



卓越工程师培养计划

▪ EDA ▪

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

王秀琴 夏洪洋 张鹏南 编著



Verilog HDL

数字系统设计 入门与应用实例



电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY

卓越工程师培养计划 · EDA

Verilog HDL 数字系统设计

入门与应用实例

王秀琴 夏洪洋 张鹏南 编著

電子工業出版社

Publishing House of Electronics Industry

北京 · BEIJING

内 容 简 介

本书系统介绍了硬件描述语言 Verilog HDL 及数字系统设计的相关知识, 主要内容包括 EDA 技术、FPGA/CPLD 器件、硬件描述语言 Verilog HDL 基础知识及设计实例、基于 CPLD/FPGA 数字系统设计实例。本书以应用为主、突出实践性, 书中的实例内容翔实、新颖, 结构严谨、由浅入深、化难为易、叙述清晰、通俗易懂。

本书适合从事电路设计和系统开发的工程技术人员阅读, 也可作为高等院校相关专业的教学用书。

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

Verilog HDL 数字系统设计入门与应用实例/王秀琴, 夏洪洋, 张鹏南编著. —北京: 电子工业出版社, 2012. 4

(卓越工程师培养计划)

ISBN 978 - 7 - 121 - 16588 - 7

I. ① V… II. ① 王… ② 夏… ③ 张… III. ① VHDL 语言 - 程序设计 IV. ① TP312

中国版本图书馆 CIP 数据核字 (2012) 第 049019 号

策划编辑: 张 剑 (zhang@phei.com.cn)

责任编辑: 徐 萍

印 刷: 涿州市京南印刷厂
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 × 1 092 1/16 印张: 20.75 字数: 531.2 千字

印 次: 2012 年 4 月第 1 次印刷

印 数: 4 000 册 定价: 49.00 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

随着电子技术、计算机应用技术的不断发展,使得现代数字系统的设计思想、设计方法及实现方式都进入了崭新的阶段。

为了适应这一趋势的变化,推动了电子设计自动化(EDA)技术快速发展,不同公司推出各类高性能的EDA工具,同时也促使了高性能FPGA/CPLD可编程逻辑器件的推出。FPGA/CPLD器件具有功能强大,开发周期短、投资小,便于修改等优点,已成为硬件设计的首选产品之一。硬件描述语言Verilog HDL作为IEEE标准的硬件描述语言,无论电子设计工程师还是高等院校的学生都应该熟练掌握,以提高工作效率。本书的主要内容就是将FPGA/CPLD器件、高性能的EDA工具和硬件描述语言Verilog HDL三者结合起来,实现现代数字系统的设计。

本书共分10章。第1章介绍EDA技术和数字系统的设计方法与流程;第2章首先对可编程逻辑器件进行概述,然后介绍FPGA/CPLD器件的结构、工作原理和主流产品;第3章介绍Quartus II使用指南,包括Quartus II的基本操作、设计输入、设计处理、设计验证和器件编程;第4章是ModelSim 6.5仿真软件使用指南;第5章介绍Verilog HDL硬件描述语言的模块结构、基本语法、基本语句、描述风格、数字电路的仿真等内容;第6章和第7章分别介绍组合逻辑电路和时序逻辑电路的程序设计;第8章介绍有限状态机的设计;第9章是数字系统设计实例,包括数字跑表、交通灯控制器、自动售货机、可控脉冲发生器的设计;第10章是基于FPGA数字系统设计实例。

本书从实用角度出发,紧密联系教学实际。语法介绍简明清晰,实例内容丰富、重点突出。在各基础知识章后面均附有综合实例,每一章后面都有思考与练习部分,建议读者在学完每一章内容后,都能完成各章的练习,以加深和巩固所学知识。

本身适合从事电路设计和系统开发的工程技术人员阅读,也可作为高等院校电子信息工程、电子科学与技术、电气自动化等相关专业的教学用书。

本书第3章和第4章由王秀琴编写;第5章及附录D由夏洪洋编写;第7章由张鹏南编写;第1、2、6、8章及附录C由陈晓洁编写;第9章由孙宇编写;第10章及附录A、B由尚春宇编写。参加本书编写的还有宋一兵、王献红、管殿柱、李文秋、谈世哲、赵景波。

在本书的编写过程中,青岛大学的管殿柱老师、黑龙江科技学院的穆秀春老师对书稿提出了宝贵的建议和意见,学生李伟、李健和张树龙在附录的编写过程中给予了很大的帮助,在此表示由衷的感谢!

由于编著者水平有限,书中难免存在疏漏,敬请广大读者批评指正。

编著者

目 录

第 1 章 绪论	1
1.1 EDA 技术	1
1.2 数字系统的设计	6
1.3 思考与练习	8
第 2 章 可编程逻辑器件	9
2.1 可编程逻辑器件概述	9
2.2 CPLD 的结构和工作原理	10
2.3 FPGA 的结构和工作原理	14
2.4 主流 FPGA/CPLD 产品	15
2.5 FPGA/PLD 的设计流程	18
2.6 FPGA 与 CPLD 的对比	19
2.7 思考与练习	19
第 3 章 Quartus II 开发软件	20
3.1 概述	20
3.2 Quartus II 9.1 管理器	22
3.3 设计输入	27
3.4 设计处理	35
3.5 层次设计	43
3.6 基于宏功能模块的设计	46
3.7 思考与练习	51
第 4 章 ModelSim 6.5 仿真软件	52
4.1 概述	52
4.2 ModelSim 6.5 使用举例	53
4.3 思考与练习	61
第 5 章 Verilog 硬件描述语言	62
5.1 Verilog HDL 概述	62
5.2 Verilog HDL 的模块结构	66
5.3 Verilog HDL 的基本语法	68
5.4 Verilog HDL 的基本语句	95
5.5 Verilog HDL 的描述风格	137

