

高等学校电子类系列教材

EDA 技术

简明教程

赵刚 等编著

四川大学出版社



耕粵技術簡明教程

赵 刚 张志亮
张 菁 植 涌 编著

四川大学出版社

前 言

数字(电子设计自动化)技术是国家教育部规定的综合性大学电子信息类专业的专业主干课,是通信工程、电子信息科学与技术、计算机科学与技术、微电子、仪器仪表、工业自动化等专业必修的专业主干课。本书是作者在四川大学电子信息学院讲授“数字技术”课程的基础上,总结多年的教学实践经验,以原有的《数字系统的逻辑设计及实现》一书为底本,并结合近年来国内外数字领域所取得的最新发展成果,反复精炼而成。

书中第1章讲述了常用可编程逻辑器件的内部结构及工作原理。第2章讲述了硬件描述语言,包括程序的基本结构、数据类型、描述方式及常用数字电路的设计方法。第3章至第5章分别讲述了目前常用的数字开发系统软件的使用方法。第6章讲述了数字逻辑电路设计仿真软件的使用方法。第7章讲述了电路原理图和印刷电路板图设计软件的使用方法。第8章和第9章讲述了电子系统设计仿真软件的使用方法。第10章介绍了目前最新的自顶向下的数字开发实现流程。

本书有四个主要特点:

覆盖面广。现代数字技术覆盖了从系统级、电路级、芯片级到版图级四个设计层次,书中对前三个层次内容进行了详尽介绍,读者通过本课程的学习,能较全面地掌握数字技术。

内容新颖。随着计算机技术的飞速发展,数字技术中的理论方法和技术手段日新月异。为了让读者掌握最新的数字技术,并考虑到国内高校在实验条件方面的实际情况,书中所介绍的各种数字工具均采用目前在实际产品开发中被广泛使用、具有代表性的开发系统软件和流程。

要点突出。在本科教学实践中,丰富的数字内容与有限的教学学时产生了严重的矛盾。本书在对数字工具讲述时,采用“功能简介—开发流程—库资源—单元设计实例—综合设计实例”为主线的引导性教学方式,通过突出教学

要点，达到有效控制学时的目的。

源注重实效。集成电路技术课程侧重于电子系统的设计及优化，是一门实践性很强的技术课程。为了培养读者的工程实践能力，书中提供了大量典型的设计范例，以帮助初学者尽快入门。

本书若作为本科生“集成电路技术”课程的教材，可安排 苑园学时（含实验课），从第 员章到第 苑章，学时分配依次为：愿 愿 愿 愿 源 愿 愿 愿 愿 源 如作为专科生“集成电路技术”课教材，主要讲授第 员 第 圆 第 猿 第 远 第 苑及第 愿章。作为电路与系统、信号处理、通信工程专业的硕士生和工程硕士的参考教材时，应增大第 源 第 愿 第 怨及第 苑章的学时数，以便安排更多的实验上机时间。

本书亦可作为从事电子信息产业的产品研发人员、工程技术人员的参考书。

本书是由参加编写的教师集体讨论、分工编写、交叉修改后完成的。参加编写的主要人员有赵刚、张志亮、张菁、植涌。本书由赵刚担任主编，并负责大纲拟定、组织编写与统稿工作。

本书在编写过程中，得到了四川大学教务处、四川大学电子信息学院、美国 粤 公司中国（西南地区）孕 技术培训中心的大力支持、帮助和鼓励。借本书出版之际，向他们表示衷心的感谢！

