

高职高专系列教材

FPGA/CPLD

技术实用教程

邹益民 主编

中国石化出版社

[HTTP://WWW.SINOPEC-PRESS.COM](http://www.sinopec-press.com)

高职高专系列教材

FPGA/CPLD 技术实用教程

邹益民 主编

中国石化出版社

内 容 提 要

本书从实际应用角度出发,以 Altera 公司的 FPGA/CPLD 芯片及相关 EDA 软件为载体,由浅入深地介绍了基于 FPGA/CPLD 器件的应用系统软硬件设计相关知识与工程技巧;Quartus II 开发工具软件的使用方法与开发技术;VHDL 语言的语法结构和编程技巧以及常用组合与时序逻辑电路的设计方法。本书同时以康芯 GW48 EDA 实验开发系统为例,简单介绍了应用开发的步骤与技巧。

本书结构清晰、内容全面并且重点突出,基础知识与大量实例相结合,突出实用性和可操作性,略去了部分抽象冷僻的内容,重点放在基本概念和常用方法的讲解上。本书可作为高职院校电子信息、机电、通信、自动化、计算机等专业的教材或教学参考书使用,也可作为从事各类电子系统设计的广大工程技术人员的培训教材或实用工具书。

图书在版编目 (CIP) 数据

FPGA/CPLD 技术实用教程 / 邹益民主编. —北京: 中国石化出版社, 2009
(高职高专系列教材)
ISBN 978 - 7 - 5114 - 0115 - 1

I. F… II. 邹… III. 可编程序逻辑器件 - 系统设计 - 高等学校: 技术学校 - 教材 IV. TP332.1

中国版本图书馆 CIP 数据核字 (2009) 第 189292 号

未经本社书面授权, 本书任何部分不得被复制、抄袭, 或者以任何形式或任何方式传播。版权所有, 侵权必究。

中国石化出版社出版发行

地址: 北京市东城区安定门外大街 58 号

邮编: 100011 电话: (010) 84271850

读者服务部电话: (010) 84289974

<http://www.sinopec-press.com>

E-mail: press@sinopec.com.cn

北京科信印刷厂印刷

全国各地新华书店经销

*

787 × 1092 毫米 16 开本 11 印张 262 千字

2010 年 1 月第 1 版 2010 年 1 月第 1 次印刷

定价: 25.00 元

前 言

随着对电路功能及性能要求的不断提升,传统的简单集成电路已不能满足设计者的需求,可编程器件逐渐成为广大硬件工程师所必需的设计器件,尤其适合于新产品的开发与小批量生产,因此深受广大工程技术人员的喜爱。

FPGA/CPLD、DSP 和 CPU 被称为未来数字电路系统的三块基石,也是目前硬件设计研究的热点。与传统电路设计方法相比,FPGA/CPLD 具有功能强大,开发过程投资小、周期短,可反复编程修改,保密性能好,开发工具智能化等特点,特别是随着电子工艺的不断改进,低成本 FPGA/CPLD 器件推陈出新,这一切促使 FPGA/CPLD 成为当今硬件设计的首选方式之一,被广泛应用于通信、仪器仪表、工业控制、信息处理等系统的设计与生产中。而掌握可编程逻辑器件 FPGA/CPLD 的设计技术,则成为电子信息类专业技术人员的一项必备设计手段和技能。

Altera 公司作为世界上最大的可编程逻辑器件供应商之一,不断地引领着世界上可编程器件领域的发展方向。其主要产品从早期的 MAX 系列、FLEX 系列、ACEX 系列到现在流行的 MAX II 系列、Stratix 系列、Stratix II 系列、Cyclone 系列、Cyclone II 系列等都有经典之作,非常有利于我们深入了解可编程器件。而作为 Altera 公司新一代开发软件的 Quartus II 更是具备了强大的功能,使我们能够非常方便地开发可编程硬件系统。故本书以 Altera 公司 FPGA/CPLD 产品及 Quartus II 开发平台作为载体,全面介绍可编程器件原理、设计、应用等基本知识及技能。

本书主要内容有:第 1 章首先给出可编程逻辑器件的基本概念、结构特点,并对其常用的开发工具、开发流程加以说明;第 2 章介绍了 VHDL 语言的基本概念、语法规则及功能特点,并通过大量实例展示其应用技巧;第 3 章以 Altera 的 Quartus II 综合开发平台为例,介绍典型 EDA 开发软件的基本功能、用户界面、设计流程;第 4~5 章介绍了典型的组合逻辑及时序逻辑电路设计的基本方法及技巧;第 6 章简单介绍了 GW48 EDA 实验系统的基本特点及使用方法,并结合实例给出了 FPGA/CPLD 硬软件应用与开发的基本知识及技能。

本书以掌握 FPGA/CPLD 及 EDA 应用技术为教学目标,以培养学生的设计 and 应用开发能力为主线,突出实用性、可操作性。从基础、应用、实践三个角度,系统地介绍相关知识与技巧。全书在取材和编排上循序渐进,注重理论联系实际。以应用为主线“以例代理”,由浅入深地提供大量经过验证的设计范例,方便自学。使学生掌握每种典型实例的设计思路和语法,进而使学生能够自行设计出比较复杂的电路。方便读者快速、全面地掌握 FPGA/CPLD 设计技

