



EDA工具应用丛书

EDA 工程的理论与实践 ——SOC系统芯片设计

曾繁泰 王 强 盛 娜 等编著



电子工业出版社

PUBLISHING HOUSE OF ELECTRONICS INDUSTRY

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

EDA 工具应用丛书

EDA 工程的理论与实践

——SOC 系统芯片设计

曾繁泰 王 强 盛 娜 等编著

電子工業出版社

Publishing House of Electronics Industry

北京 • BEIJING

内 容 简 介

本书介绍了 EDA 工程的理论基础和系统芯片 SOC 的设计方法。第 1~3 章阐述了电子设计自动化的发展历程、常用设计方法,介绍了集成电路设计的流程和集成设计环境;第 4 章介绍了 Verilog HDL 语言;第 5 章详细介绍了 VHDL 程序设计方法;第 6 章介绍了软、硬件协同设计语言——SystemC 语言;第 7 章介绍了 SOC 设计工具的使用;第 8 章阐述了集成电路的可测试设计方法、算法验证方法;第 9 章介绍了 SOC 芯片的实现方法和 EDA 工程设计方法学的进展。全书以系统芯片 SOC 为线索,阐述了 EDA 工程的设计方法、设计语言、设计流程、实现方法和验证方法。

本书可作为集成电路领域的科技工作者的读物,也可作为微电子、电子电路、通信、计算机等专业大学生、研究生的教学参考书。

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

EDA 工程的理论与实践: SOC 系统芯片设计 / 曾繁泰, 王强, 盛娜等编著. —北京: 电子工业出版社, 2004.2
(EDA 工具应用丛书)
ISBN 7-5053-9616-1

I. E… II. ①曾… ②王… ③盛… III. 数字集成电路—芯片—设计 IV. TN431.202

中国版本图书馆 CIP 数据核字 (2004) 第 003815 号

责任编辑: 雷洪勤

印 刷: 北京增富印刷有限公司

出版发行: 电子工业出版社 <http://www.phei.com.cn>

北京市海淀区万寿路 173 信箱 邮编 100036

经 销: 各地新华书店

开 本: 787×1092 1/16 印张: 30.25 字数: 774 千字

印 次: 2004 年 2 月第 1 次印刷

印 数: 6 000 册 定价: 45.00 元

凡购买电子工业出版社的图书, 如有缺损问题, 请向购买书店调换。若书店售缺, 请与本社发行部联系。联系电话: (010) 68279077。质量投诉请发邮件至 zltts@phei.com.cn, 盗版侵权举报请发邮件至 dbqq@phei.com.cn。

前 言

一、SOC 的概念

随着半导体产业进入超深亚微米乃至纳米加工技术,在单一集成电路芯片上就可以实现一个复杂的电子系统,诸如手机芯片、数字电视芯片、DVD 芯片等。事实上,高达千万个晶体管的手机基带芯片已经作为单一芯片在第二代移动电话上被广泛应用。在未来几年内,上亿个晶体管、几千万个逻辑门都可在单一芯片上实现,这将对电路的设计和整机制造带来巨大冲击。

将整个电子系统集成在同一芯片上,称为片上系统(System On Chip, SOC),或称为系统级芯片。关于 SOC 的定义,许多专家从不同侧面提出了不同的定义。笔者对片上系统(SOC)的定义为:在同一个芯片上集成了控制部件(微处理器、存储器)和执行部件(I/O 接口、微型开关、微机械),能够自成体系、独立工作的芯片称为系统芯片。数字系统芯片一般都包含三大基本要素:处理器、存储器和接口逻辑。集成了这三类电路也就相当于在单个芯片上实现了许多数字系统的所有功能。系统芯片设计和实现方法是当前 EDA 领域研究的热点课题。

系统集成的设计方法称为集成系统设计,它与集成电路设计思想不同,集成系统和集成电路的关系相当于集成电路和分立元件的关系。集成系统设计方法从整个系统的角度出发,把处理机制、模型算法、芯片结构、各层次电路直至器件的设计紧密地结合起来,在单个芯片上完成整个系统的功能,它的设计必须是从系统行为级开始的自顶向下的设计方法。

