

基于

FPGA 的 嵌入式系统设计

<Xilinx Edition>

徐欣 于红旗 等编著
易凡 卢启中



内附光盘



机械工业出版社
China Machine Press

基于 FPGA 的嵌入式系统设计

Xilinx EDITION

徐 欣 于红旗 等编著
易 凡 卢启中



机械工业出版社

随着可编程逻辑技术的不断进步和创新, FPGA 在嵌入式系统中发挥着越来越重要的作用, 已被广泛应用于通信、航天、医疗、工控等领域。本书从嵌入式系统设计角度出发, 基于全球最大的可编程逻辑器件生产厂商 Xilinx 公司的系列 FPGA 产品, 系统、全面地介绍了 Xilinx 公司最新系列 FPGA 的内部结构、设计流程和开发工具; 详细介绍了 IP 资源复用理念和 IP Core 的开发方法; 重点介绍了 RISC 处理器内核、DSP 算法在 FPGA 中的实现原理和开发流程, 向嵌入式设计工程师推荐一种基于可编程片上系统 (SOPC) 的嵌入式系统设计新理念。在随书附赠的光盘中, 收录了 Xilinx 公司设计工具——ISE WebPACK 和 MXE 5.8c, 文中设计、分析、仿真或综合时涉及的相关文件和工具, 以及附录中所有实验教程的完整工程目录、源代码和使用说明, 利于读者在阅读本书的过程中提高实际应用能力。

本书立足于工程实践, 结合作者多年嵌入式系统和可编程逻辑器件的开发和教学经验, 选用成熟的火龙刀系列 FPGA 评估系统作为例程实验平台。对于各类从事嵌入式系统设计的科研人员和应用技术工程师, 本书是一本不可多得的针对可编程片上系统 (SOPC) 设计原理和开发的参考书。本书比较适合作为 Xilinx 产品开发应用方面的专业教材, 也可作为有志于从事专用数字电路设计的高等院校高年级本科生和研究生的参考书。

图书在版编目 (CIP) 数据

基于 FPGA 的嵌入式系统设计/徐欣等编著.

-北京: 机械工业出版社, 2004.9

ISBN 7-111-15337-5

I. 基… II. 徐… III. 微型计算机-系统设计

IV. TP360.21

中国版本图书馆 CIP 数据核字 (2004) 第 098993 号

机械工业出版社 (北京市百万庄大街 22 号 邮政编码 100037)

策划编辑: 姜淑欣 责任编辑: 王金航 版式设计: 谭奕丽

北京蓝海印刷有限公司印刷·新华书店北京发行所发行

2005 年 1 月第 1 版第 1 次印刷

787mm×1092mm 1/16·34.75 印张·856 千字

0001-5000 册

定价: 49.00 元 (含 1CD)

凡购本图书, 如有缺页、倒页、脱页, 由本社发行部调换

本社购书热线电话: (010) 68993821、88379646

封面无防伪标均为盗版

前 言

1. 写作背景

嵌入式系统是一门综合性的学科，它涉及传统软硬件设计技术的方方面面，同时还融入了多任务实时操作系统的新内容。目前，国内的嵌入式系统开发一般采用嵌入式处理器加 ASIC 器件的方式，当然这种设计方式也是国际上目前最为流行的设计方式。

在目前国内流行的嵌入式设计方法中，CPLD/FPGA 等可编程逻辑器件在绝大部分场合下仅作为嵌入式处理器的胶合逻辑或只是实现诸如数据通道等简单逻辑；随着可编程逻辑器件工艺和开发工具日新月异地发展，目前的可编程逻辑器件，尤其是 Xilinx 和 Altera 公司纷纷推出高性价比的新一代超大规模可编程逻辑器件，在嵌入式产品设计中已能够实现更多的功能并发挥着越来越重要的作用。Xilinx 公司最新推出的 Spartan-III 系列 FPGA，其百万逻辑门密度产品的批量价格仅为 10 美元左右，而在这样的逻辑门密度下，可以嵌入十个以上的 32 位 RISC 处理器内核，或者同时实现多处理器并行和 DSP 算法的 FPGA，Altera 公司的 Cyclone 系列高性价比 FPGA 也是同类产品的典型代表。

