

彭澄廉 主编

陈泽文 审校

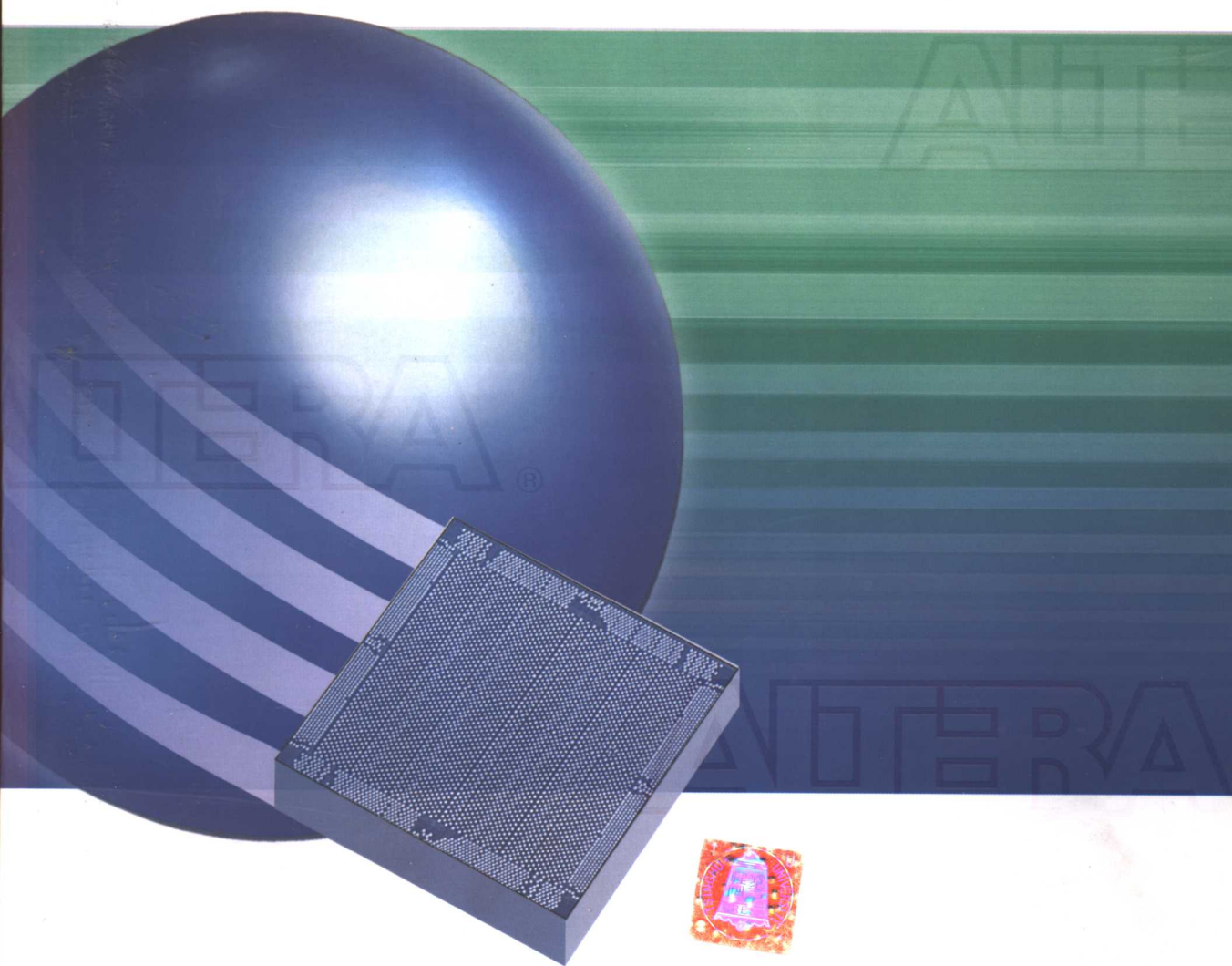
周博 邱卫东 陈燕 周学功 方茁 等编著



挑 战

SOC

——基于 NIOS 的 SOPC 设计与实践



清华大学出版社

挑战 SOC——基于 NIOS 的 SOPC 设计与实践

彭澄廉 主编

陈泽文 审校

周博 邱卫东 陈燕 周学功 方茁 等编著

清华大学出版社

北 京

内 容 简 介

本书主要介绍 Altera 公司的软核 CPU——Nios 和采用该 CPU 进行嵌入式系统设计的流程与方法。并以此为着眼点,介绍 Altera 的片上可编程系统 SOPC 的设计原理与实践技术,引领读者在低投入的情况下,较快地进入片上系统 SOC 的殿堂。

本书以 Altera 公司的技术资料为蓝本,由浅入深地阐述了 Nios 软核处理器的 CPU 结构、总线传输模式以及相关外设。同时也简要说明了软件的开发过程。既可作为学习 Nios 处理器的入门指导书,也可作为设计参考手册。