5.6	数字电路的仿真	145
5.7	综合实例	165
5.8	思考与练习	168
第 6 章	组合逻辑电路设计	171
6.1	编码器和译码器	171
6.2	数据选择器	177
6.3	加法器	180
6.4	乘法器	184
6.5	其他组合逻辑电路	185
6.6	综合实例	188
6.7	思考与练习	189
第 7 章	时序逻辑电路设计	190
7.1	触发器	190
7.2	锁存器和寄存器	196
7.3	移位寄存器	198
7.4	分频器	200
7.5	计数器	202
7.6	其他时序逻辑电路	205
7.7	综合实例	207
7.8	思考与练习	208
第 8 章	有限状态机的设计	209
8.1	有限状态机概述	209
8.2	有限状态机的设计要点	213
8.3	有限状态机设计实例	215
8.4	思考与练习	225
第 9 章	数字系统设计实例	226
9.1	数字跑表的设计	226
9.2	交通灯控制器的设计	229
9.3	自动售货机的设计	233
9.4	ADC0809 采样控制模块的设计	237
9.5	可控脉冲发生器的设计	242
9.6	思考与练习	247
第 10 章	基于 FPGA 的数字系统设计实例	248
10.1	基于 FPGA 的多功能数字钟的设计	248
10.2	基于 FPGA 的信号发生器的设计	272

10.3 基于 FPGA 的密码锁的设计	282
10.4 思考与练习	297
附录 A Verilog HDL 关键字 (IEEE Std1364 –1995)	298
附录 B Verilog HDL 关键字 (IEEE Std1364 –2001)	299
附录 C Verilog-2001 语法结构	300
附录 D Verilog-2002 语法结构	314
参考文献	324

第1章 绪 论

微电子和计算机领域的原理创新、技术创新、应用创新层出不穷，极大地推动了科学技术的发展。电子设计自动化技术（EDA）是现代电子工程领域的一门新技术，促使电子系统向速度更快、体积更小、质量更轻、功耗更小、稳定性更高的方向发展。专家预言，未来的电子技术将是 EDA 技术时代。



1.1 EDA 技术

EDA 技术促使电子电路设计者仅利用硬件描述语言和 EDA 软件平台，即可对系统硬件实现功能仿真，极大地提高了设计效率，缩短了设计周期，节省了设计成本。

1. EDA 技术的主要内容

从 20 世纪 60 年代起，一些电子和计算机技术较先进的国家，开始探索新的电子电路设计方法，并在设计方法、工具等方面进行了彻底的变革，取得了巨大成功。20 世纪 90 年代后，电子系统已经从电路板级系统集成发展为包括 ASIC、FPGA 和嵌入系统的多种模式，这为数字系统的设计带来了极大的灵活性。可以通过软件编程而对其硬件结构和工作方式进行重构，从而使得硬件的设计可以如同软件设计那样方便快捷。这一切极大地改变了传统的数字系统设计方法、设计过程和设计观念，促进了电子设计自动化（Electronic Design Automatic, EDA）技术的迅速发展。过去几十年里，IC 设计方法主要包括以下几个阶段。

1) 手工设计（Hand Design） 手工设计的主要目标是进行组合逻辑的优化和简化。

2) 电路仿真（Circuit Simulation） 这里的电路指 IC 中的晶体管电路。在 20 世纪 70 年代早期，像 Calma 这样的公司就把数字化系统推向了市场，并采用 Spice 进行电路仿真。

3) 原理图输入和逻辑仿真（Schematic Capture and Logic Simulation） 工作站技术与图形用户界面的结合让电路设计师能够直接用原理图输入工具来描述他们的设计；同时，以这种方式输入的设计可以在逻辑层次仿真，帮助电路设计师进行开发和调试。

4) 布局和布线（Placement&Routing） 从 20 世纪 80 年代开始，功能强大的 IC 布局工具开始出现，包括自动布局布线、设计规则检查、布局与原理图一致性检查等。同时，Verilog HDL 和 VHDL 等硬件描述语言的诞生，也促进了逻辑和寄存器传输级仿真器的发展。

5) 综合（Synthesis） 1987 年发布了商用的逻辑综合工具，这款工具可以帮助设计师自动将网络表映射到各个不同的单元库上。

【EDA 技术的概念】 电子设计技术的核心就是 EDA 技术。广义的 EDA 技术包括半导体工艺设计自动化、可编程器件设计自动化、电子系统设计自动化、印制电路板设计自动化、仿真与测试、故障诊断自动化、形式验证自动化，这些统称为 EDA 工程。狭义方面而言，EDA 技术是以计算机为工具，设计者在 EDA 软件平台上，用硬件描述语言 HDL（Hardware Description Language）完成设计文件，然后由计算机自动地完成逻辑编译、化简、

