



电子信息与电气学科规划教材 · 电子信息科学与工程类专业

FPGA/ASIC

高性能数字系统设计

李洪革 编著



电子工业出版社

PUBLISHING HOUSE OF ELECTRONICS INDUSTRY

<http://www.phei.com.cn>

电子信息与电气学科规划教材·电子信息科学与工程类专业

FPGA/ASIC 高性能 数字系统设计

李洪革 编著

電子工業出版社

Publishing House of Electronics Industry

北京·BEIJING

本书是高性能数字集成系统设计的基础教材，作者从硬件描述语言 Verilog HDL 描述入手，重点阐述了高性能集成化数字电路的电路结构、面积优化、时序优化、速度优化、功耗优化和可重构设计等问题。本书还给出了复杂数字系统的两种实现方案 FPGA/ASIC 的具体实现方案。全书共分 11 章，主要包含复杂数字系统设计问题前瞻、Verilog HDL 语言基础、电路结构优化、状态机及数据路径、时序/时钟域、低功耗、可重构设计及其具体 FPGA/ASIC 设计实现方法。本书通过大量设计实例讨论高性能设计思想和方法，同时，针对当前工业界人士的问题和需求，有的放矢地分析和解释了相关具体设计案例。

本书可作为普通高等院校、科研院所电子信息、通信工程、电气工程、计算机等相关专业的本科生和研究生教材，还可作为数字集成系统领域工程技术人员的参考书。

未经许可，不得以任何方式复制或抄袭本书之部分或全部内容
版权所有·侵权必究

图书在版编目(CIP)数据

FPGA/ASIC 高性能数字系统设计 / 李洪革编著. —北京: 电子工业出版社, 2011.1
(电子信息与电气学科规划教材·电子信息科学与工程专业)
ISBN 978-7-121-12070-1

I. ①F… II. ①李… III. ①可编程逻辑器件—数字系统—系统设计—高等学校—教材②集成电路—电路设计—高等学校—教材 IV. ①TP332.1②TN402

中国版本图书馆 CIP 数据核字(2010)第 205917 号

策划编辑: 段丹辉

责任编辑: 段丹辉

特约编辑: 王 纲

印 刷: 北京市李史山胶印厂

装 订:

出版发行: 电子工业出版社

北京市海淀区万寿路 173 信箱 邮编 100036

开 本: 787×1092 1/16 印张: 21.25 字数: 628 千字

印 次: 2011 年 1 月第 1 次印刷

印 数: 4000 册 定价: 36.00 元

凡所购买电子工业出版社图书有缺损问题，请向购买书店调换。若书店售缺，请与本社发行部联系，联系及邮购电话: (010)88254888。

质量投诉请发邮件至 zlts@phei.com.cn，盗版侵权举报请发邮件至 dbqq@phei.com.cn。

服务热线: (010)88258888。

前 言

电子信息科学是世界各国争相探索和研究的的前沿科学领域,是引领世界科技发展的重要源动力。电子信息科学的发展强烈地依赖于微电子、信息、通信和计算机技术等,其中随着集成化制造技术的迅猛发展,电子技术的核心——集成电路已经成为电子信息科学高速发展的基石。2000年,国务院18号文件《鼓励软件产业和集成电路产业发展的若干政策》的出台极大地鼓励和推动了我国信息产业的发展。2006年,国家中长期科学和技术发展规划纲要中公布了16项重大科技专项,其中前两项就是集成电路设计和制造专项课题。由此可见,集成化技术在科学发展中的地位和作用。

近年来,随着国家对电子信息产业的大力扶持和推进,也随着读者对相关图书的需求的激增,国内有关数字系统集成化设计的书籍也成为炙手可热的信息来源。目前,市场中相关的图书可以归纳为以下三类:(1)语言介绍类,主要以讲述硬件描述语言为主,包含 Verilog HDL 和 VHDL。(2)工具实现类,以介绍 FPGA 的应用实现方法为主,主要是对 Xilinx 产品和 Altera 产品的介绍等。(3)代码描述类,主要以讲述数字系统中的模块代码的编写。部分图书综合了上述三个方面。这些图书既包括国外引进的版本,也包括国内原创的版本。由于硬件描述语言与传统的高级程序语言所实现的最终目标存在本质区别,硬件设计人员不能仅仅考虑所描述数字系统的逻辑功能,更重要的是要考虑所实现集成化系统的物理性能。高级数字电路/系统的物理性能通常包含电路面积、功耗、速度、时滞和吞吐率等多方面因素。而国内基于 FPGA/ASIC 集成化数字系统的性能优化与设计的图书资源却寥寥无几,这必将制约数字系统集成化设计水平的提升。

为弥补上述图书资源的不足,本书以讲解 Verilog HDL 语言为基础,并较深入地分析数字系统的硬件语言的实现方法,从集成化的视角重点讲述系统的结构、时序、速度、面积和功耗等物理性能的设计优化,通过对上述多重物理性能的折中分析,实现 Verilog HDL 描述方法的高级数字系统的设计方案。本书的具体内容如下。

第1章概述了电子信息社会条件下,高级复杂数字系统设计所面临的集成化挑战、设计流程、发展趋势和未来的应用前景等。

第2章介绍 Verilog 硬件描述语言,其中包括基本结构、模块与声明、数据类型与运算符、行为建模和 Verilog-2001 设计规则等几部分。

第3章讨论了基于 Verilog 语言的数字系统集成化设计的性能优化。其中包括电路面积的优化、速度的优化、模块间接口的设计、复位信号与结构的优化等。本章介绍了在相同逻辑功能实现的前提下,不同 Verilog 描述方法对应的不同物理结构,从而对面积、速度、功耗带来不同的结果。