由于集成系统设计方法能够综合并全盘考虑整个系统的各种情况,集成系统设计方法依靠的是广泛的知识背景。软、硬 IP 设计中更多体现电路、器件、物理、工艺,甚至分子、原子等物理背景,而集成系统设计将更多地体现功能、行为、算法、架构,甚至思路、构想等系统背景。

集成系统设计方法的基础构件是功能模块和子系统。那些芯片面积最小、运行速度最快、功率消耗最低、工艺容差最大的设计模块具有很大的 IP 价值,将被集成系统设计工程师认可,并被集成系统复用。实现集成系统设计的关键技术之一是建立功能模块和子系统的 IP 库。这种 IP 库与传统的单元库不同,它的知识含量更高、规模更大。例如,CPU、运算器、存储器、驱动器、放大器等都可以是 IP 模块。

集成电路设计方法向集成系统设计方法转变,不仅是一种概念上的突破,同时也是信息技术发展的必然结果,它必将导致又一次以微电子技术为基础,以 EDA 工程方法为主导的信息技术革命。

二、系统芯片 SOC 带来的挑战

对集成电路制造行业而言,系统芯片使得难度本已很高的芯片设计进一步面临双重压力。其一,来自高密度的生产技术。高密度和高复杂度是对立统一的两个方面:高密度为

高复杂度提供了可能性,但高复杂度又限制了高密度的实现。随着水平线宽的进一步缩小,布线的分布效应变得越发难以处理。其二,来自于高复杂度的系统结构。由于芯片进一步吸纳系统功能,从而使得芯片的复杂度大幅度提高。就一个系统芯片而言,它不但要集中多种功能,如视频、音频、通信、控制数字信号处理等,还可能要用到多重复合技术,如数字电路和模拟电路并存,高频和低频电路并存,可编程和固定功能电路并存等。

集成电路的蓬勃发展有赖于两大动力:一是以晶片代工所形成的新兴产业供求链,二是自动化设计工具对芯片开发的有力支持。前者使得芯片设计和制造分离,进而造就了大批的芯片设计公司。这种产业供求链的重新合理布局,不仅令芯片业释放出巨大的生产力,也奠定了今天无工厂(FABLESS)芯片制造的市场基础,打破了多年的半导体生产模式的垄断。推动这一行业迅速发展的,是功不可没的电子自动化设计(EDA)工具。EDA工具以计算机自动设计和仿真的方法,使芯片无须经过制片,便可验证功能和性质指标。通过标准参数库的接口,晶片厂的加工处理要求可直接通过设计工具自动产生,并能在不同工厂代工投片。

另一方面,EDA工业也将面临其自身发展的困境。因为超深亚微米技术的引入,使自动布线的仿真遇到了极大的挑战,过高的投入和复杂的技术,使得EDA产品过于昂贵。更大的压力来自于上层高复杂度的系统设计需求,这些迫使EDA公司提供设计服务,而这些业务是与其以往的主流业务相矛盾的,因此增加了其未来产品方向的不确定性。半导体行业“一代方法造就一代产业,一代工具造就一代芯片”的法则,正在随着系统芯片的出现而悄然改变。

三、系统芯片 SOC 设计方法

SOC 设计有两种解决方案:可编程器件实现的 SOC 设计和基于平台的 SOC 设计。

1. 可编程器件改进 SOC 实现方法

可编程系统级芯片可以在一块现成的可编程芯片上提供系统级集成。这些新器件所提供的系统级功能包括处理器、存储器和可编程逻辑,而没有与 ASIC 相关的 NRE 费用或制造周期长的问题。

可编程系统级芯片还提供了掩模 ASIC 的高集成度(低功率、小尺寸、低成本)及 FPGA 的低风险、灵活性和快速上市特性。目前,IC 供应商能够提供这种类型的可编程 SOC。这些单片可编程 IC 上集成了一个微控制器,并带有不同密度的可编程逻辑、存储器和外围器件。因此,设计人员可以充分利用系统级集成的优势,而不用担心与掩模 ASIC 有关的复杂性和风险问题。事实上,在设计人员的桌面上就可直接完成 SOC 设计和生产。

