

高等院校信息技术规划教材

数字信号处理的 FPGA 实现

王旭东 潘明海 编著

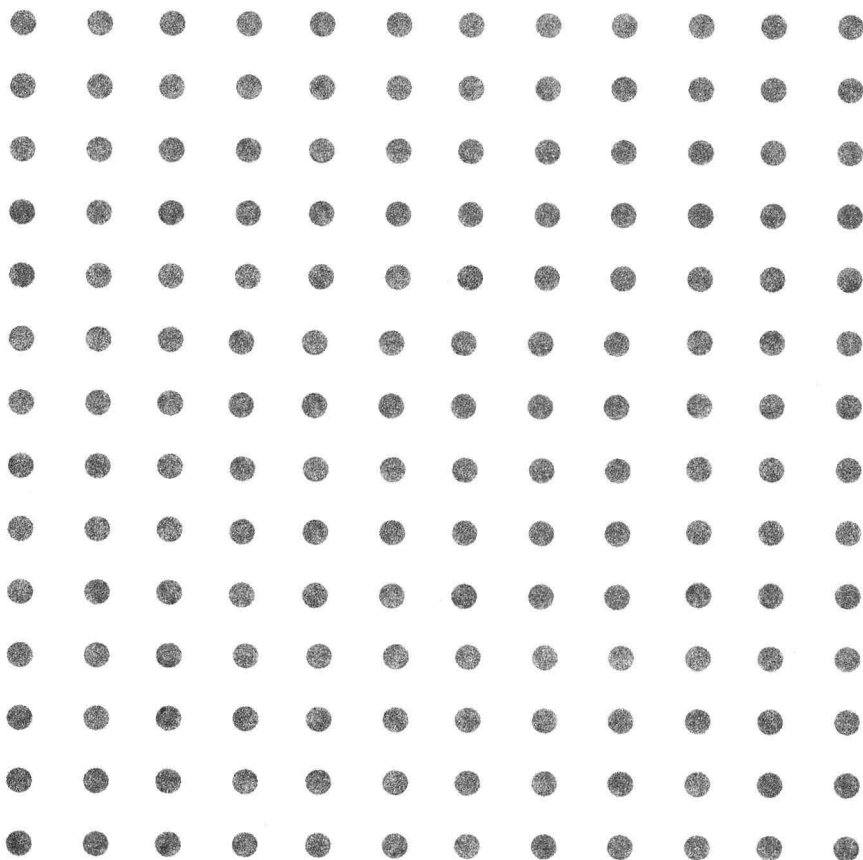


清华大学出版社

高等院校信息技术规划教材

数字信号处理的 FPGA 实现

王旭东 潘明海 编著



清华大学出版社
北京

内 容 简 介

本书比较全面地阐述了 FPGA 在数字信号处理中的应用问题。全书共分 8 章, 主要内容包括典型 FPGA 器件的介绍、VHDL 硬件描述语言、FPGA 设计中常用软件简介、用 FPGA 实现数字信号处理的数据规划、多种结构类型的 FIR 数字滤波器的 FPGA 实现、不同结构 FFT 的 FPGA 实现、数字正交下变频的 FPGA 实现、CORDIC 和 DDS 的 FPGA 实现等。全书紧密贴合工程实践, 以一个 FPGA 设计开发人员的切身体会去叙述每一个应用实例, 以一个 FPGA 教育工作者的实践经验去梳理和组织繁杂的知识点。

本书可作为高等院校通信、数字信号处理、电子工程等专业的本科生教材, 也可供相关专业的研究生和从事雷达、电子侦察、通信等工作的技术人员参考。

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

数字信号处理的 FPGA 实现/王旭东等编著. —北京: 清华大学出版社, 2011. 3

(高等院校信息技术规划教材)