术。可作为高职院校电子信息等专业的教材或教学参考书，也可作为电子系统设计工程技术人员的培训教材或实用工具书。还可作为院校科技创新实践、电子产品制作、电子信息类课程设计和毕业设计等实践活动的指导书。

本书由邹益民主编，贾达、汪霞、杜韦辰、马芙蓉等参加编写。

在编写本书的过程中参考了相关文献，在此向这些文献的作者深表感谢！由于EDA技术是一门发展迅速的新技术，加上作者水平有限，书中出现的疏漏及不妥之处，恳请读者和各位同仁批评指正。

目 录

第1章 可编程逻辑器件简介	(1)
1.1 可编程逻辑设计技术简介	(1)
1.1.1 可编程逻辑器件发展简史	(1)
1.1.2 可编程逻辑器件特点	(2)
1.1.3 可编程逻辑器件分类	(2)
1.1.4 主要可编程逻辑器件生产厂商及典型器件	(3)
1.2 FPGA/CPLD 的基本结构	(4)
1.2.1 CPLD 的基本结构	(4)
1.2.2 FPGA 的基本结构	(6)
1.2.3 CPLD 和 FPGA 的比较及选用	(8)
1.3 FPGA/CPLD 的设计流程	(9)
1.4 FPGA/CPLD 的开发工具 - EDA 软件	(12)
1.4.1 设计输入工具	(13)
1.4.2 综合工具	(14)
1.4.3 仿真工具	(14)
1.4.4 实现与优化工具	(14)
1.4.5 后端辅助工具	(15)
1.4.6 验证调试工具	(15)
1.4.7 系统级设计环境	(15)
1.5 下一代可编程逻辑设计技术展望	(15)
1.5.1 下一代可编程逻辑器件硬件发展趋势	(15)
1.5.2 下一代 EDA 软件设计方法发展趋势	(17)
1.6 Altera 典型 FPGA/CPLD 的结构	(19)
1.6.1 Altera 高密度 FPGA	(19)
1.6.2 Altera 低成本 FPGA	(21)
1.6.3 Altera 的 CPLD 器件	(22)
1.7 本章小结	(25)
1.8 习题	(25)
第2章 VHDL 硬件描述语言	(26)
2.1 VHDL 概述	(26)
2.1.1 VHDL 语言的特点	(26)
2.1.2 VHDL 程序的一般结构	(27)
2.2 VHDL 语言的程序结构	(28)
2.2.1 实体(ENTITY)	(28)
2.2.2 结构体(ARCHITECTURE)	(29)

2.2.3	库(LIBRARY)	(35)
2.2.4	程序包(PACKAGE)	(36)
2.2.5	配置(CONFIGURATION)	(37)
2.3	VHDL 语言的数据类型	(38)
2.3.1	VHDL 的文字规则	(38)
2.3.2	VHDL 的数据对象(Data Object)	(40)
2.3.3	VHDL 的数据类型(Data Type)	(41)
2.3.4	VHDL 的类型转换	(45)
2.4	VHDL 的操作符(Operator)	(46)
2.4.1	操作符的种类	(46)
2.4.2	操作符的优先级	(47)
2.4.3	逻辑操作符(Logical Operator)	(47)
2.4.4	关系操作符(Relational Operator)	(47)
2.4.5	算术操作符(Arithmetic Operator)	(47)
2.5	VHDL 的顺序语句(Sequential Statement)	(48)
2.5.1	赋值语句	(49)
2.5.2	流程控制语句	(50)
2.5.3	等待语句(WAIT)	(54)
2.5.4	子程序调用语句	(55)
2.5.5	返回语句(RETURN)	(56)
2.5.6	空操作语句(NULL)	(57)
2.5.7	其他顺序语句	(57)
2.6	VHDL 的并行语句(Concurrent Statement)	(60)
2.6.1	块语句(Block Statement)	(60)
2.6.2	进程语句(Process Statement)	(62)
2.6.3	并行过程调用语句(Concurrent Procedure Call)	(64)
2.6.4	并行信号赋值语句(Concurrent Signal Assignment)	(65)
2.6.5	元件例化语句(Component Instantiation)	(68)
2.6.6	生成语句(Generate Statement)	(71)
2.7	本章小结	(73)
2.8	习题	(74)
第3章	Quartus II 软件及应用	(76)
3.1	Quartus II 软件概述	(76)
3.1.1	Quartus II 软件的功能简介	(76)
3.1.2	Quartus II 软件的用户界面	(77)
3.2	基本设计流程	(78)
3.2.1	工程文件管理	(79)
3.2.2	创建 Quartus II 的工程	(79)
3.2.3	设计输入方式	(79)
3.2.4	基于图形编辑输入法的设计过程	(80)
3.2.5	基于文本编辑输入法的设计过程	(90)

