

高等学校教材

FPGA设计基础

王传新 主编



高等教育出版社

高等学校教材

TP332.1/51

2007

FPGA设计基础

王传新 主编



高等教育出版社

内容简介

本书以培养学生实际工程设计能力为目的,以FPGA工作原理为基础,以FPGA设计流程为主线,系统完整地介绍了当前在电子技术领域应用较广的FPGA设计软件Quartus II 6.0及其辅助工具MATLAB 7.0、DSP Builder v6.0、Nios II EDS 6.0、MegaCore 6.0的基本使用方法。

全书共6章。第1章简介EDA技术和PLD工作原理;第2章着重介绍Quartus II 6.0的基本使用方法和技巧;第3章通过大量应用电路介绍VHDL语法规则和编程要领;第4章介绍FPGA设计的常用方法和方案;第5章讲述基于FPGA的SOPC技术及其基本设计方法;第6章讲述完成数字系统设计的全过程和设计范例。书中列举的VHDL程序和FPGA设计范例全部通过了硬件环境测试。本书选取典型范例,重在应用,概念简明,深入浅出,结构清晰,学习者可以在短时间内了解和掌握EDA技术的基本理论和FPGA设计的实用技术,并为今后进一步学习打下坚实的理论与实践基础。

本书可作为高等院校电气、电子信息类专业开设EDA技术、SOPC技术、FPGA/CPLD设计等课程的理论课教材或实践课指导书,还可供开发FPGA应用技术的工程技术人员参考。

图书在版编目(CIP)数据

FPGA设计基础/王传新主编. —北京:高等教育出版社,2007.9

ISBN 978-7-04-022483-2

I. F… II. 王… III. 可编程逻辑器件-系统设计-高等学校-教材 IV. TP332.1

中国版本图书馆CIP数据核字(2007)第124833号

出版发行	高等教育出版社	购书热线	010-58581118
社 址	北京市西城区德外大街4号	免费咨询	800-810-0598
邮政编码	100011	网 址	http://www.hep.edu.cn
总 机	010-58581000		http://www.hep.com.cn
经 销	蓝色畅想图书发行有限公司	网上订购	http://www.landracom.com
印 刷	中国农业出版社印刷厂		http://www.landracom.cn
		畅想教育	http://www.widedu.com
开 本	787×960 1/16	版 次	2007年9月第1版
印 张	16.5	印 次	2007年9月第1次印刷
字 数	310 000	定 价	20.90元

本书如有缺页、倒页、脱页等质量问题,请到所购图书销售部门联系调换。

版权所有 侵权必究

物料号 22483-00

前 言

近年来 EDA 技术在电子技术领域引发的技术革命,推动着电子技术的迅猛发展,为世人所瞩目。而以 FPGA 为代表的可编程逻辑器件的应用,更是受到业内人士的普遍关注。伴随着大规模集成电路和计算机技术的高速发展,在涉及工业自动化、仪器仪表、计算机设计与应用、通信、国防等领域的电子系统中,FPGA 技术的含量正以惊人的速度提升。将尽可能大的、完整的电子系统在单一 FPGA 芯片中实现已成为现实,电子类新技术项目的开发也更多地依赖于 FPGA 技术的应用。

当今的电子系统设计已不再是利用各种通用 IC 进行 PCB 板级的设计与调试,而是转向以 ASIC 或大规模 FPGA 为物理载体的系统芯片的设计,这就是 SOC 设计或 SOPC 设计。FPGA 芯片可靠性高,功耗低,操作灵活,可以重复擦写无限次,开发软件界面友好,为学习、研发 SOC 或 SOPC 提供了便利的条件。围绕 FPGA 应用技术展开的教学活动已被列入电气、电子信息类及相关专业本科学生学习电子技术、EDA 技术的教学计划中。

淡化理论,强调实践,利用较少学时(40~60 学时),让学生在实践中体验进行 FPGA 设计的全过程,掌握规范的 FPGA 设计流程、方法、要领,明确应做什么,该怎样去做,进而在具备基本的 FPGA 应用能力的基础上,独立完成一个小型的、实用的电子系统设计项目;用较小的篇幅,简洁、通俗的语言,帮助学生掌握 FPGA 设计的基本理论、基本技能,是编者多年的教学期望和教学经验。

