

计算机系列教材

可编程逻辑器件 原理与设计

吕高焕 编著

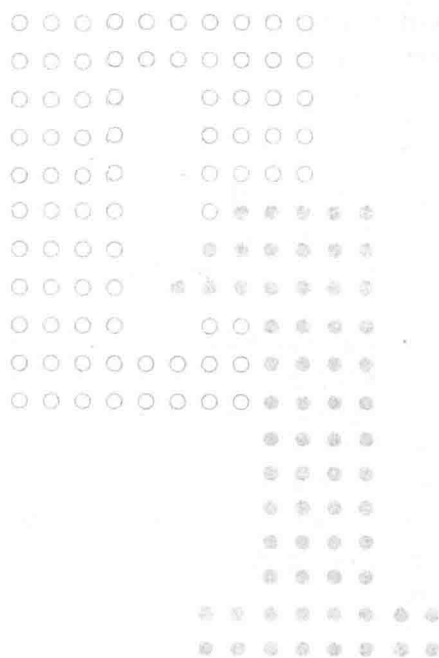
清华大学出版社



计算机系列教材

吕高焕 编著

可编程逻辑器件 原理与设计



清华大学出版社
北京

内 容 提 要

本书由浅入深、循序渐进地介绍可编程逻辑器件的基本原理、内部结构和设计方法,系统地介绍了用于 CPLD/FPGA 开发的 VHDL 语言。对于可编程器件的基本原理,首先从基本逻辑门出发,讲述控制逻辑函数表达式的设计与分解,然后详细介绍 SPLD(包括 PLA 和 PAL)、CPLD 和 FPGA 的组成原理及其区别。对于 VHDL 语言,则先从 VHDL 基本元素、基本语法、描述模型开始,依次讲解并行语句、顺序语句、元件、库和包、有限状态机等,并配有丰富的实例,有助于学习者对概念的理解和用法的掌握。

本书适合于学习芯片设计的理工科学生和 VHDL 初学者,可作为高等学校电子类专业的选修教材或有志于研发数字集成电路芯片的工程技术人员参考书。

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

版权所有,侵权必究。侵权举报电话:010-62782989 13701121933

图书在版编目(CIP)数据

可编程逻辑器件原理与设计/吕高焕编著. —北京:清华大学出版社,2016

(计算机系列教材)

ISBN 978-7-302-45647-6

I. ①可… II. ①吕… III. ①可编程序逻辑器件—高等学校—教材 IV. ①TP332.1

中国版本图书馆 CIP 数据核字(2016)第 270208 号

责任编辑:白立军

封面设计:常雪影

责任校对:李建庄

责任印制:李红英

出版发行:清华大学出版社

网 址: <http://www.tup.com.cn>, <http://www.wqbook.com>

地 址:北京清华大学学研大厦 A 座 邮 编:100084

社 总 机:010-62770175 邮 购:010-62786544

投稿与读者服务:010-62776969, c-service@tup.tsinghua.edu.cn

质量反馈:010-62772015, zhiliang@tup.tsinghua.edu.cn

课件下载: <http://www.tup.com.cn>, 010-62795954

印 装 者:北京国马印刷厂

经 销:全国新华书店

开 本:185mm×260mm

印 张:13.5

字 数:329 千字

版 次:2016 年 10 月第 1 版

印 次:2016 年 10 月第 1 次印刷

印 数:1~2000

定 价:29.00 元

产品编号:071732-01

微电子技术的飞速发展,不仅加快了微处理器的速度,增强了信号处理电路的功能,扩大了系统的存储容量,也推动了可编程逻辑器件(Programmable Logic Device,PLD)在数字电子设计中的广泛应用。借助于常用的集成了各种优化算法的电子设计自动化(Electronic Design Automation,EDA)软件,PLD开发人员可以优化设计电路,在较短的时间内实现用户需求。基于此,PLD原理与电子设计自动化技术成为了电子设计人员进行高效电路设计的基础,在电子类相关专业中开设该课程是必不可少的。

