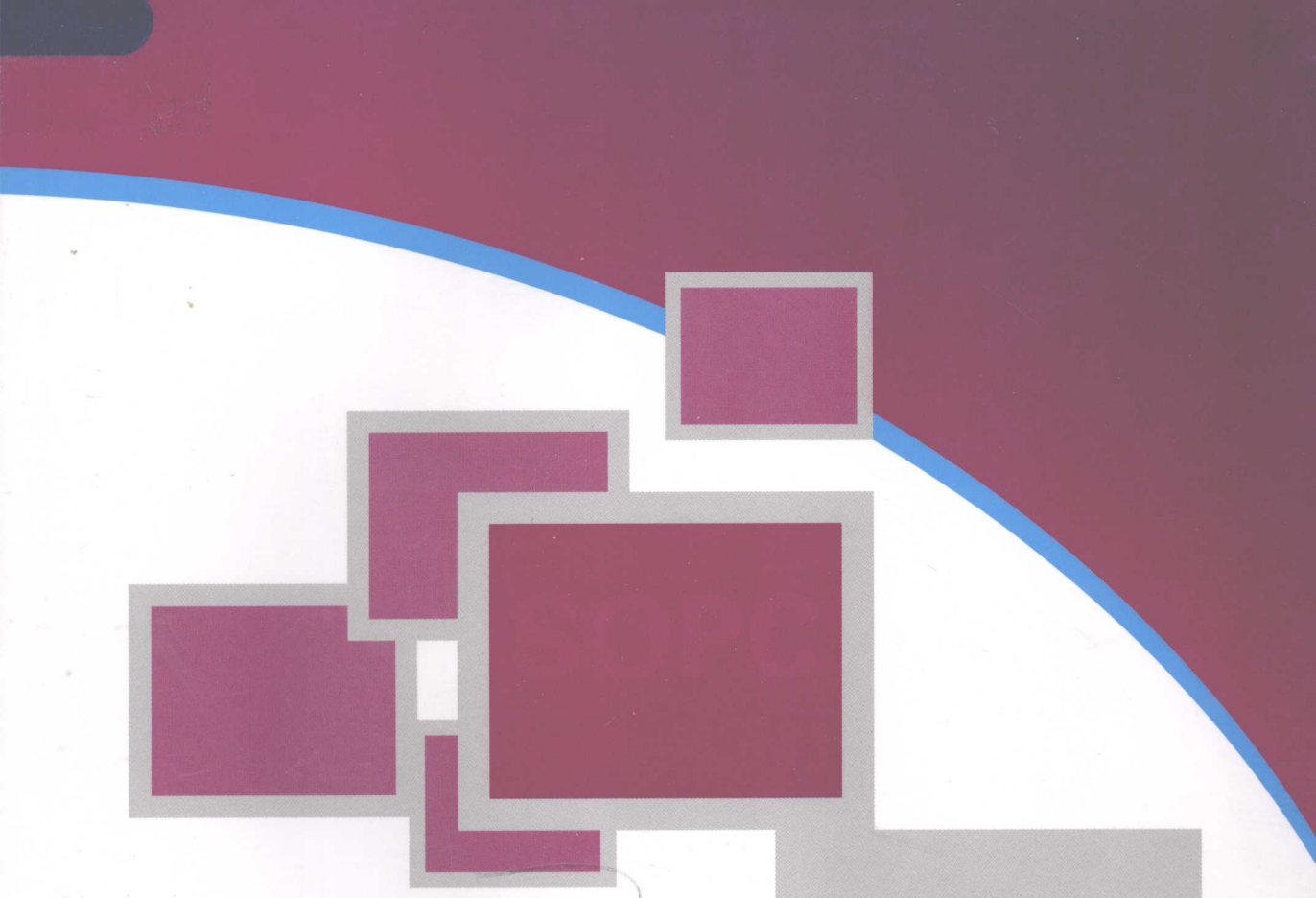




**SOPC**

王晓迪 张景秀 编著

# 系统设计与实践



北京航空航天大学出版社

TP332/148D

2008

# SOPC 系统设计与实践

王晓迪 张景秀 编著

北京航空航天大学出版社

## 内 容 简 介

本书根据 Altera 公司的最新技术资料,详细讲解了 SOPC 系统设计的相关概念,如 Avalon 总线规范、SOPC Builder 和 Nios II 软件设计及嵌入式外围设备等,并结合实例,对软件设计的方法及流程进行了深入分析。本书以讲解 Nios II 软件设计及其实验为主,初学者通过本书可以学习到 Nios II 设计的基本流程及方法,掌握其设计原理,具备软件开发能力。