本书在出版过程中，得到了四川大学出版社领导及周树琴老师的诸多帮助，在此表示诚挚的谢意！

限于编者水平，书中难免有不妥和错误之处，恳请读者批评指正（ ）。

编著者

圆 年 远 月于四川大学

目 录

上篇 芯片级 集成电路技术

第 1 章 可编程逻辑器件.....	(猿)
1.1 基于“乘积项”(可编程原理)的可编程逻辑器件.....	(猿)
1.1.1 乘积项和与或阵列.....	(猿)
1.1.2 可编程.....	(源)
1.1.3 可编程.....	(苑)
1.1.4 可编程.....	(怨)
1.1.5 可编程.....	(员)
1.1.6 可编程.....	(猿)
1.1.7 非易失性可编程元件的编程原理.....	(员)
1.2 基于“查找表”(可编程原理)的可编程逻辑器件.....	(员)
1.2.1 可编程.....	(员)
1.2.2 可编程.....	(员)
1.2.3 可编程与可编程使用上的比较.....	(员)
习 题.....	(员)
第 2 章 可编程硬件描述语言.....	(员)
2.1 概 述.....	(员)
2.2 可编程程序基本结构.....	(员)
2.2.1 实体(可编程)说明.....	(员)
2.2.2 构造体(可编程).....	(员)
2.2.3 库(可编程).....	(猿)
2.3 可编程语言的基本数据类型和操作符.....	(猿)
2.3.1 数据对象及其分类.....	(猿)
2.3.2 数据类型.....	(猿)
2.3.3 运算操作符.....	(猿)
2.4 可编程构造体的描述方式.....	(猿)
2.4.1 顺序描述语句.....	(猿)
2.4.2 并行描述语句.....	(源)

圆缘	基本逻辑电路设计	(缘)
圆缘	组合逻辑电路设计	(缘)
圆缘	时序逻辑电路设计	(缘)
	习 题.....	(缘)

第 猿章	配粤哉匀粤译 开发软件及其使用	(猿)
猿	配粤哉匀粤译 开发软件简介	(猿)
猿	配粤哉匀粤译 开发流程	(猿)
猿	配粤哉匀粤译 的原理图编辑器与库资源	(猿)
猿	猿位半加器的设计	(猿)
猿	猿位全加器的设计	(猿)
猿	使用 灾阅盒进行设计	(猿)
猿	配粤哉匀粤译 上的设计实例 简易电子琴设计	(猿)
猿	简易电子琴概述	(猿)
猿	源文件的编辑	(猿)
猿	项目的编译	(猿)
猿	系统仿真	(猿)
猿	目标芯片的编程实现	(猿)
	习 题.....	(猿)

第 源章	匝粤粤译 开发软件及其使用	(源)
源	匝粤粤译 开发软件简介	(源)
源	匝粤粤译 开发流程	(源)
源	匝粤粤译 的库资源	(源)
源	匝粤粤译 上的设计实例	(源)
源	利用向导建立新工程	(源)
源	源文件的编辑	(源)
源	设置编译选项	(源)
源	启动编译过程	(源)
源	设计仿真	(源)
源	目标芯片的编程实现	(源)
源	使用框图进行设计	(源)
	习 题.....	(源)

第 缘章	杂粤粤粤粤粤粤综合器	(缘)
缘	杂粤粤粤粤粤粤综合器简介	(缘)
缘	杂粤粤粤粤粤粤的基本使用	(缘)
缘	有限状态机编译器的使用	(缘)

缘源	利用 酝葬贼皂 进行适配	(猿猿)
缘缘	利用 酝葬皂 进行适配	(猿猿)
缘苑	设计优化技术	(猿愿)
缘苑源	资源共享	(猿愿)
缘苑缘	流水线设计	(猿京)
缘苑猿	寄存器配平	(猿缘)
习 题		(猿苑)

中篇 电路级 猿粤技术

第 远章	猿粤数模混合电路仿真软件	(猿猿)
远源	猿粤简介	(猿猿)
远缘	猿粤的运行环境介绍	(猿源)
远缘源	主菜单与工具栏	(猿源)
远缘缘	元件库	(猿苑)
远缘猿	虚拟仪器库	(猿猿)
远缘源	仿真控制	(猿苑)
远猿	模拟电路的仿真分析	(猿愿)
远猿源	电路原理图的输入	(猿愿)
远猿缘	电路仿真	(猿怨)
远猿猿	电路分析	(猿京)
远源	数字电路的仿真	(猿苑)
远缘	子电路的生成与使用	(猿苑)
习 题		(猿怨)

第 苑章	猿粤 猿粤 印刷电路板设计软件	(猿圆)
苑源	猿粤 猿粤 简介	(猿圆)
苑缘	印刷电路板设计流程	(猿圆)
苑缘源	设计准备	(猿圆)
苑缘缘	参数设置	(猿圆)
苑缘猿	加载元件封装库与网络表	(猿圆)
苑缘源	布局和布线	(猿猿)
苑缘缘	存盘和输出	(猿猿)
苑猿	电路原理图设计	(猿猿)
苑猿源	进入 猿粤 猿粤 集成环境	(猿猿)
苑猿缘	创建新的设计数据库	(猿猿)
苑猿猿	在设计数据库中创建新的文档	(猿缘)
苑猿源	原理图编辑	(猿苑)