PLD分为简单可编程逻辑器件(Simple PLD,SPLD)、复杂可编程逻辑器件(Complex PLD,CPLD)和现场可编程门阵列(Field Programmable Gate Arrays,FPGA)3类。这3类器件出现时间由先到后,容量由低到高。SPLD最早出现在20世纪70年代,由Phillips公司制造出第一个可编程逻辑阵列开始,其应用市场不断扩大,但可编程开关、逻辑平面设计的复杂性限制了其进一步发展。随着电子制作工艺的不断改进,出现了CPLD,它实际上就是将SPLD通过内部连线组合起来,扩大了芯片容量。SPLD和CPLD都是基于与平面和或平面的。随着设计需求规模的不断扩大,工程师设计出了FPGA,它不再是基于逻辑平面的概念,而是使用可配置逻辑块,不仅扩大了芯片容量,也提高了设计的灵活性。由于CPLD和FPGA的结构不同,因此电路的设计优化算法亦不相同。为了统一两者的开发方式,工程师们研究了与硬件无关的VHDL、Verilog HDL语言。使用VHDL和Verilog HDL语言,开发人员借助于EDA软件,对不同类型的器件采用相应的优化综合算法,方便了电路的设计和开发。本书介绍VHDL语言在PLD中的设计方法。

本书可分两部分:第一部分是从第1章到第4章,讲述PLD的基本原理;第二部分是从第5章到第11章,讲述VHDL语言在电路设计中的基本语法及典型应用。

第1章到第4章,从基本逻辑门出发,介绍了构成逻辑函数表达式的基于CMOS器件的电路结构及设计方法,它是理解SPLD、CPLD、FPGA等复杂电路的基础。对于SPLD,详细介绍了可编程逻辑阵列(Programmable Logic Array,PLA)和可编程阵列逻辑(Programmable Array Logic,PAL)的原理,并以此为基础,介绍了CPLD。对于FPGA,则重点介绍逻辑块和可编程开关的原理,并结合不同厂家的FPGA结构进行介绍。

第5章到第11章,由浅入深、循序渐进地介绍了VHDL语法要素、描述模型、并行语句、顺序语句、元件、包库和有限状态机。在讲解过程中结合常见的基本逻辑电路,强化读

者对概念的理解和设计理念的把握。

CPLD/FPGA 技术发展日新月异,新技术、新工艺、新算法、新软件层出不穷,虽然本书在撰写过程中力求加入最新的资料,但仍有些地方赶不上工艺的发展,因此本书重点讲述原理和基础,使读者在面对新工艺或新技术时能灵活理解其基本原理。在讲述 VHDL 过程中,本书力求基于语法原理介绍电路设计中各模块间“相互独立、各尽其责”思想,启发读者面对大型设计时能够自上而下地合理划分设计模块,采用模块化、层次化思想实现设计需求。

本书在撰写过程中,参阅了大量的国内外网站,以及与本课程相关的教育工作者和技术培训人员的相关幻灯片及技术资料;同教研室的郝金光老师、徐明铭老师、石磊老师提出了很多宝贵的意见;本书底稿讲义在十余年的科研培训、教学中不断得到补充和发展,学员们提出了各种有助于学习和理解的建议;在写作过程中作者得到清华大学出版社白立军编辑的热情帮助。在此一并向他们表示崇高的敬意和深深的感谢。