本书可供高等院校计算机、自动化、电子工程等专业学生及相关工程技术人员使用。

## 图书在版编目(CIP)数据

SOPC 系统设计与实践/王晓迪,张景秀编著. —北京:  
北京航空航天大学出版社,2008.8  
ISBN 978-7-81124-321-5

I. S… II. ①王…②张… III. 微处理器—系统设计  
IV. TP332

中国版本图书馆 CIP 数据核字(2008)第 110761 号

©2008,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本书及其所附光盘内容。

侵权必究。

## SOPC 系统设计与实践

王晓迪 张景秀 编著

责任编辑 董立娟

\*

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(100191) 发行部电话:010-82317024 传真:010-82328026

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

北京市松源印刷有限公司印装 各地书店经销

\*

开本:787×960 1/16 印张:18.75 字数:420 千字

2008 年 8 月第 1 版 2008 年 8 月第 1 次印刷 印数:5 000 册

ISBN 978-7-81124-321-5 定价:32.00 元(含光盘 1 张)

# 前言

当今的嵌入式设计工程师面临很棘手的挑战：如何寻找一款能够实现成本、性能和生命周期完美组合的处理器。而 Altera Nios II 处理器所具有的完全可定制性能、较低的产品和实施成本、易用性、适应性和不会过时等优势，使其在每次设计中都能够实现完美的配合。

Nios II 系列 32 位 RISC 嵌入式处理器具有超过 200 DMIP 的性能，在 FPGA 中实现成本只有 35 美分。由于处理器是软核形式，具有很大的灵活性，因此，用户可以在多种系统设置组合中进行选择，达到性能和成本的要求。采用 Nios II 处理器进行设计，可以帮助用户将产品迅速推向市场，延长产品生命周期，防止出现处理器逐渐过时的问题。

采用 Nios II 处理器，用户不会局限于现有的处理器技术，而是根据自己的标准定制处理器，按照需要选择合适的外设、存储器和接口；此外，用户还可以轻松集成自己专有的功能，使用户的设计具有独特的竞争优势。

用户所需要的处理器，应该能够满足当前和今后的设计性能需求。由于今后的发展具有不确定性，因此，Nios II 设计人员必须能够更改其设计，如加入多个 Nios II CPU、定制指令集及硬件加速器等，以满足新的性能目标。采用 Nios II 处理器，用户可以通过 Avalon 交换架构来调整系统性能，该架构是 Altera 公司专有的互联技术，支持多种并行数据通道，可以实现大吞吐率的应用。

在选择处理器时，为了实现需要的功能，用户可能要购买比实际所需数量多的处理器；也可能为了节省成本，而不得不购买比实际需要数量少的处理器。低成本、可定制 Nios II 处理器能够帮助用户解决这一难题。采用 Nios II 处理器，用户可以根据需要设置功能，甚至在价格低至 35 美分的 Cyclone II FPGA 等低成本 Altera 器件中都可以实施。在单个 FPGA 中实现处理器、外设、存储器和 I/O 接口，可以降低系统总体成本。

## 前言

为实现一个成功的产品,需要将其尽快推向市场,同时,增强其功能特性以延长使用时间,避免处理器逐渐过时。用户可以在短时间内将 Nios II 嵌入式处理器由最初概念设想转为系统实现,这种基于 Nios II 处理器的系统具有永久免版税设计许可,完全经得起时间考验。此外,由于在 FPGA 中使用软核处理器,因此可以方便地实现现场硬件和软件升级,且产品能够符合最新的规范,具备最新特性。

