

21世纪高等学校计算机规划教材

21st Century University Planned Textbooks of Computer Science

EDA技术 教程

Electronic Design Automation

梁勇 王留奎 编著

- 注重可读性，深入浅出便于自学
- 注重实践性，列举典型工程实例
- 注重系统性，理论指导设计实践



高校系列



人民邮电出版社
POSTS & TELECOM PRESS

21世纪高等学校计算机规划教材

21st Century University Planned Textbooks of Computer Science

EDA技术 教程

Electronic Design Automation

梁勇 王留奎 编著



高校系列

人民邮电出版社

北京

图书在版编目(CIP)数据

EDA技术教程 / 梁勇, 王留奎编著. — 北京: 人民邮电出版社, 2010.5
21世纪高等学校计算机规划教材
ISBN 978-7-115-22339-5

I. ①E… II. ①梁… ②王… III. ①电子电路—电路设计: 计算机辅助设计—高等学校—教材 IV. ①TN702

中国版本图书馆CIP数据核字(2010)第047262号

内 容 提 要

本书从实际应用的角度出发, 全面系统地介绍了 EDA 技术和硬件描述语言 VHDL, 将 VHDL 的基础知识、编程技巧、实用方法与实际工程开发技术在 EDA 软件设计平台上很好地结合起来, 使读者能够通过本书的学习迅速了解并掌握 EDA 技术的基本理论和工程开发实用技术, 并为后续的深入学习和发展打下坚实的理论与实践基础。

本书第一部分主要介绍了可编程逻辑器件和 EDA 设计技术的基本知识, 然后讨论了可编程逻辑器件的基本原理和 Altera 公司的主流 CPLD 和 FPGA 器件。第二部分主要介绍了 EDA 设计技术中的 VHDL 设计方法, 然后重点讨论了 Altera 公司的 EDA 开发工具 Quartus II 7.2。第三部分通过大量的应用实例来讨论可编程逻辑器件和 EDA 设计技术的结合应用。本书内容丰富、技术新颖、视点独特、实用性很强, 可以使读者快速、全面地掌握可编程逻辑器件和 EDA 设计技术。书中列举的 VHDL 示例, 都经编译通过或经硬件测试。

本书主要面向高等院校本、专科 EDA 技术和 VHDL 语言基础课, 推荐作为微电子、电子工程、通信、自动化、计算机应用技术、仪器仪表等专业的授课教材或主要参考书, 同时也可以作为从事电子系统设计的设计工程师和科研人员的技术参考书。

21 世纪高等学校计算机规划教材

EDA 技术教程

-
- ◆ 编 著 梁 勇 王留奎
责任编辑 刘 博
 - ◆ 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>
三河市海波印务有限公司印刷
 - ◆ 开本: 787×1092 1/16
印张: 13
字数: 343 千字 2010 年 5 月第 1 版
印数: 1—3 000 册 2010 年 5 月河北第 1 次印刷

ISBN 978-7-115-22339-5

定价: 25.00 元

读者服务热线: (010)67170985 印装质量热线: (010)67129223
反盗版热线: (010)67171154

[illegible]

本书是根据不断发展的 EDA 技术以及编者多年的教学经验和工程实践,并在参阅同类教材和相关文献的基础上编写完成。在知识结构、基本概念、应用实例等方面的安排和取舍上,既考虑了 EDA 技术理论的系统性、完整性和简洁性,又注重了 EDA 技术教学的可操作性和实践性,尽量做到用理论指导电子设计实践,用设计实例验证理论技术,实现了理论与实践的有机结合。

本书在取材和编排上,由浅入深,循序渐进,便于读者自学;同时也注重实践性,列举了典型的工程实例。

本书可作为高等院校微电子、电子、通信等专业的教材或教学参考书，同时也可作为广大硬件电路设计工程师的工具书或培训教材。

本书由梁勇和王留奎共同编写，书中包含了作者多年的教学、实验和开发工程项目的经验总结。其中，第1章、第2章、第4章、第6章由梁勇编写，第3章、第5章由王留奎编写，全书由梁勇统稿。在本书编写的过程中，闫战强、张锦龙、李新营、赵高峰、马兴平、张大蔚等参与了全书的校对和程序调试工作，这里向他们表示由衷的感谢。此外，本书在写作过程中参考了众多国内外同类书籍，吸收了很多新的知识，在这里也一并表示感谢。