为了使本书的内容尽量追随电子技术前进的步伐,本书使用当前在教学中应用较广的由 Altera 公司研发的 FPGA 设计软件平台 Quartus II 6.0 及其辅助工具软件 MATLAB 7.0、DSP Builder v6.0、Nios II EDS 6.0、MegaCore 6.0, FPGA 目标芯片选用 Cyclone 系列器件 EP1C6Q240C8,硬件描述语言采用的是 VHDL。然而,EDA 技术的发展日新月异,全球致力于 FPGA 研发的公司、厂家不胜枚举,有关 FPGA 的新技术、新方法、新器件更是层出不穷(据悉,Quartus II 每半年升级一次)。希望通过对本书的学习,读者能够对 FPGA 的设计、开发过程有较清晰的概念、较系统的认识,能够举一反三,能够建立自主学习、渴求知识、接受挑战的能力,这也是作者编写本书的又一初衷。

本书共分为 6 章。第 1 章以简介 EDA 技术和可编程逻辑器件工作原理为

II 前 言

基础,帮助读者建立 EDA 设计流程和 FPGA 编程下载的概念。第 2 章通过软件安装、文本输入设计和图形输入设计三个板块系统讲述了 FPGA 开发软件 Quartus II 6.0 的基本使用方法和技巧,其中包括授权、编辑、综合、编译、仿真、纠错、适配、下载、配置、检验等环节以及层次化设计、生成 RTL 电路、使用嵌入式逻辑分析仪等 Quartus II 特有的功能。第 3 章从硬件描述语言 VHDL 基本结构入手,本着够用易懂的原则,讲述 VHDL 的基本语言要素和语句。实践证明,写一个 VHDL 程序不难,写好一个 VHDL 程序不易。为了帮助读者尽快掌握 VHDL 的语法规则和编程要领,列举了数字电路中常用单元电路的设计范例,且针对每种单元电路的逻辑功能特点采用了多种描述方法。第 4 章从有限状态机设计、多位显示器动态扫描、调用 LPM 元件、使用 IP 核、消除“毛刺”等多方面介绍了实施 FPGA 设计的常用方法。第 5 章通过范例讲述了基于 FPGA 的 SOPC 技术及其基本设计方法,其中包括 SOPC Builder、Nios II 软核、DSP Builder、Simulink 信号编译器的基本用法。第 6 章同样通过设计范例讲述了从接受设计选题、设计方案论证到完成设计验证、写出设计报告的 FPGA 设计的全过程,最后作为对 FPGA 的设计实践,向读者推荐了适合本科生进行初步的 FPGA 应用的设计选题。设计选题体现典型性、实用性、灵活性、创意性和趣味性,以适应不同层次、不同能力、不同爱好者的需要。全书力求结构清晰简洁,内容深入浅出,考虑到基础知识与认知规律的有机结合,注重培养学生的自学能力、动手能力、应变能力和创新意识。

本书由王传新、廖文江、曲金泽、程洪波共同编写。其中廖文江执笔第 2 章和 5.3 节,曲金泽执笔第 6 章和附录部分,程洪波执笔第 3 章,朱璐对本书的 VHDL 程序进行了调试和验证,崔妍、张娜为本书绘制了部分图例,北京宇华新创科技有限公司的曹力彬先生提供了部分设计范例。王传新担任主编,执笔其它章节,并负责全书的修改与统稿工作。

杭州电子科技大学潘松教授、北京宇华新创科技有限公司杨洪伟经理对本书的出版提出了宝贵的意见。在本书的编写和出版过程中,受到北京联合大学自动化学院、信息学院和电子信息实验实训基地领导的关心和指导,受到同事李淑芬、耿钰、贺玲芳、张明莉、王珏等的大力支持和帮助。首都师范大学信息工程学院孙卫真研究员作为本书的主审,认真审阅了全部书稿,并提出了许多中肯的修改意见。在此,一并向他们表示敬意和感谢。