苑源	生成网络表文件	(苑源)
苑缘	电路板规划	(苑源)
苑远	加载封装信息库与载入网络表	(苑缘)
苑远源	加载封装信息库	(苑缘)
苑远圆	载入网络表	(苑远)
苑苑	元件布局	(苑源)
苑苑源	自动布局	(苑源)
苑苑圆	手工布局	(苑源)
苑愿	布线	(苑缘)
苑愿源	自动布线	(苑缘)
苑愿圆	手工布线	(苑愿)
苑怨	电路板 预览	(苑愿)
习 题.....	(苑怨)

下篇 系统级 集成电路技术

第 愿章	杂端包大端系统仿真软件	(苑愿)
愿源	杂端包大端简介	(苑愿)
愿源源	用户界面灵活友好	(苑愿)
愿源圆	库资源丰富	(苑愿)
愿源猿	仿真分析与数据处理功能强大	(苑愿)
愿源源	硬件设计接口灵活	(苑愿)
愿源缘	自我诊断功能完善	(苑源)
愿源远	方便的 粤端功能	(苑源)
愿圆	杂端包大端的运行环境	(苑源)
愿圆源	系统窗口	(苑源)
愿圆圆	分析窗口	(苑远)
愿猿	杂端包大端的设计仿真示例	(苑源)
愿猿源	正弦波的平方	(苑源)
愿猿圆	粤端超外差接收机.....	(苑远)
习 题.....	(苑远)
第 怨章	杂端包大端动态仿真系统	(苑怨)
怨源	杂端包大端简介.....	(苑怨)
怨圆	杂端包大端基本使用方法.....	(苑怨)
怨圆源	创建模型	(苑怨)
怨圆圆	在模型中添加模块	(苑怨)
怨圆猿	修改各模块的参数属性	(苑源)

怨源 连接模块完成模型	(缘园)
怨缘 系统仿真	(缘苑)
怨猿 杂音信道子系统	(缘愿)
怨源 杂音信道基本模块简介	(缘园)
怨缘 杂音信道综合实例:数字信号载波传输系统仿真设计	(缘怨)
习 题	(缘园)
第 苑章 灾网蕴代码自动生成	(缘缘)
苑源 概述	(缘缘)
苑源 传统的 灾网粤开发流程	(缘缘)
苑源 灾网蕴代码自动生成流程	(缘园)
苑源 灾网蕴自动生成代码工具	(缘愿)
苑猿 基于 灾网蕴的设计流程	(缘园)
苑源 灾网蕴的库资源	(缘怨)
苑缘 灾网蕴应用实例	(缘缘)
苑缘 云砸滤波器	(缘缘)
苑缘 云栽	(缘园)
苑源 小结	(缘园)
习 题	(缘苑)
参考文献	(缘愿)

上篇 芯片级EDA技术

本篇主要讲述芯片级 EDA 技术。随着工艺技术的发展、集成度的提高和价格成本的下降，可编程逻辑器件 FPGAs (Field Programmable Gate Arrays) 得到了越来越多的应用，并使得 SOC (System on Chip，片上系统) 成为现实，当今世界上最大的两家可编程器件生产商 Altera 和 Xilinx 均可提供该类的解决方案。使用单个的可编程逻辑器件，不仅可以取代原来众多的 74 系列、54 系列中小规模数字集成电路器件，实现从简单到复杂的数字逻辑系统，还可以实现高速的语音处理、图像处理、通信等数字信号处理，甚至可以在其上实现多个用户定制的 32 位并行运算与控制。可编程逻辑器件已经成为现今数字系统实现的一大有力工具。

本篇首先介绍可编程逻辑器件的发展历程以及 Altera 公司的典型器件结构与特性，然后介绍芯片设计和比较常用的硬件描述语言 Verilog；接着介绍 Altera 公司的两套集成开发工具 Quartus II 与 F10K10K；最后针对 Quartus II 综合器性能的不足，补充介绍了现今比较流行的一款逻辑综合优化工具 Synplify。读者学习完本篇各章的内容后，可基本掌握可编程逻辑器件的设计流程与开发技术。