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

电子邮箱：liangyong@henu.edu.cn。

编 者

2009 年 12 月

目 录

第 1 章 概述.....1

1.1 EDA 技术的发展概况.....1

1.1.1 EDA 的概念和发展历史.....1

1.1.2 PLD 的发展概况.....3

1.2 可编程逻辑器件概述.....4

1.2.1 简单 PLD 的基本结构.....5

1.2.2 CPLD 的基本结构.....7

1.2.3 FPGA 的基本结构.....11

1.2.4 可编程逻辑器件的主要厂商.....18

1.2.5 Altera 公司可编程逻辑器件 综述.....19

1.2.6 ISP.....21

1.3 可编程逻辑器件的设计.....24

1.3.1 一般设计流程.....24

1.3.2 基本设计方法.....27

1.3.3 EDA 的软件系统.....29

1.4 EDA 的应用及发展趋势.....30

小结.....32

习题.....32

第 2 章 硬件描述语言 VHDL.....33

2.1 VHDL 简介.....33

2.1.1 VHDL 的发展及特点.....33

2.1.2 传统设计与 VHDL 设计对照.....35

2.2 VHDL 程序的基本结构.....36

2.2.1 VHDL 程序的基本单元与构成.....36

2.2.2 实体.....37

2.2.3 结构体.....39

2.2.4 库、程序包和配置.....40

2.3 VHDL 的语法要素.....43

2.3.1 VHDL 的文字规则.....43

2.3.2 VHDL 的数据对象.....45

2.3.3 VHDL 的数据类型.....47

2.3.4 运算操作符.....51

2.4 VHDL 结构体的描述方式.....53

2.4.1 顺序描述语句.....56

2.4.2 并行描述语句.....62

2.4.3 属性描述语句.....68

小结.....69

习题.....69

第 3 章 Quartus II 开发软件.....71

3.1 Quartus II 简介.....71

3.2 Quartus II 的安装.....73

3.3 Quartus II 菜单简要说明.....79

3.4 设计输入.....85

3.4.1 文本法.....86

3.4.2 图形法.....91

3.4.3 混合输入的层次化设计方法.....94

3.5 设计项目的编译.....98

3.5.1 项目(工程)的管理.....99

3.5.2 运行编译器.....100

3.5.3 观察适配结果.....102

3.5.4 功耗分析.....104

3.6 模拟仿真和时序分析.....105

3.7 引脚锁定及器件编程.....110

3.7.1 引脚锁定.....110

3.7.2 器件编程.....112

小结.....114

习题.....114

第 4 章 VHDL 设计提高.....115

4.1 VHDL 设计逻辑电路的基本思想和 方法.....115

4.1.1 逻辑函数表达式方法.....115

4.1.2 真值表方法	116	5.3.1 MegaCore 的安装	159
4.1.3 电路连接描述方法	117	5.3.2 MegaCore 的使用实例	160
4.1.4 不完整条件语句方法	118	小结	163
4.1.5 层次化设计方法	120	习题	163
4.2 常用逻辑电路的 VHDL 实现	122	第 6 章 VHDL 设计应用实例	164
4.2.1 基本组合逻辑电路设计	122	6.1 数字频率计的设计	164
4.2.2 基本时序逻辑电路设计	127	6.2 数字钟的设计	169
4.2.3 状态机的设计	134	6.3 函数发生器的设计	173
小结	143	6.4 交通灯信号控制器的设计	179
习题	143	6.5 乐曲演奏电路的设计	182
第 5 章 Quartus II 使用提高	145	6.6 VGA 显示控制器的设计	187
5.1 IP 在 Quartus II 中的体现	145	小结	194
5.2 参数化宏功能块在 Quartus II 的 例化方法及实例	149	习题	194
5.2.1 参数化宏功能模块的例化方法	149	附录 A VHDL 关键字 (保留字)	195
5.2.2 参数化宏功能模块在原理图中 的使用	154	附录 B VHDL 编程中常见的错误	198
5.2.3 参数化宏功能模块在 VHDL 中 的使用	156	附录 C 自己动手做 PLD 器件测试 电路板的一般思路	201
5.2.4 参数化宏功能模块在混合电路 设计中的使用	157	参考文献	202
5.3 MegaCore 的安装及使用实例	159		

第 1 章

概述

在计算机技术的推动下，电子技术获得了飞速的发展，电子产品几乎渗透了工业、生产、生活的各个领域。电子技术发展的根基是微电子技术的进步，它表现为大规模集成电路加工技术，即半导体工艺技术的发展上。微电子技术和现代电子设计技术相互促进相互推动又相互制约。前者代表了物理层在广度和深度上硬件电路实现的发展，后者反映了现代先进的电子理论、电子技术、仿真技术和设计工艺与新的计算机软件的融合和升华，二者的融合成就了 EDA（Electronics Design Automation）技术。

1.1 EDA 技术的发展概况

1.1.1 EDA 的概念和发展历史

EDA 技术一般有广义和狭义之分。

广义 EDA 技术指的是以计算机硬件和系统软件为基本工作平台，继承和借鉴前人在电路和系统、数据库、图形学、图论和拓扑逻辑、计算数学、优化理论等多学科的最新科技成果而研制的商品化 EDA 通用支撑软件和应用软件包，旨在帮助电子设计工程师在计算机上完成电路的功能设计、逻辑设计、性能分析、时序测试及 PCB（印刷电路板）的自动设计。与早期电子 CAD（Computer Aided Design）软件相比，EDA 软件的自动化程度更高，功能更完善，运行速度更快，而且操作界面友好，有良好的数据开放性和互换性，即不同厂商的 EDA 软件可相互兼容。因此，EDA 技术很快在世界各大公司、企业和科研单位得到了广泛应用，它已成为衡量一个国家电子技术发展水平的重要标志。

广义 EDA 技术的范畴应包括电子工程师开发产品的全过程，以及电子产品生产过程中期望由计算机提供的各种辅助功能。一方面，EDA 技术可粗略地分为系统级、电路级和物理实现级 3 个层次上的辅助设计过程，另一方面，EDA 技术应包括电子线路从低频到高频，从线性到非线性，从模拟到数字，从分立电路到集成电路的全部设计过程。

狭义 EDA 技术是以大规模可编程逻辑器件为设计载体，以硬件描述语言为系统逻辑描述的主要表达方式，以计算机、大规模可编程逻辑器件的开发软件及实验开发系统为设计工具，通过有关的开发软件，自动完成用软件的方法设计电子系统到硬件系统的逻辑编译、逻辑化简、逻辑分割、逻辑综合及优化、逻辑布局布线、逻辑仿真，直至对于特定目标芯片的适配编译、逻辑映射、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新技术。

本教材主要介绍狭义 EDA 技术的相关知识。

EDA 技术是伴随着计算机技术和集成工艺制造技术的发展而成长壮大起来的。回顾近 40 年电子设计技术的发展历程, 可将 EDA 技术分为 3 个阶段。

1. 20 世纪 70 年代的计算机辅助设计 (CAD) 阶段

这个阶段分别研制了一些相对独立的软件工具, 典型的有 PCB 制板布线设计, 以及其他用于电路仿真的工具。该阶段的主要贡献使设计者从繁琐、重复的计算和绘图中解脱出来。该阶段的产品主要有如 AutoCAD、TANGO、Protel、SPICE 等软件。20 世纪 80 年代随着集成电路规模不断发展, EDA 技术也有了较大的突破, 针对产品开发的设计、分析、生产、测试等工具包不断出现, 有力地促进了微电子技术的发展。但该时期的 EDA 软件局限性是明显的, 各个软件工具包相互独立而且是由不同公司开发的, 一般每个工具包只完成一个任务, 因此各工具包之间的衔接需要人工干预, 这就对使用者提出挑战, 不仅要电路设计的知识有全面掌握, 而且要同时熟悉多家公司互不兼容的软件, 严重影响了设计速度。同时, 该时期的 EDA 软件不能处理复杂电子系统设计中的系统级综合与仿真。