本书在编写过程中,得到了冯瑞波、卢海涛、葛昊圳、王浩、柳向龙、曹文林、杜剑、殷树胜、苗钰雨、王相臣、于淑华、封福东和李斌等的帮助和关心,在此表示衷心的感谢。

SOPC 的概念涉及范围广,本书只是作者个人学习的一些总结,只是想为初学者提供一些帮助。随书光盘含有书中的 DEMO 程序以及 Sch 原理图,供读者参考。由于时间仓促,书中不妥之处在所难免,敬请读者批评指正。有兴趣的朋友,可以到作者的个人博客——EDACHina(<http://wangxiaodi213.blog.sohu.com>)来做客;或者登陆 SOPC 技术联盟([www.sopc.net.cn](http://www.sopc.net.cn))参与技术讨论。

作者

2008 年 4 月



# 录

|                             |    |
|-----------------------------|----|
| 第 1 章 总体概述 .....            | 1  |
| 1.1 学习平台概述 .....            | 1  |
| 1.1.1 软件平台 .....            | 1  |
| 1.1.2 硬件平台 .....            | 2  |
| 1.2 Nios 概述 .....           | 2  |
| 1.2.1 第一代 Nios 嵌入式处理器 ..... | 2  |
| 1.2.2 第二代 Nios 嵌入式处理器 ..... | 3  |
| 1.2.3 Nios II 处理器的特性 .....  | 3  |
| 第 2 章 Avalon 总线规范 .....     | 5  |
| 2.1 概 述 .....               | 5  |
| 2.1.1 术 语 .....             | 5  |
| 2.1.2 功 能 .....             | 7  |
| 2.2 Avalon 总线信号 .....       | 8  |
| 2.2.1 完整的信号类型列表 .....       | 9  |
| 2.2.2 信号极性 .....            | 12 |
| 2.2.3 信号命名 .....            | 12 |
| 2.2.4 信号时序 .....            | 13 |
| 2.2.5 传输特性 .....            | 13 |
| 2.3 从端口传输 .....             | 14 |
| 2.3.1 从设备信号 .....           | 14 |
| 2.3.2 基本的从端口读传输 .....       | 16 |
| 2.3.3 基本的从端口写传输 .....       | 20 |
| 2.4 主端口传输 .....             | 23 |
| 2.4.1 主信号 .....             | 24 |

## 目 录

|       |                                 |    |
|-------|---------------------------------|----|
| 2.4.2 | 基本的主端口读传输                       | 25 |
| 2.4.3 | 基本的主端口写传输                       | 26 |
| 2.4.4 | 等待状态、建立时间与保持时间                  | 27 |
| 2.5   | 流水线读传输                          | 27 |
| 2.5.1 | 具有固定延时的从端口流水线读传输                | 28 |
| 2.5.2 | 具有可变延时的从端口流水线读传输                | 29 |
| 2.5.3 | 主端口的流水线读传输                      | 30 |
| 2.6   | 流传输控制                           | 31 |
| 2.6.1 | 从端口传输的流控制                       | 31 |
| 2.6.2 | 具有流控制的主端口传输                     | 34 |
| 2.7   | 三态传输                            | 34 |
| 2.7.1 | 三态从端口的传输                        | 35 |
| 2.7.2 | 三态主端口的传输                        | 38 |
| 第3章   | SOPC Builder                    | 40 |
| 3.1   | SOPC Builder 简介                 | 40 |
| 3.1.1 | 概 述                             | 40 |
| 3.1.2 | SOPC Builder 系统结构               | 41 |
| 3.1.3 | SOPC Builder 功能介绍               | 43 |
| 3.2   | SOPC Builder 用户接口概述             | 44 |
| 3.2.1 | SOPC Builder 的使用                | 44 |
| 3.2.2 | System Contents 选项卡             | 45 |
| 3.2.3 | Nios II More “CPU” Settings 选项卡 | 49 |
| 3.2.4 | Board Settings 选项卡              | 50 |
| 3.2.5 | System Generation 选项卡           | 50 |
| 3.2.6 | 首选项                             | 53 |
| 3.3   | 存储器映射接口的系统互联结构                  | 54 |
| 3.3.1 | 概 述                             | 54 |
| 3.3.2 | 地址译码                            | 57 |
| 3.3.3 | 数据的多路复用                         | 58 |
| 3.3.4 | 等待状态嵌入                          | 58 |
| 3.3.5 | 高性能的流水线                         | 59 |
| 3.3.6 | 流水线的管理                          | 60 |
| 3.3.7 | endian 转换                       | 60 |

