT)实现,然后通过层次化的金属线互连。LUT和连线带来较大(且不固定)的延时。FPGA资源丰富,富含触发器和存储器等资源。可以支持很大规模的设计(百万门级)。但容量较大的FPGA一般价格昂贵。FPGA的供应商一般还提供丰富的IP资源。另外还有一种基于反熔丝(Antifuse)连接的FPGA,只能一次编程,但体积小、集成度高、速度高、易加密、抗干扰、耐高温。CPLD基于Flash技术,逻辑写入后可以保持。逻辑通过多路复用器(MUX)和或门实现,然后驱动触发器或直接输出。由于逻辑结构简单,实现逻辑的延时和功耗都相对于FPGA较低。但资源较少,通常只有数十到数百个触发器,单片的价格很低。使用PLD的优点是大大缩短了设计实现的时间,简化了设计的难度,缺点是单片成本高,设计的性能受PLD的限制,很难实现高性能或有特殊要求的设计。

为了应付这些挑战，一些 FPGA 厂商也推出了所谓嵌入式 FPGA (Embedded FPGA)，即在 FPGA 嵌入预先制造好的逻辑功能模块。它同嵌入式门阵列一样，也属于结构化 ASIC 的一种。

芯片设计者必须在系统设计时决定使用哪种类型的设计方法 (如图 1-1)，对于除 PLD 外的其余三种方法还要决定使用哪种类型的工艺 (如  $0.35\mu\text{m}$  或  $0.13\mu\text{m}$  CMOS 等等)，这主要取决于系统性能、开发周期和单片成本三方面的权衡，对于全定制设计或较小尺寸的基于标准单元的设计，最终生成的集成电路性能较高，面积较小，单个晶片可制造的成品多，因此单片制造成本也较低，但与此同时它的开发成本高，分摊到单个芯片的设计费用较高，因此当芯片的产量很大或系统要求的性能较高时 (例如 CPU)，设计者常采用全定制设计或较小尺寸的基于标准单元的设计。与之相反，基于门阵列或 PLD 的设计门使用率较低，因此实现同样的设计面积较大，单片制造成本高，但设计周期短，设计成本低，分摊到单片的设计成本低，如果所需要生产的芯片数量不多，性能要求并不是很高 (如某些通信系统的控制芯片)，通常采用基于门阵列或 PLD 的设计。

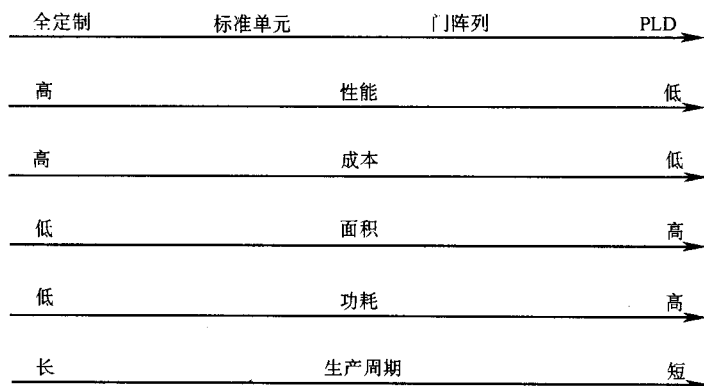


图 1-1 不同芯片类型的比较

需要指出的是，我们通常会使用专用集成电路 (ASIC: Application Specific Integrated Circuit) 这个词来泛指需要在芯片生产厂家制造的集成电路，即前面的 3 种方式。如无特别指出，本书后面都使用“芯片”这个名称泛指所有的集成电路类型。

### 1.1.3 集成电路设计方法的演进

集成电路设计方法经历了手工设计、计算机辅助设计 (CAD: Computer Aided Design)、电子设计自动化 (EDA: Electronic Design Automation) 设计等几个阶段。在手工设计阶段，设计是根据实际电子系统的设计要求，先进行功能