第4章描述了复杂数字系统设计常用的运算单元和结构。其中包括逻辑电路的数值计算、加法器、乘法器、数字信号处理等内容。

第5章讨论了数字系统中重点的问题——状态机与数据路径。本章介绍了状态机的概念、分类、HDL 描述法、编码风格、可综合化、性能优化等方面;描述了数据路径中时间调度与分配和相应集成化设计实例。

第6章介绍了数字电路中重要的概念——时序。随着数字系统计算速度的提升,电路时序要求日趋严峻。为此描述了时序参数定义、时钟抖动与偏差、时钟分布、延迟时间等问题,进而讨论了同步与异步逻辑电路的概念和设计法。

第7章介绍了集成电路设计中的重点问题——低功耗。7.1节描述了功耗的种类、定义等概念；7.2节讨论了低功耗设计的方法，如系统级、算法级、结构级、电路级和器件级等。

第8章介绍了可重构电路FPGA和可重构计算。其中包括可重构器件的现状和分类、FPGA电路结构、可重构系统等。

第9章通过AES加解密系统和基带调制解调系统设计实例，讲解了复杂数字系统设计的方法。

第10章介绍了集成化数字系统设计的方法之一：FPGA设计实现方法。

第11章介绍了集成化数字系统设计的方法之二：ASIC设计实现方法。

本书由李洪革构思、执笔完成，全书凝结了作者十余年集成化数字系统设计的工作经验并吸收、总结了多位学者的最新研究成果。参与本书编写的人员有邓征、杨奇桦、徐启成、于攀，多名研究生参与了书稿的校对和文字的复查工作。在该书的编写过程中，得到了多方面的支持与帮助。国家集成电路人才培养基地——北京航空航天大学电子信息工程学院的领导和师生一直对本书给予了大力的支持和帮助，电子工业出版社对本书的出版提供了直接而热情的帮助。在此谨向所有为本书的编写、出版给予鼓励和帮助的社会各界表示最衷心的感谢！

为方便教师教学，本书配有免费电子教学课件，可以登录华信教育资源网(<http://www.hxedu.com.cn>)注册下载或发送电子邮件至 duandh@phei.com.cn 索取。

尽管作者对书稿进行了多次严密的修改和推敲，但由于集成化系统设计的先进性和快速发展的特点，且作者学识所限，书中的错误和不当之处在所难免，恳请使用本书的师生和社会各界人士给予批评、指正。作者的邮箱：honggeli@buaa.edu.cn。

编者

2010年6月

目 录

第1章	FPGA/ASIC 设计方法概述	(1)
1.1	电子系统发展历史	(1)
1.2	高性能集成化设计	(2)
1.3	数字集成化设计流程	(3)
1.4	数字系统实现方法	(5)
1.5	集成化设计发展趋势	(8)
1.6	集成设计应用前景	(10)
	习题	(11)
	参考文献	(11)
第2章	Verilog 硬件描述语言	(12)
2.1	基本概念	(12)
2.2	Verilog HDL 基本结构	(13)
2.3	模块与声明	(14)
2.3.1	模块命名	(14)
2.3.2	信号命名	(15)
2.3.3	端口声明	(16)
2.3.4	变量声明	(16)
2.3.5	'include 与define	(16)
2.3.6	代码编写规范	(16)
2.4	数据类型与运算符	(17)
2.4.1	数字声明	(17)
2.4.2	数值逻辑	(18)
2.4.3	常量数据类型	(18)
2.4.4	数据类型	(19)
2.4.5	运算符和表达式	(21)
2.5	行为建模	(22)
2.5.1	行为描述模块	(22)
2.5.2	条件语句	(27)
2.5.3	循环语句	(28)
2.5.4	任务与函数	(30)
2.5.5	混合设计模式	(31)
2.5.6	测试激励	(32)
2.6	Verilog—2001 设计规则	(36)
2.7	Verilog 基本模块	(40)

2.7.1 组合逻辑	(40)
2.7.2 时序逻辑	(44)
2.8 本章小结	(47)
习题	(48)
参考文献	(48)
第3章 高性能电路设计	(49)
3.1 电路面积优化	(50)
3.1.1 代码编写优化	(50)
3.1.2 条件语句处理	(51)
3.1.3 资源共享	(53)
3.1.4 时序电路的优化	(58)
3.2 高速电路设计	(62)
3.2.1 逻辑与结构	(62)
3.2.2 关键路径	(65)
3.2.3 迟置信号处理	(66)
3.2.4 流水线设计	(69)
3.3 模块接口设计	(74)
3.3.1 数据流	(74)
3.3.2 模块间的协议传输	(75)
3.4 复位信号与毛刺消除	(79)
3.4.1 复位信号	(79)
3.4.2 毛刺消除	(84)
习题	(86)
参考文献	(88)
第4章 运算单元与结构	(89)
4.1 数值计算	(89)
4.2 加法器	(91)
4.2.1 加法器	(91)
4.2.2 超前进位加法器	(92)
4.2.3 进位旁路加法器	(95)
4.2.4 进位选择加法器	(98)
4.3 乘法器	(99)
4.3.1 阵列乘法器	(100)
4.3.2 高速乘法器	(104)
4.4 数字信号处理	(113)
4.4.1 有限冲激响应滤波器	(114)
4.4.2 无限冲激响应滤波器	(118)
4.4.3 脉动阵列	(118)
4.5 有限域 $GF(2^n)$ 运算	(121)
4.5.1 定义	(121)

