

可编程逻辑器件实用开发技术丛书

面向 CPLD/FPGA 的 VHDL 设计

王开军 姜宇柏 等编著



机械工业出版社

目前随着对电路功能及性能要求的不断提高,传统的简单集成电路已不能满足设计者的需求,所以在功能要求不断提高的背景下,可编程器件逐渐成为广大硬件工程师所必需的设计器件。

本书通过先对 Altera 公司的 CPLD/FPGA 的介绍,使读者对可编程器件有一个深入的了解;然后通过标准 VHDL 的介绍,并且穿插具体工程的实例,使读者对 VHDL 有一个准确的掌握,不再对可编程器件感到神秘,而且能够达到轻松上手的目的。

本书内容丰富、全面系统、实用性很强,可以使读者快速全面地掌握 VHDL 设计的知识。本书既可以作为高等学校相关专业的教材或参考书,同时也可以作为广大硬件电路设计工程师必不可少的工具书或培训教材。

图书在版编目 (CIP) 数据

面向 CPLD/FPGA 的 VHDL 设计/王开军,姜宇柏等编著. —北京:机械工业出版社,2006.10

(可编程逻辑器件实用开发技术丛书)

ISBN 7-111-20109-4

I. 面... II. ①王...②姜... III. 可编程序逻辑器件—系统设计
IV. TP332.1

中国版本图书馆 CIP 数据核字 (2006) 第 124061 号

机械工业出版社 (北京市百万庄大街 22 号 邮政编码 100037)

责任编辑:张俊红 版式设计:霍永明 责任校对:唐海燕

封面设计:马精明 责任印制:杨 曦

北京机工印刷厂印刷

2007 年 1 月第 1 版·第 1 次印刷

184mm×260mm·19.25 印张·471 千字

0 001—4 000 册

定价:33.00 元

凡购本书,如有缺页、倒页、脱页,由本社发行部调换

本社购书热线电话 (010) 68326294

编辑热线电话 (010) 88379768

封面无防伪标均为盗版

丛 书 序

随着科学技术的迅猛发展,电子工业界经历了巨大的飞跃。集成电路的设计正朝着速度快、性能高、容量大、体积小和微功耗的方向发展。这种发展必将导致集成电路的设计规模日益增大,复杂程度日益增高。基于这种情况,可编程逻辑器件的出现和发展大大改变了传统的系统设计方法,这种方法使得电子系统设计变得更加简单方便、灵活快速,因此掌握可编程逻辑器件和相应的设计技术已经成为从事电子系统设计的设计师和科研人员的一项重要设计手段和技能。

可编程逻辑器件和相应的设计技术体现在三个主要方面:一是可编程逻辑器件的芯片技术;二是适用于可编程逻辑器件的硬件编程技术,即 VHDL 技术和 Verilog 技术;三是可编程逻辑器件设计的 EDA 开发工具,它主要用来进行可编程逻辑器件应用的具体实现。

可编程逻辑器件(Programmable Logic Device, PLD),它的逻辑功能是由设计人员根据系统设计的具体要求,通过相应的器件编程来实现的。另外,由于 PLD 的集成度很高,因此它可以满足大多数数字系统设计的需要。历史上,可编程逻辑器件经历了 PROM、PLA、PAL、GAL、EPLD 到 CPLD 和 FPGA 的发展过程,它在结构、制造工艺、集成度、逻辑功能、速度和功耗上都有了很大的提高和改进。其中,CPLD 和 FPGA 由于集成度非常高,因此这两种可编程逻辑器件成为目前可编程逻辑器件的主流。

可编程逻辑器件的硬件编程技术主要体现在硬件描述语言的应用中,目前广泛使用的硬件描述语言是 VHDL 和 Verilog。这两种描述语言具有强大的功能和硬件描述能力,易于共享和复用,同时还具有独立于器件和工艺的设计能力,因此它们得到了各种 EDA 工具和集成电路厂商的普遍认同和推广,目前正在全球范围内的电子系统设计领域获得广泛应用。如今,国内外一些用户在购买和使用各种 EDA 工具时,通常都把是否支持 VHDL 和 Verilog 作为 EDA 工具是否先进的标准之一。

EDA 开发工具主要包括编辑器、仿真工具、检查/分析工具和优化/综合工具等。其中,编辑器用来对设计输入进行图形或者文本等方面的编辑操作;仿真工具用来完成设计仿真操作的 EDA 开发工具,它主要包括逻辑仿真工具和时序仿真工具;检查/分析工具用来对设计的逻辑产生可能性、电路的电气特性以及时序关系等进行检查和分析;优化/综合工具用来把一种硬件描述转化为底层描述,在转化的过程中伴随着设计的某种优化。现在,高级的 EDA 开发工具都是一种集成的开发环境,即集成了上述的所有开发工具,这样就可以用一种集成开发环境来完成所有的设计工作。

可见,对于可编程逻辑器件的设计来说,上面三个主要方面是相辅相成不可分割

的, 一个高水平的可编程逻辑设计人员必须掌握这三个方面的技术, 这样才能够满足实际设计工作的需要, 从而实现成本低、设计简单和资源优化的完美设计。现在, 可编程逻辑器件获得了极其广泛的应用, 无论是电子设计工程师还是高等院校的学生, 都迫切需要系统地学习相应的技术, 因此也就需要一套理论严谨、内容新颖、实用性较强的可编程逻辑器件丛书来满足广大读者的学习需要。基于这一点, 机械工业出版社的领导和编辑组织了这套“可编程逻辑器件实用开发技术丛书”, 这套丛书重点介绍了 Xilinx 公司和 Altera 公司的 CPLD/FPGA、相应的 EDA 开发工具 ISE 和 MAX + plusII/Quartus II 以及相应的 VHDL 和 Verilog 设计技术。本套丛书具体包括以下几本:

- 《VHDL 设计实例与仿真》
- 《通信收发信机的 Verilog 实现与仿真》
- 《面向 CPLD/FPGA 的 Verilog 设计》
- 《面向 CPLD/FPGA 的 VHDL 设计》
- 《MAX + plus II 和 Quartus II 应用与开发技巧》
- 《ISE 应用与开发技巧》
- 《Xilinx 可编程逻辑器件的应用与设计》
- 《Altera 可编程逻辑器件的应用与设计》

为了保证这套丛书的高质量和实用性, 特组织了一批具有丰富可编程逻辑器件设计经验的工程师来进行相应丛书的编写。这套丛书从实际应用的角度出发, 全面系统、由浅入深地介绍了可编程逻辑器件的各个相关技术, 可以使广大读者快速高效地掌握可编程逻辑器件的知识。本套丛书读者范围十分广泛, 它既可作为高等学校计算机和电子工程专业的研究生和本科生的教材或教学参考书, 也可作为广大电子电路设计工程师、ASIC 设计人员和系统设计人员的参考书。

由于可编程逻辑器件技术发展十分迅速, 加上编写时间相对紧张, 书中难免存在不足, 恳请广大读者和专家批评指正, 联系信箱 buptzjh@163.com。

丛书编委会

前 言

目前,随着对电路功能及性能要求的不断提高,传统的简单集成电路已不能满足设计者的需求,所以在功能要求不断提高的背景下,可编程器件逐渐成为广大硬件工程师所必需的设计器件。这种发展必将导致集成电路的设计规模日益增大,复杂程度日益增高。经过 40 多年的发展,集成电路已经能够很容易地完成在每个芯片上集成数百万个以上的晶体管。如今,大规模和超大规模集成电路技术已经成为高科技研发领域的基础。

通用可编程逻辑器件,通常也称为 PLD,即 Programmable Logic Device。这里,PLD 虽然是作为一种通用集成电路来进行生产的,但是它的逻辑功能是由设计人员根据系统设计的具体要求通过相应的器件编程来实现的。另外,由于 PLD 的集成度很高,因此它可以满足大多数数字系统设计的需要。可见,设计人员通过相应的器件编程就可以把设计的系统集成在一片 PLD 上,而不再需要由厂商来设计和制造相应的专用集成电路了,这样便解决了专用集成电路的专用性和成本较高以及开发周期较长的主要矛盾。

采用传统方法设计数字系统,特别是当电路系统非常庞大时,设计者必须具备较好的设计经验,而且繁杂多样的原理图的阅读和修改也给设计者带来诸多的不便。现在,随着系统级 FPGA 以及系统芯片的出现,软硬件协调设计和系统设计变得越来越重要。传统意义上的硬件设计越来越倾向于与系统设计和软件设计相结合。

VHDL 自从出现的那一刻起,便以它卓越的优点迅速得到大多数硬件工程师们的认可,现在仍然是世界上比较流行的硬件描述语言。在中国乃至东南亚地区,使用 VHDL 的人占据着大多数。所以,对于初学者来说,从 VHDL 入手比较容易获得更多的可利用的资源,从而更容易掌握这种硬件语言。

Altera 公司作为世界上最大的可编程逻辑器件供应商之一,不断地引领着世界上可编程器件领域的发展方向。其主要产品从早期的 MAX 系列、FLEX 系列、ACEX 系列到现在流行的 MAX II 系列、Stratix 系列、Stratix II 系列、Cyclone 系列、Cyclone II 系列等都有经典之作,非常有利于我们深入了解可编程器件。而作为 Altera 公司新一代开发软件的 Quartus II,更是具备了强大的功能,使我们能够非常方便地开发可编程硬件系统。

现在市场上介绍 VHDL 的书很多,但是大部分图书或者是只单单针对硬件描述语言,或者是没有具体的完整的实例,使广大读者读完之后并不能马上上手。本书根据作者本人在这方面的经验以及与他人的探讨,总结了大量实际应用时应该注意的细节要点,非常适合刚接触可编程器件的初学者,使初学者更容易上手,并且能从开始就养成良好的编程习惯。

随着可编程器件广泛地被应用，VHDL 将成为未来硬件工程师必不可少的一项技能。相信通过本书的学习，读者不仅能够掌握 VHDL，而且能够设计出性能优良、可靠性高的 VHDL 程序，从而成为一名出色的硬件工程师。

本书将从实际应用的角度出发，全面系统地介绍了 VHDL 设计的各方面知识。全书共分为 4 个部分：先介绍可编程器件的基本理论知识，主要的内容包括可编程器件的发展历史、分类、结构以及 Altera 公司器件介绍；再重点讨论硬件描述语言的发展历史、特点、VHDL 的基本语法以及 Altera 公司 Quartus II 的介绍；然后介绍了常见的基本 VHDL 程序的设计方法和技巧，重点讨论了组合逻辑电路的设计、时序逻辑电路的设计和有限状态机的设计；最后主要讨论对工程仿真的设计，这里将介绍 ModelSim 的具体使用以及作者在工作中积累的经验。

本书内容丰富、全面系统、实用性很强，可以使读者快速、全面地掌握 VHDL 设计的知识。本书既可以作为高等学校相关专业的教材或者参考书，同时也可以作为广大硬件电路设计工程师必不可少的工具书或者培训教材。需要说明的是，为了保持芯片资料的原样，书中部分图形和文字并未按国家标准做统一修改处理，这点请读者注意。

本书由王开军和姜宇柏共同编写，书中包含着作者多年来 VHDL 设计的经验总结。另外，张钰、高延帅、魏孔明也参与了部分章节的编写，全书由王开军统稿。在编写本书的过程中，吴鹏、李晓凯、李玉红、邹德智、姜海燕、姜海亭、潘天保、张蓬、蒋建新、葛树涛、王涛、渠丰沛、渠丽娜、杜平、赵鑫、郭建、夏钦东和丁海波也作出了很大的贡献，这里对他们的辛勤劳动表示感谢。另外，作者在编写本书的过程中参考了不少专家和学者的著作、学术论文和经验总结等，在此对他们表示最诚挚的谢意！