由于 FPGA 应用技术发展快,更新快,而编者的能力和水平有限,本书中必存在许多不妥之处,恳请读者斧正。

编 者

2007 年 6 月

目 录

第 1 章 概述

1.1	关于 EDA 技术	1
1.1.1	EDA 技术	1
1.1.2	EDA 设计流程	2
1.1.3	硬件描述语言 VHDL	4
1.2	可编程逻辑器件与 FPGA	5
1.2.1	PLD 的工作原理	5
1.2.2	PLD 的编程工艺	11
1.2.3	PLD 的开发软件	13

第 2 章 Quartus II 设计向导

2.1	Quartus II 6.0 软件的安装	15
2.1.1	对 PC 机系统的配置要求	15
2.1.2	Quartus II 6.0 软件的安装	16
2.1.3	Quartus II 6.0 软件的授权	18
2.2	文本输入设计	20
2.2.1	设计输入	20
2.2.2	创建工程	23
2.2.3	综合分析	26
2.2.4	仿真分析	28
2.2.5	生成 RTL 电路	35
2.2.6	锁定引脚	36
2.2.7	编程下载	38
2.2.8	使用嵌入式逻辑分析仪	44
2.3	图形输入设计	48

2.3.1	基本设计流程	49
2.3.2	层次化设计	55

第 3 章 VHDL 设计初步

3.1	VHDL 程序结构	61
3.2	VHDL 语言要素及规则	65
3.2.1	数据对象	65
3.2.2	数据类型	67
3.2.3	基本运算符	70
3.2.4	属性	72
3.3	VHDL 基本并行语句	73
3.3.1	信号赋值语句	73
3.3.2	进程语句	74
3.3.3	元件例化语句	75
3.3.4	子程序调用语句	77
3.3.5	块语句	78
3.4	VHDL 基本顺序语句	79
3.4.1	赋值语句	79
3.4.2	条件语句	79
3.4.3	CASE 语句	80
3.4.4	循环语句	81
3.5	组合电路设计范例	82
3.5.1	译码器	82
3.5.2	编码器与编码转换器	86
3.5.3	多路选择器	88
3.5.4	运算器	91
3.6	时序电路设计范例	94
3.6.1	触发器	94
3.6.2	锁存器	96
3.6.3	移位寄存器	97
3.6.4	计数器	99
3.6.5	分频器	103

第 4 章 FPGA 设计的常用方法

4.1	有限状态机设计	105
4.1.1	使用状态机的优势	105
4.1.2	Moore 型有限状态机设计	106
4.1.3	Mealy 型有限状态机设计	115
4.2	实施多位显示器动态扫描	118
4.2.1	扫描显示器的必要性	118
4.2.2	扫描显示器电路设计	119
4.3	调用 LPM 元件	122
4.3.1	调用 LPM_ROM 元件	123
4.3.2	调用 LPM_RAM 元件	131
4.3.3	调用 LPM_FIFO 元件	132
4.3.4	调用嵌入式锁相环	133
4.4	使用 IP 核	135
4.4.1	关于 IP 核	135
4.4.2	安装 MegaCore	137
4.4.3	IP 核应用范例	138
4.5	消除“毛刺”的常用措施	146
4.5.1	出现“毛刺”的主要原因	146
4.5.2	消除“毛刺”的常用措施	146
4.6	防止按键抖动的设计方案	148
4.6.1	按键及其抖动	148
4.6.2	防抖动电路的设计方案	149

第 5 章 SOPC 设计入门

5.1	关于 SOPC 技术	155
5.2	SOPC 设计软件的安装	159
5.2.1	安装 Nios II EDS 6.0	159
5.2.2	安装 MATLAB 7.0	160
5.2.3	安装 DSP Builder v6.0	162
5.3	Nios II 系统设计入门	163
5.3.1	Nois II 软核处理器设计	164

5.3.2	Nios II 应用程序开发	178
5.4	DSP 系统设计入门	181
5.4.1	DSP Builder 模型设计	182
5.4.2	Simulink 模型仿真	189
5.4.3	SignalCompiler 设计文件转换及综合	192
5.4.4	Quartus II 仿真、适配、下载	193