2. 20 世纪 80 年代的计算机辅助工程设计 (Computer Aided Engineering, CAE) 阶段

伴随计算机和集成电路的发展, EDA 技术进入到计算机辅助工程设计阶段。20 世纪 80 年代初推出的 EDA 工具以逻辑模拟、定时分析、故障仿真、自动布局和布线为核心, 重点解决电路设计没有完成之前的功能检测等问题。利用这些工具, 设计师能在产品制作之前预知产品的功能与性能, 能生成产品制造文件, 在设计阶段对产品性能的分析前进了一大步。

如果说 20 世纪 70 年代的自动布局布线的 CAD 工具代替了设计工作中绘图的重复劳动, 那么, 到了 20 世纪 80 年代出现的具有自动综合能力的 CAE 工具则代替了设计师的部分工作, 对保证电子系统的设计, 制造出最佳的电子产品起着关键的作用。到了 20 世纪 80 年代后期, EDA 工具已经可以进行设计描述、综合与优化和设计结果验证, CAE 阶段的 EDA 工具不仅为成功开发电子产品创造了有利条件, 而且为高级设计人员的创造性劳动提供了方便。但是, 大部分从原理图出发的 EDA 工具仍然不能适应复杂电子系统的设计要求, 而具体化的元件图形制约着优化设计。

3. 20 世纪 90 年代电子系统设计自动化 (EDA) 阶段

为了满足千差万别的系统用户提出的设计要求, 最好的办法是由用户自己设计芯片, 让他们把想设计的电路直接设计在自己的专用芯片上。

这个阶段发展起来的 EDA 工具, 目的是在设计前期将设计师从事的许多高层次设计由工具来完成, 如可以将用户要求转换为设计技术规范, 有效地处理可用的设计资源与理想的设计目标之间的矛盾, 按具体的硬件、软件和算法分解设计等。由于电子技术和 EDA 工具的发展, 设计师可以在不太长的时间内使用 EDA 工具, 通过一些简单标准化的设计过程, 利用微电子厂家提供的设计库来完成数万门 ASIC (Application Specific Integrated Circuits) 和集成系统的设计与验证。

20 世纪 90 年代, 设计师逐步从使用硬件转向设计硬件, 从单个电子产品开发转向系统级电子产品开发, 即片上系统集成 (System On a Chip, SOC)。因此, EDA 工具是以系统级设计为核心, 包括系统行为级描述与结构综合, 系统仿真与测试验证, 系统划分与指标分配, 系统决策与文件生成等一整套的电子系统设计自动化工具。这时的 EDA 工具不仅具有电子系统设计的能力, 而且能提供独立于工艺和厂家的系统级设计能力, 具有高级抽象的设计构思手段。例如, 提供方框图、状态图和流程图的编辑能力; 具有适合层次描述和混合信号描述的硬件描述语言 (如 VHDL、

AHDL 或 Verilog-HDL), 同时含有各种工艺的标准元件库。只有具备上述功能的 EDA 工具, 才可能使电子设计工程师在不熟悉各种半导体工艺的情况下, 完成电子系统的设计。

设计工具完全集成化, 可以实现以 HDL 语言为主的系统级综合与仿真, 从设计输入到版图的形成, 几乎不需要人工干预, 因此整个流程实现自动化。该阶段 EDA 的发展还促进了设计方法的转变, 由传统的自底向上的设计方法逐渐转变为自顶向下的设计方法。

未来的 EDA 技术将向广度和深度两个方向发展, EDA 将会超越电子设计的范畴进入其他领域。随着基于 EDA 的 SOC 设计技术的发展, 软硬核功能库的建立, 以及基于 HDL 自顶向下设计理念的确立, 未来的电子系统的设计与规划将不再是电子工程师们的专利。EDA 技术将是对 21 世纪电子技术产生重大影响的技术之一。

1.1.2 PLD 的发展概况

当今社会是数字化的社会, 是数字集成电路广泛应用的社会。数字集成电路本身在不断地进行更新换代。它由早期的电子管、晶体管、小中规模集成电路、发展到超大规模集成电路(VLSIC, 几万门以上) 以及许多具有特定功能的专用集成电路。但是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。系统设计师们更愿意自己设计专用集成电路芯片, 而且希望 ASIC 的设计周期尽可能短, 最好是在实验室里就能设计出合适的 ASIC 芯片, 并且立即投入实际应用之中, 因而出现了现场可编程器件, 其中应用最广泛的当属 FPGA 和 CPLD。

最早的可编程逻辑器件出现在 20 世纪 70 年代初, 主要是可编程只读存储器(PROM) 和可编程逻辑阵列(PLA)。20 世纪 70 年代末 AMD 公司推出了可编程阵列逻辑(Programmable Array Logic, PAL) 器件。20 世纪 80 年代初期, 美国 Lattice 公司推出了一种新型的 PLD 器件, 称为通用阵列逻辑(Generic Array Logic, GAL), 一般认为它是第二代 PLD(Programable Logic Device) 器件。随着技术的进步, 生产工艺的不断改进, 器件规模不断扩大, 逻辑功能不断增强, 在 EPROM 基础上出现了高密度可编程逻辑器件(称为 EPLD 或 CPLD)。现在一般把超过某一集成度的 PLD 器件都称为 CPLD。在 20 世纪 80 年代中期, 美国 Xilinx 公司首先推出了现场可编程门阵列(Field Programmable Gate Array, FPGA) 器件。FPGA 器件采用逻辑单元阵列结构和静态随机存取存储器工艺, 设计灵活, 集成度高, 可无限次反复编程, 并可现场模拟调试验证。在 20 世纪 90 年代初, 美国 Lattice 公司又推出了在系统可编程大规模集成电路(ispLSI)。