限于作者水平,书中错误之处在所难免,恳请读者批评指正。

作者

2016 年 7 月

F O R E W O R D

第 1 章 绪论 /1

1.1 可编程逻辑器件与数字电路设计 /1

1.2 可编程逻辑器件的发展 /2

1.3 可编程逻辑器件设计 /7

1.3.1 电子设计自动化 /7

1.3.2 电子设计自动化的发展 /8

1.3.3 EDA 工具的主要特征 /9

1.3.4 有代表性的 EDA 软件 /11

1.3.5 设计方法 /13

1.3.6 设计流程 /14

思考题 /17

第 2 章 数字逻辑 /18

2.1 基本逻辑门及其运算 /18

2.2 基本扩展逻辑门 /19

2.3 逻辑门的扩展 /20

2.4 基本逻辑门的实现 /23

2.4.1 MOS 管 /23

2.4.2 非门的 CMOS 实现 /25

2.4.3 基本与非门的实现 /25

2.4.4 基本或非门的实现 /26

2.4.5 逻辑函数表达式的 CMOS 实现 /26

思考题 /28

第 3 章 可编程逻辑器件原理 /29

3.1 简单可编程逻辑器件 /29

3.1.1 可编程逻辑阵列 /29

3.1.2 可编程阵列逻辑 /30

3.2 复杂可编程逻辑器件 /32

3.2.1 Altera MAX 系列 CPLD /33

- 3.2.2 AMD MACH 系列 CPLD /34
- 3.2.3 Lattice pLSI 和 ispLSI 系列 CPLD /35
- 3.2.4 Xilinx XC 7000 系列 CPLD /36
- 3.2.5 Altera FlashLogic /36
- 3.3 现场可编程逻辑门阵列 /37
 - 3.3.1 逻辑块 /39
 - 3.3.2 可编程开关 /43
 - 3.3.3 典型 FPGA 内部结构 /48
- 3.4 CPLD 和 FPGA 比较 /51
- 思考题 /53

第 4 章 图形和文本输入 /54

- 4.1 Altera Quartus II 9.0 工作环境 /54
 - 4.1.1 基于工程的管理环境 /54
 - 4.1.2 工程设计工具 /55
- 4.2 图形输入法 /56
 - 4.2.1 4-1 选择器 /56
 - 4.2.2 建立工程 /56
 - 4.2.3 电路设计 /60
 - 4.2.4 利用 4-1 选择器设计 8-1 选择器 /66
- 4.3 文本输入法 /69
- 4.4 配置文件下载 /69
- 思考题 /71

第 5 章 VHDL 基础 /72

- 5.1 对象 /72
 - 5.1.1 对象命名规则 /72
 - 5.1.2 对象声明规则 /72
 - 5.1.3 常量 /73
 - 5.1.4 信号 /74
 - 5.1.5 变量 /75
 - 5.1.6 别名 /76

5.2	标准数据类型	/77
5.2.1	bit	/77
5.2.2	bit_vector	/77
5.2.3	boolean	/78
5.2.4	boolean_vector	/78
5.2.5	integer	/78
5.2.6	natural	/79
5.2.7	positive	/79
5.2.8	integer_vector	/79
5.2.9	character	/79
5.2.10	string	/80
5.3	标准逻辑数据类型	/80
5.4	数值表达方法	/82
5.5	数据类型转换	/83
5.6	自定义数据类型	/84
5.6.1	自定义整数类型	/84
5.6.2	枚举类型	/85
5.6.3	子数据类型	/85
5.6.4	数组类型	/85
5.7	预定义属性	/86
5.7.1	标量数据类型的预定义属性	/86
5.7.2	数组类型的预定义属性	/87
5.7.3	信号的预定义属性	/88
5.8	VHDL 中的运算	/88
5.8.1	赋值运算符	/89
5.8.2	逻辑运算符	/89
5.8.3	算术运算符	/90
5.8.4	关系运算符	/90
5.8.5	移位运算	/91
5.8.6	合并运算符	/91
5.8.7	运算符的优先级	/92
	思考题	/92

第 6 章 VHDL 语言的程序结构 /93

6.1 VHDL 设计模型 /93

6.1.1 数据流模型 /93

6.1.2 行为模型 /93

6.1.3 结构化模型 /94

6.2 VHDL 程序结构 /94

6.2.1 实体 /95

6.2.2 架构 /97

6.2.3 库和包 /98

6.2.4 配置 /100

6.3 简单的例子 /100

思考题 /104

第 7 章 并行语句 /105

7.1 简单信号赋值语句 /105

7.2 条件信号赋值语句 /110

7.3 选择信号赋值语句 /114

7.4 产生语句 /118

7.5 块语句 /121

7.6 多驱动源赋值问题 /123

思考题 /124

第 8 章 顺序语句 /125

8.1 锁存器和触发器 /125

8.2 进程 /127

8.3 IF 语句 /128

8.3.1 IF...THEN...END IF /128

8.3.2 IF...THEN...ELSE...END IF /129

8.3.3 IF...THEN...ELSIF...THEN... END IF /129

8.3.4 IF...THEN...ELSIF...THEN... ELSE...END IF /130

8.3.5	嵌套式 IF 语句	/133
8.4	CASE 语句	/138
8.5	WAIT 语句	/141
8.6	LOOP 语句	/143
8.6.1	无条件循环	/143
8.6.2	FOR...LOOP 循环	/143
8.6.3	WHILE...LOOP 循环	/146
8.6.4	LOOP...EXIT 循环	/146
8.6.5	LOOP...NEXT 循环	/147
8.7	寄存器的引入问题	/147
8.8	信号和变量的再讨论	/148
	思考题	/155
第 9 章 元件 /156		
9.1	元件的声明	/156
9.2	元件例化	/157
9.3	元件声明和例化方法	/157
	思考题	/172
第 10 章 库、包与子函数 /173		
10.1	库	/173
10.2	包	/174
10.3	子程序	/177
10.3.1	函数	/178
10.3.2	过程	/181
10.4	过程、函数和进程讨论	/185
10.4.1	子程序与进程	/185
10.4.2	函数与过程	/186
	思考题	/186
第 11 章 有限状态机 /187		
11.1	FSM 的系统图和状态图	/187