随着可编程逻辑密度的不断提高，我们可以在 FPGA 中嵌入更多的数字设计，在 FPGA 中以硬件描述语言的形式嵌入越来越多的标准数字单元，如 PCI 控制器、以太网控制器等，使得 FPGA 和 ASIC 在很多应用领域的分割界限被打破，FPGA 在很多对上市时间和设计灵活性有苛刻需求的场合取代了 ASIC 的应用。但是 ASIC 设计的复杂性使得 FPGA 应用工程师无法在短时间内完成产品的功能设计，知识产权复用技术因此也开始出现在 FPGA 工程应用领域。在可编程逻辑器件厂商开始提供 IP Core 参考设计和出现如 OpenCore 等开放源代码 IP Core 开发团体外，商业化的 IP Core 产品和专业公司也开始不断涌现，而且这种趋势正是迎合日趋复杂和日益普遍的 FPGA 开发应用需求。

随着嵌入式处理器、专用数字器件和 DSP 算法以 IP Core 的形式嵌入到 FPGA 中，以单片 FPGA 完成以往整个嵌入式系统数字部分设计的设想已经成为现实。目前，国外学术和工程领域对于可编程 SOC 的研究已经开展得如火如荼，而由 Xilinx、Altera 等可编程逻辑器件厂商倡导的可编程片上系统设计（System on a Programmable Chip, SOPC）技术已在通信、工业控制等很多领域得到实际应用。

目前，国内在 SOPC 方面的研究和应用也已展开，但这方面的中文资料却较少，特别是工程应用往往受制于学术研究和相关的参考资料。我们的课题组多年来一直从事可编程逻辑设计和嵌入式系统设计方面的学术研究和工程应用工作，在我们使用了 Xilinx 公司最新的开发工具：嵌入式开发套件（Embedded Development Kit, EDK），DSP 的 FPGA 实现开发工具（System Generator）和片内逻辑分析仪（ChipScope ILA Pro）等专门为 SOPC 优化设计的工具后，就有一种强烈的欲望，要将这些设计理念和设计工具推荐给工程设计领域的工程师和广大有志于 SOPC 设计的同仁们。



2. 内容安排

本书的内容安排如下:

- 第1章为 Xilinx 公司的可编程逻辑器件综述, 包含了技术参数列表和选型指南。
- 第2~3章针对 Xilinx 公司 Virtex 系列 Platform FPGA 和 Spartan 系列高性价比 FPGA 作了详细介绍。
- 第4章介绍了基于 FPGA 的嵌入式系统设计基础, 推荐了两款 FPGA 评估系统。
- 第5~6章介绍了硬件描述语言 VHDL、VerilogHDL 以及基于 C/C++ 的 FPGA 设计方法。
- 第7~11章详细介绍了 Xilinx FPGA 设计方法和设计工具的使用。
- 第12章为读者推荐了一种基于 Internet 硬件可重构逻辑的设计方法, 这种设计理念将对嵌入式产品开发、维护和调试带来新的思路。
- 第13章详细阐述了 IP 资源复用理念和 IP Core 设计方法, 包括 HDL 编码风格。
- 第14~15章重点介绍了 8 位处理器内核 PicoBlaze、32 位处理器内核 MicorBlaze 和 PowerPC 405 的原理、结构特点和开发方法以及 Xilinx 嵌入式开发套件 EDK 的使用和实例流程。
- 第16章讨论了 DSP 算法在 FPGA 中的硬件实现方法, 介绍了 Xilinx 专用 DSP 开发工具 System Generator 的使用和设计实例。
- 本书所附光盘中提供了 Xilinx 相关产品的使用说明, 部分章节中设计、分析、仿真或者综合时涉及的相关文件和工具, 以及附录中关于火龙刀系列 FPGA 评估系统的详细资料以及本部分实验教程的全部源代码和相关文件; 并且提供了 Xilinx 公司的免费设计工具: ISE WebPACK 和 MXE 5.8c (即 ModelSim XE II 5.8c Starter Kit)。