目前 PLD 的单片集成度达 1000 万系统门以上, 速度达 420MHz 以上, 线宽达 90nm, 属深亚微米技术。从 FPGA 工艺发展来看, 0.13 μm 的产品技术已臻成熟, 而 90nm 产品技术正步入应用阶段, 65nm 产品也已有产品推出。

工艺的不断进步必然导致价格的不断下跌, FPGA 产品的价格也会因技术迁移和架构采用流水线设计而下降。正是因为 PLD 低廉的价格, 使得它在消费性电子市场以及车用市场的增长率相对强劲, 尤其是消费性电子领域。

简单地讲, PLD 是这样一种 ASIC, 内部有大量的门电路, 通过用软件编程可以实现这些门电路不同的连接关系, 从而整个 PLD 就完成了不同的功能, 并且这些门电路的连接关系可以用软件来改变。

PLD 与分立元件相比, 具有速度快、容量大、功耗小和可靠性高等优点。由于集成度高, 设计方法先进、现场可编程, 可以设计各种数字电路, 因此, 在通信、网络、仪器、数据处理、汽车、存储/服务器、工业和航空/国防等众多领域内得到了广泛应用。Xilinx 也把 FPGA 从可编程逻辑领域扩展到技术领域, 如高性能 DSP、高性能嵌入式处理和高速串行连接。不久的将来, PLD

将全部取代分立数字元件。目前一些数字集成电路生产厂商已经停止了分立数字集成电路的生产, 因此应该学会 PLD 的设计技术。

目前我国常见的 PLD 生产厂家有 Xilinx、Altera、Actel、Lattice、Atemel、Microchip 和 AMD 等, 其中 Xilinx 和 Altera 为两个主要生产厂商。

1.2 可编程逻辑器件概述

常见的可编程逻辑器件有 PROM、PLA、PAL、GAL、EPLD、CPLD 和 FPGA 等。由于历史的原因, 对可编程逻辑器件的命名不很规范, 一种器件往往具备几种器件的特征, 并不能够严格地分类, 因此可编程逻辑器件有多种分类方法, 没有统一的标准。下面介绍其中几种比较常见的分类方法。

1. 按可编程逻辑器件集成度分类

集成度是可编程逻辑器件的一项很重要的指标。从集成密度上分类, 可编程逻辑器件可分为低密度可编程逻辑器件 (LDPLD) 和高密度可编程逻辑器件 (HDPLD)。PROM、PLA、PAL 和 GAL 属于低密度可编程逻辑器件, 而 EPLD、CPLD 和 FPGA 则属于高密度可编程逻辑器件, 如图 1.1 所示。

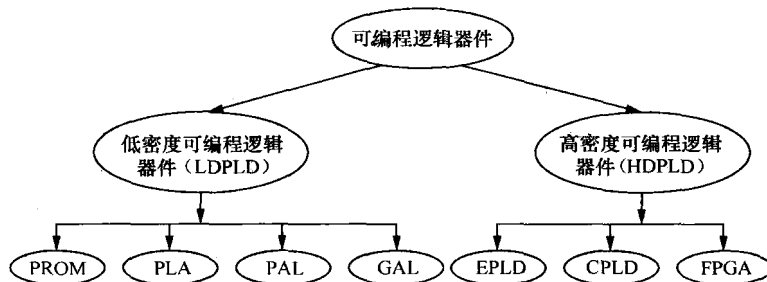


图 1.1 可编程逻辑器件分类

2. 按可编程逻辑器件结构分类

目前常用的可编程逻辑器件都是从“与-或阵列”和“门阵列”两类基本结构发展起来的, 所以又可从结构上将其分为两大类:

- PLD 器件——基本结构为与-或阵列的器件;
- FPGA 器件——基本结构为门阵列的器件。

PLD 是最早的可编程逻辑器件, 它的基本逻辑结构由与阵列和或阵列组成, 能够有效地实现“积之和”形式的布尔逻辑函数。FPGA 是最近 10 年发展起来的另一种可编程逻辑器件, 它的基本结构类似于门阵列, 能够实现一些较大规模的复杂数字系统。PLD 主要通过修改具有固定内部电路的逻辑功能来编程, FPGA 主要通过改变内部连线的布线来编程。

3. 按可编程逻辑器件编程工艺分类

所有的 CPLD 器件和 FPGA 器件均采用 CMOS 技术, 但它们在编程工艺上有很大的区别。如果按照编程工艺划分, 可编程逻辑器件又可分为 4 个种类。

① 熔丝 (Fuse) 或反熔丝 (Antifuse) 编程器件。PROM 器件、Xilinx 公司的 XC5000 系列器件和 Actel 的 FPGA 器件等采用这种编程工艺。

- ② EPROM 编程器件,即紫外线擦除可编程器件。大多数的 FPGA 和 CPLD 用这种方式编程。
- ③ EEPROM 编程器件,即电擦写可编程器件。GAL 器件、ispLSI 器件用这种方法编程。
- ④ SRAM 编程器件。Xilinx 公司的 FPGA 是这一类器件的代表。

在可编程逻辑器件的术语中,将前三类器件称为非易失性器件,它们在编程后,配置数据保持在器件上;将第4类器件称为易失性器件,每次掉电后配置数据会丢失,因而在每次上电时需要进行重新配置。由于熔丝或反熔丝器件只能写一次,所以称为一次性编程(One Time Programmable, OTP)器件,其他种类的器件均可以多次编程。