3.3	SignalProbe 及 SignalTap II 逻辑分析器	(91)
3.3.1	SignalProbe	(91)
3.3.2	SignalTap II 逻辑分析器	(93)
3.4	本章小结	(97)
3.5	习题	(97)
第4章	组合逻辑电路设计	(99)
4.1	基本门电路的设计	(99)
4.1.1	设计要求	(99)
4.1.2	VHDL 语言输入	(99)
4.1.3	软件仿真	(100)
4.2	数据选择器的设计	(100)
4.2.1	设计要求	(100)
4.2.2	VHDL 语言输入	(100)
4.2.3	软件仿真	(101)
4.3	1 对 2 数据分配器的设计	(101)
4.3.1	设计要求	(101)
4.3.2	VHDL 语言输入	(102)
4.3.3	软件仿真	(102)
4.4	4 位 BCD 译码器的设计	(102)
4.4.1	实验原理	(103)
4.4.2	VHDL 语言输入	(103)
4.4.3	软件仿真	(104)
4.5	三态门的设计	(104)
4.5.1	设计要求	(104)
4.5.2	VHDL 语言输入	(104)
4.5.3	软件仿真	(105)
4.6	半加器的设计	(105)
4.6.1	设计要求	(105)
4.6.2	VHDL 语言输入	(105)
4.6.3	软件仿真	(106)
4.7	全加器的设计	(106)
4.7.1	设计要求	(106)
4.7.2	VHDL 语言输入	(106)
4.7.3	软件仿真	(107)
4.8	6 位加法器的设计	(107)
4.8.1	设计要求	(107)
4.8.2	VHDL 语言输入	(107)
4.8.3	软件仿真	(109)
4.9	4 位加减法器的设计	(109)
4.9.1	设计要求	(109)
4.9.2	VHDL 语言输入	(109)

4.9.3 软件仿真	(110)
4.10 3 位乘法器的设计	(110)
4.10.1 设计要求	(110)
4.10.2 VHDL 语言输入	(110)
4.10.3 软件仿真	(111)
4.11 本章小结	(111)
4.12 习题	(111)
第5章 时序逻辑电路设计	(113)
5.1 VHDL 程序中时钟信号变化的检测处理	(113)
5.1.1 使用进程的敏感信号表检测时钟变化	(113)
5.1.2 使用 WAIT 语句检测时钟变化	(113)
5.2 D 触发器的设计	(114)
5.2.1 设计要求	(114)
5.2.2 VHDL 语言输入	(114)
5.2.3 软件仿真	(115)
5.3 JK 触发器的设计	(116)
5.3.1 设计要求	(117)
5.3.2 VHDL 语言输入	(117)
5.3.3 软件仿真	(118)
5.4 带异步复位/置位端的同步使能 T 触发器的设计	(119)
5.4.1 设计要求	(120)
5.4.2 VHDL 语言输入	(120)
5.4.3 软件仿真	(120)
5.5 简单计数器的设计	(121)
5.5.1 设计要求	(121)
5.5.2 VHDL 语言输入	(121)
5.5.3 软件仿真	(121)
5.5.4 功能拓展: 同步清零的计数器	(122)
5.6 同步清零的可逆计数器	(123)
5.6.1 设计要求	(123)
5.6.2 VHDL 语言输入	(123)
5.6.3 软件仿真	(124)
5.7 同步预置数的计数器	(124)
5.7.1 设计要求	(124)
5.7.2 VHDL 设计输入	(124)
5.7.3 软件仿真	(125)
5.7.4 功能拓展	(125)
5.8 带进制的计数器	(128)
5.8.1 设计要求	(128)
5.8.2 VHDL 设计输入	(128)
5.8.3 软件仿真	(129)

5.9	基本寄存器的设计	(129)
5.9.1	设计要求	(129)
5.9.2	VHDL 语言输入	(129)
5.9.3	软件仿真	(130)
5.10	基本移位寄存器的设计	(130)
5.10.1	设计要求	(130)
5.10.2	VHDL 设计输入	(130)
5.10.3	软件仿真	(132)
5.11	同步预置数串行输出移位寄存器	(132)
5.11.1	设计要求	(133)
5.11.2	VHDL 设计输入	(133)
5.11.3	软件仿真	(133)
5.12	循环移位寄存器的设计	(134)
5.12.1	设计要求	(134)
5.12.2	VHDL 设计输入	(134)
5.12.3	软件仿真	(134)
5.13	6 位双向移位寄存器的设计	(135)
5.13.1	设计要求	(135)
5.13.2	VHDL 设计输入	(135)
5.13.3	软件仿真	(136)
5.14	有限状态机的设计	(137)
5.14.1	莫尔型状态机	(137)
5.14.2	设计要求	(137)
5.14.3	设计输入	(137)
5.14.4	软件仿真	(138)
5.14.5	米里型状态机	(139)
5.14.6	设计要求	(139)
5.14.7	设计输入	(139)
5.15	本章小结	(141)
5.16	习题	(142)
第 6 章	GW48 EDA 实验系统使用方法简介	(145)
6.1	GW48 系统简介	(145)
6.1.1	基本特点	(145)
6.1.2	系统基本结构及主要模块功能	(145)
6.2	实验电路结构图	(147)
6.2.1	实验电路信号资源符号图说明	(147)
6.2.2	各实验电路结构图特点及应用	(148)
6.2.3	系统插口信号使用说明	(155)
6.3	基于 GW48 的实验示例	(160)
6.4	本章小结	(164)
6.5	习题	(164)

