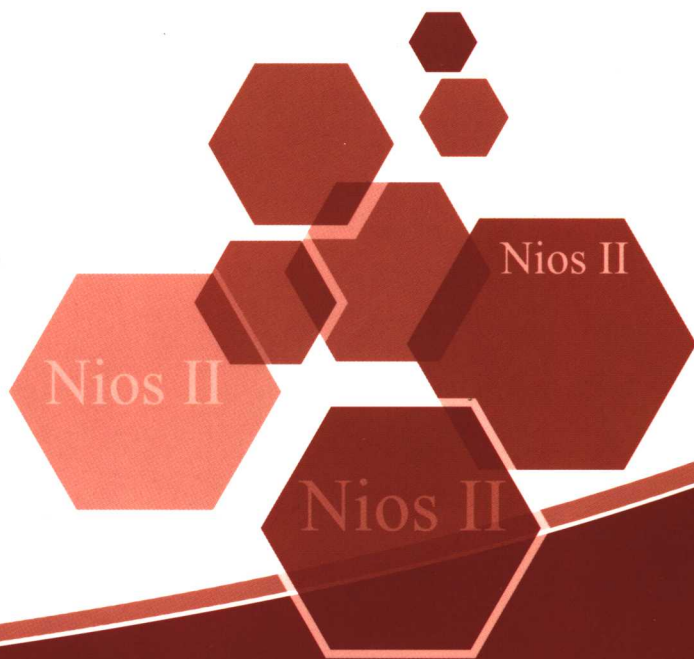


高等院校通用教材·SOPC嵌入式开发系列教程

① 本书配套多媒体教学课件

SOPC

嵌入式系统基础教程



周立功 等编著



北京航空航天大学出版社

中国高等院校计算机专业系列教材

第 2 版

SOPC

嵌入式系统基础教程



清华大学出版社



清华大学出版社

高等院校通用教材·SOPC 嵌入式开发系列教程

TP332

83

2006

SOPC 嵌入式系统 基础教程

周立功 等编著

北京航空航天大学出版社

内 容 简 介

本书为《SOPC 嵌入式开发系列教程》的理论教材,介绍了在嵌入式系统中日益广泛应用的 SOPC 技术基础知识。主要以 Altera 公司的技术资料为基础,加之作者的实践及理解,全面、深入浅出地介绍了 SOPC 技术。全书分为 3 部分:第 1、3、4 章为基础部分,主要介绍 SOPC 技术、Nios II 处理器的体系结构、Nios II 处理器的常用外设;第 2、5、6 章为应用部分,重点介绍 SOPC 开发流程、FPGA 配置、Flash 编程、软件开发等;第 7、8、9 章为提高部分,主要介绍 Avalon 接口规范、SOPC 深入设计、混合语言编程等。

本书可作为高等院校电子工程、计算机、微电子、通信、自动控制等相关专业 SOPC 嵌入式系统课程的理论教材,也可作为从事 SOPC 嵌入式系统应用开发工程师的参考资料。

本书有配套实验教程以及多媒体教学课件。

图书在版编目(CIP)数据

SOPC 嵌入式系统基础教程/周立功编著. —北京:北京航空航天大学出版社,2006. 11
ISBN 7-81077-902-8

I. S… II. 周… III. 微处理器—系统设计—高等院校—教材 IV. TP332

中国版本图书馆 CIP 数据核字(2006)第 126477 号

© 2006, 北京航空航天大学出版社, 版权所有。

未经本书出版者书面许可,任何单位和个人不得以任何形式或手段复制或传播本书内容。
侵权必究。

SOPC 嵌入式系统基础教程

周立功 等编著

责任编辑 张军香

*

北京航空航天大学出版社出版发行

北京市海淀区学院路 37 号(100083) 发行部电话:010-82317024 传真:010-82328026

<http://www.buaapress.com.cn> E-mail: bhpess@263.net

涿州市新华印刷有限公司印装 各地书店经销

*

开本:787 mm×960 mm 1/16 印张:23.25 字数:521 千字

2006 年 11 月第 1 版 2006 年 11 月第 1 次印刷 印数:5 000 册

ISBN 7-81077-902-8 定价:29.50 元

前 言

微电子技术的发展历史是一个不断创新的过程,这种创新包括理论创新、技术创新和应用创新。每一种创新都能开拓出一个新的领域,带来新的市场,产生重大的影响。

在集成电路(IC)发展初期,电路设计主要是器件的物理版图设计。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。随着IC设计与工艺水平的提高,原先由许多IC组成的电子系统可以集成到一个芯片上,构成片上系统(System on aChip, SOC)。SOC的设计思想有别于普通IC;SOC从整个系统的角度出发,把处理机制、模型算法、芯片结构直至器件的设计紧密结合起来,在单个(或少数几个)芯片上实现整个系统的功能。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,并需要借助EAD工具。研究表明,与IC组成的系统相比,由于SOC设计能够综合考虑整个系统的各种情况,因而可以在同样的工艺条件下,实现更高性能的系统指标。

SOPC(System On Programmable Chip, 可编程的片上系统),是Altera公司提出的一种灵活、高效的SOC解决方案,是一种新的软硬件协同设计的系统设计技术。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可编程器件上,构成一个可编程的片上系统。

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、I/O设备和可编程逻辑的SOPC芯片,在应用灵活性和价格上有相当大的优势,所以有人认为SOPC代表了半导体产业未来的发展方向。

本书为《SOPC嵌入式开发系列教程》的理论教材,而事实上,本书的内容并非纯理论的,也包含很多实践部分。以作者的经验,SOPC技术只有通过大量的实战与实践才能很好并快速地掌握,因此本书的内容编排力求实用、深入浅出。本书内容编排如下:

第1章以简短的篇幅介绍了SOPC技术及Altera公司系列产品中可用于SOPC的FPGA,重点介绍了32位Nios II软核处理器。

第2章首先详细介绍了SOPC开发的基本流程,然后非常详细地描述了一个简单SOPC系统的设计过程,包括使用Quartus II、SOPC Builder定制Nios II系统以及利用Nios II IDE进行应用程序开发。本章力求通过实例让读者以最快的方式了解SOPC开发以及各软件的使

前 言

用,从而激起读者对 SOPC 的兴趣。

第 3 章详述了 Nios II 处理器的体系结构,以便让读者了解 Nios II 处理器的工作细节。虽然在软件开发过程中我们多数情况下只使用 C 语言,无须关心体系结构的细节。但熟悉体系结构对开发出高效、健壮的程序是非常有好处的。

第 4 章介绍了 Nios II 处理器常用外围设备的特点、配置以及软件编程,供读者在使用这些外设定制 Nios II 系统时查阅。

第 5 章对较难掌握的 FPGA 配置、Flash(包括 EPCS)编程、目标板 Flash 编程设计的创建以及 Nios II 系统用户程序引导等内容进行了详细阐述。

第 6 章首先简单介绍 Nios II IDE 开发环境的使用;然后重点介绍了硬件抽象层(HAL)系统库,包括 HAL 下的应用程序开发以及开发 HAL 下的设备驱动两部分内容;最后讲述了 MicroC/OS-II 实时操作系统及其应用程序开发实例。本章还简单介绍了最新的 Nios II C 语言至硬件加速编译器。

第 7 章详细介绍了 Avalon 接口规范,着重从 Avalon 外设的角度论述 Avalon 接口。如果用户只使用已经定制好的外设来创建自己的系统,就无须了解有关 Avalon 的相关知识。但对于需要自己设计外设的用户来说,就需要掌握诸如传输类型、信号时序等全面的 Avalon 接口知识,第 7 章的内容就显得尤为重要。


第 8 章讲述有关 SOPC 的深入设计,以帮助读者掌握如何定制用户逻辑外设和用户指令。定制用户逻辑外设和定制用户指令是使用 Nios II 嵌入式软核处理器的 SOPC 系统的重要特性,用户还可以通过定制用户逻辑外设和定制用户指令来实现各种应用要求。

第 9 章首先简单地回顾了程序语言发展的历史,并从中总结了汇编程序的重要性;随后概要地介绍了 Nios II 的指令集;最后重点介绍了 Nios II 汇编程序的设计方法以及混合语言程序开发。

附录 1 介绍了广州致远电子有限公司的 SmartSOPC 多功能教学实验开发平台,本书的许多内容都在该开发平台上通过验证。

附录 2 首先简单介绍了逻辑分析仪的基础知识;重点比较了逻辑分析仪与示波器的区别,以此得出逻辑分析仪在数字电路设计中的重要作用。最后向读者简单介绍了广州致远电子有限公司的 LA 系列高性能逻辑分析仪。

在阅读本书的过程中,请读者注意以下几点:

- ▶ 本书的部分内容是基于 Altera 公司的 6.0 版本的英文手册,Altera 公司会不断更新和修改各参考手册的内容,如遇到不符,请读者以最新的手册为准。
- ▶ 本书列出了作者们在验证内容过程中遇到很多问题的解决方法和注意事项,书中特别用表示出来。
- ▶ 本书对一些术语在不同场合下并没有严格进行统一,事实上 Altera 公司的英文手册也是如此。不统一的术语主要是外设、元件、IP Core。在 SOPC Builder 中通常使用元件

(component)或 IP Core 来描述硬件模块,而在软件编程中通常把在 SOPC Builder 中称之为元件或 IP Core 的硬件模块描述为外设或设备。

- ▶ 由于中文资料缺乏,一些内容是在 Altera 英文文档及软件帮助内容的基础上翻译的,在翻译过程中一些表达可能不恰当,为此在括号内给出了相应的英文表达,以便于读者更好地理解。

最后建议读者在学习本书的过程中要进行实际的动手操作,“纸上得来终觉浅,绝知此事要躬行”。再好的讲述也比不上亲身的体验,如果不亲自动手,可能永远都不知道什么是困难。在实验的过程中,推荐使用本书的配套实验教程《Nios II SOPC 嵌入式系统实验教程(一)》以及 SmartSOPC 多功能教学实验平台。

本书适合于高等院校电子工程、计算机、微电子、通信、自动控制等专业以及对此类专业感兴趣的高年级本科生和研究生。作者相信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,就能较快地进入 SOPC 技术世界。

参与本书编写工作的主要人员有周书武、刘英斌、张晶、芮文彬、杜佳、许惠泉、余文亮等,本书由周立功负责规划、内容编排、定稿与修改。感谢北京航空航天大学出版社的大力支持,使本书得以快速出版。

由于作者水平有限,书中的内容难免有疏忽、不恰当甚至错误的地方,恳请各位老师及同行指正,并请您将阅读中发现的错误或建议发送到 tools@embedtools.com,以使本书不断完善。

如果读者在阅读本书时有什么问题,或需要技术支持,可与广州周立功单片机发展有限公司联系(联系方式见书后)。

- ▶ 本书配套实验教材基于《SOPC 嵌入式系统基础教程》以及 SmartSOPC 多功能教学实验开发平台。读者若需要购买配套的教学实验开发平台,请与广州周立功单片机发展有限公司联系(联系方式见本书最后)。
- ▶ 本书配套有多媒体教学课件,有需要的教师可与广州周立功单片机发展有限公司或北航出版社联系。北航出版社联系方式如下:

通信地址:北京海淀区学院路 37 号 北京航空航天大学出版社教材推广部

邮 编:100083

电 话:010-82317027

传 真:010-82328026

E-mail: zhujunwei_218@163.com

周立功

2006 年 7 月 20 日

目 录

第 1 章 概 述	1
1.1 SOPC 及其技术	1
1.1.1 基于 FPGA 嵌入 IP 硬核的 SOPC 系统.....	1
1.1.2 基于 FPGA 嵌入 IP 软核的 SOPC 系统.....	1
1.1.3 基于 HardCopy 技术的 SOPC 系统	2
1.2 Nios II 软核处理器	3
1.2.1 Nios II 软核处理器简介	3
1.2.2 可配置嵌入式软核处理器的优势	5
1.3 Altera 支持 SOPC 的 FPGA 简介	9
1.3.1 Cyclone 系列	9
1.3.2 Cyclone II 系列	11
1.3.3 Stratix 系列	12
1.3.4 Stratix II 系列	14
第 2 章 SOPC 开发流程及开发平台简介	16
2.1 SOPC 开发流程	16
2.1.1 硬件开发	18
2.1.2 软件开发	18
2.1.3 SOPC 基本开发流程简介	19
2.2 简单 SOPC 实例开发任务及步骤	21
2.3 分析系统需求.....	22
2.4 使用 Quartus II 建立工程	23
2.4.1 打开 Quartus II 软件并建立工程	23
2.4.2 建立顶层模块	26
2.5 使用 SOPC Builder 创建 Nios II 系统.....	27
2.5.1 SOPC Builder 简介	27
2.5.2 启动 SOPC Builder	29
2.5.3 指定目标 FPGA 和时钟设置	31

2.5.4	添加 Nios II 处理器内核	31
2.5.5	添加片内存储器	33
2.5.6	添加 PIO	35
2.5.7	添加系统 ID 外设	35
2.5.8	指定基地址和中断请求优先级	36
2.5.9	设置 Nios II 复位和异常地址	36
2.5.10	生成 Nios II 系统	37
2.6	集成 Nios II 系统到 Quartus II 工程	39
2.6.1	添加 Nios II 系统模块到 Quartus II 顶层模块	39
2.6.2	添加引脚和其他基本单元	40
2.6.3	选择器件型号	41
2.6.4	分配 FPGA 引脚	42
2.6.5	器件和引脚的其他设置	43
2.7	设置编译选项并编译硬件系统	44
2.7.1	设置编译选项	44
2.7.2	编译硬件系统	44
2.7.3	查看编译报告	44
2.8	下载硬件设计到目标 FPGA	46
2.9	使用 Nios II IDE 建立用户程序	47
2.9.1	创建一个新的 C/C++ 应用工程	47
2.9.2	设置 C/C++ 应用工程系统属性	52
2.9.3	编译连接工程	53
2.10	调试/运行程序	53
2.10.1	在目标硬件上运行程序	54
2.10.2	在目标板上调试程序	54
第 3 章	Nios II 体系结构	57
3.1	Nios II 处理器结构	57
3.2	Nios II 的寄存器文件	58
3.2.1	Nios II 的通用寄存器	58
3.2.2	Nios II 的控制寄存器	60
3.3	算术逻辑单元(ALU)	61
3.3.1	未实现的指令	61
3.3.2	用户指令	61
3.3.3	浮点指令	61

3.4	复位信号	62
3.5	Nios II 处理器运行模式	62
3.6	异常和中断控制器	63
3.6.1	异常控制器	63
3.6.2	中断控制器	63
3.7	Nios II 的异常处理	64
3.7.1	异常类型	64
3.7.2	异常硬件处理流程	64
3.7.3	异常判别及优先级	64
3.7.4	异常的嵌套	65
3.7.5	异常返回	66
3.7.6	异常响应时间	66
3.8	存储器及 I/O 结构	66
3.8.1	指令与数据总线	67
3.8.2	高速缓存	68
3.8.3	紧耦合存储器	69
3.8.4	地址映射	70
3.9	存储器和外设访问	70
3.9.1	寻址方式	70
3.9.2	高速缓存访问	71
3.10	Nios II 处理器性能	71
第 4 章	Nios II 外围设备	73
4.1	并行输入/输出内核	73
4.1.1	PIO 寄存器描述	73
4.1.2	PIO 内核配置选项	75
4.1.3	软件编程	77
4.2	SDRAM 控制器内核	77
4.2.1	SDRAM 控制器内核概述	77
4.2.2	SDRAM 内核配置选项	79
4.2.3	软件编程	81
4.2.4	SDRAM 应用	81
4.3	CFI 控制器内核	82
4.3.1	CFI 控制器内核综述	83
4.3.2	CFI 控制器内核设置	83

4.3.3	软件编程	85
4.4	EPCS 控制器内核	85
4.4.1	EPCS 控制器内核综述	85
4.4.2	软件编程	86
4.5	定时器内核	87
4.5.1	定时器内核综述	87
4.5.2	定时器寄存器描述	88
4.5.3	定时器内核配置选项	90
4.5.4	软件编程	92
4.6	UART 内核	93
4.6.1	UART 内核综述	93
4.6.2	UART 内核的寄存器描述	95
4.6.3	UART 内核配置选项	99
4.6.4	软件编程	101
4.7	JTAG_UART 内核	104
4.7.1	JTAG_UART 的寄存器描述	105
4.7.2	JTAG_UART 内核配置选项	106
4.7.3	软件编程	108
4.8	SPI 内核	110
4.8.1	SPI 内核综述	110
4.8.2	SPI 内核的寄存器描述	112
4.8.3	SPI 内核配置选项	115
4.8.4	软件编程	117
4.9	DMA 内核	118
4.9.1	DMA 控制器内核综述	118
4.9.2	DMA 寄存器描述	119
4.9.3	DMA 控制器配置选项	122
4.9.4	软件编程	123
4.10	带 Avalon 接口的互斥内核	125
4.10.1	互斥内核描述	125
4.10.2	互斥内核配置选项	126
4.10.3	软件编程	126
4.11	带 Avalon 接口的邮箱内核	129
4.11.1	邮箱内核描述	129

4.11.2	邮箱内核配置选项	129
4.11.3	软件编程	130
4.12	System ID 内核	132
4.12.1	System ID 寄存器描述	133
4.12.2	软件编程	133
第 5 章	FPGA 配置和 Flash 编程	134
5.1	FPGA 配置	134
5.1.1	FPGA 配置方式	134
5.1.2	FPGA 配置过程	136
5.2	Cyclone 及 Cyclone II FPGA 配置	138
5.2.1	主动串行配置	140
5.2.2	主动串行配置(AS)的配置时间估算	143
5.2.3	JTAG 配置	143
5.3	配置芯片	145
5.4	配置的软件设置	146
5.4.1	配置方式及配置芯片选择	146
5.4.2	复用配置引脚处理	147
5.5	配置文件	148
5.5.1	配置文件的格式	148
5.5.2	配置文件设置和转换	149
5.5.3	配置文件的压缩	150
5.6	配置可靠性及电路设计注意事项	152
5.6.1	配置的可靠性	152
5.6.2	电路设计注意事项	153
5.7	下载配置文件到 FPGA	153
5.8	下载配置文件到 EPCS	155
5.8.1	AS 接口的 EPCS 直接编程	155
5.8.2	JTAG 接口的 EPCS 间接编程	157
5.8.3	SOF 文件转换为 JIC 文件	158
5.8.4	JIC 文件转换为 JAM 文件	161
5.8.5	JIC 和 JAM 文件的内容	163
5.8.6	使用 JIC 文件和 Quartus II Programmer 对 EPCS 编程	163
5.8.7	使用 JAM 文件和 Quartus II Programmer 对 EPCS 编程	163
5.9	IDE Flash Programmer 介绍	164

目 录

5.9.1	IDE Flash Programmer 编程工作过程	165
5.9.2	Flash 编程设计	165
5.9.3	目标板 Flash 编程设计	165
5.9.4	Flash 编程内容的类型	166
5.9.5	Flash 文件	166
5.10	用户程序引导	167
5.10.1	从 CFI Flash 引导	167
5.10.2	从串行配置器件 EPCS 引导	167
5.10.3	引导复制程序	168
5.10.4	用户程序 ELF 文件连接设置	169
5.10.5	用户程序引导小结	171
5.11	使用 IDE Flash Programmer 编程	172
5.12	创建目标板 Flash 编程设计	176
第 6 章	Nios II 应用程序及其外设 HAL 驱动开发	184
6.1	Nios II IDE 简介	184
6.2	设置工程系统库属性和编译选项	186
6.2.1	C/C++ Build 设置	187
6.2.2	C/C++ Indexer 设置	188
6.2.3	System Library 设置	188
6.3	调试/运行程序	190
6.3.1	调试器目标连接设置	191
6.3.2	调试器设置	192
6.3.3	运行程序	194
6.4	下载程序到 Flash	194
6.5	硬件抽象层(HAL)系统库	194
6.5.1	HAL 的构成	195
6.5.2	HAL 的通用设备模型	196
6.6	使用 HAL 开发应用程序	197
6.6.1	Nios II IDE 工程结构	197
6.6.2	与目标系统相关的 system.h 系统描述文件	198
6.6.3	数据宽度及 HAL 类型定义	200
6.6.4	HAL 对系统启动的支持	200
6.6.5	HAL 与异常处理	201
6.6.6	提高处理性能的措施	203

6.7	设备驱动概述	204
6.7.1	设备的寄存器抽象	205
6.7.2	处理器外围设备的分类	205
6.7.3	标准 I/O 访问方法	206
6.7.4	标准 I/O 函数到具体驱动程序的映射	206
6.7.5	驱动程序与设备交互的方法——轮询和中断	208
6.7.6	调用设备驱动的方式	208
6.8	开发 HAL 下的设备驱动	209
6.8.1	创建一个描述设备寄存器的头文件	209
6.8.2	为各类设备创建驱动程序	210
6.8.3	把设备驱动集成到 HAL 中	218
6.9	$\mu\text{C}/\text{OS}-\text{II}$	221
6.9.1	$\mu\text{C}/\text{OS}-\text{II}$ 简介	222
6.9.2	设备驱动的线程安全	222
6.9.3	Newlib C 标准库的线程安全	223
6.9.4	HAL 系统时钟与 $\mu\text{C}/\text{OS}-\text{II}$ 系统时钟	224
6.9.5	$\mu\text{C}/\text{OS}-\text{II}$ 代码编译和初始化	225
6.9.6	使用 Nios II IDE 开发 $\mu\text{C}/\text{OS}-\text{II}$ 应用程序	225
6.9.7	$\mu\text{C}/\text{OS}-\text{II}$ 基本常识	230
6.10	Nios II C 语言至硬件加速编译器(C2H)简介	231
第 7 章	Avalon 接口规范	233
7.1	Avalon 总线简介	233
7.2	术语和概念	235
7.2.1	Avalon 交换结构	235
7.2.2	Avalon 外设	237
7.2.3	Avalon 信号	238
7.2.4	Avalon 端口	238
7.2.5	主从端口对	239
7.2.6	传 输	239
7.2.7	周 期	239
7.2.8	PTF 文件与 SOPC Builder	239
7.3	Avalon 信号	240
7.3.1	信号命名规则	241
7.3.2	信号类型的完全列表	241

7.3.3	信号极性	244
7.3.4	信号序列和时序	244
7.3.5	传输属性	245
7.4	从端口传输	246
7.4.1	从端口基本信号	246
7.4.2	从端口基本读传输	247
7.4.3	带传输属性的从端口读传输	248
7.4.4	从端口基本写传输	252
7.4.5	带传输属性的从端口写传输	252
7.5	主端口传输	255
7.5.1	主端口信号	256
7.5.2	主端口基本读传输	257
7.5.3	主端口基本写传输	258
7.5.4	主端口传输的等待周期、建立时间及保持时间属性	259
7.5.5	主端口传输的流水线、突发及三态属性	260
7.6	流水线传输属性	260
7.6.1	具有固定延迟的从端口流水线读传输	260
7.6.2	具有可变延迟的从端口流水线读传输	261
7.6.3	主端口流水线读传输	263
7.7	流控制属性	264
7.7.1	具有流控制的从端口传输	265
7.7.2	具有流控制的主端口传输	267
7.8	三态传输属性	269
7.8.1	三态从端口传输	270
7.8.2	三态主端口传输	276
7.9	突发传输属性	277
7.9.1	主端口突发传输	277
7.9.2	从端口突发传输	280
7.10	非传输相关的信号	283
7.10.1	中断请求信号	283
7.10.2	复位控制信号	285
7.11	地址对齐	286
7.11.1	静态地址对齐	286
7.11.2	动态地址对齐	287

7.11.3	为 Avalon 端口选择地址对齐参数	288
7.11.4	不同地址对齐方式的连线及编程操作	289
第 8 章	Nios II 系统深入设计	291
8.1	定制基于 Avalon 的用户外设	291
8.1.1	定制用户外设简介	291
8.1.2	PWM 设计说明	293
8.1.3	将 PWM 设计封装为 SOPC Builder 元件	296
8.1.4	使用并验证新创建的 PWM 元件	300
8.1.5	手工修改创建的元件	302
8.2	定制 Nios II 用户指令	302
8.2.1	Nios II 定制指令综述	303
8.2.2	定制指令实现方式	307
8.2.3	定制指令设计示例	308
8.2.4	定制指令的使用方法	312
第 9 章	Nios II 汇编程序设计及混合语言程序开发	317
9.1	历史回顾	317
9.2	Nios II 的指令集	318
9.2.1	数据传输指令	318
9.2.2	寄存器间数据拷贝指令	319
9.2.3	算术逻辑运算指令	319
9.2.4	比较指令	320
9.2.5	移位指令	321
9.2.6	跳转、分支和子程序调用指令	321
9.2.7	断点及异常指令	322
9.2.8	用户指令	322
9.2.9	未实现的指令	323
9.2.10	空操作	323
9.3	Nios II 汇编程序设计	323
9.3.1	利用 Nios II 汇编语言实现“Hello World!”	323
9.3.2	在 C 语言中内嵌汇编	334
9.4	使用混合语言进行程序开发	341
附录 1	SmartSOPC 多功能教学实验开发平台	344
附录 2	LA 系列高性能逻辑分析仪	349
参考文献	354

第 1 章

概 述

SOPC(System On Programmable Chip,可编程的片上系统)是 Altera 公司提出来的一种灵活、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可编程器件上,构成一个可编程的片上系统。

SOPC 是 PLD 和 ASIC 技术融合的结果,可以认为 SOPC 代表了半导体产业未来的发展方向。

1.1 SOPC 及其技术

现今,可将 SOPC 视为是基于 FPGA 解决方案的 SOC。与 ASIC 的 SOC 解决方案相比,SOPC 系统及其开发技术具有更多的特色,也有多种途径可构成 SOPC 方案。

1.1.1 基于 FPGA 嵌入 IP 硬核的 SOPC 系统

基于 FPGA 嵌入 IP 硬核的 SOPC 系统是指在 FPGA 中预先植入处理器。目前最常用的嵌入式处理器大多是采用含有 ARM32 位知识产权处理器核的器件。为了达到通用性,必须为常规的嵌入式处理器集成诸多通用和专用的接口,但这样无疑会增加芯片的成本和功耗。如果将 ARM 或其他处理器核以硬核方式植入 FPGA 中,利用 FPGA 的可编程逻辑资源,按照系统功能需求来添加接口功能模块,既能实现目标系统功能,又能降低系统的成本和功耗。这样就使得 FPGA 灵活的硬件设计与处理器的强大软件功能有机地结合在一起,高效地实现 SOPC 系统。

1.1.2 基于 FPGA 嵌入 IP 软核的 SOPC 系统

将 IP 硬核直接植入 FPGA 的解决方案存在以下不足:

(1) 由于此类硬核多来自第三方公司,FPGA 厂商通常无法直接控制其知识产权费用,从而导致 FPGA 器件价格相对偏高。

(2) 由于硬核是预先植入的,设计者无法根据实际需要改变处理器的结构,如总线规模、接口方式、指令形式,更不可能将 FPGA 逻辑资源构成的硬件模块以指令的形式嵌入硬件加