可编程 SOC 的出现给所有设计师带来了全新的系统级集成技术。这一新技术将给未来各种各样的产品提供高度集成的高性能开发手段。要确定最合适的可编程 SOC 方案,需要全面考虑每个设计的成本、功耗和性能等约束因素,同时采用设计人员最熟悉的设计方法也是很重要的一个因素。

2. 开发平台提高 SOC 设计能力

一般来说,芯片平台是具有某种特征的通用电路,如 FPGA 是一种连接型平台,单一微处理器的平台是一种计算型平台。但这里讲的平台是具有某种相对稳定性的系统应用 IP

模块, 适合与某一类功能结合应用, 如高清晰度电视平台、无线通信平台、数字音响平台等。该解决方案区别于传统的平台方式在于: 使用相对专用的平台和相对通用的 IP 核。由于平台并不涉及具体产品细节, 故不随产品的更新换代而变化; 同时这类平台可利用高密度集成电路加工技术, 硬平台较易实现。例如, 一个无线平台, 可由微控制器和两个数字信号处理器及大量存储器构成。至于该平台是用于 GSM 还是 CDMA 芯片, 则可依靠调整芯片上的软件来完成。这样, 大量复杂的系统功能可转化成软件设计的工作, 这也十分符合当今逐渐软化设计的思路与潮流。

这种平台设计方案之所以具有优势, 其根本原因在于: 它把高复杂度的结构和高密度实现分割成软硬两个区域。硬件领域通过不断地优化平台电路以追求高效的物理性能, 软件领域通过不断地优化算法和软件以追求高效的系统性能。所以说, 硬件平台和嵌入式软件既是 SOC 设计的必然, 又是芯片工业的机遇。

SOC 开发平台通过提供针对特殊应用的预集成和预验证内核, 大大减轻了与 IP 集成和系统级验证有关的工作负担。而且, 这种平台将内核嵌入一种带有简单互连机制的结构中, 这样就可非常容易地增强基础平台的性能, 从而满足设计人员的特殊应用要求。另外, 开发平台进一步推动了硬、软件协同开发的策略, 有助于正确而快速地完成硬件和软件的协同设计任务。

目前有两种类型的平台可供设计人员选择: 一种是通用平台, 另一种是专用平台。通用平台可以满足单芯片嵌入式系统的设计需要, 这些系统通常由作为特殊内核的处理器、存储器和外围功能模块组成, 所有内核都是经过预集成和预验证的。同样地, 专用平台也为某些特定应用提供完善的设计, 必要时也可配置和修改以满足最终目标要求。这些专用开发平台能够进一步增强通用平台的功能, 利用其配置的特殊内核和功能, 可以满足许多特殊要求。通过更高层次的抽象思维, 设计人员能够为其具体的应用尽早实现成功的项目。

四、系统芯片带来的机遇

系统芯片 (SOC) 正在迅速地进入主流产品的行列。由此引发的“芯片就等于整机”的现象, 将对整个电子产业形成重大的冲击。种种迹象表明, 整个电子产业正在酝酿着一场深刻的产业重组, 这将为许多新兴的企业提供进入这一行业的最佳契机。目前, 半导体供求链最大的瓶颈在于设计高复杂度芯片。针对这一难题, 一种行之有效的方案, 是将芯片设计环节进一步细化, 以形成一个新的产业链。目前, 行业中被认可的方案是把芯片设计分成两部分, 即知识产权 IP 模块设计和系统芯片集成。在信息产业领域, IP 厂商可以通过硬平台或硬内核的设计重拾 FABLESS 的地位, 而且可以利用硬平台加上附加逻辑最终成片。软件公司可以通过嵌入式软件和硬平台达到无设计制片, 进而成为 SOC 芯片新秀。新的供求链更需要新的设计工具和新的开发工具, 尤其是横跨软硬件的 EDA 工具和 SOC 平台工具, 将造就一代新型的电子设计自动化工具公司。

20 世纪 70 年代 IC 问世, 80 年代计算机应用开始普及, 90 年代通信和信息基础设施建设成主流, 下一个机会便是接口器件, 2010 年之前微电子技术就会融合纳米技术。当然, 如果 IC 设计行业要抢占下一轮商业先机, 就应该探求新的设计方法和途径。