11.2	FSM 的编程框架	/188
11.3	Moore 型 FSM 设计	/189
11.3.1	系统图设计	/189
11.3.2	状态机描述	/189
11.3.3	编程实现	/190
11.4	Mealy 型 FSM 设计	/191
11.5	综合设计	/193
11.6	FSM 中的问题	/200
	思考题	/201

附录 A	VHDL 中的保留字	/202
------	------------	------

附录 B	缩略语	/203
------	-----	------

参考文献	/204
------	------

第 1 章 绪 论

可编程逻辑器件(Programmable Logic Device,PLD)的出现极大地改变了电子工程师的电路设计方式和理念,它同电子设计自动化(Electronic Design Automation,EDA)软件一起,改善了设计过程,缩短了产品的开发周期,提高了设计质量。本章将介绍 PLD 的历史及发展概况,PLD 的设计方法和设计工具、设计流程等。

1.1 可编程逻辑器件与数字电路设计

在早期的电路设计中,使用分立元件搭建各种逻辑门,不仅费时、费力,而且调试时非常麻烦,生产周期长、投入成本高、实现的逻辑功能也相对简单,而且占据面积大,在实际应用中很不方便。

随着集成电路工艺的发展,出现了小规模集成电路。小规模集成电路的发展大幅降低了基于分立元件的设计过程和设计成本。通常采用已有的集成电路芯片,如 74H00、74LS32、74HC14 等根据电路图直接组成控制逻辑电路,然后在此基础上生产印制板,最后将相关逻辑器件焊接到印制板上,通过调试、修改,直到定型,实现所需要的功能。但由于其规模小,门数有限,只能实现简单的控制逻辑,而且执行速度慢、耗能大。另外,由于设计中需要大面积的印制板,虽然可通过简单的电路仿真验证电路的功能,但在调试过程中仍然需要投入大量的人力和物力。

大规模集成电路的出现使电路设计过程出现了质的飞跃,它改变了人们以往的电路设计理念。通常大规模集成电路的设计分两类,一类是以单片机和数字信号处理器为代表的微处理器设计,它通常是通过编码的形式使微处理器完成一定的控制和计算功能。该方法通过微处理器辅以外围电路实现对外围信号的处理。微处理器通过逐句执行软件指令实现,其专长是处理非常复杂的数学运算任务。这种系统是通过软件控制实现的,其特点是易编程(通常使用标准 C),有固定的数据位宽,例如 24 位加法器,但这种情况下对 5 位加法实现效率不高;另外芯片资源受限,不易扩展。另一类是以现场可编程门阵列(Field Programmable Gate Arrays,FPGA)为代表的 PLD 系统设计,它通常通过编程、综合后实现控制逻辑电路,包括组合逻辑和时序逻辑。该方法直接将逻辑函数以电路的形式在 PLD 中实现,执行速度快,耗能少。它有大量的引脚和逻辑门容量,可以用来实现大部分的数字电路。这些器件都需要硬件描述语言编程实现控制逻辑功能。PLD 有大量的门和资源,但不支持浮点运算,需要自定义实现。早期它在计算应用方面有一定的局限性,但目前这些器件都可以内嵌 IP 核和嵌入式微处理器,弥补了计算功能上的不足,是当前数字电路设计的主流。

复杂可编程逻辑器件(Complex PLD,CPLD)的发展极大地改变了数字器件的设计过程。以往的设计是板级设计,使用各种包含基本逻辑门的简单芯片组成,而现在的数字