本书附录中刊载了 Altera 2003 年 Nios 嵌入式软核心处理器设计大赛中中国大陆赛区和中国台湾赛区的全部获奖作品,这一部分是由获奖者撰写的,读者可以从中获得有益的启示。

本书附带一张光盘,包含了 Altera 提供的 2004 年新版 QuartusII 4.0 软件和 Nios 处理器的开发包,读者可以藉此开始 SOPC 的旅程。

本书可供电子和通信等领域从事硬件设计和系统开发的工程技术人员和教师阅读参考,也可以作为相关专业研究生和高年级本科生的教材。

版权所有,翻印必究。举报电话:010-62782989 13901104297 13801310933

本书封面贴有清华大学出版社激光防伪标签,无标签者不得销售。

图书在版编目(CIP)数据

挑战 SOC——基于 NIOS 的 SOPC 设计与实践/彭澄廉主编;周博等编著.

—北京:清华大学出版社,2004.7

ISBN 7-302-08990-6

I. 挑… II. ①彭… ②周… III. 微处理器-系统设计 IV. TP332

中国版本图书馆 CIP 数据核字(2004)第 066367 号

出 版 者:清华大学出版社 地 址:北京清华大学学研大厦

<http://www.tup.com.cn> 邮 编:100084

社 总 机:010-62770175 客户服务:010-62776969

责任编辑:曾 刚

封面设计:秦 铭

版式设计:俞小红

印 刷 者:清华大学印刷厂

装 订 者:三河市新茂装订有限公司

发 行 者:新华书店总店北京发行所

开 本:185×260 印张:20.5 字数:448 千字

版 次:2004 年 7 月第 1 版 2004 年 7 月第 1 次印刷

书 号:ISBN 7-302-08990-6/TP·6359

印 数:1~5000

定 价:38.00 元(附光盘 1 张)

本书如存在文字不清、漏印以及缺页、倒页、脱页等印装质量问题,请与清华大学出版社出版部联系调换。联系电话:(010)62770175-3103 或(010)62795704

前 言

《Altera®SOPC 设计与实践》主要是由参加 Altera 2003 年 Nios 嵌入式软核心处理器设计大赛并获一等奖的研究生们编译成书的。2003 年 4 月~8 月, 复旦大学信息科学与工程学院的研究生周博等, 在指导教师的支持下, 参加了该项设计比赛。通过近 4 个月日夜勤奋的学习和设计实践, 他们以“汽车运行信息记录仪”的设计和演示, 荣获一等奖。这些研究生又在 Altera 公司的支持下, 从该项设计中抽取出包括多种输入输出和通信接口的设计平台, 并自制 40 套, 用于 2003 年 10 月 14 日举行的 Altera 全国大学教师会议, 作为培训 Nios 3.0 & SOPC 2.8 的教学实验板, 获广泛好评。适逢 Altera 公司试图将竞赛获奖成果与设计原理介绍汇集成书, Altera 的 Keith 先生给出了英文素材目录的初稿, 周博等根据学习和实践体会, 又做了调整和取舍。经过多位研究生翻译, 反复交换审校, 对内容的删简理顺, 教师审稿, 最后由 Altera 的工程师审阅后定稿。

Altera 的 Max+Plus II 曾是应用广泛、非常受欢迎的 PLD (可编程逻辑器件) 设计软件, 对于 FPGA 和 CPLD 这些 PLD 器件的应用推广、电路部件和嵌入式系统的设计, 以及高校有关专业的教学实习, 做出了非常重要的贡献。近年来, 为适应微电子技术及其应用的飞速发展, 尤其是 SOC (片上系统) 技术发展的需要, Altera 推出了新版本的 PLD 设计软件 Quartus II。它继承了 Max+Plus II 的所有优点, 是更加完善的 PLD 设计工具。它包括不同的设计输入手段 (原理图、VHDL 或 Verilog-HDL)、综合仿真工具、时限分析工具、功率评估工具、PLD 布局布线工具和产品验证工具。其中的设计工具 SOPC Builder 更是开创了嵌入式系统设计的新理念。

SOPC (片上可编程系统) 是以 PLD 取代 ASIC, 更加灵活、高效的 SOC 解决方案。SOPC 代表一种新的系统设计技术, 也是一种初级的软硬件协同设计技术。设计师使用 Altera 的 SOPC Builder, 在 PLD 器件上创建硬软件开发的基础平台, 也即用 SOPC Builder 创建软核 CPU——Nios 和参数化的接口总线 Avalon。在此基础上, 可以很快地将硬件系统 (包括处理器、存储器、外设接口和用户逻辑电路) 与常规软件集成在单一可编程芯片中。SOPC 是 PLD 和 ASIC 技术融合的结果, 它可编程的灵活性和 IP 设计的重用性, 易于保证产品的差异性和缩短面市时间; 它无需库存费用, 也无需一次性投片费用, 降低了投资风险。SOPC 相对于 SOC 的这个优势, 也反映了 FPGA 日益占有 ASIC 市场的趋势。不过, 由于 ASIC 电路仍有大批量产品的低价格优势, 目前已与 FPGA 形成共存、互补、相互渗透融合的局面, 并出现了结构化 ASIC 芯片研发和产品市场。

