在系统编程技术及其应用

（第二版）

黄正瑾　编著



东南大学出版社

在系统编程技术及其应用

（第二版）

黄正瑾　编著

东南大学出版社

内 容 提 要

本书介绍现代数字电子系统设计中使用的在系统编程技术，全书共分三篇。第一篇介绍 PLD 的工作原理、分类，着重介绍现场可编程 CPLD 的原理、特点、发展趋势，及其与现场可编程门阵列(FPGA)的比较和选用。在 ispLSI 器件的开发与编程一章中除介绍 ISP 器件的设计流程外，还对逻辑电路的语言(ABEL)描述方法、Synario 的源文件格式以及如何编写测试向量文件和调用宏器件作了介绍；在 VHDL 语言一章中介绍了 VHDL 语言的基本知识及其应用实例。第二篇为在系统编程实习，主要介绍 ISP Synario System 开发工具的使用，分别提供了组合电路和时序电路设计实例和若干设计课题。第三篇在介绍适合 PLD 的逻辑设计技巧的基础上重点介绍数字系统自顶向下的模块化设计方法，并提供了设计实例和课题。附录则对 Lattice 系统库中宏器件的名称、格式和功能作了必要的说明。

本书是从事电子产品开发和生产的工程技术人员以及其他应用电子技术的科研人员学习、掌握在系统编程技术的必备参考书。也可作为大专院校电子类专业开设有关选修课、实验课、数字系统课程设计等课程以及进行企业培训的合适教材。

图书在版编目(CIP)数据

在系统编程技术及其应用/黄正瑾编著.—2 版，—南京：
东南大学出版社，1999.5
ISBN 7-81050-485-1

Ⅰ.在… Ⅱ.黄… Ⅲ.可编程序控制器-程序设计
Ⅳ.TP332.3

中国版本图书馆 CIP 数据核字(1999)第 17543 号

东南大学出版社出版发行
(南京四牌楼 2 号 邮编 210096)
出版人：洪焕兴
江苏省新华书店经销 南京雄州彩色印刷厂印刷
开本：787mm×1092mm 1/16 印张：15 字数：365 千字
1999 年 5 月第 2 版 1999 年 5 月第 3 次印刷
印数：6001—10000 册 定价：19.00 元

前　言

在现代数字系统设计中，现场可编程器件(FPGA和CPLD)的使用越来越广。ISP(在系统编程)器件是20世纪90年代出现的最先进的可编程器件，它无须编程器，可直接对安装在用户目标板上的ISP器件编程，从而给电子产品的设计和生产带来许多革命性的变化，被誉为第四代PLD器件，代表着PLD的方向。掌握在系统编程技术是新一代电子设计工程师及从事电子技术开发和研究人员的必备技能，也是现代电子技术类专业学生学习的重要内容。编写本书的首要目的就是向读者介绍这种新技术的原理和应用，以促进我国电子技术水平的迅速提高。

现代可编程器件(FPGA和CPLD)的出现，使得电子设计工程师或科研人员有条件在实验室内快速方便地开发专用集成电路(ASIC)，这些专用集成电路往往就是一个数字系统。而在用现场可编程器件设计数字系统时，传统的试凑法常常难以胜任，更不能高效率、高质量地完成，因此，本书向读者推荐一种自顶向下的模块化设计方法，以提高设计的速度和水平。这是编写本书的第二目的。

本书部分内容参考了国外文献资料，为避免文字生涩，方便读者，特别是初学者阅读，作者根据教学和认识规律的要求，将有关文献资料综合整理，去粗取精，选择最基本的内容，充以必要的实例和作者的经验体会，系统地介绍了ISP器件的开发方法和开发软件的使用方法，同时按由浅入深，循序渐进的原则，安排了若干实习课题。愿读者通过对这些课题实际编程练习，能很快学会在系统编程技术和自顶向下的数字系统模块化设计方法。当然，这里所指只是最基本的技术和技能，更深入地掌握这项技术和技能，还得阅读更多的文献资料，并进行反复认真地训练。

由上述编写宗旨可知，本书不仅适合工程技术人员和其他科技工作者，特别是初次接触ISP器件和自顶向下的数字系统模块化设计方法者阅读，更适合于作为高等学校有关课程和企业单位培训的教材使用。除用本书为教材开设有关选修课外，《数字电路》类课程的实验，可使用本书的前二篇，为学生开设部

分组合逻辑电路和时序逻辑电路的ISP器件实验;《数字系统课程设计》课程,可用本书第三篇作为基本教材,再结合前二篇内容,以在系统编程技术为主要实现手段,培养学生的现代化工程设计能力和工程素质。