本书第1~4章、第14~16章由徐欣编写, 第5~13章由于红旗编写, 易凡和卢启中负责全书的审定。

3. 致谢

首先感谢国防科技大学·湖南大学嵌入式系统联合研究中心和国防科技大学综合电子战实验室的广大教员和研究生对于本书的大力支持, 其中冯道旺、徐海源、龙霞飞、张望、吴佳、伍涛、喻小虎、刘凯等也参与了本书的编写工作, 每一章节和实例教程也渗透了他们辛勤的汗水和不倦的工作。同时也感谢所有参与原始资料翻译工作的学生们。

感谢湖南大学黎福海教授为本书使用的 FPGA 评估系统提供的宝贵的意见, 以及为本书的出版起到的很大的促进作用。感谢湖南省电子设计竞赛专家组对本书提供的指导性意见和鼓励。

感谢我们亲密的战友孙广富、薛以辉、王建东为本书使用的 FPGA 评估系统设计作出的不懈努力, 并为本书提供了很多宝贵的指导性意见。

感谢 Xilinx (赛灵思) 公司亚太区高端产品市场经理梁晓明先生提供本书所附光盘软件的支持, 以及对编写所提供的支援和鼓励。

感谢亚太科汇 AK 先生和夏祖泉小姐帮助我们申请大学计划援助和提供大量的软件和开



发工具支持，感谢科汇的 FAE 工程师们提供的无私技术支援。

本书第一作者在此感谢妻子樊一华在本书出版过程中给予的理解和支持。

本书第二作者在此感谢妻子田苗苗在本书出版过程中给予的理解和支持。

感谢机械工业出版社为我们出版了本书，没有出版社同仁尤其是姜淑欣编辑认真的工作，本书也无法与读者见面。

4. 再版目标

由于时间和研究进度的限制，本书还没有涉及 MicroBlaze 处理器内核的 RTOS 移植，不能不说是本书的缺憾。

作者希望在进一步研究和实际工程应用后再版本书，修正本书第一版本中出现的错误与不足，吸收第一版读者的意见；增加 RTOS 移植和应用部分内容；提供在百万门级 FPGA 评估系统上的设计实例；加强 DSP 算法的 FPGA 实现部分内容；并引入更多的工程应用实例。

5. 反馈

本书编者力求把本书内容写好，但由于能力所限，本书中纰漏在所难免，对于本书的任何意见和建议，欢迎读者联系本书编辑 (jsx@cmpbook.com)，也可以在网站 www.sopc.cn 上的 BBS 里留言，本书作者在此表示诚挚的感谢。

编者
于国防科大



目 录

前言

第 1 章 Xilinx 现场可编程逻辑器件综述	1
1.1 可编程逻辑器件的基本特征	1
1.2 SOPC 成为 FPGA 的发展趋势	2
1.2.1 Actel 公司的 VariCore 内核和 ProASIC Plus FPGA	2
1.2.2 Altera 公司的 Excalibur 嵌入式处理器方案和 Stratix 器件	3
1.2.3 Atmel 公司的 FPSLIC 系列产品	4
1.2.4 Lattice 公司的 FPSC 和 ORCA FPGA	4
1.2.5 QuickLogic 公司的 QuickMIPS	4
1.2.6 Xilinx 公司的 Virtex-II Pro FPGA	5
1.2.7 总结	5
1.3 可编程逻辑器件的基本开发流程	6
1.4 Xilinx 可编程逻辑器件产品选型	7
1.4.1 Virtex 系列 FPGA	7
1.4.2 Spartan 系列 FPGA	7
1.4.3 扩展温度范围汽车 IQ 产品	11
1.4.4 军品及宇航级产品	14
1.4.5 配置存储器解决方案	15
1.5 Xilinx 创新平台 FPGA 架构 ASMBL	16
1.6 Xilinx 可编程逻辑器件网络资源	17
1.7 小结	18
第 2 章 Virtex™ 系列高端 Platform FPGA	19
2.1 Virtex-II 系列 Platform FPGA 产品	19
2.1.1 概述与订购信息	19
2.1.2 结构与功能描述	21
2.2 Virtex-II Pro/Pro X 系列 Platform FPGA 产品	28
2.2.1 概述与订购信息	28
2.2.2 结构与功能描述	30
2.3 小结	33
第 3 章 Spartan™ 系列高性价比 FPGA 产品	34
3.1 Spartan-II 系列 FPGA 产品	34
3.1.1 概述与订购信息	34



