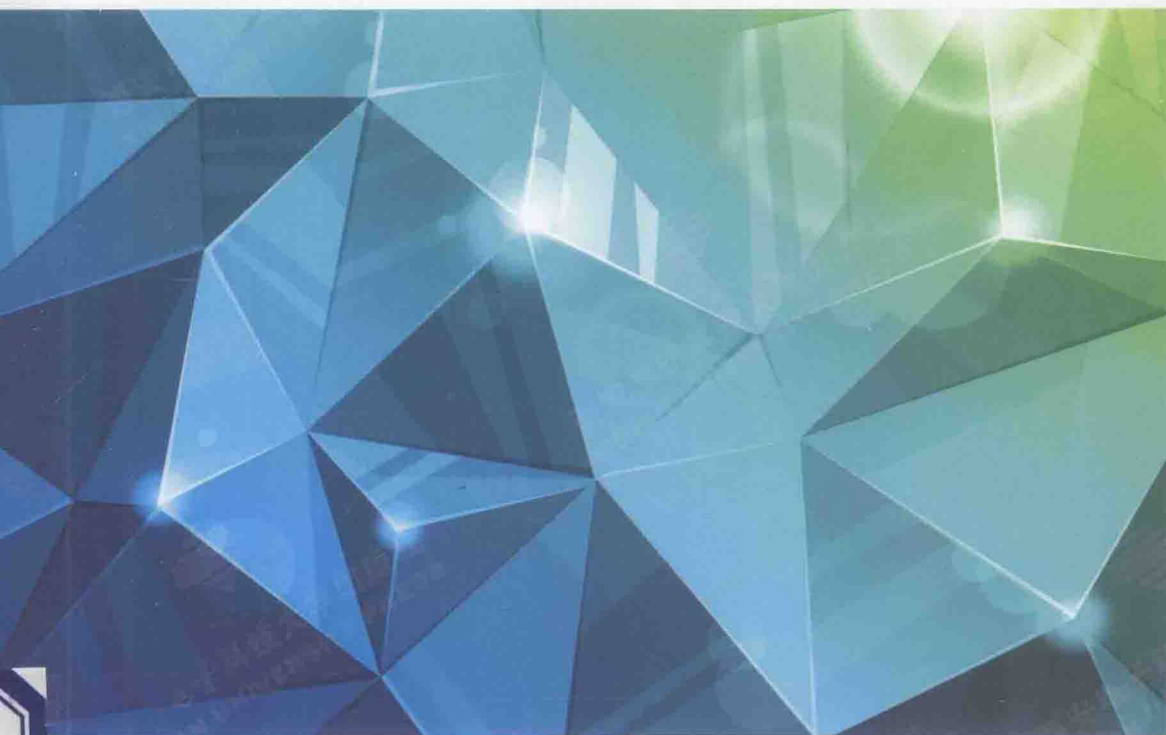


应用型本科“十二五”规划教材

数字电路EDA实用教程

主编 顾 斌 魏 欣
姜志鹏 李立早



西安电子科技大学出版社
<http://www.xduph.com>

应用型本科“十二五”规划教材

数字电路 EDA 实用教程

主 编 顾 斌 魏 欣 姜志鹏 李立早

副主编 施慧彬 孙 玥

主 审 李正权

西安电子科技大学出版社

内 容 简 介

全书从实用的角度出发,通过大量工程实例,详细介绍了利用 EDA 技术设计可编程逻辑器件的方法和技巧。全书分为 3 篇,共 6 章。上篇为可编程逻辑器件基础知识篇,简要介绍了 EDA 技术的应用领域、FPGA 和 CPLD 硬件结构,详细叙述了 Altera 公司和 Xilinx 公司常用开发工具的使用方法;中篇为硬件描述语言编程基础篇,通过大量实例详细介绍了 VHDL 和 Verilog 这两种常用的硬件描述语言的语法和常用数字电路功能的实现;下篇为数字系统开发实例篇,包括实用数字钟电路设计、串口通信设计、FIR 滤波器的设计和数字频率计的设计。

本书语言简洁、结构清晰,以实际工程为背景,实例典型、丰富,全部来自于实践并且调试通过,是作者多年开发经验的推广与总结,代表性和指导性强,利于读者举一反三。

本书适合计算机、自动化、电子及硬件等相关专业的大学生,以及从事 FPGA 开发的科研人员使用。

图书在版编目(CIP)数据

数字电路 EDA 实用教程/顾斌等主编. —西安:西安电子科技大学出版社,2014.7

ISBN 978-7-5606-3397-8

I. ①数… II. ①顾… III. ①数字电路—电路设计—计算机辅助设计—教材 IV. ①TN79

中国版本图书馆 CIP 数据核字(2014)第 119096 号

策 划 马乐惠

责任编辑 马乐惠 董柏娴

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

网 址 www.xduph.com 电子邮箱 xdupfb001@163.com

经 销 新华书店

印刷单位 陕西华沐印刷科技有限责任公司

版 次 2014 年 7 月第 1 版 2014 年 7 月第 1 次印刷

开 本 787 毫米×1092 毫米 1/16 印 张 15

字 数 353 千字

印 数 1~3000 册

定 价 26.00 元

ISBN 978-7-5606-3397-8/TN

XDUP 3689001-1

*** 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

IT 产业飞速发展的今天,电子技术逐步跨入了“数字时代”,以可编程逻辑器件和硬件描述语言为载体的数字电路 EDA 技术已成为数字电子技术的重要发展方向。可编程逻辑器件的基本特点如下:

(1) 采用可编程逻辑器件设计的 ASIC 电路,用户不需要投片生产,就能得到适合的芯片。

(2) 可编程逻辑器件可作为其他全定制或半定制 ASIC 电路的中试样片。

(3) 可编程逻辑器件是 ASIC 电路中设计周期短、开发费用低、风险小的器件之一。

可以说,可编程逻辑器件芯片是小批量系统提高系统集成度、可靠性的很好选择之一。

目前可编程逻辑器件在数字系统、通信系统、网络开发及汽车电子方面得到了深入应用。目前市场上都是一些介绍基础原理的书,缺少有大量实例导航的工程实践书,本书的出版正好可以填补这个空白。

全书从实用的角度出发,通过大量工程实例,详细介绍了可编程逻辑器件的设计方法与技巧。全书共分为 3 篇,共 6 章。上篇为可编程逻辑器件基础知识篇,简要介绍了 EDA 技术的应用领域、FPGA 和 CPLD 硬件结构,详细叙述了 Altera 公司和 Xilinx 公司常用开发工具的使用方法;中篇为硬件描述语言基础篇,通过大量实例详细介绍了 VHDL 和 Verilog 这两种常用的硬件描述语言的语法和常用数字电路功能的实现;下篇为数字系统开发实例篇,包括实用数字钟电路设计、串口通信设计、FIR 滤波器的设计和数字频率计的设计。

与同类型书相比,本书具有以下特色:

(1) 本书提供了丰富的设计实例,范围涉及数字系统常用的功能电路。实例典型,且全部调试通过,易学易懂。

(2) 全书对可编程逻辑器件开发的步骤和设计思路进行详细讲解,穿插介绍开发经验技巧与注意事项,对程序代码进行详细注释,利于读者理解和巩固知识点,举一反三。

(3) 本书中的实例读者稍加修改,便可应用于实际工作中或者完成相关的课题(课程设计或毕业设计),有实用价值。

本书适合计算机、自动化、电子及硬件等相关专业的大学生,以及从事 FPGA 开发的科研人员使用。

本书主要由顾斌、魏欣、姜志鹏和李立早编写。施慧彬老师和孙玥老师提出了很多宝贵意见。依元素科技有限公司提供了大量丰富的工程设计实例。本书还参考了大量的书刊资料,并引用了部分资料,在此一并向这些书刊资料作者表示衷心的感谢。

由于编者水平有限,本书不足之处在所难免,恳请广大专家和读者批评指正。

编 者

目 录

上篇 可编程逻辑器件基础知识

第 1 章 绪论	2
1.1 概述	2
1.2 EDA 技术的应用领域	3
1.3 EDA 的设计步骤	4
1.4 TOP-DOWN 设计方法	6
1.5 硬件描述语言	7
1.5.1 ABEL-HDL	7
1.5.2 Verilog-HDL	7
1.5.3 VHDL	8
1.5.4 Verilog-HDL 和 VHDL 的比较	8
1.6 可编程逻辑器件开发工具	8
1.6.1 ispLEVER	9
1.6.2 ISE	9
1.6.3 Quartus II	10
1.7 IP 核概述	11
第 2 章 Xilinx 公司产品介绍及开发平台应用	13
2.1 Xilinx 公司及其 FPGA 产品	13
2.1.1 概述	13
2.1.2 可配置逻辑块 CLB	14
2.1.3 输入/输出模块 IOB	16
2.1.4 Block RAM	18
2.2 ISE Design Suite 集成开发环境	19
2.2.1 ISE Design Suite 各功能模块简介	20
2.2.2 ISE Foundation 软件介绍	20
2.3 ISE Foundation 设计流程	25
2.3.1 问题分析	25
2.3.2 设计输入	26
2.3.3 工程编译	36
2.3.4 仿真验证	36
2.3.5 器件配置与编程	40
2.3.6 IP 核的使用	51

习题 2.....	55
第 3 章 Altera 公司产品介绍及开发平台应用.....	56
3.1 Altera 公司 CPLD 芯片.....	56
3.1.1 概述.....	56
3.1.2 功能描述.....	59
3.1.3 逻辑阵列块.....	60
3.1.4 用户 Flash 存储区.....	62
3.2 ModelSim 的设计过程.....	64
3.2.1 新建工程与源文件.....	64
3.2.2 ModelSim 仿真.....	67
3.3 Quartus II 的设计过程.....	68
3.3.1 设计输入.....	68
3.3.2 编译.....	72
3.3.3 编译前的约束设置.....	73
3.3.4 仿真前的参数设置.....	75
3.3.5 仿真.....	76
3.3.6 引脚分配.....	79
3.4 Quartus II 与 ModelSim 联合仿真.....	80
3.4.1 存储器初始化文件.....	80
3.4.2 MegaWizard Plus-In Manager 定制 ROM.....	81
3.4.3 Quartus II 与 ModelSim 联合仿真.....	86
习题 3.....	92