本书第一版完成于1997年初,当时CPLD(复杂的可编程逻辑器件)最大规模为2.4万门/片,曾几何时,此纪录已被多次刷新,而且利用深亚微米工艺制作的FPGA的集成规模已高达50万门/片以上。人们已不只是讨论怎样用PLD芯片去实现某些数字系统或专用CPU的主要部件,而是在研究如何将一些包括CPU在内的复杂系统作为一个软核嵌入PLD芯片中了。在这样的形势下,第一版原有的内容就有诸多不适应之处:例如书中着墨较多的pDS开发系统,因该系统设计时通常需按GLB逐个地进行,只适用较小的数字系统,现已很少使用,而书中未曾涉及的VHDL语言,由于它能适应的集成规模大,并且能适用于ASIC芯片开发,已越来越受到人们的重视。所以,第二版删去了关于pDS的内容,增补了Synario的某些操作,并由高礼忠老师增写了"VHDL语言"作为第6章。虽然新器件和新技术不断涌现,但器件的基本原理并未有重大改变,因而关于其它PLD器件的介绍基本维持原状。

本书写作过程中始终得到了上海Lattice公司的鼎力支持,第一版由章开和教授、赵源平先生审阅,第二版增写部分由陈恒先生审阅,他们对本书的写作和再版都提供了详细的极为宝贵的意见,在此谨向他们致以最诚挚的谢意。东南大学无线电工程系宋继亮、梁俊、徐莹隽、熊明珍、章小丽老师,南京铁路运输学校王瑞林老师以及黄小鹏先生、梅霆先生都对本书的写作给予热情帮助,特别是团雷明等老师研制的SE-1型数字系统实验板,为作者取得若干实践经验提供了必要条件,于此一并致谢。

由于作者水平所限,错漏之处,恳请读者指正。

作　者

1999.4于东南大学

目　　录

0 绪　论
0.1 数字系统的概念 …… (1)
0.2 现代数字系统的实现手段 …… (1)
0.3 现代数字系统的设计方法 …… (3)
0.4 现代数字系统的设计过程 …… (3)

第一篇　在系统编程技术

1 PLD 概述
1.1 PLD 的基本结构 …… (7)
1.2 PLD 的分类 …… (7)
2 PAL 和 GAL 的原理与使用
2.1 PAL 的原理与使用 …… (12)
2.2 GAL 的原理与特点 …… (20)
3 HDPLD 的原理与使用
3.1 阵列型 HDPLD …… (23)
3.2 Lattice pLSI/ispLSI 1016 简介 …… (28)
3.3 FPGA 的原理与特点 …… (39)
3.4 FPGA 与 HDPLD 的比较和选用 …… (45)
4 PLD 的开发与在系统编程技术
4.1 PLD 的开发过程 …… (48)
4.2 在系统编程技术 …… (50)
4.3 互连的在系统编程 …… (60)
4.4 在系统编程技术对数字系统的贡献 …… (63)
4.5 在系统编程的未来 …… (65)
5 ispLSI 的开发与编程
5.1 ispLSI 的开发工具 …… (67)
5.2 ISP 器件的设计流程 …… (68)
5.3 逻辑电路的语言描述 …… (70)
5.4 源文件的编写格式 …… (83)
5.5 测试向量及其序列的编排 …… (86)
5.6 宏器件及其调用 …… (94)
6 VHDL 语言
6.1 引言 …… (96)
6.2 VHDL 的基本结构 …… (97)
6.3 VHDL 的数据类型和常用语句 …… (105)

6.4 常用电路的 VHDL 描述 …… (113)
6.5 VHDL 设计举例 …… (117)

第二篇 在系统编程实习

7 实习的环境
7.1 必备环境 …… (129)
7.2 SE-1 型数字系统实验板 …… (129)
8 isp Synario System 的操作
8.1 设计开始 …… (134)
8.2 原理图输入方式 …… (137)
8.3 ABEL-HDL 语言输入方式 …… (142)
8.4 ABEL-HDL 语言与原理图混合输入方式 …… (144)
8.5 逻辑仿真(模拟) …… (146)
8.6 引脚锁定、JED 文件生成与下载编程 …… (149)
8.7 用 IDCD 软件下载编程 …… (151)
8.8 实验 …… (153)
9 组合电路与时序电路的设计实习
9.1 组合电路设计实例——GRAY/BIN 变换电路 …… (154)
9.2 组合逻辑电路设计实习课题 …… (157)
9.3 时序逻辑电路设计实例——可变模计数器 …… (159)
9.4 时序逻辑电路设计实习课题 …… (164)

第三篇 现代数字系统设计

10 适合 PLD 的逻辑设计
10.1 门、触发器级设计 …… (169)
10.2 模块级设计 …… (175)
11 数字系统设计
11.1 确定初步方案 …… (185)
11.2 确定详细方案 …… (188)
11.3 具体电路设计 …… (191)
11.4 系统设计实例 …… (196)
12 数字系统设计实习
12.1 数字系统设计实习实例 …… (201)
12.2 数字系统设计课题 …… (210)
附录 A Lattice 系统宏 …… (219)
附录 B 缩略语词汇表 …… (231)
参考文献 …… (233)

0 绪 论

0.1 数字系统的概念

现在的电子设备，单纯用模拟电路实现的已经很少见了。通常情况是，只在微弱信号放大，高速数据采集和大功率输出等局部采用模拟电路，其余部分如信号处理等均采用数字电路。也就是说，对大多数电子设备而言，其主体部分是数字系统。

系统(system)这个名词的定义是比较含混的。大到计算机系统，小到一个简单的数字密码锁控制电路，皆可称为系统。通常将门、触发器称为逻辑器件；将由逻辑器件构成，能执行某个单一功能的电路，如计数器、译码器、加法器等称为逻辑功能部件；而由若干逻辑部件组成，能实现复杂功能的电路则称为数字系统。有些规模较大的系统还可分成若干个子系统。如此说来，系统与部件之间的区别在于功能是否单一：一个存储器，尽管其规模很大，可达数兆字节，但因其功能单一，只能称之为部件；而只需几片 MSI 器件即可实现的数字锁控制电路却应称之为系统。近年来，数字系统的设计大多仿效计算机组成方式，将整个系统分为控制器和受控部分，或分成控制器、处理器和存储器三个部分。在这种结构下，系统与部件的分野就在于其中有无控制器：一个数字电路，不论其规模大小，必须具有控制器才可称之为系统。

0.2 现代数字系统的实现手段

随着集成电路技术和计算机技术的发展，数字系统的实现方法也经历了由分离元件、SSI、MSI 到 LSI、VLSI 的过程。SSI 时代直接用集成门、触发器为基本器件构成系统，MSI 时代用 MSI 器件，如计数器、译码器、数据选择器(MUX)等功能电路作为模块构成系统。有的文献把这些 SSI、MSI 逻辑器件统称为标准产品。它们都属于通用电路，任何数字系统皆可使用，其产量高、价格低，然而一个复杂系统往往要许多片 SSI、MSI 器件才可能实现，因而制成的设备体积大、重量大、功耗大、生产周期长、成本高，更重要的是用 SSI、MSI 器件做成的设备可靠性差。可靠性差的原因有两个：一是由于所用器件多，需连接的环节多，因而出现故障的机会多；二是每个 IC 器件有一定的失效时间(这是与生产工艺有关的)，一个系统所用器件数量多，其平均失效时间将下降。因而用标准产品实现数字系统的方法现在已很少使用。

在集成技术高度发展的今天，数字系统理应用 LSI、VLSI 实现，然而数字系统功能千变

万化，极其复杂，要制作一个各个系统皆能接受的通用全硬件 LSI 电路是不可能的，解决方法只有两个，一是使用微处理器，二是制作专用集成电路。

微处理器是制作在一片集成片上的 CPU，它也是一种通用型器件，只要配以一定的程序（通常固化于 EPROM 中）和一些附属电路，原则上可以实现任意复杂的逻辑功能。用微处理器做成的系统所用器件少，避免了用标准产品组成系统的缺点。最近问世的一些微处理器，利用 E^2ROM 技术，将 ROM 也制作在芯片中，使得系统的器件数更少，使用更加灵活。由于不同系统的差别在于程序不一样，硬件结构变化不大，而程序的修改方便，试制时承担的风险小，因而这种方法得到广泛的使用。但是，微处理器方法也有其缺点，其一是速度低，试想系统的一个操作，用硬件实现时，通常只需若干级器件延迟时间即可完成，而用微处理器实现时，却需一段程序，即耗费若干时钟周期才行，可见微处理器在要求处理速度很高的情况下是不适用的。其二是微处理器系统除了微处理器、ROM 外，通常还需一些辅助电路，尚不能使所用器件最少。其三，采用微处理器系统，设计者独特的构思，高超的技巧都体现在程序中，而程序又是固化在 ROM 中，就目前水平，尚无法对程序可靠地加密，因而很容易为他人非法复制，剽窃成果。

专用集成电路（ASIC）是专为某一数字系统设计、生产的集成电路。一个复杂的数字系统只要一片或数片 ASIC 即可实现，因而制成的设备体积小，重量轻，功耗低，速度高，生产周期（指整机）短，成本低，且大多数 ASIC 皆有保密性，可保护设计者的成果不被盗用。对于生产批量极大的成熟的产品可以采用全定制方法将成本降至很低很低，而对于生产批量少或处于试制阶段的产品，可利用可编程逻辑器件（PLD）或其它半定制方法，其研制时间短，设计费用低，无风险，因而 ASIC 成为现代数字系统设计的主要手段。有人认为自 1985 年起，集成电路的生产进入了 ASIC 时代，这个说法不是没有根据的。

制作 ASIC 的方法可粗略地划分为两类，一类是掩膜方法，即由半导体生产厂家制造，另一类则是现场可编程方法，也就是由数字系统的设计者通过某种方式利用半导体厂生产的 PLD 半成品芯片制作 ASIC。

集成电路生产工艺流程中需经数次光刻或蚀刻等处理，每次处理都需制作一块掩膜，设计制作掩膜的成本是相当高的，动辄数万元。当然这笔成本（通常称为先期成本）应由同一批生产出的所有集成片分担，这样每片集成电路的成本应按下式计算：

$$C = \frac{A}{N} + B$$

式中 B 是生产每片集成片所花的实际成本，A 是先期成本，N 是产品的批量。可见，当生产批量 N 较小的情况下，上式前面一项 $\frac{A}{N}$ 的值将大得难以被使用者接受。因此，只有生产批量达数万片以上，掩膜方法才有价值。何况此种方法还需承担很大风险，因为一旦设计中有错误或设计不完善，则全套掩膜便不能再用，巨额设计费用将付之东流，所以设计不到非常成熟之时，使用掩膜方法也是不合适的。再则，设计掩膜的周期也很长，例如设计一片含有万只以上晶体管的集成电路，即使使用人机交互式计算机图形辅助技术，也需耗费若干人年，想要在极短时间内推出新产品以占领市场是难以奏效的。

现场可编程逻辑器件（FPLD）是工厂专门生产的一种半成品芯片，在这些芯片中，门、触发器或其它各种电路单元被排列成阵列形式。系统的设计者可将所设计的电路通过计算机和开发工具生成关于阵列连接的信息（文件），并将这些信息通过烧录器“编程”到芯片上去。

这样，一个不懂半导体生产工艺的电气工程师也可在实验室中自行制作 ASIC 了。显然，这种方法设计时间短，费用低，一个不太大的系统可能只需数天即可完成。更为可贵的是，现在的现场可编程器件都有可改写功能，倘若设计有错，可以擦去重写，这就彻底免除了设计者承担的风险，达到 100%的可编程，这对于正在试制阶段的工程师而言，无疑是十分诱人的。这里尤要向读者推荐一种最新的在系统编程(ISP)器件，它不需烧录器，直接将 PLD 器件装在自己设计的目标系统或线路板上就可编程和修改，这在 PLD 范畴中，不啻是一场革命，因而本书主要介绍这种 ISP 编程技术。

0.3 现代数字系统的设计方法

早期的数字系统多采用试凑法设计，此法无固定的套路可循，主要凭借设计者的经验。所设计出的电路，虽然不乏构思巧妙者，但交流和修改不方便，设计所花费时间也较多。

本书介绍的是一种自上而下的模块化设计方法。自上而下也称自顶向下(top-down)，顶就是系统的功能，向下指的是将系统分割成若干功能模块。完整的意思是：从整个系统功能出发，按一定原则将系统分为若干子系统，再将每个子系统分为若干功能模块，再将每个模块分成若干较小的模块……直至分成许多基本模块实现。在自上而下划分过程中，最重要的是将系统或子系统按计算机组成思想那样划分成控制器和若干个受控制的功能模块(即受控部分)。受控部分通常为设计者们所熟悉的各种功能电路，无论是取用现成模块还是自行设计都有一些固定方法可依，无须花费更多的精力。主要的任务是设计控制器，而控制器通常相当于一个规模不大的时序机，且控制器在系统或子系统中又只有一个，设计工作量不大。这样就把一个复杂的系统设计任务化为一个较小规模的时序机设计问题，从而大大简化了设计的难度，缩短了设计时间，而且修改设计也很方便，通常只需对控制器作适当调整，无牵一发动全身之虞。所以，本书主要介绍这种自上而下的模块化设计方法。

本书介绍的数字系统最后要用现场可编程器件实现，对于 ASIC 制作过程而言，上述设计只是一种构思设计，最后还要落实到具体芯片的设计上。通常 PLD 是以门和触发器为基本单元的，也就是说，各种模块还要“下”到门和触发器级，因而需要用到各种数字电子技术书上介绍的 SSI 设计方法，但结合 PLD 的特点，有其独特之处。本书将介绍一些 PLD 中适用的 SSI 设计方法。另一方面，在现代 PLD 开发软件的数据库中都有许多功能模块(相应于各种 MSI 器件)，设计者可以随便调用，因而各种教科书所介绍的 MSI 设计方法也有其用武之地。只是一般说来，用 MSI 模块方法设计方便、快捷，但芯片利用率不如 SSI 方法高。本书对 MSI 设计方法也将专门给予介绍。

0.4 现代数字系统的设计过程

在 FPLD 大量问世以前，电子工程师们设计数字系统的过程是：书面设计——硬件搭试——制作样机。众所周知，硬件搭试是很费时间的，往往因接线紊乱和接触不良带来各式各样的麻烦，所用器件越多，搭试难度越大，当系统规模大到一定程度，系统复杂到一定程度，这种搭试实际是不可行的。

由于计算机技术的发展和 FPLD 的使用，改变了数字系统设计的程式，硬件搭试被仿真

所取代。所谓仿真，就是在计算机上建立起系统的模型，然后加进合适的测试码（对组合电路）或测试序列（对时序电路），对此模型进行测试，以验证系统是否符合预期的设计，如不符合再行修改，直至满足设计要求，然后便可制作样机。换言之，在最后制作样机（印制板）以前，都处于“纸上谈兵”状态，而此纸上所谈之兵应保证准确无误，方可节省时间和避免不必要的损失。

对于 ASIC 设计，电路的设计与连接全反映在芯片之中（印制板却是很简单的）。使用掩膜方法时，由于设计费用高，投资风险大，一定要保证其万无一失，即使使用 FPLD 方法，也要力争在烧录芯片之前使系统设计结果通过仿真。所以，现代数字系统设计过程是书面设计——仿真——（烧录器件）制作样机，这是一种快捷而且节约的方法。现在流行的仿真软件很多，设计时可以选用。本书第一篇第 5 章介绍的 ISP Synario System 设计软件中就包含有仿真程序，使用时送入合适的测试向量序列，即可通过与预期结果相比对或输出波形显示两种形式，对所设计电路仿真。本书将从应用角度，对如何编排测试向量序列作适当的讨论。

第一篇

在系统编程技术

1 PLD 概述

1.1 PLD 的基本结构

PLD 的基本结构如图 1.1 所示。电路的主体是由门构成的“与阵列”和“或阵列”，逻辑函数靠它们实现。为了适应各种输入情况，“与阵列”的每个输入端(包括内部反馈信号输入端)都有输入缓冲电路，如图 1.2 所示，从而降低对输入信号的要求，使之具有足够的驱动能力，并产生原变量(A)和反变量($\overline{A}$)两个互补的信号。有些 PLD 的输入电路还包含锁存器(latch)，甚至是一些可以组态的输入宏单元，可对输入信号进行预处理。PLD 的输出方式有多种，可以由或阵列直接输出(组合方式)，也可以通过寄存器输出(时序方式)；输出可以是低电平有效，也可以是高电平有效；不管采用什么方式，在输出端口上往往做有三态电路，且有内部通路可以将输出信号反馈到与阵列输入端。新型的 PLD 器件则将输出电路做成宏单元(macro cell)，使用者可以根据需要对其输出方式组态，从而使 PLD 的功能更灵活，更完善。

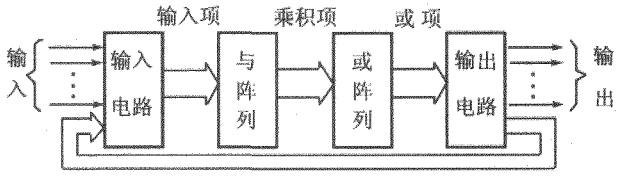


图 1.1 PLD 的基本结构框图

众所周知，任何组合逻辑函数均可化为与或式，从而用“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元件(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的意义。

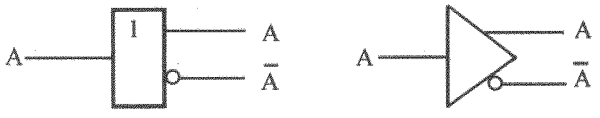


图 1.2 PLD 输入缓冲电路

1.2 PLD 的分类

1) 按可编程的部位分类

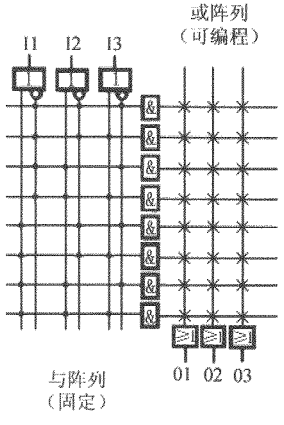
图 1.1 示出的 PLD 各个方框中，通常只有部分可以编程或组态。根据它们的可编程情况，可将 PLD 分为 PROM(可编程只读存储器)、PLA(可编程逻辑阵列)、PAL(可编程阵列逻

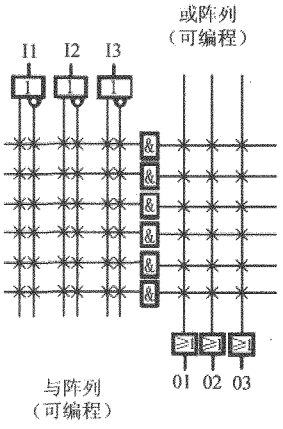
辑)和 GAL(通用阵列逻辑)等 4 类,如表 1.1 所示。

表 1.1 PLD 的分类

分类	与阵列	或阵列	输出电路
PROM	固 定	可编程	固 定
PLA	可编程	可编程	固 定
PAL	可编程	固 定	固 定
GAL	可编程	固 定	可组态

图 1.3、图 1.4 和图 1.5 分别画出了 PROM、PLA 和 PAL(GAL)的阵列结构。这里采用了一种省略画法,图 1.6 是这种画法的示意图,其中图(b)的每个与门只画一条输入线,凡是通过"·"或"×"连到此线上的输入信号都是该与门的一个输入端。该门最多可以有 8 个输入,图 1.6 中只有 3 个输入信号加在与门输入端,因而与门输出 $Z=\bar{I}_2\cdot I_1\cdot I_0$。此省略规则对于或门同样适用。至于图中连接符号·和×的含义如图 1.7 所示:·表示固定连接,就是厂家在生产芯片时连好的,不可编程;×则是可编程联接,PLD 芯片出厂时,每一点都被连上,使用者对其编程时将不需连接的×去掉。





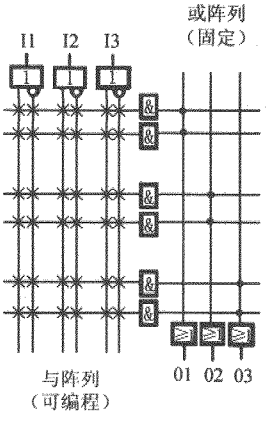
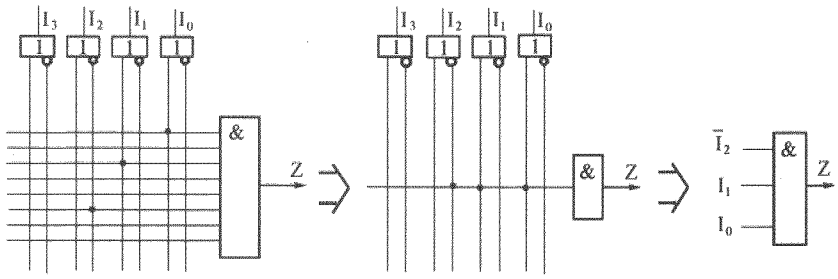


图 1.3 PROM 的阵列结构 **图 1.4 PLA 的阵列结构** **图 1.5 PAL(GAL)的阵列结构**

上述各种 PLD 中最先问世的是 PROM,它诞生于 70 年代初期,原是用于存放数据、微程序的存储器。从存储器的角度看,其与阵列实际是一个向或阵列(存储矩阵)提供地址的全译码电路,因而如与阵列有 n 个输入端,就会有 2^n 个输出,即有 2^n 个 n 输入的与门存在。如用 PROM 来实现逻辑函数,则与阵列中每个与门的输出分别代表函数的一个最小项。显然用 PROM 实现逻辑函数是不经济的,例如一个 10 变量的逻辑函数,经过逻辑简化处理,其积项通常不会超过 40 个,而 10 输入 PROM 的与阵列却有 1024 个 20 输入与门。这使得芯片面积很大,利用率很低,而且阵列面积越大,信号的开关延迟时间就越大,工作速度就越低,所以 PROM 除了制作函数表、译码显示电路外,很少用作 ASIC,通常只作为存储器出现。

70 年代中期出现的 PLA 就是为解决上述问题而设计的。因为它的与、或阵列皆可编程,可以利用逻辑化简技术,因而阵列规模可以比输入数相同的 PROM 小得多,且使用灵活。



(a)传统画法　　(b)PLD 省略画法　　(c)等效电路

图 1.6　与阵列的省略画法

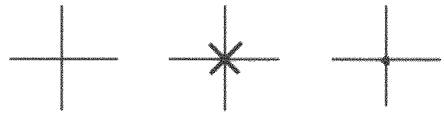


图 1.7　PLD 连接画法

但迄今为止,PLA 编程缺少高质量的支撑软件和编程工具,器件价格较贵,门的利用率也不够高,因而使用不广泛。

PAL 出现于 70 年代末、80 年代初,其特点是或阵列固定、与阵列可编程,因而又称为固定或阵列。因为这种结构速度高,价格低,其输出有极性转换和 I/O 方式,也可用触发器输出并对与阵列有反馈,通常都有触发器上电清零和加密功能,使用者可借助于编程器自行编程(称为现场可编程),因而很受用户欢迎。随后出现的通用阵列逻辑和其它高密度可编程逻辑器件(HDPLD)都是采用这种固定或阵列结构,只是用输出逻辑宏单元代替了固定的输出电路,使得输出方式也可以由用户自行组态,因而功能更强,使用更灵活,应用更广泛。可以这样认为,真正有效地使用 PLD 制作 ASIC 是从 PAL 开始的。

除了以上几种 PLD 外,还有可编程二极管阵列和可编程 MUX 阵列等特殊的 PLD。

另一种广泛用作 ASIC 的器件——现场可编程门阵列(FPGA)是由若干宏单元组成的阵列,每个宏单元中也包括有与门、或门、触发器等,相当于用 PLD 原理做成的逻辑元胞,因而 FPGA 实际上是一个现场可编程的逻辑元胞阵列。之所以称为 FPGA,只是因为其阵列排列方式与门阵列类似而已。

上述各种 PLD 都是一般意义上的 PLD,近年来,可编程技术使用越来越广,又出现了可编程互联、可编程模块和可编程模拟电路等广义的可编程器件。这些已超出本书的范围,此处不予讨论。

2) 按编程方法分类

前面说到,在按编程部位划分的类别中最先出现的是 PROM,现在就以 ROM 为例,介绍各种编程和改写的方法,而这些方法又是与半导体器件的生产工艺密切相关的。

最初的 ROM 是由半导体生产厂制造的,阵列中各点间的连线用厂家专门为用户设计的

掩膜板制作，因而将这种编程方法称为掩膜编程。由于其设计成本高，需有一定的生产批量才有价值，且有一定的风险，所以这种方法只用来生产存放固定数据、固定程序的 ROM 以及函数表、字符发生器等器件。

为省去设计掩膜的昂贵费用，人们研制了一种熔丝编程的 PROM。图 1.8 是这种 PROM 编程机理的示意图，图中每个横线与纵线的交点处皆做有熔丝，因而任何一条横线与纵线都是相联的，编程时利用某一形状特殊，其幅度比工作电流大得多的电流将不需联接处的熔丝烧断即可。显然，因为熔丝一旦烧断便不能恢复导通，这种方法只能一次性编程，故这种方式也是有风险的。用 MOS 工艺制造的 PROM 中，各连接点放的不是熔丝，而是一对反向串联的肖特基二极管，如图 1.9 所示，未编程时纵线与横线间是不通的，编程时对需要联接处加上高的反向电压使其中一个二极管永久性击穿而短路，从而达到该点逻辑联接的目的，这种 PN 结击穿法编程的 PROM 以及后面提到的各种 PROM 都已不使用熔丝作为可编程联接材料了，但习惯上仍将描述阵列中各点的联接情况的图称为熔丝图。