产品都使用高度密集的集成电路。目前的集成电路技术,不仅可以完成自定义配置处理器、存储器等,还可以生成逻辑电路,诸如状态机控制器、计数器、寄存器、解码器和编码器等。当这些器件需要组成更大的系统时,设计人员可将它们集成入更高密度的门阵列。但初期的门阵列生产周期长、费用大,并不适合生产原型机和其他低成本产品,因此人们研究了现场可编程逻辑器件(Field Programmable Logic Device,FPLD)。FPLD 的最具竞争力的优势是低成本、低商业风险。由于器件的编程是现场的,因此它有较短的生产制造周期,且易于在线更改设计,及时发现设计缺陷,改进设计。

在过去的几十年里 FPLD 的市场得到迅猛发展。很多芯片供应商可提供多种不同的器件系列供设计人员选择。为选择一种产品,设计人员需了解设计的规模、器件的性质等,以充分利用器件,达到最优性价比。另外,还需要学习相关的编程语言和编程软件。为此,本书将对 PLD 的原理(电路基础)、PLD 的设计工具(EDA 技术)、PLD 的设计方法(图形化和 VHDL 编程)等从理论和工程上进行系统的介绍。

1.2 可编程逻辑器件的发展

PLD 市场有低容量和高容量器件两部分。低容量器件称为简单可编程逻辑器件(Simple PLD,SPLD),通常包含低于 600 个可用门,系列产品包括可编程逻辑阵列(Programmable Logic Arrays,PLA)、可编程阵列逻辑(Programmable Array Logic,PAL)、通用阵列逻辑(General Array Logic,GAL)等。SPLD 的工艺为 CMOS 工艺,提供 EPROM、EEPROM 和 Flash 存储单元。通常将门数高于 600 的 PLD 称为高容量器件,包含 CPLD 和 FPGA。高容量器件使用 CMOS 工艺,可选 EPROM、EEPROM、Flash、SRAM 和反熔丝技术。高容量器件可通过内连结构区分,CPLD 使用连续内连结构,而 FPGA 使用分段内连结构。图 1.1 展示了 PLD 的分类图。

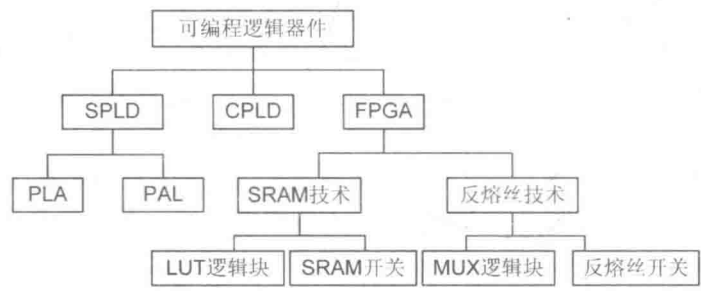


图 1.1 PLD 的分类图

SPLD 是早期的 PLD,其设计框架如图 1.2 所示。这类 PLD 的共同点是使用 AND 平面和 OR 平面两级实现。这两级平面至少有一级是可编程的。

第一个可实现自定义逻辑电路的用户可编程芯片是 可编程只读存储器 (Programmable Read-Only Memory,PROM),其中,地址线作为逻辑输入,数据线作为输出,逻辑函数仅需要几个乘积项,如图 1.3 所示。PROM 包含一个译码器作为地址输入,在 AND 平面上是全编码的,它对实现逻辑电路效率不高,因此很少有设计人员用它来实

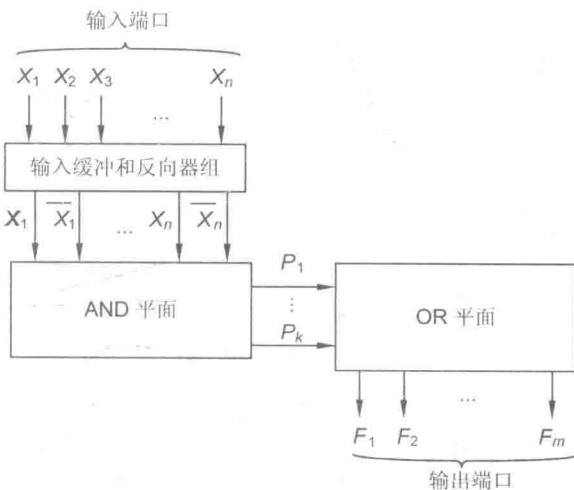


图 1.2 SPLD 框架图

现逻辑功能。

第一种真正意义上的可编程逻辑器件是 PLA。PLA 包含两级逻辑门：一级是可编程的 AND 平面；另一级是可编程的 OR 平面。PLA 的这种结构可使任意输入(或它们的非逻辑)在 AND 平面以乘积项的形式输出,如图 1.4 所示。通过配置 OR 平面,产生任意的控制逻辑函数。这种结构下,PLA 比较适合逻辑函数表达为乘积和的形式的电路实现。