ISBN 978-7-302-24266-6

I. ①数… II. ①王… III. ①现场可编程门阵列—应用—数字信号—信号处理—高等学校—教材 IV. ①TN911.72

中国版本图书馆 CIP 数据核字(2010)第 250050 号

责任编辑: 张瑞庆 顾 冰

责任校对: 白 蕾

责任印制: 李红英

出版发行: 清华大学出版社

地 址: 北京清华大学学研大厦 A 座

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

邮 编: 100084

社 总 机: 010-62770175

邮 购: 010-62786544

投稿与读者服务: 010-62795954, jsjic@tup.tsinghua.edu.cn

质 量 反 馈: 010-62772015, zhiliang@tup.tsinghua.edu.cn

印 装 者: 北京国马印刷厂

经 销: 全国新华书店

开 本: 185×260 印 张: 19.75

字 数: 469 千字

版 次: 2011 年 3 月第 1 版

印 次: 2011 年 3 月第 1 次印刷

印 数: 1~3000

定 价: 35.00 元

产品编号: 036931-01

FPGA(Field Programmable Gate Array,现场可编程门阵列)是在 PAL、GAL、EPLD 等可编程器件的基础上进一步发展的产物,具有更高的集成度、更强的逻辑实现能力和更好的设计灵活性。它是作为专用集成电路(ASIC)领域中的一种半定制电路而出现的,既解决了定制电路的不足,又克服了原有可编程器件门电路数有限的缺点。采用 FPGA 设计 ASIC 电路,用户不需要投片生产,就能得到可用的芯片。FPGA 还可以做其他全定制或半定制 ASIC 电路的中试样片。FPGA 是 ASIC 电路中设计周期最短、开发费用最低、风险最小的器件之一。可以说,FPGA 芯片是小批量系统提高系统集成度、可靠性的最佳选择之一。同时,相对 ASIC 来说,FPGA 技术难度大、生产成本低,符合现阶段中国国情。

数字信号处理(Digital Signal Processing, DSP)是一门涉及许多学科而又广泛应用于许多领域的新兴学科。20 世纪 60 年代以来,随着计算机和信息技术的飞速发展,数字信号处理技术应运而生并得到迅速的发展。在过去的几十年时间里,数字信号处理已经在雷达、通信、电子侦察等领域得到极为广泛的应用。

数字信号处理的实现方法一般有以下几种:

- (1) 在通用的计算机(如 PC)上用软件(如 FORTRAN、C 语言)实现;
- (2) 在通用计算机系统中加上专用的加速处理机实现;
- (3) 用通用的单片机(如 MCS-51、96 系列等)实现;
- (4) 用通用的可编程 DSP 芯片实现;
- (5) 用 FPGA 芯片实现。

在上述几种方法中,基于 DSP 芯片的方法在过去的 30 年里取得了长足进步,为数字信号处理的工程应用作出了突出贡献。但是,随着数字技术的提高,要求的信号处理速度也越来越高,用通用 DSP 芯片很难实现。

近年来,基于 FPGA 的 DSP 技术发展迅速,各种大容量、高性能、低功耗的可编程逻辑器件不断推出。同时,出现了许多 FPGA



设计辅助工具,这些工具大大提高了新型集成电路的设计效率,使更低成本、更短周期的复杂数字系统开发成为可能。设计者只需编写硬件描述语言代码,然后选择目标器件,在集成开发环境里进行编译、仿真、综合,最后在线下载调试。整个过程,大部分工作由软件完成。这些软件的推出,极大地促进了集算法设计、芯片编程、电路板设计于一体的可编程技术的发展。

本书在内容安排上共分8章、3大部分内容。第1章主要讲述FPGA器件的基础知识,重点讲述书中设计实例中经常用到的FPGA芯片;第2、3章主要介绍开发FPGA所用硬件描述语言VHDL,以及FPGA设计中常用的软件;第4、5、6、7、8章首先简单介绍了用FPGA实现数字信号处理的一般方法,然后对数字信号处理中常用的FIR、FFT、CORDIC、DDS等算法的FPGA实现进行了详细设计和仿真。

本书在编写过程中,得到了清华大学出版社的大力支持;另外还得到了Altera公司、Xilinx公司和www.fpga.com.cn网站的支持;南京航空航天大学信息科学与技术学院电子侦察实验室的刘渝教授、高翔(2003年毕业)、黄铮(2004年毕业)、徐文明(2005年毕业)、马健(2006年毕业)、张棣棣(2006年毕业)、成大伟(2007年毕业)、王琦(2008年毕业)、陈役涛(2008年毕业)、谢欲进(2009年毕业)、张鑫鑫(2009年毕业)、姜建军(2010年毕业)、杨健(在读)、叶金伟(在读)研究生在本书的编写过程中给予作者诸多鼓励和帮助,在此对他们一并表示感谢,感谢南京航空航天大学基本科研业务费专项科研项目(No. NS2010109)、国家自然科学基金项目(No. 61071164)对本书的资助。