1.2.1 简单 PLD 的基本结构

PLD 器件种类较多,不同厂商生产的 PLD 器件结构差别较大,图 1.2 为 PLD 器件的基本结构框图,它由输入缓冲电路、与阵列、或阵列和输出缓冲电路 4 部分组成。其中“与阵列”和“或阵列”是 PLD 器件的主体,逻辑函数靠它们实现;输入缓冲电路主要用来对输入信号进行预处理,用户可以根据需要选择各种灵活的输出方式(组合方式、时序方式)。任何组合逻辑函数均可化为与或式,用“与门-或门”二级电路实现,而任何时序电路又都是由组合电路加上存储单元(触发器)构成的,因而 PLD 的这种结构对实现数字电路具有普遍意义。

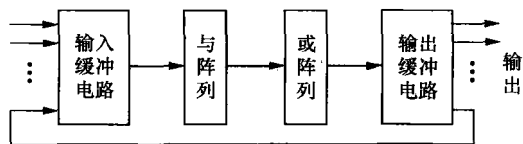


图 1.2 PLD 器件基本结构

可编程逻辑器件有一个相同的基本结构,其核心由与阵列和或阵列构成,为了能紧凑地描述 PLD 的内部电路结构,并便于识读,现广泛采用如图 1.3 所示逻辑表示方法。

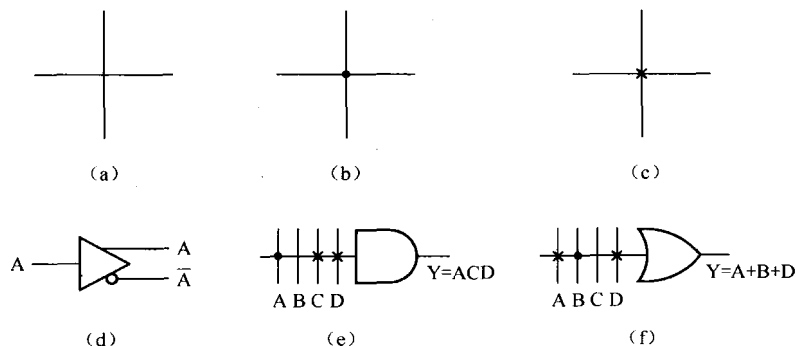


图 1.3 PLD 阵列线连接表示和逻辑图形符号

图 1.3 表示的是一些 PLD 电路的逻辑图形符号,为了使输入信号具有足够的驱动能力并产生原变量和反变量两个互补的信号,PLD 的输入缓冲器和反馈缓冲器都采用互补的输出结构,如图 1.3 (d) 所示,图 1.3 (e) 所示为一个三输入与门的 PLD 表示。图 1.3 (f) 所示为一个三输入或门的 PLD 表示。图 1.3 (a)、图 1.3 (b)、图 1.3 (c) 所示为 PLD 中阵列交叉点上 3 种连接方式的表示法,其中硬线连接是不可编程的,而接通和开断连接是靠编程实现的。在熔丝式工艺的 PLD 中(如 PAL),接通对应于熔丝未熔断,开断对应于熔丝被熔断;在 EEPROM 工艺的 PLD 中(如 GAL),接通对应于一个基本单元的导通状态,此单元被称为被编程单元;开断对应于该单元的截止状态,此单元被称为被删除单元。

早期的 PLD 主要是可编程只读存储器(Programmable Read Only Memory, PROM)。在 PROM

中,与门阵列固定,或门阵列可编程,如图 1.4 所示。与门阵列是“全译码”阵列,即输入项的每一种可能组合对应有一个乘积项。对于这种全译码阵列,若输入项数为 n ,则与门数为 2^n 个。

与门阵列可以做得很大,但阵列越大,开关延迟时间越长,速度就越慢。而且大多数逻辑函数不需要使用输入的全部可能组合,因为其中许多组合是无效的或不可能出现的,这就使得 PROM 的与阵列不能得到充分利用。PROM 除了用于随机逻辑设计,其最早的和主要的用途在存储器方面。

后来,出现了 PLA 器件,PLA 是在 PROM 结构的基础上发展而来的。在 PLA 中,与门阵列和或门阵列都是可编程的,其阵列结构如图 1.5 所示。虽然 PLA 的存储单元利用率相对较高,但是其与阵列和或阵列都可编程,造成软件算法复杂,运行速度大幅下降。由于很少有软件支持,因此 PLA 存在的时间很短。

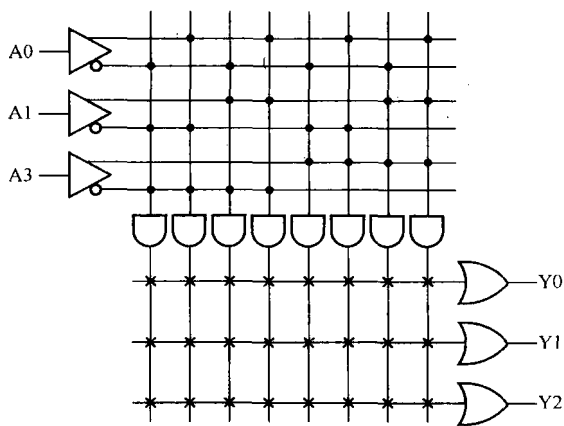


图 1.4 PROM 阵列结构

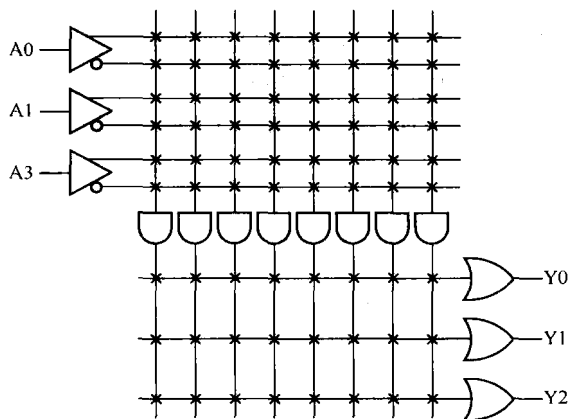


图 1.5 PLA 阵列结构

20 世纪 70 年代末期,AMD 公司率先推出可编程阵列逻辑 (Programmable Array Logic, PAL) 器件,在 PAL 中与门阵列是可编程的,而或阵列是固定的,其阵列结构如图 1.6 所示。在 PAL 产品中,最多的乘积项数可达 8 个。PAL 的这种基本门阵列结构,可以提供很高的速度。对于大多数逻辑函数,这种结构也是最有效的,因为大多数逻辑函数都可以方便地化简为若干个乘积项之和,即与—或表达式。