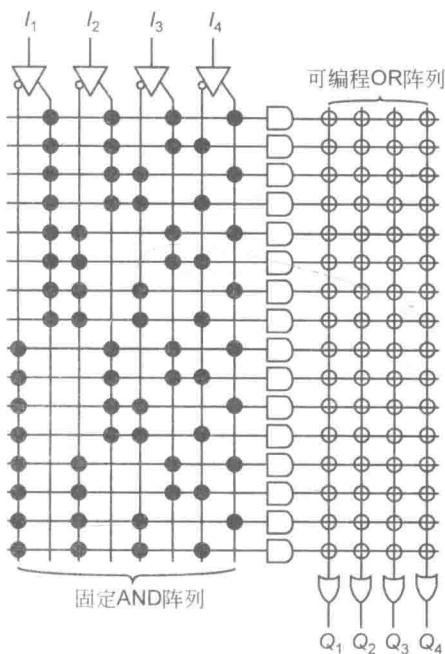


图 1.3 PROM 内部结构图

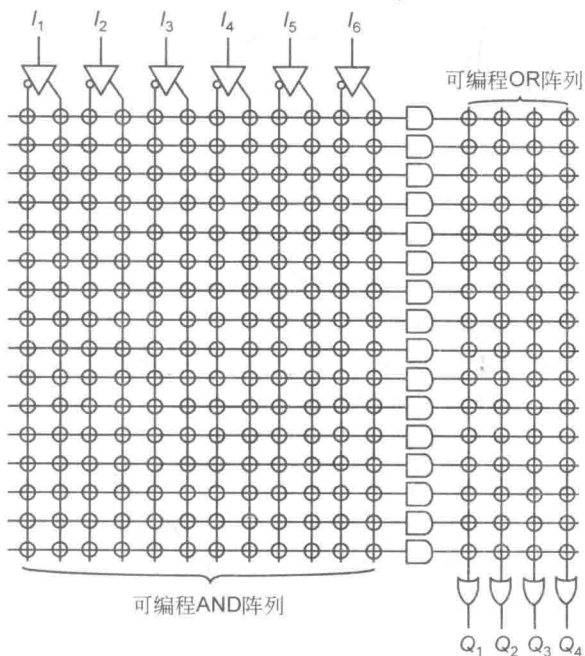


图 1.4 PLA 内部结构图

在 Phillips 公司于 20 世纪 70 年代生产出 PLA 之后,其主要局限性在于其造价高昂,速度性能不佳。这两个缺点来源于其制作工艺和逻辑函数实现的两级方案。由于可编程逻辑开关制作起来工艺复杂,因此可编程逻辑平面制作起来十分困难,而且伴有显著的传输延时。为克服这些弱点,Monolithic Memories 公司开发了 PAL 器件。在该类器件的结构中,仅在 AND 平面有一级可编程功能,而在 OR 平面则是固定的,如图 1.5 所示。PAL 器件的引入对数字硬件设计影响深远,它是构成复杂电路结构的基础。PLA、PAL 都归为简单可编程逻辑器件(SPLD),其特点是成本低、规模小(约 200 个等效门)、端口间延时相对较小等。