中篇 硬件描述语言编程基础

第 4 章 VHDL 编程基础.....	94
4.1 VHDL 概述.....	94
4.2 VHDL 结构.....	95
4.2.1 实体说明.....	96
4.2.2 构造体.....	98
4.2.3 程序包、库及配置.....	107
4.3 VHDL 顺序语句与并发语句.....	110
4.3.1 顺序语句.....	110
4.3.2 并行语句.....	117
4.4 VHDL 要素.....	125
4.4.1 VHDL 文字规则.....	125
4.4.2 VHDL 数据对象.....	126
4.4.3 VHDL 数据类型.....	128
4.4.4 VHDL 运算符.....	132

4.5 数字电路的 VHDL 实现	136
4.5.1 组合逻辑电路设计	136
4.5.2 时序逻辑电路的 VHDL 实现	141
4.5.3 状态机的 VHDL 实现	146
知识梳理与总结	151
习题 4	151
第 5 章 Verilog 编程基础	152
5.1 Verilog 简介	152
5.1.1 模块	153
5.1.2 时延	153
5.1.3 数据流描述方式	153
5.1.4 行为描述方式	154
5.1.5 结构化描述形式	156
5.1.6 混合设计描述方式	157
5.2 Verilog 要素	158
5.2.1 标识符	158
5.2.2 注释	158
5.2.3 格式	159
5.2.4 编译指令	159
5.2.5 值集合	161
5.2.6 数据类型	163
5.3 表达式	166
5.3.1 算术操作符	167
5.3.2 关系操作符	168
5.3.3 相等关系操作符	168
5.3.4 逻辑操作符	169
5.3.5 按位操作符	169
5.3.6 归约操作符	169
5.3.7 移位操作符	170
5.3.8 条件操作符	170
5.3.9 连接和复制操作	171
5.4 Verilog HDL 的基本语法	171
5.4.1 赋值语句	171
5.4.2 块语句	175
5.4.3 条件语句	178
5.4.4 循环语句	184
5.4.5 结构说明语句	187
5.5 数字电路的 Verilog 实现	190
5.5.1 简单的组合逻辑设计	190

5.5.2 简单时序逻辑电路的设计	190
5.5.3 利用条件语句实现较复杂的时序逻辑电路	191
5.5.4 用 always 块实现较复杂的组合逻辑电路	192
5.5.5 在 Verilog HDL 中使用函数	193
5.5.6 利用有限状态机进行复杂时序逻辑的设计	193
5.5.7 模块之间的调用实现自顶向下的设计	197
知识梳理与总结	199
习题 5	200

下篇 数字系统开发实例

第 6 章 典型数字系统设计	204
6.1 实用数字钟电路设计	204
6.1.1 分频器	205
6.1.2 时分秒计时器	206
6.1.3 数码管驱动模块	208
6.1.4 报时电路	211
6.1.5 顶层设计	212
6.1.6 数字钟的硬件验证	212
6.2 串口通信	213
6.2.1 异步串口数据传送格式	214
6.2.2 波特率发生器	214
6.2.3 数据发送模块程序	215
6.2.4 数据接收模块程序	216
6.2.5 顶层设计	218
6.2.6 串口通信的硬件验证	218
6.3 FIR 滤波器的设计	219
6.3.1 FIR 滤波器原理	219
6.3.2 FIR 滤波器直接运算法	220
6.3.3 FIR 滤波器 IP 核法	221
6.4 数字频率计的设计	223
6.4.1 测频原理	223
6.4.2 频率计的组成结构分析	223
6.4.3 频率计的 VHDL 程序	224
6.4.4 频率计的仿真结果	225
6.4.5 频率计的硬件验证	225
习题 6	226
附录 Nexys3 开发板	227
参考文献	232

上

篇



可编程逻辑器件基础知识

可编程逻辑器件设计是 EDA 技术的重要应用领域。应用 EDA 技术设计可编程逻辑器件时,设计者只需正确描述所需逻辑功能即可,然后由 EDA 软件平台根据设计者提供的逻辑描述完成对指定目标可编程逻辑器件内部的布局布线工作。由于主要逻辑功能由可编程逻辑器件内部电路承担,而可编程逻辑器件内部连线很短,因此基于可编程逻辑器件的数字电路可以达到较高的运行速度与可靠性。此外,EDA 软件平台通常提供软件仿真功能,也可以使用专门的软件仿真工具对已有的设计结果进行功能仿真、时序仿真、驱动仿真甚至电磁兼容方面的验证。当仿真结果显示不能达到设计要求时,一般只需修改设计者的设计描述而不需重新设计硬件电路,即使是对硬件电路的修改也只是软件中部分语句的修改,所消耗资源较少。本篇将重点介绍可编程逻辑器件内部结构和 EDA 软件平台的使用方法。