第 1 章 可编程逻辑器件简介

本章首先给出可编程逻辑器件的基本概念,简单介绍现代可编程逻辑器件的发展过程、结构特点及分类,并对其常用开发工具、开发流程加以说明,同时简要介绍了目前市场上常见的可编程器件及其供应商,最后以 Altera 公司典型器件为例,进一步说明可编程逻辑器件的详细结构及特点。

1.1 可编程逻辑设计技术简介

1.1.1 可编程逻辑器件发展简史

随着微电子设计技术与工艺的发展,数字集成电路由电子管、晶体管、中小规模集成电路、超大规模集成电路逐步发展到专用集成电路(ASIC, Application Specialized Integrated Circle)。ASIC 是指应特定用户要求和特定电子系统的需要而设计、制造的集成电路,其特点是面向特定用户的需求,品种多、批量少,要求设计和生产周期短。作为集成电路技术与特定用户需求紧密结合的产物,ASIC 与通用集成电路相比具有体积更小、功耗更低、性能更高、保密性更强等优点。然而,ASIC 器件也存在着改版投资大、灵活性差等缺陷,制约了它的应用范围。

ASIC 的设计主要有全定制(full custom)设计方法和半定制(semi - custom)设计方法。全定制方法是完全由设计师根据工艺要求,以尽可能高的速度、尽可能小的面积以及完全匹配的封装,独立地进行芯片设计。这种方法虽然针对性强,有望达到最优的设计性能,但却需要花费大量的时间与人力来进行专业化设计,而且一旦需要修改内部设计,则将造成重大损失,故其设计成本相对较高,适合于大批量的 ASIC 芯片设计。相比之下,半定制方法是一种基于已有库元件的约束性设计,可大大简化设计过程、缩短设计周期,并提高芯片的成品率,对于小规模设计生产和实验具有明显的优势。

目前,作为 ASIC 领域中的一种半定制方式,用可编程逻辑器件 PLD(Programmable Logic Device)来进行 ASIC 设计已得到广泛应用。用户借助于特定的电子设计自动化(EDA, Electronic design automation)软件,即可在实验室内快速、方便地开发出功能独特的专用集成电路,并且还可以方便地实现加密和重新编程,从而大大加快硬件系统的设计速度、降低了设计成本、提高了系统的可靠性、灵活性和保密性。

可编程逻辑器件 PLD 的发展大致可分为 4 个阶段,即从 20 世纪 70 年代初到 70 年代中为第 1 阶段,20 世纪 70 年代中到 80 年代中为第 2 阶段,20 世纪 80 年代中到 90 年代末为第 3 阶段,20 世纪 90 年代末到目前为第 4 阶段。

第 1 阶段的可编程器件只有简单的可编程只读存储器(PROM, Programmable Read Only Memory)、紫外线可擦除只读存储器(EPROM, ultraviolet Erasable PROM)和电可擦除只读存储器(EEPROM, Electrically Erasable PROM)3 种。由于结构的限制,它们只能完成简单的数字逻辑功能。

第 2 阶段出现了结构上稍微复杂的可编程阵列逻辑(PAL, Programmable Array Logic)和

通用阵列逻辑(GAL, Generic Array Logic)器件,正式被称为 PLD,能够完成各种逻辑运算功能。典型的 PLD 由“与-或”阵列组成,借助“积之和”逻辑表达式可实现任意的组合逻辑。

第3阶段出现了可以完成超大规模复杂组合逻辑与时序逻辑的复杂可编程逻辑器件 CPLD(Complex Programmable Logic Device)和现场可编程逻辑阵列 FPGA(Field Programmable Gate Array),这类器件具有体系结构和逻辑单元灵活、集成度高以及适用范围广等特点,编程方式也更加灵活,成为产品原型设计和中小规模产品生产的首选。

第4阶段出现了片上可编程系统 SOPC(System On Programmable Chip)技术,新一代的 FPGA 可将中央处理器 CPU(Central Processing Unit)或数字信号处理器 DSP(Digital Signal Processor)内核集成进来,配合强大的专用 EDA 开发软件,在一片 FPGA 上即可完成软硬件协同设计,实现了高速与灵活性的完美结合,为构建片上可编程系统提供了强有力的硬件平台。

1.1.2 可编程逻辑器件特点

可编程逻辑器件具有以下特点:

(1) 研制周期短,设计灵活性高

可编程逻辑器件可以像通用器件一样在市场上购买,借助于专用的 EDA 软件开发工具,用户可任意定制其功能,并可反复修改,且设计与编程过程十分便捷。相对于由工厂编程的掩膜逻辑器件或全定制的 ASIC 芯片,产品的研制周期大大缩短,并具有更高的设计灵活性,极大地方便了设计更改及产品升级。

(2) 设计成本低

制作掩膜逻辑器件的一次性工程费用 NRE(Non-Recurring Engineering)往往较高,只有在生产批量足够大的情况下才有价值。同时,这种设计方法还需承担极大的风险,一旦设计出错或不完善,则前功尽弃,而采用可编程逻辑器件可随意修改其设计功能,为降低投资风险提供了更为适当的选择。