限于作者的理论水平和实际开发经验，书中难免存在一些不足之处或错误，恳请广大读者和相关专家批评指正。

作 者

2006 年 8 月

目 录

丛书序

前言

第1章 可编程器件及 Altera 公司可

编程器件简介 1

- 1.1 可编程逻辑器件简介 1
 - 1.1.1 可编程器件的发展历史及前景 1
 - 1.1.2 可编程逻辑器件的基本结构 3
 - 1.1.3 可编程器件的分类 4
- 1.2 Altera 系列器件简介 14
 - 1.2.1 PLD 厂商简介 14
 - 1.2.2 Altera 公司的复杂可编程器件 14
 - 1.2.3 Altera 公司的现场可编程门阵列 22
- 1.3 如何根据项目选择器件 25

第2章 硬件描述语言简介 27

- 2.1 硬件描述语言的由来和发展 27
- 2.2 各种硬件描述语言的介绍及特点 28
 - 2.2.1 VHDL 28
 - 2.2.2 Verilog HDL 28
 - 2.2.3 Superlog 29
 - 2.2.4 SystemC 30
- 2.3 VHDL 的基本语法 30
 - 2.3.1 VHDL 的基本结构 31
 - 2.3.2 VHDL 的基本语句 40

第3章 Altera 公司 Quartus II 介绍 66

- 3.1 Quartus II 简介 66
- 3.2 Quartus II 安装及界面介绍 68
 - 3.2.1 Quartus II 安装 68
 - 3.2.2 Quartus II 界面简介 69
 - 3.2.3 Quartus II 常用的设置 82

第4章 组合逻辑电路设计 85

- 4.1 组合逻辑电路概述 85
 - 4.1.1 组合逻辑电路的定义 85
 - 4.1.2 组合逻辑电路的分析 85
 - 4.1.3 组合逻辑电路的设计 86
- 4.2 我在第一个项目中遇到的问题 87
- 4.3 典型的组合逻辑电路分析 90

- 4.3.1 译码器 90
- 4.3.2 加法器 92
- 4.3.3 只读存储器 95
- 4.3.4 比较器 96
- 4.3.5 多路选择器 98
- 4.3.6 三态总线 102
- 4.4 工程师们的经验 102
 - 4.4.1 组合逻辑电路的竞争冒险 103
 - 4.4.2 选择器设计和 FPGA 资源 107

第5章 时序逻辑电路的设计 108

- 5.1 时序是一切硬件工作的基础 108
 - 5.1.1 时序逻辑电路的定义 108
 - 5.1.2 时序逻辑电路的分类 108
 - 5.1.3 时序逻辑电路的分析 109
 - 5.1.4 时序逻辑电路的设计 113
- 5.2 设计中应考虑时序问题 119
 - 5.2.1 时钟信号 119
 - 5.2.2 清零信号和置位信号 122
 - 5.2.3 建立时间和保持时间 123
 - 5.2.4 触发器及其应用 124
- 5.3 典型的时序逻辑电路分析与描述 133
 - 5.3.1 分频器 133
 - 5.3.2 计数器 135
 - 5.3.3 移位寄存器 138
 - 5.3.4 存储器 141
- 5.4 怎样才能避免潜在的危险 143
 - 5.4.1 FPGA/CPLD 中的竞争冒险 143
 - 5.4.2 时序电路中的竞争冒险 144
 - 5.4.3 如何消除时序电路中的竞争冒险 145
- 5.5 工程师们的经验 145
 - 5.5.1 毛刺的产生 145
 - 5.5.2 如何消除毛刺 146
 - 5.5.3 计数器设计与 FPGA 资源 147

第6章 有限状态机 149

- 6.1 什么是状态机 149
- 6.2 有限状态机分类及 VHDL 描述 151

6.2.1 摩尔型状态机	151	第8章 电路的仿真	222
6.2.2 米勒型状态机	153	8.1 什么是电路的仿真	222
6.3 有限状态机的编码	154	8.2 ModelSim 功能介绍	222
6.3.1 状态机的编码方式	154	8.2.1 ModelSim 窗口说明	224
6.3.2 状态方程和输出方程	156	8.2.2 波形窗口调试方法	238
6.3.3 剩余状态的处理	157	8.3 怎样写 VHDL 测试基准	242
6.4 有限状态机的 VHDL 设计	159	8.3.1 测试基准常用的 VHDL 语句	244
6.4.1 有限状态机的设计流程	159	8.3.2 测试基准分析	245
6.4.2 有限状态机的复位	160	8.4 一个功能仿真实例	250
6.5 状态机与时序逻辑电路	163	8.4.1 基本仿真流程	250
6.6 典型状态机电路的 VHDL 描述	166	8.4.2 工程仿真流程	255
6.7 工程师们的经验	179	第9章 基于 FPGA/CPLD 的 VHDL	
6.7.1 状态机速度的优化	179	设计经验总结	257
6.7.2 状态机的容错性设计	180	9.1 养成良好的编程习惯	257
第7章 典型的 VHDL 设计解析	182	9.2 怎样优化你的程序	266
7.1 分频电路	182	9.2.1 如何优化 VHDL 设计	266
7.1.1 2 的幂次分频电路	182	9.2.2 如何在 VHDL 设计中提高综合效率	274
7.1.2 非 2 的幂次分频电路	187	9.3 FPGA/CPLD 的设计和优化	276
7.1.3 非整数分频电路	198	9.3.1 哪些因素影响电路结构的复杂程度	276
7.2 倍频电路	201	9.3.2 速度和面积的优化	281
7.3 多位加法器电路	205	9.4 系统级层次式设计	292
7.4 伪随机序列发生器	206	参考文献	295
7.5 并/串转换器	208		
7.6 FIFO 存储器	212		
7.7 双向数据转换器	214		
7.8 数字频率计	216		

第 1 章 可编程器件及 Altera 公司

可编程器件简介

1.1 可编程逻辑器件简介

1.1.1 可编程器件的发展历史及前景

随着科学技术的发展,集成电路的设计正朝着速度快、性能高、容量大、体积小和微功耗的方向发展,这种发展必将导致集成电路的设计规模日益增大,复杂程度日益增高。经过 40 多年的发展,集成电路已经从小规模集成 (SSI)、中规模集成 (MSI)、大规模集成 (LSI) 发展到超大规模集成 (VLSI) 和特大规模集成 (ULSI),每个芯片可以集成数百万个以上的晶体管。如今,大规模和超大规模集成电路技术已经成为高科技研发领域的基础。

如果从具体的逻辑功能特点上来进行分类,那么大规模和超大规模集成电路可以分为通用集成电路和专用集成电路两大类。例如,常见的小规模数字集成电路 74 系列、CC4000 系列、74HC 系列和常用的大容量动态存储器等都属于通用集成电路,它们的特点是逻辑功能比较简单,并且固定不变。由于这些集成电路在组成复杂的电子系统中经常用到,因此可以看出它们具有很强的通用性。

通常,采用芯片厂商提供的中、小规模集成电路可以组成任何复杂的电子系统。但为了减小系统电路的体积、重量、功耗和提高可靠性,设计人员经常会把设计的系统直接做成一片大规模或超大规模集成电路。这种为某种专用用途设计的集成电路就称作专用集成电路,通常也称为 ASIC,即 Application Specific Integrated Circuit。例如,微处理器就是一种常见的专用集成电路,它只是在某一类计算机中可以使用。可以看出,这类集成电路的通用性比较差,而且设计和制造它们的成本较高、周期较长。

因此,为了满足人们日益增长的对专用集成电路的需求,以及对时间、开发周期的要求,通用可编程器件逐渐地在电子设计领域发展起来。

通用可编程逻辑器件,通常也称为 PLD,即 Programmable Logic Device。这里,PLD 虽然是作为一种通用集成电路来进行生产的,但是它的逻辑功能是由设计人员根据系统设计的具体要求通过相应的器件编程来实现的。另外,由于 PLD 的集成度很高,因此它可以满足大多数数字系统设计的需要。可见,设计人员通过相应的器件编程就可以把设计的系统集成在一片 PLD 上,而不再需要由厂商来设计和制造相应的专用集成电路了,这样便解决了专用集成电路的专用性和成本较高以及开发周期较长的主要矛盾。

PLD 能做什么呢?可以毫不夸张地讲,PLD 能完成任何数字器件的功能,上至高性能 CPU,下至简单的 74 电路,都可以用 PLD 来实现。PLD 如同一张白纸或是一堆积木,工程师可以通过传统的原理图输入法,或是通过硬件描述语言,自由地设计一个数字系统。通过软件仿真,我们可以事先验证设计的正确性。在 PCB (印制电路板) 完成以后,还可以利

用 PLD 的在线修改能力随时修改设计，而不必改动硬件电路。使用 PLD 来开发数字电路，可以大大缩短设计时间，减少 PCB 面积，提高系统的可靠性。PLD 的这些优点使得 PLD 技术在 20 世纪 90 年代以后得到飞速的发展，同时也大大推动了 EDA（电子设计自动化）软件和硬件描述语言（HDL）的进步。

从可编程逻辑器件的发展历史上看，可编程逻辑器件经历了 PROM、PLA、PAL、GAL、EPLD 到 CPLD（复杂可编程逻辑器件）和 FPGA（现场可编程门阵列）的发展过程，它在结构、制造工艺、集成度、逻辑功能、速度和功耗上都有了很大的提高和改进。其中，CPLD 和 FPGA 由于集成度非常高，因此这两种可编程逻辑器件也常被称作高密度可编程逻辑器件。通常，可编程逻辑器件的发展历史如下所示：

1) 20 世纪 70 年代，采用熔丝编程的 PROM 和 PLA 器件可以称作是最早的可编程逻辑器件，它可以根据用户的需要写入相应的信息来完成一定的逻辑功能。但是由于熔丝烧断后不能再次接通，因此这时的器件编程是一次性的，写入后不能进行修改。

2) 20 世纪 70 年代末，MMI 公司率先推出了一种新的可编程逻辑器件 PAL，它也采用双极性型工业制作，熔丝编程方式。从逻辑特性上来看，PAL 器件不如 PLA 器件那样灵活和方便，但是它的成本较低、价格便宜、编程方便、编程器也较为便宜，另外它还具有保密位来防止非法读出，因此它在当时获得了广泛的应用。

3) 20 世纪 80 年代初，Lattice 公司推出了另外一种新型的可编程逻辑器件 GAL，它克服了 PAL 器件存在的缺点，应用起来更加灵活和方便。GAL 器件采用了电可擦除的 CMOS 工艺制作，采用电压信号可以擦除原来的信息而重新进行编程操作。另外，GAL 器件的输出端口设置了可编程的输出逻辑宏单元（Output Logic Macro Cell, OLMC），设计人员通过编程可以将 OLMC 设置成不同的工作状态。这样可以采用同一型号的 GAL 器件来实现 PAL 器件的各种电路输出工作方式，大大提高了器件的通用性。

4) 20 世纪 80 年代中期，Xilinx 公司提出了现场可编程的概念，同时生产了世界上第一片 FPGA 器件。FPGA 的编程方式与 PAL 和 GAL 器件大为不同，它不是通过专门的编程器来完成的。通常，它的设计过程是利用一套专用的设计软件来完成的，最后生成一个用来对 FPGA 器件进行编程的文件。一般来说，FPGA 中的编程信息将会存储在专用的 RAM 中，这样上电后外部 RAM 中的编程信息将会传入到器件中的静态 RAM 中，从而实现相应的功能。

同一时期，Altera 公司推出了它的新一代可编程逻辑器件 EPLD，它采用 UVEPROM 和 EEPROM 工艺制作，同时集成度要比 PAL 和 GAL 器件高得多，其产品多半属于高密度的可编程逻辑器件。