第 6 章 FPGA 设计实践

6.1	设计范例	195
6.1.1	系统设计基本流程	195
6.1.2	频率计设计	197
6.1.3	交通灯控制系统设计	213
6.2	设计选题	224
6.2.1	数字跑表	224
6.2.2	多功能电子表	224
6.2.3	数字密码锁	225
6.2.4	出租车计费器	225
6.2.5	交通灯控制器	226
6.2.6	自适应频率计	227
6.2.7	数字式频率合成器	228
6.2.8	多功能信号发生器	228
6.2.9	电梯控制器	229
6.2.10	数字式竞赛抢答器	230
6.2.11	乒乓球比赛游戏机	230
6.2.12	彩灯控制器	231
6.2.13	简易乐曲发生器	232
6.2.14	简易存储示波器	234

附录 GW48 系列 SOPC/EDA 开发系统使用说明

附录 1	开发系统使用介绍	236
附录 2	电路模式使用说明	244
附录 3	开发系统信号名与目标芯片引脚号对照表	251

参考文献

概 述

1.1 关于 EDA 技术

1.1.1 EDA 技术

EDA(Electronic Design Automation)技术是指以计算机为工作平台,融合应用电子技术、计算机技术、信息处理及智能化技术,进行电子产品自动化设计的一门新技术。

EDA 技术作为现代电子设计技术的核心,依赖于功能强大的计算机,在 EDA 工具软件平台上,对以硬件描述语言为逻辑描述手段完成的设计文件,自动地完成逻辑编译、逻辑化简、逻辑分割、逻辑综合、布局布线,以及逻辑优化和仿真测试,直至实现既定的电子电路的功能。

在硬件方面,EDA 技术融合了大规模集成电路制造技术、集成电路版图设计技术、可编程器件编程技术、自动测试技术等;在计算机辅助工程方面融合了计算机辅助设计 CAD(Computer Assist Design)、计算机辅助制造 CAM(Computer Assist Manufacturing)、计算机辅助测试 CAT(Computer Assist Test)、计算机辅助分析 CAA(Computer Assist Analyzer)、计算机辅助工程 CAE(Computer Assist Engineering Design)技术以及多种计算机语言的设计概念;在现代电子学方面容纳了诸如计算机设计技术、电子线路设计理论、数字信号处理技术、数字系统建模和优化技术以及基于微波技术的长线技术理论等。因此,EDA 技术为现代电子理论和设计的表达与实现提供了可能。

当前,利用 EDA 技术进行电子电路设计的主要方法,是基于可编程器件完成专用集成电路 ASIC(Application Specific Integrated Circuit)的实现。ASIC 作为最终的物理平台,是集中容纳用户通过 EDA 技术将电子应用系统的既定功能、技术指标和个性创意具体实现的硬件实体。

在传统的电路设计中,分立元件、中小规模集成电路的功能、参数、规格是相对固定的,人们把大量的精力和时间花费在元器件的选配和电路结构的可行性分析上,采用的设计方法只能是自底向上,即整体电路是由底层器件、电路一层层级联起来的。在这个设计过程中的任一时刻,若发生底层目标器件的缺损,总体或局部技术参数的变更,甚至由于市场竞争而临时提出降低系统成本、提高运行速度等不可预测的外部因素,都将可能使前期的工作前功尽弃,设计工作又得重新从底层开始。可见,自底向上是一种低效、低可靠性、高成本的设计方法。

在 EDA 技术应用中广泛采用自顶向下的方法设计电路,工程项目的设计流程包括:用自然语言描述功能特性和技术指标,用硬件描述语言建立系统描述→行为描述→结构描述→逻辑描述,用计算机开发软件进行功能仿真→时序仿真→硬件测试。设计者在整个设计过程中可根据需要,随心所欲地改变器件内部结构乃至器件外部引脚的功能,可以将系统电路分解为各个模块,也可以将多个模块集合在一起,而不必顾及目标器件的技术细节。正是采用了基于可编程器件的、利用计算机完成的自顶向下的设计方法,大大减少了功能芯片的数量,减轻了设计电路板图的工作量,缩小了整体电路的体积,提高了系统的可靠性,加快了研发速度,降低了产品设计成本。