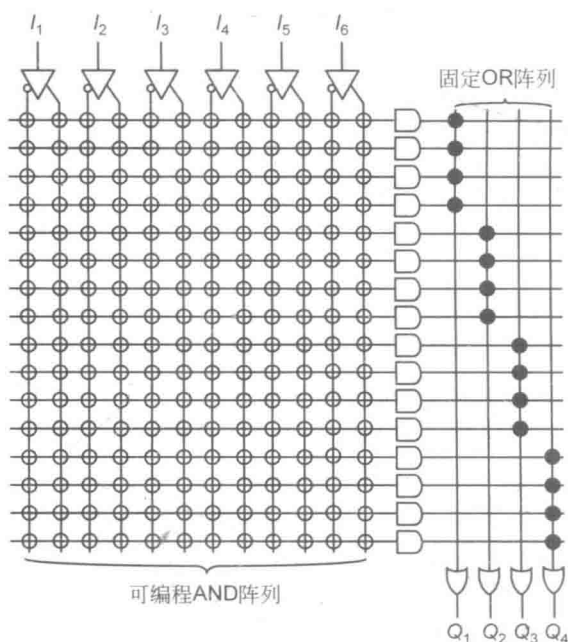


图 1.5 PAL 的内部结构

随着工艺和技术的发展,越来越多的公司可生产集成度更高、容量更大的器件。基于 SPLD 结构的一种可行扩展方法是将多个 SPLD 通过可编程内部连线组合起来,形成一个更大容量的集成电路,称为 CPLD。CPLD 元件基本上是由 SPLD 块组合而成的。一般来说,CPLD 的规模一般在 50 个典型的 SPLD 之上,等效门数在 1000~7000 之间。图 1.6 展示了典型的 CPLD 结构。

Altera 公司在 CPLD 领域做了开拓性的工作,其首创的代表产品为 EPLD 芯片系列,包括 MAX 5000、MAX 7000、MAX 9000 系列以及 FlashLogic 系列。由于大规模可编程逻辑器件日益增大的市场因素,其他制造商也开发 CPLD 器件,其中有些器件在市场上可以购买得到,如 Cypress 的 Max 340 及 Flash 370 系列,AMD 公司的 MACH 系列,Xilinx 公司的 XC 7000 系列、Lattice 公司的 ispLSI 3000 系列等。

CPLD 使用几种不同的 CMOS 工艺和结构以解决不同的逻辑设计问题。基于 EPROM、EEPROM、Flash 的器件,例如 Altera Classic MAX 5000、MAX 7000、MAX 9000 以及 FlashLogic 系列产品,使用乘积项结构适于做组合逻辑设计,这类器件是可编

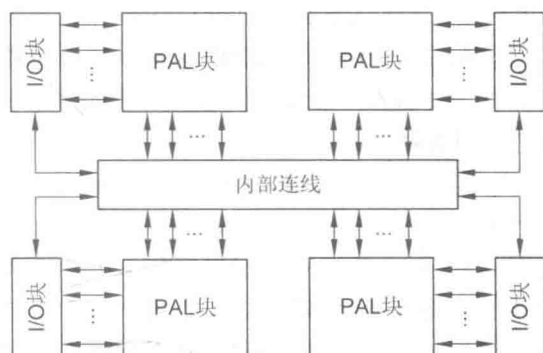


图 1.6 典型的 CPLD 结构

程的且非易失性的。基于 SRAM 的 CPLD,例如 Altera FLEX8000 系列,使用查找表结构,适于做时序逻辑电路设计,这类器件支持在线重配置功能。

CPLD 在设计需要使用大量的 AND/OR 门且不需要大量的触发器电路时有很大优势。通常用于图形控制器、LAN 控制器、UART 和 Cache 控制等。其优点是易于在系统编程、电路实现可预期、速度快。

为提高设计质量,改善电路性能,提高系统集成度,工程师们又研究了新型的芯片结构设计方法,可编程门阵列是其中最有代表性的技术之一。早期出现的可编程门阵列逻辑芯片是掩膜可编程门阵列(Mask Programmable Gate Arrays,MPGA)。一个 MPGA 由预制晶体管阵列组成,这些阵列通过内部连线实现用户逻辑电路。但这种器件造时长、成品周期长、花费高。随后出现的是 FPGA,它是由一组电路单元(逻辑块)和内部联系组成,其最终逻辑功能是通过配置逻辑块编程获得。图 1.7 展示了一个典型的 FPGA 结构。