3.1.2 结构与功能描述	36
3.2 Spartan-IIIE 系列 FPGA 产品	37
3.2.1 概述与订购信息	37
3.2.2 结构与功能描述	38
3.3 Spartan-III 系列 FPGA 产品	42
3.3.1 概述与订购信息	42
3.3.2 结构与功能描述	44
3.4 小结	51
第 4 章 基于 FPGA 的嵌入式系统概述	52
4.1 嵌入式系统概述	52
4.1.1 嵌入式系统的定义	52
4.1.2 嵌入式系统的基本特征	52
4.1.3 嵌入式系统的基本组成	54
4.1.4 嵌入式处理器的分类	56
4.1.5 实时多任务操作系统 RTOS	58
4.2 FPGA 在嵌入式系统中的地位和作用	58
4.2.1 在 FPGA 中实现 RISC 处理器内核	59
4.2.2 在 FPGA 中实现高速 DSP 算法	59
4.2.3 在 FPGA 中嵌入式 ASIC 模块	59
4.2.4 在 FPGA 中实现数字 IP Core	60
4.3 基于 FPGA 的嵌入式系统设计方法	60
4.3.1 可编程片上系统设计框架	60
4.3.2 微处理器内核开发工具 EDK	61
4.3.3 DSP 算法硬件实现工具 System Generator	62
4.4 火龙刀系列 FPGA 评估系统设计	62
4.4.1 火龙刀 I 代 Spartan-II 评估系统	62
4.4.2 火龙刀 II 代 Spartan-IIIE 评估系统	71
4.5 小结	76
第 5 章 VHDL & Verilog HDL 简明教程	77
5.1 数字系统的表示方法和硬件描述语言	77
5.2 VHDL 语言	79
5.2.1 VHDL 语言的基本结构	79
5.2.2 结构体的子结构描述	92
5.2.3 标识符 (Identifier)	93
5.2.4 数据对象 (Data Object)	94
5.2.5 数据类型	95
5.2.6 属性	98
5.2.7 运算符	99



基于FPGA的嵌入式系统设计

5.2.8	VHDL 的语句和结构体	99
5.2.9	用 VHDL 设计基本的逻辑电路	102
5.3	Verilog HDL 语言	120
5.3.1	Verilog HDL 语言的发展及其特点	120
5.3.2	Verilog HDL 的结构	121
5.3.3	空白符和注释	126
5.3.4	标识符	127
5.3.5	常量的数据类型	127
5.3.6	变量常用的数据类型	129
5.3.7	运算符	130
5.3.8	语句	133
5.3.9	块语句	136
5.3.10	编译预处理	137
5.3.11	基本逻辑电路的设计	139
5.4	小结	151
第 6 章	使用 C/C++ 开发 FPGA 介绍	152
6.1	为什么要使用 C/C++ 开发 FPGA	152
6.1.1	传统的设计流程	152
6.1.2	SystemC 及其系统设计流程	153
6.1.3	Handel-C 及其开发 FPGA 的设计流程	153
6.2	如何对 SystemC 进行功能仿真	154
6.3	使用 Visual C++ 和 ModelSim 来进行仿真	155
6.3.1	设计准备	155
6.3.2	编译生产 SystemC 的库文件	156
6.3.3	新建 SystemC 工程	156
6.3.4	为设计添加源文件	158
6.3.5	为工程添加 systemc.lib 文件	160
6.3.6	编译执行	160
6.3.7	使用 ModelSim 转换波形格式	161
6.3.8	使用 ModelSim 查看波形	161
6.4	使用 Borland C++ 和 SystemC_Win 进行功能仿真	161
6.4.1	设计准备	161
6.4.2	打开一个设计	163
6.4.3	编译、执行	163
6.5	小结	164
第 7 章	Xilinx ISE 6.1i 简明教程	165
7.1	设计准备	165
7.1.1	ISE 6.1i 软件的安装	165