|       |                      |     |
|-------|----------------------|-----|
| 3.3.8 | 本地地址定位及动态总线大小        | 60  |
| 3.4   | SOPC Builder 组件      | 61  |
| 3.4.1 | 概 述                  | 61  |
| 3.4.2 | 组件目录的结构与内容           | 63  |
| 3.4.3 | 组件目录位置               | 64  |
| 3.5   | 组件编辑器                | 65  |
| 3.5.1 | HDL Files 选项卡        | 66  |
| 3.5.2 | Signals 选项卡          | 67  |
| 3.5.3 | Interfaces 选项卡       | 69  |
| 3.5.4 | SW Files 选项卡         | 71  |
| 3.5.5 | Component Wizard 选项卡 | 71  |
| 3.6   | 目标板描述                | 73  |
| 3.6.1 | 概 述                  | 73  |
| 3.6.2 | 创建目标板描述              | 74  |
| 3.6.3 | 目标板描述编辑器的使用          | 78  |
| 3.7   | 引脚映射                 | 84  |
| 3.7.1 | 概 述                  | 84  |
| 3.7.2 | 设计流程                 | 85  |
| 3.7.3 | 引脚映射 GUI             | 86  |
| 第 4 章 | 嵌入式外围设备              | 88  |
| 4.1   | SDRAM 控制器内核          | 88  |
| 4.1.1 | 概 述                  | 88  |
| 4.1.2 | 在 SOPC Builder 中示例内核 | 91  |
| 4.1.3 | 硬件仿真                 | 93  |
| 4.1.4 | 时钟、PLL 和时序           | 95  |
| 4.2   | CFI Flash 控制器        | 99  |
| 4.2.1 | 概 述                  | 99  |
| 4.2.2 | 在 SOPC Builder 中综合内核 | 100 |
| 4.2.3 | 软件设计模型               | 101 |
| 4.3   | EPCS 器件控制器内核         | 102 |
| 4.4   | DMA 控制器内核            | 104 |
| 4.4.1 | 概 述                  | 104 |
| 4.4.2 | 基本功能描述               | 104 |

## 目 录

|        |                      |     |
|--------|----------------------|-----|
| 4.4.3  | 在 SOPC Builder 中示例内核 | 106 |
| 4.4.4  | 软件设计模型               | 107 |
| 第 5 章  | Nios II 软件设计         | 108 |
| 5.1    | Nios II 概述           | 108 |
| 5.1.1  | 概 述                  | 108 |
| 5.1.2  | 开发环境                 | 108 |
| 5.1.3  | 第三方支持                | 110 |
| 5.2    | Nios II IDE          | 110 |
| 5.2.1  | 概 述                  | 110 |
| 5.2.2  | Nios II IDE 工作台      | 110 |
| 5.2.3  | 建立新的工程               | 110 |
| 5.2.4  | 组建并编辑工程              | 112 |
| 5.2.5  | 运行和调试程序              | 114 |
| 5.2.6  | 编辑 Flash             | 117 |
| 5.2.7  | 帮助系统                 | 118 |
| 5.3    | HAL 系统库简介            | 119 |
| 5.3.1  | 概 述                  | 119 |
| 5.3.2  | 开始设计                 | 119 |
| 5.3.3  | HAL 的结构              | 119 |
| 5.4    | 使用 HAL 进行程序开发        | 122 |
| 5.4.1  | 概 述                  | 122 |
| 5.4.2  | Nios II IDE 工程结构     | 122 |
| 5.4.3  | system.h 系统说明文件      | 123 |
| 5.4.4  | 数据宽度和 HAL 类型的定义      | 124 |
| 5.4.5  | UNIX 风格的接口           | 124 |
| 5.4.6  | 文件系统                 | 125 |
| 5.4.7  | 使用字符模式器件             | 126 |
| 5.4.8  | 使用定时器                | 127 |
| 5.4.9  | 使用 Flash 器件          | 129 |
| 5.4.10 | 使用 DMA 器件            | 134 |
| 5.4.11 | 缩减代码                 | 138 |
| 5.4.12 | 引导程序序列和入口            | 140 |
| 5.4.13 | 存储器的使用               | 142 |

|        |                       |     |
|--------|-----------------------|-----|
| 5.4.14 | HAL 系统库文件路径 .....     | 145 |
| 5.5    | 为 HAL 开发硬件驱动 .....    | 145 |
| 5.5.1  | 概 述 .....             | 145 |
| 5.5.2  | 创建硬件驱动的开发流程 .....     | 146 |
| 5.5.3  | SOPC 编辑器 .....        | 146 |
| 5.5.4  | 硬件访问 .....            | 147 |
| 5.5.5  | 为 HAL 器件创建驱动程序 .....  | 148 |
| 5.5.6  | 将器件驱动集成到 HAL .....    | 155 |
| 5.5.7  | 缩减代码量 .....           | 158 |
| 5.5.8  | 命名空间的分配 .....         | 159 |
| 5.5.9  | 越过默认器件驱动 .....        | 160 |
| 5.6    | 异常处理 .....            | 160 |
| 5.6.1  | 概 述 .....             | 160 |
| 5.6.2  | Nios II 异常概述 .....    | 160 |
| 5.6.3  | 中断服务程序 .....          | 162 |
| 5.6.4  | 中断服务程序性能 .....        | 165 |
| 5.6.5  | 提升中断服务程序的性能 .....     | 165 |
| 5.6.6  | 调试中断 .....            | 168 |
| 5.6.7  | 中断编写的指导总结 .....       | 168 |
| 5.6.8  | HAL 异常控制器 .....       | 169 |
| 5.7    | 高速缓存与紧耦合存储器 .....     | 172 |
| 5.7.1  | 概 述 .....             | 172 |
| 5.7.2  | 高速缓存初始化 .....         | 173 |
| 5.7.3  | 写器件驱动程序 .....         | 174 |
| 5.7.4  | 编写装载程序或者自修改代码 .....   | 175 |
| 5.7.5  | 多 CPU 系统的高速缓存管理 ..... | 176 |
| 5.7.6  | 紧耦合存储器 .....          | 177 |
| 第 6 章  | 实验板电路 .....           | 178 |
| 6.1    | 实验板特点 .....           | 178 |
| 6.2    | 存储电路 .....            | 179 |
| 6.3    | 配置电路 .....            | 181 |
| 6.4    | 按键及 LED 电路 .....      | 182 |
| 6.5    | LCD Module 接口电路 ..... | 185 |

## 目 录

|       |                                               |     |
|-------|-----------------------------------------------|-----|
| 6.6   | E <sup>2</sup> PROM 及 Buzzer 电路 .....         | 186 |
| 6.7   | PS/2 及 VGA 接口电路 .....                         | 187 |
| 6.8   | 红外发射及接收电路 .....                               | 188 |
| 6.9   | RS232 接口电路 .....                              | 188 |
| 6.10  | 时钟及锁相环电路 .....                                | 190 |
| 6.11  | I/O 分配 .....                                  | 190 |
| 6.12  | 电源电路 .....                                    | 191 |
| 第 7 章 | Nios 基础实验 .....                               | 193 |
| 7.1   | 流水灯实验 .....                                   | 193 |
| 7.2   | JTAG UART 通信实验 .....                          | 207 |
| 7.3   | LCM 显示实验 .....                                | 216 |
| 7.4   | 按键中断实验 .....                                  | 219 |
| 7.5   | 计数显示实验 .....                                  | 223 |
| 7.6   | 建立带 Flash 的 Nios II 系统及配置方法 .....             | 225 |
| 7.7   | 定时器编程 .....                                   | 229 |
| 7.8   | I <sup>2</sup> C Controller IP Core 的使用 ..... | 233 |
| 第 8 章 | 基于 Nios 控制的 DDS 信号发生器 .....                   | 236 |
| 8.1   | 设计简介 .....                                    | 236 |
| 8.1.1 | 设计需求 .....                                    | 236 |
| 8.1.2 | DDS 原理简介 .....                                | 236 |
| 8.2   | 设计方案论证 .....                                  | 238 |
| 8.2.1 | 使用 MCU + FPGA + DAC 的方案 .....                 | 238 |
| 8.2.2 | FPGA + DAC 的方案 .....                          | 239 |
| 8.2.3 | 方案论证与选择 .....                                 | 240 |
| 8.3   | 系统框图及实现原理 .....                               | 240 |
| 8.3.1 | 系统框图 .....                                    | 240 |
| 8.3.2 | 实现原理 .....                                    | 241 |
| 8.4   | 软硬件设计 .....                                   | 241 |
| 8.4.1 | DAC 模块电路设计 .....                              | 241 |
| 8.4.2 | Nios II 硬件部分设计 .....                          | 242 |
| 8.4.3 | DDS 硬件部分设计 .....                              | 243 |
| 8.4.4 | C 语言部分设计 .....                                | 244 |

|                                             |            |
|---------------------------------------------|------------|
| 8.5 小 结 .....                               | 245        |
| <b>第 9 章 Nios II Flash 编程器 .....</b>        | <b>246</b> |
| 9.1 在 SOPC Builder 下定制目标板 .....             | 246        |
| 9.2 Nios II Flash 编程器的使用 .....              | 248        |
| 9.2.1 Nios II Flash 编程器概述 .....             | 249        |
| 9.2.2 烧写内容到 Flash .....                     | 252        |
| 9.2.3 在 SOPC Builder 系统中例化 Flash .....      | 255        |
| 9.3 协控制器 EPM240 的工作原理 .....                 | 258        |
| <b>第 10 章 通过实例讲解 IP Core 的设计过程 .....</b>    | <b>260</b> |
| 10.1 概 述 .....                              | 260        |
| 10.2 SOPC 设备设计流程 .....                      | 261        |
| 10.3 设计实例一 .....                            | 263        |
| 10.4 设计实例二 .....                            | 270        |
| <b>第 11 章 基于 Nios II 控制的 TFT 液晶显示 .....</b> | <b>276</b> |
| 11.1 概 述 .....                              | 276        |
| 11.2 硬件部分设计 .....                           | 277        |
| 11.2.1 硬件电路接口 .....                         | 277        |
| 11.2.2 建立系统 .....                           | 277        |
| 11.3 软件控制部分设计 .....                         | 279        |
| 11.3.1 驱动设计 .....                           | 279        |
| 11.3.2 汉字库及其显示 .....                        | 283        |
| 11.4 小 结 .....                              | 286        |
| <b>参考文献 .....</b>                           | <b>287</b> |

# 第 1 章

## 总体概述

### 1.1 学习平台概述

#### 1.1.1 软件平台

##### 1. Quartus II

Quartus II 是 Altera 公司用于取代 MAX+PLUS II 的升级版本。MAX+PLUS II 在 2000 年已经停止更新,而 Quartus 目前已经更新到 6.0 版本,里面集成了很多有用的工具,如 SOPC Builder(这个工具相比其他同类 EDA 开发工具,同 MAX+PLUS II 一样仍是最好用的)等。MAX+PLUS II 用户可以方便地转入 Quartus II,因为用户可以选择与 MAX+PLUS II 类似的操作界面,同时,大量的说明也可以帮助用户更加详细地了解 Quartus II 的使用。

Quartus 较之 MAX+PLUS II 有如下几个特点:

- 支持更多的新器件;
- 具有更多、更先进的功能以及快速的升级性能,如 SOPC Builder;
- 更多 IP Core 的支持;
- 更多功能的加强,如综合及布线等功能的加强。

##### 2. Nios II IDE

这个工具用于在 Nios II 上开发软件工程,类似于单片机的软件开发环境,但其设置简单。在 Nios II IDE 中,可以完成所有软件开发任务,如工程管理、编辑和编译、调试以及 Flash 器件编程等,这些将在第 3 章的 Nios 实验中详细介绍。在这里要说的是,Nios II IDE 6.0 版本

## 第1章 总体概述

的开发软件需要 Quartus II 6.0 版本的支持,也就是说,Quartus II 的其他低级版本不支持 Nios II IDE 6.0 软件开发环境。因此,常将 Quartus II 和 Nios II IDE 安装在同一个目录下(建议放在 D:\altera 目录下)。

### 1.1.2 硬件平台

本书实验基于 SOPC 技术联盟 SOPC\_EP2C8 开发平台,该平台使用的是性价比较高的 Cyclone II 系列 FPGA 芯片 EP2C8Q208C8,以及丰富的外围设备。其详细特性如下:

- Cyclone II 主控芯片: EP2C8Q208 主芯片+4 MB 配置芯片;
- 8 MB SDRAM: SANSUNG SDRAM;
- 4 MB Flash: 选用 Intel Flash 芯片(方便 Flash 软件编程);
- 大功率输出电源: 方便用户使用板上电源用于扩展(2 次开发);
- 2 个晶振时钟电路: 已有一个 50 MHz 的,预留一个;
- 4 个用户按键及 1 个 Reset 按键: 满足用户最小需求;
- 16×2 字符液晶显示模块: 显示字符及 ASCII 码;
- 4 位七段数码管: 用于数字显示;
- 8 个 LED: 用于调试及状态显示;
- E<sup>2</sup>PROM 芯片: 练习 I<sup>2</sup>C 设备控制;
- 红外发射及接收模块: 练习红外发射及接收;
- RS-232 串口电路: 系统通信接口;
- PS/2 接口电路: 用于键盘设备的控制;
- VGA 接口(8 色)电路: 用于简单逻辑实现 VGA 显示;
- 1 个 24 针扩展接口: 方便用户外接其他模块或用于 2 次开发;
- 板上 BBII 下载电缆: 更好的电气连接特性;
- SD 卡接口: 可练习 SD 卡的读取数据;
- 使用 PI5C3384 作为 I/O 接口缓冲: 以便支持 3.3 V 和 5 V 外设。

## 1.2 Nios 概述

### 1.2.1 第一代 Nios 嵌入式处理器

第一代 Nios 已经体现出嵌入式软核的强大优势,但仍不够完善: 它没有提供软件开发的

集成环境,用户需要在 Nios SDK Shell 中以命令行的形式对软件进行编译、运行、调试,且程序的编辑、编译、调试都是分离的,而且不支持对项目的编译。这对用户来说不够方便,因此,还需要功能更为强大的软核处理器和开发环境。

### 1.2.2 第二代 Nios 嵌入式处理器

2004 年 6 月,Altera 公司在继全球范围内推出 Cyclone II 和 Stratix II 器件系列后,又推出了这些新款 FPGA 系列的 Nios II 嵌入式处理器。Nios II 嵌入式处理器和 Cyclone II FPGA 的组合,使得 Nios 在 Cyclone II 中具有超过 100 DMIP 的性能,允许设计者在很短的时间内构建一个完整的可编程系统,且风险和成本比中小规模的 ASIC 要小。

Nios II 嵌入式处理器、使用 32 位的指令集结构,与二进制代码完全兼容,适用于广泛的嵌入式应用。它包括 3 种内核:快速的(Nios II/f)、经济的(Nios II/e)和标准的(Nios II/s)内核,每种内核对应不同的性能范围和成本。使用 Altera 公司的 Quartus II 软件、SOPC Builder 工具以及 Nios II 集成开发环境(IDE),用户可以轻松地将 Nios II 处理器嵌入到系统中。

Altera 公司推出的 Nios II 系列嵌入式处理器扩展了目前世界上最流行的软核嵌入式处理器的性能,把 Nios II 嵌入到 Altera 公司的所有 FPGA 中,例如 Stratix II, Stratix, Cyclone II, Cyclone, APEX, ACEX 和 HardCopy 系列器件中,用户可以获得超过 200 DMIP 的性能,用户可以从 3 种处理器以及超过 60 个的 IP 核中选择所需 Nios II 系统,用以提供最基本的多功能性,也可以以此创建一个最适用的嵌入式系统。

### 1.2.3 Nios II 处理器的特性

#### 1. 可定制特性集

采用 Nios II 处理器,将不会局限于现有的处理器技术,而是根据自定的标准处理器,根据需要选择合适的外设、存储器和接口。此外,还可以轻松集成特有的功能,使设计更具有竞争优势。

#### 2. 配置系统性能

所需要的处理器应该能够满足当前和今后的设计性能需求。而今后的发展具有不确定性,因此,要求 Nios II 设计人员必须能够更改其设计,通过加入多个 Nios II CPU、定制指令集、硬件加速器,以达到新的性能目标。采用 Nios II 处理器,可以通过 Avalon 交换架构来调整系统性能。该架构是 Altera 公司专有的互联技术,支持多种并行数据通道,因此,可实现大吞吐量应用。

## 第1章 总体概述

### 3. 低成本实现

选择处理器时,为实现需要的功能,可能要购买比实际所需数量多的处理器,也可能为节省成本,不得不购买比实际需要数量少的处理器。而具有低成本、可定制特性的 Nios II 处理器,可以根据需要设置相应功能,甚至在价格低至 35 美分的 Cyclone II FPGA 等低成本 Altera 器件中实施。在单个 FPGA 中实现处理器、外设、存储器和 I/O 接口,可以降低系统总体成本。

### 4. 产品生存周期管理

为实现一个成功的产品,需要将其尽快推向市场,并不断增强其功能特性以延长使用时间,避免处理器逐渐过时。可以在短时间内,将 Nios II 嵌入式处理器由最初概念设想转为系统实现,这种基于 Nios II 处理器的系统具有永久免版税设计许可,完全经得起时间考验。此外,由于在 FPGA 中实现软核处理器,因此,可以方便地实现现场硬件和软件升级,使产品符合最新的规范,具备最新特性。

### 5. 灵活性高

Nios II 具有完全可定制和重新配置的特性,所实现的产品可满足现在和今后的需求。

### 6. 定制指令

Nios II 处理器定制指令扩展了 CPU 指令集,提高了对时间和软件运行速度的要求,从而使开发人员能够提高系统的性能。采用定制指令,可以实现传统处理器无法达到的最佳系统性能。Nios II 系列处理器支持多达 256 条的定制指令,加速一些通常由软件实现的逻辑和复杂的数学算法。

### 7. 硬件加速

专用硬件加速器可以作为 FPGA 中的定制协处理器,用于协助 CPU 同时处理多个数据块。如果实现循环冗余编码,则采用硬件加速器处理 64 KB 缓冲,比软件速度快 530 倍。SOPC Builder 含有一个输入向导,用于将加速逻辑和 DMA 通道引入系统。

此外,可以访问 Altera 的官方网站 [www.altera.com.cn](http://www.altera.com.cn) 或 [www.sopc.net.cn](http://www.sopc.net.cn) 和 [www.ednchina.net](http://www.ednchina.net) 深入了解 Nios。

## 第 2 章

# Avalon 总线规范

## 2.1 概 述

随着 Nios II 的发展, Avalon 总线也产生了一些变化。由于之前的 Nios 系统对应的总线结构在原理上没有大的变化, 因此, 可通过对比发现其变化。

Avalon 接口是为 SOPC 系统开发环境下外设的开发而设计的, 用于提供描述主外设和从外设中基于地址读/写接口的基础, 例如, 微处理器、存储器、UART 及定时器等。接口规范定义了外设和 Avalon 开关互联结构之间的数据传输。在没有主或从接口限制的情况下, 规范的互联策略允许任何主外设连接到任何从外设; Avalon 接口描述了一个可配置的互联策略, 允许外设的设计者限制某种特定传输所需的信号类型。

Avalon\_MM 接口定义了一组信号类型、信号的行为及信号支持的传输类型。例如, Avalon\_MM 接口可以用来描述一个传统外设接口 (SRAM), 仅支持简单的、固定周期的读/写传输。另外, Avalon\_MM 接口也可以描述一个较复杂的、支持突发传输的流水接口。

### 2.1.1 术 语

本小节定义了一些与 Avalon 接口规范相关的术语和概念。

#### (1) Avalon 外设和 Avalon\_MM Interconnect Fabric

使用 Avalon\_MM 接口连接多个外设的系统, 如图 2.1 所示。其中, system interconnect fabric 是系统的片上互连结构, 用于将 Avalon\_MM 的各个外设连接起来。Avalon\_MM 接口定义了外设和系统互连的接入点, 本章主要从外设的角度集中讨论 Avalon\_MM 接口, 定义了接口的系统互连结构, 而未介绍具体的内部应用。