(3) 器件规模大,功能强

新型 FPGA 中内嵌 CPU 或 DSP 内核,支持软硬件协同设计,为实现 SOPC 片上可编程系统创造了条件。同时,FPGA 内嵌的高性能逻辑器件 Hard IP Core(硬知识产权内核),可以实现某些特定的高速复杂设计,例如,SPI 4.2、PCI Express、SDRAM 内存控制器等成熟标准和接口,从而可帮助用户加速研发进程,提高研发效率,规避研发风险。

(4) 开发工具简单易用

可编程逻辑器件开发工具功能强大,可以实现从设计输入、综合、实现到器件编程等一系列功能,并具有完善的仿真、优化、约束、调试等功能,可使设计人员专注于电路设计,提高研发效率。

1.1.3 可编程逻辑器件分类

广义上讲,可编程逻辑器件是指所有可通过软件手段更改、配置器件内部连接结构、逻辑单元及 IO 单元,完成特定逻辑功能的数字集成电路。目前常用的可编程逻辑器件一般可划分为简单可编程逻辑器件 SPLD(Simple Programmable Logic Device)、复杂可编程逻辑器件 CPLD 和现场可编程逻辑阵列 FPGA 三大类。

1. 简单可编程逻辑器件 SPLD

简单可编程逻辑器件主要指早期开发的 PLD 器件,它们通常由“与-或”阵列组成,能

够用于实现任意以“积之和”形式表示的各种布尔逻辑函数。常见的 SPLD 器件包括 PROM、可编程阵列逻辑 PAL 和通用阵列逻辑 GAL。PROM 具有固定的“与”阵列和可编程的“或”阵列，而 PAL 和 GAL 则具有可编程的“与”阵列和固定的“或”阵列。

PAL/GAL 是早期得到广泛应用的可编程逻辑器件。PAL 器件常用熔丝链路作为可编程开关，属于一次性可编程的；GAL 器件则采用 E²CMOS 工艺；可反复编程，为设计和修改带来了更大的方便。由于 PAL/GAL 结构较为简单，可编程逻辑单元密度较低，故只适用于相对简单的数字逻辑电路设计，尤其适用于对成本敏感的应用需求。目前比较大的 GAL 器件供应商主要是 Lattice 公司。

2. 复杂可编程逻辑器件 CPLD

CPLD 是 20 世纪 80 年代后期得到迅速发展的可编程逻辑器件。Altera 公司为了突出其产品特性，曾将自己的 CPLD 器件称为增强型可编程逻辑器件 EPLD(Enhanced Programmable Logic Device)。事实上，EPLD 和 CPLD 属于同质的逻辑器件，目前 Altera 为了遵循称呼习惯，已经将其 EPLD 统称为 CPLD。

CPLD 是在 PAL/GAL 的基础上发展起来的可编程逻辑器件，通过大量增加输出宏单元，提供更大的“与”阵列以及采用分层逻辑阵列结构等，使 CPLD 兼有成本低、速度高、设计简便等优点。CPLD 多采用 E²CMOS 工艺，也有少数厂商采用 Flash 工艺，掉电后其逻辑功能具有非易失性，方便用户使用。其基本结构一般包括：可编程 I/O 单元、基本逻辑单元、布线池和其他辅助功能模块等。在互连特性上，CPLD 采用连续互连方式，这种结构有利于方便地预测设计时延，保证了 CPLD 的高速性能。CPLD 的集成度一般可达数千甚至数万门，从而可完成较复杂、较高速度的逻辑功能，如接口转换、总线控制等。目前 CPLD 器件的主要供应商有 Altera、Lattice 和 Xilinx 等。

3. 现场可编程逻辑阵列 FPGA

FPGA 是在 CPLD 基础上发展起来的新型高性能可编程逻辑器件。其基本结构一般包括：可编程 I/O 单元、基本可编程逻辑单元、嵌入式块 RAM、可编程的布线资源、底层嵌入功能单元、内嵌专用硬核等。FPGA 的集成度很高，其器件密度从数万门到数千万门不等，可以完成极其复杂的时序与组合逻辑功能，适用于高速、高密度的高端数字逻辑电路设计。高端 FPGA 中还包含了其他丰富的内嵌资源，如锁相环(PLL/DLL)、存储器、数字信号处理器 DSP、中央处理单元 CPU 等嵌入式功能单元，为实现片上系统(SOPC)提供了丰富的硬件资源。典型的 FPGA 通常采用分段式互连结构，这种互连结构具有走线灵活、便于复杂功能的多级实现等优点，但同时也带来了布线复杂度增加、输入至输出的延时较大及总的时延性能难于估计等问题。

除少数 FPGA 专用器件采用 Flash 工艺或反熔丝(Anti-Fuse)工艺外，FPGA 一般采用 SRAM 工艺，可实现芯片功能的动态重构，但掉电后则会丢失内部逻辑配置，故需在每次上电时，从外部非易失存储器中导入配置信息。系统上电时，这些配置代码被读入 FPGA 内由 SRAM 构成的配置存储器，再由这些配置存储单元控制 FPGA 中的各种可编程资源，最终实现用户的定制功能。目前 FPGA 的主要器件供应商有 Xilinx、Altera 等。