分割、综合、优化、布局、布线和仿真，直至对于特定目标芯片的适配编译、逻辑映射和编程下载等工作，最终形成集成电子系统或专用集成芯片的一门多学科融合的新技术。

EDA 技术的出现，极大地提高了电路设计的效率和可操作性，减轻了设计者的劳动强度。利用 EDA 工具，电子设计师可以从概念、算法、协议等开始设计电子系统，大量工作可以通过计算机完成，并可将电子产品从电路设计、性能分析到设计出 IC 版图或 PCB 版图的整个过程在计算机上自动处理完成。这一过程可以缩短设计周期，节省设计成本。

EDA 的范畴很宽，包括机械、电子、通信、航空航天、化工、矿产、生物、医学、军事等各个领域。基于可编程逻辑器件的数字系统的 EDA 技术，其应用范畴如图 1-1 所示。

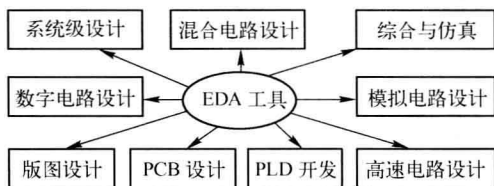


图 1-1 EDA 应用范畴

2. EDA 技术的发展趋势和特点

近年来微电子技术以惊人的速度发展，其工艺水平已达到纳米级，在一个芯片上可集成数百万乃至上千万个晶体管，工作速度可达到 Gb/s，这为制造出规模更大、速度更快和信息容量更高的芯片系统提供了基础条件。微电子技术的快速发展也对 EDA 系统提出了更高的要求，并大大地促进了 EDA 技术的发展。20 世纪 90 年代后期出现了以高级语言描述、系统仿真和综合技术为特征的 EDA 技术，不仅极大地提高了系统的设计效率，而且使设计者摆脱了大量的辅助性工作，可以将精力集中于创造性的方案与概念的构思上。

近年来 EDA 技术的发展主要有以下几方面的特点：

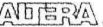
- 高层综合（High Level Synthesis, HLS）的理论与方法的进展，从而将 EDA 设计层次由 RT 级提高到了系统级（又称行为级），并且推出了相应的行为级综合工具，大大缩短了复杂 ASIC 的设计周期，改进了设计质量，提高了设计效率。
- 采用硬件描述语言（Hardware Description Language, HDL）来描述 10 万门以上的设计，并形成了 VHDL 和 Verilog HDL 两种标准硬件描述语言，更适合描述大规模系统，能够使用户对数字系统进行抽象层次的描述。它们均支持不同层次的描述，使得复杂 IC 的描述更加规范化，有利于传递、交流、保存与修改，并可建立独立工艺的设计文档。
- 采用平面规划（FloorPlaning）技术对逻辑综合和物理版图设计进行联合管理，做到在逻辑综合早期设计阶段就考虑到物理设计信息的影响。通过这些信息，设计者能进行更深入的综合与优化，并保证所做的修改只会提高性能而不会对版图设计带来负面影响。在纳米级布线延时已成为主要延时的情况下，对加速设计过程的收敛和提高成功率都是有帮助的。
- 可测性综合设计。随着 ASIC 的规模与复杂性的增加，测试的难度与费用急剧上升，由此而产生了将可测性电路结构做在 ASIC 芯片上的思想，于是开发了扫描插入、BLST（内建自测试）、边界扫描等可测性设计（DFT）工具，并已集成到 EDA 系统中。
- 为带有嵌入 μP 核的 ASIC 设计提供软、硬件协同设计工具。
- 建立并行设计工程（Concurrent Engineering, CE）框架结构的集成化、系统化设计环境，以适应当今 ASIC 的如下特点：规模大而复杂；数字与模拟电路并存；硬件与软件设计并存。产品上市速度快。该框架可以将不同公司的优秀工具集成为一个完整的 EDA 系统，并能在 UNIX 与

Windows 两种平台之间实现平滑过渡。各种 EDA 工具在该框架中可以并行使用。通过统一的集成化设计环境,能够保证各个设计工具之间的相互联系与管理。在这种集成化设计环境中,使用统一的数据管理系统与完善的通信管理系统,由若干个相关的设计小组共享数据库和知识库,同时并行地进行设计。一旦系统设计完成,相应的电路设计、版图设计、可测性设计与嵌入软件的设计等也都基本完成,适应现代数字系统设计的开发特点。

EDA 技术的发展趋势主要体现在超大规模集成电路的集成度和工艺水平不断提高两个方面。市场对系统的集成度不断提出更高的要求,高性能的 EDA 工具,其自动化和智能化程度不断提高,为嵌入式系统设计提供了功能强大的开发环境,计算机硬件平台性能大幅提高,为复杂的 SOC 设计提供了物理基础。

3. 常见 EDA 开发软件的开发商

随着 EDA 技术的不断发展,进入我国并具有广泛影响的 EDA 软件是系统设计软件辅助类和可编程芯片辅助设计软件: Protel、Altium Designer、PSPICE、MultiSIM10 (原 EWB 的最新版本)、OrCAD、PCAD、LSIlogic、MicroSim、ISE、ModelSim、Matlab 等。这些工具都有较强的功能,一般可用于几个方面,例如,很多软件都可以进行电路设计与仿真,同进还可以进行 PCB 自动布局布线,可输出多种网络表文件与第三方软件接口。按主要功能或主要应用场合,分为电路设计与仿真工具、PCB 设计软件、IC 设计软件、PLD 设计工具及其他 EDA 软件。ASIC 设计领域相当有名的软件供应商包括 Altera 公司、Xilinx 公司、Lattice 公司。其中,Altera 和 Xilinx 占有了 60% 以上的市场份额。