7.1.2	ISE 软件的运行及 ModelSim 的配置	166
7.2	用 VHDL 语言设计输入	167
7.2.1	创建一个新工程	167
7.2.2	创建一个计数器源文件	169
7.2.3	利用计数器模板向导生成设计	170
7.3	仿真	172
7.3.1	创建 Testbench 波形源文件	172
7.3.2	设置输入仿真波形	173
7.3.3	生成预期输出响应	174
7.3.4	调用 ModelSim 进行仿真简介	174
7.3.5	调用 ModelSim 进行行为仿真 (Simulate Behavioral Model)	176
7.3.6	转换后仿真 (Simulate Post-Translate VHDL Model)	177
7.3.7	调用 ModelSim 进行映射后仿真 (Simulate Post-Map VHDL Model)	177
7.3.8	布局布线后的仿真 (Simulate Post-Place&Route VHDL Model)	178
7.3.9	使用 ModelSim 5.7SE 对设计进行仿真	178
7.4	用原理图设计输入	179
7.4.1	将创建的 VHDL 模块生成一个原理图符号	179
7.4.2	创建一个顶层原理图 (Top-Level Schematic)	180
7.4.3	例化 VHDL 模块 (Instantiating VHDL Module)	180
7.4.4	添加原理图连线 (Wiring the Schematic)	181
7.4.5	为连线添加网络名	182
7.4.6	为总线添加网络名	182
7.4.7	添加输入/输出引脚标记 (I/O Markers)	183
7.4.8	查看原理图生成的 VHDL 文件	184
7.4.9	查看综合后的 RTL 级电路图	185
7.5	对顶层文件进行仿真	186
7.6	设计实现	189
7.7	用 EDIF 设计输入	191
7.7.1	设计输入	191
7.7.2	设计实现	192
7.8	用 Verilog HDL 设计输入	193
7.9	下载配置	194
7.10	小结	199
第 8 章	使用 ModelSim 进行设计仿真	200
8.1	ModelSim 的 License 及其加密方法	200
8.2	菜单和工具栏介绍	202
8.2.1	标题栏	202
8.2.2	菜单栏	203



8.2.3	工具栏	208
8.2.4	状态栏	208
8.3	使用图形界面对设计进行仿真	208
8.3.1	创建新工程	209
8.3.2	为工程添加源文件	210
8.3.3	编译	211
8.3.4	装载设计	211
8.3.5	查看仿真波形窗口	212
8.3.6	设置信号驱动	212
8.3.7	开始仿真	213
8.3.8	仿真结果分析	213
8.4	使用命令行方式对设计进行仿真	213
8.5	TestBench 及其在仿真中的应用	216
8.5.1	TestBench 概述	216
8.5.2	使用 TestBench 对设计进行仿真	216
8.6	TEXTIO 在仿真中的应用	218
8.7	ModelSim 的配置	225
8.8	ModelSim 中常用的几个命令	231
8.9	小结	233
第 9 章	基于 Synplify/Synplify Pro 的 FPGA 高级综合设计	234
9.1	Synplify/Synplify Pro 简介	234
9.2	以一个例子来熟悉基本操作	235
9.2.1	新建一个工程	235
9.2.2	为工程添加设计文件	236
9.2.3	编译、综合	237
9.2.4	查看综合后的 RTL 视图	238
9.2.5	查看技术视图	238
9.2.6	添加约束文件	239
9.2.7	查看综合报告	240
9.2.8	修改约束文件	241
9.2.9	在 Synplify /Synplify Pro 中调用 ISE	242
9.3	在 ISE 6.1 中调用 Synplify Pro	243
9.3.1	对 ISE 进行设置	243
9.3.2	以一个例子来说明	244
9.4	使用 ISE 6.1 自带的综合工具 XST 对设计进行综合	245
9.5	小结	248
第 10 章	FPGA 设计技巧——ISE 高级设计工具	249
10.1	Floorplanner 概述	249



10.2 使用 Floorplanner 手动布局逻辑块.....	249
10.2.1 新建 ISE 工程.....	250
10.2.2 为工程添加源文件.....	250
10.2.3 综合设计.....	252
10.2.4 查看综合后的 RTL 视图.....	252
10.2.5 进入 Floorplanner.....	254
10.2.6 使用 Floorplanner 进行资源分配.....	254
10.2.7 为设计添加约束文件.....	256
10.2.8 查看布局布线结果.....	256
10.2.9 总结.....	256
10.3 使用 FPGA Editor 进行手动布局布线.....	257
10.3.1 打开工程.....	257
10.3.2 进入 FPGA Editor.....	257
10.3.3 导入设计.....	258
10.3.4 自动布线.....	259
10.3.5 手动布线.....	260
10.3.6 规则检查.....	260
10.3.7 查看布局结果.....	261
10.3.8 总结.....	261
10.4 使用 XPower 分析设计的功耗.....	261
10.4.1 打开工程.....	261
10.4.2 新建仿真波形文件.....	262
10.4.3 仿真生成 VCD 文件.....	262
10.4.4 进行功耗分析.....	263
10.4.5 查看电池使用时间.....	265
10.4.6 参数设置.....	265
10.5 小结.....	266
第 11 章 片内逻辑分析仪工具——ChipScope Pro	267
11.1 ChipScope Pro 概述.....	267
11.2 使用 ChipScope Pro 内核生成器.....	268
11.2.1 使用 ISE 6.1 建立一个新工程.....	268
11.2.2 打开 ChipScope Pro Core Generator.....	269
11.2.3 产生 ICON 核.....	269
11.2.4 产生 ILA 内核.....	271
11.2.5 在 VHDL 设计流程中使用内核.....	272
11.3 使用 ChipScope Pro 内核插入器.....	273
11.3.1 使用 ISE 6.1 新建工程.....	274
11.3.2 Core Inserter 参数设置.....	274



11.3.3	使用 Xilinx ISE 将插入的核整个实现流程	278
11.4	使用 ChipScope Pro 分析器	279
11.4.1	启动边界扫描链	279
11.4.2	配置目标器件	280
11.4.3	设置触发条件	281
11.4.4	运行并观察波形	283
11.5	小结	284
第 12 章	基于 Internet 的可重构逻辑 技术及应用	285
12.1	系统高级配置环境简介	285
12.2	System ACE 的实现	286
12.2.1	System ACE CF	286
12.2.2	System ACE MPM	287
12.2.3	System ACE SC	288
12.3	基于嵌入式处理器的配置技术	289
12.4	基于 Internet 的可重构逻辑实现	293
12.4.1	IRL 的基本组成	294
12.4.2	基于 8 位单片机的 IRL 实现	295
12.4.3	基于 32 位微处理器的 IRL 实现	297
12.4.4	比较与应用	299
12.5	PAVE	299
12.6	小结	301
第 13 章	IP 资源复用与 IP Core 开发	302
13.1	IP 资源复用理念与 IP Core 概述	302
13.1.1	IP 资源概述	302
13.1.2	IP 技术的形成	302
13.1.3	IP 开发的现状	303
13.1.4	IP 的发展面临的挑战	304
13.1.5	IP Core 设计	304
13.2	HDL 编码风格与编码指导	306
13.2.1	编写代码前的准备工作	306
13.2.2	HDL 编码风格	307
13.2.3	HDL 编码指导	309
13.2.4	Verilog 编码指导原则	310
13.2.5	VHDL 编码指导原则	310
13.2.6	了解适合综合工具的代码风格	311
13.3	Xilinx IP Core 打包工具 (IP Capture)	311
13.3.1	IP Capture 工具概述	311
13.3.2	IP Capture 用户界面	312



13.3.3	IP Capture 输出文件	316
13.4	Xilinx IP Core 生成工具 (Core Generator)	317
13.4.1	CORE Generator 综述	317
13.4.2	IP 核生成器用户界面	317
13.4.3	使用 CORE Generator 生成 IP 核的方法	320
13.5	Xilinx IP Core 更新工具 (Updates Installer)	323
13.5.1	Updates Installer 概述	323
13.5.2	安装包的定义	323
13.5.3	设置用户计算机环境	324
13.5.4	代理设置	324
13.5.5	浏览器的路径	325
13.5.6	用户注册	325
13.5.7	IP 升级包所需的输入	325
13.5.8	使用 GUI 安装 IP Core	326
13.5.9	安装所选择的封装包	326
13.5.10	运行 Get Models	327
13.6	基于 IP Core 的设计实例	327
13.6.1	实例概述	327
13.6.2	IIR 数字滤波器的运算结构	327
13.6.3	数字滤波器参数设计	328
13.6.4	IIR 滤波器的硬件设计	328
13.7	小结	339
第 14 章	PicoBlaze 处理器 IP Core 开发与应用	341
14.1	PicoBlaze 处理器概述	341
14.2	基于 Virtex-II 系列器件应用的 PicoBlaze 处理器	341
14.2.1	PicoBlaze 处理器功能及结构分析	341
14.2.2	PicoBlaze 的特性设置	344
14.2.3	PicoBlaze 指令集	346
14.2.4	PicoBlaze 处理器的控制信号	359
14.2.5	PicoBlaze 汇编程序	360
14.2.6	程序语法	365
14.2.7	程序指令约束	366
14.2.8	汇编程序中的指令	367
14.2.9	与 KCPSM 代码兼容性	369
14.2.10	中断处理	370
14.2.11	CALL/RETURN 堆栈	373
14.2.12	对于比较操作的一些提示	373
14.3	基于 Virtex-E 和 Spartan-II/III 系列器件应用的 PicoBlaze 处理器	374



- 14.3.1 PicoBlaze 处理器功能及结构分析 374
- 14.3.2 PicoBlaze 的特性设置 377
- 14.3.3 PicoBlaze 指令集 377
- 14.3.4 PicoBlaze 处理器的控制信号 390
- 14.3.5 PicoBlaze 汇编程序 391
- 14.3.6 程序语法 393
- 14.3.7 程序指令 394
- 14.3.8 汇编程序中的指令 394
- 14.3.9 与 KCPSM2 代码兼容性 396
- 14.3.10 中断处理 397
- 14.3.11 CALL/RETURN 堆栈 400
- 14.3.12 PicoBlaze Macro 的应用 400
- 14.3.13 对于应用的一些小提示 402
- 14.4 基于 CPLD 系列器件应用的 PicoBlaze 处理器 413
 - 14.4.1 PicoBlaze 处理器功能及结构分析 413
 - 14.4.2 PicoBlaze 的特性设置 413
 - 14.4.3 PicoBlaze 指令集 414
 - 14.4.4 PicoBlaze 汇编程序 420
 - 14.4.5 程序语法 421
 - 14.4.6 汇编程序中的指令 422
 - 14.4.7 PicoBlaze Macro 的应用 422
- 14.5 小结 424
- 第 15 章 Xilinx SOPC 集成开发环境 EDK 425**
 - 15.1 EDK 概述 425
 - 15.1.1 系统要求 425
 - 15.1.2 软硬件平台支持 425
 - 15.1.3 集成 IP Core 支持 426
 - 15.2 系统描述文件 427
 - 15.2.1 MHS 文件 427
 - 15.2.2 MSS 文件 427
 - 15.2.3 MVS 文件 428
 - 15.3 EDK 开发流程 429
 - 15.3.1 XPS 介绍 429
 - 15.3.2 EDK 设计流程实例 431
 - 15.3.3 几种软硬件模式的说明 441
 - 15.3.4 几个需要用户修改的文件说明 442
 - 15.4 小结 443



第 16 章 DSP 开发工具 System Generator	444
16.1 System Generator 概述.....	444
16.1.1 系统需求	445
16.1.2 软件安装	445
16.1.3 利用 System Generator 进行系统级建模.....	445
16.2 Xilinx Blockset 介绍	452
16.2.1 Xilinx 模块定义	453
16.2.2 在 Simulink 模型中引用 Xilinx 模块.....	453
16.2.3 模块参数设置	453
16.3 Xilinx Blockset 库中的模块介绍	456
16.3.1 基本单元 (Basic Elements)	456
16.3.2 DSP 模块	460
16.3.3 数学运算模块	460
16.3.4 存储器模块	460
16.3.5 通信模块	461
16.3.6 数据类型模块	461
16.4 System Generator 工程设计流程及实现.....	461
16.4.1 利用辅助工具学习 System Generator.....	462
16.4.2 设计实现	463
16.4.3 基于 EDIF 的设计流程.....	464
16.4.4 设计仿真	464
16.4.5 约束文件	465
16.4.6 设计实例	467
16.5 小结	472
附录 实验指导	473
实验 1 数码管显示时钟	473
实验 2 音频信号发生器	481
实验 3 彩条信号显示	488
实验 4 从 SRAM 中读取数据并显示	497
实验 5 液晶模块显示字符串	506
实验 6 EDK 设计 MicroBlaze 定时器中断	512
实验 7 SystemGenerator 使用.....	516
实验 8 串行通信实验	519
实验 9 虚拟信号发生器与示波器实验.....	530
参考文献	535

第 1 章 Xilinx 现场可编程逻辑器件综述

1.1 可编程逻辑器件的基本特征

嵌入式系统行业是一个面向应用、技术密集、资金密集、高度分散、不可垄断的产业。随着嵌入式系统的各个领域应用需求的多样化，嵌入式设计技术和芯片技术也经历着一次又一次的革新。虽然 ASIC 的成本很低，但设计周期长，投入费用高，风险较大，而可编程逻辑器件（Programmable Logical Device, PLD）设计灵活，功能强大，尤其是高密度现场可编程逻辑阵列（Field Programmable Gate Array, FPGA）的设计性能已完全能够与 ASIC 媲美，而且由于 FPGA 的逐步普及，其性能价格比已足以与 ASIC 抗衡。因此，以 FPGA 为核心的可编程逻辑器件在嵌入式系统设计领域已占据着越来越重要的地位。

可编程逻辑器件是逻辑器件家族中发展最快的一类器件，这两个根本原因：其一，每个可编程逻辑器件具有愈来愈多的门电路，因而可完成本来要由许多个分立逻辑器件和存储芯片完成的功能。这种方法缩小了最终系统的尺寸，降低了最终系统的功耗和成本，提高了最终系统的性能和可靠性；其二，同样重要的是，只需花几秒或几分钟的时间，就可以在 workstation 或系统组装线上配置或重新配置这些器件。这种能力使设计人员有很大的灵活性，可以对最后一分钟的设计变动作出反应，并可以在实施之前根据想法做出原型，也可以满足因客户需求和竞争压力而导致的产品上市的最后期限。

与 ASIC 相比，可编程逻辑器件研制周期较短，先期 NRE 费用较低，也没有最少订购数量的限制，所有这一切都简化了库存管理。随着每个门电路成本的降低和每个器件中门电路数量的增加，可编程逻辑器件正在大举打入传统的门阵列领域，并已有少量的打入了标准单元 ASIC 的领域。系统设计师和制造商们刚刚开始探索和利用“系统可重编程性”的性能，其目的是为了在最终系统交付给用户使用后仍可纠正其错误和进行功能升级，或是为了实现“可重新配置计算”，即用固定数量的逻辑门电路来实现分时复用多功能。

正如各公司利用可编程逻辑器件的灵活性来使自己的产品不同于竞争对手的产品那样，半导体厂商们都已经开发出了独特的 PLD 和 FPGA，以达到性能、功耗、集成度和成本等多方面的具体目标。这种目标多样性也许是人们设计过程中所面临的最复杂问题，原因是在许多情况下，必须仔细分析每种可编程逻辑器件体系结构，然后才能挑选出满足自己需要的一种可编程逻辑体系结构。市场上一些领先厂商正在不断推进事实上的行业标准化，从而简化了这种挑选工作。

可编程逻辑器件厂商的硅片产品的技术领先程度和文档的全面程度，并不是决定可编程逻辑器件厂商成败的惟一因素，同样重要的是可编程逻辑器件公司内部开发的或由第三方提供的软件工具支持的深度和广度。

片上 RAM 和单芯片 ASIC/可编程逻辑混合器件的数量越来越多，以及可预测的摩尔定律集成度发展趋势，正在促进有效门电路数量的爆炸性增长。这些因素最终将使人们期盼已久