1.1.4 主要可编程逻辑器件生产厂商及典型器件

(1) Altera

Altera 公司在 20 世纪 90 年代以后发展很快，是最大的可编程逻辑器件供应商之一。其主要的 CPLD 产品有 MAX3000/7000、MAX II 等系列，而主要的 FPGA 产品有 FLEX10K、

APEX20K、ACEX1K、Stratix、Cyclone 等系列，Altera 的 EDA 软件开发工具有 MAXPlus II 和 Quartus II。

(2) Xilinx

Xilinx 是 FPGA 技术的发明者，也是目前最大的可编程逻辑器件供应商之一。其产品种类较全，主要的 CPLD 产品有 XC9500/4000、CoolRunner (XPLA3) 等系列，而主要的 FPGA 产品有 Spartan、Virtex 等系列，其软件开发工具有 Foundation、Alliance 和 ISE。

(3) Lattice

Lattice 是在系统可编程技术 (ISP, In System Programmable) 的发明者。与 Altera 和 Xilinx 相比，Lattice 的中小规模 SPLD 比较有特色，但在大规模 FPGA/CPLD 器件方面的竞争力略逊一筹。

(4) Actel

Actel 是一次性烧写的反熔丝 PLD 技术领导者。由于反熔丝 PLD 抗辐射、耐高低温、功耗低、速度快，因此在军品和宇航级上有较大优势。

其他如 Cypress、QuickLogic、Lucent、Atmel 等公司，也有一些各具特色的 PLD 产品。

1.2 FPGA/CPLD 的基本结构

本节将在讨论 CPLD 与 FPGA 基本结构的基础上，比较两者的异同，以加深对这两种最常用的可编程逻辑器件的认识。

1.2.1 CPLD 的基本结构

人们发现，无论是简单还是复杂的数字电路系统都可由与、或、非基本门电路构成。由基本门可构成两类数字电路，一类是组合电路，其输出总是当前输入状态的函数；另一类是时序电路，其输出是当前系统状态与输入状态的函数，它含有存储元件。事实上，任何的组合逻辑函数都可以化为“与-或”表达式 (或称乘积项: Product-Term)，即任何的组合电路可以用“与门-或门”二级电路实现 (加上输入状态的非门)。同样，任何时序电路都可由组合电路加上存储元件，即锁存器、触发器等构成。

CPLD 一般基于乘积项结构，采用可编程的“与”阵列和固定的“或”阵列结构，再加上一个全局共享的可编程“与”阵列，把多个宏单元连接起来，并根据需要增加可编程的 I/O 控制模块。典型的 CPLD 如 Altera 的 MAX7000、MAX3000 系列器件，Lattice 的 ispMACH4000、ispMACH5000 系列器件，Xilinx 的 XC9500、CoolRunner II 系列器件等都是基于乘积项的 CPLD。

CPLD 主要包含以下逻辑资源：可编程 I/O 单元、基本逻辑单元、布线池和其他辅助功能模块，如图 1.1 所示。

(1) 可编程 I/O 单元

输入/输出单元是芯片与外界电路的接口部分，用于实现不同电气特性下对 I/O 信号的驱动与匹配需求，比如可以设定不同的 I/O 电压级别，指定集电极开路输出、三态输出及终端匹配电阻等。

(2) 基本逻辑单元

基本逻辑单元构成 CPLD 器件的逻辑构造核心。CPLD 中的基本逻辑单元常被称为宏单元 MC (Macro Cell)。所谓宏单元，其本质是由一些“与”、“或”阵列加上触发器构成。在

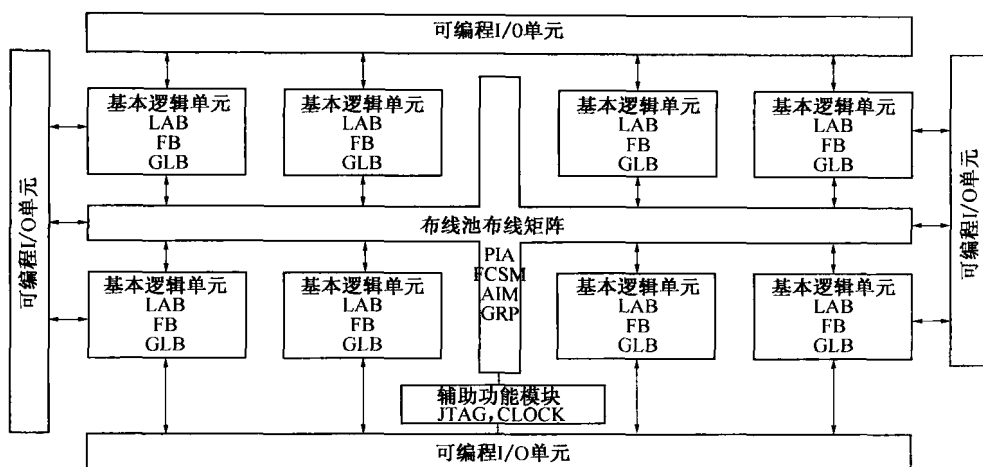


图 1.1 CPLD 的结构示意图

CPLD 中，将 MC 中“与”阵列的输出称为乘积项，而在“与”阵列后一般还有一个“或”阵列，“与-或”阵列配合工作，则可实现复杂的组合逻辑功能。MC 中的可编程触发器则一般还包含时钟、复位/置位配置功能，可用以实现时序逻辑的寄存器/锁存器等功能。多个 MC 按需要连接起来即可完成更复杂的逻辑功能。