1) Altera 公司  Altera 是世界上“可编程芯片系统”(SOPC)解决方案倡导者。Altera 结合带有软件工具的可编程逻辑技术、知识产权(IP)和技术服务。普遍认为 Maxplus II 曾经是最优秀的 PLD 开发平台之一,适合开发早期的中小规模 PLD/FPGA,它是由 Altera 公司推出的第三代 PLD 开发系统,其全称为 Multiple Array Matrix and Programmable Logic User System,设计速度快,对于几千门的电路设计,从设计输入到器件编程完毕,只需几小时,设计处理在几分钟内完成。特别是在原理图输入等方面,被公认为最易使用、人机界面最友善的 PLD 开发软件。目前已经由 Quartus II 替代,不再推荐使用。

Altera 公司提供的开发 CPLD/FPGA 的集成化设计环境 Quartus II,简单、易学易用,其可视化、集成化设计环境等优点为大家公认,设计输入、仿真、编译、综合、布局布线和下载都可以使用这个集成环境来完成。Quartus II 支持 Altera 公司推出的所有最新的 FPGA 器件,提供了一种与结构无关的设计环境,用户只需使用自己熟悉的开发工具,通过软件提供的各种输入方式进行编译、仿真和综合,便可设计出需要的可编程逻辑器件。而且 Quartus II 软件还提供了可编程片上系统(SOC)设计的一个综合开发环境,包括以下内容:系统级设计,嵌入式软件开发,可编程逻辑器件设计。适合新器件和大规模 FPGA 的开发。

此外, SOPC Builder 配合 Quartus II,可以完成集成 CPU 的 FPGA 芯片的开发工作。DSP Builder 是 Quartus II 与 Matlab 的接口,利用 IP 核在 Matlab 中快速完成数字信号处理的仿真和最终 FPGA 实现。

2) Xilinx 公司  Xilinx (赛灵思)是全球领先的可编程逻辑完整解决方案的供应商。Xilinx 研发、制造并销售范围广泛的高级集成电路、软件设计工具,以及作为预定义系统级功能的 IP (Intellectual Property) 核。Xilinx ISE 设计套件为嵌入式、DSP 和逻辑设计人员提供 FPGA 设计工具和 IP 产品方面确立了业界新标准,是更简单、更智能的设计方法。

主要体现在：基于用户群体而优化的多种配置版本；改进了 Project Navigator 和 System Generator for DSP、Platform Studio (EDK) 和 Core Generator System 之间的交互通信；突破性的生产力、功耗和性能优势；更大的灵活性。

3) **Lattice 公司**  Lattice 是 ISP (在线可编程) 技术的发明者, ISP 技术极大地促进了 PLD 产品的发展。Lattice 在 PLD 领域发展多年, 拥有众多产品系列, 目前主流产品是 ispMACH 4000 系列 PLD 和 Lattice EC/ECP 系列 FPGA。此外, 在混合信号芯片上, 也有诸多建树, 如可编程模拟芯片 ispPAC、可编程电源管理、时钟管理等。是世界第三大可编程逻辑器件供应商。该公司的 ispLEVER 5.0 支持新的 Lattice XP 非易失、可无限重构 FPGA。

4. 常见 EDA 开发软件

根据设计流程与功能划分, EDA 工具可以分为设计输入工具、综合工具、仿真工具、实现与优化工具、后端辅助工具、验证与调试工具和系统级设计环境 7 类。

1) **设计输入工具** 这是任何一种 EDA 软件必须具备的基本功能。像 Cadence 的 Composer, Viewlogic 的 Viewdraw, 硬件描述语言 VHDL、Verilog HDL 都是主要设计语言, 许多设计输入工具都支持 HDL (如 Multisim 等)。另外, 像 Active-HDL 和其他的设计输入方法, 包括原理和状态机输入方法, 设计 FPGA/CPLD 的工具大都可作为 IC 设计的输入手段。

- **HDL 语言输入。**这种设计输入方法是目前最普遍的方法, 包括 VHDL、Verilog HDL 两种语言。个别采用 Altera 公司的 AHDL。一般来说任何文本编辑器都可以用 HDL 语言完成输入。Quartus II 内嵌的文本编辑器是 Text Editor, 它根据语法来彩色显示关键字。此外, 常用的文本编辑器还有 Ultra Edit, 通过修改 Ultra Edit 的 WORDFILE.TXT 也可以支持彩色语法显示。
- **原理图设计输入方式**在早期应用广泛, 目前已被 HDL 语言输入方式代替, 仅在某些设计的顶层描述时才会使用。Quartus II 内嵌的文本编辑器是 Schematic Editor。
- **IP Core 输入方式**是 FPGA 设计中的一个重要设计输入方式。所谓 IP Core, 是指已经设计好且受知识产权保护的标准模块单元。适当使用 IP Core, 能大幅度地减少设计工作量, 提高设计质量。
- **其他辅助性输入方法**还包括状态机输入、真值表输入和波形输入等。较流行的状态机输入工具是 StateCAD, 设计者只需画出状态转移图, 状态机编辑器自动生成相应的 HDL 语言模型。使用真值表输入方法时, 用户在工具中填充设计对应的真值表, 可自动生成 HDL 代码。波形输入方法是指用户画出输入的激励波形和输出的相应波形, 波形工具自动生成符合输入、输出关系的功能代码。