由于作者水平有限,书中错误、不妥之处,恳请广大读者批评指正。

编者

2011年1月

目录

Contents

第 1 章	FPGA 基础知识	1
1.1	PLD 演化过程	1
1.2	PLD 分类	2
1.3	FPGA 的诞生	3
1.4	FPGA 结构特点	4
1.4.1	基于查找表的 FPGA 结构	4
1.4.2	基于乘积项的 CPLD	8
1.4.3	其他结构 FPGA 器件	10
1.5	FPGA 内部编程方式	11
1.6	Altera 公司典型 FPGA 器件	11
1.7	FPGA 实现数字信号处理的流程	14
1.7.1	FPGA 实现数字信号处理系统概述	14
1.7.2	基于单一软件的 FPGA 设计流程	16
1.7.3	基于多种软件的 FPGA 设计流程	19
1.8	PLD 器件选型	22
1.8.1	FPGA 与 CPLD 的比较	22
1.8.2	CPLD 的应用场合	23
1.8.3	FPGA 的应用场合	25
第 2 章	VHDL 硬件描述语言	26
2.1	硬件描述语言综述	26
2.1.1	硬件描述语言现状	26
2.1.2	硬件描述语言的发展历史	26
2.1.3	使用硬件描述语言的理由	27
2.1.4	硬件描述语言的主要特征	28
2.1.5	硬件描述语言设计流程及设计方法	29
2.1.6	Verilog HDL 与 VHDL 比较	31

2.1.7	HDL 与计算机语言的区别	32
2.1.8	硬件描述语言发展趋势	32
2.2	VHDL 硬件描述语言基本结构	33
2.2.1	VHDL 的基本组成	33
2.2.2	实体	35
2.2.3	结构体	37
2.2.4	进程	41
2.3	VHDL 数据对象	42
2.3.1	常数	42
2.3.2	信号	42
2.3.3	变量	43
2.3.4	信号与变量的区别	43
2.4	VHDL 数据类型	47
2.4.1	枚举型	47
2.4.2	整数型、实数型	47
2.4.3	阵列数据类型	48
2.4.4	记录型	49
2.4.5	VHDL 数据类型及子类型	49
2.4.6	VHDL 数据类型转换	50
2.5	属性	51
2.6	VHDL 运算符	51
2.6.1	逻辑运算符	52
2.6.2	关系运算符	53
2.6.3	算术运算符	54
2.6.4	并置(连接)运算符	54
2.7	组合逻辑电路设计	55
2.7.1	并行语句	55
2.7.2	顺序语句	57
2.7.3	几种语句的比较	63
2.8	同步时序逻辑电路设计	63
2.9	状态机的优化设计	64
2.10	层次化设计	70
2.11	库	72
2.12	包	74
2.13	元件	75
2.14	函数	77
2.15	过程	78

第 3 章	FPGA 设计常用软件	80
3.1	FPGA 设计时常用 MATLAB 功能	80
3.1.1	MATLAB 编程基本操作	80
3.1.2	基本运算操作	81
3.1.3	MATLAB 中的矩阵操作	83
3.1.4	MATLAB 程序参数输入和输出操作	89
3.1.5	MATLAB 函数及调用	92
3.1.6	MATLAB 绘图功能	104
3.2	ISE 软件基本操作	109
3.2.1	ISE 功能简介	109
3.2.2	ISE 软件的基本操作	110
3.2.3	ISE 软件的项目管理	115
3.2.4	ISE 软件的开发流程	117
3.3	Quartus- II 软件基本操作	123
3.3.1	Quartus- II 项目操作	123
3.3.2	Quartus- II 设计输入	127
3.3.3	Quartus- II 与其他 EDA 软件连接	133
3.3.4	Quartus- II 编译	137
3.3.5	Quartus- II 延时分析	138
3.3.6	Quartus- II 仿真	139
3.3.7	Quartus- II 下载	142
3.4	ModelSim 软件基本操作	143
3.4.1	ModelSim 简介	143
3.4.2	代码仿真	143
3.4.3	门级仿真和时序仿真	147
3.4.4	仿真需要的文件	147
3.4.5	仿真步骤	147
第 4 章	FPGA 实现数字信号处理概述	151
4.1	数字信号处理简介	151
4.2	FPGA 实现数字信号处理的数据表示	152
4.2.1	数的定标	152
4.2.2	FPGA 定点算术运算	154
4.2.3	非线性运算的定点快速实现	156