划分, 然后对每个电路画出真值表, 用卡诺图方法进行逻辑化简, 写出化简后的逻辑表达式, 据此画出与逻辑表达式相对应的逻辑线路图, 再按照逻辑线路图选择元器件, 搭制电路板, 最后进行实测与调试。如果是进行集成电路的设计, 就必须手工完成版图绘制、红膜的刻制等一系列复杂的手工操作。在手工设计时期, 系统的设计、调试都十分困难, 设计的测试需要等到实际的芯片生产出来之后才能够进行, 在这之前, 对于实际设计的功能和时序以及物理特征根本没有办法了解。

随着计算机辅助设计技术的出现, 传统的手工设计电子系统的方法走向了灭亡, 人们在设计过程中更多地借助于计算机进行辅助设计, 完成图形(逻辑图、版图等)输入与编辑工作, 在设计过程中, 各种软件开发环境还可以提供各个阶段的模拟仿真手段。对于集成电路版图设计而言, 还可以进行设计过程的集合规则检查(DRC: Design Rule Check)、电学规则检查(ERC: Electrical Rule Check)。在后期还可进一步实现自动参数提取, 版图/逻辑图对比(LVS: Logic Versus Schematic)验证等一系列辅助手段。与传统的纯手工设计相比, CAD技术的出现使集成电路设计技术向前迈进了一大步。

到了20世纪80年代后期, CAD技术日益成熟, 从最初的单纯作为辅助设计手段, 逐渐发展成为可以提供各种自动综合(Synthesis)工具, 如逻辑综合、版图综合、测试综合等, 从而真正跨入了电子设计自动化阶段。这一阶段, 硬件描述语言(HDL: Hardware Description Language)能够实现从抽象的行为与功能描述到具体的内部线路结构描述, 国际标准化组织IEEE定义了一系列的标准来完成各个设计阶段图形和语言到实际内部线路的映射。到了20世纪90年代末, 物理综合、高级验证语言等技术不断涌现, 使我们暂时还能够应对日益增加的设计复杂性。然而, 由于EDA业界的发展速度落后于CMOS工艺的发展速度, 集成电路设计方法学以及EDA工具的发展还需要有更重大的突破, 更大的挑战还在前方。

## 1.2 目前面临的问题和发展方向

从前面的介绍可以看出, 集成电路的特征尺寸不断减小, 设计的复杂性也随之不断提高。当集成电路工艺的特征尺寸小于 $0.35\mu\text{m}$ 时, 我们称之为深亚微米(DSM: Deep Sub-micron)工艺。由于深亚微米工艺中, 互连线(Interconnect)的延时超过了逻辑门延时, 传统的逻辑门主导的(Gate-dominated)设计方法将转为互连线主导(Interconnect-dominated)的设计方法, 它对集成电路设计及其设计方法学提出了新的问题和挑战, 举例来说, 由于集成电路工作频率的提高、时钟周期的缩短, 使得允许的时延变小, 例如如果芯片工作在 $1\text{GHz}$ , 那么一个

时钟周期只有 1 ns，两个触发器之间的组合逻辑的延时就要小于 1 ns。随着互连线延时的增大，传输延时在整个逻辑延时中的影响就变大，按照原有的 EDA 设计流程，每进行一次逻辑优化，都需要重新进行版图综合，而新的版图综合结果又产生不同的延时分布，每进行一次逻辑优化，都需重新进行版图综合，这导致逻辑设计阶段的布线和版图设计阶段的布线很可能无法“收敛 (Closure)”，所以，传统的芯片设计流程中对于逻辑设计阶段和版图设计阶段的划分不再适用，在设计前端不得不开始考虑后端布局布线的影响，许多 EDA 厂商推出的物理综合工具就是为了解决这一问题。

此外，深亚微米设计带来的另一个问题是，由于深亚微米电路中线条间距的缩短和时钟频率的提高，必须开始考虑电路的感性负载，随之而来的信号线之间可能发生串扰，使信号的可靠性受到挑战。尽管有许多潜在的困难，深亚微米电路设计带来的益处使得全世界的研究人员正在向多个方向努力来解决这方面问题，随着系统的集成度的持续增加，特征尺寸的持续减小，在不远的将来很可能使设计方法还会有很大的改变。