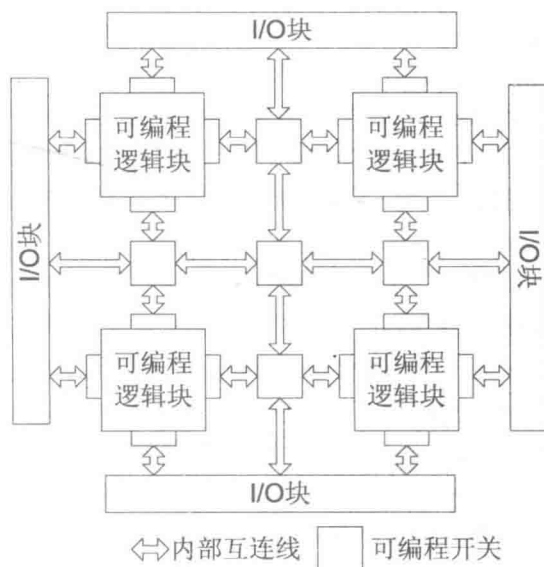


图 1.7 典型的 FPGA 内部结构

该结构使用预制硅器件,通过电气编程可以变成任意种类的数字电路或数字系统。它含有非常大的可编程逻辑块阵列,被可编程互连线包围,方便放置和布线。FPGA 使用逻辑块代替 AND 平面和 OR 平面,可以有几百万门的容量,时钟频率可达 500MHz,可通过终端用户编程实现特定的功能。

第一颗 FPGA 是 1967 年 Wahlstrom 设计的基于 SRAM 的 FPGA。1984 年 Xilinx 设计了具有现代意义的 FPGA,有 64 个逻辑块,58 个输入输出引脚。当前的 FPGA 生产商主要有 4 个: Altera、Xilinx、Actel、Lattice。当前使用的 FPGA 通常超过 30 万个逻辑块,约 100 万个等效门,1100 个输入输出引脚。

FPGA 是目前市场上应用量最大的半导体器件,其种类分为 SRAM 和反熔丝两大类。前者生产商以 Altera、Xilinx 为代表,后者生产商以 Actel、QuickLogic 和 Cypress 为代表。

Altera FPGA 分为两大类:一种侧重低成本应用,其特点是容量中等,性能可以满足一般的逻辑设计要求,如 Cyclone、Cyclone II;还有一种侧重于高性能应用,其特点是容量大,性能能满足各类高端应用,如 Stratix、Stratix II 等,用户可以根据自己实际应用要求进行选择。在性能可以满足的情况下,优先选择低成本器件。

Altera FPGA 的最基本的资源是逻辑单元(Logic Element, LE),一个 LE 主要包括一个四输入查找表(Look-Up Table, LUT),从数据存储角度看, LUT 本质上就是一个随机存储器(Random Access Memory, RAM)。目前 Cyclone 系列使用四输入的 LUT,所以每一个 LUT 可以看成是一个有 4 位地址线的 16×1 的 RAM。当用户用程序描述了一个逻辑电路以后, FPGA 开发工具会自动计算逻辑电路的所有可能的结果,并把结果事先写入 RAM,这样,每输入一个信号进行逻辑运算就等于输入一个地址进行查表,找出地址对应的内容,然后输出即可。除了一个四输入查找表以外,还有一个可编程寄存器,可以通过程序来配置为异步或者同步触发器。通常看 Altera 的 FPGA 资源大小,就是看芯片中有多少个 LE。

例如, Altera Cyclone 系列芯片里面 16 个 LE 组成一个逻辑阵列块(Logic Array Block, LAB),而每个 LAB 都有 LAB 控制信号,按照顺序排列在 FPGA 内部,每个 LAB 内部之间的 LE 数据通信通过寄存器链传输,每个 LAB 之间的通过纵横两个通道互连,相互连接的 LAB、LAB 周围的 RAM、乘法器、PLL 都可以通过纵横线驱动其周围的 LAB。

FPGA 的输入输出端口(I/O)资源相当丰富,它可以根据参考电平(REF)不同接收或输出不同的电平标准的信号,如 3.3V LVTTTL、2.5V LVDS、LVPECL 或者 SSTL-2 等。Cyclone III 以及 Cyclone IV 还支持片内上拉电阻、输入输出延时等。Cyclone IV GX 系列还带有高速收发器、PCIe 硬核等。

FPGA 产品的应用领域已经从原来的通信扩展到消费电子、汽车电子、工业控制、测试测量等广泛的领域,而应用的变化也使 FPGA 产品近几年的演进趋势越来越明显:一方面, FPGA 供应商致力于采用当前最先进的工艺来提升产品的性能,降低产品的成本;另一方面,越来越多的通用 IP(知识产权)或客户定制 IP 被引入 FPGA 中,以满足客户产品快速上市的要求。此外, FPGA 企业都在大力降低产品的功耗,满足业界越来越苛刻的