本书由曾繁泰、王强、盛娜、阚玉利执笔完成。参加本书编写工作的还有陈美金、陈洪敏、曾波、侯亚宁、孙刚见、曾峰等同志。在本书编写过程中，受到山东大学信息科学与工程学院梅良模教授、郝修田副教授、张新副教授的大力支持和悉心指导，在此表示深深的谢意。由于作者水平有限，书中谬误之处在所难免，希望广大读者批评指正。

曾繁泰

于山东大学多屏幕微机研究所

2003 年 10 月

目 录

第 1 章 绪论	(1)
1.1 EDA 工程概论	(1)
1.1.1 概述	(1)
1.1.2 EDA 工程的实现载体	(1)
1.1.3 EDA 工程的设计语言	(2)
1.1.4 EDA 工程的基本特征	(3)
1.1.5 EDA 工程的学科范畴	(4)
1.2 EDA 工程发展历程	(5)
1.3 EDA 工程与其他学科	(7)
1.3.1 EDA 工程学科与微电子技术的关系	(7)
1.3.2 EDA 工程促进其他学科的发展	(7)
第 2 章 EDA 工程设计方法	(9)
2.1 EDA 工程的设计方法	(9)
2.2 IC 设计描述法	(11)
2.2.1 集成电路设计的描述方法	(11)
2.2.2 行为描述法	(12)
2.3 IP 复用方法	(14)
2.3.1 问题的提出	(14)
2.3.2 软 IP 核与硬 IP 核	(15)
2.3.3 设计复用方法	(17)
2.3.4 基于 IP 模块的设计技术	(20)
2.4 以集成平台为基础的设计方法	(23)
2.4.1 集成平台的概念	(24)
2.4.2 集成平台的结构	(25)
2.4.3 集成平台的发展	(26)
2.5 EDA 工程集成设计环境 IDE	(27)
2.5.1 集成设计环境的概念	(27)
2.5.2 趋向集成化的 EDA 工具平台	(30)
2.5.3 集成设计环境介绍	(32)
2.6 虚拟器件协同设计环境	(33)
2.7 软、硬件协同设计方法	(37)
2.7.1 软、硬件协同设计语言	(37)
2.7.2 软、硬件划分的问题	(38)
2.7.3 软、硬件协同设计工具	(39)
2.8 EDA 工程的分层设计方法	(43)
2.8.1 层次设计方法概述	(43)

2.8.2 层次设计方法实例——4 位微处理器设计	(43)
2.9 EDA 工程的仿生学方法	(54)
2.9.1 概述	(54)
2.9.2 进化硬件	(54)
2.9.3 POE 模型	(55)
2.9.4 电子胚胎结构模型	(55)
2.9.5 仿生 SOC 芯片模型	(56)
2.10 EDA 工程综合方法	(58)
2.10.1 综合的概念	(58)
2.10.2 逻辑电路综合	(58)
2.10.3 时序电路综合	(58)
2.10.4 用 EDA 工具进行自动综合	(60)
2.11 EDA 工程仿真方法	(61)
2.11.1 概述	(61)
2.11.2 仿真方法	(62)
2.11.3 功能仿真	(66)
第 3 章 SOC 设计流程规划	(68)
3.1 流程的概念	(68)
3.1.1 EDA 工程方法与设计流程	(68)
3.1.2 集成电路产业流程	(69)
3.1.3 系统层与算法层设计流程	(70)
3.1.4 高层次综合设计流程	(70)
3.2 系统级芯片验证流程	(71)
3.3 基于模块的设计流程	(76)
3.3.1 设计环境	(76)
3.3.2 设计流程	(78)
3.4 系统芯片 SOC 设计方法	(80)
3.4.1 系统芯片 SOC 概念	(80)
3.4.2 系统芯片对 IC 产业的影响	(81)
3.4.3 系统芯片的一般设计方法	(84)
3.4.4 系统芯片的分层设计方法	(87)
3.4.5 系统芯片的集成设计方法	(90)
3.4.6 系统芯片设计的关键问题	(97)
3.5 可编程系统级芯片	(99)
3.5.1 可编程系统级芯片的结构	(99)
3.5.2 可编程系统级芯片的设计方法	(102)
3.6 系统芯片的测试方法	(105)
3.7 系统芯片的设计实例	(109)
3.7.1 单片微处理器芯片	(109)