可以说,EDA 技术打破了软件设计和硬件设计间的壁垒,是一门综合性学科,一种多方位技能技术。它将设计效率和产品性能合二为一,代表了电子设计技术和电子应用技术的发展方向。

1.1.2 EDA 设计流程

可编程器件的开发系统由硬件和软件两部分组成。硬件包括计算机和专用的编程器、编程电缆等,软件泛指各种编程软件。

可编程器件有模拟、数字之分。目前应用广泛、技术完善、便于开发者,当属数字式可编程逻辑器件。

了解利用 EDA 技术进行可编程逻辑器件设计开发的流程,对于正确选择和使用 EDA 软件、优化设计项目、提高设计效率十分有益。一个完整的 EDA 设计流程,既是自顶向下设计方法的具体实施途径,也是 EDA 工具软件本身的组成结构。

图 1.1.1 是应用于可编程逻辑器件的 EDA 设计流程。

一、编辑输入(Editing)

首先,依据人们描述电路功能的习惯,将设计系统以一定的方式输入计算机。使用 EDA 工具软件的设计输入方法有文本输入、原理图输入、状态图输入、波形图输入等多种,当然也可以将它们混合使用。

① 文本输入。在 EDA 工具软件的文本编辑界面上,输入用某种硬件描述语言表述的电路设计文本。这种方法与传统的计算机软件语言编辑输入类似。

② 原理图输入。在 EDA 工具软件的图形编辑界面上,绘制能完成预定功能的电路原理图。原理图由逻辑器件的符号和连线构成。图中的逻辑器件可以是 EDA 软件库中预留的功能模块,如各种逻辑门、触发器、74 系列器件等,也可以是一些 IP(Intellectual Property)功能模块或设计者自定义的模块。

③ 状态图输入。依据电路的控制条件和状态转换的因果关系,在 EDA 工具软件的状态图编辑界面上,绘制时序电路的状态转换流程图。

④ 波形图输入。在 EDA 工具软件的波形图编辑界面上,绘制电路输入、输出信号间的波形关系,然后由 EDA 编译器据此完成电路的设计。

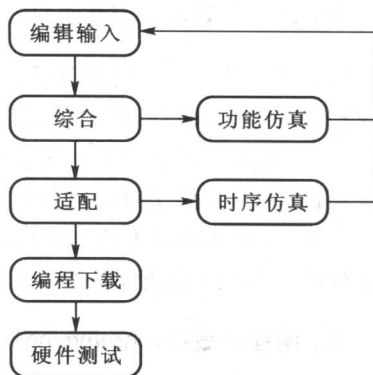


图 1.1.1 应用于可编程逻辑器件的 EDA 设计流程

二、综合(Synthesis)

综合是将由硬件描述语言表述的电路行为描述转换为低级的、与可编程逻辑器件基本结构相映射的网表文件或电路连线图。简单地讲,综合的过程就是将软件设计与硬件结构挂钩,是文字描述与硬件实现间的一座桥梁,是将软件转换为硬件电路的关键步骤。

综合与软件语言的编译(Compilation)不同,因为根据对设计电路速度、面积的不同要求,综合出的电路结构不可能是唯一的。

三、适配(Fitting)

适配是为了将综合器产生的网表文件装配于指定的目标器件中,而产生最终的编程文件。因为在 EDA 工具软件中的综合器可由专业的第三方 EDA 公司提供,而适配器需由目标可编程逻辑器件供应商提供,即由适配器生成的编程文件与目标器件的结构细节相映射。

四、功能仿真与时序仿真(Functional Simulation and Timing Simulation)

仿真是用 EDA 工具软件根据一定的算法和一定的仿真库对 EDA 设计进行模拟,以便验证设计,排除错误。

① 功能仿真是指直接对文本设计、原理图设计及其它设计方式的逻辑功能进行测试模拟,以便了解所设计电路实现的功能是否符合要求。

② 时序仿真的结果接近真实器件的运行特性,因此仿真文件必须是针对具体器件的适配测试结果,而且仿真的耗时也较长。

五、编程下载(Programming)

