

嵌入式开发系列

FPGA应用开发入门 与典型实例

(修订版)

华清远见嵌入式培训中心 姚远 李辰 编著

从FPGA基础讲起，引导读者快速入门

21个典型实例，全面讲解FPGA在各个领域的应用

提供Altera的QuartusII和Xilinx的ISE两个版本的源文件下载

图文并茂，轻松阅读



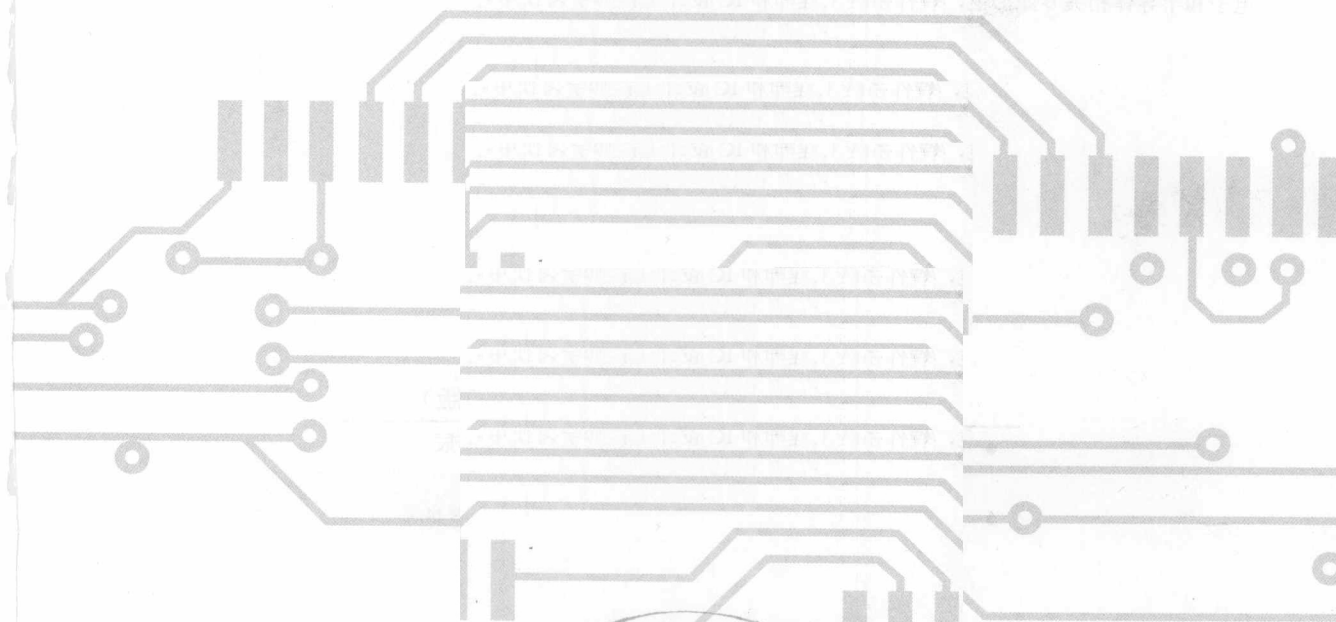
人民邮电出版社
POSTS & TELECOM PRESS

嵌入式开发系列

FPGA应用开发入门 与典型实例

(修订版)

华清远见嵌入式培训中心 姚远 李辰 编著



人民邮电出版社

北京

图书在版编目 (C I P) 数据

FPGA应用开发入门与典型实例 / 姚远, 李辰编著
— 2版 (修订本). — 北京: 人民邮电出版社,
2010.6
ISBN 978-7-115-22740-9

I. ①F… II. ①姚… ②李… III. ①可编程序逻辑器
件—系统开发 IV. ①TP332.1

中国版本图书馆CIP数据核字 (2010) 第066939号

内 容 提 要

FPGA (现场可编程逻辑器件) 以其体积小、功耗低、稳定性高等优点被广泛应用于各类电子产品的设计中。本书全面讲解了 FPGA 系统设计的背景知识、硬件电路设计, 硬件描述语言 Verilog HDL 的基本语法和常用语句, FPGA 的开发工具软件的使用, 基于 FPGA 的软核嵌入式系统, FPGA 设计的基本原则、技巧、IP 核, FPGA 在接口设计领域的典型应用, FPGA+DSP 的系统设计与调试, 以及数字变焦系统和 PCI 数据采集系统这两个完整的系统设计案例。

本书内容全面、实例丰富, 适合 FPGA 系统设计初学者, 大专院校通信工程、电子工程、计算机、微电子和半导体相关专业师生, 硬件系统工程师和 IC 设计工程师学习使用。

FPGA 应用开发入门与典型实例 (修订版)

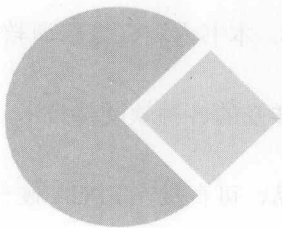
- ◆ 编 著 华清远见嵌入式培训中心 姚远 李辰
责任编辑 黄 焱
 - ◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街 14 号
邮编 100061 电子函件 315@ptpress.com.cn
网址 <http://www.ptpress.com.cn>
北京顺义振华印刷厂印刷
 - ◆ 开本: 787×1092 1/16
印张: 27.5
字数: 671 千字
印数: 5 801—9 800 册
- 2010 年 6 月第 2 版
2010 年 6 月北京第 1 次印刷

ISBN 978-7-115-22740-9

定价: 49.80 元

读者服务热线: (010) 67132692 印装质量热线: (010) 67129223

反盗版热线: (010) 67171154



前言

FPGA 系统设计技术背景

目前数字电路系统设计领域公认的基础性技术分别是 CPU、DSP 和 FPGA。其中 FPGA 技术发展迅速，正在逐渐融合 CPU 和 DSP 的功能。FPGA 不仅可以解决电子系统小型化、低功耗、高可靠性等问题，而且其开发周期短、投入少，芯片价格又在不断下降。

正是因为这些优势，FPGA 已经被广泛应用于各类电子产品的设计之中，从高端的通信系统设备，如无线基站、千兆网络路由器等，到低成本、大批量的消费类产品，如智能手机、便携式产品、数码相机等领域。

因此，FPGA 系统设计技术已经成为了高级硬件工程师和 IC 逻辑设计工程师必备的技能之一。

FPGA 系统设计如何入门

FPGA 是基于硬件可编程的器件，设计方法与 CPU 和 DSP 的设计方法有本质上的区别。设计者不仅需要掌握硬件设计语言，还要具备硬件的概念和调试的经验才能设计出高质量的 FPGA 系统。

因此，FPGA 系统设计最好的入门方法就是实践，不能仅仅停留在看书和软件仿真的阶段。初学者在学习的过程中如果能够结合 FPGA 的开发板，将自己的设计在 FPGA 硬件系统上真正的运行起来，可以达到更好的学习效果。

怎样全面掌握 FPGA 设计的相关技术

FPGA 系统设计涉及的相关技术很多，包括了 FPGA 的结构原理，电路硬件设计与调试，硬件描述语言 (HDL)，开发工具 EDA 软件，仿真验证技术，以及 FPGA 和其他处理器的互联接口技术等。

在短时间内全面掌握以上技术对于初学者来说是非常困难的。因此，本书在典型实例讲解了 FPGA 系统设计涉及的相关技术中最重要的部分。

本书的特点

(1) 针对性更强。针对初学者的习惯来设计章节的顺序和内容。本书内容来自培训教材，经过课堂实践，学员的普遍性困惑在书中都做了相应的解答。

(2) 全面而系统。本书全面、系统地介绍了 FPGA 系统设计涉及的各种软硬件技术，尤其详细介绍了 FPGA 硬件电路设计的技术细节。

(3) 实用价值高。本书的全部实例来自于工程实践和实际产品，可直接在 FPGA 硬件平台上运行。本书同时提供了配套的硬件开发板供读者选用。

(4) 适合面广。本书同时介绍了 Altera 公司和 Xilinx 公司的开发软件，同时提供 QuartusII 和 ISE 两个版本的典型实例。读者无论是正在开发 Altera 芯片还是 Xilinx 芯片，都可以阅读本书，了解 FPGA 设计的通用技术。

本书的内容

本书共 13 章。

第 1 章主要讲解 FPGA 系统设计的背景知识，通过这一章的学习，读者可以掌握 FPGA 系统设计技术的现状和发展趋势，FPGA 的典型应用领域，以及产品开发过程中如何选择 FPGA 芯片等。

第 2 章以红色飓风 II 代 FPGA 开发平台为例，介绍了如何构建 FPGA 的硬件系统。这一章详细介绍了 FPGA 相关的电路设计方法，读者可以参考并设计出 FPGA 的硬件最小系统。

第 3 章和第 4 章主要介绍了硬件描述语言 (Verilog HDL) 的基本语法和常用语句，详细介绍了 Verilog HDL 语言中最常用和最关键的语法。读者掌握这些语法后，就可以开始独立进行 HDL 代码的设计工作了。

第 5 章和第 6 章分别介绍了 Altera 公司的开发工具软件 QuartusII 和 Xilinx 公司的开发工具软件 ISE。读者可以根据自己的需要进行选择。

第 7 章主要介绍了 FPGA 的仿真验证原理和方法。以 ModelSim 软件为工具，演示了如何对一个 FPGA 设计进行仿真和验证。

第 8 章主要介绍了基于 FPGA 的软核嵌入式系统设计技术。本章以 Altera 公司的 NIOSII 软核为例进行讲解，因此 Xilinx 的用户可以跳过本章。本书的配套光盘上也提供了 Xilinx 公司的 MicroBlaze 软核的相关设计资料和典型实例。

第 9 章主要介绍了 FPGA 设计的 3 个基本原则、3 种常用技巧以及 3 类常用的 IP 核。掌握这些原则和技巧并能够运用到 FPGA 设计中，可以大大提高设计的效率。

第 10 章主要介绍了 FPGA 在接口设计领域的典型应用，如 RS-232C、USB 2.0、字符 LCD 以及 VGA 等。

第 11 章主要介绍了 FPGA+DSP 的双芯片协同处理系统的设计方法和调试技巧。这种新型的系统设计构架正在得到越来越广泛的应用，因此本书对这种技术进行了深入讲解。

第 12 章和第 13 章分别介绍了数字变焦系统和 PCI 数据采集系统，这两个系统都已经批

量生产，因此具有较强的工程意义。这两个案例综合运用了前面各章的内容，读者可以通过该实例综合掌握前面各章的内容。

实例相关资料下载

读者可以登录 <http://www.fpgadev.com> 下载书中所有实例的完整工程文件、设计源文件和相关的说明文件，而且每个典型实例提供了 Altera 的 QuartusII 和 Xilinx 的 ISE 两个版本，供读者选择使用。

每个实例目录包括了完整的项目文件、源代码以及编译后的结果文件。读者可以直接在 QuartusII 或者 ISE 软件下面打开工程，直接运行。

本书的全部实例都可以运行在红色飓风 II 代 FPGA 开发平台上。如果读者能够结合开发板学习，可以达到最佳的学习效果。

红色飓风 II 代开发平台的相关信息请见：<http://www.fpgadev.com>。

适合的读者

- FPGA 系统设计初学者。
- 大专院校通信工程、电子工程、计算机、微电子和半导体相关专业师生。
- 硬件系统工程师和 IC 设计工程师。

参与本书编写的人员

本书内容来源于北京华清远见嵌入式培训中心 (<http://www.farsight.com.cn>) 的 FPGA 培训课程资料，书中相关参考资料下载请见 <http://www.farsight.com.cn/download>。

本书由姚远、李辰执笔。由于时间仓促，加之水平有限，书中不足之处在所难免，敬请读者批评指正。本书责任编辑的联系方式是 huangyan@ptpress.com.cn，欢迎来信交流。

编者

2010年5月

目 录

第 1 章 FPGA 系统设计基础 1

1.1 FPGA 技术的发展历史和

动向 1

1.1.1 FPGA 技术的发展历史 1

1.1.2 FPGA 技术的发展动向 2

1.2 FPGA 的典型应用领域 3

1.2.1 数据采集和接口逻辑
领域 3

1.2.2 高性能数字信号处理
领域 4

1.2.3 其他应用领域 4

1.3 FPGA 的工艺结构 5

1.3.1 基于 SRAM 结构的
FPGA 5

1.3.2 基于反融丝结构的
FPGA 5

1.3.3 基于 Flash 结构的
FPGA 6

1.4 主流的 FPGA 芯片厂家及其 代表产品 6

1.4.1 Xilinx 公司的代表产品 6

1.4.2 Altera 公司的代表产品 7

1.5 工程项目中 FPGA 芯片选择 策略和原则 8

1.5.1 尽量选择成熟的产品
系列 8

1.5.2 尽量选择兼容性好的

封装 8

1.5.3 尽量选择一个公司的

产品 9

第 2 章 从零开始设计 FPGA 最小系统 ... 10

2.1 FPGA 最小系统的概念 10

2.2 FPGA 最小系统电路分析 11

2.2.1 FPGA 管脚设计 11

2.2.2 下载配置与调试接口
电路设计 14

2.2.3 高速 SDRAM 存储器接口
电路设计 14

2.2.4 异步 SRAM (ASRAM)
存储器接口电路设计 16

2.2.5 Flash 存储器接口电路
设计 16

2.2.6 开关、按键与发光 LED
电路设计 18

2.2.7 VGA 接口电路设计 19

2.2.8 PS/2 鼠标及键盘接口
电路设计 20

2.2.9 RS-232 串口 20

2.2.10 字符型液晶显示器接口
电路设计 21

2.2.11 USB 2.0 接口芯片
CY7C68013 电路设计 ... 22

2.2.12	电源电路设计	25	3.4.3	关键词	87
2.2.13	复位电路设计	26	3.5	Verilog HDL 语言的条件语句	88
2.2.14	时钟电路设计	27	3.5.1	if 语句	88
2.3	FPGA 硬件系统的设计技巧	27	3.5.2	case 语句	90
2.3.1	管脚兼容性设计	27	3.5.3	其他条件语句	93
2.3.2	根据电路布局来分配管脚功能	28	3.6	Verilog HDL 语言的其他常用语句	93
2.3.3	预留测试点	28	3.6.1	循环语句	93
2.4	FPGA 硬件系统的调试方法	28	3.6.2	结构说明语句	96
2.5	典型实例 1: 在 Altera 的 FPGA 开发板上运行第一个 FPGA 程序	29	3.7	Verilog HDL 语言实现组合逻辑电路	98
2.5.1	实例的内容及目标	29	3.7.1	assign 语句实现组合逻辑	98
2.5.2	平台简介	30	3.7.2	always 块实现组合逻辑	99
2.5.3	实例详解	30	3.8	Verilog HDL 语言实现时序逻辑电路	100
2.5.4	小结	40	3.9	Verilog HDL 语言与 C 语言的区别与联系	101
2.6	典型实例 2: 在 Xilinx 的 FPGA 开发板上运行第一个 FPGA 程序	40	3.10	Verilog HDL 程序设计经验	104
2.6.1	实例的内容及目标	40	3.11	典型实例 3: 数字跑表	105
2.6.2	平台简介	41	3.11.1	实例的内容及目标	105
2.6.3	实例详解	41	3.11.2	原理简介	105
2.6.4	小结	65	3.11.3	代码分析	106
3.11.4	参考设计	107	3.12	典型实例 4: PS/2 接口控制	107
第 3 章	硬件描述语言 Verilog HDL 基础	66	3.12.1	实例的内容及目标	107
3.1	Verilog HDL 语言简介	66	3.12.2	原理简介	108
3.1.1	Verilog HDL 的历史和进展	66	3.12.3	实例详解	109
3.1.2	VHDL 和 Verilog HDL 语言对比	67	3.12.4	参考设计	109
3.2	Verilog HDL 程序基本结构	67	3.13	典型实例 5: 交通灯控制器	109
3.2.1	Verilog HDL 程序入门	68	3.13.1	实例的内容及目标	109
3.2.2	模块的框架	70	3.13.2	原理简介	110
3.3	Verilog HDL 语言的数据类型和运算符	71	3.13.3	代码分析	111
3.3.1	常用数据类型	71	3.13.4	参考设计	113
3.3.2	常用运算符	76	第 4 章	硬件描述语言 Verilog HDL 设计进阶	114
3.4	Verilog HDL 语言的赋值语句和块语句	83	4.1	task 和 function 说明语句的区别	114
3.4.1	非阻塞赋值和阻塞赋值	83			
3.4.2	块语句	84			

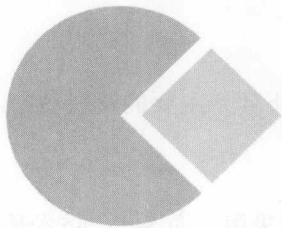
4.2 Verilog HDL 高级语法结构—— 任务 (TASK)	115	5.5.1 创建工程	149
4.3 Verilog HDL 高级语法结构—— 任务 (function)	116	5.5.2 添加设计文件	151
4.4 有限状态机的设计原理及其 代码风格	118	5.6 编译及仿真工程	153
4.4.1 有限状态机 (FSM) 设计 原理	118	5.6.1 编译	153
4.4.2 FSM 设计实例	123	5.6.2 仿真	154
4.4.3 设计可综合状态机的 指导原则	124	5.7 约束及配置工程	156
4.5 逻辑综合的原则以及可综合的 代码设计风格	125	5.7.1 器件选择	157
4.5.1 always 块语言指导原则	125	5.7.2 管脚分配	158
4.5.2 可综合风格的 Verilog HDL 模块实例	126	5.7.3 时序约束	158
4.6 典型实例 6: 状态机应用	132	5.7.4 配置工程	159
4.6.1 实例的内容及目标	132	5.8 LogicLock 逻辑锁定工具使用 技巧	160
4.6.2 实例详解	133	5.8.1 逻辑锁定方法学	160
4.6.3 参考设计	140	5.8.2 逻辑锁定优势	160
4.7 典型实例 7: 自动转换量程 频率计控制器	140	5.8.3 逻辑锁定参数设置	161
4.7.1 实例内容及目标	140	5.8.4 逻辑锁定流程	161
4.7.2 原理简介	140	5.9 SignalTap II 在线逻辑分析仪的 使用方法	162
4.7.3 代码分析	141	5.9.1 SignalTap II 介绍	162
4.7.4 参考设计	143	5.9.2 使用 SignalTap II 操作 流程	163
4.8 典型实例 8: 使用函数实现 简单的处理器	143	5.9.3 设置触发器	163
4.8.1 实例的内容及目标	143	5.10 典型实例 9: SignalTap II 功能 演示	164
4.8.2 原理简介	143	5.10.1 实例的内容及目标	164
4.8.3 代码分析	143	5.10.2 实例详解	165
		5.10.3 小结	177
		5.11 典型实例 10: LogicLock 功能 演示	177
		5.11.1 实例的内容及目标	177
		5.11.2 实例详解	178
第 5 章 FPGA 设计开发软件 Quartus II 的使用技巧	145	第 6 章 FPGA 设计开发软件 ISE 使用技巧	184
5.1 Quartus II 软件简介	145	6.1 ISE 软件简介	184
5.2 Quartus II 软件新特性	145	6.1.1 ISE 软件简介	184
5.3 Quartus II 软件的安装与启动	147	6.1.2 ISE 7.1i 特点	185
5.4 Quartus II 软件设计流程	148	6.2 ISE 软件的安装与启动	186
5.5 创建工程设计文件	149	6.2.1 ISE 软件的安装	186

6.2.2 ISE 软件的启动	188	7.2 ModelSim 仿真工具简介	243
6.3 ISE 软件的设计流程	189	7.2.1 标题栏	244
6.4 创建设计工程	193	7.2.2 菜单栏	244
6.5 编译与仿真设计工程	197	7.2.3 工具栏	249
6.6 增量式设计 (Incremental Design) 技巧	203	7.2.4 工作区	249
6.6.1 增量式设计的必要性	203	7.2.5 状态栏	249
6.6.2 增量设计流程	204	7.3 ModelSim 的仿真流程	249
6.6.3 小结	207	7.3.1 ModelSim 的安装	249
6.7 片上逻辑分析仪 (ChipScope Pro) 使用技巧	208	7.3.2 使用 ModelSim 进行功能仿真	250
6.7.1 ChipScope Pro 概述	208	7.4 功能仿真和时序仿真的区别和实现方法	253
6.7.2 ChipScope Pro 设计流程	209	7.4.1 功能仿真	253
6.7.3 ChipScope Pro Core Inserter 简介	209	7.4.2 时序仿真	256
6.7.4 ChipScope Pro Analyzer 简介	215	7.5 仿真测试文件 (Testbench) 的设计方法	257
6.7.5 小结	219	7.5.1 测试文件的用途	257
6.8 典型实例 11: ChipScope 功能演示	220	7.5.2 测试文件设计方法	258
6.8.1 实例的内容及目标	220	7.5.3 测试常用语句	260
6.8.2 基于 ChipScope Pro Core Generator 的实现流程	221	7.6 典型实例 13: SDRAM 读写控制的实现与 Modelsim 仿真	262
6.8.3 基于 ChipScope Pro Core Inserter 的实现流程	230	7.6.1 实例的内容及目标	262
6.8.4 小结	234	7.6.2 SDRAM 简介	262
6.9 典型实例 12: 增量式设计 (Incremental Design) 演示	234	7.6.3 SDRAM 控制器的 Modelsim 仿真	265
6.9.1 实例的内容及目标	234	7.6.4 小结	267
6.9.2 实例详解	234	第 8 章 基于 FPGA 的片上可编程系统 (SOPC) 设计	269
6.9.3 小结	240	8.1 基于 FPGA 的 SOPC 系统组成原理和典型方案	269
第 7 章 FPGA 系统设计的仿真实验	241	8.1.1 SoC 及 SOPC 简介	269
7.1 FPGA 设计仿真实验的原理和方法	241	8.1.2 Xilinx 的 SOPC 解决方案	270
7.1.1 FPGA 设计仿真实验	241	8.2 Altera 公司的 NIOS II 解决方案	271
7.1.2 FPGA 设计仿真的切入点	241	8.2.1 NIOS 的主要特点	271
		8.2.2 NIOS 技术实现方式	272
		8.3 基于 NIOS II 的开发设计流程	274

8.3.1	硬件开发流程	274	10.1.2	灵活的可编程逻辑	327
8.3.2	软件开发流程	281	10.1.3	支持多种电平接口标准	327
8.4	典型实例 14: 基于 NIOSII 处理器的“Hello LED”程序设计	287	10.2	利用 FPGA 实现 RS-232C 串行接口	327
8.4.1	实例的内容及目标	287	10.2.1	RS-232C 接口概述	327
8.4.2	实例详解	287	10.2.2	RS-232C 接口的电气标准	327
8.4.3	小结	295	10.2.3	RS-232C 的通信协议	328
8.5	典型实例 15: 基于 NIOSII 处理器的数字钟设计	296	10.2.4	RS-232C 接口的典型应用	329
8.5.1	实例的内容及目标	296	10.3	利用 FPGA 实现 USB 2.0 通信接口	329
8.5.2	软件规划	296	10.3.1	USB 2.0 接口的实现方式	329
8.5.3	实例步骤	296	10.3.2	FX2 接口芯片的 Slave FIFO 传输模式	330
8.5.4	小结	297	10.3.3	Slave FIFO 模式的典型操作时序	331
第 9 章	FPGA 系统设计原则和技巧	298	10.3.4	FX2 的固件程序设计	333
9.1	FPGA 系统设计的 3 个基本原则	298	10.3.5	USB 2.0 接口的典型应用	334
9.1.1	面积与速度的平衡互换原则	298	10.4	利用 FPGA 实现常用显示接口 (Display Interface)	334
9.1.2	硬件可实现原则	300	10.4.1	七段数码显示接口的设计与实现	334
9.1.3	同步设计原则	301	10.4.2	字符型 LCD 显示接口的设计与实现	336
9.2	FPGA 系统设计的 3 种常用技巧	302	10.4.3	VGA 显示接口的设计与实现	337
9.2.1	乒乓操作技巧	302	10.5	利用 FPGA 实现 A/D、D/A 转换器接口	339
9.2.2	串并/并串转换技巧	303	10.5.1	A/D、D/A 转换器接口的特点	339
9.2.3	硬件流水线操作技巧	304	10.5.2	A/D、D/A 转换器接口的实现方法	339
9.3	FPGA 系统设计的 3 种常用 IP 模块	305	10.5.3	AD/DA 转换器接口的注意事项	340
9.3.1	片上存储器的使用方法	305			
9.3.2	锁相环的使用方法	316			
9.3.3	高速串行收发器的使用方法	321			
第 10 章	利用 FPGA 实现外设通信接口	326			
10.1	FPGA 在外设接口实现方面的优势	326			
10.1.1	充足的用户 I/O 资源	326			

10.6	典型实例 16: RS-232C (UART)	平台的优势和适用领域	359
	接口的设计与实现	11.1.1 基于 FPGA 的信号处理	
10.6.1	实例内容及目标	系统的优点	359
10.6.2	实例详解	11.1.2 基于 DSP 的信号处理	
10.6.3	BlockRAM 的实现	系统的优点	360
	方法	11.1.3 基于 FPGA+DSP 的信号处	
10.6.4	FPGA 代码的设计	理系统优势和应用领域	360
	实现	11.2 基于 FPGA+DSP 的协同处理平台	
10.6.5	波特率的设定	的设计流程	361
10.6.6	ModelSim 仿真验证	11.2.1 FPGA 与 DSP 的功能	
10.6.7	小结	划分	361
10.7	典型实例 17: USB 2.0 接口的	11.2.2 FPGA+DSP 的系统设计	
	设计与实现	流程	362
10.7.1	实例的内容及目标	11.3 FPGA 与 DSP 的通信接口	
10.7.2	USB 接口通信实战步骤	设计	363
10.7.3	USB 接口通信实例	11.3.1 基于 TMS320C64x 系列的	
	结果	EMIF 接口设计	363
10.7.4	FPGA 代码的设计	11.3.2 基于 TMS320 系列 DSP 的	
	实现	主机接口 (HPI) 设计	369
10.7.5	小结	11.4 FPGA+DSP 协同平台的调试	
10.8	典型实例 18: 字符 LCD	技巧和注意事项	371
	接口的设计与实现	11.4.1 FPGA 和 DSP 的隔离	
10.8.1	实例的内容及目标	调试技术	371
10.8.2	字符 LCD 接口实例	11.4.2 FPGA 测试点的设计	
	内容	372	
10.8.3	FPGA 代码设计实现	11.4.3 借助 FPGA 的内部逻辑	
10.8.4	ModelSim 仿真验证	分析仪来辅助调试	372
10.8.5	小结	11.5 典型实例 20: FPGA 片上硬件	
10.9	典型实例 19: VGA 接口的	乘法器的使用	372
	设计与实现	11.5.1 实例的内容及目标	
10.9.1	实例内容及目标	372	
10.9.2	VGA 接口实例内容	11.5.2 硬件乘法 IP 的使用	
10.9.3	FPGA 代码设计实现	方法	373
10.9.4	ModelSim 仿真验证	11.5.3 小结	
10.9.5	小结	377	
第 11 章	FPGA 与 DSP 协同处理	11.6 典型实例 21: 整数 DCT 变换的	
	系统设计	设计与实现	378
11.1	基于 FPGA+DSP 协同处理	11.6.1 实例的内容及目标	
		378	
		11.6.2 整数 DCT 变换的	
		原理	378
		11.6.3 实例步骤	
		380	
		11.6.4 小结	
		382	

第 12 章 数字图像倍焦系统设计与实现综合实例383	
12.1 设计需求分析与芯片选型383	
12.1.1 需求分析383	
12.1.2 芯片选型384	
12.2 系统工作原理分析385	
12.3 系统原理框图387	
12.4 FPGA 内部结构设计388	
12.4.1 FPGA 内部结构框图388	
12.4.2 各个模块功能描述389	
12.5 系统硬件配置方案389	
12.5.1 FPGA 的配置389	
12.5.2 Video Decoder (SAA7113H) 的配置390	
12.5.3 Video Encoder (SAA7128) 的配置390	
12.5.4 印刷电路板 (PCB) 设计391	
12.6 FPGA 在其他视频和图像处理系统中的应用392	
第 13 章 高速 PCI 信号采集卡设计与实现综合实例393	
13.1 设计需求分析与功能定义393	
13.1.1 信号与信号采集系统393	
13.1.2 设计需求分析394	
13.1.3 选型及功能定义395	
13.2 系统工作原理分析395	
13.2.1 数据总线396	
13.2.2 控制总线397	
13.2.3 地址总线397	
13.2.4 信号采集系统控制机制398	
13.3 PCI 接口芯片 PCI9054 与 FPGA 的接口设计398	
13.3.1 PCI9054 的特性 398	
13.3.2 PCI9054 工作模式 399	
13.3.3 PCI 设备空间配置 400	
13.3.4 PCI9054 与 FPGA 接口设计 401	
13.4 PCI 卡的驱动程序设计 405	
13.4.1 WDM 驱动程序模型 405	
13.4.2 设备和驱动程序的层次结构 406	
13.4.3 PCI 设备驱动程序例程 407	
13.5 主机应用程序和驱动程序的接口设计 412	
13.5.1 驱动程序例程的封装 412	
13.5.2 PLX API 函数 413	
13.5.3 API 函数调用 416	
13.5.4 PCI9054 驱动程序安装 416	
13.6 FPGA 内部结构设计 416	
13.6.1 构框图 416	
13.6.2 设计方法 417	
13.7 硬件系统实现 418	
13.7.1 FPGA 配置 418	
13.7.2 PCI9054 配置 419	
13.7.3 PCI9054 PCI 总线连接规范 420	
13.7.4 电源系统 421	
13.8 样机的调试方法和技巧 422	
13.8.1 没有 PCI 控制的跑马灯 422	
13.8.2 基于 PCI 控制的跑马灯 423	
13.9 产品稳定性和可靠性测试 425	
13.10 产品定型和设计文档备案 426	



第 1 章 FPGA 系统设计基础

本章目标

- 了解 FPGA 技术的发展历史和动向
- 了解 FPGA 的典型应用领域
- 了解主流的 FPGA 芯片厂家及其代表产品
- 了解工程项目中 FPGA 芯片选择策略和原则

1.1 FPGA 技术的发展历史和动向

1.1.1 FPGA 技术的发展历史

纵观数字集成电路的发展历史，经历了从电子管、晶体管、小规模集成电路到大规模以及超大规模集成电路等不同的阶段。发展到现在，主要有 3 类电子器件：存储器、处理器和逻辑器件。

存储器保存随机信息（电子数据表或数据库的内容）；处理器执行软件指令，以便完成各种任务（运行数据处理程序或视频游戏）；而逻辑器件可以提供特殊功能（器件之间的通信和系统必须执行的其他所有功能）。

逻辑器件分成两类：

- ① 固定的或定制的。
- ② 可编程的或可变的。

其中，固定的或定制的逻辑器件通常称为专用芯片（ASIC）。ASIC 是为了满足特定的用途而设计的芯片，例如 MP3 解码芯片等。其优点是通过固化的逻辑功能和大规模的工业化生产，降低了芯片的成本，同时提高了产品的可靠性。随着集成度的提高，ASIC 的物理尺寸也在不断的缩小。

但是，ASIC 设计的周期很长，而且投资大，风险高。一旦设计结束后，功能就固化了，以后的升级改版困难比较大。电子产品的市场正在逐渐细分，为了满足快速产品开发，产生了现场可编程逻辑器件（FPGA）。

自 1984 年 Xilinx 公司推出了第一片现场可编程逻辑器件（FPGA）至今，FPGA 已经历了 20 几年的快速发展历程。特别是近几年来，更是发展迅速。FPGA 的逻辑规模已经从最初

的 1000 个可用门发展到现在的 1000 万个可用门。

FPGA 技术之所以具有巨大的市场吸引力,其根本原因在于:FPGA 不仅可以解决电子系统小型化、低功耗、高可靠性等问题,而且其开发周期短、投入少,芯片价格不断下降。FPGA 正在越来越多地取代传统上 ASIC,特别是在小批量、个性化的产品市场方面。

1.1.2 FPGA 技术的发展动向

随着芯片设计工艺水平的不断提高,FPGA 技术呈现出了以下 4 个主要的发展动向。

1. 基于 FPGA 的嵌入式系统 (SoPC) 技术正在成熟

System on Chip (SoC) 技术在芯片设计领域被越来越广泛地采用,而 SoPC 技术是 SoC 技术在可编程器件领域的应用。这种技术的核心是在 FPGA 芯片内部构建处理器。Xilinx 公司主要提供基于 Power PC 的硬核解决方案,而 Altera 提供的是基于 NIOSII 的软核解决方案。

Altera 公司为 NIOSII 软核处理器提供了完整的软硬件解决方案,可以让客户短时间完成 SoPC 系统的构建和调试工作。

如图 1.1 所示,是 Altera Stratix III FPGA 基于 NIOS II 解决方案的内部结构图。

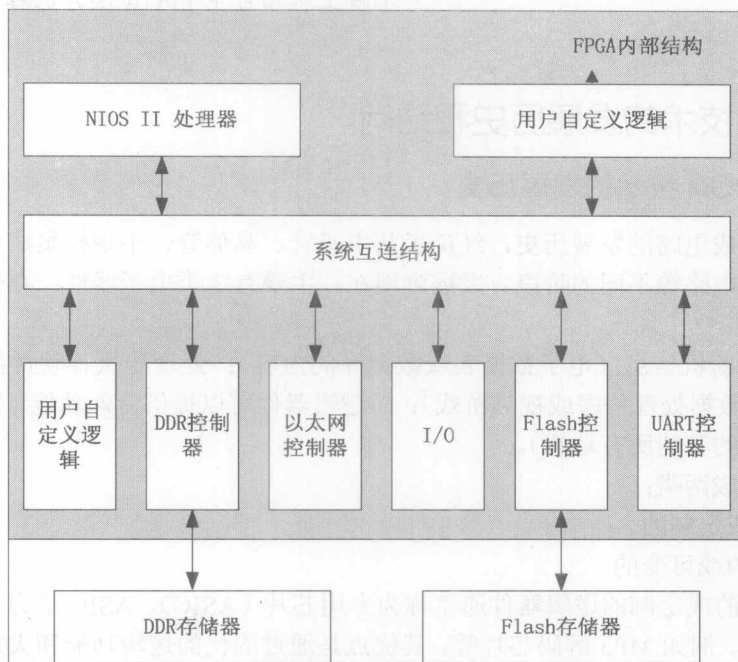


图 1.1 NIOS II 解决方案

2. FPGA 芯片向高性能、高密度、低压和低功耗的方向发展

随着芯片生产工艺不断提高,FPGA 芯片的性能和密度都在不断提高。早期的 FPGA 主要是完成接口逻辑设计,比如 AD/DA 和 DSP 的粘合逻辑。现在的 FPGA 正在成为电路的核心部件,完成关键功能。

在高性能计算和高吞吐量 I/O 应用方面,FPGA 已经取代了专用的 DSP 芯片,成为最佳

的实现方案。因此,高性能和高密度也成为衡量 FPGA 芯片厂家设计能力的重要指标。

随着 FPGA 性能和密度的提高,功耗也逐渐成为了 FPGA 应用的瓶颈。虽然 FPGA 比 DSP 等处理器的功耗低,但是要明显高于专用芯片(ASIC)的功耗。FPGA 的厂家也在采用各种新工艺和技术来降低 FPGA 的功耗,并且已经取得了明显的效果。

例如,Altera 公司的 StratixIII 系列 FPGA 的功耗比上一代产品 StratixII 系列降低了 50% 以上。

3. 基于 IP 库的设计方法

未来的 FPGA 芯片密度不断提高,传统的基于 HDL 的代码设计方法很难满足超大规模 FPGA 的设计需要。随着专业的 IP 库设计公司不断增多,商业化的 IP 库种类会越来越全面,支持的 FPGA 器件也会越来越广泛。

作为 FPGA 的设计者,主要的工作是找到适合项目需要的 IP 库资源,然后将这些 IP 整合起来,完成顶层模块设计。由于商业的 IP 库都是通过验证的,因此整个项目的仿真和验证工作主要就是验证 IP 库的接口逻辑设计的正确性。

目前,由于国内的知识产权保护的相关法律法规还不尽完善,基于 IP 库的设计方法还没有得到广泛应用。但是随着 FPGA 密度不断提高和 IP 库的价格逐渐趋于合理化,这种设计方法将会成为主流的 FPGA 设计技术。

4. FPGA 的动态可重构技术

FPGA 动态重构技术主要是指对于特定结构的 FPGA 芯片,在一定的控制逻辑的驱动下,对芯片的全部或部分逻辑资源实现高速的功能变换,从而实现硬件的时分复用,节省逻辑资源。

由于密度不断提高,FPGA 能实现的功能也越来越复杂。FPGA 全部逻辑配置一次的时间也变长了,降低了系统的实时性。局部逻辑的配置功能可以实现“按需动态重构”,大大提高了配置的效率。

动态可重构的 FPGA 可以在系统运行中对电路功能进行动态配置,实现硬件的时分复用,节省了资源,主要适用于以下两个系统设计。

① 最新通信系统。

FPGA 的动态重构特性可以适应不同体制和不同标准的通信要求,满足软件无线电技术的发展和第三代(3G)和第四代(4G)移动通信系统的需要。

② 重构计算机: FPGA 具有并行处理能力和动态配置能力,可自动改变硬件来适应正在运行的程序,产生了基于这种软硬件环境的全新概念的计算机。

1.2 FPGA 的典型应用领域

1.2.1 数据采集和接口逻辑领域

1. FPGA 在数据采集领域的应用

由于自然界的信号大部分是模拟信号,因此一般的信号处理系统中都要包括数据的采集

功能。通常的实现方法是利用 A/D 转换器将模拟信号转换为数字信号后, 送给处理器, 比如利用单片机 (MCU) 或者数字信号处理器 (DSP) 进行运算和处理。

对于低速的 A/D 和 D/A 转换器, 可以采用标准的 SPI 接口来与 MCU 或者 DSP 通信。但是, 高速的 A/D 和 D/A 转换芯片, 比如视频 Decoder 或者 Encoder, 不能与通用的 MCU 或者 DSP 直接接口。在这种场合下, FPGA 可以完成数据采集的粘合逻辑功能。

2. FPGA 在逻辑接口领域的应用

在实际的产品设计中, 很多情况下需要与 PC 机进行数据通信。比如, 将采集到的数据送给 PC 机处理, 或者将处理后的结果传给 PC 机进行显示等。PC 机与外部系统通信的接口比较丰富, 如 ISA、PCI、PCI Express、PS/2、USB 等。

传统的设计中往往需要专用的接口芯片, 比如 PCI 接口芯片。如果需要的接口比较多, 就需要较多的外围芯片, 体积、功耗都比较大。采用 FPGA 的方案后, 接口逻辑都可以在 FPGA 内部来实现了, 大大简化了外围电路的设计。

在现代电子产品设计中, 存储器得到了广泛的应用, 例如 SDRAM、SRAM、Flash 等。这些存储器都有各自的特点和用途, 合理地选择存储器类型可以实现产品的最佳性价比。由于 FPGA 的功能可以完全自己设计, 因此可以实现各种存储接口的控制器。

3. FPGA 在电平接口领域的应用

除了 TTL、COMS 接口电平之外, LVDS、HSTL、GTL/GTL+、SSTL 等新的电平标准逐渐被很多电子产品采用。比如, 液晶屏驱动接口一般都是 LVDS 接口, 数字 I/O 一般是 LVTTTL 电平, DDR SDRAM 电平一般是 HSTL 的。

在这样的混合电平环境里面, 如果用传统的电平转换器件实现接口会导致电路复杂性提高。利用 FPGA 支持多电平共存的特性, 可以大大简化设计方案, 降低设计风险。

1.2.2 高性能数字信号处理领域

无线通信、软件无线电、高清影像编辑和处理等领域, 对信号处理所需要的计算量提出了极高的要求。传统的解决方案一般是采用多片 DSP 并联构成多处理器系统来满足需求。

但是多处理器系统带来的主要问题是设计复杂度和系统功耗都大幅度提升, 系统稳定性受到影响。FPGA 支持并行计算, 而且密度和性能都在不断提高, 已经可以在很多领域替代传统的多 DSP 解决方案。

例如, 实现高清视频编码算法 H.264。采用 TI 公司 1GHz 主频的 DSP 芯片需要 4 颗芯片, 而采用 Altera 的 StratixII EP2S130 芯片只需要一颗就可以完成相同的任务。FPGA 的实现流程和 ASIC 芯片的前端设计相似, 有利于导入芯片的后端设计。

1.2.3 其他应用领域

除了上面一些应用领域外, FPGA 在其他领域同样具有广泛的应用。

- (1) 汽车电子领域, 如网关控制器/车用 PC 机、远程信息处理系统。
- (2) 军事领域, 如安全通信、雷达和声纳、电子战。
- (3) 测试和测量领域, 如通信测试和监测、半导体自动测试设备、通用仪表。