第 1 章 可编程逻辑器件

可编程逻辑器件 (Programmable Logic Device, PLD) 是 20 世纪末出现的一种新型大规模电子集成电路。它的出现,使电子工程师能在实验室甚至家庭里制造出用户自行设计的专用集成电路。可编程逻辑器件经历了从 PAL 器件到 CPLD 器件的发展过程,在结构、工艺、集成度、功能、速度和灵活性方面逐步得到了改进和提高。

可编程逻辑器件的发展大致经历了如下过程:

- ① 20 世纪 70 年代中期,熔丝编程的 PAL 器件和 PLD 器件是最早的可编程逻辑器件。
- ② 20 世纪 70 年代末期,Intel 公司开始推出 CPLD 器件。
- ③ 20 世纪 80 年代初期,Altera 公司发明电擦写的,使用上比 CPLD 更灵活的 FPLD 器件。

④ 20 世纪 80 年代中期,Xilinx 公司提出现场可编程概念,同时生产出世界上第一片现场可编程门阵列 (FPGA) 器件。同一时期,Altera 公司推出可擦除的可编程逻辑器件 (EEPROM),也称为早期的 (FPGA) 较 FPLD 器件有更高的集成度,可以用紫外线或电擦除。

⑤ 20 世纪 90 年代末,Altera 公司又提出在系统可编程 (ISP) 技术,并且推出了一系列具有在系统可编程能力的复杂可编程逻辑器件 (CPLD)。

近年来,可编程逻辑器件 (FPGA) 已在工业上得到了广泛应用,市场份额逐年提高。随着制造工艺的进步,可编程逻辑器件正向高密高速方向迅速发展,使其在一块芯片内实现一个完整的电子系统 (单片系统) 成为可能。目前,各大集成电路厂商正在大量地研究模拟可编程器件,在不远的将来,数模混合可编程器件将使电子设计及实现技术发生质的飞跃。

1.1 基于“乘积项 (AND-OR) 的可编程逻辑器件

1.1.1 乘积项和与或阵列

乘积项是布尔代数的一个术语,指与运算的输出项。

具体实现过程如下:输入信号首先通过一个与矩阵,产生一系列输入信号相与的组合,每组组合都称为一个乘积项,然后这些乘积项在或矩阵中相加,再经输出单元或宏单元输出。

基于乘积项的可编程逻辑器件包含两个基本部分：一个是逻辑阵列，另一个是输出单元或宏单元。逻辑阵列是用户可编程的部分，它由与矩阵、或矩阵和反相器所组成。宏单元的作用是使设计者能改变阵列的输出结构。

与或这种结构可直接实现任何以积之和形式表达的逻辑，任何组合逻辑功能都可以通过采用卡诺图和摩根定理的方法得到积之和的逻辑方程。

与或阵列的结构可以通过改变与或阵列的连接实现不同的逻辑功能。无论改变与阵列还是或阵列的连接，都可以使所实现的逻辑函数发生变化。表 1-1 给出各种器件对应的与或阵列结构：

表 1-1 各种可编程器件对应的与或阵列结构表

器件类别	与或阵列结构
PLD	与阵列固定、或阵列可编程
CPLD	与阵列、或阵列均可编程
FPGA	与阵列可编程、或阵列固定

随着对可编程器件研究的深入，前两种形式的与或阵列结构暴露出一定的缺陷。第 1 种结构的器件在输入数目增加时，与阵列的输出信号线数目以 2 的级数增加，第 2 种结构的器件制造工艺复杂，器件工作速度慢。

基于前两类与或阵列结构的可编程器件处于被淘汰的边缘，相对地，第 3 种形式具备一定的技术优势，是 FPGA 目前发展的主流。

【例 1-1】 试用与或阵列实现一位全加器。

解：一位全加器的输入变量为加数 A_i 和低位进位 C_{i-1} ，输出变量为和数 S_i 及进位输出 C_i 。经过化简的最简与或表达式的逻辑方程如下：

$$S_i = A_i \oplus C_{i-1} \oplus A_i C_{i-1}$$

$$C_i = A_i C_{i-1} + A_i B_i + B_i C_{i-1}$$

可用一个 3 输入 2 输出的与或阵列实现该逻辑函数，如图 1-1 所示。

与或阵列的与阵列输入线、乘积项、或门和输出线等在具体器件内都是有限的，被称为可编程逻辑器件的资源。上面这个例子中，使用的资源有 3 根与阵列输入线、2 个乘积项、2 个 4 输入或门和 2 个三输入或门。