把适配后生成的编程文件通过编程器或编程电缆载入目标可编程逻辑器件,以便进行硬件验证和调试,进而落实项目设计。

六、硬件测试(Hardware Debugging)

对含有载入了编程文件的可编程逻辑器件的硬件系统进行统一测试,验证设计项目在目标系统上的实际工作情况,以便排除故障,改进设计,最终使设计成功。

1.1.3 硬件描述语言 VHDL

用于描述硬件电路的语言有多种,如 VHDL、Verilog HDL、System Verilog、System C、ABEL HDL、AHDL 等。目前得到众多 EDA 公司支持的语言以 VHDL 和 Verilog HDL 为主。专家预言,在新的世纪中,VHDL 和 Verilog HDL 语言将承担起电子工程领域多数数字系统的设计任务。

VHDL 语言(Very High Speed Integrated Circuit Hardware Description Language,超高速集成电路硬件描述语言)诞生于 1982 年。在美国国防部和 IEEE(The Institute of Electrical and Electronics Engineers,电气与电子工程师学会)的共同努力下,1987 年发布了 VHDL 语言的“IEEE_1076”标准,1993 年又升级为“IEEE_1164”标准。1995 年,中国国家技术监督局组织编辑出版《CAD 通用技术规范》,推荐 VHDL 语言作为我国电子设计自动化硬件描述语言的国家标准。本书将重点介绍 VHDL 的编程方法和使用技术。

VHDL 具有与具体硬件电路无关和与设计开发平台无关的特性,支持自顶向下的设计模式,并且具有良好的硬件电路描述和建模的能力,在语言的易读性和层次化、结构化设计方面具有强大的应用潜力。利用 EDA 工具软件可以将 VHDL 源码程序自动地转换为基本逻辑元件连接图,即设计网表文件,从而大大简化硬件电路的设计任务,提高设计的效率和可靠性。

必须强调的是,VHDL 语言与任何基于 CPU 的软件程序语言的不同之处

在于:基于 CPU 的软件语言依据顺序控制信号,按时钟节拍逐条执行语句,且每执行一条指令需数个时钟周期。而 VHDL 语言适应实际数字电路的工作方式,以并行和顺序语句方式描述在同一时刻电路可能发生的一切事件,具有描述多维的、并发的、组合的、复合的系统电路的能力。因为电路本质上是由许多并行工作的门电路构成,它们之间是通过仅存在传输延迟时间的电信号传递信息。VHDL 语言的这一特征,是任何基于 CPU 的软件程序语言所不能替代的。所以,用 VHDL 语言进行程序设计,清楚 VHDL 语言与硬件结构、电路功能之间的依存关系,尤为重要。

1.2 可编程逻辑器件与 FPGA

可编程逻辑器件 PLD(Programmable Logic Device)是 20 世纪 70 年代发展起来的新型的大规模集成器件。运用 PLD,结合计算机软件技术,可以方便、快捷地构建数字系统电路。

1.2.1 PLD 的工作原理

常见的 PLD 就其集成度而言可分为简单 PLD 和复杂 PLD 两大类。简单 PLD 包括 PROM(Programmable Read Only Memory,可编程只读存储器)、PLA(Programmable Logic Array,可编程逻辑阵列)、PAL(Programmable Array Logic,可编程阵列逻辑)、GAL(Generic Array Logic,通用阵列逻辑)等,它们的集成度较低,每只器件中可用的逻辑门大约在 500 门以下。复杂 PLD 就是现在常用的 CPLD(Complex Programmable Logic Device,复杂可编程逻辑器件)和 FPGA(Field Programmable Gate Array,现场可编程门阵列),它们是 20 世纪 80 年代中期发展起来的高密度芯片,每只器件可含上万门可用的逻辑门。

一、可编程只读存储器 PROM

可编程只读存储器 PROM 由不可编程的与阵列和可编程的或阵列构成,如图 1.2.1 所示。其中与阵列对于 n 个输入变量 $A_0 \sim A_{n-1}$ 输出 2^n 个全部最小项 $W_0 \sim W_{2^n-1}$,或阵列通过编程从中选择所需的最小项将它们或起来,即可构成由变量最小项表述的组合逻辑函数。