3.7.2 多处理器系统芯片	(112)
3.8 系统芯片 SOC 展望	(115)
第 4 章 Verilog HDL 语言	(118)
4.1 概述	(118)
4.1.1 硬件描述语言的发展历史	(118)
4.1.2 HDL 语言的主要特征	(119)
4.1.3 Verilog HDL 与 VHDL 的比较	(119)
4.1.4 设计方法	(120)
4.1.5 硬件描述语言的发展	(121)
4.2 程序结构	(121)
4.3 词法约定	(125)
4.3.1 词法	(125)
4.3.2 字符串	(126)
4.3.3 标志符关键字和系统名称	(127)
4.4 数据类型	(128)
4.4.1 物理数据类型	(128)
4.4.2 抽象数据类型	(128)
4.5 运算符和表达式	(129)
4.5.1 算术运算符	(129)
4.5.2 符号运算符	(130)
4.5.3 关系运算符	(130)
4.5.4 逻辑运算符	(132)
4.5.5 位逻辑运算符	(132)
4.5.6 一元约简运算符	(133)
4.5.7 其他运算符	(134)
4.5.8 运算符优先级排序	(135)
4.6 控制结构	(135)
4.6.1 选择结构	(136)
4.6.2 重复结构	(137)
4.7 其他语句	(139)
4.7.1 参数语句	(139)
4.7.2 连续赋值语句	(140)
4.7.3 阻塞和无阻塞过程赋值	(141)
4.7.4 任务和函数结构	(142)
4.8 时序控制	(143)
4.8.1 延迟控制	(143)
4.8.2 事件	(144)
4.8.3 等待语句	(145)
4.8.4 延迟定义块	(146)

4.9 Verilog_XL 仿真	(147)
4.10 设计练习	(149)
4.10.1 简单的组合逻辑设计	(150)
4.10.2 简单时序逻辑电路的设计	(151)
4.10.3 利用条件语句实现较复杂的时序逻辑电路	(152)
4.10.4 设计时序逻辑时采用阻塞赋值与非阻塞赋值的区别	(154)
4.10.5 用 always 块实现较复杂的组合逻辑电路	(157)
4.10.6 在 Verilog HDL 中使用函数	(159)
4.10.7 在 Verilog HDL 中使用任务	(161)
4.10.8 利用有限状态机进行复杂时序逻辑的设计	(163)
第 5 章 VHDL 程序设计基础	(167)
5.1 VHDL 程序结构	(167)
5.1.1 实体及实体说明	(167)
5.1.2 类属说明和端口说明	(168)
5.1.3 结构体及其描述方法	(170)
5.1.4 库、程序包及其配置	(173)
5.2 VHDL 语言的客体及其分类	(181)
5.2.1 标志符	(181)
5.2.2 对象	(182)
5.2.3 数据类型	(185)
5.2.4 类型转换	(188)
5.2.5 运算操作符	(190)
5.3 VHDL 语法基础	(192)
5.3.1 并行语句	(192)
5.3.2 顺序语句	(210)
5.4 组合逻辑设计	(219)
5.4.1 门电路	(219)
5.4.2 编码器、译码器和选择器电路	(222)
5.5 时序逻辑电路设计	(228)
5.5.1 时钟信号的 VHDL 描述方法	(228)
5.5.2 时序电路中复位信号 Reset 的 VHDL 描述方法	(230)
5.5.3 时序电路基础模块之一 —— 触发器的设计	(231)
5.5.4 时序电路基础模块之二 —— 寄存器的设计	(236)
5.5.5 时序电路基础模块之三 —— 计数器的设计	(238)
5.6 测试平台程序的设计方法	(241)
5.6.1 实体描述可简化	(241)
5.6.2 程序中应包含输出错误信息的语句	(241)
5.6.3 配置语句	(241)
5.6.4 不同仿真目的对测试平台设计的要求	(242)

第 6 章 软、硬件协同设计语言 SystemC	(244)
6.1 SystemC 概述	(244)
6.1.1 系统级设计所面临的挑战	(244)
6.1.2 SystemC 概述	(245)
6.1.3 SystemC 引起系统级设计方法的变化	(245)
6.1.4 SystemC 开发平台	(247)
6.2 SystemC 数据类型	(248)
6.2.1 sc_bit 类型	(248)
6.2.2 sc_logic 类型	(249)
6.2.3 固定精度的有符号和无符号整数	(250)
6.2.4 任意精度符号和无符号整数类型	(252)
6.2.5 任意长度的位矢量类型	(252)
6.2.6 定点类型	(252)
6.3 用 SystemC 创建 RTL 模型	(253)
6.3.1 定义进程	(253)
6.3.2 创建模块	(255)
6.3.3 用单一 SC_METHOD 进程的模块	(264)
6.3.4 含有多个 SC_METHOD 进程的模块	(266)
6.3.5 创建层次化 RTL 模型	(269)
6.4 使用可综合的子集	(276)
6.4.1 可综合子集转换	(276)
6.4.2 系统综合前数据修改	(278)
6.4.3 系统综合前修改建议	(283)
6.5 寄存器传输级 (RTL) 编程	(283)
6.5.1 寄存器	(283)
6.5.2 三态输出	(301)
6.5.3 状态机	(305)
6.6 行为级建模和综合	(310)
6.6.1 RTL 级建模和行为级建模的比较	(310)
6.6.2 行为级综合简介	(311)
6.6.3 选择用于仿真的正确抽象方式	(314)
6.6.4 RTL 代码和行为级代码示例	(315)
6.7 SystemC 与 VHDL 设计对比	(320)
6.7.1 DFF 实例	(320)
6.7.2 移位寄存器	(323)
6.7.3 计数器	(325)
6.7.4 状态机	(327)
6.7.5 存储器	(332)
第 7 章 SOC 设计工具 Cocentric System Studio	(335)
7.1 Cocentric System Studio 概述	(335)

7.1.1 Cocentric System Studio 的组织结构	(335)
7.1.2 启动设计平台	(336)
7.1.3 浏览用户界面	(337)
7.1.4 打开工作区	(337)
7.1.5 打开设计模型	(338)
7.1.6 设计项目的编译	(340)
7.1.7 设计项目的仿真	(340)
7.1.8 小结	(341)
7.2 数据流图表 DFG	(341)
7.2.1 创建个人库	(341)
7.2.2 创建层次化 DFG	(342)
7.3 Prim 模型	(351)
7.3.1 单元级数字转换器	(351)
7.3.2 二进制到八进制转换	(353)
7.4 Control 模型	(355)
7.4.1 宏调试或门模型	(355)
7.4.2 OR 模型—复位计数器	(359)
7.4.3 OR 模型——0 与 1 计数器	(361)
7.4.4 层次化 OR 模型——加法器/乘法器转换开关	(362)
7.5 单元模型	(362)
7.5.1 单元模型错误演示	(362)
7.5.2 外部函数中的数量积	(364)
7.5.3 交换指针符号	(369)
7.5.4 利用已存在的模型建立新模型	(372)
7.5.5 仿真	(376)
7.5.6 调试	(382)
7.6 控制模型练习实验	(387)
7.6.1 “与模型”——调幅波	(387)
7.6.2 动态切换发生器	(389)
7.7 在系统平台中浮点到定点的转换	(390)
7.7.1 开启演示	(390)
7.7.2 仿真测试平台	(390)
7.7.3 建立质量量化标准	(391)
7.7.4 创建混合模型及添加字长参量	(391)
7.7.5 系统仿真时输入端口分配	(392)
7.7.6 改变模式、字长设置	(394)
7.7.7 计算系数、器件固定	(395)
7.7.8 创建定点模型	(398)
7.7.9 创建定点测试平台并仿真	(399)

7.8 算法建模——可综合的 SystemC RTL 代码产生	(400)
7.8.1 从 PRIM 模型中输出 HDL	(400)
7.8.2 从控制模型中输出 HDL	(402)
7.8.3 从 DFG 设计中输出 HDL	(403)
7.9 Cocentric 高级系统平台——使用 DAVIS	(406)
7.9.1 创建数据组、调用 DAVIS	(407)
7.9.2 选择数据组和初始化设置	(407)
7.9.3 显示 DAVIS 结果	(408)
7.9.4 计算	(408)
7.9.5 创建并连接分散窗口	(409)
7.9.6 图形显示	(409)
第 8 章 EDA 工程可测试、验证设计方法	(411)
8.1 概述	(411)
8.1.1 可测试设计	(411)
8.1.2 内建自测试	(412)
8.1.3 可测试设计中的功耗优化问题	(414)
8.1.4 可测试设计技术的发展	(415)
8.2 测试方法的范畴	(416)
8.3 可测试性分析	(417)
8.4 测试矢量生成	(418)
8.4.1 组合电路测试	(419)
8.4.2 时序电路测试	(420)
8.5 可测试性结构设计	(422)
8.5.1 分块测试	(422)
8.5.2 扫描测试设计	(422)
8.5.3 内建自测试	(427)
8.6 测试平台程序的设计方法	(430)
8.6.1 测试平台的搭建	(430)
8.6.2 不同仿真目的对测试平台设计的要求	(431)
8.6.3 用子程序方式建立测试平台	(432)
8.7 深亚微米工艺的时序分析方法	(435)
8.7.1 动态模型	(436)
8.7.2 行为模式	(436)
8.8 故障测试概述	(437)
8.8.1 故障模型	(438)
8.8.2 故障仿真	(439)
8.9 验证方法概述	(440)
8.9.1 FPGA 器件的仿真验证	(441)
8.9.2 嵌入式处理器验证环境	(443)

第9章 SOC实现方法和设计方法进展 (446)

9.1 设计实现方法的概念 (446)

9.1.1 设计实现初步 (446)

9.1.2 设计实现与逻辑综合的区分 (448)

9.2 EDA工程的CPLD实现方法 (448)

9.2.1 可编程技术 (449)

9.2.2 复杂可编程器件(CPLD) (451)

9.2.3 现场可编程门阵列(FPGA) (452)

9.2.4 百万门级FPGA的设计方法 (457)

9.3 系统芯片SOC设计方法进展 (461)

9.3.1 硬、软IP设计方法..... (461)

9.3.2 C语言用于IC系统级设计..... (462)

9.3.3 物理设计转向COT设计方法 (462)

9.3.4 EDA向EDO转变 (463)

9.4 IC设计技术的发展 (463)

9.4.1 共享RTL设计方法..... (466)

9.4.2 动态可重构技术 (466)

参考文献 (469)

第1章 绪 论

1.1 EDA 工程概论

1.1.1 概述

计算机的应用促进了新学科的诞生。她使人类克服智力局限，加深了对自然科学的认识、理解。EDA 工程方法学是计算机科学衍生出的众多学科中的一门关于电子设计自动化的学科，它是信息产业发展的核心技术之一，对国民经济发展起着推动作用。

EDA (Electronic Design Automation) 工程就是以计算机为工作平台，以 EDA 软件工具为开发环境，以硬件描述语言为设计语言，以可编程器件为实验载体，以 ASIC, SOC 芯片为设计目标，以电子系统设计为应用方向的电子产品自动化设计过程。

EDA 工程广义的定义范围应是半导体工艺版图设计自动化、可编程器件设计自动化、电子系统设计自动化、印刷电路板设计自动化、功能仿真、故障测试、形式验证自动化。EDA 工程是指电子设计自动化工程，不包含电子生产自动化。

EDA 工程学科，是近几年迅速发展起来的计算机软硬件、微电子交叉的现代电子设计学科。涉及到计算机操作系统、数据库管理、编译原理、计算数学、人工智能、电路理论、微电子学集成电路设计等领域。现代电子设计技术的发展，主要体现在 EDA 工程领域。

EDA 工程方法学是现代电子设计的基础，是电子线路原理设计、电子系统整机设计、集成电路芯片设计的方法学。

EDA 工程的主要设计目标是超大规模专用集成电路，怎样对一片超大规模集成电路进行功能划分、行为描述、逻辑综合、时序分析、故障测试、形式验证是 EDA 工程解决的主要问题。本书主要讨论 EDA 工程的基本理论和方法，涉及 EDA 工具开发以及利用 EDA 工具设计 ASIC, SOC 产品。不同领域的技术人员，不同专业的学生可有选择地阅读。

现代电子设计技术的核心就是 EDA 工程。EDA 工程的理论基础涉及硬件语言、设计方法学、系统建模、逻辑综合、故障测试、功能仿真、时序分析、形式验证等内容。EDA 工程概念如图 1.1 所示。图中的每一个环节都是 EDA 工程的一个分支领域。

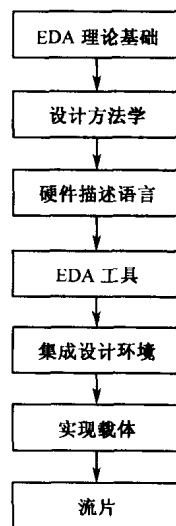


图 1.1 EDA 工程概念

1.1.2 EDA 工程的实现载体

现代电子产品的复杂度日益加深，集成电路规模急剧扩大。一个电子系统可能由数万

个中小规模集成电路构成，这就带来了体积大、功耗大、可靠性差的问题，解决这一问题的有效方法就是采用 ASIC (Application Specific Integrated Circuits) 芯片进行设计。ASIC 按照设计方法的不同可分为以下几类：全定制 ASIC、半定制 ASIC 和可编程 ASIC (也称为可编程专用集成电路)。

设计全定制 ASIC 芯片时，设计师要定义芯片上所有晶体管的几何图形和工艺规则，最后将设计结果交由 IC 厂家掩模制造完成。其优点是：芯片可以获得最优的性能，即面积利用率高，速度快，功耗低。其缺点是：开发周期长，费用高，只适合大批量的产品开发。

半定制 ASIC 芯片的版图设计方法有所不同，分为门阵列设计法和标准单元设计法，这两种方法都是约束性的设计方法，其主要目的就是简化设计，以牺牲芯片性能为代价来缩短开发周期。

可编程 ASIC 芯片设计与上述掩模 ASIC 设计的不同之处在于：设计人员完成设计后，在实验室内就可以烧制出自己的芯片，无须 IC 厂家参与，从而大大缩短了开发周期。

可编程逻辑器件自 20 世纪 70 年代以来，经历了 PAL, GAL, CPLD/FPGA 几个发展阶段。其中 CPLD/FPGA 属高密度可编程逻辑器件，目前集成度已高达 200 万门/片，它将掩模 ASIC 集成度高的优点和可编程逻辑器件设计生产方便的特点结合在一起，特别适合于样品研制或小批量产品开发，使产品能以最快的速度上市，而当市场扩大时，它可以很容易地转由掩模 ASIC 实现，因此开发风险也大为降低。

可编程 ASIC 芯片，尤其是 CPLD/FPGA 器件，已成为 EDA 工程设计方法的实现载体之一。

1.1.3 EDA 工程的设计语言

硬件描述语言是一种用于设计电子系统硬件的计算机语言，它以软件编程的方式来描述电子系统的逻辑功能、电路结构和连接形式，适合大规模电子系统的设计。

以 VHDL 语言为代表的硬件描述语言，是电子系统设计承包商的书面语言，是撰写工程承包合同文本的语言，是工程设计技术档案的规范语言。利用硬件描述语言设计电子系统和集成电路比电路图法更加简捷、准确、方便。VHDL 是一种全方位的硬件描述语言，支持系统行为级、寄存器传输级和逻辑门级多个设计层次，支持结构描述、数据流描述和行为描述及三种形式的混合描述。VHDL 语言覆盖了多种硬件语言的功能，“自顶向下”或“自底向上”的电子系统设计过程都可以用 VHDL 语言来完成。

Verilog HDL 是另一个被 IEEE 组织承认的标准的硬件描述语言，是民间的、非官方的硬件描述语言，有着广泛的民众支持，因而资源丰富，IP Core 涉及面广。

两种语言的侧重点稍有区别：VHDL 更适合大系统的描述；Verilog HDL 更适合硬件细节的描述。

在电子系统设计开发中，使用 VHDL 和 Verilog HDL 之类硬件描述语言的硬件设计人员和使用 C 及 C++ 编程语言的软件设计人员同时进行功能仿真和形式验证。这种硬件设计和软件设计使用不同语言的现象，给设计带来了不便，增加了产品开发成本，延长了设计周期。有没有更先进的语言来统一硬件、软件设计呢？回答是肯定的。SystemC (系统 C 语言) 统一了硬件与软件的描述，解决了硬件、软件设计长期分家的局面。增加定