与或阵列在可编程器件中只能实现组合电路功能，时序电路的功能需由包含触发器或寄存器的宏单元来实现。

1.2.1 只读存储器

只读存储器 ROM 存储原封不动的数据只能读出，不能写入。工作时，在地址码

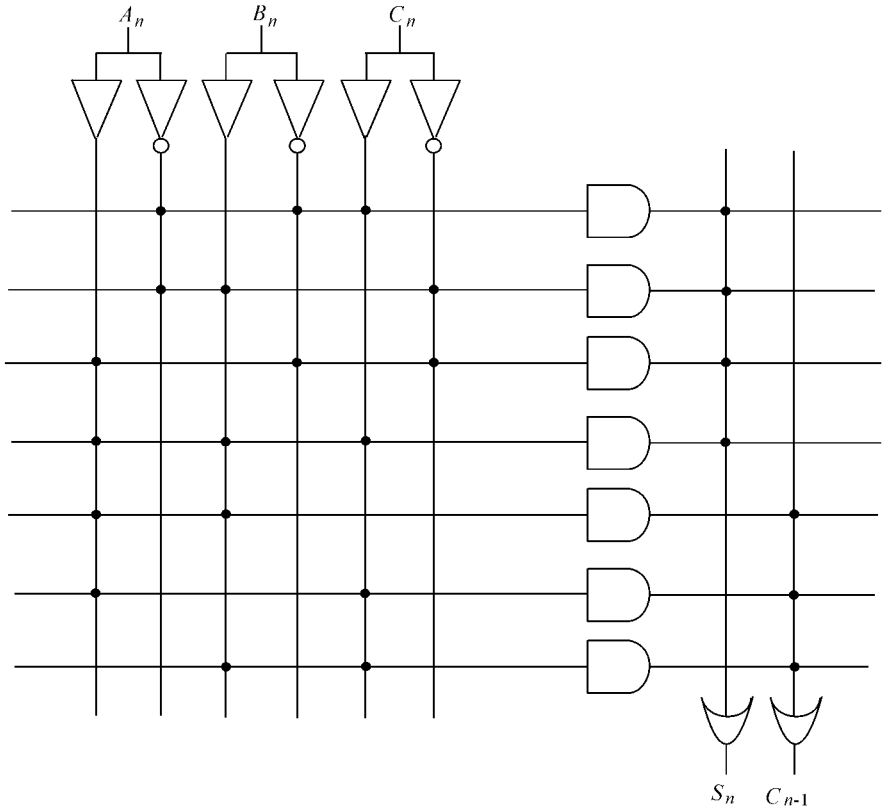


图 4.1.6 与或阵列实现的全加器

输入端输入一个确定的地址信号,便可以在输出端得到一个事先存入的确定数据。它的最大优点是具有非易失性,即使停止供电,阵列中存储的信息也不会丢失,因此可应用于一些需要永久保存数据的场合。

阵列包含一个不可编程的与阵列和一个可编程的或阵列,分别用于地址译码和用作存储阵列。图 4.1.6 是阵列的基本结构框图。图中地址信号 $A_n \sim A_0$ 是与阵列的 n 个输入变量,经不可编程的与阵列全译码后输出 n 条字线 $W_n \sim W_0$,每条字线对应一个最小项(乘积项)。当地址译码器选中某一位后,可编程的或阵列按编程结果将该字的所有位同时读出。

用户对可编程只读存储器阵列中的或阵列可以编程,并仅可以编程一次(即只能写入一次数据)。若我们改变某一位线与译码所得字线的连接情况,便可以改变这一位上的输出。从存储器角度看,我们可以说是改变了这一位所存的数据。从组合逻辑电路来看,我们可以说是改变了这一位的逻辑函数。只要对或阵列进行适当的编程,即一次性地向阵列中写入适当的数据,我们就可在任一输出位上实现输入的任意组合逻辑函数。正是基于这一点,阵列可作为逻辑器件使用。阵列的容量用字个数与位个数的乘积来表示。