20 世纪 80 年代中期,Lattice 公司在 PAL 的基础上,设计出了通用逻辑阵列 (Generic Array Logic, GAL) 器件。GAL 在阵列结构上保留了 PAL 与阵列可编程、或阵列固定的结构。GAL 在输出结构上采用输出逻辑宏单元 (Output Logic Macro Cell, OLMC) 电路。输出逻辑宏单元设有多种组态,可配置成专用组合输入、专用组合输出、组合输出双向口、寄存器输出、寄存器输出双向口等,从而为逻辑设计提供了很大的灵活性。

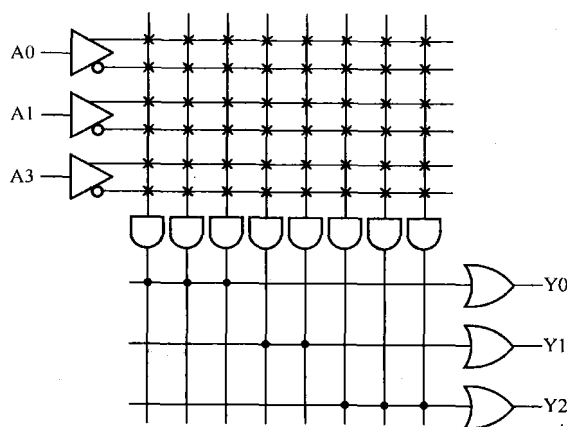


图 1.6 PAL 阵列结构

上述 PLD 器件的结构可汇总成表 1.1。

表 1.1

PLD 结构汇总表

	阵 列		输 出
	AND	OR	
PROM	固定的	可编程的	TS、OC
PLA	可编程的	可编程的	TS、OC、H、L
PAL	可编程的	固定的	TS、I/O、寄存器型
GAL	可编程的	固定的	由用户定义

注：TS——三态输出；OC——集电极开路输出；H——高电平输出；L——低电平输出。

1.2.2 CPLD 的基本结构

复杂可编程逻辑器件（Complex Programmable Logic Device, CPLD）是由 PAL 和 GAL 发展而来的，其结构与 PAL 和 GAL 器件基本相同，它通常是由可编程逻辑的功能块围绕一个位于中心的、延时固定的可编程互连矩阵构成的。典型的复杂可编程逻辑器件有 Altera 公司的 MAX 系列和 Lattice 公司的 ispLSI/PLSI 系列等。下面以 Altera 公司的 MAX7000S 系列器件为例介绍复杂可编程逻辑器件的基本结构。

MAX7000S 系列器件结构中主要包含 3 个主要部分，分别是逻辑阵列块（Logic Array Block, LAB）、可编程连线阵列（Programmable Interconnect Array, PIA）和 I/O 控制块（I/O Control Blocks, IOC）。图 1.7 表示的是 MAX7000S 系列器件的内部结构。

1. 逻辑阵列块

MAX7000S 主要是由多个相互关联的逻辑阵列块（Logic Array Block, LAB）构成的，每个 LAB 都由 16 个宏单元（Macrocells）的阵列构成。多个 LAB 是通过可编程连线阵列（PIA）连接在一起的。对于 PIA，这个全局总线包括所有的专用输入、I/O 引脚和宏单元的信号引线。LAB 都有如下的输入信号：来自通用逻辑输入的 PIA 的 36 个信号、用于寄存器辅助功能的全局控制信号、用于 I/O 引脚到寄存器的直接输入通道信号。