表 1.2.1 为 1 位全加器的逻辑真值表,输入信号 A_2 、 A_1 、 A_0 分别为本位加数、本位被加数和从低位来的进位信号,输出信号 F_1 、 F_0 是本位和数与本位送往高位的进位信号,据此可得到 1 位全加器的最小项之和表达式(1.2.1)和(1.2.2)。如果提供输入信号的非信号,可以构建如图 1.2.2 所示的仅由与门和或门实现的 1 位全加器电路。

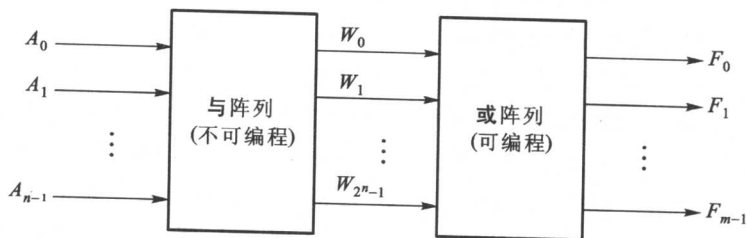


图 1.2.1 PROM 逻辑结构框图

表 1.2.1 1 位全加器真值表

输 入			输 出	
A_2	A_1	A_0	F_1	F_0
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

$$F_0 = \bar{A}_2 A_1 A_0 + A_2 \bar{A}_1 A_0 + A_2 A_1 \bar{A}_0 + A_2 A_1 A_0 \quad (1.2.1)$$

$$F_1 = \bar{A}_2 \bar{A}_1 A_0 + \bar{A}_2 A_1 \bar{A}_0 + A_2 \bar{A}_1 \bar{A}_0 + A_2 A_1 A_0 \quad (1.2.2)$$

在图 1.2.2 中用到了 3 输入与门、4 输入或门和互补的输入信号。为了简明地表示多输入变量的逻辑门, PLD 采用专用的电路符号, 见图 1.2.3。在

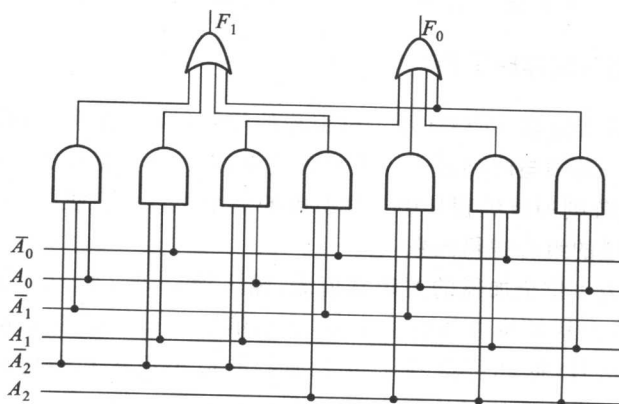


图 1.2.2 用与门和或门实现的 1 位全加器

PLD 的电路符号中,输入信号与门电路的输入线十字相交。若交点上打了黑点,表示固定连接(不可编程);若交点上打了叉,表示已被连接或可编程;否则表示信号与该门电路没有关联关系。