第 1 章 绪 论

【教学导航】 本章介绍了数字电路 EDA(Electronic Design Automation, 电子设计自动化)技术的基本概念、应用领域与设计步骤, 简要介绍了常用的高速集成电路硬件描述语言(high-speed-integrated-circuit Hardware Description Language)和常用 EDA 开发工具及其特点, 最后介绍了 EDA 中 IP 核技术的发展。主要内容如下:

- EDA 技术概述;
- EDA 应用领域;
- EDA 设计步骤;
- HDL 概述;
- 常用 EDA 开发工具;
- IP 核概述。

1.1 概 述

进入 21 世纪的十多年来, 随着计算机技术与微电子技术的持续发展, 数字化社会的特征进一步彰显, 以数字集成电路为代表的数字电路已进入社会生活的各个领域。数字电路应用领域扩大的同时, 其相应的功能设计也越来越复杂, 这就对数字电路的设计方法提出了更高的要求。

传统数字电路的主要功能模块由功能固定的中、小规模集成电路(SSI、MSI)、大规模集成电路(LSI)搭建而成。设计者在明确设计要求后, 需要根据设计要求选择功能已知的 SSI、MSI 与 LSI, 然后根据所选择的芯片考虑整个系统的硬件设计方案。概括起来说, 传统的数字电路设计具有以下缺点:

(1) 由于所选择的集成电路功能固定, 因此一旦设计方案确定并制造交付, 硬件电路便不能修改、升级。

(2) 如果硬件经测试不能满足设计要求或者需要对逻辑功能进行调整、升级, 则必须重新设计并制造, 而实现复杂逻辑功能需要成百上千的 SSI、MSI 芯片与大量 LSI, 此时如果重新设计并制造硬件, 需要消耗较多的人力与物力。

(3) 数字电路的相应控制全部由连线完成, 只要参照成品的连线即可以仿制电路, 电路的保密性低。

(4) 由大量 SSI、MSI、LSI 搭建而成的电路, 其复杂的芯片外围连线对电路工作速度及工作的可靠性产生了不利影响: 一方面其连线长度制约了所能达到的工作速度, 另一方

面过多的连线使电路也易受到外界的干扰。

鉴于上述缺点,传统的数字电路设计已经越来越不适应当下经济对电子设计的实时快速、易于检修、保密和升级的要求,而 EDA 技术与可编程逻辑器件的出现与发展弥补了传统数字电路设计方法的不足。EDA 是电子设计自动化的简称,这里的“自动化”主要指电子设计的关键工作由计算机自动完成。

可编程逻辑器件设计是 EDA 技术的重要应用领域。应用 EDA 技术设计可编程逻辑器件时,设计者只需正确描述所需逻辑功能即可,然后由 EDA 软件平台根据设计者提供的逻辑描述完成对指定目标可编程逻辑器件内部的布局布线工作。由于主要逻辑功能由可编程逻辑器件内部电路承担,而可编程逻辑器件内部连线很短,因此基于可编程逻辑器件的数字电路可以达到较高的运行速度与可靠性。此外,EDA 软件平台通常提供软件仿真功能,也可以使用专门的软件仿真工具对已有的设计结果进行功能仿真、时序仿真、驱动仿真甚至电磁兼容方面的验证。当仿真结果显示不能达到设计要求时,一般只需修改设计者的设计描述而不需重新设计硬件电路,即使是对硬件电路的修改也只是软件中部分语句的修改,所消耗资源较少。

EDA 技术已成为当今电子设计领域的重要技术。基于 EDA 技术,目前绝大多数数字电路均可由 CPU 与可编程逻辑器件及必要的外围电路(如存储器等)配合实现。学会使用数字电路设计这一强大的工具、掌握 EDA 技术,是 21 世纪相关专业人员掌握数字技术的重要环节。



1.2 EDA 技术的应用领域

EDA 技术在电子设计领域的主要应用包括电子 CAD(Computer Aided Design)与集成电路设计。

电子 CAD 即计算机辅助设计,它是 EDA 最早的应用领域。电子 CAD 的使用可以追溯到 20 世纪 70 年代,当时的 CAD 软件主要利用计算机软件帮助设计者进行 PCB(Printed Circuit Board)布线。进入 20 世纪 80 年代,CAD 软件在电路仿真方面有了很大的发展,设计者在 CAD 软件帮助下对电路进行功能检验,以期在设计交付之前能够发现问题。CAD 软件代替了一部分手工计算与操作,提高了电子设计的效率与可靠性。

随着电子 CAD 的发展,EDA 技术也日益应用于集成电路设计,尤其是 ASIC(Application Specific Integrated Circuit,专用集成电路)设计中。ASIC 是一种为满足某种特定应用目的而设计的集成芯片,其“专用”是相对通用集成电路而言的。ASIC 通常分为模拟 ASIC、数字 ASIC、模/数混合 ASIC 与微波 ASIC,本节只讨论数字 ASIC。

数字 ASIC 可以划分为全定制 ASIC、半定制 ASIC 和可编程 ASIC 三大类。

全定制 ASIC 需要设计者借助全定制 IC 版图设计工具,由设计者手工设计 IC 版图,包括芯片内部的布局布线、规则验证、参数提取、一致性检查等。这种 ASIC 对设计人员提出了很高的经验要求,设计周期长且设计成本高,适用于批量很大的芯片。

半定制 ASIC 实际上是一种半成品的 ASIC,这种 ASIC 内部已经预制好单元电路,但各单元之间的连线掩膜尚未制造,有待设计确定。半定制 ASIC 包括门阵列 ASIC 与标准单

元 ASIC。门阵列 ASIC 片上提供了大量规则排列的单元(早期的单元是门, 故称门阵列), 将这些单元按不同规则连接到一起就可以产生不同的功能。标准单元 ASIC 的特征是采用标准单元库, 设计时通过调用库中提供的标准单元的版图完成版图设计。由于标准单元库的内容经过精心设计, 因此通过调用其所设计的版图往往能用较短的设计周期获得较高的性能。

无论是全定制 ASIC 还是半定制 ASIC, 当版图设计出来后, 仍然要返回到 IC 生产厂家去制造。而可编程 ASIC 与此不同, 这种 ASIC 出厂时其制造工艺已全部完成, 用户只要借助个人电脑与相应软件即可进行“编程”, 经过“编程”的芯片可直接应用于系统。可编程 ASIC 的典型应用是 PLD(可编程逻辑器件)。

可编程逻辑器件的核心价值体现为“可编程”。可编程是指器件的内部硬件连接可修改, 大部分的可编程逻辑器件可以多次修改其内部布局布线, 从而改变其所具有的逻辑功能, 这为设计的修改完善与产品升级带来了很大的灵活性。由于其主要逻辑功能在 PLD 内部实现, 外界只能看到输入、输出引脚, 不能轻易知悉 PLD 内部的连接情况, 因而也增加了数字电路设计的保密性。

可编程逻辑器件早期的产品包括 PROM、PAL、PLA、GAL 等, 集成度较低, 一般将其称为低密度 PLD; 而 CPLD、FPGA 由于集成度较高而称为高密度 PLD。可编程逻辑器件经过数十年的发展, 使用越来越普及, 集成度越来越高, 以往需要多个芯片构成的数字系统如今可以在一片超大规模 PLD 芯片上实现。本书讲述的重点是 EDA 技术在可编程逻辑器件方面的应用, 其中, 第 2 章将介绍 Xilinx 公司的 FPGA 产品系列和基本结构, 第 3 章将介绍 Altera 公司的 CPLD 产品系列和基本结构。



1.3 EDA 的设计步骤

EDA 的设计步骤主要包括设计输入、设计实现、设计验证与器件下载。

1. 设计输入

EDA 设计输入指设计者采用某种描述工具描述出所需的电路逻辑功能, 然后将描述结果交给 EDA 软件进行设计处理。设计输入的形式有硬件描述语言输入、原理图输入、状态图输入、波形输入或几种方式混合输入等, 其中硬件描述语言输入是最重要的设计输入方法。目前业界常用的硬件描述语言有 VHDL、Verilog-HDL、ABEL-HDL, 本书主要介绍 VHDL 的设计输入方法。

2. 设计实现

设计实现的过程由 EDA 软件承担。设计实现是将设计输入转换为可下载至目标器件的数据文件的全过程。设计实现主要包括优化(Optimization)、合并(Merging)、映射(Mapping)、布局(Placement)、布线(Routing)、产生下载数据等步骤。

优化是指 EDA 软件对设计输入进行分析整理, 使得逻辑最简, 并将其转换为适合目标器件实现的形式。

合并是指将多个模块文件合并为一个网表文件。

映射是指根据具体的目标器件内部结构对设计进行调整,使逻辑功能的分割适合于用指定的目标器件内部逻辑资源实现。映射之前软件产生的网表文件与器件无关,主要是以门电路和触发器为基本单元的表述;映射之后产生的网表文件对应于具体的目标器件的内部单元电路,比如针对 Xilinx 公司的 FPGA 芯片,映射后的网表文件将逻辑功能转换为以 CLB 为基本单元的表述形式,便于后续布局。

映射将逻辑功能转换为适合于目标器件内部硬件资源实现的形式后,实施具体的逻辑功能分配,即用目标器件内不同的硬件资源实现各个逻辑功能,这一过程称为布局。针对 Xilinx 公司的 FPGA 芯片,布局就是将映射后的各个逻辑子功能分配给具体的某个 CLB 的过程。

布线是指在布局完成后,根据整体逻辑功能的需要,将各子功能模块用硬件连线连接起来的过程。

产生下载数据是指产生能够被目标器件识别的编程数据。对于可编程逻辑器件而言,CPLD 的下载数据为熔丝图文件,即 JEDEC 文件;FPGA 的下载数据为位流数据文件,即 bitstream 文件。

3. 设计验证

设计验证包括功能仿真、时序仿真与硬件测试。这一步通过仿真器来完成,利用编译器产生的数据文件自动完成逻辑功能仿真和延时特性仿真。在仿真文件中加载不同的激励,可以观察中间结果以及输出波形。必要时,可以返回设计输入阶段,修改设计输入,以满足最终的设计要求。

基于 EDA 软件强大的仿真功能,设计者可以在将数据下载至目标芯片之前或在制造出芯片之前通过软件对设计效果进行评估,这极大地节约了成本。高档的仿真软件还可以对整个系统设计的性能进行评估。仿真不消耗资源,仅消耗少许时间,而这些时间与设计成本相比完全值得。

功能仿真与时序仿真统称为软件仿真。二者的主要区别在于仿真时是否需要针对具体的目标器件考虑时序延迟。

功能仿真主要验证设计结果在逻辑功能上是否满足设计要求,这种仿真不考虑逻辑信号实际运行时不可避免的延迟信息,可以在选择指定目标器件之前进行,或者在指定了目标器件但尚未进行布局布线之前进行,因此有时也称之为前仿真。

时序仿真由仿真软件根据目标器件内部的结构与连线情况,在仿真时考虑信号的延迟,尽可能地模拟实际运行状况。时序仿真必须在指定了目标器件且已经实现了布局布线后才能进行,因此有时也称之为后仿真。显然,在评估设计结果的性能、分析时序关系、消除竞争冒险等情况下必须进行时序仿真。

硬件测试是指将下载数据下载至目标器件中,然后从硬件实际运行效果的角度验证设计是否达到预期要求。

4. 器件下载

器件下载也称为器件编程,是将设计实现阶段产生的下载数据通过下载电缆下载至目标器件的过程。

使用查找表(LUT)技术和基于 SRAM 的 FPGA 器件(如 Altera 的 APEX、Cyclone, Xilinx

的 Spartan、Virtex), 将下载的数据存入 SRAM, 而 SRAM 掉电后所存数据将丢失, 为此, 需将编程数据固化入 EEPROM 内。FPGA 上电时, 由器件本身或微处理器控制 EEPROM 将数据“配置”入 FPGA 器件。FPGA 调试期间, 由于编程数据改动频繁, 没有必要每次改动都将编程数据下载到 EEPROM, 此时可用下载电缆将编程数据直接下载到 FPGA 内查看运行结果, 这种过程称为在线重配置(ICR)。

使用乘积项逻辑、基于 EEPROM 或 Flash 工艺的 CPLD 器件(如 Altera 的 MAX 系列、Xilinx 的 XC9500 系列以及 Lattice 的多数产品)进行下载编程时, 应使用器件厂商提供的专用下载电缆, 该电缆一端与 PC 的打印机并行口相连, 另一端接 CPLD 器件所在 PCB(印刷电路板)上的 10 芯插头(PLD 只有 4 个引脚与该插头相连)。编程数据通过该电缆下载到 CPLD 器件中, 这个过程称为在系统编程(ISP)。ISP 过程如图 1-1 所示。



图1-1 在系统编程示意图

部分 CPLD 与 FPGA 不能进行 ISP 或 ICR, 下载数据时需要将目标芯片放入专门的编程器进行数据下载, 下载之后再目标芯片焊到系统电路板上。



1.4 TOP-DOWN 设计方法

TOP-DOWN 设计方法即自顶向下设计方法, 是数字系统设计常用的设计方法, 其本质是模块化设计, 其精髓在于对系统功能按层逐渐分解, 按层进行设计, 按层进行验证仿真。采用 TOP-DOWN 方法设计某系统时, 需要将设计的逻辑功能从上到下分解为功能子块, 再分别对每个功能子块进行功能划分, 从而得到各个功能子块下一层的若干功能子块。依此类推, 对每一层的各个功能子块都可进行功能划分, 从而得到下一层功能子块。功能划分的目标是将总体系统功能具体化、模块化, 功能划分的最底层是具体寄存器与逻辑门电路或其他单元电路。功能划分结束后, 从上至下对各层的各功能子块进行设计描述。最底层以上的每一层各功能子块的更详细设计描述在子块的下一层中说明。最后利用 EDA 工具, 经过逻辑综合与适配, 把决定 PLD 内部硬件连接的编程数据下载至有关器件, 即完成了电子设计自动化过程。

功能划分时, 一部分较高层次的设计描述比较抽象, 这些高层与具体的硬件实现无关, 因此不需考虑具体的目标器件, 可以对其进行功能仿真, 从而在设计的早期阶段就可以验证设计方案的可行性。一旦高层次的逻辑功能满足要求, 就可以在底层针对具体的目标器件进行具体描述。此外, 由于高层的抽象描述未涉及具体的器件, 因此后期选择目标器件时更加自由。

自顶向下的设计方法是一种对系统功能由粗到细进行设计描述的过程, 这一过程符合大多数人思考解决问题的习惯, 很容易被广大设计者接受并使用。由于这种方法本质是模

块化设计,可以在合适的层次上将各功能子块分配给不同的设计者进行设计,从而极大地节省了设计时间,非常适合系统功能越来越复杂的现状。

1.5 硬件描述语言

自顶向下的设计方法在高层次描述系统功能时,不涉及具体的器件,属于形式化抽象描述。硬件描述语言就是一种形式化描述语言,可以较抽象地描述数字电路的逻辑功能,是以自顶向下为主要设计方法的数字系统设计中主要的设计描述方法。常用的 HDL 有 ABEL-HDL、Verilog-HDL 和 VHDL。

1.5.1 ABEL-HDL

ABEL-HDL 是美国 DATA I/O 公司开发的硬件描述语言。用户使用 ABEL-HDL 进行设计时,无需考虑或较少涉及目标器件的内部结构,只需输入符合语法规则的逻辑描述。ABEL-HDL 语言支持布尔方程、真值表、状态图等逻辑表达方式,能准确地表达计数器、译码器等逻辑功能。

由于 ABEL-HDL 是在早期的简单可编程逻辑器件(如 GAL)的基础上发展而来的,因此进行较复杂的逻辑设计时,ABEL-HDL 与 VHDL、Verilog-HDL 这些由集成电路发展起来的 HDL 相比稍显逊色。

目前支持 ABEL-HDL 语言的开发工具很多,有 DOS 版的 ABEL4.0(主要用于 GAL 的开发)、DATAT I/O 的 Synario、Lattice 的 ispEXPERT、Xilinx 的 Foundation 等软件。通过文件转换,ABEL-HDL 程序可以被转换为 VHDL 等其他 HDL。

ABEL-HDL 的基本结构可包含一个或几个独立的模块,每个模块包含一整套对电路或子电路的完全逻辑描述。无论有多少模块都能结合到一个源文件中,并同时予以处理。ABEL-HDL 源文件模块可分成五段:头段、说明段、逻辑描述段、测试向量段和结束段。

1.5.2 Verilog-HDL

Verilog-HDL 是目前应用较广泛的一种硬件描述语言。设计者可以用它来进行各种级别的逻辑设计,可以用它进行数字逻辑系统的仿真验证、时序分析、逻辑综合等。Verilog-HDL 是在 1983 年由 GDA(Gateway Design Automation)公司的 Phil Moorby 首创的。Phil Moorby 后来成为 Verilog-XL 的主要设计者和 Cadence 公司的第一个合伙人。20 世纪 80 年代中期,Mooby 设计出了第一个关于 Verilog-XL 的仿真器,他对 Verilog-HDL 的另一个巨大的贡献是于 1986 年提出了用于快速门级仿真的 XL 算法。随着 Verilog-XL 算法的成功,Verilog-HDL 得到迅速的发展。1989 年,Cadence 公司收购了 GDA 公司。1990 年,Cadence 公司决定公开 Verilog-HDL,于是成立了 OVI(Open Verilog International)组织来负责 Verilog-HDL 的发展。基于 Verilog-HDL 的优越性,IEEE 于 1995 年制定了 Verilog-HDL 的 IEEE 标准,即 Verilog-HDL 1364-1995。

Verilog-HDL 是专门为 ASIC 设计而开发的,本身即适合 ASIC 设计。在亚微米和深亚微米 ASIC 已成为电子设计主流的今天,Verilog-HDL 的发展前景是非常远大的。

Verilog-HDL 较为适合算法(Algorithm)级、寄存器传输(RTL)级、逻辑(Logic)级和门(Gate)级设计。对于特大型的系統级设计, VHDL 则更为适合。

Verilog-HDL 把一个数字系統当作一组模块来描述, 每一个模块都具有接口以及关于模块内容的描述, 一个模块代表一个逻辑单元, 这些模块用网络相互连接、相互通信。

1.5.3 VHDL

VHDL(Very-high-speed-integrated-circuits Hardware Description Language, 超高速集成电路硬件描述语言)是美国国防部于 20 世纪 80 年代后期出于军事工业的需要开发的。1984 年, VHDL 被 IEEE 确定为标准化的硬件描述语言。1994 年, IEEE 对 VHDL 进行了修订, 增加了部分新的 VHDL 命令与属性, 增强了系统的描述能力, 并公布了新版本的 VHDL, 即 IEEE 标准版本 1046-1994 版本。VHDL 已经成为系統描述的国际公认标准, 得到众多 EDA 公司的支持, 越来越多的硬件设计者使用 VHDL 描述系统的行为。

VHDL 涵盖面广, 抽象描述能力强, 支持硬件的设计、验证、综合与测试。VHDL 能在多个级别上对同一逻辑功能进行描述, 如可以在寄存器级别上对电路的组成结构进行描述, 也可以在行为描述级别上对电路的功能与性能进行描述。无论哪种级别的描述, 都有赖于优良的综合工具将 VHDL 描述转化为具体的硬件结构。

相对于其他硬件描述语言, VHDL 的抽象描述能力更强。运用 VHDL 进行复杂电路设计时, 非常适合自顶向下分层设计的方法。首先从系統级功能设计开始, 对系統的高层模块进行行为与功能描述并进行高层次的功能仿真, 然后从高层模块开始往下逐级细化描述。

VHDL 设计描述的基本结构包含有一个实体和一个结构体, 而完整的 VHDL 结构还包括配置、程序包与库。本书第 4 章将对 VHDL 进行详细介绍。

1.5.4 Verilog-HDL 和 VHDL 的比较

Verilog-HDL 和 VHDL 都已成为 IEEE 标准。其共同的特点在于: 能形式化地抽象表示电路的结构和行为, 支持逻辑设计中层次与领域的描述, 可借用高级语言的精巧结构来简化电路的描述, 具有电路仿真与验证机制以保证设计的正确性, 支持电路描述由高层到低层的综合转换, 便于文档管理, 易于理解 and 设计重用。VHDL 是一种高级描述语言, 适用于电路高级建模, 综合的效率和效果都比较好。Verilog-HDL 是一种较低级的描述语言, 最适于描述门级电路, 易于控制电路资源。

VHDL 直接描述门电路的能力不如 Verilog-HDL, 反之, Verilog-HDL 在高级描述方面不如 VHDL。VHDL 入门较难, 但在熟悉以后, 设计效率明显高于 Verilog-HDL, 生成的电路性能也与 Verilog-HDL 不相上下。在 VHDL 设计中, 综合器完成的工作量是巨大的, 设计者所做的工作就相对减少了; 而在 Verilog-HDL 设计中, 设计者的工作量通常比较大, 因为设计者需要搞清楚具体电路结构的细节。本书以介绍 VHDL 硬件描述语言为主。



1.6 可编程逻辑器件开发工具

可编程逻辑器件的设计离不开 EDA 开发工具。现在有多种支持 CPLD 和 FPGA 的设

计软件,有的设计软件是由芯片制造商提供的,如 Lattice 开发的 ispLEVER 软件包、Xilinx 开发的 ISE 软件包、Altera 开发的 Quartus II 软件包。

由专业 EDA 软件商提供的 EDA 开发工具称为第三方设计软件,例如 Cadence、Mental、Synopsys、Viewlogic 和 DATA I/O 公司的设计软件。但利用第三方软件设计具体型号的器件时,需要器件制造商提供器件库和适配器软件。

本节将简要介绍由芯片制造商提供的三款开发工具。

1.6.1 ispLEVER

Lattice 公司于 1983 年成立,1992 年发明第一个 ISP 可编程逻辑器件。ispLEVER 是 Lattice 公司提供的新款 EDA 软件。这款软件提供设计输入、HDL 综合、仿真、器件适配、布局布线、编程和在系统设计调试等功能。ispLEVER 作为一个软件包,不但包含了 Lattice 自己开发的用于器件编程的 ispVM 以及用于 ispClock 和 ispPAC 器件设计的 PAC-Designer 软件工具等,还包含了众多的第三方工具。如用于综合的第三方工具包括 Synplicity 公司的 Synplify 和 Exemplar Logic 公司的 Leonado 综合工具,用于仿真的第三方工具包括 Mentor Graphics 的 ModelSim。

ispLEVER 的设计输入可采用原理图、硬件描述语言、原理图和硬件描述语言混合输入三种方式。其中硬件描述语言支持 ABEL-HDL、VHDL 与 Verilog-HDL。此外,ispLEVER 还可以采用 EDIF(Electronic Design Interchange Format, 电子设计交换格式)输入。

ispLEVER 的仿真工具支持功能仿真和时序仿真。其时序分析与仿真功能强大,能够帮助设计者进行充分的时序分析检查,确保设计从硬件上满足时序要求。ispLEVER 的仿真分析工具采用布线延迟估计算法,产生的结果更接近布线后的结果。通过时序分析检查工具提供的报告,设计者可以迅速确定系统的关键路径与元件,也可以判断时钟域是否被约束或如何被约束、时钟域之间的路径是否被约束或如何被约束、时钟域之间的数据通路是否需要修正等。

较新版本的 ispLEVER 支持 Lattice 公司的 ECP2/MFPGA 系列芯片,该系列芯片可以使用 ispLEVER 的时钟提速功能,通过在传递路径中改变时钟的边沿来平衡该路径,从而提高最高时钟频率。

随着数字系统的工作频率越来越快,保持时间不足成为系统中最可能发生的时序问题之一。保持时间不足通常发生在时钟偏移大于数据时延的场合。较新版本的 ispLEVER 提供了自动校正保持时间不足的功能,极大地减轻了设计的工作量。

ispLEVER 软件适用于 Lattice 公司的从 ispLSI、MACH、ispGDX、ispGAL、GAL 器件到 FPGA、FPSC、ispXPGATM 和 ispXPLDTM 产品系列的所有可编程逻辑产品。

1.6.2 ISE

ISE 是 Xilinx 公司提供的 EDA 设计软件,该软件有 Foundation 版和 WebPack 版,二者主要功能相同,区别在于 WebPack 版是免费版本,但支持的器件型号相对较少。这款软件提供设计输入、综合、仿真、布局布线、配置和在线调试等功能。ISE 是一个软件包,除了 ISE 集成环境 Project Navigator 外,还集成了众多的软件工具。

ISE 的设计输入工具包括: