



北京市高等教育精品教材立项项目

高等院校信息与通信工程系列教材

专用集成电路设计与电子设计自动化

路而红 主编



清华大学出版社



北京市高等教育精品教材立项项目



高等院校信息与通信工程系列教材

专用集成电路设计与电子设计自动化

路而红 主编

清华大学出版社
北 京

内 容 简 介

本书由器件篇、工具篇、语言篇和应用篇组成。器件篇重点介绍常用的和最新的可编程逻辑器件结构及其性能指标;工具篇重点介绍 PC 环境下的电子设计自动化工具,如 MAX+plus II、Quartus II 和 ISE;语言篇介绍两种国际标准化硬件描述语言 VHDL 和 Verilog HDL 及其应用实例;应用篇重点介绍数字系统设计中的新技术,如 IP 核重用技术、SoC 设计技术等,另外还介绍了数字系统设计应用实例。

全书从硬件到软件、从基础到应用对专用集成电路和电子设计自动化的相关技术做了较为全面的介绍。本书 2003 年被列为北京市高等教育精品教材立项项目,可作为高等学校电子信息类、计算机类专业的本科生教材,也可供从事电子设计的工程技术人员参考。

版权所有,翻印必究。举报电话:010-62782989 13901104297 13801310933

本书封面贴有清华大学出版社激光防伪标签,无标签者不得销售。

图书在版编目(CIP)数据

专用集成电路设计与电子设计自动化/路而红主编. —北京:清华大学出版社,2004

(高等院校信息与通信工程系列教材)

ISBN 7-302-08605-2

I. 专… II. 路… III. ①集成电路—电路设计—高等学校—教材 ②电子电路—电路设计:计算机辅助设计—高等学校—教材 IV. ①TN402 ②TN702

中国版本图书馆 CIP 数据核字(2004)第 041660 号

出 版 者:清华大学出版社

<http://www.tup.com.cn>

社 总 机:010-62770175

地 址:北京清华大学学研大厦

邮 编:100084

客户服务:010-62776969

责任编辑:陈国新

版式设计:刘祎森

印 装 者:北京鑫海金澳胶印有限公司

发 行 者:新华书店总店北京发行所

开 本:185×260 印张:25.75 字数:590 千字

版 次:2004 年 7 月第 1 版 2004 年 7 月第 1 次印刷

书 号:ISBN 7-302-08605-2/TN·186

印 数:1~4000

定 价:35.00 元

本书如存在文字不清、漏印以及缺页、倒页、脱页等印装质量问题,请与清华大学出版社出版部联系调换。联系电话:(010)62770175-3103 或(010)62795704

前言

近年来,专用集成电路(ASIC)和电子设计自动化(EDA)技术的发展给数字系统设计带来了革命性的变化。EDA 包含的技术基础有:可编程逻辑器件(PLD)、EDA 设计工具、硬件描述语言。这些技术的发展正在使 ASIC 设计和 EDA 技术应用走向普及。

本书由器件篇、工具篇、语言篇和应用篇组成。

器件篇由第 1 章~第 4 章组成。第 1 章介绍可编程逻辑器件的基本概念和基本原理,第 2 章介绍复杂可编程逻辑器件,第 3 章介绍现场可编程门阵列,第 4 章介绍专用集成电路。本篇的编写原则是,希望读者在了解 PLD 的基本概念和原理的基础上,也能了解目前世界上主流公司 Altera 公司、Xilinx 公司的 PLD 产品,对于实际的可编程逻辑器件的原理、特点及性能指标,给出了详尽的介绍,以方便读者选择和使用 PLD 器件。另外,还介绍了专用集成电路的分类、设计流程、故障分析与测试。

工具篇由第 5 章和第 6 章组成。第 5 章介绍 Altera 公司的 MAX+plus II 10. x 和 Quartus II 3. 0,第 6 章介绍 Xilinx 公司的 ISE 5. x。这些工具的共同特点是:对硬件环境要求低,占用系统资源少,易学易用。书中介绍了这些工具的主要功能及其使用,并按照 FPGA/CPLD 的开发流程介绍工具中的各功能模块,使读者更容易掌握工具的基本使用方法。

语言篇由第 7 章和第 8 章组成。第 7 章介绍 VHDL,第 8 章介绍 Verilog HDL。由于硬件描述语言越来越受到从事硬件设计,特别是从事数字系统设计的人们的关注,所以本书详细介绍了这两种国际标准化硬件描述语言的结构和应用实例,对每个应用实例都作了仿真和综合,在许多实例中都给出了综合电路,希望读者在学习硬件描述语言时真正掌握语言与电路之间的内在联系。

应用篇由第 9 章和第 10 章组成。第 9 章对数字系统设计的基本概念、设计过程及其方法作了阐述,特别介绍了数字系统设计中的新技术,如 IP 核重用技术、SoC 设计流程。第 10 章列举了不同应用领域的数字系统设计实例。

2015/2/10

本书第4章、第6章以及第5章的部分内容由路而红编写,第1章和第3章由洗立勤编写,第8章和第5章的部分内容由李雪梅编写,第7章由高献伟编写,第2章由高献伟和周志全共同编写,第9章由路而红、靳济方和武玉华共同编写,第10章的应用实例由李莉、王曼珠、高献伟和路而红编写。全书由路而红教授负责统稿。

本书的写作与出版得到了2003年北京市高等教育精品教材建设项目和2002年北京市高等学校教学内容及教学方法改革项目的资助。本书的撰写和出版得益于各个时期从事EDA教学和科研工作老师们的成果,没有他们的宝贵经验,也就没有本书的顺利出版。

特别感谢清华大学自动化系阎石教授、北京电子科技学院李战勇教授、北京联合大学王毓银教授、北京交通大学侯建军教授提供的帮助。在本书的编写过程中,还得到了北京电子科技学院电子信息工程系领导和同事们的支持与合作,以及通信工程系许萍老师、王传海同学和电子信息工程系的孔庆钢、王建承、丁杰、袁睿、朱文杰等同学的帮助,在此表示感谢。

由于EDA技术发展快,各种可编程逻辑器件的产品更新更快,加之编者水平有限,难免有疏漏或错误之处,真诚地希望广大读者提出批评和建议。

作者

2004年1月

于北京电子科技学院

E-mail: luerhong@besti.edu.cn

目 录

第 1 章 可编程逻辑器件	1
1.1 引言	1
1.2 PLD 分类	2
1.2.1 按集成度分类	2
1.2.2 按编程工艺分类	3
1.2.3 按与或阵列可编程性分类	4
1.3 PLD 的基本结构	6
1.3.1 低密度 PLD 的基本结构	6
1.3.2 高密度 PLD 的基本结构	6
1.4 PLD 的早期产品	6
1.4.1 PAL 器件	7
1.4.2 PAL 器件输出与反馈结构	8
1.4.3 GAL 器件	10
1.4.4 普通型 GAL 器件	11
1.5 PLD 的发展趋势	17
习题	18
第 2 章 复杂可编程逻辑器件	19
2.1 MAX 系列	19
2.1.1 MAX7000 系列	19
2.1.2 其他 MAX 系列	26
2.2 ACEX1K 系列	29
2.2.1 器件性能	29
2.2.2 结构原理	30
2.3 Cyclone 系列	40
2.4 Stratix 系列	42
2.4.1 器件性能	42

2.4.2	结构原理	43
2.4.3	逻辑阵列块	43
2.4.4	存储器块	50
2.4.5	数字信号处理块	58
习题		70
第3章	现场可编程门阵列	71
3.1	FPGA 概述	71
3.2	XC4000 系列	73
3.2.1	器件性能	73
3.2.2	结构原理	74
3.3	Spartan 系列	79
3.3.1	器件性能	79
3.3.2	结构原理	80
3.3.3	分布式 RAM	86
3.3.4	配置	89
3.4	Virtex 系列	94
3.4.1	器件性能	94
3.4.2	结构原理	96
习题		101
第4章	专用集成电路	102
4.1	引言	102
4.2	ASIC 分类	103
4.3	ASIC 设计流程	104
4.4	ASIC 设计实现	105
4.4.1	全定制设计实现	105
4.4.2	半定制设计实现	106
4.5	ASIC 故障分析与测试	109
4.5.1	ASIC 测试概述	109
4.5.2	ASIC 故障分析	110
4.5.3	ASIC 的可测性设计	116
4.5.4	边界扫描测试	128
习题		135

第 5 章 Altera CPLD 开发工具	137
5.1 MAX+plus II 操作指南	137
5.1.1 MAX+plus II 的安装	139
5.1.2 设计输入	140
5.1.3 设计处理	148
5.1.4 设计检验	153
5.1.5 器件编程	159
5.2 Quartus II 3.0	162
5.2.1 Quartus II 3.0 的安装	163
5.2.2 设计输入	165
5.2.3 设计处理	168
5.2.4 设计仿真	172
5.2.5 器件编程	175
5.2.6 模块编辑器的使用	176
5.2.7 LPM 宏单元库的使用	181
习题	183
第 6 章 Xilinx FPGA 开发工具	184
6.1 ISE 系统简介	184
6.2 ISE 的设计输入	187
6.2.1 工程管理器	187
6.2.2 设计输入	189
6.3 ISE 的功能仿真	203
6.3.1 测试激励的生成	204
6.3.2 启动 ModelSim	206
6.3.3 ModelSim 的仿真	209
6.4 ISE 的设计综合	210
6.4.1 XST 综合属性	211
6.4.2 XST 综合流程	214
6.5 ISE 的设计约束	215
6.5.1 约束编辑器	216
6.5.2 引脚与区域约束编辑器	220
6.6 ISE 的设计实现	225
6.6.1 翻译	227
6.6.2 映射	227
6.6.3 布局布线	229
6.6.4 器件配置	231
6.6.5 功耗分析	235

习题	238
第 7 章 硬件描述语言 VHDL	239
7.1 VHDL 语言要素	239
7.1.1 标识符	239
7.1.2 数据对象	239
7.1.3 数据类型	240
7.1.4 属性	242
7.2 VHDL 运算符	243
7.3 VHDL 模型构成	244
7.3.1 VHDL 模型构成	244
7.3.2 实体说明	244
7.3.3 结构体	246
7.3.4 配置	246
7.3.5 子程序	246
7.3.6 程序包和库	248
7.4 并行语句	250
7.4.1 进程语句	250
7.4.2 信号赋值语句	251
7.4.3 块语句	254
7.4.4 元件例化语句	255
7.4.5 生成语句	257
7.5 顺序语句	259
7.5.1 变量赋值语句	259
7.5.2 if 语句	259
7.5.3 case 语句	260
7.5.4 loop 语句	261
7.5.5 next 语句	262
7.5.6 exit 语句	263
7.5.7 return 语句	263
7.5.8 null 语句	263
7.5.9 wait 语句	265
7.6 常用数字电路的 VHDL 描述	266
7.6.1 常用组合逻辑电路	266
7.6.2 常用时序逻辑电路	273
7.6.3 有限状态机	277
7.6.4 存储器	279
习题	280

第 8 章 硬件描述语言 Verilog HDL	286
8.1 Verilog HDL 引言	286
8.2 Verilog HDL 程序结构	287
8.3 Verilog HDL 数据类型	290
8.3.1 常量	290
8.3.2 变量	291
8.4 运算符	292
8.4.1 算术运算符	292
8.4.2 关系运算符	293
8.4.3 等式运算符	293
8.4.4 逻辑运算符	293
8.4.5 位运算符	294
8.4.6 缩减运算符	294
8.4.7 移位运算符	295
8.4.8 条件运算符	295
8.4.9 位拼接运算符	295
8.4.10 优先级	296
8.5 赋值语句	296
8.5.1 连续赋值语句	296
8.5.2 过程赋值语句	297
8.6 条件语句	299
8.6.1 if-else 语句	299
8.6.2 case 语句	300
8.7 循环语句	302
8.7.1 for 语句	302
8.7.2 while 语句	303
8.8 结构说明语句	304
8.8.1 initial 语句	304
8.8.2 always 块语句	304
8.8.3 task 语句	307
8.8.4 function 语句	307
8.9 语句的顺序执行与并行执行	309
8.10 不同抽象级别的 Verilog 模型	310
8.11 常用数字电路的 Verilog HDL 描述	314
8.11.1 常用组合逻辑电路	314
8.11.2 常用时序逻辑电路	318
8.11.3 存储器	322

8.11.4 有限状态机	323
习题	326
第9章 数字系统设计	328
9.1 概述	328
9.2 EDA 技术	329
9.2.1 EDA 技术发展	330
9.2.2 EDA 系统构成	331
9.2.3 EDA 技术特点	332
9.3 SoC 技术	333
9.3.1 SoC 的硬件结构	334
9.3.2 SoC 的软件特征	335
9.3.3 SoC 的层次结构设计	335
9.3.4 SoC 的软硬件协同设计	336
9.3.5 SoC 的仿真和测试	336
9.4 IP 核重用技术	337
9.4.1 Atlera 公司的 IP 核及其使用	338
9.4.2 Xilinx 公司的 IP 核及其使用	340
9.4.3 具有自主知识产权 IP 核的开发	341
9.5 嵌入式 Nios CPU 设计流程	341
9.5.1 Nios CPU 简介	341
9.5.2 Nios CPU 硬件开发流程	343
9.5.3 Nios CPU 系统模块的创建	344
9.5.4 Nios CPU 的配置与调试	357
9.6 网上相关技术资源	363
习题	364
第10章 数字系统设计举例	365
10.1 RC6 算法的硬件设计	365
10.1.1 RC6 算法基本原理	365
10.1.2 RC6 算法硬件设计	367
10.1.3 RC6 算法的综合与仿真	374
10.2 信号发生器设计	374
10.2.1 信号发生器设计要求	374
10.2.2 信号发生器设计实现	375
10.2.3 信号发生器的仿真	380
10.3 ADC 采样控制电路设计	381
10.3.1 ADC 采样控制电路设计要求	381

10.3.2	ADC 采样控制电路设计实现	382
10.3.3	ADC 采样控制电路的仿真	384
10.4	电梯控制系统设计.....	384
10.4.1	电梯控制系统设计要求.....	384
10.4.2	电梯控制系统设计实现.....	385
10.4.3	电梯控制系统仿真.....	390
10.5	VGA 图像显示控制电路设计	390
10.5.1	VGA 图像显示控制电路设计原理	390
10.5.2	VGA 图像显示控制电路设计实现	392
	习题.....	396
	参考文献.....	398

第 1 章 可编程逻辑器件

1.1 引言

可编程逻辑器件(programmable logic device, PLD)是一种由用户编程以实现某种逻辑功能的新型逻辑器件。它诞生于 20 世纪 70 年代。此后,它随着集成电路技术和计算机技术的进步而迅速发展。

在 PLD 问世之前,要实现一个复杂的数字系统大致有两种途径,一是选用具有固定功能的通用型中小规模数字集成电路(如 74 系列或 CC4000 系列),通过设计制版布线,将它们搭建成系统。二是采用专用集成电路(ASIC)实现,通常是由用户提出特定要求,由半导体厂家完成设计和制造。由于 ASIC 是属于大规模乃至超大规模集成电路,跟通用型集成电路相比,用 ASIC 来实现一个系统有许多优点,例如体积小,重量轻,功耗低,可靠性高,保密性强,性能高,大批量生产时成本低。但是它也有缺点,主要是设计周期长,先期设计成本高,设计风险大,灵活性差,不适用于小批量生产。为了解决上述矛盾,PLD 应运而生。

PLD 虽然是作为一种通用器件生产的,但是它的逻辑功能是由用户通过对器件的编程来设定的。PLD 的出现,打破了由中小规模通用型集成电路和大规模专用集成电路垄断的局面。跟中小规模通用型集成电路相比,用 PLD 实现数字系统,有集成度高,速度快,功耗小,可靠性高等优点。跟大规模专用型集成电路相比,用 PLD 实现数字系统,有研制周期短,先期投资少,投资风险小,修改逻辑设计灵活方便,小批量生产成本低等优势。

PLD 自问世以来,经历了从低密度的可编程只读存储器 PROM,可编程逻辑阵列 PLA,可编程阵列逻辑 PAL,通用阵列逻辑 GAL 到高密度的现场可编程门阵列 FPGA,复杂可编程逻辑器件 CPLD 的发展过程。在此期间,PLD 芯片的集成度规模由每片几百门发展到几百万门,时钟频率由不到 10MHz 发展到几百兆赫,同时,PLD 的功能不断增强,功耗和成本不断降低,结构更加趋于合理,使用变得更加灵活方便。

PLD 的问世及其发展实现了系统设计师和科研人员的一个梦想:利用价格低廉的软件工具在实验室里自己动手快速设计、开发、仿真和

测试数字系统,然后可以快速将设计编程到一块芯片中,并立即投入到实际应用之中。

PLD 电路表示法跟传统表示法有所不同,主要是因为 PLD 的阵列规模十分庞大,若用传统表示法则极不方便。在图 1-1 中给出了 PLD 的三种连接方式。连线交叉处有实点的表示固定连接;有符号“×”的表示编程连接;交叉连线表示不连接或者是擦除单元。

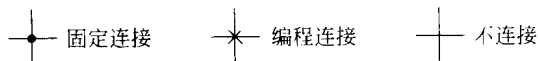


图 1-1 PLD 的连接方式

图 1-2 中是一个三输入与门的两种表示法,即传统表示法和 PLD 电路表示法。在多输入的情况下,PLD 表示法显得简洁。图 1-3 为三输入的与阵列。其中输入变量 A,B,C 分别通过具有互补输出端的输入缓冲器,获得输入变量的原变量和反变量。第一个与门输出为 $D=\overline{A}B$, 第二个与门输出为 $E=A\overline{A}B\overline{B}C\overline{C}=0$, 这种状态称作与门的缺省状态,为了表示方便,可以在相应与门符号中加一个“×”,以代替所有输入项的“×”。如第三个与门所表示的那样, $F=0$ 。第四个与门与所有输入都不接通,即与门输入端悬空,因此 $G=1$,一般将其称作“悬浮 1”状态。

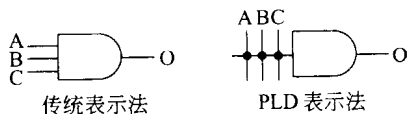


图 1-2 与门表示法

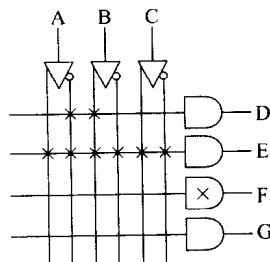


图 1-3 三输入的与阵列

1.2 PLD 分类

1.2.1 按集成度分类

随着集成工艺的发展,PLD 的集成规模越来越大,集成度从每片几百门发展到每片几百万门,甚至每片上千万门。如根据集成度来分类,PLD 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)两大类。

1. 低密度可编程逻辑器件(LDPLD)

低密度 PLD 通常指那些集成度小于 1000 门/片的 PLD。从 20 世纪 70 年代初期至 80 年代中期产生的 PLD,如 PROM、PLA、PAL 和 GAL 均属于 LDPLD,低密度 PLD 同中小规模集成电路相比,有着集成度高、速度快、设计灵活方便、设计周期短等优点,因此得到广泛应用。但随着科学技术发展,低密度 PLD 由于集成密度低的局限,而很难满足

复杂的数字系统规模和性能的要求。

2. 高密度可编程逻辑器件(HDPLD)

高密度 PLD 通常指那些集成度大于 1000 门/片的 PLD。20 世纪 80 年代中期以后产生的 EPLD、CPLD 和 FPGA 均属于 HDPLD。

EPLD(erasable programmable logic device)即可擦除可编程逻辑器件,从结构上看类似于 GAL。但它与 GAL 相比,无论是与阵列的规模还是输出逻辑宏单元的数目都有了大幅度的增加。EPLD 的缺点主要是内部互连能力较弱。

CPLD(complex programmable logic device)为复杂可编程逻辑器件。不同的半导体器件公司推出的 CPLD 结构差异很大,但至少包含可编程逻辑宏单元,可编程 I/O 单元和可编程内部连线这几部分。在 CPLD 中数目众多的逻辑宏单元被排成若干个阵列块,丰富的内部互连线为块与块之间提供了快速、具有固定时延的通路。Xilinx 公司的 XC7000 和 XC9500 系列、Lattice 公司的 ispLSI 系列、Altera 公司的 MAX9000 系列,以及 AMD 公司的 MACH 系列都属于 CPLD。

FPGA(field programmable gate array)为现场可编程门阵列。从结构上看,它包含可编程逻辑块、可编程 I/O 模块和可编程内连线。可编程逻辑块排列成阵列,可编程内连线围绕着阵列。通过对内连线编程,将逻辑块有效地组合起来,实现一定的逻辑功能。FPGA 与 CPLD 之间主要的差别是 CPLD 通过修改具有固定内连电路的逻辑功能来进行编程,而 FPGA 则通过修改内部连线来进行编程。许多半导体器件公司都有自己的 FPGA 产品,例如,Xilinx 公司的 Spartan 系列和 Virtex 系列,Altera 公司的 Stratix 系列和 Cyclone 系列,Actel 公司的 Axcelerator 等。

1.2.2 按编程工艺分类

1. 熔丝或反熔丝编程器件

熔丝编程器件在每个可编程的互连接点上都接有熔丝开关,若接点需要连接则保留熔丝,若接点需要断开则用编程大电流熔断熔丝。熔丝开关的缺点是熔断后不能再恢复,而且熔丝开关很难测试其可靠性,此外熔丝开关占芯片面积较大,对提高器件集成度不利。

反熔丝编程器件以反熔丝开关作为编程元件,同熔丝开关相比,反熔丝开关占芯片面积小得多。反熔丝开关的核心是介质,未编程时开关呈现很高的阻抗,当编程电压加在开关上将介质击穿后,开关则呈现导通状态。

以上两种器件均为非易失一次性编程器件。PROM、PAL 采用了熔丝编程工艺,Actel 的 FPGA 采用了反熔丝编程工艺。

2. 浮栅编程器件

浮栅编程器件采用了浮栅编程技术,包括紫外线擦除电编程的 UVEPROM、电擦除电编程的 EEPROM。它们都是采用悬浮栅存储电荷的方法来保存数据的。

UVEPROM 存储单元的核心是一个浮栅 MOS 管。该 MOS 管绝缘层中埋置一个栅

极,称悬浮栅。编程电压脉冲可对浮栅 MOS 管中的悬浮栅注入电子,使浮栅 MOS 管处于截止状态。而经紫外线照射又可将浮栅中的电子泄放使得浮栅 MOS 管恢复导通。

EEPROM 中的浮栅 MOS 管跟 UV PROM 中的浮栅 MOS 管有所不同,采用了浮栅隧道氧化物工艺。在编程时加编程电脉冲使浮栅注入电子,在擦除时也是靠电脉冲使浮栅中的电子通过隧道效应泄放。

浮栅编程器件属于非易失可重复擦除器件。GAL、EPLD、CPLD 大都采用了以上工艺。

3. SRAM 编程器件

SRAM 即静态存储器,在片中的作用是存储决定系统逻辑功能和互连的配置数据,所以又称配置存储器。SRAM 属于易失性元件,所以每次系统加电时,先要将驻存在外部 EPROM 或硬盘中的编程数据加载到 SRAM 中去。采用 SRAM 技术可以方便地装入新的配置数据实现在线重置。Xilinx 的 FPGA 采用了这种技术。

1.2.3 按与或阵列可编程性分类

根据与阵列及或阵列是否可编程,可以将低密度 PLD 分为三种基本类型。

1. 与阵列固定、或阵列可编程的 PLD

如图 1-4 所示,与阵列是一个全译码的固定阵列,输入为 n 个变量,输出则为 n 个变量的 2^n 个最小项。或阵列是可编程的(由用户编程)。每一个输出可根据需要任意选择一些最小项相或,从而在输出端产生若干个最小项之和的函数。

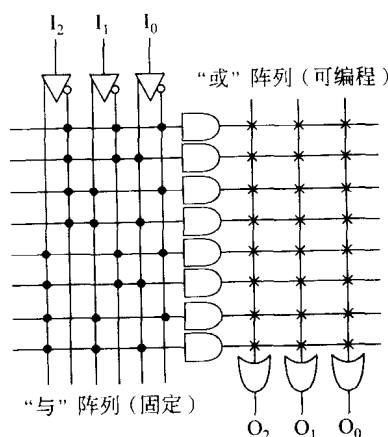


图 1-4 与阵列固定、或阵列可编程的 PLD

最早的产品可编程只读存储器 PROM(programmable read only memory)就是采用这种形式。PROM 能够较方便地实现多输入多输出组合函数。由于它以最小项为基础,因此在设计中无须对函数化简。它的缺点主要是当它的输入增加时,它的与阵列输出(即乘积项)个数以 2 的级数增加,这样导致与乘积项成正比的芯片面积、成本和开关延时相

应迅速增加,再加上总有相当一部分最小项没被使用,使得它的芯片利用率较低。

2. 与阵列和或阵列均可编程的 PLD

如图 1-5 所示,与阵列不再采用全译码方式,而是采用部分译码方式。通过编程使其产生函数所需要的乘积项,乘积项不一定是全部 n 个输入的组合。或阵列也是可编程的,它选择所需要的乘积项相或,在输出端产生乘积项之和的函数。

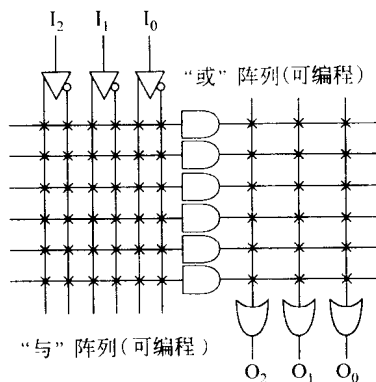


图 1-5 与阵列和或阵列均可编程的 PLD

可编程逻辑阵列 PLA(programmable logic array)采用了以上结构。PLA 跟 PROM 相比,有效地提高了芯片利用率,缩小了系统体积。但由于制造工艺复杂,器件工作速度不够高,PLA 现在已不常使用。

3. 与阵列可编程或阵列固定的 PLD

如图 1-6 所示的 PLD,与阵列可编程,或阵列固定。这种结构不仅能实现大多数逻辑功能,而且可提供最高的性能和速度。

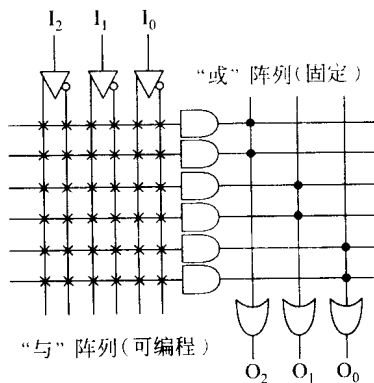


图 1-6 与阵列可编程或阵列固定的 PLD

可编程阵列逻辑 PAL(programmable array logic)采用了以上结构。PAL 的输出结构种类很多,不同的输出结构对应不同的型号。用户可以根据不同的需要,灵活地选择相