例如，若组合逻辑的输出要求为 $f = (A + B) \times C \times \bar{D}$ ，对上式进行逻辑变换转换成“与-或”表述式，可得 $f = A \times C \times \bar{D} + B \times C \times \bar{D}$ 。A、B、C、D 信号由 CPLD 芯片的引脚输入后进入互连矩阵，在内部会产生 A 、 $\bar{A}$ 、 B 、 $\bar{B}$ 、 C 、 $\bar{C}$ 、 D 、 $\bar{D}$ 8 个信号。图 1.2 中每一个点表示相连(可编程开关接通)，从而可得： $f = f_1 + f_2 = A \times C \times \bar{D} + B \times C \times \bar{D}$ ，这样就实现了所需的组合逻辑。

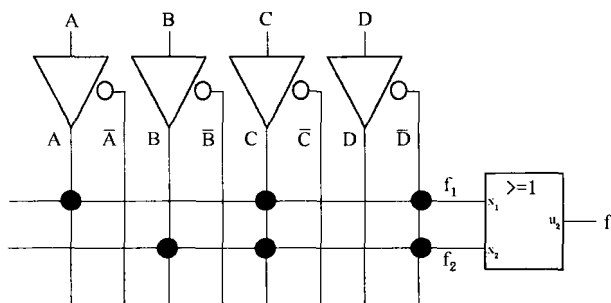


图 1.2 CPLD 中“与-或”逻辑实现示意图

不同厂家的 MC 宏单元功能相仿，但结构略有不同，其命名方式也有差异。如 Altera 的 MAX7000、MAX3000 系列 CPLD 称之为逻辑阵列模块 LAB (Logic Array Block)；Xilinx 9500 和 CoolRunner II 称之为功能模块 FB (Function Block)；Lattice 的 LC4000、ispLSI5000、ispLSI2000 系列 CPLD 则称之为通用逻辑模块 GLB (Generic Logic Block)。

(3) 布线池、布线矩阵

CPLD 的结构比较简单，其布线资源也相对有限，一般采用集中式布线池结构。所谓布线池其本质就是一个开关矩阵，通过打结点可以完成不同 MC 的输入与输出项之间的连接。Altera 的布线池叫做可编程互联阵列 (PIA, Programmable Interconnect Array)；Xilinx 9500 系

列 CPLD 的布线池被称为高速互联与交叉矩阵(Fast Connect Switch Matrix), CoolRunler II 系列 CPLD 被称之为先进的互联矩阵(AIM, Advanced Interconnect Matrix); Lattice 的布线池则被称为全局布线池(GRP, Global Routing Pool)。

由于 CPLD 的布线池结构固定, 故其输入管脚到输出管脚的标准延时也固定, 被称为 Pin to Pin 延时 Tpd, 它决定了 CPLD 器件可以实现的最高工作频率。

(4) 其他辅助功能模块

CPLD 中还有一些辅助的功能模块, 如 JTAG(IEEE 1532、IEEE 1149.1)边界扫描测试功能模块, 一些全局时钟、全局使能、全局复位/置位单元等。

1.2.2 FPGA 的基本结构

简化的 FPGA 基本由六部分组成, 分别为: 可编程 I/O 单元、基本可编程逻辑单元、嵌入式块 RAM、布线资源、底层嵌入功能单元和内嵌专用硬核等, 如图 1.3 所示。