【例 4.1.1】用阵列实现源位二进制码到格雷码的转换。

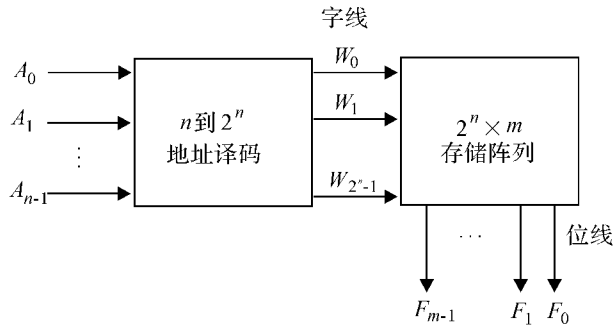


图 1.10 译码的一般结构框图

解：①列出源二进制码转换为格雷码的真值表，见表 1.11。我们只需要把 $W_{m-1}, W_{m-2}, \dots, W_0$ 作为地址信号，把 $F_{m-1}, F_{m-2}, \dots, F_0$ 作为数据写入到一片容量为 $2^n \times m$ 的存储器中，即可实现代码的转换。使用时，将待转换的二进制码加在存储器地址输入端，在存储器的数据输出端就可得到转换后的格雷码。

表 1.11 四位二进制码转换为格雷码的真值表

二进制码				格雷码			
W_{m-1}	W_{m-2}	W_{m-1}	W_0	F_{m-1}	F_{m-2}	F_{m-1}	F_0
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1
0	0	1	0	0	0	1	1
0	0	1	1	0	0	1	0
0	1	0	0	0	1	1	0
0	1	0	1	0	1	0	1
0	1	1	0	0	1	0	0
0	1	1	1	0	1	0	1
1	0	0	0	1	1	0	0
1	0	0	1	1	1	0	1
1	0	1	0	1	1	1	1
1	0	1	1	1	1	1	0
1	1	0	0	1	0	1	0
1	1	0	1	1	0	0	1
1	1	1	0	1	0	0	0
1	1	1	1	1	0	0	1

② 由真值表写出最小项表达式：

是固定的,而是可以按照用户使用的要求来设计的。它所产生的乘积项数目总是小于 圆项,每一个乘积项也不一定是全部 灶个输入信号的组合,而是根据需要来选择。孕阵的或阵列则和孕阵的或阵列相似,或阵列的每个输出都可以任意选用所需要的乘积项。即是说,孕阵的与阵列和或阵列都是可以编程的。孕阵的容量用阵列的与门数伊或门数来表示,图 员源所示的孕阵的容量为 远伊 猿

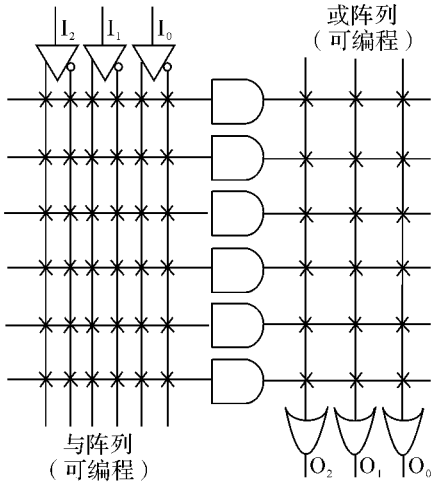


图 员源 孕阵的阵列结构图

【例 员源】用孕阵实现源位二进制码到格雷码的转换。

解: ①为尽可能地减少孕阵的容量,应首先对多输出函数进行化简,得到如下最简与原或表达式:

$$\begin{aligned}
 G_3 &= B_3 \\
 G_2 &= B_3 \oplus B_2 \\
 G_1 &= B_2 \oplus B_1 \\
 G_0 &= B_1 \oplus B_0
 \end{aligned}$$