由于 SOPC 带来的灵活性, 使过去耗费、刻板的硬件设计变得像软件设计一样容易调试修改, 并出现了诸如可变结构、网上传送硬件、远程升级硬件、远程维修硬件等丰富多

彩的新概念、新技术。同时,由于 Altera 公司提供了多种廉价的 FPGA 器件和研发工具,并有较完善的大学支持计划,使得一般高校,甚至个人就有能力问津嵌入式系统的设计以及 SOC 技术领域。而在过去,我国的许多高校和研究机构,由于高昂的研究经费所限,不得不放弃包括系统结构技术在内的硬件技术研究。

嵌入式系统飞速发展,渗透于各个领域和行业。诸如宽带通信、移动通信、网络分布计算、网格计算、普适计算(Pervasive Computing)、可变结构等新概念、新技术,更使得嵌入式系统将涉足从微观的精细控制到宏观的超级计算。计算机软件的发展,也必然反映这个计算机结构变化的现实。软件公司已开始关注嵌入式软件的开发,我国许多高校的软件学院,也充分注意了嵌入式软件的研究教学,甚至直接关注嵌入式系统的设计和应用,设立了专门的实验室。这不能不说是有远见之举。

本书适合于电子工程、计算机、自动控制专业,以及对此类专业有兴趣的高年级本科生和研究生学习和自学 SOPC 设计技术。如果有电子工程或者 EDA 软件方面的经验,学习会更轻松。本书附录中刊有 Altera 2003 年 Nios 嵌入式软核心处理器设计大赛(中国大陆赛区和中国台湾赛区)的获奖作品说明,都是一些年轻的学生在不足 4 个月的时间里完成的作品。如果在数年前,这些作品本身就构成一些相当规模的项目。本书的读者只要认真、勤奋地学习好书中的内容、并勇于实践,就能具备挑战 SOC 的实力。

本书的编译者有周博、邱卫东、陈燕、周学功、方茁等。陈泽文高级工程师审阅了全稿。彭澄廉作为指导教师,组织了编写过程,参加了最后审校工作。

在本书编译过程中,得到了 Altera 公司各方面的大力支持。除了提供相关软件的光盘外,Altera 公司香港代表处的陈国裕经理和 Keith 先生为我们提供了英文素材,上海代表处的徐平波经理则在整个编译过程中给予了热心的帮助和督促。最后,Altera 全球副总裁 Ben Lee 先生还在百忙之中特意为本书做序,在此一并表示致敬和谢意。

本书因是限时多人合作编译成书,时间不容编译审校者进行充分的句句斟酌。错误疏漏之处,敬请读者批评指正。

彭澄廉

2004 年 3 月 31 日

Preface

The world of electronic system design is getting ever more complex. Both university students and professional engineers are faced with the challenges of having to do more with less time. System-on-a-Programmable Chip (SOPC) design methodology with Altera's Quartus II design tool is an ideal solution to meet these challenges. Nios, Altera's popular soft core RISC CPU, is at the heart of SOPC design.

Programmability is the future of electronic system design. Fixed function SOC's (System-on-a-Chips) will become less and less practical. The high development cost, long development time and lack of flexibility often cause SOC's to be obsolete soon after they come out of initial production. For example, even as 2G wireless technology was being installed, 2.5G basestations were being developed. Now before 2.5G is completely in place, 3G is being launched in many countries. Just as 3G is being launched, R&D teams are developing 4G hardware. The time period between each generation is getting shorter. There is simply not enough time to develop SOC's. Therefore, flexibility and programmability is the key. Hardware specifications are changing in real time. Some of Altera's customers are even remotely upgrading their hardware after they are installed in the field. This is possible because of SOPC technology.

Altera pioneered SOPC by putting the "P" in "SOC". P = Programmability = Flexibility. With programmability enabled by simple-to-use tools such as SOPC Builder in Quartus II, even novice designers can develop complex systems in a very short period of time by using a standard PC. This low cost but powerful methodology is perfect for students and universities. Through a number of Nios design contests conducted in China and throughout Asia Pacific recently, graduate students have developed and demonstrated very complex electromechanical systems in areas such as automotive electronics, robotics and industrial control. Some of these designs rival those of professional companies in the industry and I'm sure will be eventually commercially deployed.

More and more electronic systems, including consumer applications such as televisions, are depending on programmable logic. In the future, very few electronic equipment can exist without programmable logic. Thus, it will be a requirement in the near future, if not already so,

for every electronic engineering graduate to know how to design with PLDs and even SOPCs. Altera intends to continue to work closely with teaching professionals and universities to help students learn SOPC design. This book written by Professor 彭澄廉 and his students is a great example. On behalf of Altera, I would like to thank Professor 彭 for his tireless effort in publishing this textbook in order to help develop the next generation of great SOPC designers.

Ben Lee
Vice President & Managing Director
Altera International Limited
Apr. 28, 2004

目 录

第 1 章 绪论	1
1.1 概述	1
1.2 本书内容安排	3
1.3 设计步骤	4
1.4 支持 NIOS CPU 的 FPGA 型号	5
第 2 章 SOPC Builder 开发环境	9
2.1 创建 Quartus II 工程	9
2.2 创建 NIOS 系统模块	12
2.2.1 创建新的 .bdf	12
2.2.2 开始使用 SOPC Builder	13
2.2.3 系统主频	14
2.2.4 加入 CPU 和 IP 模块	14
2.2.5 指定基地址	25
2.2.6 配置 NIOS 系统	25
2.2.7 生成 NIOS32 并把它加入到设计中	27
2.2.8 把符号 (symbol) 加入到 BDF 文件中	28
2.2.9 加入引脚和基本单元	28
2.2.10 命名引脚	31
2.2.11 进行最后连接	31
2.3 编译 (Compilation)	32
2.3.1 创建编译器设置	32
2.3.2 指定器件系列 (Device Family) 与型号	33
2.3.3 为器件引脚分配信号	34
2.3.4 确认引脚分配	35
2.3.5 指定设备与编程设置	35
2.3.6 指定配置文件类型	36
2.3.7 编译用户设计	36
2.4 编程 (Programming)	38
2.4.1 配置 FPGA	38
2.4.2 在 NIOS 系统上运行软件	40
第 3 章 NIOS CPU 结构	43
3.1 NIOS 处理器概述	43

3.1.1	指令总线主端口	44
3.1.2	数据总线主端口	44
3.1.3	缓存	45
3.1.4	移位单元	45
3.1.5	乘法支持	45
3.1.6	中断支持	46
3.1.7	NIOS 片上调试模块	46
3.1.8	开发环境	47
3.2	内部寄存器	47
3.2.1	通用寄存器文件	47
3.2.2	K 寄存器	48
3.2.3	%r0 (%g0) 寄存器	49
3.2.4	程序计数器	49
3.2.5	控制寄存器	49
3.3	流水线	53
3.4	存储器组织	54
3.5	寻址方式	59
3.6	程序流程控制	62
第 4 章	Avalon 总线规范	64
4.1	概述	64
4.2	术语和概念	65
4.3	Avalon 总线传输	69
4.3.1	主端口接口与从端口接口	69
4.3.2	Avalon 总线时序	69
4.3.3	Avalon 总线信号	70
4.3.4	并发多主端口的 Avalon 总线需考虑的事项	71
4.4	Avalon 从端口传输	71
4.4.1	从传输的 Avalon 总线信号	72
4.4.2	Avalon 总线上的从端口读传输	73
4.4.3	在 Avalon 总线上的从端口写传输	78
4.5	Avalon 主端口传输	83
4.5.1	主传输的 Avalon 信号	84
4.5.2	Avalon 总线上的基本主端口读传输	85
4.5.3	Avalon 总线上的基本主端口写传输	86
4.6	高级 Avalon 总线传输	89
4.6.1	带延迟的 Avalon 读传输	89
4.6.2	流传输模式	96
4.6.3	Avalon 总线控制信号	102

4.7	片外设备与 Avalon 总线的接口	103
4.7.1	从传输的 Avalon 三态信号	104
4.7.2	无延迟的 Avalon 三态从端口读传输	105
4.7.3	带固定延迟的 Avalon 三态从端口读传输	106
4.7.4	Avalon 三态从端口写传输	108
4.8	Avalon 总线地址对齐方式	110
4.8.1	地址对齐概述	110
4.8.2	为 Avalon 外设选择地址对齐参数	111
4.8.3	静态地址对齐	112
4.8.4	动态地址对齐	115
4.9	连接到外部设备	117
第 5 章	软件设计流程和方法	119
5.1	软件开发包	119
5.1.1	软件开发流程	119
5.1.2	SDK 树目录简介	120
5.1.3	编译后代码结构	124
5.1.4	NIOS 库函数	125
5.1.5	高级 C 语言支持	128
5.2	软件开发常用命令	128
5.2.1	hexout2flash	129
5.2.2	nios_build	130
5.2.3	nios-console	131
5.2.4	nios-convert	132
5.2.5	nios-debug	133
5.2.6	nios-run	133
5.2.7	srec2flash	134
第 6 章	外设的组织与使用	136
6.1	SOPC Builder 与 PTF 文件	136
6.1.1	SOPC Builder 组成	136
6.1.2	PTF 文件的两种类型	137
6.1.3	SOPC Builder 设计流程	141
6.2	SOPC Builder 的外设	148
6.2.1	定时器	149
6.2.2	并行输入输出模块	153
6.2.3	异步收发器 (UART)	156
6.2.4	DMA 控制器	164
第 7 章	中断处理	170
7.1	中断处理过程	170

7.1.1	异常处理的基础知识	170
7.1.2	NIOS 中断处理基础	171
7.1.3	NIOS 异常处理步骤	174
7.2	中断处理的例子: UART 收发程序	176
7.2.1	在 NIOS 系统中实现 ISR	176
7.2.2	串口通信基础	177
7.2.3	实现 NIOS UART ISR	178
7.2.4	代码示例	180
第 8 章	调试手段	189
8.1	GERMS 监控程序	189
8.1.1	监控程序命令	189
8.1.2	GERMS 引导过程	190
8.1.3	采用 GERMS 从闪存中引导系统	190
8.2	FS2 公司的 System Analyzer	194
8.2.1	简介	194
8.2.2	设置调试器	196
8.2.3	使用调试器	198
8.2.4	调试命令列表	205
8.2.5	OCI 调试模块相关软件子程序	212
第 9 章	高级特性	214
9.1	用户自定义指令	214
9.1.1	用户自定义指令的概念	214
9.1.2	硬件接口	214
9.1.3	软件接口	219
9.1.4	加速效果	220
9.1.5	用户自定义指令实例	221
9.2	并发多主设备总线结构	226
9.2.1	传统总线结构	226
9.2.2	并发多主设备总线结构	227
9.3	NIOS 嵌入式处理器设计的模拟	232
9.3.1	软件配置	232
9.3.2	模拟设置	232
9.3.3	使用 ModelSim 进行模拟	237
9.3.4	分析模拟结果	240
9.3.5	增加或删除波形图信号	247
9.3.6	片外存储器模拟	248
第 10 章	新版 NIOS II 的特点	251
10.1	简介	251

10.2	处理器的实现	252
10.3	寄存器文件	252
10.4	算术逻辑单元	252
10.4.1	未实现的指令	253
10.4.2	用户自定义指令	253
10.5	异常和中断的控制	253
10.5.1	异常控制器	253
10.5.2	集成的中断控制器	253
10.6	存储器与 I/O 组织	254
10.6.1	指令和数据总线	255
10.6.2	缓存	256
10.6.3	地址映像	256
10.7	硬件辅助调试模块	257
附录 A	汇编指令及伪指令参考	258
	符号说明	258
	32 位指令集	259
	IFS 和 SKPS 指令的条件代码	263
	汇编语言伪指令与宏	263
附录 B	Altera 2003 年 NIOS 嵌入式软核心 处理器设计大赛获奖单位	265
	一、中国大陆部分	265
	二、中国台湾地区	299

第 1 章 绪 论

1.1 概 述

微电子技术与计算机技术的发展历史是一个不断创新的过程,这种创新包括原始创新、技术创新和应用创新等。每一项创新都能开拓出一个新的领域,带来新的巨大的市场,对我们的生产、生活方式产生重大的影响。

自集成电路发明以后,集成电路芯片的发展基本上遵循了 Intel 公司创始人之一的 Gordon E Moore 1965 年预言的摩尔定律,即每隔 3 年集成度增加 4 倍,特征尺寸缩小 1.4 倍。

在集成电路 (IC) 发展初期,电路设计都是从器件的物理版图设计入手。后来出现了集成电路单元库 (Cell-Lib),使得集成电路设计从器件级进入逻辑级,极大地推动了 IC 产业的发展。不过,集成电路只有安装在整机系统中才能发挥它的作用。IC 芯片是通过印刷电路板 (PCB) 等技术实现整机系统的。尽管 IC 的速度可以很高,功耗可以很小,但由于 PCB 板中 IC 芯片之间的连线延时、PCB 板可靠性及重量等因素的限制,整机系统的性能受到了很大的限制。随着系统向高速度、低功耗、低电压和多媒体、网络化、移动化的发展,系统对电路的要求越来越高。传统集成电路设计技术已无法满足性能日益提高的整机系统的要求。同时,由于 IC 设计与工艺技术水平提高,集成电路规模越来越大,复杂程度越来越高,整个系统已可以集成在一个芯片上。目前已经可以在一个芯片上集成 $10^8 \sim 10^9$ 个晶体管。SOC (System on Chip: 片上系统)就是在这种条件下应运而生的。

SOC 从整个系统的角度出发,把处理机制、模型算法、芯片结构、各层次电路,直至器件的设计紧密结合起来,在单个 (或少数几个) 芯片上完成整个系统的功能。SOC 的设计以 IP 核为基础,以分层次的硬件描述语言为系统功能和结构的主要描述手段,借助于以计算机为平台的 EDA 工具进行。研究表明,与 IC 组成的系统相比,由于 SOC 设计能够综合并全盘考虑整个系统的各种情况,因而可以在同样的工艺技术条件下,实现更高性能的系统指标。SOC 技术,也大大促进了软硬件协同设计及计算机系统设计自动化的发展。

SOPC (System on a Programmable Chip, 片上可编程系统)是 Altera®公司提出来的的一种灵活、高效的 SOC 解决方案。它将处理器、存储器、I/O 口、LVDS、CDR 等系统设计需要的功能模块集成到一个 PLD 器件上,构建成一个可编程的片上系统。它是可编程系统,具有灵活的设计方式,可裁减、可扩充、可升级,并具备软硬件在系统可编程的功能。可编程器件内,还具有小容量高速 RAM 资源。由于市场上有丰富的 IP Core 资源可供灵活选择,用户可以构成各种不同的系统,如单处理器、多处理器系统。有些可编程器件内还可以包含部分可编程模拟电路。除了系统使用的资源外,可编程器件内还具有足够的可编程

逻辑资源，用于实现其他的附加逻辑。

SOPC 是 PLD 和 ASIC 技术融合的结果，目前 $0.13\mu\text{m}$ 的 ASIC 产品制造价格仍然相当昂贵，而集成了硬核或软核 CPU、DSP、存储器、外围 I/O 及可编程逻辑的 SOPC 芯片在应用的灵活性和价格上有极大的优势。所以，有人认为 SOPC 代表了半导体产业未来发展的方向。

在很多对速度的要求不是很高的低端应用，Altera 将一个软核放入 PLD，这个软核就是 NIOS，它只占芯片内部很少的一部分逻辑单元，成本很低。同 ASIC 相比较，如果将处理器放到 ASIC 中，生产的每片芯片都要付给处理器厂商专利费。况且 ASIC 的 NRE（一次性投资）大，风险也大；NIOS 则没有这个问题。NIOS 的开发工具包价格很低，在速度要求高的高端应用中，如通信领域，软核的处理速度不够，Altera 就将硬核（ARM9）集成到 APEX 器件中，还集成入 RAM 和 RAM 控制器。同时 Altera 本身在 PLD 的结构方面也不断发展和创新，近期推出的 HardCopy Stratix 器件系列，是一个针对大容量设计的，从原型设计到批量生产的完整解决方案，试图成为 ASIC 的全面替代方案。

Altera 的 SOPC 开发工具，将软硬件的设计结合起来，提供给客户一个很好的开发环境。Altera 的 Max+Plus II 曾是应用广泛、非常受欢迎的 PLD（可编程逻辑器件）设计软件，它对于 FPGA 和 CPLD 这些 PLD 器件的应用推广、电路部件和嵌入式系统的设计，以及高校有关专业的教学实习，做出了非常重要的贡献。近年来，为适应微电子技术及其应用的飞速发展，尤其是 SOC（片上系统）技术发展的需要，Altera 推出了新版本的 PLD 设计软件 Quartus II。它继承了 Max+Plus II 的所有优点，是更加完善的 PLD 设计工具。它包括不同的设计输入手段（原理图、VHDL 或 Verilog-HDL）、综合仿真工具、时限分析工具、功率评估工具、PLD 布局布线工具和产品验证工具。其中的设计工具 SOPC Builder 更是开创了嵌入式系统设计的新理念。

SOPC Builder 是一个软件工具，它属于一种基于 IP 或者平台的设计方法。本书介绍的平台包括 Altera 的 NIOS 处理器、Avalon 总线，以及片内外存储器。利用 SOPC Builder，用户可以很方便地将处理器、存储器和其他外设模块连接起来，形成一个完整的系统。SOPC Builder 中已包含了 NIOS 处理器及其他一些常用的外设 IP 模块。用户也可以设计自己的外设 IP。

从用户的角度来看，SOPC Builder 是一个能够生成复杂硬件系统的工具。但从内部来看，SOPC Builder 包含两个主要部分：一个图形用户界面（GUI）和一个系统生成程序。SOPC Builder 图形用户界面提供管理 IP 模块、配置系统和报告错误等功能。用户通过图形用户界面设计系统。完成设计之后，单击 Generate，启动系统生成程序。系统生成程序通常从图形用户界面中启动。系统生成程序执行大量的功能，创建几乎所有的 SOPC Builder 输出文件（HDL 逻辑文件、C 程序的头文件和库文件、模拟文件，等等）。所以，SOPC Builder 可看作是一个以 IP 模块为输入，集成的系统为输出的工具。

SOPC Builder 会提示用户设置参数，并提示使用哪些可选的端口和外设。一旦向导生成了 NIOS 系统模块，则可以在设计文件中生成实例。一些常规硬软件接口、中断子程序等，都可被 SOPC Builder 包办，节约用户很多时间，而且更可靠。

NIOS CPU 是一种采用流水线技术、单指令流的 RISC 处理器，其大部分指令可以在一

个时钟周期内完成。NIOS 处理器又是一种软核 CPU，专门针对 Altera 的可编程逻辑器件及片上可编程系统的设计思想，做了相应优化。作为一种可配置的通用 RISC 处理器，它可以与用户自定义逻辑（user logic）结合构成 SOC 系统，并下载到 Altera 的可编程器件中去。32 位 NIOS 软核，结合外部闪存，以及大容量存储器，可构成一个功能强大的 32 位嵌入式处理器系统。NIOS 软核处理器分为 32 位和 16 位两种版本，本书主要介绍 32 位版本的 NIOS 3.0 嵌入式处理器。

在 Altera 的 NIOS 嵌入式处理器中，用户可以在 NIOS 指令系统中增加用户自定义指令，以增强其对强实时软件算法的处理能力。用户自定义指令可以通过单周期或多周期操作来完成复杂的处理任务。另外，增加的用户自定义指令同样可以访问存储器或 NIOS 系统外的逻辑。采用用户自定义指令，用户可以把一个复杂的标准指令序列，简化为一条用硬件实现的单个指令。这一特性可以用于多种情况，例如对数字信号处理（DSP）、数据包处理，以及计算密集型软件进行优化。

NIOS 采用的 Avalon 总线是一种相对简单的总线结构，主要用于连接片内处理器与外设，以构成片上可编程系统（SOPC）。它描述了主从构件间的端口连接关系，以及构件间通信的时序关系。Avalon 总线规范提供了各种选项，来剪裁总线信号和时序，以满足不同类型外设的需要。SOPC Builder 自动产生 Avalon 总线，Avalon 总线也包括许多特性和约定，用以支持 SOPC Builder 软件自动生成系统、总线和外设。

片上可编程系统（SOPC）的设计人员在嵌入式系统开发和调试时，还有其他辅助工具，诸如 ModelSim 模拟器及其他监控、调试工具等。

1.2 本书内容安排

本书共分为 10 章，以下是其他各章介绍，了解这些可以帮助读者取舍内容。

第 2 章以 Altera®公司的 NIOS 开发板为基础，介绍了一个典型 NIOS 系统的设计过程，由于其中绝大部分工作是在 PC 机上完成，更多情况下类似于一份软件操作说明，所以读者即使在没有 Altera®公司的开发板的情况下，仍然可以通过本章了解完整的开发过程。通过本章，初学者可以对 SOPC Builder 开发环境有一定的感性认识，并可在后面的学习中比照本章学习。而对于已经具有相当设计经验的读者，可以略过本章。

第 3 章介绍 3.0 版本的 32 位 NIOS CPU 内部结构。包括寄存器组织、流水线结构和汇编指令等。

第 4 章介绍 Avalon 总线规范。Avalon 总线是一种相对简单的片上总线结构，主要用于连接 CPU 和片内外设。但它仍然包含了相当多的高级特性，包括支持延迟读操作、流模式传输以及并发多主设备访问等。这些特性极大地扩展了 Avalon 总线的适应能力并能够有效提升系统性能。

第 5 章介绍 NIOS 处理器的软件设计流程与方法，包括 NIOS SDK Shell 环境及软件开发包（SDK）的结构。

第 6 章介绍 SOPC Builder 中的 IP 组织方法,并细致说明了 4 个常用的外设。其他外设的使用基本类似,用户可以参考 Altera®公司的相关文档。

第 7 章讲述 NIOS 处理器的中断处理机制及如何在软件中书写中断服务程序。特别是以 UART 为例,详细地介绍了中断向量的安装与中断的不同处理方法。

第 8 章介绍 NIOS 处理器的调试手段。Altera®公司并不提供 All in one 的编译和调试工具,只提供基本的监控程序或者对调试硬件支持。因此用户如果需要更优秀的调试工具,可以从第三方那里得到,如 Mentor 公司的 Codelab 就是一款比较优秀的调试软件。

第 9 章介绍 NIOS 处理器的高级特性,这些特点有效地提升了系统性能,并反映了片上总线的特点。

第 10 章介绍即将发布的 NIOS II CPU,用户可以通过介绍了解 NIOS 处理器的发展方向及一些最新的特点。

由于本书不仅作为 NIOS 处理器的入门教程,同时也作为 NIOS 处理器的参考手册。所以在章节顺序上是按照通常的处理器介绍模式安排的。但是由于 NIOS 软核处理器与通常的固核处理器之间存在较大区别,特别是 Altera 提供了大量与硬件相关的底层函数,隐藏了硬件细节,使得用户在不了解外设寄存器设置或 CPU 汇编指令的情况下,仍然可以进行系统设计工作。所以,读者在阅读本书时,特别是第一次接触 NIOS 处理器的读者,没有必要完全按照章节顺序来阅读,而是可以根据需要进行取舍。读者应重点掌握 SOPC 的设计概念、流程、方法和相关工具。

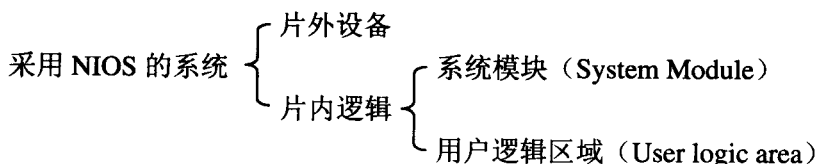
1.3 设计步骤

在采用 NIOS 处理器设计嵌入式系统时,通常会按照以下步骤:

- (1) 分析系统需求说明,包括功能需求和性能约束等;
- (2) 根据分析结果,选择片外外设或片内 IP (知识产权核)。除此以外,还要对选择的 IP 和外设进行初步性能评估,以保证能够满足系统需求;
- (3) 设定 IP 和系统参数;
- (4) 确定系统互联逻辑,分配 FPGA 的引脚等;
- (5) 结合 NIOS 提供的软件开发包进行软件开发。

图 1.1 是典型 NIOS 系统的框图。

图 1.1 的结构可以划分为:



片内逻辑是指实现在 FPGA 内部的电路设计,系统模块指的是由 SOPC Builder 自动生成的设计。SOPC Builder 会根据用户选择的 IP 生成相应的 HDL 描述文件(系统模块文件),

这些文件与用户逻辑区域内的设计描述文件一起由 Quartus 软件综合, 然后下载到 FPGA 内, 这样就构成了系统的硬件基础。

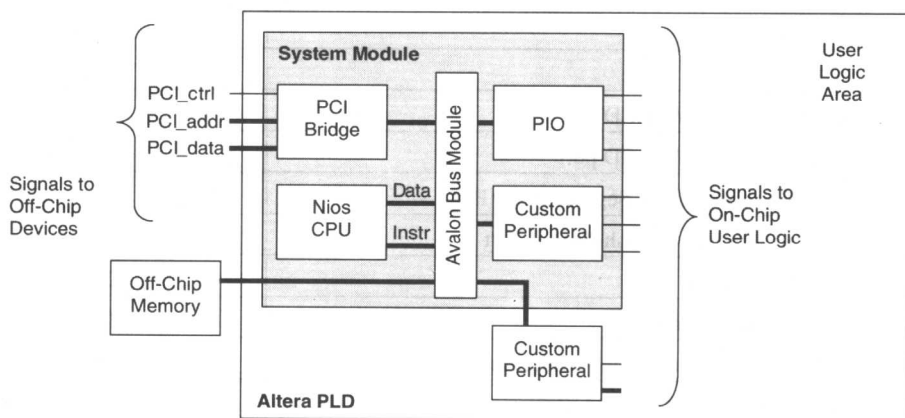


图 1.1 采用 NIOS CPU 的系统框图

系统模块中包含至少一个 Avalon 主外设和整个 Avalon 总线模块。系统模块通常还包含一些 Avalon 从外设, 例如 UART、PIO 和定时器等。SOPC Builder 可以帮助设计者从 IP 库寻找合适的 IP 并很快地集成一个系统, 它采用图形用户界面 (GUI) 显示和组织 IP 模块, 能够自动生成 IP 模块互连逻辑, 并生成用于综合和模拟的文件, 使得设计者可以轻松完成系统设计。

当用户使用 SOPC Builder 创建一个新的系统时, SOPC Builder 会为该系统自动生成一个 PTF 文件, 所有的设计信息都存储在该 PTF 文件里。当使用 SOPC Builder 重新打开一个已有的系统时, SOPC Builder 会从并且只从 PTF 文件中读取系统具体设计信息。

用户逻辑区内可以包含用户自定义的 Avalon 外设, 以及与系统模块无关的其他的用户自定义逻辑。

硬件系统建立起来以后, 用户可以利用 Altera®公司提供的 NIOS SDK Shell 工具, 结合 SOPC Builder 生成的与硬件系统对应的软件开发包来开发用户软件。

1.4 支持 NIOS CPU 的 FPGA 型号

并不是所有 Altera®公司的 FPGA 都支持 NIOS 软核处理器, 目前只有 APEX、Stratix、StratixII 和 Cyclone 4 个系列支持, SOPC 软件会针对相应的器件进行优化。以 Cyclone 系列为例, 一个典型的 32 位 NIOS 系统大约只占用 3000 个 LE (详细资源占用情况见表 1.1)。如果用户不再添加其他逻辑, 恰好容纳在 Cyclone 系列中的最小容量的型号——EP1C3 中。当然, 如果用户采用 16 位的 NIOS CPU, 占用的 LE 数量会更小。表 1.2 至表 1.6 是不同系列的 FPGA 片内资源列表, 用户可以参考以便选择相应器件。