2) **综合工具** 综合工具可以把 HDL 变成门级网络表。这方面 Synopsys 工具占有较大的优势。主流的综合工具包括 Synplify 公司的 Synplify/Synplify Pro、Synopsys 公司的 FPGA Compiler II/Express、Exemplar Logic 公司的 LeonardoSpectrum 等。

- Synplify/Synplify Pro 在综合策略和优化手段上有较大幅度的提高, 特别是其先进的 Timing Driven (时序驱动) 和 BEST (Behavioral Extration Synthesis Technology, 行为级综合提取技术) 算法引擎, 使其综合结果往往面积较小、速度较快。
- Synopsys 公司的 FPGA Express 是最早的 FPGA/CPLD 综合工具之一。FPGA Express 的综合结果比较忠于原设计, 其升级版本 FPGA Compiler II 是最好的 ASIC/FPGA 设计工具之一。
- Mentor 的子公司 Exemplar Logic 出品的 LeonardoSpectrum 也是一款非常流行的综合工具, 其综合优化能力非常高。此外, 它也支持 Altera 器件。

3) **仿真工具** 几乎每个公司的 EDA 产品都有仿真工具。Verilog-XL、NC-verilog 用于 Verilog 仿真, Leapfrog 用于 VHDL 仿真, Analog Artist 用于模拟电路仿真。Viewlogic 的仿真

器有: viewsim 门级电路仿真器, speedwaveVHDL 仿真器, VCS-verilog 仿真器。Mentor Graphics 有其子公司 Model Tech 出品的 VHDL 和 Verilog 双仿真器 ModelSim。Cadence、Synopsys 用的是 VSS (VHDL 仿真器)。现在的趋势是各大 EDA 公司都逐渐使用 ModelSim。此外, Aldec 公司的 ActiveHDL 也有相当广泛的用户群。

- ModelSim 可以说是业界最流行的仿真工具之一, 其特点是仿真速度快、精度高, 支持 VHDL、Verilog HDL 及 VHDL 和 Verilog HDL 混合编程的仿真。ModelSim 的 PC 版的仿真速度很快, 甚至和 workstation 版不相上下。

4) 实现与优化工具 实现与优化工具包含的面比较广。Quartus II 集成的实现工具主要有 Assignment Editor (约束编辑器)、LogicLock (逻辑锁定工具)、PowerFit Fitter (布局布线器)、Timing Analyzer (时序分析器)、Floorplan Editor (布局规划器)、Chip Editor (底层编辑器)、Design Space Explorer (设计空间管理器) 和 Design Assistant (检查设计可靠性) 等。

- Assignment Editor 是图形界面的用户约束输入工具。约束文件包含时钟属性、延时特性、引脚位置、寄存器分组、布局布线要求和特殊属性等信息, 这些信息指导实现过程。如果约束文件设计得当, 会帮助 Quartus II 达到用户的设计目标。
- Quartus II 内嵌的 LogicLock 用以完成模块化设计流程, 通过划分每个模块的设计区域, 然后单独设计和优化每个模块, 最后将每个模块融合到顶层设计中。LogicLock 支持模块化设计流程、增量设计流程、团队设计流程等设计方法。
- PowerFit Fitter 是 Quartus II 内嵌的布局布线器。
- Timing Analyzer 是 Quartus II 内嵌的 STA 分析工具, 用以定位、分析并改善设计的关键路径, 从而提高设计的工作效率。
- Floorplan Editor 用以观察、规划、修改芯片内部的实际布局布线情况, 是用户分析设计结构、指导布局布线的重要工具。
- Chip Editor 也是分析修改芯片内部布线情况的重要工具。通过 Chip Editor 可以观察芯片时序的关键路径, 将 Chip Editor 与 SignalTap II 和 SignalProbe 调试工具配合使用, 可以加快设计验证及增量化修改在设计验证期间未解决的错误。
- DSE (Design Space Explorer, 设计空间管理器) 是控制 Quartus II 布局布线的另一种有效方法。对应一个名为 dse.tcl 的 tcl 脚本, 可以使用 quartus_sh 可执行文件从命令行运行它, 用以优化设计。DSE 界面能自动试验一系列的 Quartus II 选项和设置, 从而确定优化设计的最佳参数设置。
- Design Assistant 内嵌工具用以检查设计的可靠性, 在 HardCopy 设计流程中非常有用。

5) 后端辅助工具 Quartus II 内嵌的后端辅助工具主要有 Assembler (编程文件生成工具)、Programmer (下载配置工具) 和 PowerGauge (功耗仿真器)。

- Assembler 是 Quartus II Compiler 的一个内嵌工具, 用以完成 FPGA/CPLD 配置文件的生成。
- Programmer 是 Quartus II 内嵌的下载配置工具, 用于对 FPGA/CPLD 进行下载配置。
- PowerGauge 是 Quartus II 内嵌的功耗仿真器, 用以估算设计的功耗。