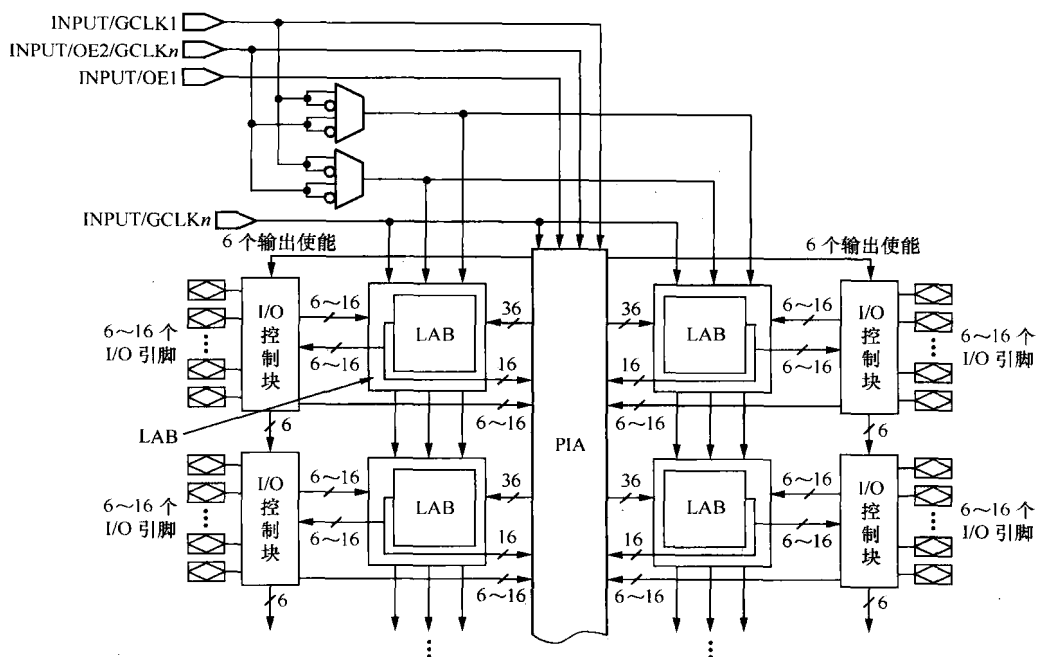


图 1.7 MAX7000S 系列器件的内部结构

(1) 宏单元

图 1.8 所示为 MAX7000S 系列器件宏单元的结构。宏单元（Macrocells）是 MAX7000S 系列器件的具体逻辑单元，由逻辑阵列、乘积项选择矩阵和可编程寄存器 3 个功能块构成。其中逻辑

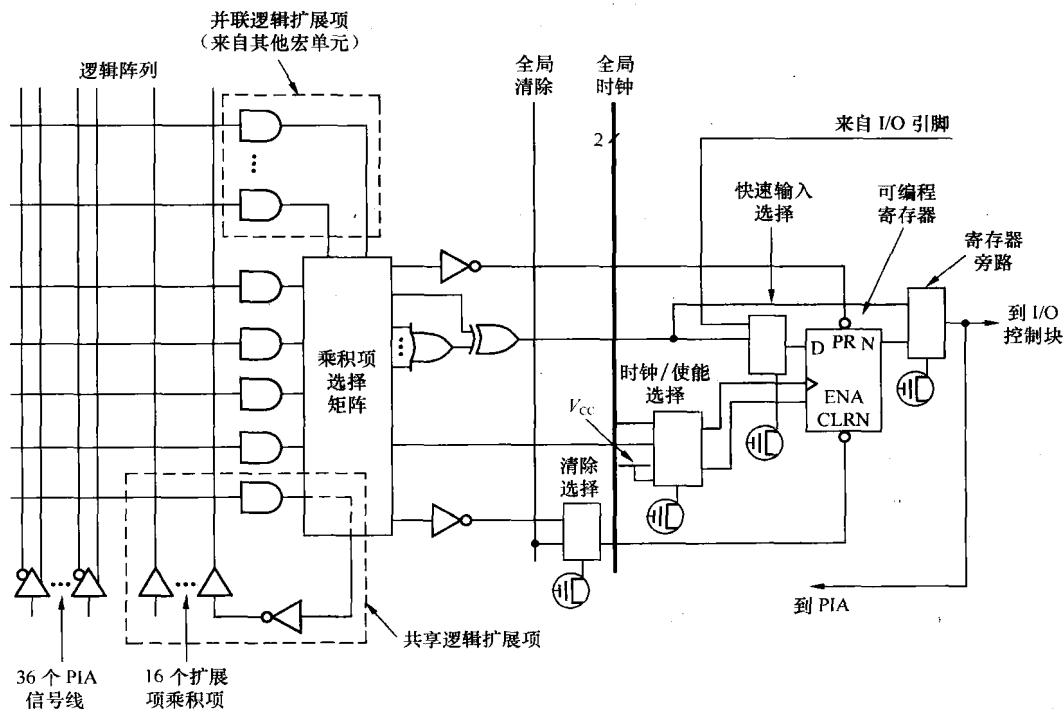


图 1.8 MAX7000S 系列器件的宏单元的结构

阵列是实现组合逻辑的，每个逻辑阵列可以给每个宏单元提供 5 个乘积项；通过乘积项选择矩阵分配这些乘积项作为主要逻辑输入（如作为或门和异或门逻辑输入）以实现组合逻辑函数功能，或者是把这些乘积项作为宏单元中寄存器的辅助输入（清零、置位、时钟和时钟的使能）。

宏单元中的可编程寄存器可以单独地配置为带有可编程时钟控制的 D、T、JK 或 SR 触发器，也可以被旁路掉，以实现组合逻辑工作方式。每个可编程寄存器都有以下 3 种不同的时钟工作模式。

- ① 全局时钟信号。该模式能提供最快的时钟控制。
- ② 带有高电平使能的全局时钟信号。该模式能够给每个触发器提供时钟使能信号。该模式能提供较快的时钟控制。
- ③ 利用乘积项实现阵列时钟。在这种模式下，触发器由来自隐含宏单元或 I/O 引脚的信号进行时钟控制。该模式具有较慢的时钟控制。

（2）扩展乘积项

在 MAX7000S 结构中有两种扩展乘积项类型，一种是共享扩展乘积项，另一种是并联扩展乘积项。MAX7000S 结构允许利用共享和并联扩展乘积项作为附加的乘积项，直接送到同一逻辑阵列块的任一宏单元中，从而利用扩展乘积项就可以实现单个宏单元不能完成的复杂函数。利用扩展乘积项可保证在实现逻辑综合时，用尽可能少的逻辑资源，得到尽可能快的工作速度。

① 共享扩展项。共享扩展项就是由每个宏单元提供一个未使用的乘积项，并将它们反向后反馈到逻辑阵列块中，每个逻辑阵列块有 16 个共享扩展项。每个共享扩展项都可以被逻辑阵列块内任何一个宏单元或全部宏单元使用和共享，以便实现复杂的逻辑函数。图 1.9 表示出共享扩展项是如何馈送到多个宏单元的。