第 5 章	数字滤波器的 FPGA 设计	160
5.1	分布式结构 FIR 滤波器设计	160
5.1.1	算法分析	160
5.1.2	FPGA 设计	162
5.1.3	仿真验证	169
5.1.4	设计扩展	169
5.1.5	Quartus-II 与 MATLAB 联合仿真	170
5.2	流水线结构 FIR 滤波器设计	173
5.2.1	算法分析	173
5.2.2	模型设计	175
5.2.3	设计的 HDL 描述	175
5.2.4	细化设计	176
5.2.5	参数设置	176
5.2.6	乘法器的变形	177
5.2.7	仿真验证	178
5.3	基于 Altera IP 核的 FIR 滤波器设计	182
5.3.1	参数设置	183
5.3.2	仿真验证	190
5.4	基于 Xilinx IP 核的 FIR 滤波器设计	194
5.4.1	参数设置	194
5.4.2	仿真验证	199
第 6 章	离散傅里叶变换的 FPGA 设计	202
6.1	高速并行结构 FFT 的 FPGA 实现	202
6.1.1	用 FPGA 实现 FFT 简介	202
6.1.2	高速并行 FFT 算法简介	203
6.1.3	高速并行 FFT 算法 MATLAB 仿真	206
6.1.4	高速并行 FFT 算法 FPGA 实现	207
6.1.5	用 ModelSim 与 MATLAB 联合仿真验证设计	211
6.2	固定几何结构 FFT 的 FPGA 实现	215
6.2.1	算法简介	215
6.2.2	数据格式的选择	216
6.2.3	算法 FPGA 实现	218
6.2.4	设计验证	221
6.3	基-4 串行结构 FFT 的设计实现	223
6.3.1	设计简介	223

6.3.2	设计框图及接口时序	223
6.3.3	FPGA 设计实现	224
6.3.4	设计的仿真验证	225
6.4	基于 FIR 滤波结构的高速 DFT 设计	229
6.4.1	算法简介	229
6.4.2	FPGA 设计	230
6.4.3	仿真验证	232
6.5	串并混合结构 FFT 的 FPGA 设计实现	234
6.5.1	算法分析	234
6.5.2	FPGA 设计实现	234
6.5.3	仿真验证	238
6.6	基于 Altera IP 核的 FFT 设计实现	239
6.6.1	IP 核简介	239
6.6.2	下载和安装 FFT 核	239
6.6.3	创建一个新项目	239
6.6.4	用 MATLAB 与 Quartus-II 对设计进行仿真验证	241
6.7	基于 Xilinx IP 核的 FFT 设计实现	243
6.7.1	参数设置	243
6.7.2	IP 核的引用	245
6.7.3	仿真验证	247
第 7 章	数字正交变换算法的 FPGA 设计	250
7.1	正交变换简介	250
7.2	模拟正交变换	250
7.3	直接采样数字正交变换的 FPGA 设计	251
7.3.1	算法分析	251
7.3.2	算法 MATLAB 仿真	253
7.3.3	算法 FPGA 实现	255
7.4	基于 FFT/IFFT 的数字正交变换法的 FPGA 设计	259
7.4.1	算法分析	259
7.4.2	FPGA 设计	260
7.4.3	仿真验证	262
7.5	多相滤波结构数字正交变换的 FPGA 设计	263
7.5.1	算法分析	263
7.5.2	FPGA 设计	264
7.5.3	设计验证	266
7.6	大带宽数字正交变换法的 FPGA 设计	271
7.6.1	算法分析	271

7.6.2	算法 MATLAB 仿真	273
7.6.3	算法 FPGA 实现	274
7.6.4	FPGA 实现后的时序验证	275
第 8 章	CORDIC 与 DDS 的 FPGA 设计	278
8.1	CORDIC 和 DDS 算法简介	278
8.1.1	坐标旋转数字计算机	278
8.1.2	CORDIC 原理	278
8.1.3	三角函数的计算	279
8.1.4	CORDIC 算法的 MATLAB 验证	280
8.1.5	用 CORDIC 实现数控振荡器	280
8.1.6	DDS 基本原理	281
8.2	CORDIC 算法的 FPGA 实现	282
8.2.1	用 HDL 实现	282
8.2.2	基于 Xilinx 的 IP 核实现	286
8.3	DDS 算法的 FPGA 实现	291
8.3.1	用 HDL 在 Altera FPGA 中实现 DDS	291
8.3.2	用 HDL 在 Xilinx FPGA 中实现 DDS	294
8.3.3	用 Xilinx IP 核实现 DDS	296
	参考文献	301

FPGA 基础知识

FPGA(Field Programmable Gate Array, 现场可编程门阵列)是在 PAL、GAL、EPLD 等可编程器件的基础上进一步发展的产物。可编程逻辑器件(Programmable Logic Device, PLD)是一种由用户编程来实现某种逻辑功能的新型逻辑器件,芯片内的逻辑门、触发器等硬件资源可由用户配置连接来实现专用的用户逻辑功能。与只能实现固定功能的传统的标准逻辑器件(如 74 系列的 TTL 器件)相比,PLD 器件可反复修改,并且在满足专用的、个性化的设计需求方面具有更大的灵活性和竞争力。同时与大规模专用集成电路相比,用 PLD 实现的数字系统,有研制周期短、先期投资少、无风险、修改逻辑设计方便和小批量生产成本低等优势。

近年来,随着 PLD 性能价格比的不断提高,电气自动化设计(EDA)开发软件的不断完善,现代电子系统的设计将越来越多地使用 PLD,特别是大规模 PLD。如果说一个电子系统可以像积木块一样堆积起来的话,那么现在构成许多电子系统仅需 3 种标准的积木块:微处理器、存储器和 PLD,甚至只需一块大规模 PLD。

1.1 PLD 演化过程

最早的 PLD 出现在 20 世纪 70 年代初,主要是可编程只读存储器(PROM)和可编程逻辑阵列(PLA)。70 年代末出现了可编程阵列逻辑(Programmable Array Logic, PAL)器件。80 年代初期,美国 Lattice 公司推出了一种新型的 PLD 器件,称为通用阵列逻辑(Generic Array Logic, GAL),一般认为它是第二代 PLD 器件,其详细说明见本章的后续章节。

随着技术的进步,生产工艺的不断改进,器件规模不断扩大,逻辑功能不断增强,各种 PLD 如雨后春笋般涌现,如 PROM、EPROM、E²PROM 等。最初把在 EPROM 基础上出现的高密度 PLD 称为 EPLD 或 CPLD,而现在一般把超过某一集成度的 PLD 器件都称为 CPLD,其内部结构如图 1.1 所示,其详细说明见本章的后续章节。

在 20 世纪 80 年代中期,美国 Xilinx 公司首先推出了现场可编程门阵列器件(FPGA)。不同于 CPLD 的是,FPGA 器件采用逻辑单元阵列结构和静态随机存取存储器工艺,设计灵活、集成度高、可无限次反复编程,并可现场模拟调试验证。随后,在 90 年代初,Lattice 公司又推出了在系统可编程大规模集成电路(ispLSI)。

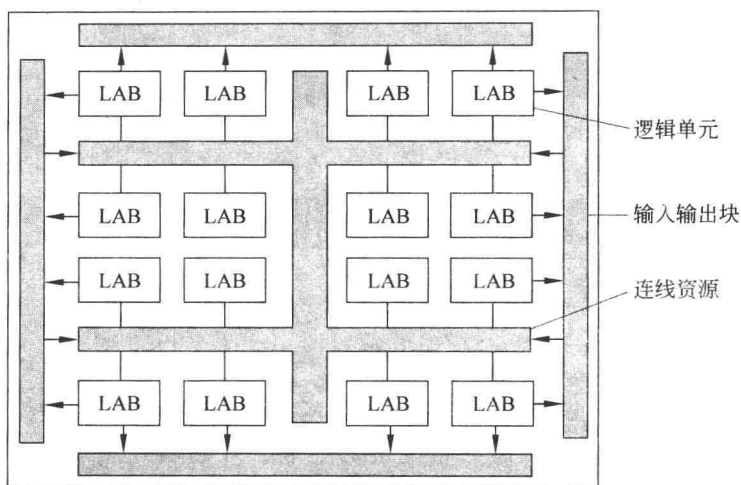


图 1.1 典型 CPLD 的总体内部框图

近年来,随着集成电路的深亚微米制造技术和设计技术的迅速发展,集成电路进入了片上系统,即所谓的系统级芯片(System On a Chip,SOC)时代。它在单一芯片上实现信号采集、转换、存储、处理和 I/O 等功能,或者说在单一芯片上集成了数字电路、模拟电路、信号采集和转换电路、存储器、MPU、MCU、DSP、MPEG 等,实现了过去需要多片集成电路才能完成的功能。随着 SOC 技术的发展,也出现了面向 SOC 的 PLD。Xilinx 公司推出的百万门 Virtex 系列 FPGA,就为解决系统级设计问题提供了新的 FPGA 平台。Altera 公司于 1999 年第一季度推出的 APEX20K 系列,成功地将乘积项、查找表以及与内嵌存储器结合为一体,同时它是第一个密度在百万门、系统性能可支持 64 位/66MHz PCI 标准的产品系统,同时也使整个复杂的系统集成在一片 PLD 上成为可能。Altera 公司的 APEX II 系列还集成一个 60MIPS 的 CPU。目前 Xilinx 公司和 Altera 公司的 SOC PLD 芯片密度已经达到上千万门。

目前世界各著名半导体器件公司,如 Altera、Xilinx、Lattice 等公司,均提供了不同类型的 CPLD、FPGA 产品,新的 PLD 产品不断面世。众多公司的竞争促进了可编程集成电路技术的提高,使其性能不断完善。

1.2 PLD 分类

随着微电子技术的发展,PLD 的品种越来越多,型号也越来越复杂。每种器件都有自己的特征,而不同器件之间也存在许多共同点。因此了解 PLD 的分类情况,对于正确选用 PLD 就显得尤为重要。目前可编程器件尚无严格的划分标准,下面仅从集成度、可编程原理及结构特点 3 个方面对 PLD 进行简单的分类。

1. 按集成度分类

PLD 从集成密度上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)两类。通常,当 PLD 中的等效门数超过 500 门时,则认为它是 HDPLD。传统的 PAL 和 GAL 是典型的 LDPLD, EPLD、CPLD、FPGA 和 pLSI/ispLSI 则称为 HDPLD。

2. 按可编程原理分类

从可编程特性上可将 PLD 分成一次性可编程和重复可编程两类。一次性可编程的典型产品是 PROM、PAL 和熔丝型 FPGA,其他的可编程产品(如 EPROM、E²PROM、Flash)大多是重复可编程的。同时根据各种可编程元件的结构及编程方式,又可分为 4 类:采用一次性可编程的熔丝(Fuse)或反熔丝(Anti-fuse)元件的可编程器件;采用紫外线擦除、电可编程的 EPROM 工艺结构的可编程器件;采用电可擦除、电可编程的 E²PROM 工艺结构的可编程器件;基于静态存储器 SRAM 结构的器件。

3. 按结构特点分类

目前常用的 PLD 都是从与阵列、或阵列和门阵列发展起来的,所以可以从结构上将其分为两大类。

(1) 阵列型 PLD。阵列型 PLD 的基本结构由与阵列和或阵列组成。简单 PLD (PROM、PLA、PAL 和 GAL)、EPLD 和 CPLD 都属于阵列型 PLD。

(2) 现场可编程门阵列 FPGA。FPGA 具有门阵列的结构形式,它是由许多可编程逻辑单元(或称逻辑功能块)排成阵列组成的,这些逻辑单元的结构和与阵列、或阵列的结构不同,所以也将 FPGA 称为单元型 PLD。

1.3 FPGA 的诞生

相对普通 PLD, FPGA 具有更高的集成度、更强的逻辑实现能力和更好的设计灵活性。FPGA 由许多独立的可编程逻辑模块组成,用户可以通过编程将这些模块连接起来实现不同的设计,它是作为专用集成电路(ASIC)领域中的一种半定制电路而出现的,既解决了定制电路的不足,又克服了原有可编程器件门电路数有限的缺点。FPGA 具有掩膜可编程门阵列的通用结构,它由逻辑功能块排成阵列组成,并由可编程的互连资源连接这些逻辑功能块以及相应的输入/输出单元来实现不同的设计。其中, FPGA 的功能由逻辑结构的配置数据决定。工作时,这些配置数据存放在片内的 SRAM 或熔丝上。基于 SRAM 的 FPGA 器件,在工作前需要从芯片外部加载配置数据,配置数据可以存储在片外的 EPROM 或其他存储体上。用户可以控制加载过程,在现场修改器件的逻辑功能,即所谓的现场编程。

FPGA 采用了逻辑单元阵列(Logic Cell Array, LCA)这样一个新概念,内部包括可配置逻辑模块(Configurable Logic Block, CLB)、输入输出模块(Input Output Block, IOB)和内部连线(Interconnect)3 个部分。FPGA 的基本特点主要如下。

- (1) 采用 FPGA 设计 ASIC 电路,用户不需要投片生产,就能得到可用的芯片。
- (2) FPGA 可做其他全定制或半定制 ASIC 电路的中试样片。

- (3) FPGA 内部有丰富的触发器和 I/O 引脚资源。
 - (4) FPGA 是 ASIC 电路中设计周期最短、开发费用最低、风险最小的器件之一。
 - (5) FPGA 采用高速 CHMOS 工艺,功耗低,可以与 CMOS、TTL 电平兼容。
- 可以说,FPGA 芯片是小批量系统提高系统集成度、可靠性的最佳选择之一。

目前 FPGA 的品种很多,有 Xilinx 公司的 XC 系列、Altera 公司的 STRATIX 系列和 TI 公司的 TPC 系列等。

FPGA 是由存放在片内 RAM 中的程序来设置其工作状态的,因此,工作时需要对片内的 RAM 进行编程。用户可以根据不同的配置模式,采用不同的编程方式。

加电时,FPGA 芯片将 EPROM 中数据读入片内编程 RAM 中,配置完成后,FPGA 进入工作状态。掉电后,FPGA 恢复成空白状态,内部逻辑关系消失。因此,FPGA 能够反复使用。FPGA 的编程无须专用的 FPGA 编程器,只需通用的 EPROM、PROM 编程器即可。当需要修改 FPGA 功能时,只需修改 EPROM 中的内容即可。这样,同一片 FPGA,不同的编程数据,可以产生不同的电路功能。因此,FPGA 的使用非常灵活。

FPGA 有多种配置模式:并行主模式为一片 FPGA 加一片 EPROM 的方式;主从模式可以支持一片 PROM 编程多片 FPGA;串行模式可以采用串行 PROM 编程 FPGA;外设模式可以将 FPGA 作为微处理器的外设,由微处理器对其编程。

随着集成电路密度的不断提高,FPGA 器件几乎可以实现所有的数字电路功能。目前,器件集成度已经达到了数百万门数量级,工作频率可达 300MHz 以上。一般可在几个小时或几天内完成一个设计,并可随时修改。这加速了产品设计开发的进程,免去了昂贵的 ASIC(专用集成电路)器件半导体生产工艺费用及投资风险,因此,FPGA 技术已经成为电子工程师不可缺少的好帮手。

在发达国家,PLD 技术已经成为大学电子类专业的必修课程,它已经像单片机一样,广泛地应用在各种需要数字逻辑的领域。

1.4 FPGA 结构特点

各种 PLD 在结构上的差异主要反映在 CLB 上,常见的结构主要有 4 种类型,即查找表结构、乘积项结构、多路开关结构和多级与非门结构。

1.4.1 基于查找表的 FPGA 结构

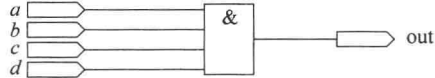
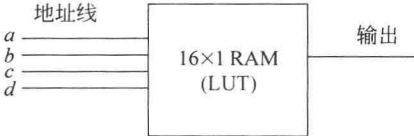
1. 查找表的基本原理

查找表(Look-Up-Table,LUT)本质上就是一个 RAM。采用这种结构的 PLD 芯片一般称为 FPGA,如 Altera 公司的 ACEX、APEX 系列器件,Xilinx 公司的 Spartan、Virtex 系列器件等。查找表型 FPGA 的 CLB 是查找表,由查找表构成函数发生器,通过

查找表来实现逻辑函数。查找表的物理结构是静态存储器(SRAM)。M 个输入项的逻辑函数可以由一个 2^M 位容量的 SRAM 实现,函数值存放在 SRAM 中。SRAM 的地址线起输入线的作用,地址即输入变量值,SRAM 的输出为逻辑函数值,由连线开关实现与其他功能块的连接。查找表结构的函数功能非常强大。M 个输入的查找表可以实现任意一个 M 个输入项的组合逻辑函数,这样的函数有 2^M 个。用查找表实现逻辑函数时,把对应函数的真值表预先存放在 SRAM 中即可实现相应的函数运算。

目前 FPGA 中多使用 4 输入的 LUT,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 语言描述了一个逻辑电路以后,PLD/FPGA 开发软件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 所示是一个用 LUT 实现 4 输入与门的例子。

表 1.1 用 LUT 实现 4 输入与门

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 中存储的内容
0000	0	0000	0
0001	0	0001	0
⋮	0	⋮	0
1111	1	1111	1

从表 1.1 中可以看出,当 4 输入与门的输入端 a、b、c、d 变化时,对应的输出端也跟着变化,如果把与门的输入端的 4 个引脚看做是 LUT 的输入地址,其寻址空间为 16,只要在这 16 个存储单元中存放和与门对应的输出逻辑值,就实现了用 LUT 替代与门的作用。

2. 基于查找表(LUT)的 FPGA 内部结构

下面以两大主要 FPGA 生产公司生产的主流 FPGA 器件来介绍 FPGA 的内部结构。图 1.2 所示是 Xilinx 公司的 Spartan-II 系列 FPGA 器件的内部结构。图 1.3 所示为该器件的内部 Slices 模块结构。

Xilinx 公司的 Spartan-II 系列 FPGA 器件主要包括 CLB 模块、I/O 块、RAM 块和可编程连线。在 Spartan-II 中,一个 CLB 包括两个 Slices 模块,每个 Slices 模块又包括两个 LUT、两个触发器和相关逻辑。Slices 可以看成是 Spartan-II 器件实现逻辑功能的最基本结构。Xilinx 公司的其他系列的 FPGA 器件的内部结构与此类似。

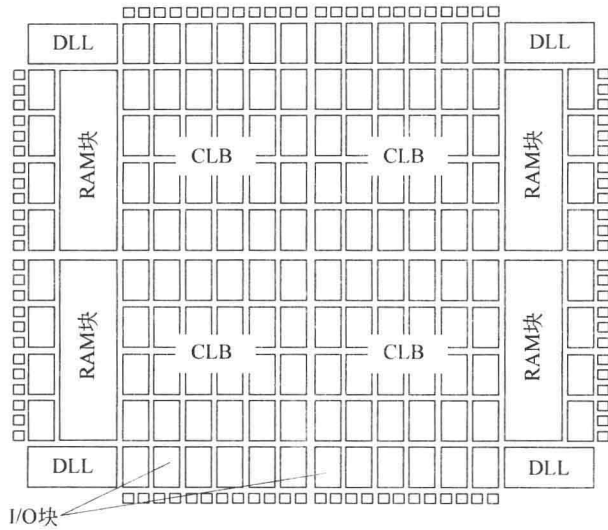


图 1.2 Xilinx 公司 Spartan-II 系列 FPGA 芯片内部结构

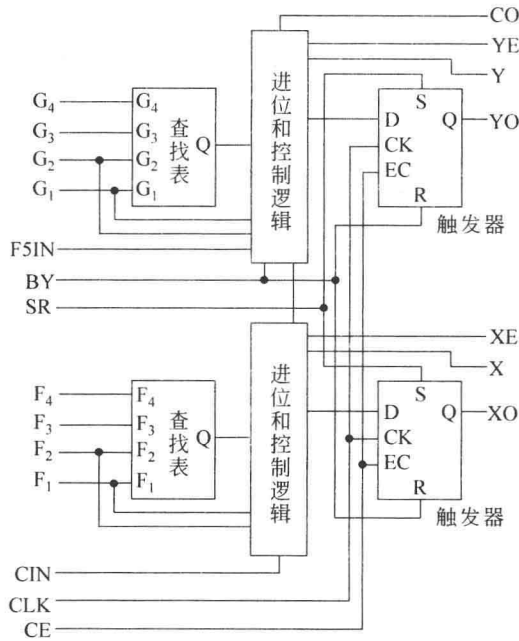


图 1.3 Spartan-II 器件内部 Slices 结构

Altera 公司的 FLEX/ACEX 系列 FPGA 芯片的内部结构如图 1.4 所示。其内部 LE(逻辑单元)结构如图 1.5 所示。

FLEX/ACEX 的结构主要包括 LAB、I/O 块、RAM 块和可编程行/列连线。在 FLEX/ACEX 中,一个 LAB 包括 8 个 LE,每个 LE 包括一个 LUT、一个触发器和相关逻辑。LE 是 FLEX/ACEX 芯片实现逻辑的最基本结构,Altera 其他系列,如 APEX 的结构与此基本相同,具体可参阅 Altera 公司为每款芯片提供的数据手册。