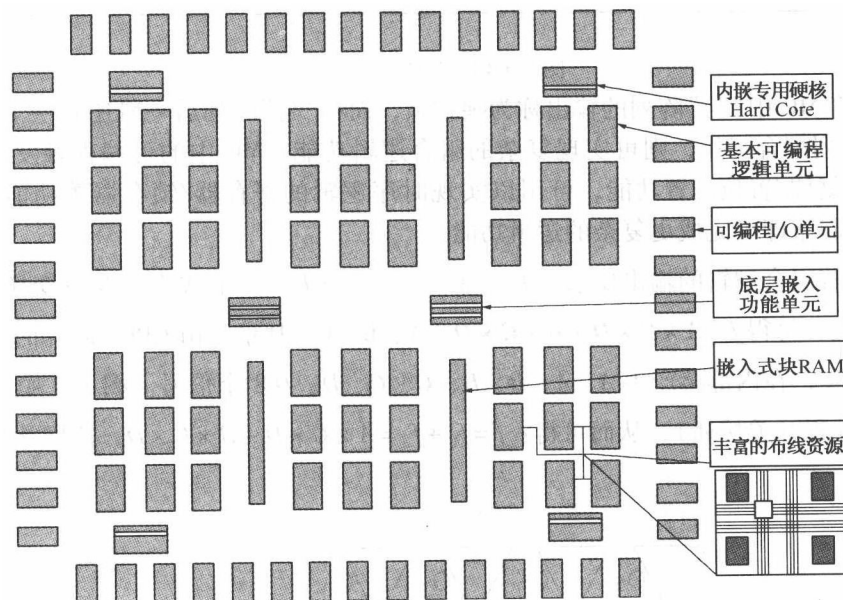


图 1.3 FPGA 结构原理图

(1) 可编程 I/O 单元

为了使 FPGA 能够适应各种 I/O 需求, 其 I/O 单元被设计为可编程模式, 以适配不同的电气标准与 I/O 物理特性。常见的电气标准有 LVTTTL、LVCMOS、LVDS、LVPECL 等。此外, I/O 单元的上下拉电阻, 输出驱动电流大小及匹配阻抗特性等常常也可通过编程指定。

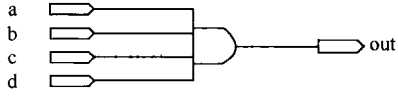
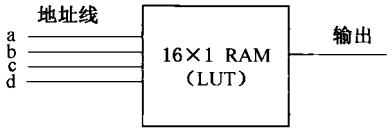
(2) 基本可编程逻辑单元

基本可编程逻辑单元是 FPGA 的逻辑核心, 可以根据用户的需要灵活地改变其内部连接与配置, 以实现不同的逻辑功能。与 CPLD 不同, FPGA 的基本可编程逻辑单元一般都是由查找表(LUT, Look Up Table)和寄存器(Register)组成。前者主要用于实现纯组合逻辑功能, 而后者则可用于实现同步/异步时序逻辑功能。

LUT 本质上就是一个 RAM。目前 FPGA 中多使用 4 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。当用户通过原理图或 HDL 语言描述一个逻辑电路后, FPGA 开发软件则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。表 1.1 所示为 LUT 实现一个 4 输入与门的例子。

表 1.1 LUT 实现一个 4 输入与门的例子

实际逻辑电路		LUT 的实现方式	
			
a, b, c, d 输入	逻辑输出	地址	RAM 中储存的内容
0000	0	0000	0
0001	0	0001	0
.....	0		0
1111	1	1111	1

常见的基本可编程单元的配置是一个寄存器加一个查找表，实际上不同厂商与器件族的寄存器和查找表的内部结构及组合模式往往不同。例如，Altera 可编程逻辑单元通常被称为逻辑单元(LE, Logic Element)，由一个寄存器加一个查找表构成。Altera 大多数 FPGA 还将 10 个 LE 有机地组合起来，构成更大功能单元 – 逻辑阵列模块(LAB, Logic Array Block)，LAB 中除了 LE 还包含 LE 间的进位链、LAB 控制信号、局部互联线资源、查找表级联链、寄存器级联链等连线与控制资源。Xilinx 可编程逻辑单元叫 Slice，由两个部分构成，每个部分都由一个查找表加一个寄存器组成，被称为逻辑单元(LC, Logic Cell)，两个 LC 之间有一些共用逻辑，可以完成 LC 之间的配合与级联。Lattice 的底层逻辑单元叫可编程功能单元(PFU, Programmable Function Unit)，它由 8 个查找表和 8 ~ 9 个寄存器构成。目前新型 FPGA 可编程单元的配置结构也在不断发展更新之中，常常根据设计需求推出一些新的 LUT 和 Register 配置比率，并优化其内部的连接构造。

很多 FPGA 器件手册上常用器件的逻辑门数表示器件的规模，但是，由于 FPGA 内部除了基本可编程逻辑单元外，常常还包含有丰富的嵌入式块 RAM，专用硬知识产权功能核(Hard IP Core)等，这些功能模块均会等效出一定规模额外的逻辑门。所以，用逻辑门数衡量基本可编程逻辑单元的数量并不准确，而比较简单的方法是用器件的寄存器或查找表数量(两者的比率一般为 1:1)来衡量。例如，Xilinx 的 XC3S1000 有 15360 个查找表，Lattice 的 LFEC15E 也有 15360 个查找表，而 Altera 的 EPIC12 仅有 12060 个查找表，所以，前两款 FPGA 的可编程逻辑单元数量基本相当，属于同一规模的产品，而 EPIC12 则比前两款 FPGA 规模略小。

(3) 嵌入式块 RAM

设计数字信号处理(DSP)、数据加密或数据压缩等复杂数字系统时，经常要用到存储器。FPGA 中内嵌的存储器不仅可以简化系统设计，提高系统工作速度，提高芯片内外数据交换的可靠性，而且还可降低数据存储的成本。

目前大多数 FPGA 都有内嵌的块 RAM(Block RAM)，并可灵活配置为单端口 RAM(SPRAM, Single port RAM)、双端口 RAM(DPRAM, Double Ports RAM)、伪双端口 RAM(Pseudo DPRAM)、CAM(Content Addressable Memory)、FIFO(First In First Out)等常用存储