5) 20 世纪 80 年代末，Lattice 公司又提出了在系统可编程技术，同时推出了一系列具有在系统可编程能力的 CPLD。与 EPLD 相比，CPLD 规模更大，结构更为复杂，能够完成较为复杂的系统功能设计。

6) 20 世纪 90 年代后，可编程逻辑器件进入到了飞速发展的阶段。在生产工艺方面，可编程逻辑器件的线宽越来越小，集成门数越来越大，功耗越来越低；在具体的器件编程方面，EEPROM 逐渐取代了 UVEPROM 工艺，同时 FPGA 的现场可编程技术和 CPLD 的在系统可编程技术使得可编程逻辑器件在使用上更加方便；在测试技术方面，可编程逻辑器件大多数均可采用边界扫描测试技术，大大加强了器件、电路板和系统的可测试性；在逻辑功能上，可编程逻辑器件内部嵌有微处理器，大大提高了器件的逻辑功能和处理能力。

总之，可编程逻辑器件的出现大大改变了传统的电路系统设计方法，从而形成了一种新的基于芯片的设计方法。通过基于芯片的设计方法可以大大减少系统设计中的器件数量，缩小系统体积和降低功率消耗等，从而提高设计系统的性能和可靠性。

1.1.2 可编程逻辑器件的基本结构

前面已经对通用集成电路和专用集成电路的概念进行了介绍，从而引出了可编程逻辑器件的基本概念。由于在用量不大的情况下，设计和制造专用的集成电路不仅成本高，而且设计和制造的周期也很长，而可编程逻辑器件的出现则顺利地解决了这一矛盾，因此它在业界获得了极其广泛的应用。

通常，可编程逻辑器件虽然作为一种通用器件来生产，但它的逻辑功能是由用户通过对器件编程来设定的。而且有些可编程逻辑器件的集成度很高，足以满足设计一般数字系统的需要。这样就可以由设计人员自行编程把一个数字系统“集成”到一个可编程逻辑器件上，而不必去请芯片制造商设计和制造专用集成电路芯片了。

自从可编程逻辑器件出现以后，相继出现的可编程逻辑器件产品主要包括：可编程只存储器（PROM）、可编程逻辑阵列（PLA）、可编程阵列逻辑（PAL）、通用阵列逻辑（GAL）、可擦除的可编程逻辑器件（EPLD）、复杂的可编程逻辑器件（CPLD）和现场可编程门阵列器件（FPGA）。

为了更好地掌握各种可编程逻辑器件的基本原理，这里给出可编程逻辑器件的基本结构。类似于最基本的组合逻辑电路——与或式两级电路，可编程逻辑器件的基本结构也是一种与或结构，如图 1-1 所示。

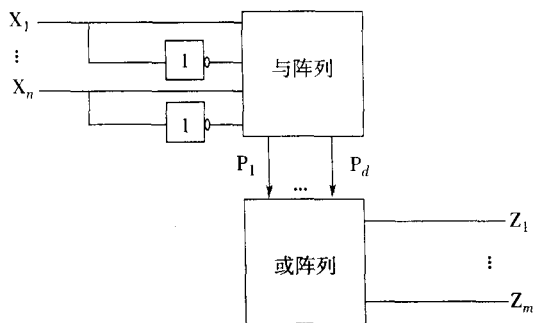


图 1-1 可编程逻辑器件的基本结构

在可编程逻辑器件的基本结构中，由于与电路、或电路的数量很多，同时它们的排列又具有一定的规律性，因此设计人员常常将其称为与阵列以及或阵列。另外，基本结构中，与阵列的输出 $P_1 \sim P_d$ 称作乘积项，它是可编程逻辑器件中非常重要的一个概念。这里，假设可编程逻辑器件的输入数为 n 、输出数为 m 、乘积项的数目为 d ，那么器件的芯片面积可以表示为如下的公式：

$$S = (2n + m)d$$

除了现场可编程门阵列器件（FPGA）外，几乎所有的 PLD 器件在其结构中都含有由与门和或门构成的与或阵列，可见与或阵列是可编程逻辑器件中的重要组成单元。由于与或阵列在硅片上非常容易实现，同时对逻辑设计也十分方便，因此它可以直接实现与或表达式形式的逻辑函数。在数字电路中，与或表达式是布尔代数的常用表达形式。根据布尔代数原理，所有的逻辑函数都可以用与或表达式来表示。因此，由这种与或阵列构成的可编程逻辑器件可以很容易地实现数字系统的设计。

对于与或阵列来说，设计人员可以通过改变与或阵列的具体连接关系来实现不同的逻辑功能。通常，只要与或阵列中的任何一个阵列的连接关系发生了变化，那么相应可编程逻辑器件的功能也会发生一定的变化。在数字电路中，根据可编程逻辑器件中与或阵列的不同组

合形式或者编程方式，可以将简单的可编程逻辑器件分为 3 种类型：

- 1) 可编程只读存储器 (PROM)，它的特点是与或阵列中与阵列固定、或阵列可编程；
- 2) 可编程逻辑阵列 (PLA)，它的特点是与或阵列中与阵列可编程、或阵列也可编程；
- 3) 可编程阵列逻辑 (PAL)，它的特点是与或阵列中与阵列可编程、或阵列固定。

随着对可编程逻辑器件的不断深入研究，其中前两种形式的与或阵列已经逐渐暴露出一定的缺点，这些缺点的具体体现是：可编程只读存储器 (PROM) 在输入数目增加时，与阵列的输出信号将会以 2 的级次递增，这样占用资源较大；可编程逻辑阵列 (PLA) 制造工艺复杂，而且器件的工作速度很慢。因此，基于这两种与或阵列结构的 PLD 器件正处于被淘汰的边缘，目前只有很少的制造商在继续生产这种 PLD 器件。相对而言，由于可编程阵列逻辑 (PAL) 中的与或阵列具有一定的技术优势，目前已经成为 PLD 发展的主流。

下面给出可编程逻辑器件的具体设计流程，目的是给设计人员一个设计参考的标准。这里将同时给出采用通用集成电路和可编程逻辑器件进行设计的具体流程，这样便于对两者进行对照比较。采用通用集成电路进行设计的流程如图 1-2 所示，而采用可编程逻辑器件进行设计的流程如图 1-3 所示。

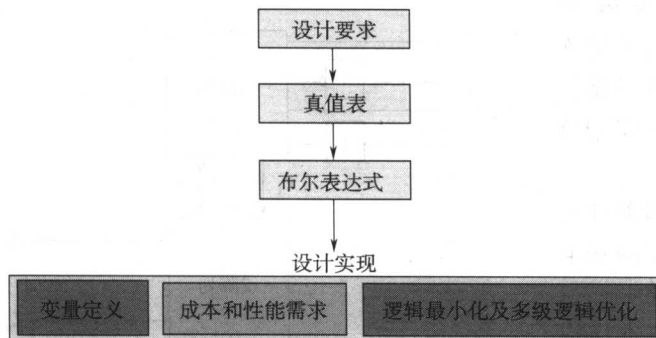


图 1-2 通用集成电路的设计流程

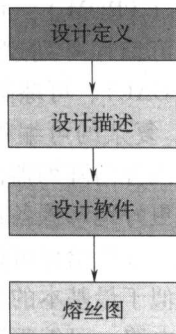


图 1-3 可编程逻辑器件的设计流程

1.1.3 可编程器件的分类

一般来说，设计人员经常按照可编程逻辑器件的集成度和结构复杂度、基本结构、逻辑单元以及编程工艺来进行分类，下面对其进行简单介绍。

1.1.3.1 按照集成度和结构复杂度进行分类

在可编程逻辑器件中，集成度和结构复杂度是一项非常重要的技术指标，因此根据集成度和结构复杂度进行分类是一种常用的分类方法。根据可编程逻辑器件的集成度和结构复杂度，它可以分为低集成度可编程逻辑器件和高集成度可编程逻辑器件。这里将会提出这样一个问题：低集成度和高集成度可编程逻辑器件的划分标准是什么？

通常，业界往往以 PALCE22V10 或者 GAL22V10 的集成度为标准来划分低集成度和高集成度可编程逻辑器件，即集成度大于 PALCE22V10 或者 GAL22V10 的可编程逻辑器件视为高集成度可编程逻辑器件，否则为低集成度可编程逻辑器件。按照这样的划分标准可以看出，

PROM、PLA、PAL 和 GAL 器件属于低集成度可编程逻辑器件，而 EPLD、CPLD 和 FPGA 则属于高集成度可编程逻辑器件，如图 1-4 所示。

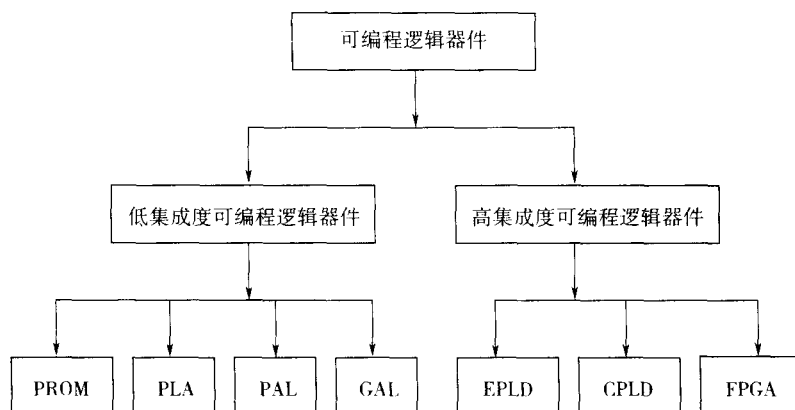


图 1-4 按照集成度和结构复杂度进行分类

按照上面的划分标准，低集成度可编程逻辑器件主要包括 PROM、PLA、PAL 和 GAL 四种可编程逻辑器件。由于低集成度可编程逻辑器件易于编程、开发成本较低、价格低廉，同时对于开发软件的要求也很低，因此在 20 世纪 80 年代得到了广泛的应用。但是随着科学技术的不断发展和不断进步，低集成度可编程逻辑器件在集成度和性能方面的局限性变得非常明显，而且它的寄存器、I/O 引脚和时钟资源的数目有限，大大限制了设计的灵活性。随着 EPLD、CPLD 和 FPGA 高集成度可编程逻辑器件的出现，它们已经渐渐退出了历史舞台。

PROM (Programmable Read Only Memory) 是一种与阵列固定、或阵列可编程的与或阵列，通常它是由地址译码器、存储矩阵和输出缓冲器三个部分组成的。PROM 采用熔丝工艺编程，因此 PROM 的编程是一次性的，写入之后不可以进行擦除或重写操作。在具体的应用中，PROM 主要用来实现组合逻辑函数，特别是可以方便地实现多输出函数，因此它的应用场合还是相对较多的。随着制造工艺的不断发展，目前常用的 PROM 器件包括两种，分别是紫外线擦除可编程只读存储器 (UltraViolet Erasable Programmable Read Only Memory, UVEPROM) 和电擦除可编程只读存储器 (Electrically Erasable Programmable Read Only Memory, EEPROM)。

PLA (Programmable Logic Array) 是一种与阵列可编程、或阵列也可进行编程的与或阵列。实际上，PLA 就是把 PROM 结构中的地址译码器改成乘积项发生器的一种可编程逻辑器件。PLA 曾经被认为是极有发展前途的可编程逻辑器件，但是由于器件的资源利用率低，现在已经基本退出了应用场合。

PAL (Programmable Array Logic) 是一种与阵列可编程、或阵列固定的与或阵列。这里，与阵列可编程是指它产生的乘积项可以根据设计的需要来进行安排；或阵列固定是指 PAL 每个输出所分配的乘积项是固定不变的，不仅每个输出所具有的乘积项数目是固定的，而且固定数目的乘积项的位置也是固定的。由于 PAL 具有多种输出结构形式，因而它的型号很多。

GAL (Generic Array Logic) 是一种在 PAL 器件的基础上发展起来的新型可编程逻辑器

件,从基本结构上看,它仍然属于与或阵列结构的可编程逻辑器件。与 PAL 器件相比,它采用了 E²CMOS 工艺,可以进行电擦除和进行重写操作;另外 GAL 采用了一个可编程的逻辑宏单元输出 (OLMC),通过对 OLMC 进行配置就可以得到多种形式的输出和反馈。由于 GAL 器件对 PAL 器件具有 100% 的兼容性,因此 GAL 器件几乎完全替代了 PAL 器件。

通常,可编程逻辑器件 PAL 和 GAL 非常适用于小规模的逻辑设计,例如选用一片或者几片简单的 PAL 或者 GAL 芯片就可以代替印制电路板中的多片中、小规模通用 IC 芯片,从而实现缩减原来的组合电路和时序电路,目的是减少印制电路板中的连线,缩小电路板的体积和重量,同时提高其可靠性。

同样按照上面的划分标准,高集成度可编程逻辑器件主要包括 EPLD、CPLD 和 FPGA 三种器件。由于这些高集成度可编程逻辑器件规模大、结构复杂,能够完成大多数比较复杂的电子系统设计,因此它们获得极其广泛的应用。目前,EPLD、CPLD 和 FPGA 不但可以解决设计系统的速度快、性能高、容量大、体积小、微功耗和高可靠性等问题,而且其开发周期短、开发软件成本低、芯片价格不断降低,因此它们获得了越来越广阔的应用市场。

20 世纪 80 年代中期,Altera 公司推出了一种新型的、电可擦除的可编程逻辑器件,它被称作 EPLD,即 Erasable Programmable Logic Device。这种新型的可编程逻辑器件采用 UVEPROM 和 EEPROM 工艺制作,同时集成度要比 PAL 和 GAL 器件高得多,其产品多半属于高密度的可编程逻辑器件。通常 EPLD 的基本逻辑单位是宏单元,宏单元是由可编程的与或阵列、可编程寄存器和可编程 I/O 三部分组成的。宏单元和整个器件的逻辑功能均由 EPROM 来定义和进行规划。

一般来说,世界著名的半导体器件公司,如 Altera、Xilinx、AMD、Lattice 和 Atmel 等公司均生产 EPLD 产品,但是结构差异很大。由于 EPLD 的内部互连能力较弱,在 20 世纪 80 年代末期受到了另外一种新兴的可编程逻辑器件 FPGA 的冲击,应用市场越来越小。直到 20 世纪 90 年代 EPLD 的改进器件 CPLD 出现以后,这种情况才有所改进。

CPLD (Complex Programmable Logic Device) 通常被称作复杂可编程逻辑器件,它是 EPLD 的改进形式,规模更大、结构更为复杂,因此它属于大规模集成电路的范围。一般情况下,CPLD 器件至少包含了三种结构:可编程逻辑宏单元、可编程 I/O 单元和可编程内部连线。此外,有些 CPLD 内部还集成了 RAM、FIFO 或双口 RAM 等存储器。其中,Lattice 公司的 ispLSI/pLSI1000 系列、AMD 公司的 MACH5 系列、Xilinx 公司的 XC9500 系列、Altera 公司的 MAX7000 系列是 CPLD 的代表性产品。

通常,CPLD 在集成度和结构上呈现出来的特点是具有更大规模的与或阵列,增加了大量的逻辑宏单元和布线资源,触发器的数目也明显增多。另外,由于 CPLD 内部采用固定长度的金属线进行各种逻辑块的互连,因此其设计的电路具有时间可预测性,从而避免了分段式互连结构时序不可完全预测的缺点。可见,常见的数字电路如高速译码器、多位计数器、寄存器和移位寄存器、时序有限状态机、网络适配器和总线控制器等较大规模的逻辑设计,均可以通过 CPLD 来进行实现。CPLD 由于具有同 FPGA 相似的集成度和易用性,而且它在速度和时序预测上还有一定的优势,因此,CPLD 能够在可编程逻辑器件技术的竞争中与 FPGA 并驾齐驱,成为两支领导可编程逻辑器件技术发展的力量之一。

近年来,各个器件厂商又推出了一些更大规模 CPLD,从而使得 CPLD 的应用更加深入和广泛。Lattice 公司的 ispLSI/pLSI3256 集成度已经达到了 14000 个等效的 PLD 门,触发

器数量也达到了 480 个。另外, ispLSI6000 系列的集成度达到了 25000 个等效的 PLD 门, 它具有 320 个宏单元。Altera 公司推出的 MAX9000 系列最高集成度可以达到 24000 个等效的 PLD 门, 逻辑宏单元数目可以达到 1024 个。因此, 具有复杂算法的数字滤波器、傅里叶变换等数字信号处理单元的逻辑设计, 都可以通过这些新型的具有更高集成度的 CPLD 来实现。

FPGA (Field Programmable Gate Array) 通常被称作现场可编程门阵列, 它最初是由 Xilinx 公司提出的。自从 20 世纪 80 年代中期 Xilinx 公司生产了世界上第一片 FPGA 器件起, 这种新型的可编程逻辑器件便以其优越的性能获得了业界设计人员的青睐, 应用迅速普及起来。与其他的可编程逻辑器件相比, FPGA 的规模可以做得很大, 一片芯片集成几十万乃至上百万的逻辑门是常有的事。另外它的逻辑功能单元绝不限于逻辑门, 而是可以具有较为复杂的逻辑功能, 从而使得整个芯片的功能大大加强。

FPGA 的功能由逻辑结构的配置数据决定, 工作时这些配置数据存放在芯片内部的 SRAM 或熔丝图上。FPGA 的编程方式与 PAL 和 GAL 器件大为不同, 它不是通过专门的编程器来完成的。通常, 它的设计过程是利用一套专用的设计软件来完成的, 最后生成一个用来对 FPGA 进行编程的文件。通常, FPGA 的配置数据可以存放在片外的 EPROM 或其他存储体上, 设计人员可以控制芯片的加载过程, 并现场修改器件的逻辑功能, 即所谓现场编程。可见, 采用 FPGA 可以十分方便地实现相应的逻辑功能设计。

前面介绍的可编程逻辑器件中, 它们都采用了与或逻辑电路或者是与或逻辑电路加上输出逻辑单元的结构形式。但是 FPGA 的电路形式与它们则是完全不同的, 它是由若干个独立的可编程逻辑模块组成的, 用户可以通过对这些模块进行编程来设计所需要的逻辑功能。通常, 由于这些可编程逻辑模块的排列形式和门阵列 (Gate Array) 中单元的排列形式十分相似, 因此这里沿用了门阵列这个名称。

一般来说, 大多数 FPGA 器件的逻辑块和布线是分开的; 而有一些 FPGA 器件的逻辑块和布线则是紧密相连的, 很难从结构上加以区分。常见的 FPGA 器件的结构主要包括三种类型, 分别是查找表结构、多路开关结构和多级与非门结构, 下面将对这 3 种结构进行一下简单介绍。

1. 查找表结构

查找表 (Look Up Table, 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。图 1-5 所示是一个 4 输入与门的例子。

查找表结构 FPGA 的可编程逻辑单元的主体是查找表, 通常它是由静态存储器 (SRAM) 来构成的, 而查找表又可以进一步构成相应的函数发生器。经过相应的计算, 可以得出 M 个输入项的逻辑函数可以由一个 2^M 位容量的 SRAM 实现。在相应的查找表结构中, 函数值存放在 SRAM 中, SRAM 的地址线起输入线的作用, 地址即输入变量值; SRAM 的输出为逻辑函数值, 由连线开关实现与其他功能快的连接。

通常, 查找表具有非常强大的函数实现功能, M 个输入的查找表可以实现任意一个 M 个输入项的组合逻辑函数, 而这样的函数可以达到 2^M 个。可见, 采用查找表来实现具体的

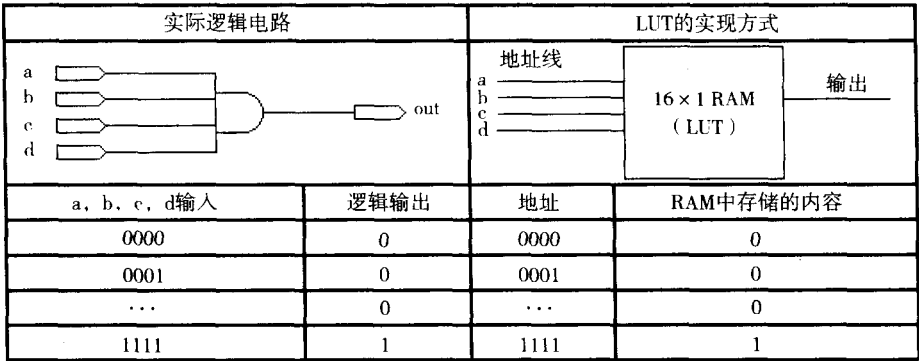


图 1-5 查找表结构

逻辑函数时，一般只需要将对应函数的真值表预先存放在 SRAM 中，即可实现相应的逻辑函数运算。因此从理论上来看，增加输入信号线和扩大存储器容量可以实现任意多的输入逻辑函数，但是这样将会带来一系列的问题，其中最主要的问题就是容量问题。

实际上对于 FPGA 的查找表结构来说，每增加一个输入项，相应查找表 SRAM 的容量就会扩大一倍，当输入项数目超过 5 个时，查找表 SRAM 容量的增加将变得十分巨大。另外，如果查找表的输入多于 5 个，那么它的存储单元数将会高于 32 个，这时它可以实现的函数数目增加得太多，而这些附加的函数在逻辑设计中有时候会用不到，同时也很难通过逻辑综合工具去开发利用。综上所述，一般查找表结构 FPGA 的查找表输入一般不多于 5 个。而对于多于 5 个输入项的逻辑函数，则是由多个查找表逻辑块组合或级联实现的。

当逻辑函数输入项少于器件查找表的输入项数目时，逻辑函数很容易实现；而当逻辑函数输入项数目大于器件查找表的输入项数目时，则需要由多个查找表组合和级联起来实现。相应地，逻辑函数也需要作些变换以适应查找表的结构要求，这一步在器件设计中称为逻辑分割。至于怎样变换逻辑函数才能用最少数目的查找表来实现逻辑函数，这是一个求最优解的问题。针对于具体的结构由相应的算法来解决这一问题，在 EDA 技术中这属于逻辑综合的范畴。在 EDA 的应用层次，这些工作则是完全由软件代替设计人员进行的。

为了加强查找表逻辑块的功能，大多数芯片制造商对基本查找表的结构进行了改进，出现了多种查找表的变形。例如，Xilinx 公司的 XC 系列 FPGA 中的查找表结构有：

- 1) XC2000 系列的逻辑块是由 4 输入/1 输出的查找表构成的，可以实现任何 4 输入变量的逻辑函数。
- 2) XC3000 系列的逻辑块是由 5 输入/1 输出的查找表构成的，而且可以配置成两个 4 输入的查找表形式。在输入项数目少于 5 个变量的情况下，可以更好地利用器件资源。
- 3) XC4000 系列的逻辑块采用两个 4 输入查找表和一个 3 输入查找表组成，由 4 输入查找表接到 3 输入查找表的多层结构来构成逻辑单元。两个查找表之间采用固定连接，这样既可以增加输入数目，又可以减少速度上的损失。
- 4) XC5200 系列由 4 个 4 输入查找表构成 4 个独立的逻辑单元，一对逻辑单元可以组合成一个 5 输入的查找表。

图 1-6 给出了 XC2000 系列中 CLB 的电路结构，CLB 中的组合逻辑是由一个 4 输入/1 输出的查找表构成的，它可以产生任何一个 4 输入变量的逻辑函数。