②选择孕阵芯片实现代码转换器。

化简后的多输出函数共有 苑个不同的与项和 源个输出,可选用容量为 源输入的 愿伊原孕阵来实现。图 员缘给出了一个实现代码转换的孕阵阵列图。与图 员源所示的孕阵实现的电路相比较,两者的逻辑功能完全相同,但前者孕阵容量是 员远伊原而后者的孕阵容量是 愿伊原,实际只需 苑伊原,这证明了孕阵阵列具有较高利用率的特点。

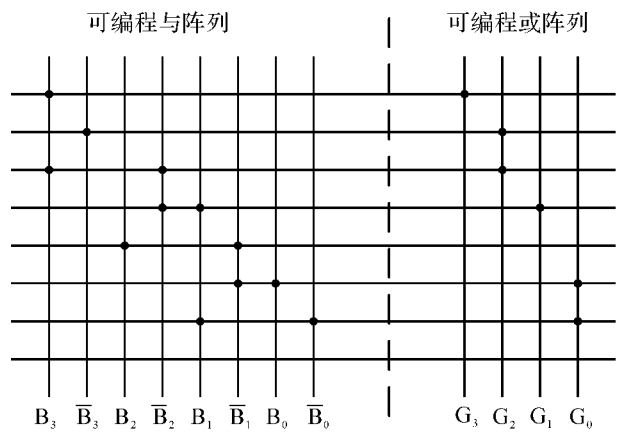


图 员缘 四位二进制码到格雷码转换的孕阵阵列图

孕阵因芯片面积大,制造工艺复杂,工作速度较慢等原因,随着孕阵的发展,现已很少使用。

4.1 缓冲器、与门、或门

4.1.1 缓冲器、与门、或门在 PLD 中的表示方法

由于 PLD 结构与通常的集成电路有很大的不同, 因而对 PLD 器件采用新的逻辑符号表示法。

(1) 输入缓冲器的表示方法

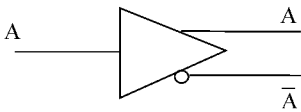
图 4-1-1 给出了 PLD 使用的缓冲器的符号, 它的两个输出分别是输入的原码和反码。

(2) 与门的表示方法

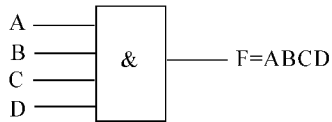
图 4-1-2 给出了与门的标准逻辑符号, 图 4-1-3 为与门在 PLD 中常用的表示方法。在这种描述方法中, 四输入与门的输入部分只画一根线, 通常称为乘积线。四个输入分别用四根与乘积线相垂直的竖线送入, 这种多输入的和门在 PLD 中构成乘积项。竖线和乘积线的交叉点均有一个耦合元件, 交叉点的“·”表示可编程连接, “·”表示固定连接或编程后的连接, 无任何标记则表示不连接。

(3) 或门的表示方法

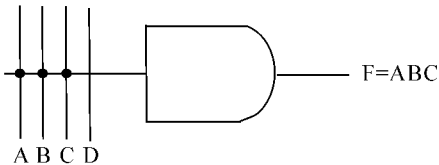
图 4-1-4 分别给出了或门的标准逻辑符号和 PLD 中采用的表示方法。图 4-1-5 表示该或门有四个乘积项 P_1, P_2, P_3, P_4 输入, 因此有 $F=P_1+P_2+P_3+P_4$ 。



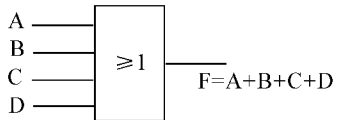
(a) PLD 输入缓冲器



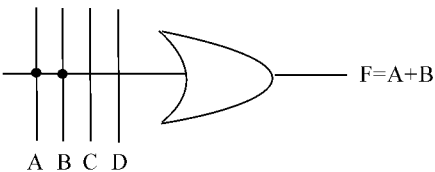
(b) 与门标准逻辑符号



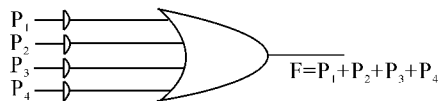
(c) 与门在 PLD 中的表示



(d) 或门标准逻辑符号



(e) 或门在 PLD 中的表示



(f) 四个乘积项的或门

图 4-1-5 缓冲器、与门、或门表示方法图

(4) 阵列图

阵列图是用来描述 PLD 内部与或阵列连接关系的一种特殊的逻辑电路图。图 4-1-6 给