6) 验证与调试工具 Quartus II 内嵌的调试工具有 SignalTap II (在线逻辑分析仪) 和 SignalProbe (信号探针)。常用的板级仿真验证工具还有 Mentor Tau、Synopsys HSPICE 和 Innoveda BLAST 等。

- SignalTap II (在线逻辑分析仪) 和 SignalProbe (信号探针) 功能配合使用, 用以分析器件内部节点和 I/O 引脚上的信号。SignalTap II 逻辑分析仪使用内嵌的逻辑分析器将信号数据通过 JTAG 端口送往 SignalTap II 逻辑分析器或外部逻辑分析仪、示波器。SignalProbe 采用增量式路由方式, 将选定信号送往外部逻辑分析仪或示波器。

➤ Mentor Tau 和 Innoveda BLAST 板级验证工具用以分析 Quartus II 生成 STAMP 模型。

7) 系统级设计环境 Quartus II 的系统级设计环境主要包括 SOPC Builder (可编程片上系统设计环境)、DSP Builder (内嵌 DSP 设计环境) 和 Software Builder (软件开发环境)。

➤ SOPC Builder 为用户提供了一个标准化的 SOPC 图形设计环境。Altera 的 SOPC 标准结构由 CPU、存储器接口、标准外围设备和用户自定义的外围设备等组件组成。SOPC Builder 允许选择和自定义系统模块的各个组件和接口。

➤ DSP Builder 是一个图形化的 DSP 算法开发环境。DSP Builder 是由 Altera 提供的一个可选软件包, 并且 DSP 开发工具包中也包含它。DSP Builder 还可使用 Signal Tap 在线逻辑分析仪对系统进行调试, 并且可以通过 MATLAB/Simulink 接口综合、编译和下载设计, 然后执行调试。

Software Builder 是 Quartus II 内嵌的软件开发环境, 用以将软件源代码转换为配置 Excalibur 单元的 Flash 格式文件或无源格式文件, Excalibur 嵌入式处理器带分区结构的存储器初始化数据文件。Software Builder 可以转化 Excalibur 设计的软件源文件, 以及使用 SOPC Builder 和 DSP Builder 系统级设计工具创建的设计。



1.2 数字系统的设计

1. 现代电子系统设计

现代电子系统设计流程 (Top-down 设计): 对整个系统进行方案设计和功能划分, 系统的关键电路用一片或几片专用集成电路 (ASIC) 实现, 然后采用硬件描述语言 (HDL) 完成系统行为级设计, 最后通过综合器和适配器生成最终的目标器件, 这样的设计方法称为高层次的电子设计方法。

1) 设计方法 如图 1-2 所示, 自顶向下是指将数字系统的整体逐步分解为各个子系统或模块, 若子系统的规模较大, 则还需将子系统进一步分解为更小的子系统, 层层分解, 直至整个系统中各个子系统关系合理, 且便于逻辑电路级的设计和实现为止。Top-down 的设计须经过“设计—验证—修改设计—再验证”的过程, 如图 1-3 所示, 从系统最高层开始进行逐层分解, 逐层描述, 逐层仿真, 如果某一层上仿真发现问题, 则返回上一层, 寻找和修改相应的错误, 然后再向下继续进行未完的工作, 不断反复, 最后得到设计的硬件电路能够实现所要求的功能, 并在速度、功耗、价格和可靠性方面实现较为合理的平衡。它是基于芯片的设计方法, 发展趋势以硬件描述语言 (HDL) 为主。