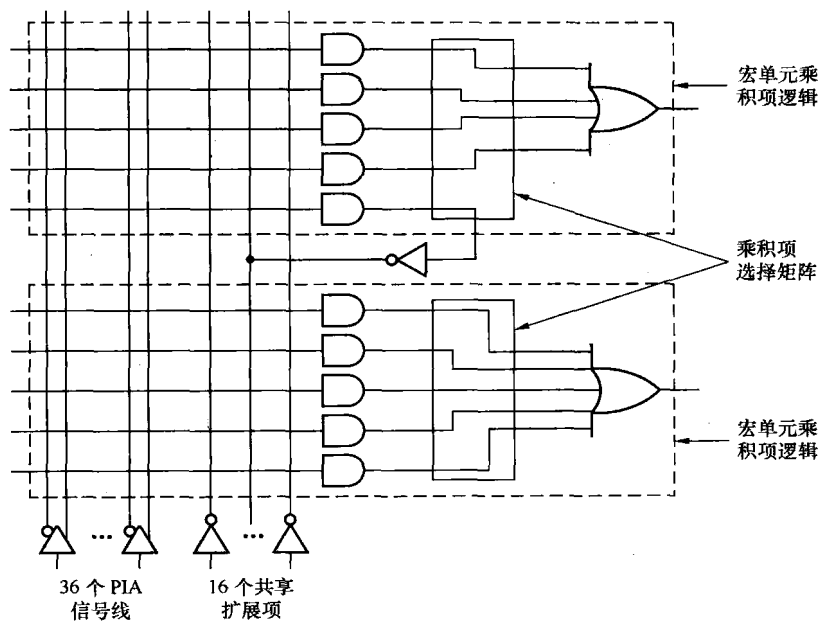


图 1.9 利用共享扩展项实现多个宏单元之间的连接

② 并联扩展项。并联扩展项是指宏单元中没有被使用的、可分配到邻近的宏单元去的乘积项。并联扩展项允许最多 20 个乘积项直接馈送到宏单元的“或”逻辑，其中 5 个乘积项是由宏单元本

身提供的,另外 15 个并联扩展项是从同一个逻辑阵列块中邻近宏单元提供的。当需要并联扩展时,宏单元“或”逻辑的输出通过选择分频器,送往下一个宏单元的并联扩展“或”逻辑输入端。图 1.10 表示出并联扩展项是如何从邻近的宏单元借用的。

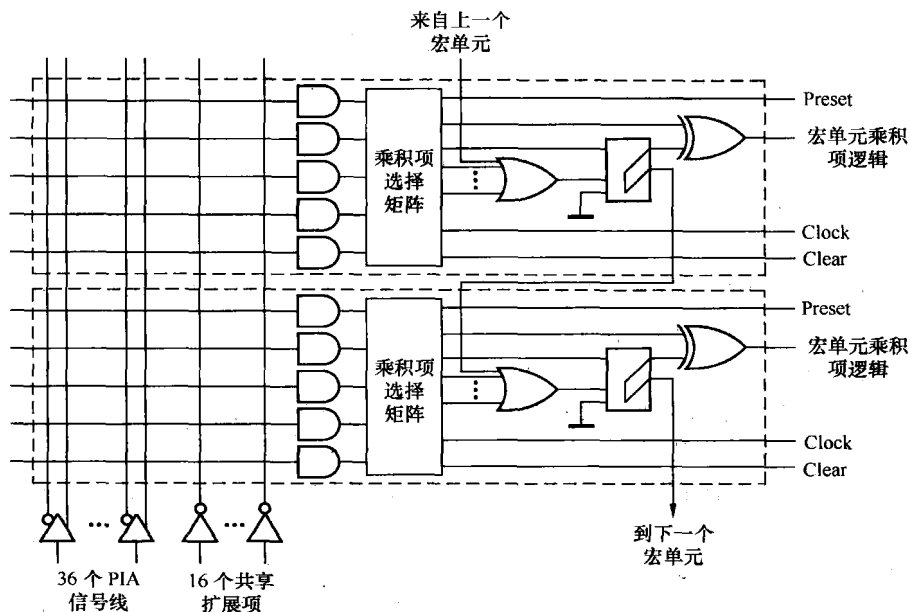


图 1.10 利用并联扩展项实现多个宏单元之间的连接

2. 可编程连线阵列

通过可编程连线阵列 (Programmable Interconnect Array, PIA), 可以把不同的逻辑阵列块相互连接, 以实现用户所需要的逻辑功能。通过对可编程连线阵列合适编程, 就可以把器件中的任何信号连接到其目的地。MAX7000S 器件所有的专用输入、I/O 引脚和宏单元输出都是连接到可编程连线阵列的, 而通过可编程连线阵列能够把这些信号送到整个器件内的任何地方。只有每个逻辑阵列块需要的信号才布置从可编程连线阵列到逻辑阵列块的连线。图 1.11 表示出可编程连线阵列连接到逻辑阵列块的方式。

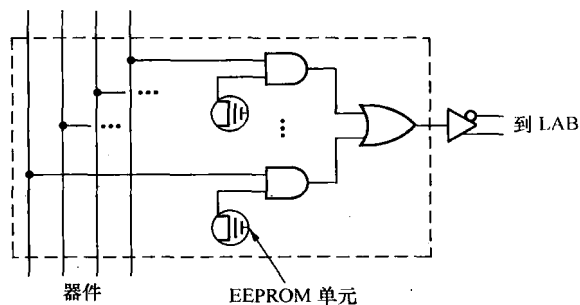


图 1.11 PIA 连接到 LAB 的方式

3. I/O 控制块

I/O 控制块 (I/O Control Blocks, IOC) 主要是由三态门和使能控制电路构成的, 在每个逻辑阵列块和 I/O 引脚之间都有一个 I/O 控制块。I/O 控制块允许每个 I/O 引脚被独立配置为输入、输出或双向工作方式。所有 I/O 引脚都有一个三态缓冲器, 它的使能端可以受到全局输出使能信号的其中一个使能信号控制, 或者是直接连到地 (GND) 或电源 V_{CC} 上。MAX7000S 系列器件的 I/O 控制块如图 1.12 所示。