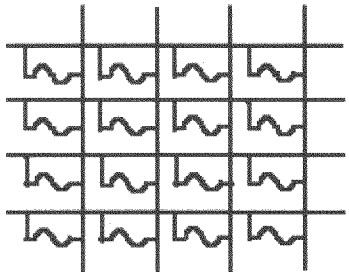


图 1.8　熔丝编程 PROM 示意图

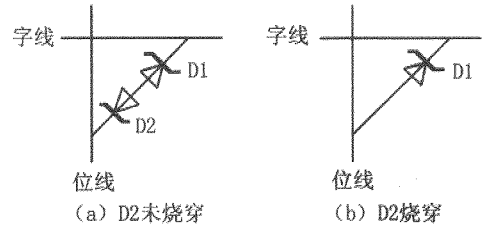


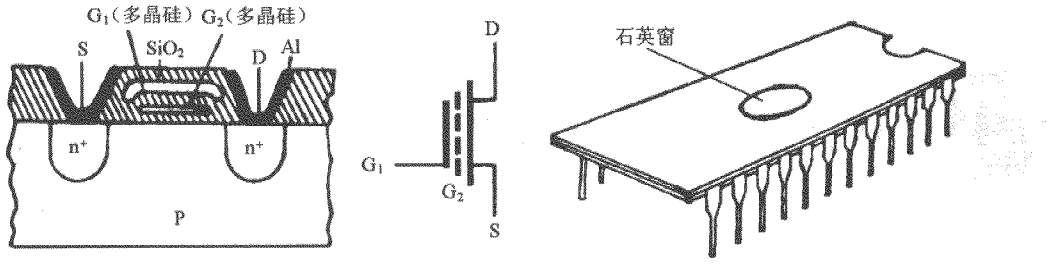
图 1.9　PN 结击穿法 PROM

第三类编程方式称为可擦除 PROM，简称 EPROM(Erasable Programmable ROM)，其编程“熔丝”是一只浮栅雪崩注入型 MOS 管，其结构如图 1.10 所示。该器件与普通 MOS 管的区别在于控制栅 G_1 与沟道之间有一个被包围在二氧化硅之中的多晶硅浮栅(G_2)，当漏源间加有几十伏脉冲电压时沟道中产生雪崩，若在控制栅 G_1 加正电压，则沟道中因雪崩产生的高能电子会穿过二氧化硅绝缘层注入到浮栅 G_2 上。当编程电压去除后，因浮栅被绝缘层所包围，G_2 在编程时所积累的电子无处泄漏，可以长期保持(通常情况下，存放 10 年后，G_2 上的电荷仍能保存 70%左右)。经过这样处理的 MOS 管，因为浮栅上负电荷的作用，其开启电压被提高到电源电压附近，电路正常工作时处于截止状态，而未经处理的 MOS 管，正常工作时(G_1 加正电压)处于导通状态，从而达到了编程的目的。

擦除 EPROM 的方法是将器件放在紫外光(光强 12000 $\mu W/cm^2$)下 2.54 cm(1 in)处照射 20 min，浮栅中的电子获得足够能量，得以穿过氧化层回到衬底中，MOS 管便回到了未编程时的状态，从而将编程信息全部擦去。EPROM 器件的上方有一个石英窗(如图 1.10(c)所示)就是为擦去编程信息而设置的，正常运用时，应用黑色胶纸将其封盖住。

对 EPROM 的编程是在编程器上进行的，编程器也称烧录器，通常与微机联用，用户可以在自己的实验室编程，因而是一种现场可编程器件。

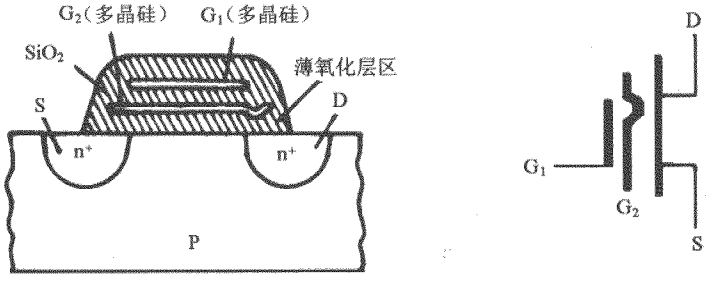
另一种可擦除的 PROM 器件称为 EEPROM 或 E^2PROM，它是一种电擦除的可编程的器件，其编程“熔丝”与 EPROM 结构相仿，只是浮栅 G_2 与漏极间有一薄氧化层，其厚度只有 10



(a) 结构图　　(b) 逻辑符号　　(c) EPROM 外形图

图 1.10　EPROM 的"熔丝"结构

~15 μm,可以产生"隧道效应"。编程时,漏极接地,控制栅加正 20 V 脉冲电压,衬底中的电子通过隧道效应注入浮栅,电压脉冲撤除后浮栅上的电子也可长久保留,使得该管正常工作时(漏极加正电压,栅极上所加正电压通常不高于漏极电压)处于截止状态。而若将其漏极加 20 V 正脉冲电压,控制栅 G_1 接地,则浮栅上的电子又会通过隧道效应回到衬底,正常工作时又处于导通状态,从而达到对该管擦除的目的。由于 E^2ROM 的编程和擦除都是通过在漏极和控制栅上加一定幅度和极性的电脉冲实现,且都是由用户在"现场"用编程器完成的,事实上编程和擦除是同时进行的,每编程一次,就以新的信息代替了原来的信息(有些文献上称为电改写 PROM),整个时间不到 1 秒钟,因而这种方法最受用户青睐。



(a) 结构图　　(b) 逻辑符号

图 1.11　E^2ROM 的"熔丝"结构

综上所述,ROM 的编程方法是按"掩膜 ROM→PROM→EPROM→E^2PROM"次序发展的,PLA、PAL 等 PLD 器件也一样。通常把一次性编程的(如 PROM)称为第一代 PLD,把紫外光(UV)擦除的(如 EPROM)称为第二代 PLD,把电擦除的(如 E^2ROM)称为第三代 PLD。

第二代、第三代 PLD 器件的编程都是在编程器上进行的。1991 年,美国 Lattice 公司又推出一种在系统编程(ISP)器件,编程工作可以不用编程器而直接在目标系统或线路板上进行,因而称为第四代 PLD 器件。

3) 其他分类方法

按照器件工艺,可将 PLD 分为双极型、MOS 型和 CMOS 型;按其集成密度,可以分低密类和高密类等。

原则上,各种分类方法是并行不悖的,但实际上又有一定联系。例如目前大量生产和广泛使用的是高密度 PLD,它们都是 MOS 或 CMOS 型,采用 UV 擦除或电可擦方式的器件。