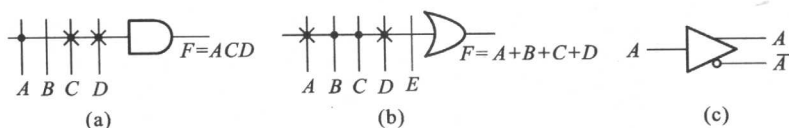


图 1.2.3 PLD 采用的常用电路符号

(a)与门 (b)或门 (c)互补缓冲器

用 PROM 实现的 1 位全加器如图 1.2.4 所示。显然,用 PROM 可实现任何扩展为最小项之和表达式的组合逻辑函数。

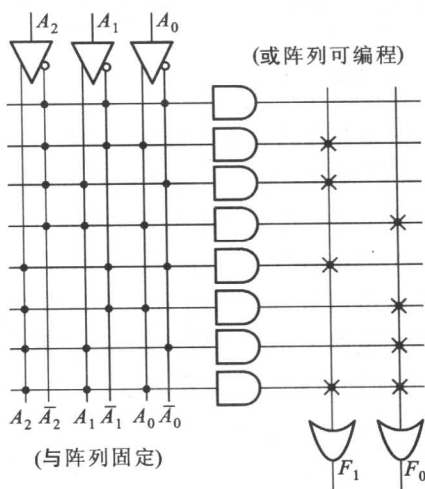


图 1.2.4 用 PROM 实现的 1 位全加器

二、可编程逻辑阵列 PLA

PROM 的与阵列实质上是全译码器,可产生输入变量的全部最小项,然而在实际应用中,绝大多数的组合逻辑函数并不需要所有的最小项。于是,诞生了与阵列和或阵列均可编程的可编程逻辑阵列 PLA。由 PLA 实现的 1 位全加器如图 1.2.5 所示。

在用 PLA 实现组合逻辑时,首先需要把函数化为最简的与或表达式,然后用可编程的与阵列构成与项,用可编程的或阵列完成或运算。在有多输出时,要尽量运用公共的与项,以提高阵列的利用率。

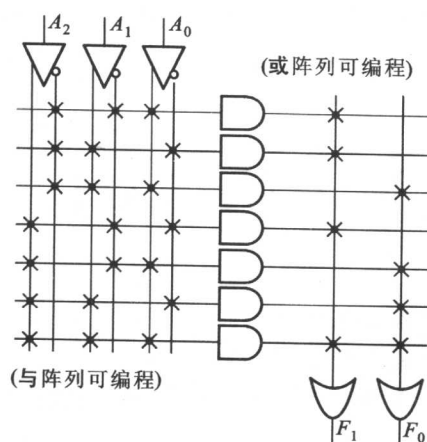


图 1.2.5 用 PLA 实现的 1 位全加器

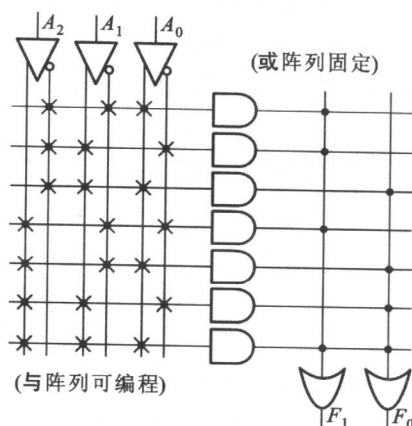


图 1.2.6 用 PAL 实现的 1 位全加器

三、可编程阵列逻辑 PAL

PLA 的利用率高,但是与阵列和或阵列均可编程,必然造成软件算法复杂,影响运行速度。不久,人们研制了或阵列固定仅与阵列可编程的可编程阵列逻辑 PAL。由 PAL 实现的 1 位全加器如图 1.2.6 所示。

图 1.2.6 所示的 PAL 只允许逻辑函数 F_0 和 F_1 有 4 个与项。事实上,对于多个乘积项, PAL 允许输出信号再馈送给另一个与阵列。为了适应不同的需要,生产者在 PAL 的输出电路中还加入各种触发器、异或逻辑、三态逻辑,使得人们能够用 PAL 实现对时序电路的可编程。

四、通用阵列逻辑 GAL

不同型号的 PAL 有着不同的 I/O 结构,给生产者、使用者都带来一些不便。1985 年, Lattice 公司在 PAL 的输出部分增加了输出逻辑宏单元 OLMC (Output Logic Macro Cell),即成功研制了通用阵列逻辑 GAL,如图 1.2.7 所示。通过对结构控制字编程可将 OLMC 设置成不同的工作模式,如专用输入、专用组合输出、反馈组合输出、时序电路中的组合输出、寄存器输出等,从而增强了器件的通用性和灵活性。由于 OLMC 单元具有结构重构的功能,在一定程度上简化了电路板的布局布线,因而使系统的可靠性进一步提高。

五、复杂可编程逻辑器件 CPLD

复杂可编程逻辑器件 CPLD 诞生于 20 世纪 80 年代,是以 GAL 结构为基础扩展而来的超大规模集成芯片,适于设计较大规模的数字逻辑电路。CPLD