

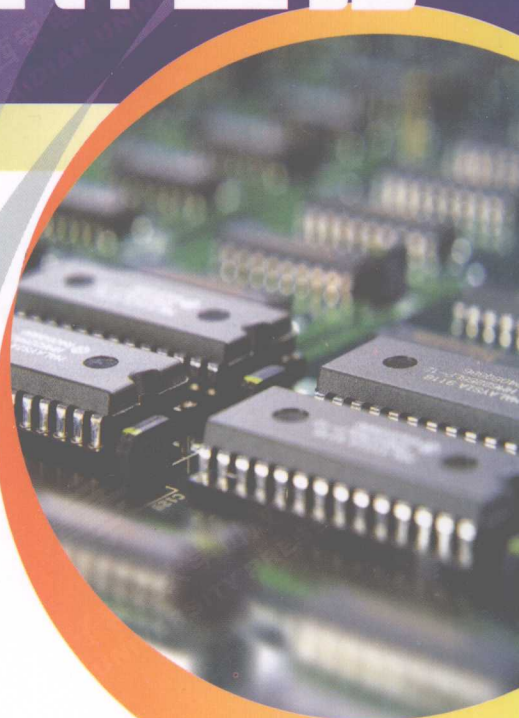


高等学校信息工程类专业规划教材

Xilinx FPGA 设计基础

(VHDL版)

李云松 宋 锐
雷 杰 杜建超 编著



西安电子科技大学出版社
<http://www.xduph.com>

面向 21 世纪高等学校信息工程类专业规划教材

Xilinx FPGA 设计基础 (VHDL 版)

李云松 宋锐 雷杰 杜建超 编著

西安电子科技大学出版社

2008

内 容 简 介

本书系统地介绍了 Xilinx 公司 FPGA 的结构特点和相关开发软件的使用方法, 详细描述了 VHDL 语言的语法和设计方法, 并深入讨论了 Xilinx FPGA 相关硬件的设计问题。

全书共分为七章。第 1 章概要介绍了 FPGA 知识。第 2 章介绍了 Xilinx FPGA 开发的常用软件及一般的开发流程。第 3 章详细讲述了 VHDL 硬件描述语言。第 4 章讨论了 Xilinx FPGA 开发中常用的 IP 核及其使用方法, 并着重描述了时钟管理 IP 核的参数配置。第 3 章和第 4 章的内容是 Xilinx FPGA 开发的基础。第 5 章讲述了 Xilinx FPGA 的相关硬件设计方法, 并给出了参考电路。第 6 章和第 7 章是实验部分, 包括程序设计实验和 FPGA 逻辑设计实验。

本书是在 Xilinx 公司大学计划的支持下完成的, 内容浅显易懂, 简洁精炼, 实验可操作性强, 是 Xilinx FPGA 开发的入门教材, 可作为高等院校电子类和通信类专业本科生、研究生 EDA 课程的教材。

图书在版编目 (CIP) 数据

Xilinx FPGA 设计基础: VHDL 版 / 李云松等编著.

—西安: 西安电子科技大学出版社, 2008.2

面向 21 世纪高等学校信息工程类专业规划教材

ISBN 978-7-5606-1959-0

I. X… II. 李… III. ① 可编程序逻辑器件—系统设计—高等学校: 技术学校—教材
② 硬件描述语言, VHDL—程序设计—高等学校: 技术学校—教材 IV. TP332.1 TP312

中国版本图书馆 CIP 数据核字 (2007) 第 201262 号

策 划 戚文艳

责任编辑 戚文艳

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

<http://www.xduph.com> E-mail: xdupfb@pub.xaonline.com

经 销 新华书店

印刷单位 陕西天意印务有限责任公司

版 次 2008 年 2 月第 1 版 2008 年 2 月第 1 次印刷

开 本 787 毫米×1092 毫米 1/16 印张 21.75

字 数 513 千字

印 数 1~4000 册

定 价 37.00 元(含光盘)

ISBN 978-7-5606-1959-0 / TN·0398

XDUP 2251001-1

*** 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

可编程逻辑器件作为一类特殊的集成电路芯片，其功能可由用户通过编程加以实现并可在线修改。设计人员可利用价格低廉的软件工具快速开发、仿真和测试其设计，并可快速将设计结果加载到器件中，在实际运行的电路中对整个系统进行完整测试。如果需要修改，只需重复以上开发过程而无需修改电路，这样就大大缩短了系统的研制周期并有效降低了产品的制造成本和设计成本。

过去几年时间里，可编程逻辑器件取得了巨大的技术进步。在芯片的集成度、处理速度以及系统功能等各个方面都取得了显著的进展。如今，可编程逻辑器件已经广泛应用于通信、自动控制、信息处理、大型计算机以及专用集成电路前端功能验证等诸多领域。熟练掌握可编程逻辑器件的设计技术已经是对电子设计工程师的基本要求。

本书的作者有多年从事 FPGA 教学和科研的经验，在负责的多项国家重点项目中使用了 Xilinx 公司的 FPGA 作为核心器件，深切感受到 FPGA 技术是提高我国数字电路自主设计的一项关键技术，对提高我国数字电路自主设计能力、产品开发能力和超大规模专用集成电路器件设计能力具有十分重要的意义。

本书系统地介绍了可编程逻辑器件的基本知识，着重讲述了 Xilinx 公司 FPGA 的结构特点、相关软件的使用方法以及 VHDL 语言设计方法。为了增强本书的实用性，提高读者的实际动手能力，本书结合 Xilinx 公司捐赠的大学计划开发板 Spartan-3E 给出了大量实验供读者参考练习。本书共分为七章。第 1 章介绍了可编程逻辑器件的发展历程、分类、基本结构和 Xilinx 公司主要的 FPGA 产品。第 2 章介绍了 Xilinx FPGA 开发的通用软件 ISE 和常用第三方软件 Modelsim 的使用方法以及一般的开发流程。第 3 章详细讲述了 VHDL 硬件描述语言。读者在学习第 3 章内容的过程中可以对照书中的指导逐步完成这部分实验内容。第 4 章简要讨论了 Xilinx FPGA 开发过程中可用的 IP 核及其使用方法，并着重描述了时钟管理 IP 核的配置参数。第 5 章讲述了 Xilinx FPGA 的配置和电源设计，并给出了参考电路。第 6 章是 VHDL 程序设计实验。第 7 章是 Xilinx XUP 提供的针对 Spartan-3E 开发板的逻辑设计实验，涵盖了 Xilinx FPGA 开发的整个流程。实验内容的开发环境为 ISE，读者可以从 Xilinx 公司网站上下载 ISE 的 60 天试用版，完成工程验证。本书配有光盘，光盘的内容包含书稿中实验部分的相关代码。

本书由李云松主编，第 1 章由杜建超、李云松编写，第 2、3、4、6 章由宋锐、李云松、杜建超编写，第 5、7 章由雷杰、李云松编写。西安电子科技大学吴成柯教授在百忙之中审阅了全书，提出了许多宝贵意见。

在本书的撰写过程中，得到了西安电子科技大学出版社和西安电子科技大学—Xilinx 联合实验室的大力支持。Xilinx 公司向本书作者提供了硬件开发板、开发软件以及其他技术资料。Xilinx 公司大学计划中国区负责人谢凯年博士为本书提供了许多技术资料和技术支持。西安工程大学张卫军老师参与了本书第 3 章部分内容的编写工作。西安电子科技大学杨立成、王玉国、赵嘉鑫、张兆亮、郭瑞等研究生对本书内容做了大量的校对工作，并对书中的例程和实验进行了验证。在此一并表示衷心的感谢！

由于作者水平有限，错误遗漏在所难免，恳请读者批评指正。

编 者

2007 年 10 月

目 录

第 1 章 绪论	1
1.1 逻辑器件概述	1
1.2 可编程逻辑器件 PLD 的发展历程	2
1.3 FPGA 的特点	4
1.4 CPLD/FPGA 的基本结构	5
1.4.1 CPLD 的基本结构	5
1.4.2 FPGA 的基本结构	7
1.4.3 CPLD 与 FPGA 比较	10
1.5 Xilinx FPGA 产品介绍	10
1.5.1 Xilinx 公司概况	10
1.5.2 Xilinx FPGA 产品	11
第 2 章 开发软件与开发流程	19
2.1 开发软件简介	19
2.1.1 ISE 开发软件简介	20
2.1.2 ISE9.1i 的安装	20
2.1.3 ModelSim 仿真软件简介	23
2.2 一个简单的开发项目	23
2.3 开发软件使用进阶	36
2.3.1 ISE9.1i 集成开发环境界面	37
2.3.2 设计输入	40
2.3.3 设计综合	51
2.3.4 功能仿真	53
2.3.5 工程实现	60
2.3.6 时序仿真	61
2.3.7 器件配置	64
第 3 章 VHDL 硬件描述语言	67
3.1 VHDL 的历史和概况	67
3.2 VHDL 基本设计思想	67
3.3 VHDL 语言设计的基本单元	69
3.3.1 实体	70
3.3.2 构造体	74
3.3.3 配置	78
3.3.4 包集合	82
3.3.5 库	83

3.4 VHDL 语言的对象和数据类型	85
3.4.1 VHDL 语言的对象类型	85
3.4.2 VHDL 语言的数据类型	88
3.4.3 不同数据类型之间的转换	98
3.5 VHDL 语言运算操作符	99
3.6 VHDL 语言的描述语句	103
3.6.1 有关规则和基本语句	104
3.6.2 并发描述语句	113
3.6.3 顺序描述语句	122
3.6.4 其他语句	130
3.7 VHDL 的层次结构设计	135
3.7.1 参数与参数配置	135
3.7.2 元件与元件例化	137
3.7.3 <i>generate</i> 语句	138
3.7.4 子程序(Subprogram)	143
3.7.5 VHDL 的行为级建模与 RTL 建模	146
3.8 有限状态机(FSM)	148
3.8.1 有限状态机(FSM)	148
3.8.2 一个 FSM 的 RTL 代码实例	155
第 4 章 Xilinx IP 核	158
4.1 Xilinx 的 IP 介绍	158
4.2 Xilinx IP 配置工具及使用方法	159
4.3 时钟管理 IP	164
4.3.1 DCM 模块	164
4.3.2 DCM 的使用方法	165
第 5 章 FPGA 的配置和电源设计	171
5.1 FPGA 的配置	171
5.1.1 FPGA 的配置引脚	171
5.1.2 FPGA 的配置模式	171
5.1.3 FPGA 的配置流程	172
5.1.4 FPGA 的配置电路	173
5.2 FPGA 的电源设计	175
5.2.1 FPGA 的电源指标	175
5.2.2 FPGA 的功耗估计	176
5.2.3 FPGA 的电源解决方案	177
第 6 章 VHDL 程序设计实验	181
6.1 实验一 层次化工程的创建	181
6.2 实验二 仿真测试平台的创建	188
6.3 实验三 存储器和记录类型实验	193

6.4 实验四 n 比特计数器及 RTL 验证实验	197
6.5 实验五 比较器实验	203
6.6 实验六 算术逻辑单元实验	208
6.7 实验七 状态机实验	213
6.8 实验八 计数器实验	220
6.9 实验九 IP 核应用实验	229
6.10 实验十 数字时钟管理 IP 核实验	239
第 7 章 FPGA 逻辑设计实验	253
7.1 实验一 熟悉 Xilinx 开发工具	253
7.2 实验二 结构体生成向导和 PACE	261
7.3 实验三 全局时钟约束实验	270
7.4 实验四 综合技巧实验	282
7.5 实验五 IP 核生成实验	287
7.6 实验六 Chipscope 调试实验	293
附录 A VHDL 关键字	302
附录 B VHDL 中的运算操作符	303
附录 C VHDL 中的描述语句及用法	304
附录 D VHDL 中的属性定义	310
附录 E IEEE 的标准库	313
参考文献	337

第1章

绪 论



1.1 逻辑器件概述

众所周知,在数字电子系统领域,存在三种基本的器件类型:存储器、微处理器和逻辑器件。这几种器件具有不同的功能和作用:存储器用来存储信息;微处理器用来执行指令以完成规定的任务;逻辑器件则用来提供数据通信、数据显示、接口通信、时序等一系列控制功能。

逻辑器件通常分为两大类——固定逻辑器件和可编程逻辑器件。固定逻辑器件是指为完成一组特定功能而实现的硬连接,其电路是永久性的,一旦制造完成,就无法改变。在整个开发过程中,设计和前期验证工作需要投入大量的“非重发性工程成本”,即 NRE(Non Recurrent Expense)。NRE 表示固定逻辑器件最终从芯片制造厂制造出来以前客户需要投入的所有成本。这些成本包括工程资源、软件设计工具、用来制造芯片不同金属层的昂贵光刻掩膜组,以及初始原型器件的生产成本。NRE 成本可能从数十万美元至数百万美元不等。固定逻辑器件适合批量生产,对有些需要极高性能的应用也是最佳选择。但是,如果客户需求发生了变化,或者器件工作不合适,就必须进行全新的设计开发。

可编程逻辑器件(PLD, Programmable Logic Device)是指可通过软件编程修改其逻辑电路、为客户提供具有多种功能的标准部件。设计者利用通用的 PLD 芯片经计算机编程,实现满足需求的电路系统,形成专用的集成电路芯片。这使得硬件设计变得非常灵活和方便。设计人员通过软件工具快速进行开发、仿真和测试。然后,将设计结果编程到器件中,并立即在实际运行的电路中进行测试。原型中使用的 PLD 器件与正式生产最终设备时所使用的 PLD 完全相同。这样就节约了 NRE 成本,设计周期也比采用定制固定逻辑器件时大大缩短。

此外,PLD 现在有越来越多的知识产权 IP 核的支持。用户可利用这些预定义和预测试的软件模块在 PLD 内迅速实现系统功能。IP 核包括从复杂数字信号处理算法和存储器控制器到总线接口和成熟的软件微处理器在内的一切内容。IP 核为客户节约了大量时间和费用。否则,用户可能需要数月的时间才能实现这些功能,而且还有可能延迟产品推向市场的时间。

总体来说,PLD 的优势可以概括如下:

(1) PLD 在设计过程中为客户提供了更大的灵活性,因为对于 PLD 来说,重复设计只需要简单地改变编程文件就可以了,而且设计时改变的结果可立即在工作器件中看到。

(2) PLD 不需要漫长的前置时间来制造原型或正式产品(PLD 器件已经放在分销商的货

架上并可随时付运。)

(3) PLD 不需要客户支付高昂的 NRE 成本和购买昂贵的掩膜组。PLD 供应商在设计其可编程器件时已经支付了这些成本, 并且可通过 PLD 产品线延续多年的生命期来分摊这些成本。

(4) PLD 允许客户在需要时仅订购所需要的数量, 从而使客户可以控制库存。

(5) PLD 甚至在设备交付客户那儿以后还可以重新编程。事实上, 由于有了可编程逻辑器件, 现在一些设备制造商正在尝试为已经成型的产品增加新功能或者进行升级。要实现这一点, 只需要通过因特网将新的编程文件下载到 PLD 就可以在系统中创建出新的硬件逻辑。

可编程逻辑器件缩短了电子产品制造商的开发周期, 帮助人们更快地将产品推向市场。随着 PLD 供应商继续致力于在可编程逻辑器件中集成更多的功能、降低其成本并提高能够节约时间的 IP 核心的可用性, 可编程逻辑器件将会得到更为广泛的应用。

1.2 可编程逻辑器件 PLD 的发展历程

PLD 由两种门阵列构成: 与门阵列和或门阵列。由于任意一个组合逻辑都可以用“与—或”表达式来描述, 因此 PLD 能以乘积和的形式完成大量的逻辑功能。根据复杂程度不同, PLD 可以分为简单低密度 PLD 和复杂高密度 PLD。根据包含芯片的不同, PLD 可分为三种类型: 基于可编程只读存储器 PROM 的 PLD、基于可编程阵列逻辑 PAL 的 PLD 和基于可编程逻辑阵列 PLA 的 PLD。

早期的可编程逻辑器件有可编程只读存储器(PROM)、紫外线可擦除只读存储器(EPROM)和电可擦除只读存储器(EEPROM)。这类器件由固定的“与”门阵列和可编程的“或”门阵列构成。PROM 采用熔丝工艺编程, 电路状态只能改变一次, 因此一旦写入内容, 就不能擦除和重写。EPROM 和 EEPROM 都是可重复写入的存储器, 易于编程, 但是价格较高, 写入速度较慢。由于结构的限制, 这些可编程逻辑器件只能完成简单的数字逻辑功能。

可编程阵列逻辑(PAL, Programmable Array Logic)于 20 世纪 70 年代出现, 它由可编程的“与”门阵列和固定的“或”门阵列构成。PAL 采用熔丝编程, 双极性工艺制造, 器件的工作速度很高。PAL 是第一个得到普遍应用的可编程逻辑器件, 如 PAL16L8。在 PAL 的基础上, 又发展了一种通用阵列逻辑(GAL, Generic Array Logic), 如 GAL16V8, GAL22V10 等。GAL 采用了 EEPROM 工艺, 实现了电可擦除、电可改写, 其输出结构是可编程的逻辑宏单元, 因而它的设计具有很强的灵活性, 至今仍有许多人使用。

可编程逻辑阵列(PLA, Programmable Logic Array)也是在 20 世纪 70 年代出现的, 它的“与”门阵列和“或”门阵列都是可编程的, 因而是最灵活的器件。但是由于资源利用率低、价格较贵、编程复杂, 因而没有得到广泛的应用。

这些早期的 PLD 器件都属于简单 PLD, 虽然其结构简单、设计灵活, 对开发软件的要求较低, 但其过于简单的结构使它们难以实现复杂的逻辑功能。随着技术的发展, 简单 PLD 在集成度和性能方面的局限性也暴露了出来, 其寄存器、I/O 引脚、时钟资源的数目有限, 没有内部连接。

20 世纪 80 年代中期, Altera 公司和 Xilinx 公司分别推出了类似于 PAL 结构的扩展型 CPLD(Complex Programmable Logic Device)和与标准门阵列类似的 FPGA(Field Programmable Gate Array),它们都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。同以往的 PAL、GAL 等相比较, CPLD / FPGA 的规模比较大,它可以替代几十甚至几千块通用 IC 芯片。这两种器件兼容了 PLD 和通用门阵列的优点,可实现较大规模的电路,编程也很灵活。与其他 ASIC(Application Specific IC)相比,它们又具有设计开发周期短、设计制造成本低、开发工具先进、标准产品无需测试、质量稳定以及可实时在线检验等优点,因此被广泛应用于产品的原型设计和小批量产品生产之中。几乎所有应用门阵列、PLD 和中小规模通用数字集成电路的场合均可应用 FPGA 和 CPLD 器件。

在这两类可编程逻辑器件中, CPLD 提供的逻辑资源较少,一般来说大约在 1000~7000 门之间,最高约 1 万门。常见的 CPLD 元件有 Xilinx 公司的 XC9500 系列和 CoolRunner 系列, Altera 公司的 Max5000 及 Max7000 系列等。但是, CPLD 提供了非常好的可预测性,因此对于关键产品的控制应用非常理想。而且如 Xilinx CoolRunner 系列, CPLD 器件需要的功耗极低,并且价格低廉,从而使其对于成本敏感的、电池供电的便携式应用(如移动电话和数字手持助理)非常理想。CPLD 至少包含三种结构:可编程逻辑宏单元、可编程 I/O 单元和可编程内部连线。部分的 CPLD 器件还集成了 RAM、FIFO 或双口 RAM 等存储器。

FPGA 提供了更高的逻辑密度、丰富的特性和更高的性能。现在最新的 FPGA 器件,如 Xilinx Virtex 系列中的部分器件,可提供八百万“系统门”(相对逻辑密度)。这些先进的器件还提供诸如内建的硬连线处理器(如 IBM Power PC)、大容量存储器、时钟管理系统等特性,并支持多种最新的超快速器件至器件(device-to-device)信号技术。FPGA 是作为专用集成电路领域中的一种半定制电路出现的,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数有限的缺点。现在, FPGA 的应用非常广泛,从数据处理和存储、仪器仪表、电信到数字信号处理等。FPGA 一般分为三个部分:可编程逻辑块 CLB、可编程 I/O 模块和可编程内部连线。FPGA 内部含有多个独立的 CLB,互相可以灵活连接。CLB 的功能很强,不仅能够实现逻辑函数,还可以配置成 RAM 等复杂形式。

CPLD 和 FPGA 两者各有特点。CPLD 分解组合逻辑的功能很强,一个宏单元就可以分解十几个甚至 20~30 多个组合逻辑输入。而 FPGA 的一个 LUT 只能处理 4 输入的组逻辑,因此, CPLD 适合于设计译码等复杂组合逻辑。但 FPGA 的制造工艺确定了 FPGA 芯片中包含的 LUT 和触发器的数量非常多,往往都是几千上万, CPLD 一般只能做到 512 个逻辑单元,而且如果用芯片价格除以逻辑单元数量, FPGA 的平均逻辑单元成本大大低于 CPLD。所以如果设计中使用到大量触发器,例如设计一个复杂的时序逻辑,那么使用 FPGA 就是一个很好的选择。同时 CPLD 具有上电即可工作的特性,而大部分 FPGA 需要一个加载过程,所以,如果系统要求可编程逻辑器件上电就能工作,那么就应该选择 CPLD。

最近几年,可编程逻辑器件技术取得了巨大进步。例如,仅仅数年前,最大规模的 FPGA 器件也仅仅为数万系统门,工作频率为 40 MHz,而且价格相对较贵,当时最先进的 FPGA 器件大约要 150 美元。然而,今天具有最先进特性的 FPGA 可提供百万门级的逻辑容量、工作频率为 300 MHz,成本低至不到 10 美元,并且还提供了更高水平的集成特性,如处理器和存储器。

1.3 FPGA 的特点

FPGA 采用了逻辑单元阵列(LCA, Logic Cell Array)这样一个新概念, 内部包括可配置逻辑模块(CLB, Configurable Logic Block)、输入输出模块(IOB, Input Output Block)和内部连线(Interconnect)三个部分。

CLB 是 FPGA 内的基本逻辑单元, 具有相对独立的组合逻辑阵列, 每个 CLB 都包含一个由 4 或 6 个输入、一些选型电路(多路复用器等)和触发器组成的可配置开关矩阵。开关矩阵是高度灵活的, 可以进行配置以便处理组合逻辑、移位寄存器或 RAM。每块芯片包含几十到近千个 CLB, 相互间通过互联布线进行连通。CLB 提供了逻辑性能, 灵活的互联布线在 CLB 和 I/O 之间发送信号。有几种布线方法, 从 CLB 内部互联到实现时钟与其他全局信号低歪斜布线的水平和垂直长线。除非特别规定, 对用户而言设计软件中的互联布线任务是透明的, 这样就极大地降低了设计复杂度。

FPGA 支持很多 I/O 标准, 这样就为用户的系统提供了理想的接口连接。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。当今领先的 FPGA 提供了很多 I/O 组, 实现了 I/O 支持的灵活性。

大多数 FPGA 均提供嵌入式 Block RAM 存储器, 这可以在用户的设计中实现片上存储器。

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)。Xilinx 推出的 FPGA 提供数字时钟管理和相位环路锁定。相位环路锁定能够提供精确的时钟综合, 且能够降低抖动, 并能够实现过滤功能。

FPGA 的基本特点主要有:

(1) 采用 FPGA 设计 ASIC 电路, 用户不需要投片生产, 就能得到合适的芯片。

(2) FPGA 可做其他全定制或半定制 ASIC 电路的中试样片。

(3) FPGA 内部有丰富的触发器和 I/O 引脚。

(4) FPGA 是 ASIC 电路中设计周期最短、开发费用最低、风险最小的器件之一。

(5) FPGA 采用高速 CHMOS 工艺, 功耗低, 可以与 CMOS、TTL 电平兼容。

可以说, FPGA 芯片是小批量系统提高系统集成度、可靠性的最佳选择之一。

FPGA 是由存放在片内 RAM 中的程序来设置其工作状态的, 因此, 工作时需要对片内的 RAM 进行编程。用户可以根据不同的配置模式, 采用不同的编程方式。

加电时, FPGA 芯片将 EPROM 中的数据读入片内编程 RAM 中, 配置完成后, FPGA 进入工作状态。掉电后, FPGA 恢复成白片, 内部逻辑关系消失, 因此, FPGA 能够反复使用。FPGA 的编程无需专用的 FPGA 编程器, 只需用通用的 EPROM、PROM 编程器即可。这样, 同一片 FPGA, 不同的编程数据, 可以产生不同的电路功能, 因此, FPGA 的使用非常灵活。

FPGA 有多种配置模式: 并行主模式为一片 FPGA 加一片 EPROM 的方式; 主从模式可以支持一片 PROM 编程多片 FPGA; 串行模式可以采用串行 PROM 编程 FPGA; 外设模式可以将 FPGA 作为微处理器的外设, 由微处理器对其编程。

1.4 CPLD/FPGA 的基本结构

1.4.1 CPLD 的基本结构

1) 基于乘积项(Product-Term)的 CPLD 结构

通常把基于乘积项技术、Flash 工艺的 PLD 叫做 CPLD。采用这种结构的芯片有: Altera 的 MAX7000、MAX3000 系列(EEPROM 工艺), Xilinx 的 XC9500 系列(Flash 工艺)和 Lattice, Cypress 的大部分产品(EEPROM 工艺)。

以 MAX7000 为例, CPLD 的总体结构如图 1.1 所示。

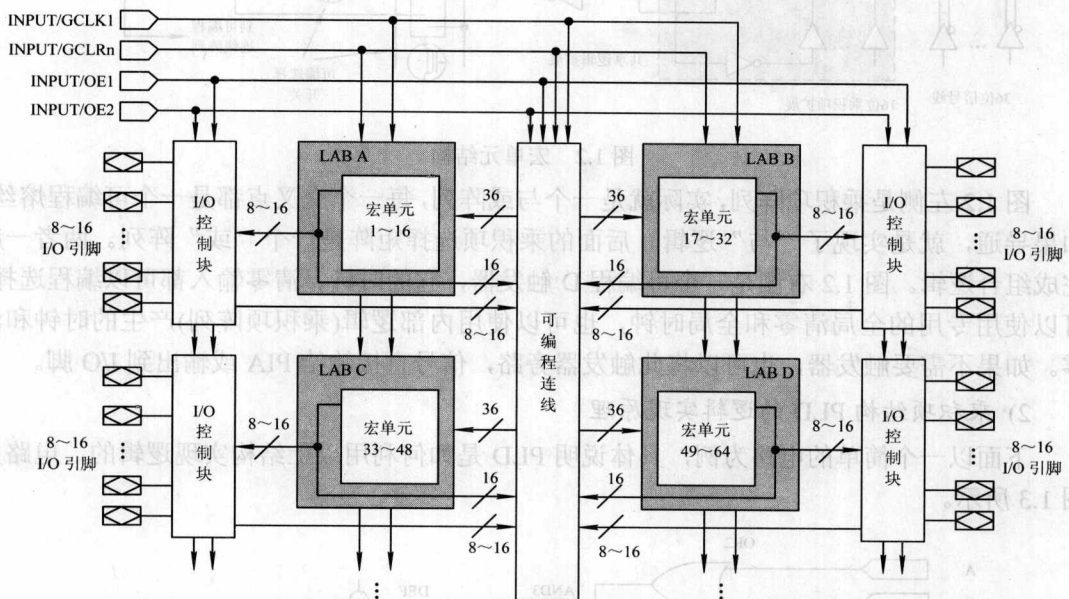


图 1.1 基于乘积项的 CPLD 内部结构

CPLD 可分为三块结构: 宏单元(Marocell)、可编程连线(PIA)和 I/O 控制块。宏单元是 PLD 的基本结构, 可以用来实现基本的逻辑功能。图 1.1 中灰色部分是多个宏单元的集合(因为宏单元较多, 没有一一画出)。可编程连线负责信号传递, 连接所有的宏单元。I/O 控制块负责输入/输出的电气特性控制, 比如可以设定集电极开路输出、摆率控制、三态输出等。图 1.1 左上方的 INPUT/GCLK1、INPUT/GCLRn、INPUT/OE1、INPUT/OE2 是全局时钟、清零和输出使能信号, 这几个信号有专用连线与 CPLD 中每个宏单元相连, 信号到每个宏单元的延时相同并且延时最短。CPLD 的逻辑块互连是集中式的, 其特点是等延时, 任意两块之间的延时是相等的, 这种结构给设计者带来很大方便。

宏单元的具体结构如图 1.2 所示。

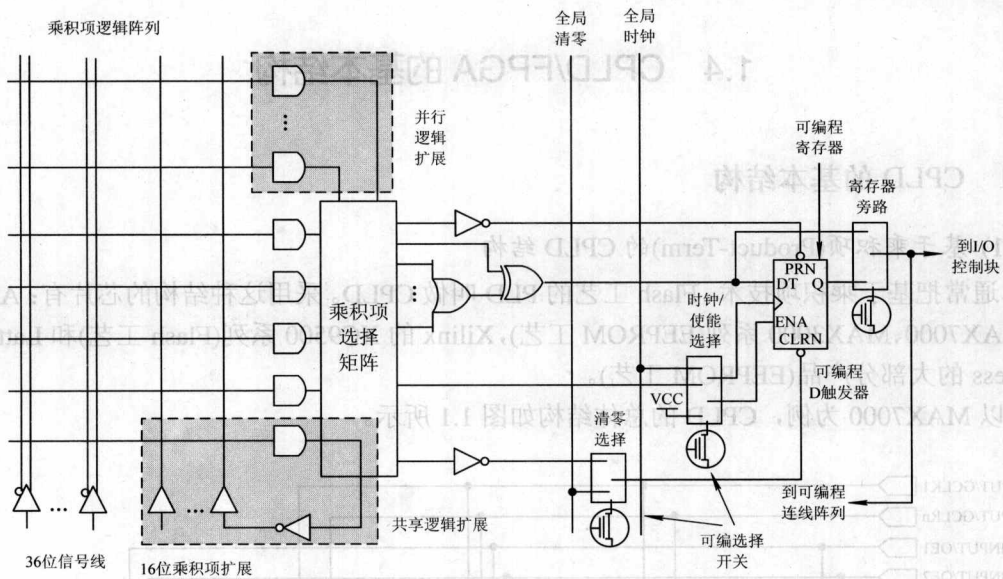


图 1.2 宏单元结构

图 1.2 左侧是乘积项阵列,实际就是一个与或阵列,每一个交叉点都是一个可编程熔丝,如果导通,就是实现了“与”逻辑。后面的乘积项选择矩阵是一个“或”阵列。两者一起完成组合逻辑。图 1.2 右侧是一个可编程 D 触发器,它的时钟、清零输入都可以编程选择,可以使用专用的全局清零和全局时钟,也可以使用内部逻辑(乘积项阵列)产生的时钟和清零。如果不需要触发器,也可以将此触发器旁路,信号直接输给 PIA 或输出到 I/O 脚。

2) 乘积项结构 PLD 的逻辑实现原理

下面以一个简单的电路为例,具体说明 PLD 是如何利用以上结构实现逻辑的,电路如图 1.3 所示。

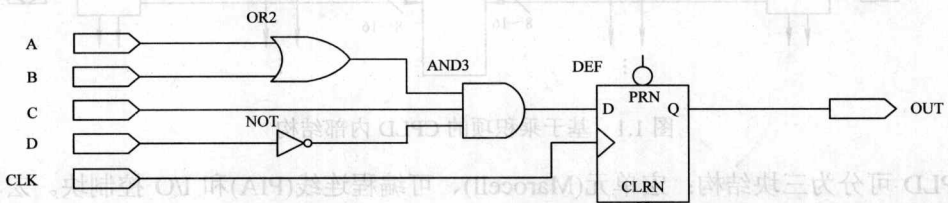


图 1.3 乘积项结构 CPLD 的逻辑实现原理

假设组合逻辑的输出(AND3 的输出)为 f , 则 $f=(A+B)*C*(!D)=A*C*!D+B*C*!D$ (!D 表示 D 的“非”)。

PLD 将以图 1.4 所示的方式来实现组合逻辑 f 。

A, B, C, D 由 PLD 芯片的管脚输入后进入可编程连线阵列(PIA),在内部会产生 A、A 反、B、B 反、C、C 反、D、D 反 8 个输出。图中每一个叉表示相连(可编程熔丝导通),所以可得到: $f=f_1+f_2=(A*C*!D)+(B*C*!D)$ 。这样组合逻辑就实现了。图 1.3 电路中 D 触发器的实现比较简单,直接利用宏单元中的可编程 D 触发器来实现。时钟信号 CLK 由 I/O

脚输入后进入芯片内部的全局时钟专用通道，直接连接到可编程触发器的时钟端。可编程触发器的输出与 I/O 脚相连，把结果输出到芯片管脚。这样 PLD 就完成了如图 1.3 所示电路的功能。(以上这些步骤都是由软件自动完成的，不需要人为干预。)

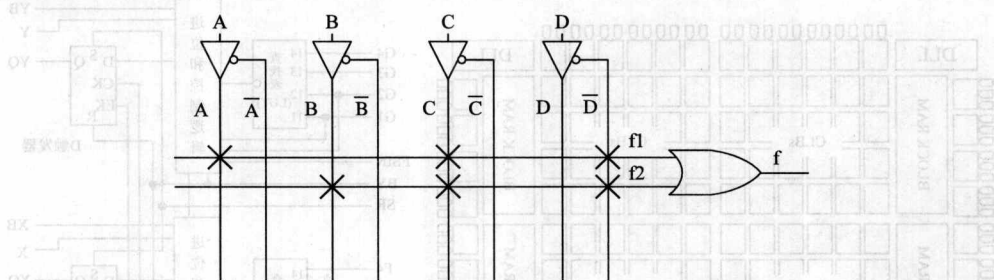


图 1.4 组合逻辑

图 1.3 的电路是一个很简单的例子，只需要一个宏单元就可以完成。但对于一个复杂的电路，一个宏单元是不能实现的，这时就需要通过并联扩展项和共享扩展项将多个宏单元相连，宏单元的输出也可以连接到可编程连线阵列，再作为另一个宏单元的输入。这样 CPLD 就可以实现更复杂的逻辑了。

这种基于乘积项的 CPLD 基本都是由 EEPROM 和 Flash 工艺制造的，一上电就可以工作，无需其他芯片配合。

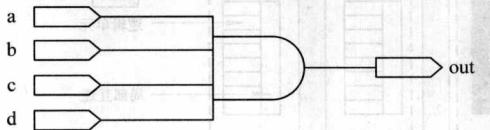
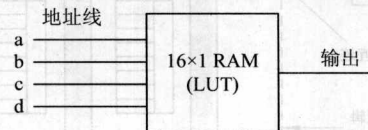
1.4.2 FPGA 的基本结构

1) 查找表(Look-Up-Table)的原理与结构

基于查找表技术、SRAM 工艺、要外挂配置 EEPROM 的 PLD 叫做 FPGA。采用这种结构的芯片有：Altera 的 ACEX 系列、APEX 系列、Xilinx 的 Spartan 系列、Virtex 系列等。

查找表(Look-Up-Table)简称为 LUT，LUT 本质上就是一个 RAM。目前 FPGA 中多使用 4 输入的 LUT，所以每一个 LUT 可以看成是一个有 4 位地址线的 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的结果，并把结果事先写入 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。

下面是一个 4 输入与门的例子：

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

2) 基于查找表(LUT)的FPGA 的结构

Xilinx Spartan- II 的内部结构如图 1.5 所示。

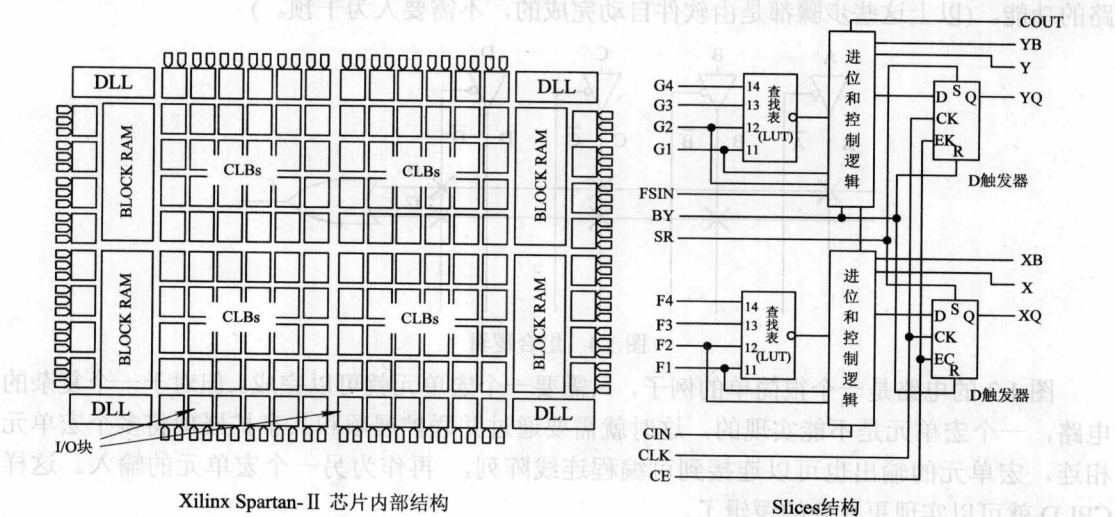
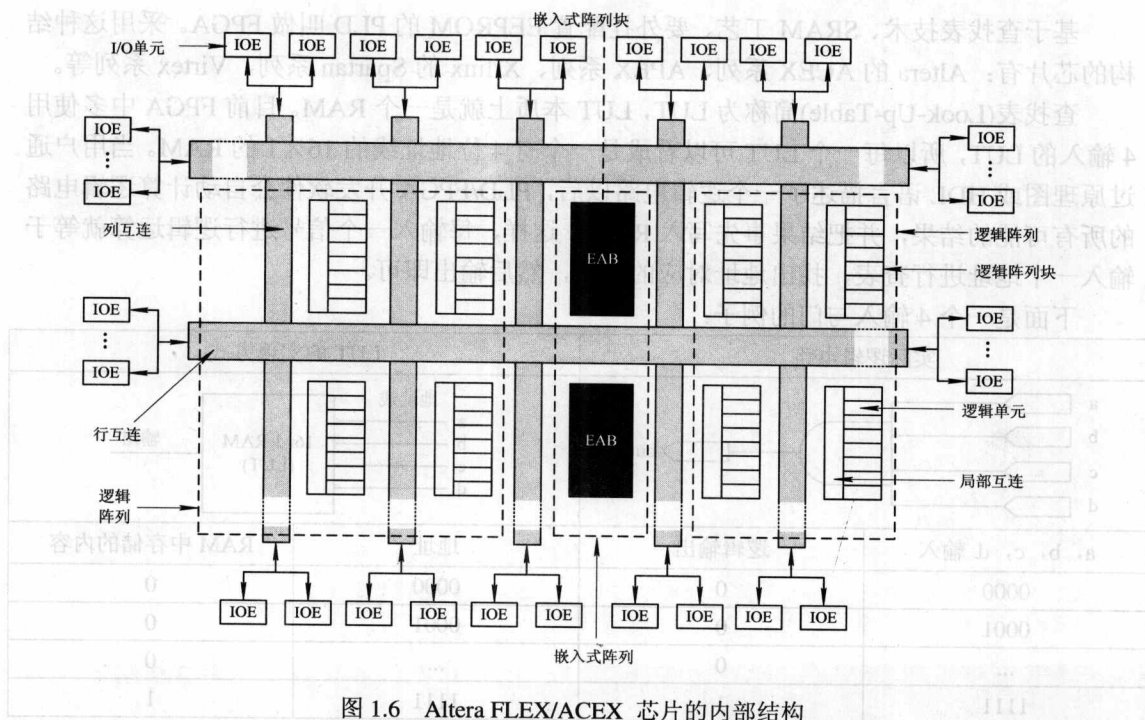


图 1.5 Spartan- II 的内部结构

Spartan- II 主要包括 CLBs、I/O 块、RAM 块和可编程连线。在 Spartan- II 中，一个 CLB 包括 2 个 Slices，每个 Slices 包括两个 LUT、两个触发器和相关逻辑。Slices 可以看成是 Spartan- II 实现逻辑的最基本结构。

Altera 的 FLEX/ACEX 等芯片的结构如图 1.6 和 1.7 所示。



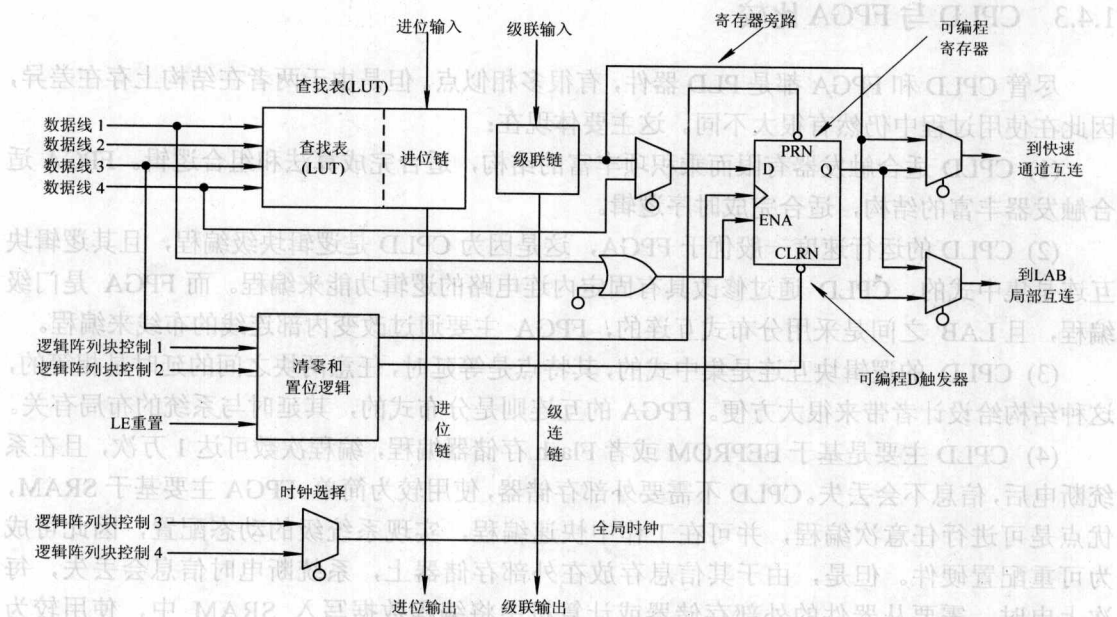


图 1.7 逻辑单元(LE)内部结构

FLEX/ACEX 的结构主要包括 LAB、I/O 块、EAB 块和内部行/列连线。在 FLEX/ACEX 中，一个 LAB 包括 8 个逻辑单元(LE)，每个 LE 包括一个 LUT、一个触发器和专用的信号通道。LE 是 FLEX/ACEX 芯片实现逻辑的最基本结构。

3) 查找表结构的 FPGA 逻辑实现原理

下面以图 1.3 所示的电路为例，介绍查找表结构的 FPGA 逻辑实现原理。

A, B, C, D 由 FPGA 芯片的管脚输入后进入可编程连线，然后作为地址线连接到 LUT，LUT 中已经事先写入了所有可能的逻辑结果，通过地址查找到相应的数据然后输出，这样组合逻辑就实现了。该电路中 D 触发器是直接利用 LUT 后面 D 触发器来实现的。时钟信号 CLK 由 I/O 脚输入后进入芯片内部的时钟专用通道，直接连接到触发器的时钟端。触发器的输出与 I/O 脚相连，把结果输出到芯片管脚。这样 PLD 就完成了如图 1.3 所示电路的功能。(以上这些步骤都是由软件自动完成的，不需要人为干预。)

图 1.3 所示的电路是一个很简单的例子，只需要一个 LUT 加上一个触发器就可以完成。对于一个 LUT 无法完成的电路，就需要通过进位逻辑将多个单元相连，这样 FPGA 就可以实现复杂的逻辑。

由于 LUT 主要适合 SRAM 工艺生产，因此目前大部分 FPGA 都是基于 SRAM 工艺的，而 SRAM 工艺的芯片在掉电后信息就会丢失，一定需要外加一片专用配置芯片，在上电的时候，由这个专用配置芯片把数据加载到 FPGA 中，然后 FPGA 就可以正常工作。由于配置时间很短，因而不会影响系统正常工作。也有少数 FPGA 采用反熔丝或 Flash 工艺，对这种 FPGA，就不需要外加专用的配置芯片。