下面讨论一些当前正在逐步使用的设计方法和今后的发展方向。

### 1.2.1 物理综合技术

传统的综合方法和布局布线是分离的，在综合过程中使用线载模型来预测布局布线后的延时。这一模型是一种统计的结果，在连线延时和门延时相比很小的时候是足够准确的。但是在深亚微米的环境中，最显著的一个特点就是连线延时超过了门延时。器件在芯片中的实际布局会大大影响芯片的时间特性。线载模型已经不能准确地反映布局布线后的延时情况了 (如图 1-2)。

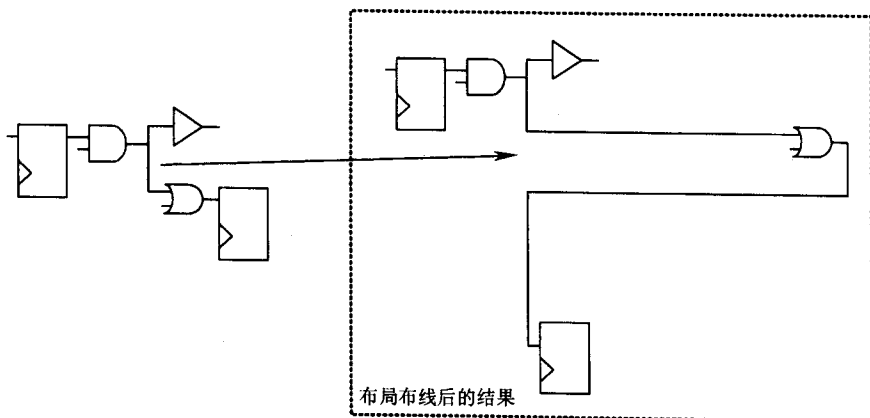


图 1-2 布局对延时的影响

在图中，左边是综合后的电路，而右边是经过布局布线以后的实际结果。可以看出，布局布线之后，由于器件布局的关系，有的连线会和综合后的假设差别很大，造成延时的差别很大。因此传统的综合与布局分离的综合方式已经不能满足深亚微米工艺的要求。而物理综合就是在这种情况下提出的新的综合方式。简单地说，物理综合就是在原有的设计流程中更早地考虑物理设计因素，改善综合的效果，保证实现时序闭合（Timing Closure，指满足设计的时间要求）。目前的物理综合技术主要是将布局和综合结合起来，在综合中根据布局信息估计延时的情况，这样就有了更准确的延时信息来驱动综合的过程。

### 1.2.2 设计重用和 SoC 设计

设计重用并非新概念，事实上前面讲述的基于标准单元的设计就是设计重用的绝佳例子，标准单元库经过单独的设计和优化后被许多设计反复使用。目前的趋势是 ASIC 设计越来越多地采用预先验证好的越来越大的模块来加速设计周期，如微处理器、存储器、视频编解码器（Codec）、发送接收器（Transceiver）等，这些预先设计好并通过了验证的模块被称为“内核”（Core）或者“知识产权”（IP: Intellectual Property）。

深亚微米技术的发展使单个芯片上集成整个系统成为可能，我们称之为片上系统（SoC: System-on-a-Chip），如图 1-3 所示。比较 PCB 板级系统，片上系统

