



Xilinx大学合作计划指定教材

超值大容量DVD光盘

内含对应本书内容的
Xilinx ISE Design Suite 10.1 Web版软件
和本书所有实例的完整工程文件



Xilinx

ISE Design Suite 10.x

FPGA开发指南

——逻辑设计篇

田耘 徐文波 胡彬 等 编著



人民邮电出版社
POSTS & TELECOM PRESS

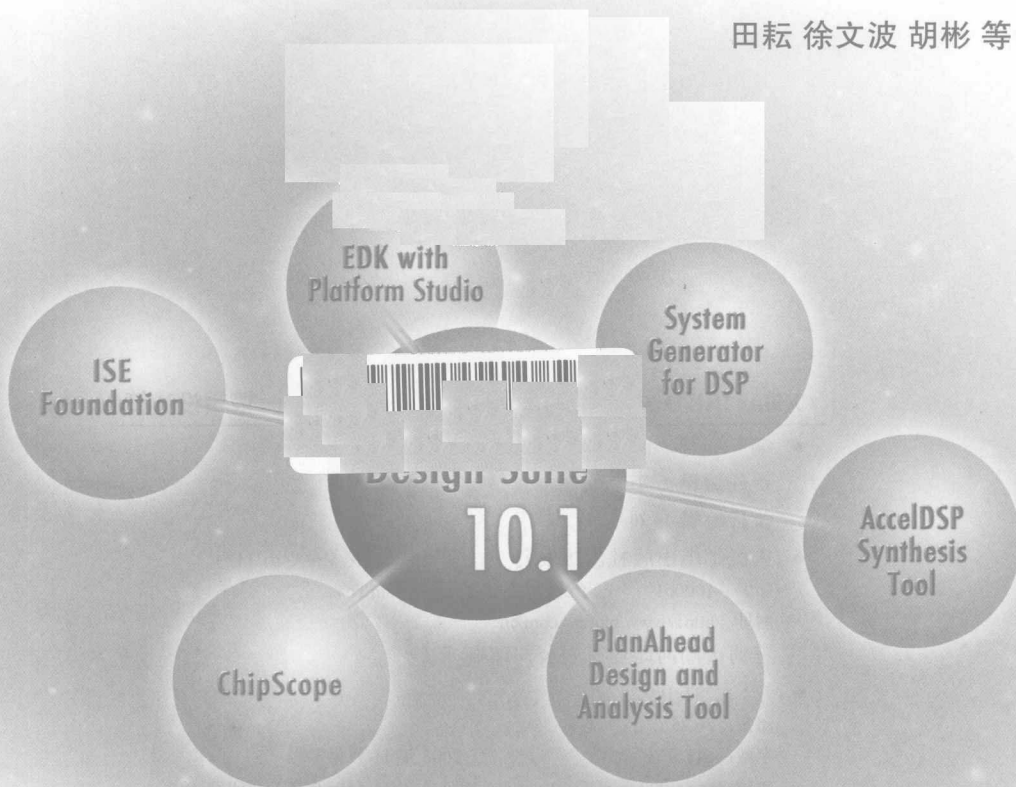


Xilinx大学合作计划指定教材

Xilinx ISE Design Suite 10.x FPGA开发指南

——逻辑设计篇

田耘 徐文波 胡彬 等 编著



人民邮电出版社

北京



图书在版编目 (CIP) 数据

Xilinx ISE Design Suite 10.x FPGA 开发指南. 逻辑设计篇 / 田耘等编著. —北京: 人民邮电出版社, 2008.11
ISBN 978-7-115-18736-9

I. X… II. 田… III. 可程序逻辑器件—系统设计
IV. TP332.1

中国版本图书馆 CIP 数据核字 (2008) 第 133161 号

内 容 提 要

本书以 Xilinx FPGA 逻辑开发流程为主线, 以浅入深出、图文并茂的方式, 全面、详细地介绍了 Xilinx 公司的终极开发套件 ISE Design Suite 10.1 中逻辑开发的操作方法, 并精选了多个实际开发案例进行深入讲解。书中内容结合了作者多年的实际开发经验, 具有很高的实践指导价值。

本书针对性较强, 可满足实际工程开发的需求。本书可作为电子和通信工程师的实用工具书, 还可作为高等院校通信工程、电子工程、计算机以及微电子与集成电路等相关专业的高年级本科生以及研究生的教材和学习参考书。

Xilinx ISE Design Suite 10.x FPGA 开发指南——逻辑设计篇

◆ 编 著 田 耘 徐文波 胡 彬 等

责任编辑 王晓明

执行编辑 刘 洋

◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街 14 号

邮编 100061 电子函件 315@ptpress.com.cn

网址 <http://www.ptpress.com.cn>

三河市潮河印业有限公司印刷

◆ 开本: 787×1092 1/16

印张: 26.25

字数: 641 千字

2008 年 11 月第 1 版

印数: 1—4 000 册

2008 年 11 月河北第 1 次印刷

ISBN 978-7-115-18736-9/TN

定价: 59.00 元 (附光盘)

读者服务热线: (010)67120142 印装质量热线: (010)67129223

反盗版热线: (010)67171154

前言

近 10 年来, FPGA 行业获得了突飞猛进的发展, 门数量的增加和生产工艺的进步使 FPGA 走到了技术的前沿, FPGA 在开发工作中的地位从胶合逻辑的配角上升到数字系统的核心处理器件。美国 Xilinx 公司是全球领先的可编程逻辑器件 (PLD) 及完整解决方案的供应商, 提供了类型多样、功能强大的 FPGA 器件, 以及软件设计工具和丰富的 IP 核。这些都为工业界和学术界提供了优秀的开发和应用平台。FPGA 的应用非常广泛, 涉及无线通信、多媒体处理、汽车和消费电子、国防和宇航工业等诸多领域。

Xilinx 公司致力于为全球范围的用户提供先进的芯片和设计理念, 同时也非常关注高校师生的创新活动。Xilinx 在中国多所高校成立了联合实验室, 帮助在校老师和学生掌握 FPGA 的开发流程和设计方法。今天的年轻学习者, 未来将可能成为专业的开发工程师。他们现在打好基础, 掌握了学习和跟踪技术发展前沿的能力, 对我国未来可编程逻辑器件产业的发展将起到积极的推动作用。

目前, Xilinx 最新的开发软件为 ISE Design Suite 10.1 版本, 这个版本首次统一了逻辑、DSP 和嵌入式等不同设计领域, 能够为用户提供更高的设计效率。为了让读者更好地学习这个软件, 我向读者推荐人民邮电出版社出版的《Xilinx ISE Design Suite 10.x FPGA 开发指南——逻辑设计篇》和《Xilinx ISE Design Suite 10.x FPGA 开发指南——DSP、嵌入式与高速传输篇》。这两本书详细讲解了 Xilinx ISE Design Suite 10.1 软件的操作方法和它在逻辑设计、DSP 开发、嵌入式设计和高速传输领域的应用。希望这两本书能让更多的 FPGA 学习者深入了解并掌握基于 ISE Design Suite 设计的基本原理和方法, 并促进 FPGA 技术在我国普及和推广。

沈嘉

工业和信息化部 (原信息产业部) 电信研究院高级工程师
英国约克大学电子学博士

2008 年 9 月于北京

前言

目前, FPGA 设计已经与固定架构芯片的设计变得同样复杂, 门数量的增加和生产工艺的进步, 使得 FPGA 的设计走到技术的前沿。FPGA 已经不再仅仅作为设计原型平台, 今天的数百万门的 FPGA 器件采用先进的 65nm 工艺生产, 价位也具有相当的竞争力, 完全能够支持高性能、大批量产品的设计。考虑到传统的高度灵活性和可编程优点, 以及设计方便性, FPGA 在许多情况下已经成为最佳的选择, 可广泛用于计算机、通信、消费电子和汽车等领域中众多要求苛刻且成本敏感的场所。因此 FPGA 设计工具必须跟上相应器件的发展。同时, 设计领域也在不断融合, 因此设计团队需要开发出满足用户需求的综合解决方案。通过一个集成开发环境, 完成逻辑、DSP 和嵌入式应用设计, 可以提高生产效率, 并通过片上系统 (SoC) FPGA, 促进真正的系统级设计。

Xilinx 公司是全球最大的可编程逻辑器件制造商, 在芯片设计领域和 FPGA 设计工具方面一直引导最新潮流。此次创新性地为其屡获殊荣的、被广泛运用的 ISE 工具套件推出了 10.1 版本, 从而再次为设计解决方案确立了新标准, 并为其最新的高性能 Virtex 5 和低成本 Spartan 3 FPGA 提供了更强大的支持。通过新推出的 ISE Design Suite 10.1, Xilinx 公司解决了采用高级 FPGA 进行设计的设计师所面对的严峻挑战, 并且第一次提供了一个统一的逻辑、DSP 和嵌入式应用设计人员需要的解决方案。

本书主要涉及 ISE Design Suite 中有关逻辑设计的内容, 分为 7 章。第 1 章介绍 FPGA 开发的基础知识, 包括 Xilinx FPGA 的器件架构和开发流程。第 2 章介绍 Xilinx FPGA 开发资源, 涉及 Xilinx 文档类型、FPGA 开发人员进阶路线以及 Xilinx 最新的 FPGA 芯片和 Xilinx 公司在诸多领域的解决方案。第 3 章简要介绍了 ISE Design Suite 10.1 套件的创新特点和主要组件。第 4 章详细讲解了逻辑开发工具 ISE 的使用方法, 这些开发工具包括 HDL 输入工具、原理图输入工具、状态机生成器、仿真工具、综合工具、实现工具、用户约束文件、各类约束编辑工具以及 FPGA 下载工具。第 5 章主要针对 ISE 中的高级开发工具 (如时序分析器、布局规划器、平面布局工具、FPGA 底层编辑器、功耗分析器以及在线调试工具) 进行详细讲解。有效使用这些工具, 可极大提高设计性能, 并提高设计生产力。第 6 章主要讲解第三方设计工具, 包括综合工具 Synplify、仿真工具 Modelsim 和非常流行的 MATLAB 软件。第 7 章给出了 FPGA 底层单元和 LMS 算法的逻辑开发实例, 期望从工程开发角度给予读者尽可能多的参考。

本书适合从事 Xilinx 系列 FPGA 设计和开发工作的工程师, 以及通信、电子和计算机等相关专业的研究生和高年级本科生使用。

本书的附带光盘包含了 ISE Design Suite 10.1 完整的 Web 版软件和本书所有工程实例的完整源文件。我们希望可以帮助读者的学习提供更多的方便。

在本书的筹划和写作过程中, 得到了 Xilinx 公司亚太区公共关系经理张俊伟女士的诸多帮助, 并最终促成本书的编写; Xilinx 公司中国区大学计划经理谢凯年先生提供了最新版的 ISE 软件; Xilinx 公司亚太区高级产品经理梁晓明先生审阅了全书内容, 并提出很多建设性意见, 极大地提高了本书的质量; 人民邮电出版社刘洋编辑在本书的定稿和审稿过程中付出了诸多努力, 才有了本书的迅速面世。此外, Openhard 社区 (www.openhw.org) 的贺满荃先生给作者共享了很多资源和参考设计实例, 在此一并表示衷心的感谢。

由于作者水平有限, 加上时间比较仓促, 书中不妥之处在所难免, 敬请读者批评指正。读者可通过电子邮件 (liuyang@ptpress.com.cn) 与本书责任编辑进行互动和交流。

目 录

第 1 章	FPGA 开发基础知识	1
1.1	可编程逻辑器件基础	1
1.1.1	可编程逻辑器件概述	1
1.1.2	可编程逻辑器件的发展历史	1
1.1.3	可编程逻辑器件开发工具	2
1.2	FPGA 器件的基础知识	3
1.2.1	FPGA 芯片的基本工作原理	3
1.2.2	Xilinx FPGA 的基本架构	4
1.2.3	典型的 FPGA 开发流程	8
1.2.4	基于 FPGA 的 SoC 设计方法	10
1.2.5	FPGA 芯片与设计的性能指标	11
1.3	本章小结	12
第 2 章	Xilinx FPGA 资源简介	13
2.1	Xilinx 最新 FPGA	13
2.1.1	高端平台 Virtex 5 系列	13
2.1.2	中低端平台 Spartan 3E 系列	24
2.2	Xilinx 主流芯片介绍	26
2.2.1	Xilinx FPGA 芯片介绍	26
2.2.2	Xilinx PROM 芯片介绍	32
2.2.3	Xilinx 芯片的选型	33
2.3	Xilinx FPGA 的开发资源	34
2.3.1	Xilinx FPGA 在通信领域中的解决方案	34
2.3.2	Xilinx FPGA 在汽车电子领域中的解决方案	36
2.3.3	Xilinx FPGA 在工业/科学/医疗领域中的解决方案	37
2.3.4	Xilinx FPGA 在宇航和国防领域中的解决方案	37
2.3.5	Xilinx FPGA 在其他领域中的解决方案	38
2.4	Xilinx FPGA 的电子文档资源	38
2.4.1	Xilinx 官方文档分类	38
2.4.2	ISE 软件自带文档	39
2.4.3	Xilinx 电子文档资源的搜索技巧	40
2.4.4	FPGA 设计人员的进阶路线	42
2.5	Xilinx FPGA 芯片管脚解读	43
2.5.1	FPGA 芯片的封装形式	43

2.5.2	FPGA 芯片的管脚介绍	45
2.5.3	FPGA 芯片管脚的分配策略	49
2.6	本章小结	49
第 3 章	新一代开发工具 ISE Design Suite 10.1	51
3.1	ISE Design Suite 10.1 简介	51
3.1.1	ISE Design Suite 10.1 综述	51
3.1.2	ISE Design Suite 10.1 的创新特性	51
3.2	ISE Design Suite 10.1 主要组件	53
3.2.1	ISE Foundation	53
3.2.2	EDK 开发工具	55
3.2.3	DSP 工具	55
3.2.4	ChipScope Pro	56
3.2.5	PlanAhead	56
3.3	本章小结	57
第 4 章	ISE Foundation 基本组件	59
4.1	ISE Foundation 的介绍与安装	59
4.1.1	ISE Foundation 简要介绍	59
4.1.2	ISE 软件的安装	59
4.1.3	ISE 软件的基本操作	62
4.1.4	ISE 软件的开发操作流程	66
4.2	基于 ISE 的工程建立与设计输入	67
4.2.1	新建工程	67
4.2.2	HDL 代码输入	68
4.2.3	状态机的输入与验证	70
4.2.4	原理图输入法	78
4.2.5	代码模板的使用	80
4.2.6	Xilinx IP Core 的使用	81
4.3	基于 ISE 的仿真	86
4.3.1	基于波形测试法的仿真	86
4.3.2	基于 HDL 测试代码的仿真	89
4.4	基于 ISE 的综合与实现	90
4.4.1	基于 Xilinx XST 的综合	91
4.4.2	基于 ISE 的实现	96
4.4.3	基于目标和用户策略的设计方法	104
4.4.4	基于 SmartXplorer/Xplorer 的实现技术	107
4.4.5	基于 SmartCompile 的设计保存技术	112

4.5	用户约束文件	116
4.5.1	约束文件的基本知识	117
4.5.2	UCF 文件的语法说明	117
4.5.3	管脚和区域约束语法	118
4.5.4	时序约束语法	120
4.6	管脚和区域约束工具 Floorplan Editor	124
4.6.1	Floorplan Editor 功能简介	125
4.6.2	利用 PACE 完成管脚分配	125
4.6.3	使用 Floorplan Editor 完成管脚分配和区域约束	127
4.6.4	Floorplan Editor 的其他功能	129
4.7	时序约束工具 Constraints Editor	132
4.7.1	Constraints Editor 功能简介	132
4.7.2	利用 Constraints Editor 添加时序约束	134
4.7.3	利用 Constraints Editor 添加分组约束	139
4.7.4	利用 Constraints Editor 添加专用约束	140
4.8	基于 ISE 的器件配置	141
4.8.1	FPGA 配置电路	141
4.8.2	iMPACT 参数设置	145
4.8.3	配置 FPGA 器件	150
4.8.4	配置 PROM 器件	153
4.9	本章小结	158
第 5 章	ISE Foundation 高级组件	159
5.1	在线逻辑分析仪 ChipScope Pro	159
5.1.1	ChipScope Pro 工具简介	159
5.1.2	ChipScope Core Generator 使用说明	162
5.1.3	ChipScope Core Inserter 使用说明	167
5.1.4	ChipScope Core Analyzer 使用说明	172
5.1.5	ChipScope Pro Serial I/O Toolkit 使用说明	175
5.1.6	ChipScope Pro 应用实例	186
5.2	平面布局规划器 PlanAhead	192
5.2.1	PlanAhead 10.1 的安装及新特性	193
5.2.2	PlanAhead 设计流程	194
5.2.3	利用 PlanAhead 进行 I/O 引脚规划	199
5.2.4	使用 ExploreAhead 优化实现结果	207
5.3	时序分析器 Timing Analyzer	213
5.3.1	时序分析基础	213
5.3.2	Xilinx FPGA 中的时钟资源	220
5.3.3	ISE 时序分析器的软件操作	224

5.3.4	Timing Analyzer 应用实例	237
5.4	布局规划器 Floorplanner	238
5.4.1	Floorplanner 简介	238
5.4.2	Floorplanner 软件操作	240
5.4.3	Floorplanner 应用实例	248
5.5	底层编辑器 FPGA Editor	250
5.5.1	FPGA Editor 简介	250
5.5.2	FPGA Editor 软件操作	251
5.5.3	FPGA Editor 应用实例	262
5.6	功耗分析工具 XPower	264
5.6.1	功耗分析简介	264
5.6.2	XPower 估计器	265
5.6.3	XPower 分析器	269
5.6.4	低功耗设计技术	275
5.6.5	XPower 分析器应用实例	277
5.7	本章小结	278
第 6 章	ISE 与第三方软件	279
6.1	ModelSim 仿真软件的使用	279
6.1.1	ModelSim 仿真软件的安装	279
6.1.2	在 ModelSim 中指定 Xilinx 的仿真库	281
6.1.3	ModelSim 的基本操作	282
6.1.4	ModelSim 的高级操作	284
6.2	综合工具 Synplify Pro	296
6.2.1	Synplify Pro 综合软件的安装	296
6.2.2	Synplify Pro 的使用	297
6.3	ISE 与 MATLAB 的联合使用	302
6.3.1	利用 MATLAB 辅助 FPGA 的逻辑设计	303
6.3.2	利用 MATLAB 完成 DSP 系统开发	304
6.3.3	利用 MATLAB 自动生成滤波器代码	306
6.4	本章小结	309
第 7 章	FPGA 底层单元与逻辑开发实例	311
7.1	FPGA 底层单元开发	311
7.1.1	Xilinx 全局时钟网络的使用	311
7.1.2	DCM 模块的使用	312
7.1.3	Xilinx 内嵌块存储器的使用	320
7.1.4	硬核乘加器 DSP48 的使用	330

7.2	FPGA 常用 IP Core 使用实例	338
7.2.1	Cordic 算法 IP Core 的使用	338
7.2.2	FFT 算法 IP Core 的使用	344
7.2.3	FIR 滤波器 IP Core 的使用	356
7.3	开发实例——LMS 算法的 Verilog 实现	370
7.3.1	LMS 算法的原理	370
7.3.2	LMS 算法的 MATLAB 实现	373
7.3.3	LMS 算法的 FPGA 实现	375
7.3.4	LMS 算法的软件调试	385
7.4	本章小结	391
附录	Verilog HDL 语言基础	393
参考文献		409

第 1 章 FPGA 开发基础知识

现场可编程门阵列 (Field Programmable Gate Array, FPGA) 属于可编程逻辑器件的一种, 在 20 世纪 90 年代获得了突飞猛进的发展, 经过近 20 年的发展, 到目前已成为实现数字系统的主流平台之一。本章主要介绍 FPGA 的起源、发展历史、芯片结构、工作原理、开发流程以及 FPGA 芯片与设计的性能指标, 为读者提供 FPGA 设计的基础知识。

1.1 可编程逻辑器件基础

1.1.1 可编程逻辑器件概述

可编程逻辑器件 (Programmable Logic Device, PLD) 起源于 20 世纪 70 年代, 是在专用集成电路 (ASIC) 的基础上发展起来的一种新型逻辑器件, 是当今数字系统设计的主要硬件平台, 其主要特点是完全由用户通过软件进行配置和编程, 从而完成某种特定的功能, 且可以反复擦写。修改和升级 PLD 不需额外地改变 PCB 电路板, 只是在计算机上修改和更新程序, 使硬件设计工作转化为软件开发工作, 缩短了系统设计的周期, 提高了实现的灵活性并降低了成本, 因此获得了广大硬件工程师的青睐, 形成了巨大的 PLD 产业规模。

目前常见的 PLD 产品有: 可编程只读存储器 (Programmable Read Only Memory, PROM)、现场可编程逻辑阵列 (Field Programmable Logic Array, FPLA)、可编程阵列逻辑 (Programmable Array Logic, PAL)、通用阵列逻辑 (Generic Array Logic, GAL)、可擦除的可编程逻辑器件 (Erasable Programmable Logic Array, EPLA)、复杂可编程逻辑器件 (Complex Programmable Logic Device, CPLD) 和现场可编程门阵列 (Field Programmable Gate Array, FPGA) 等类型。PLD 器件从规模上又可以细分为简单 PLD (SPLD)、复杂 PLD (CPLD) 以及 FPGA。它们内部结构的实现方法各不相同。

可编程逻辑器件按照颗粒度可以分为 3 类: 小颗粒度 (如“门海 (Sea of Gates)”架构), 中等颗粒度 (如 FPGA) 以及大颗粒度 (如 CPLD)。按照编程工艺可以分为 4 类: 熔丝 (Fuse) 和反熔丝 (Antifuse) 编程器件、可擦除的可编程只读存储器 (EPROM) 编程器件、电信号可擦除的可编程只读存储器 (EEPROM) 编程器件 (如 CPLD) 和静态 RAM (SRAM) 编程器件 (如 FPGA)。在工艺分类中, 前 3 类为非易失性器件, 编程后配置数据保留在器件上; SRAM 为易失性器件, 掉电后配置数据会丢失, 因此在每次上电后需要重新进行数据配置。

1.1.2 可编程逻辑器件的发展历史

可编程逻辑器件的发展可以划分为 4 个阶段, 从 20 世纪 70 年代初期到 70 年代中期为第 1 阶段, 20 世纪 70 年代中期到 80 年代中期为第 2 阶段, 20 世纪 80 年代中期到 90 年代末为

第3阶段, 20世纪90年代末到目前为第4阶段。

第1阶段的可编程器件只有简单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(EEPROM)3种, 由于结构的限制, 它们只能完成简单的数字逻辑功能。

第2阶段出现了结构上稍微复杂的可编程阵列逻辑(PAL)和通用阵列逻辑(GAL)器件, 正式被称为PLD, 能够完成各种逻辑运算功能。典型的PLD由“与”、“非”阵列组成, 用“与或”表达式来实现任意组合逻辑, 所以PLD能以乘积和形式完成大量的逻辑组合。

第3阶段由Xilinx和Altera分别推出了与标准门阵列类似的FPGA以及类似于PAL结构的扩展性CPLD, 提高了逻辑运算的速度, 具有体系结构和逻辑单元灵活、集成度高以及适用范围宽等特点, 兼容了PLD和通用门阵列的优点, 能够实现超大规模的电路, 编程方式也很灵活, 成为产品原型设计和中小规模(一般小于10000门)产品生产的首选。这一阶段, CPLD、FPGA器件在制造工艺和产品性能方面都获得了长足的发展, 达到了0.18 μm 工艺和系统门数百万门的规模。

第4阶段出现了可编程的片上系统(System On Programmable Chip, SOPC)和片上系统(System on a Chip, SoC)技术, 是PLD和ASIC技术融合的结果, 涵盖了实时化数字信号处理技术、高速数据收发器、复杂计算以及嵌入式系统设计技术的全部内容。各FPGA厂商都推出了相应的SoC FPGA产品, 制造工艺达到65nm, 系统门数也超过百万门。并且, 这一阶段的逻辑器件内嵌了硬核高速乘法器、吉比特差分串行接口、时钟频率高达数百兆赫的PowerPC微处理器以及软核处理器PicoBlaze与MicroBlaze等, 不仅实现了软件需求和硬件设计的完美结合, 还实现了高速与灵活性的完美结合, 已超越了ASIC器件的性能和规模, 也超越了传统意义上FPGA的概念, 使PLD的应用范围从单片扩展到系统级。目前, 基于PLD片上可编程的概念仍在不断向前发展。

1.1.3 可编程逻辑器件开发工具

基于高复杂度PLD器件的开发, 在很大程度上要依靠电子设计自动化(EDA)来完成。PLD的EDA工具以计算机软件为主, 将典型的单元电路封装起来形成固定模块并形成标准的硬件开发语言(如HDL语言)供设计人员使用。设计人员考虑如何将可组装的软件库和软件包搭建成为满足需求的功能模块甚至完整的系统。PLD开发软件需要自动地完成逻辑编译、化简、分割、综合及优化、布局布线、仿真以及对于特定目标芯片的适配编译和编程下载等工作。典型的EDA工具中必须包含两个特殊的软件包, 即综合器和适配器。综合器的功能是将设计者在EDA平台上完成的针对某个系统项目的HDL、原理图或状态图形描述, 针对给定的硬件系统组件, 进行编译、优化、转换和综合。适配器一般不用于FPGA设计。

随着开发规模的级数性增长, 就必须减短PLD开发软件的编译时间, 并提高其编译性能以及提供丰富的知识产权(IP核)资源供设计人员调用。此外, PLD开发界面的友好性以及操作的复杂程度也是评价其性能的重要因素。目前在PLD产业领域中, 各个芯片提供商的PLD开发工具已成为影响其成败的核心成分。只有全面做到芯片技术领先、文档完整和PLD开发软件优秀, 芯片提供商才能获得客户的认可。一个完美的PLD开发软件应当具备以下特点:

① 准确地将用户设计转换为电路模块;

② 能够高效地利用器件资源;

③ 能够快速地完成编译和综合;

④ 满足用户在不同领域的设计需求，既全面又高效；

⑤ 提供丰富的 IP 资源；

⑥ 用户界面友好，操作简单。

Xilinx 公司的 ISE 是业界公认的优秀集成 PLD 开发软件，可完成上百款可编程逻辑器件的开发。此外，综合软件 Synplify、仿真软件 ModelSim 以及 MATLAB/Simulink 等诸多第三方 EDK 开发软件也满足上述要求。

1.2 FPGA 器件的基础知识

1.2.1 FPGA 芯片的基本工作原理

如前所述，FPGA 是在 PAL、GAL、EPLD、CPLD 等可编程器件的基础上进一步发展的产物。它是作为 ASIC 领域中的一种半定制电路而出现的，既解决了定制电路的不足，又克服了原有可编程器件门电路有限的缺点。

由于 FPGA 需要被反复烧写，它实现组合逻辑的基本结构不可能像 ASIC 那样通过固定的与非门来完成，而只能采用一种易于反复配置的结构；查找表可以很好地满足这一要求。目前主流 FPGA 都采用了基于 SRAM 工艺的查找表结构，也有一些军品和宇航级 FPGA 采用 Flash 或者熔丝与反熔丝工艺的查找表结构。SRAM 工艺的 FPGA 芯片不具备非易失特性，因此断电后将丢失内部逻辑配置。在每次上电后，都需要从外部非易失存储器（PROM、Flash 存储器等）中导入配置比特流。

根据数字电路的基本知识可以知道，对于一个 n 输入的逻辑运算，不管是与或非运算还是异或运算等，最多只可能存在 $2n$ 种结果。所以，如果事先将相应的结果存放于一个存储单元，就相当于实现了与非门电路的功能。FPGA 的原理也是如此，它通过烧写文件去配置查找表的内容，从而在相同的电路情况下实现不同的逻辑功能。

查找表（Look-Up-Table, LUT）本质上就是一个 RAM。目前 Xilinx 的最新产品 Virtex 5 FPGA 芯片采用 6 输入的 LUT，而其余所有芯片都使用 4 输入的 LUT。一个 6/4 LUT 可以看成是一个有 6/4 位地址线的 64/16×1 的 RAM。当用户通过原理图或 HDL 语言描述了一个逻辑电路以后，PLD/FPGA 开发软件会自动计算逻辑电路的所有可能结果，并把真值表（即结果）事先写入 RAM，这样，每输入一个信号进行逻辑运算就等于输入一个地址进行查表，找出地址对应的内容，然后输出即可。

下面给出一个 4 与门电路的例子来说明 LUT 实现逻辑功能的原理。

例 1-1 给出一个使用 LUT 实现 4 输入与门电路的真值表，见表 1-1。

表 1-1

4 输入与门的真值表

实际逻辑电路		LUT 的实现方式	
a, b, c, d 输入	逻辑输出	RAM 地址	RAM 中存储的内容
0000	0	0000	0
0001	0	0001	0
...	...	...	...
1111	1	1111	1

从表 1-1 中可以看到, LUT 具有和逻辑电路相同的功能。实际上, LUT 具有更快的执行速度和更大的规模。

由于基于 LUT 的 FPGA 具有很高的集成度, 其器件密度从数万门到数千万门不等, 可以完成极其复杂的时序逻辑电路与组合逻辑电路, 因此适用于高速、高密度的高端数字逻辑电路设计领域。其组成部分主要包括可编程输入/输出单元、基本可编程逻辑单元、内嵌 SRAM、丰富的布线资源、底层嵌入功能单元、内嵌专用单元等, 主要设计和生产厂家有 Xilinx、Altera、Lattice、Actel、Atmel 和 QuickLogic 等公司, 其中最大的是 Xilinx、Altera、Lattice 3 家。

如前所述, FPGA 是由存放在片内的 RAM 来设置其工作状态的, 因此工作时需要对片内 RAM 进行编程。用户可根据不同的配置模式, 采用不同的编程方式。FPGA 有如下几种配置模式。

① 并行模式: 并行 PROM、Flash 配置 FPGA。

② 主从模式: 一片 PROM 配置多片 FPGA。

③ 串行模式: 串行 PROM 配置 FPGA。

④ 外设模式: 将 FPGA 作为微处理器的外设, 由微处理器对其编程。

目前, FPGA 市场占有率最高的两大公司 Xilinx 和 Altera 生产的 FPGA 都是基于 SRAM 工艺的, 需要在使用时外接一个片外存储器以保存程序。上电时, FPGA 将外部存储器中的数据读入片内 RAM, 完成配置后, 进入工作状态; 掉电后 FPGA 恢复为白片, 内部逻辑消失。这样 FPGA 不仅能反复使用, 还无需专门的 FPGA 编程器, 只需通用的 EPROM、PROM 编程器即可。Actel、QuickLogic 等公司还提供反熔丝技术的 FPGA, 只能下载一次, 具有抗辐射、耐高低温、低功耗和速度快等优点, 在军品和航空航天领域中应用较多, 但这种 FPGA 不能重复擦写, 开发初期比较麻烦, 费用也比较昂贵。Lattice 是在系统可编程 (In-System Programmability, ISP) 技术的发明者, 在小规模 PLD 应用上有一定的特色。早期的 Xilinx 产品一般不涉及军品和宇航级市场, 但目前已经有 Q Pro-R 等多款产品进入该领域。

1.2.2 Xilinx FPGA 的基本架构

目前 FPGA 芯片仍是基于查找表技术的, 但其概念和性能已经远远超出查找表技术的限制, 并且整合了常用功能的硬核模块 (如块 RAM、时钟管理和 DSP)。图 1-1 所示为 Xilinx 公司 FPGA 芯片的内部结构示意图 (由于不同系列的应用场合不同, 因此内部结构会有一些调整)。从图 1-1 中可以看出, FPGA 芯片主要由 6 部分组成, 即可编程输入输出单元、基本可编程逻辑单元、完整的时钟管理、嵌入式块 RAM、丰富的布线资源、内嵌的底层功能单元和内嵌专用硬件模块。

下面详细讲解每个模块的功能。

1. 可编程输入输出单元 (IOB)

可编程输入/输出单元简称 I/O 单元, 是芯片与外界电路的接口部分, 完成不同电气特性下对输入/输出信号的驱动与匹配要求, 提供输入缓冲、输出驱动、接口电平转换、阻抗匹配以及延迟控制等功能, 其一般结构如图 1-2 所示。FPGA 内的 I/O 按组分类, 每组都能够独立地支持不同的 I/O 标准。通过软件的灵活配置, 可适配不同的电气标准与 I/O 物理特性, 可以调整驱动电流的大小, 可以改变上、下拉电阻。目前, I/O 口的频率也越来越高, 一些高

端的 FPGA 通过 DDR 寄存器技术可以支持高达 2Gbit/s 的数据传输速率。

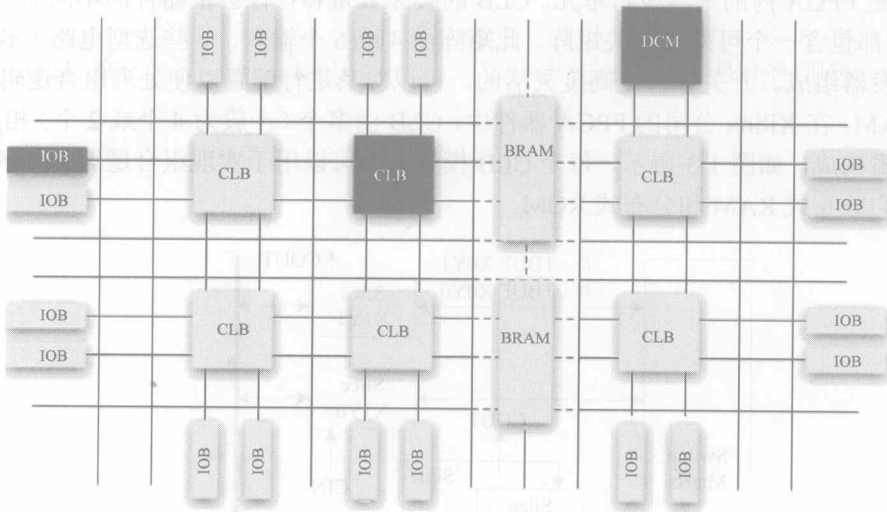


图 1-1 FPGA 芯片的内部结构

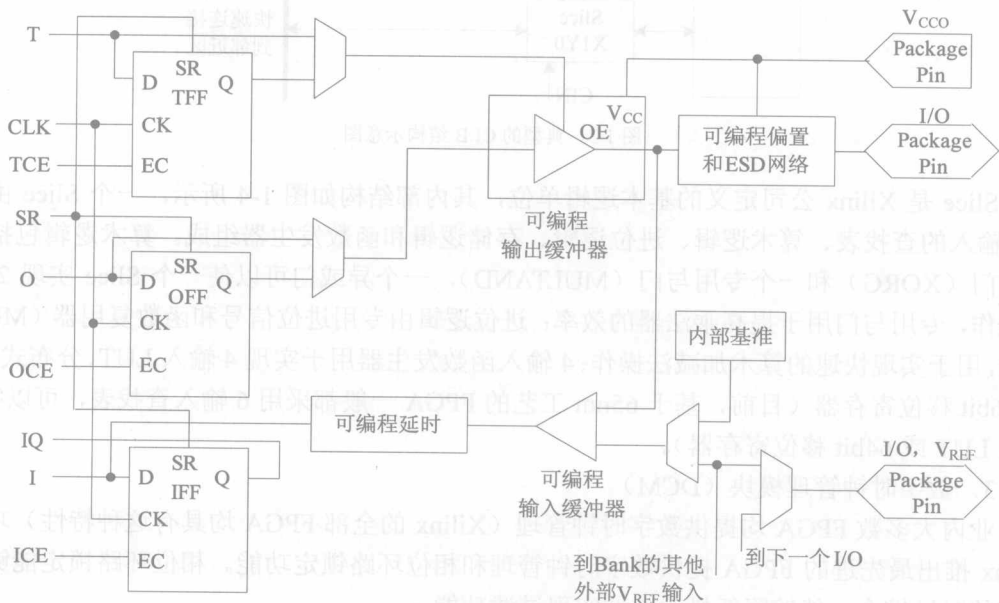


图 1-2 典型的 IOB 内部结构示意图

外部输入信号可以通过 IOB 模块的存储单元输入到 FPGA 的内部，也可以直接输入到 FPGA 内部。当外部输入信号经过 IOB 模块的存储单元输入到 FPGA 内部时，其保持时间(Hold Time)的要求可以降低，通常默认为 0。

为了便于管理和适应多种电气标准，FPGA 的 IOB 被划分为若干个组 (Bank)，每个 Bank 的接口标准由其接口电压 V_{CCO} 决定，一个 Bank 只能有一种 V_{CCO} ，但不同 Bank 的 V_{CCO} 可以不同。只有相同电气标准的端口才能连接在一起， V_{CCO} 电压相同是接口标准的基本条件。

2. 可配置逻辑块 (CLB)

CLB 是 FPGA 内的基本逻辑单元。CLB 的实际数量和特性会依器件的不同而不同,但是每个 CLB 都包含一个可配置开关矩阵,此矩阵由 4 或 6 个输入、一些选型电路(多路复用器等)和触发器组成。开关矩阵是高度灵活的,可以对其进行配置以便处理组合逻辑、移位寄存器或 RAM。在 Xilinx 公司的 FPGA 器件中,CLB 由多个(一般为 4 个或 2 个)相同的 Slice 和附加逻辑构成,如图 1-3 所示。每个 CLB 模块不仅可以用于实现组合逻辑、时序逻辑,还可以配置为分布式 RAM 和分布式 ROM。

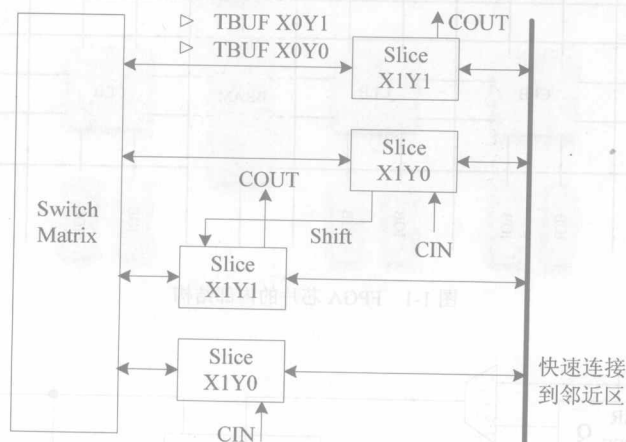


图 1-3 典型的 CLB 结构示意图

Slice 是 Xilinx 公司定义的基本逻辑单位,其内部结构如图 1-4 所示,一个 Slice 由两个 4/6 输入的查找表、算术逻辑、进位逻辑、存储逻辑和函数发生器组成。算术逻辑包括一个异或门 (XORG) 和一个专用与门 (MULTAND),一个异或门可以使一个 Slice 实现 2bit 全加操作,专用与门用于提高乘法器的效率;进位逻辑由专用进位信号和函数复用器 (MUXC) 组成,用于实现快速的算术加减法操作;4 输入函数发生器用于实现 4 输入 LUT、分布式 RAM 或 16bit 移位寄存器(目前,基于 65nm 工艺的 FPGA 一般都采用 6 输入查找表,可以实现 6 输入 LUT 或 64bit 移位寄存器)。

3. 数字时钟管理模块 (DCM)

业内大多数 FPGA 均提供数字时钟管理 (Xilinx 的全部 FPGA 均具有这种特性) 功能。Xilinx 推出最先进的 FPGA 提供数字时钟管理和相位环路锁定功能。相位环路锁定能够提供精确的时钟综合,能够降低抖动,并实现过滤功能。

4. 嵌入式块 RAM (BRAM)

大多数 FPGA 都具有内嵌的块 RAM,这大大拓展了 FPGA 的应用范围和灵活性。块 RAM 可被配置为单端口 RAM、双端口 RAM、内容地址存储器 (CAM) 以及 FIFO 等常用存储结构。RAM、FIFO 是比较普及的概念,在此不再赘述。CAM 存储器在其内部的每个存储单元中都有一个比较逻辑,写入 CAM 中的数据会和内部的每一个数据进行比较,并返回与端口数据相同的所有数据的地址,因而在路由的地址交换器中有广泛的应用。除了块 RAM,还可以将 FPGA 中的 LUT 灵活地配置成 RAM、ROM 和 FIFO 等结构。在实际应用中,芯片内部块 RAM 的数量也是选择芯片的一个重要因素。

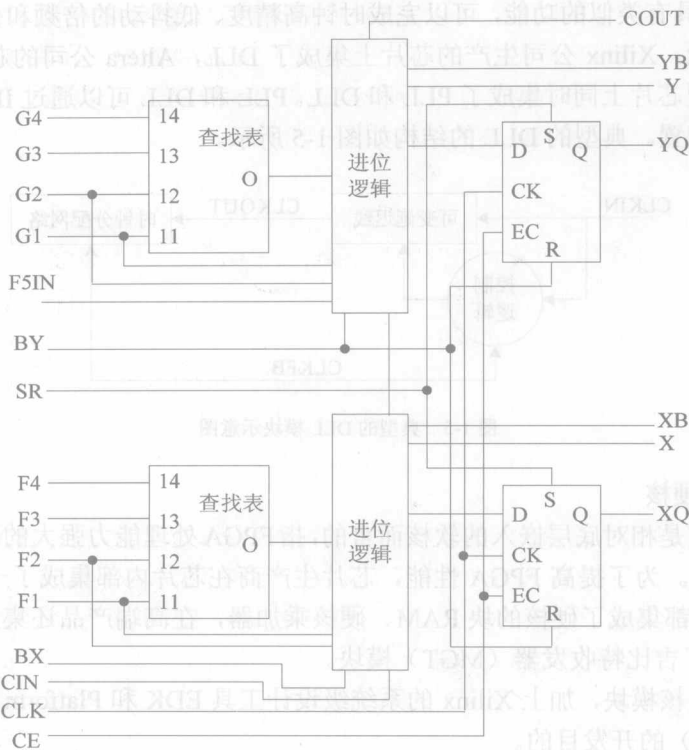


图 1-4 典型的 4 输入 Slice 结构示意图

单片块 RAM 的容量为 18kbit，即位宽为 18bit，深度为 1024。可以根据需要改变其位宽和深度，但要满足两个原则：首先，修改后的容量（位宽×深度）不能大于 18kbit；其次，位宽最大不能超过 36bit。当然，可以将多片块 RAM 级联起来形成更大的 RAM，此时只受限于芯片内块 RAM 的数量，而不再受上面两条原则的约束。

5. 丰富的布线资源

布线资源连通 FPGA 内部的所有单元，而连线的长度和工艺决定着信号在连线上的驱动能力和传输速度。FPGA 芯片内部有着丰富的布线资源，根据工艺、长度、宽度和分布位置的不同而划分为 4 类不同的类别。第一类是全局布线资源，用于芯片内部全局时钟和全局复位/置位的布线；第二类是长线资源，用以完成芯片 Bank 间的高速信号和第二全局时钟信号的布线；第三类是短线资源，用于完成基本逻辑单元之间的逻辑互连和布线；第四类是分布式的布线资源，用于专有时钟、复位等控制信号线。

在实际中设计者不需要直接选择布线资源，布局布线器可自动地根据输入逻辑网表的拓扑结构和约束条件选择布线资源来连通各个模块单元。从本质上讲，布线资源的使用方法和设计的结果有密切、直接的关系。

6. 底层内嵌功能单元

内嵌功能模块主要指 DLL (Delay Locked Loop)、PLL (Phase Locked Loop)、DSP 和 CPU 等软处理核，即内嵌处理器 (Embedded Processor)。正是集成了丰富的内嵌功能单元，才使得单片 FPGA 成为了系统级的设计工具，具备了软硬件联合设计的能力，逐步向 SoC 平台过渡。