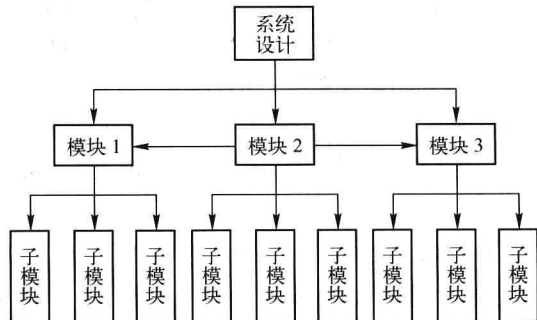


图 1-2 自顶向下设计思想

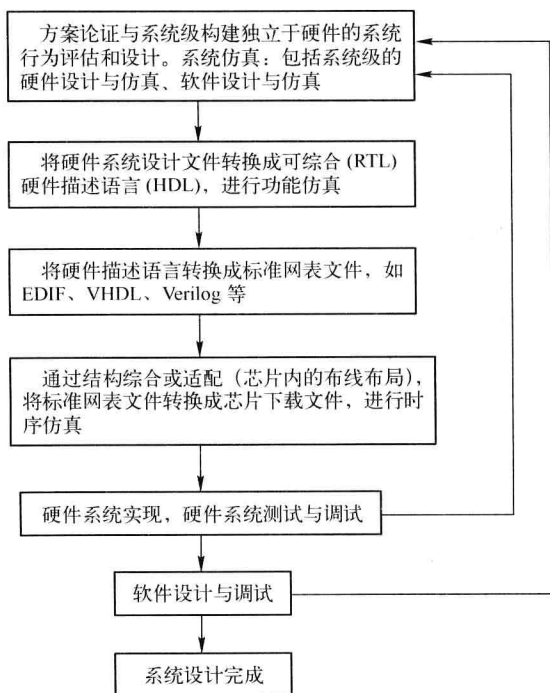


图 1-3 Top-down 设计流程

2) Top-down 设计的优点

- 对设计的描述由上而下，由粗略到详细，符合人们常规的思维习惯。
- 高层次设计与元器件无关，便于将设计结果在各种集成电路或 PLD 间移植。
- 支持模块化/层次化的设计。
- 底层模块可以被反复调用，多个底层模块也可以同时由多个设计者进行同一系统的设计。
- 可以进行软、硬件的联合设计，消除了硬件和软件开发时间上的间隔。
- 在不同层次上都易于形成用于模拟和验证的设计描述。

2. 传统电子系统设计

十几年前，电子设计的基本思路还是选用标准集成电路“自底向上”地造出一个新的系统。一般先按照电子系统的具体功能要求进行功能划分，然后对每个模块画出真值表，用卡诺图进行手工逻辑简化，写出布尔表达式，画出相应的逻辑线路图，再选择相应的元器件，设计电路板，测试。

1) 设计方法 Bottom-up 设计，即自底向上的设计，其流程如图 1-4 所示，由设计者调用设计库中的元件（如各种门电路、加法器、计数器等），设计组合出满足自己需要的系统。设计依赖于手工和经验，设计过程较复杂，尤其对于复杂的电路设计困难，工作量大也易出错。系统设计时存在的问题只有在后期才能发现，一旦系统设计中存在缺陷，就得重新设计系统。底层的设计是否满足设计要求，不能在当前层次判断，要在整个系统设计

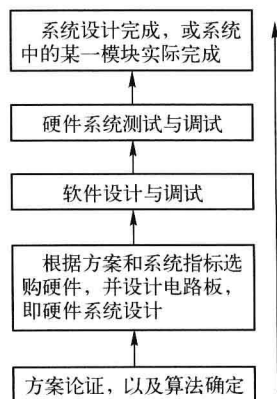


图 1-4 自底向上设计流程

完后才能做系统的测试和调试，才能确定以前的设计是否正确。这样的设计方法就如同一砖一瓦建造金字塔，先选择具体元器件，然后进行逻辑电路设计、系统的硬件设计。在整个过程中，必须始终顾及具体目标元器件的技术细节，任一时刻目标元器件的更换，或某些技术参数不满足总体要求，或不可预测的外部因素，都可能导致前功尽弃，重新开始。不仅设计的周期长，灵活性差，效率低，成本高，而且容易出错。

2) Bottom-up 设计的缺点

- 设计电路复杂，调试困难。
- 如果存在错误，查找和修改不方便。
- 设计过程产生大量文档，不易管理。
- 设计实现过程与具体工艺直接相关，因此可移植性差。
- 只有在设计出样机或生产出芯片后才能进行实测。
- 设计结果是一张电路图。



1.3 思考与练习

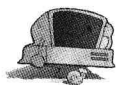
- (1) EDA 的特点有哪些？
- (2) 比较现代电子系统设计与传统电子系统设计的优缺点。
- (3) 设计方法 Top-down 与 Bottom-up 相比有哪些优点？

第2章 可编程逻辑器件

数字电子技术发展日新月异，使集成电路不断发展、更新换代，由早期的电子管、晶体管、中小规模集成电路，发展到超大规模集成电路及许多实现专门功能的专用集成电路。同时随着微电子技术的发展，设计与制造集成电路的任务已不完全由半导体厂商来独立承担，设计工程师们希望自行开发设计出可用的集成电路芯片。因此出现了各种可编程逻辑器件，至今主要有5种器件可供实现专用集成电路的要求，它们是：

- 简单可编程逻辑器件 (Programmable Logic Device, PLD)；
- 复杂可编程逻辑器件 (Complex Programmable Logic Device, CPLD)；
- 现场可编程器件 (Filed Programmable Gate Array, FPGA)；
- 标准单元 (Standard Cell)；
- 门阵列 (Gate Array)。

这些器件的出现，使电子系统的设计工程师利用与器件相应的计算机辅助设计 (Computer Aided Design, CAD) 软件，在实验室里就可以设计出应用最为广泛的 ASIC 芯片。



2.1 可编程逻辑器件概述

规模可编程逻辑器件 (Programmable Logic Devices) 是集成电路技术发展的产物。它是 EDA 得以实现的硬件基础，由设计人员自行编程将数字系统“集成”在一片 PLD 上，可灵活方便地构建和修改数字电子系统，而不必去请芯片制造厂商设计和制作专用的集成电路芯片。

1. PLD 的发展概述

随着微电子设计技术与工艺的发展，数字集成电路从电子管、晶体管、中小规模集成电路、超大规模集成电路 (VLSIC) 逐步发展到今天的专用集成电路 (ASIC)。ASIC 的出现降低了产品的生产成本，提高了系统的可靠性，缩小了设计的物理尺寸，推动了社会的数字化进程。但是 ASIC 因其设计周期长、改版投资大、灵活性差等缺陷制约着它的应用范围。硬件工程师希望有一种更灵活的设计方法，根据需要，在实验室就能设计、更改大规模数字逻辑，研制自己的 ASIC 并马上投入使用，这是提出可编程逻辑器件的基本思想。