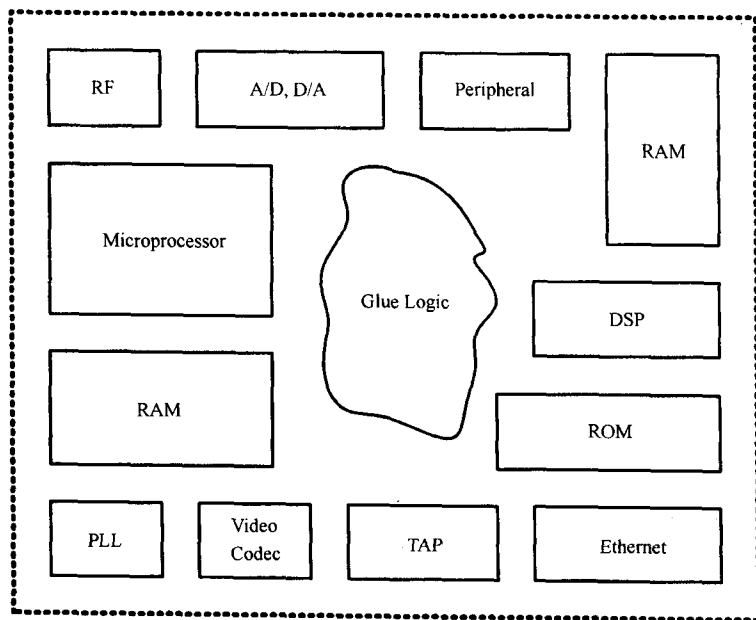


图 1-3 片上系统基本结构



的成本更低、性能更好、功耗更小、稳定性更高，此外对于一些便携式应用，采用 SoC 设计可以减小系统的体积和重量。因此，SoC 设计得以迅速普及，在目前的集成电路设计中占有主导地位。与此同时，SoC 设计相对于传统的电路设计也带来了设计方法上的变革，新的 SoC 越来越倾向于可重构性、高速自动化的设计方法，传统的功能设计方法逐渐转向功能组装的设计方法，这就要求设计中更加关注系统的规划、设计和验证，而现在对于系统的规划初期的软硬件协同验证方法的研究还很少，使得很多本可以在设计初期就发现的系统规划失误不能尽早被发现，需要等到系统实现的末期再返回，这样迭代的次数就有可能过多，不仅导致系统开发周期过长，而且可能这种迭代无法收敛，导致整个系统设计失败。此外，SoC 中嵌入式微处理器的引入，也使得嵌入式软件的开发验证成为 SoC 设计中无法回避的话题，传统的嵌入式软件开发环境只是针对板级的开发，当涉及到类似嵌入式存储器的设计时，已经不太适用了，这就要求开发新的适应 SoC 设计的嵌入式软件开发环境。SoC 设计的发展还要求现场可编程逻辑器件的厂商不断完善、扩充 IP 库，目前可重用的 IP 模块数量和质量还不够，原有的一些 IP 模块往往是针对某一个具体的应用，在应用到其它方面时，往往需要对 IP 模块进行修改，这已经失去了开发 IP 模块的最初用意，开发可重用的 IP 模块也是 SoC 设计的基础。总之，SoC 设计的概念虽然已经提出了很长的一段时间，但是，在它们的发展过程中，还存在很多如上所述的问题，只有这些问题得到解决，才可能使更多的 SoC 厂商参与进来，才能真正实现 SoC 设计的蓬勃发展。

### 1.2.3 片上网络 (NoC: Network-on-a-Chip)

随着芯片制造工艺的不断发展，在一颗芯片内集成的内核数量越来越多，在实现了强大功能的同时，同时也引入了许多新的设计问题。其中之一就是如何实现内核间的互连。首先，如何实现一个稳定可靠的内核互连机制？这种互连机制会在很大程度上影响 SoC 系统的性能甚至是功耗，简单的总线结构已无法满足 SoC 系统中的越来越复杂的通信需求。

一个由多个内核组成的系统如果使用总线连接，则会需要很长的连线，而这样的总线要达到很高速度是不可能的。另外，芯片的同步越来越困难，按照传统的方式使用单一时钟已经变得几乎是不可能的工作。所以，未来的 SoC 最有可能采用一个全局异步局部同步的工作方式 (Globally Asynchronous and Locally Synchronous)，也就是说芯片将不再使用一个统一的时钟，而是采取一种分布式的方式，而只是在局部（如一个内核内部）使用传统的单一时钟。这样，各个内核间的通信方式将变成异步的方式。最后，由于工艺尺寸的缩小和 SoC 复杂度的提高，加之工作电压的降低，使得电信号在导线上的传输不再是完全确定和可靠的，而是会有一定错误的概率。这就使得 SoC 的设计和分析必须基于确定性