4.5.2 有限域多项式	(122)
习题	(124)
参考文献	(124)
第5章 状态机与数据路径	(125)
5.1 有限状态机	(125)
5.1.1 基本概念	(125)
5.1.2 状态机分类	(126)
5.1.3 状态机描述方法	(132)
5.1.4 状态机的编码风格	(143)
5.1.5 可综合的 FSM 编码	(149)
5.1.6 状态机的优化	(151)
5.1.7 状态机容错和设计准则	(152)
5.2 数据路径	(154)
5.2.1 概述	(154)
5.2.2 时间调度与分配	(155)
5.2.3 数据路径设计实例	(161)
习题	(165)
参考文献	(166)
第6章 时序与时钟	(167)
6.1 时序电路	(167)
6.1.1 基本概念	(167)
6.1.2 稳态与亚稳态	(168)
6.1.3 时钟信号	(169)
6.1.4 时钟分布	(172)
6.1.5 电路延时	(174)
6.2 时钟域	(174)
6.2.1 同步与异步	(174)
6.2.2 异步电路通信	(176)
6.2.3 多时钟域复位问题	(190)
习题	(193)
参考文献	(194)
第7章 低功耗设计	(195)
7.1 基本原理	(196)
7.1.1 动态开关功耗	(196)
7.1.2 短路功耗	(197)
7.1.3 静态功率消耗	(198)
7.2 低功耗设计方法	(199)
7.2.1 系统级低功耗法	(200)
7.2.2 算法级低功耗法	(200)

7.2.3	结构级低功耗法	(201)
7.2.4	电路级低功耗法	(203)
7.2.5	泄漏功耗消减法	(210)
	习题	(212)
	参考文献	(212)
第 8 章	FPGA 与可重构计算	(213)
8.1	可重构器件	(213)
8.1.1	可重构器件现状	(213)
8.1.2	可重构的分类	(214)
8.2	可重构电路结构	(215)
8.2.1	FPGA 电路结构	(215)
8.2.2	动态可重构系统	(229)
8.2.3	专用可重构系统	(232)
	参考文献	(238)
第 9 章	数字系统设计实例	(239)
9.1	AES 加解密系统	(239)
9.1.1	AES 算法概述	(239)
9.1.2	AES 算法结构	(239)
9.1.3	芯片内部电路系统架构	(241)
9.1.4	芯片硬件描述	(244)
9.2	通信基带系统	(249)
9.2.1	无线通信系统	(249)
9.2.2	RFID 基带设计	(250)
	参考文献	(264)
第 10 章	FPGA 设计方法	(265)
10.1	新建工程	(265)
10.2	新建代码	(266)
10.3	代码仿真	(269)
10.4	IP Core 的使用	(276)
10.5	逻辑综合	(279)
10.6	配置实现	(281)
10.7	分析、报告	(285)
10.8	测试	(286)
	参考文献	(292)
第 11 章	ASIC 设计方法	(293)
11.1	ASIC 定义及设计流程	(293)
11.2	逻辑综合	(295)
11.2.1	逻辑综合工具 Design Compiler 介绍	(295)
11.2.2	DC 基本概念	(297)

11.2.3	DC 设计流程	(299)
11.3	布局布线物理设计	(311)
11.3.1	FloorPlan	(314)
11.3.2	Timing Setup	(318)
11.3.3	Placement	(318)
11.3.5	CTS	(318)
11.3.5	Route	(321)
11.3.6	DFM	(324)
11.4	版图验证、修正	(327)
11.4.1	版图验证	(328)
11.4.2	后仿真	(328)
11.4.3	流片	(328)
	参考文献	(328)

第1章 FPGA/ASIC 设计方法概述

电子信息系统已经是现实生活中不可缺少的一部分,无论是消费类电子产品、工业类电子产品或宇航级电子设备,都是基于硅器件条件下完成各种复杂信息处理的,而电子系统的设计和结构决定了其系统的性能。随着对信息处理速度和性能的苛刻要求,各种复杂电子信息系统越来越多由基于FPGA/ASIC的数字集成电路来完成,数字集成系统的设计质量在整个信息系统中起着至关重要的作用。

1.1 电子系统发展历史

电子系统经过几十年的历程,已经从初期的分立器件向功能集成化、系统集成的方向发展了。民用家电、航空电子系统的进步真实地再现了电子信息技术发展的过程。在20世纪50年代,电子系统都是以分立器件为核心而组建的,航空电子系统也处于分分离散结构。此时整体电子系统无中心控制计算机,每个子系统都有各自的传感器、控制器、显示器和传输链路。这时系统的集成化简单、复杂度高,因此,故障率高、体积庞大、性能有限,难以实现大量信息数据处理和交互,使系统的维护和提升都受到了严格的限制。到70年代,集成电路在电子系统中已经占有一席之地。为提升航空电子系统的功能,美国WRIGHT实验室提出了数字式航空电子信息系统(DAIS)计划,通过大量使用LSI芯片实现航空电子系统控制单元与显示单元的数字综合化。进入80年代后,在微电子技术的支撑下,微处理器、数字化通信系统也不断发展,此时,航空电子系统进入了联合式结构的时代。航空电子系统完成了全部数字化,系统在高效计算机的管理下,通过1553B数据总线将飞行器的各独立功能子系统构成分布式集中控制的网络结构,实现了信息交互和系统综合。90年代前后,微电子技术已经发展到了超大规模集成的阶段,高集成度ASIC芯片的出现,大大提高了信息处理的能力,而且减小了系统的质量和能耗,提高了可靠性。在这一背景下,美国空军实验室又提出了宝石柱计划——通过集成化设计技术实现系统各单元接口高速时分多路传输总线。在超高速集成电路和超高速总线的支持下,完成系统的资源共享、互相备份等高可靠处理。为进一步提高系统的实时性,系统综合扩展到信息采集的前端——RF电路,这样就完成了航空电子系统各单元的综合化与集成化。

20世纪80年代前,尽管集成电路也取得了飞快发展,然而,当时的集成电路设计工程师只能采用代工厂提供的专用电路图来进行手工设计。对于相对复杂的数字逻辑电路,设计师从原理设计、功能设计、电路设计到版图设计一般要一年以上的设计周期,其中仅仅版图布线一环,工程师就要花费数周的时间才能完成。随着大规模集成电路的研发,集成数百万逻辑门的电路变得非常复杂,而其功能的仿真也很难通过传统的面板测试法验证设计的系统,在此基础上,后端工程师开始寻找通过电子设计自动化(EDA)的方法将手工设计转变为计算机辅助。前端的工程师也希望使用一种标准的语言来进行硬件设计,以提高设计的复杂度和可靠性,基于此,硬件描述语言(Hardware Description Languages, HDL)应运而生。美国国防部制定了一套电子电路规范标准文档VHSIC(Very High Speed Integrated Circuit),通过对上述VHSIC改良的VHDL语言在1982年正式诞生。与此同时,Verilog HDL语言在1983年由Gateway设计自动化公司研发出来。而后,这两种语言分别被IEEE批准为大规模数字电路设计的标准语言,其中,Verilog HDL语言更接近于高级语言C,设计人员更容易理解和掌握。VHDL语言描述较复杂,其设计风格类似于PASCAL,其特点对系统设计更有优势。Verilog HDL的IEEE1364—2001标准与1995标准相比有显著的提高。

20 世纪 80 年代中期, 工程师已经开始普遍采用 HDL 进行数字电路的逻辑验证, 但工程师仍延续手工方法将逻辑功能设计转化为相互连接的逻辑门表示的电路图, 而手工设计大大延长了产品的研发周期。80 年代后期, Synopsys 公司开发了 Design_Compiler(简称 DC)的逻辑自动综合工具, 综合工具的诞生对数字电路的设计方法产生了巨大的影响。工程师可以使用 HDL 在寄存器传输级(Register Transfer Level, RTL)对电路进行功能描述。通过 DC 综合工具, 设计师只要说明数据在寄存器移动和处理的过程, 构成逻辑电路及其连线是由自动综合工具从 RTL 描述中抽取出来的, 而无须手工转化电路的门级网表。自动综合工具的诞生完全解放了设计师在逻辑门电路布局的手工劳动, 使设计师更专注于电路性能、结构的提升。

1.2 高性能集成化设计

集成化数字系统正在变得日趋复杂、日益灵活。复杂表现在电路的功能和结构上, 即现在的集成化系统不仅仅是 VLSI/ASIC/FPGA, 而是系统级的 SoC(SoPC)甚至 SIP/NoC 等。集成化的灵活性表现在可配置、可编程和可重构的柔性硬件系统上, CPLD/FPGA 器件和可重构计算 SoC 是主要实现手段。集成化硬件的复杂化、柔性化的前提是以集成化系统的高性能设计为基础的, 因此, 高性能的电路设计是完成复杂化、柔性化集成系统的前提。高性能数字电路/系统设计主要体现在如下指标, 如速度、吞吐率、功耗、面积、柔性和可靠性等。

尽管对于数字集成化设计拥有很多综合优化的 EDA 工具为我们提供了便利, 但任何一种高级工具的优化通常并不能满足多方面的设计约束, 设计工程师的高级设计艺术是 EDA 工具永远无法替代的。从 Intel 的高性能 CPU 处理器到联发科技的基带处理器, 从德州仪器的 ADC 到三星的 SDRAM, 集成化数字电路的设计无时无刻不凝结着设计师的智慧结晶。由此可见, 大规模成熟应用的集成电路产品都是无数设计师多年辛苦的智慧体现, 也是设计师在熟练驾驭集成电路设计艺术的基础上不断创新成果, 而非 EDA 工具使然。

数字集成化系统的性能主要反映在以下 4 个主要物理特性中: 速度、吞吐率、面积和功耗。通过上面 4 个物理特性参数的性能指标, 可以明确判断出集成化系统的性能。这些性能指标对于实际的硬件系统又是相互关联、相互制约的, 实际的硬件设计必须全盘考虑、折中优化才能实现所需求的高性能集成化硬件系统。数字系统速度, 也就是系统的时钟频率是考虑的首要因素。系统的高速化是高性能电路的重要指标, 它可以带来系统吞吐率的增加而使单位时间内处理的信息量增大。但是, 通过流水线或并行结构等技术获得的高速化一方面会导致系统面积的增大, 另一方面也增大了系统的功率消耗。因此, 高速化的系统设计必须是在一定的面积消耗和规定的功耗范围内达到最大时钟频率, 才能称其为高性能; 否则, 再高的速度也是无意义的。相关的内容将在第 3 章、第 5 章和第 6 章讲述。

数字系统设计的第 1 个关键物理特性是速度。系统的速度通常是由系统的时序(Timing)和时滞(Latency)两个因素决定的。系统的时钟速度由关键路径中最长延迟时间的时序所决定, 也就是所经过的两个相邻的时序元件之间的最大延时。为了优化电路的时序, 通常使用在关键的两个时序单元之间设置寄存器的方法, 以缩减它的延时。第二种方法是把关键路径的逻辑功能分解成大量可以并行操作的功能单元。其次, 可以改变关键路径与非关键路径的寄存器的平衡, 以减少任何两个寄存器之间可能出现的最坏延迟条件。另外, 当关键路径与非关键路径组合时, 可以使用关键路径最小化的策略以缓解关键路径中的时序矛盾的问题。时滞是被处理的数据通过处理器所消耗的时钟量。硬件设计中, 时滞的最小化设计是追求的目标。最小化时滞设计需要系统的并行结构、非流水线和缩减逻辑结构等, 这样的时滞化设计可能带来时钟速度降低等现象。

数字系统设计的第 2 个关键特性是吞吐量(Throughput), 也就是数据流量, 即每个时钟周期内处

理的数据量。吞吐量不仅与系统的时钟频率有关，更与输入/输出的位宽有关。吞吐量的增加主要是通过时钟频率和数据位宽，也就是吞吐量的增加不一定是系统时钟频率的必然结果。吞吐量的增加往往采用大位宽结构、并行处理等方法使面积消耗增加。另一种方法是打开具有迭代功能的回路增加数据流量，其代价也是成比例地增加面积。

数字系统设计的第 3 个关键物理特性是面积。集成电路面积的缩减主要是通过半导体器件工艺的不断研发而实现低面积消耗，但其代价是高昂的实现成本，且制造的复杂度也大幅攀升，使成功率降低。为此，在某种工艺条件下减小面积，首先，可以通过优化系统的算法来实现，系统算法的优化往往不能直接影响面积。其次，可以通过优化高层次的拓扑结构来实现。以数据路径而言，可以通过寄存器、运算器等功能单元的逻辑复用优化初始的拓扑结构，也就是逻辑复用地建立满足数据流的递归操作而达到使用面积大幅消减的目的。面积的消减是以牺牲数据的吞吐量为代价的。再次，设计低面积消耗可以通过代码编写的策略、综合优化的方法和布局布线优化来完成。其具体技术包括：逻辑资源共享、数据流结构、流水线结构复用、复位/置位设计方法等。具体的设计方法可以参阅第 3 章、第 5 章等的有关内容。

数字系统设计的第 4 个关键物理特性是功耗。随着集成系统复杂度的增加，单位面积集成的晶体管数量也剧增，为此，低功耗设计成为必然。ASIC 的低功耗设计采用的方法较多，可以降低电源电压、减低时钟频率和一些具体的低功耗设计方法来实现。对 FPGA 而言，由于电源电压已经固定，只有通过时钟频率和具体电路结构来实现。对于集成系统的低功耗设计，可以从系统级低功耗方法、算法级方法、结构级方法、RTL 电路级方法和器件(泄露)级方法来考虑。如系统实现有低功耗设计需求，FPGA 器件的功耗较大，不适合应用到成熟的产品中，ASIC 专用集成电路才是最佳选择。详细的低功耗设计方法请参阅第 3 章和第 7 章的相关内容。

1.3 数字集成化设计流程

数字集成电路已经完全实现了自动化设计，图 1.1 描述了 FPGA/ASIC 数字集成设计的典型流程。在设计流程中，系统设计师首先制定所设计电路的技术指标并对功能需求进行细节描述。从系统和抽象的角度对电路功能、指标、接口及总体结构进行描述。系统分析设计阶段只考虑系统的功能而不关注具体电路结构，采用的工具一般是 C/C++ 语言、SystemC/SystemVerilog 或 MATLAB 等。当系统功能仿真满足总体设计的性能要求后，硬件设计工程师使用 HDL 语言对系统进行行为级描述，其间主要完成电路逻辑功能、物理功能的实现，并进行性能的分析 and 解决其他高层次的问题。

系统的电路行为描述是最重要的一环，为提高描述语言的可理解性，一般根据其功能划分数个功能模块和子模块并完成可综合 (Synthesizable) 的

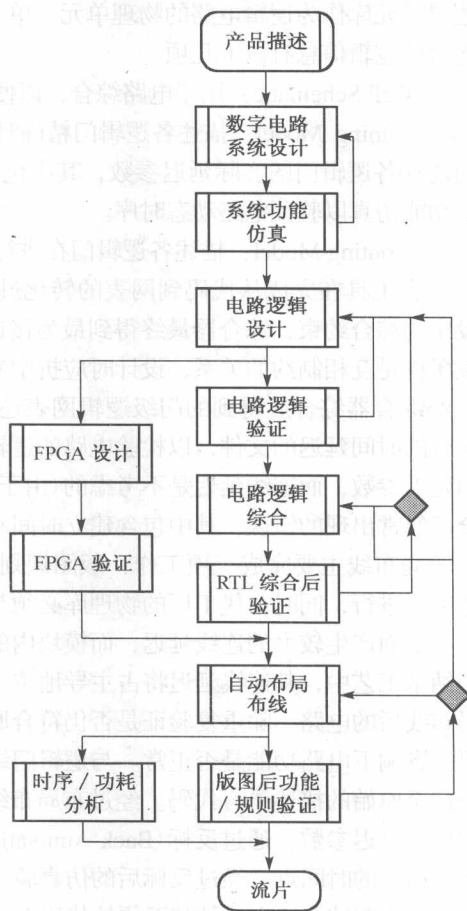


图 1.1 FPGA/ASIC 数字集成设计流程

语法描述。这种按功能需求层层分割电路单元的方法就是所谓的层次化设计(Hierarchical Design)。对于系统的行为级描述和综合化设计,设计师依赖于 EDA 工具厂商所提供的各种工具软件。在逻辑功能的仿真阶段, FPGA 设计一般使用 MentorGraphics 公司的 MondelSim 或者 FPGA 开发平台自带的功能方针平台,也就是 Windows 版本的仿真工具。而 ASIC 设计工程师一般更喜欢 NC-Verilog/Verilog-XL。对于逻辑功能的仿真,仿真器并不考虑实际逻辑门或连线所产生的时间延迟、门延迟或传输延迟等信息,而是使用单位延迟的数学模型来粗略估算电路的逻辑行为。尽管逻辑功能仿真不能得到精确物理时序等结果,但已经基本满足电路逻辑功能的设计正确性验证。为实现对电路模块的功能验证,基于 HDL 语言的测试模块(TestBench)是必要的。其中,必须考虑所有可能影响设计功能的输入信号的组合,以便发现错误的逻辑功能描述。上述仿真验证过程中,错误修改与实际的设计经验有重要的关系,初学者往往要通过大量的实验验证来总结经验。

对于 FPGA 设计,当完成电路功能验证后就可以使用相关的 FPGA 设计软件平台进行芯片设计。其平台主要有 Xilinx 公司的 ISE 开发平台和 Altera 公司的 Quartus II 平台。其设计流程主要包括功能仿真、逻辑综合、时序约束、布局布线和配置约束等几个步骤。设计可在任意开发平台下全部开发完成,不需要第三方工具软件的支持,也不需要集成电路物理层或器件布局布线的专业知识。

采用 ASIC 设计方法通过电路逻辑功能验证后,后端的工作往往更复杂也更关键。设计工作的第二阶段是逻辑综合(Logic Synthesis),此阶段依靠综合工具来实现。综合过程必须选择预计流片工厂的逻辑单元库作为逻辑电路的物理单元。单元库也可以从第三方单元库供货商获取。一般来说,单元库包含的逻辑信息有以下几项。

① Cell Schematic: 用于电路综合,以便产生逻辑电路的网表(Netlist)。

② Timing Model: 描述各逻辑门精确时序模型,设计时提取逻辑门内寄生电阻及电容进行仿真,从而建立各逻辑门的实际延迟参数,其中包含门延迟、输入/输出延迟和连线延迟等。此数据用于综合后功能仿真以验证电路动态时序。

③ Routing Model: 描述各逻辑门在进行连线时的限制,作为布线时参考。

综合工具在完成从代码到网表的转化过程中,其中心工作就是如何获得最优化的逻辑网表。根据设定的综合约束,综合器最终得到最为接近的结果。一般的约束设计有面积、功耗和速度,这三项约束条件是互相制约的关系,设计时应折中考虑以获得最优结果。

经综合器综合后得到的门级逻辑网表还要再进行第二次逻辑功能仿真,此仿真要附加反标到测试平台的时间延迟的文件,以检验电路的逻辑功能和时序约束两个方面。在综合后仿真时,一般只考虑门延迟参数,而连线延迟是不考虑的(由于无法预计实际连线的长度及使用的金属层)。时序变异是综合后经常出现的错误,其中包含建立时间和保持时间的问题,还有电磁干扰、脉冲干扰等现象。

布局布线主要完成三项工作:版图规划、布局和布线。此部分工作也必须由代工厂物理库的配合才可以进行,同时,代工厂的物理库必须与综合阶段的逻辑库相一致才可以。由于各模块之间连线较长,从而产生较大的连线延迟,而模块内的逻辑门间连线较短,因此连线延迟也较小。在深亚微米甚至纳米工艺中,其连线延迟将占主导地位。布局后的功能仿真是 ASIC 设计中最重要的一环,经过布局布线后的电路,除重复验证是否仍符合原始逻辑功能设计外,主要考虑物理实现时门延迟和连线延迟等影响下电路功能是否正常。与逻辑门级的功能验证基本相同,当发现错误时,需要修改上一级数据甚至原始的描述语言代码。经过布局布线工具所产生的标准延迟格式(SDF)文件,提供了详实的物理层次延迟参数,通过反标(Back-Annotation)仿真器能精确估算数字电路的电气行为,并可表明发生时序错误的时间点。经过反标后的仿真验证可以发现逻辑功能和时序约束的问题,对后仿时出现的问题需要修改综合约束条件甚至原始代码。

对于 ASIC 设计工程师而言,前端设计要求对 HDL 有良好的理解和对设计工作的全面把握;后

端设计则要求对所使用物理单元库的物理特性的理解、对工具的充分掌握和对流程的严格操作。布局布线后仿真尽管通过,但基于代工厂的设计规则验证和电气特性验证是流片前必须的步骤。版图验证主要包含设计规则检查(DRC),及版图与网表对比的检查(LVS),在设计中,既可以采用 Cadence 公司的 Assure 工具软件,也可使用 Mentor 的 Calibre 进行验证,深亚微米工艺一般使用后者。此时的规则检测一般来说不会有太多错误,手工修复即可,如有大量错误的话,则需要返工重做自动布局布线 APR。LVS 主要验证网表与版图的一致性、是否存在短路、断路等错误。在做 LVS 前,需要把布局布线后的网表文件转换成 Spice 网表文件,使用 Calibre 的 v2lvs 命令并配合 Spice 标准库。

以上是整个 FPGA/ASIC 设计流程的简单的描述,而在实际设计中涉及许多未提及的问题,其中包括电路性能优化、时序分析、功耗分析、可测试性设计、功能一致性验证及静态时序分析等,这些问题将在相关的章节中讲述。

1.4 数字系统实现方法

随着大规模数字集成的到来,传统的单元集成正在被系统集成所代替,即整个系统完全集成到单一芯片上,从而提高了系统的性能。数字系统的集成化实现方法主要包含现场可编程门阵列(Field Programmable Gate Array, FPGA)和专用集成电路(Application Specific Integrated Circuits, ASIC)。

1. 现场可编程门阵列

FPGA 是一个含有可编程结构单元的半导体器件,是可供使用者根据源程序代码(硬件描述语言)的修改而重复烧录的逻辑门器件,它可以分为可编程逻辑器件(Programmable Logic Device, PLD)和现场可编程逻辑阵列(FPGA)。PLD 和 FPGA 两者的功能基本相同,只是实现原理略有不同,所以有时可以忽略这两者的区别,将它们统称为可编程逻辑器件或 PLD/FPGA。由于 PLD/FPGA 可以完全免除 ASIC 芯片开发后端的大量烦琐的工作,因此备受前端数字逻辑工程师的青睐。由于现在半导体制造工艺的发展,基于纳米工艺的 FPGA 可轻松集成多达上千万门的逻辑单元,如不考虑成本和性能的条件下,FPGA 芯片完全可以取代 ASIC 产品且具有极短的开发周期。现在,FPGA 主要的产品供应商是 Xilinx 公司和 Altera 公司。以 Xilinx 公司的产品为例,介绍 FPGA 的结构。图 1.2 为 Xilinx Virtex-II Pro 的内部结构示意图,其内部单元主要包含可配置逻辑模块(Configurable Logic Block, CLB)、输入/输出接口模块(Input/Output Block, IOB)、存储模块(Block RAM)和数字延迟锁相环(DLL)。该芯片内还嵌入了 PowerPC 内核,以提高其微处理器的功能。FPGA 的大部分逻辑功能由可配置逻辑模块完成,存储模块用于完成 FPGA 内部数据的随机存储,输入/输出模块提供内部与外部的接口。

在 FPGA 结构中,可配置逻辑模块(CLB)是主要的逻辑资源,其结构如图 1.3 所示。Virtex-5 的 Slice 结构主要包含:4 个查找表(Look-Up Table, LUT),它由 6 输入端 1 bit 输出或 5 输入端 2 bit 输出配置而成;3 个用户可控制的多路复用器;专用算术逻辑(两个 1 位加法器和进位链);4 个 1 位寄存器,无论是可配置作为触发器或锁存器,这些寄存器的输入是被多路复用器所选择的。

2. 专用集成电路

在专用集成电路设计领域包括全定制设计和半定制

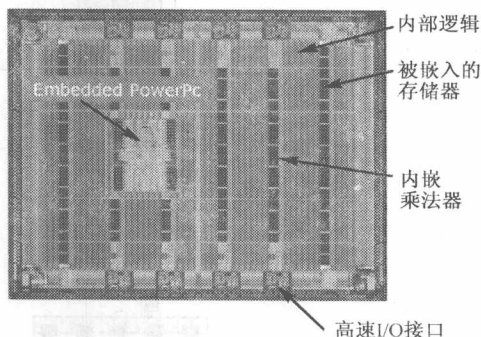


图 1.2 Xilinx Virtex-II Pro 的内部结构示意图

设计两种方法，全定制设计主要采用基于标准单元库的实现方法。对于标准单元库的设计，工艺厂或第三方提供商要开发出所有常用的逻辑单元，确定基于生产厂家的物理特性，组成一个标准单元库。标准单元库包含反相器、与非门、或非门、锁存器、寄存器等数百个单元。其中，每种逻辑门又有多种物理尺寸满足不同的扇出要求以提供足够的驱动能力。不同的单元物理尺寸可供芯片设计师选择最佳的单元以实现电路的性能指标。图 1.4 是特许公司 0.35 微米工艺的标准单元库例(反相器和或非门)。

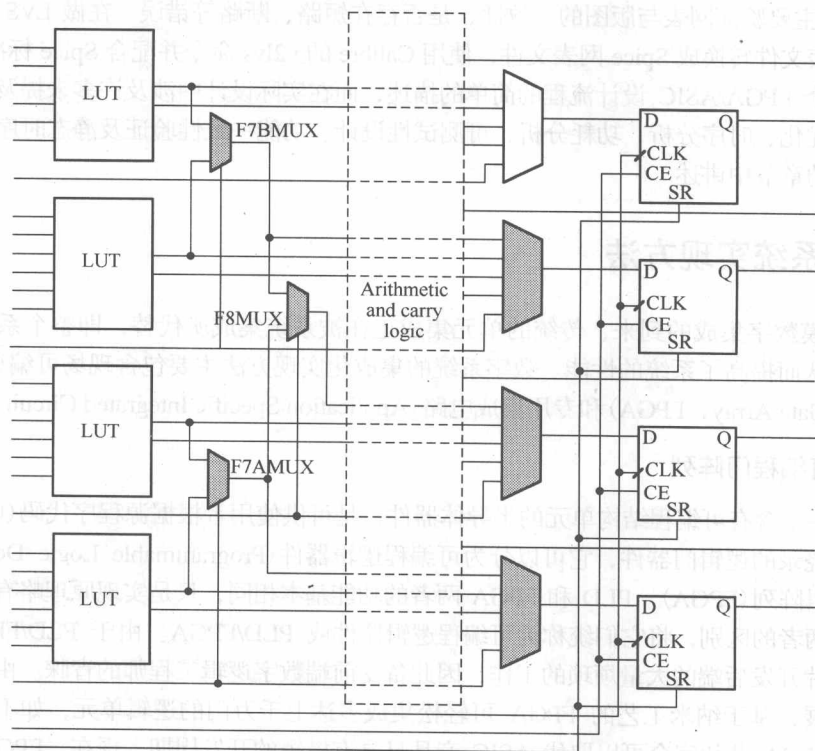


图 1.3 CLB 的结构示意图

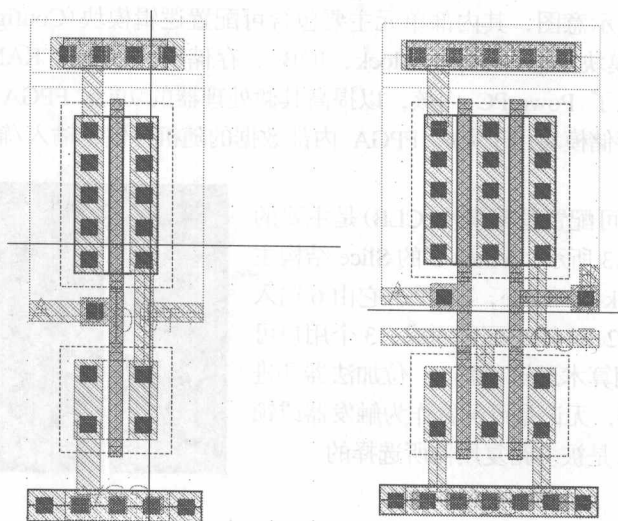


图 1.4 特许公司 0.35 微米工艺的标准单元库例

图1.5和图1.6分别示出了基于标准单元库的数字集成电路版图和基于标准单元的自动布局布线设计示意图。电路由外围 IO Pad 和内核电路组成,其中输入/输出管脚由工艺厂家提供标准库,工程师根据设计需求决定管脚的数量和位置。内核电路则是设计的核心,工程师按照产品需求通过布局布线工具实现物理层的设置。设计完成的专用集成电路系统主要考察的性能指标是芯片的面积、电路的速度和功耗等。工程师必须按照功能指标(面积、速度和功耗)实现芯片的设计,对设计的版图最终进行厂家提供的设计规则检查和电气特性检查,当完全满足设计要求时,抽取 GDSII 文件提交给工艺生产厂进行制造。这样生产厂家制造的芯片称为裸片(Die),裸片还需要进行封装以得到可使用的成品芯片。

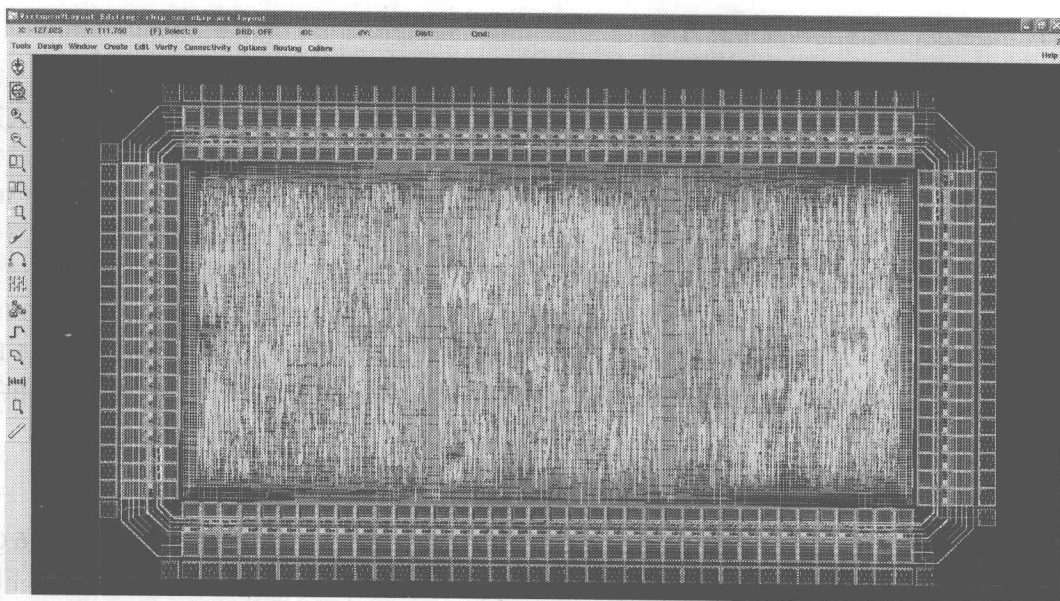


图 1.5 基于标准单元库的数字集成电路版图

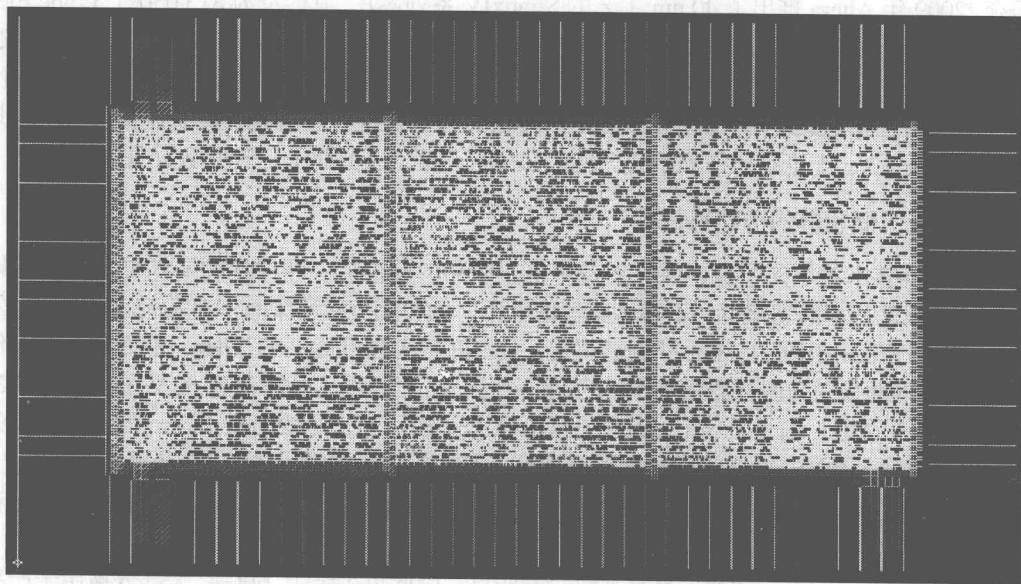


图 1.6 基于标准单元的自动布局布线设计示意图