可编程逻辑器件随着微电子制造工艺的发展取得了长足的进步。从早期的只能存储少量数据、完成简单逻辑功能的可编程只读存储器 (PROM)、紫外线可擦除只读存储器 (EPROM) 和电可擦除只读存储器 (E^2 PROM)，发展到能完成中大规模数字逻辑功能的可编程阵列逻辑 (PAL) 和通用阵列逻辑 (GAL)，今天已经发展为可完成超大规模复杂组合逻辑与时序逻辑的复杂可编程逻辑器件 (CPLD) 和现场可编程逻辑器件 (FPGA)。随着工艺技术的发展与市场需要，超大规模、高速、低功耗的新型 FPGA/CPLD 不断推出。新一代的 FPGA 甚至集成了中央处理器 (CPU) 或数字处理器 (DSP) 内核，在一片 FPGA 上进行

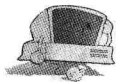
软、硬件协同设计,为实现片上可编程系统(System On Programmable Chip, SOPC)提供了强大的硬件支持。

2. PLD 的分类及特点

广义上讲,可编程逻辑器件是指一切通过软件手段更改、配置器件内部连接结构和逻辑单元,完成既定设计功能的数字集成电路。目前常用的可编程逻辑器件主要有简单的逻辑阵列(PAL/GAL)、复杂可编程逻辑器件(CPLD)和现场可编程逻辑阵列(FPGA)3大类。

- PAL/GAL: PAL (Programmable Array Logic) 即可编程阵列逻辑; GAL (Generic Array Logic) 即通用可编程阵列逻辑。PAL/GAL 是早期可编程逻辑器件的发展形式,其特点是大多基于 E^2 CMOS 工艺,结构较为简单,可编程逻辑单元多为与、或阵列,可编程单元密度较低,仅适用于某些简单的数字逻辑电路。虽然 PAL/GAL 密度较低,但是它们一出现即以其低功耗、低成本、高可靠性、软件可编程、可重复更改等特点引发了数字电路领域的巨大震动。虽然目前较复杂的逻辑电路一般使用 CPLD 甚至 FPGA 完成,但是对应很多简单的数字逻辑, GAL 等简单的可编程逻辑器件仍然被大量使用。目前,国内外很多对成本十分敏感的设计都在使用 GAL 等低成本可编程逻辑器件,越来越多的 74 系列逻辑电路被 GAL 取代。GAL 等器件发展至今已经近 20 年了,新一代的 GAL 以功能灵活、小封装、低成本、重复可编程、应用灵活等优点仍然在数字电路领域扮演着重要的角色。目前比较大的 GAL 器件供应商主要是 Lattice 半导体。
- CPLD: CPLD (Complex Programmable Logic Device) 即复杂的可编程逻辑器件。Altera 为了突出特性,曾将自己的 CPLD 器件称为 EPLD (Enhanced Programmable Logic Device),即增强型可编程逻辑器件。其实 EPLD 和 CPLD 属于同等性质的逻辑器件,目前 Altera 为了遵循称呼习惯,已经将其 EPLD 统称为 CPLD。CPLD 是在 PAL、GAL 的基础上发展起来的,也采用 E^2 CMOS 工艺,也有少数厂商采用 Flash 工艺,其基本结构由可编程 I/O 单元、基本逻辑单元、布线池和其他辅助功能模块构成。CPLD 可实现的逻辑功能比 PAL、GAL 有了大幅度的提升,一般可以完成设计中较复杂、较高速度的逻辑功能,如接口转换、总线控制等。CPLD 的主要器件供应商有 Altera、Lattice 和 Xinlinx 等。
- FPGA: FPGA (Filed Programmable Gate Array) 即现场可编程逻辑阵列,是 20 世纪 80 年代中期出现的高密度可编程器件,短短二十几年来,取得了惊人的发展,其单片集成密度从最初的 1 200 门发展到目前的几百万门,而且时钟频率由最初不到 10MHz 发展到目前的 300MHz。FPGA 是在 CPLD 的基础上发展起来的新型高性能可编程逻辑器件,一般采用 SRAM 工艺,也有一些专用器件采用 Flash 工艺或反熔丝 (Anti-Fuse) 工艺等。它可以完成极其复杂的时序与组合逻辑电路功能,适用于高速、高密度的高端数字逻辑电路设计领域。FPGA 的基本组成部分有可编程 I/O 单元、基本可编程逻辑单元、嵌入式块 RAM、丰富的布线资源、底层嵌入功能单元、内嵌专用硬核等。FPGA 的主要器件供应商有 Altera、Lattice、Actel 和 Xinlinx 等。

尽管 FPGA、CPLD 和其他类型的 PLD 的结构各有其特点和长处,但概括起来,它们都是由三大部分组成的:一个二维的逻辑块阵列,构成了 PLD 器件的逻辑组成核心;输入/输出块;连接逻辑块的互连资源。连线资源由各种长度的连线线段组成,其中也有一些可编程的连接开关,它们用于逻辑块之间、逻辑块与输入/输出之间的连接。



2.2 CPLD 的结构和工作原理

大部分的 CPLD 是基于乘积项 (Product-Term) 结构的。采用这种结构的 CPLD 芯片有: