

EDA 工 | 程 | 技 | 术 | 丛 | 书 |

Xilinx亚太地区销售与市场副总裁杨飞 作序

ARM中国区大学计划经理时昕 作序

系统论述Zynq-7000 SoC体系结构、程序设计及操作系统移植
的经典著作!



THE DESIGN GUIDE: XILINX ALL PROGRAMMABLE ZYNQ-7000 SOC

Xilinx All Programmable Zynq 7000 SoC 设计指南

何宾 编著
He Bin



清华大学出版社

FDA 工 | 程 | 技 | 术 | 丛 | 书 |



THE DESIGN GUIDE: XILINX ALL PROGRAMMABLE ZYNQ-7000 SoC

Xilinx All Programmable Zynq-7000 SoC 设计指南

何 宾 编著
He Bin

清华大学出版社
北京

内 容 简 介

本书系统地论述了 Xilinx ALL Programmable Zynq-7000 SoC 的体系结构与设计方法,全书共 23 章,分为 3 篇。Zynq-7000 基础理论篇介绍了可编程 SoC 设计和 AMBA 协议规范;Zynq-7000 体系结构篇介绍了 Zynq-7000 应用处理单元、可编程逻辑资源、系统互联结构、系统公共资源特性及功能、Zynq 调试和测试子系统、Zynq 平台的启动和配置、Zynq 平台主要外设模块、Zynq 平台描述规范和高级综合工具 HLS;Zynq-7000 设计实践篇介绍了 Zynq 基本处理器系统的建立和运行、添加 AXI IP 到设计、基于定制 IP 实现简单嵌入式系统设计、基于定制 IP 实现复杂嵌入式系统设计、软件和硬件协同调试系统、Zynq 平台配置和启动的实现、基于 Zynq HP 从端口的数据传输实现、基于 Zynq ACP 从端口的数据传输实现、XADC 在 Zynq 平台上的应用、Ubuntu 操作系统在 Zynq 平台上的实现、 μ C/OS-III 操作系统在 Zynq 平台上的实现和 HLS 在 Zynq 嵌入式系统设计中的应用。

全书基于 Xilinx 最新的 Zynq-7000 平台,理论和实践相结合;并配套提供了工程文件及课件,便于自学与教学。本书可作为高等学校计算机与电子信息类专业嵌入式系统相关课程的教学用书,也可作为从事嵌入式系统设计的工程技术人员的参考用书。

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

版权所有,侵权必究。侵权举报电话:010-62782989 13701121933

图书在版编目(CIP)数据

Xilinx All Programmable Zynq-7000 SoC 设计指南/何宾编著. —北京:清华大学出版社,2013.5
(EDA 工程技术丛书)

ISBN 978-7-302-32222-1

I. ①X… II. ①何… III. ①可编程序逻辑器件—系统设计—指南 IV. ①TP332.1-62

中国版本图书馆 CIP 数据核字(2013)第 084568 号

责任编辑:盛东亮

封面设计:李召霞

责任校对:白 蕾

责任印制:李红英

出版发行:清华大学出版社

网 址: <http://www.tup.com.cn>, <http://www.wqbook.com>

地 址:北京清华大学学研大厦 A 座 邮 编:100084

社 总 机:010-62770175 邮 购:010-62786544

投稿与读者服务:010-62776969, c-service@tup.tsinghua.edu.cn

质 量 反 馈:010-62772015, zhiliang@tup.tsinghua.edu.cn

课 件 下 载: <http://www.tup.com.cn>, 010-62795954

印 刷 者:清华大学印刷厂

装 订 者:三河市新茂装订有限公司

经 销:全国新华书店

开 本:185mm×260mm

印 张:37

字 数:896 千字

版 次:2013 年 5 月第 1 版

印 次:2013 年 5 月第 1 次印刷

印 数:1~3000

定 价:79.00 元

把握 All Programmable SoC 技术,推动中国创新事业

新年伊始,获悉北京化工大学何宾教授的《Xilinx All Programmable Zynq-7000 SoC 设计指南》图书即将出版,我感到非常振奋。

从诞生至今,赛灵思一直都是行业的先锋企业,29 年来引领了众多的行业创新。在过去的两年中,赛灵思最为自豪的,就是自 28nm 产品推出以来公司所实现的众多的行业第一,其中包括:

(1) 行业第一个 All Programmable SoC Zynq-7000 系列。该系列的发布为行业带来了软件和硬件全面可编程的 SoC 平台。

(2) 行业第一个 3D IC。该产品的发布使得半导体制造业突破了摩尔定律的限制。已经量产的 Virtex-7 2000T 3D IC 是目前半导体历史上最大的 IC,拥有 68 亿个晶体管。

(3) 行业第一个异构 3D IC。该 3D IC 让 28nm FPGA 功能和 40nm 28G SerDes 收发器功能得以完美集成,成为行业第一个拥有 400G 吞吐能力的单芯片。

(4) 行业第一个把高层次综合能力引入到 All Programmable FPGA 设计流程中,使得 FPGA 工程师能够通过构建基于 C/C++/System C 的模型而转换硬件实现。

(5) 打造行业第一个面向未来 10 年 All Programmable 设计的新一代环境——Vivado Design Suit,把集成和实现的速度提高 4 倍以上,大大提升了工程设计的生产力。

(6) Zynq 加 HLS (赛灵思高层次综合工具),让系统设计师能够基于 All Programmable SoC 实现软硬件的协同设计,大幅度缩短了其产品的上市进程。

所有这些赛灵思所创造的行业领先产品和技术,让赛灵思能够为工程业界提供领先一代的价值优势,帮助工程师们充分释放其创造和创新的巨大潜能。

芯片平台进入门槛一直是局限中国信息行业发展的瓶颈,赛灵思的 28nm All Programmable 产品,尤其是其 Zynq 系列,对于中国的工程师们来说,是一个非常合适且高效的平台,可以支持他们实现精彩纷呈的行业创新。我很高兴中国的工程师能够如此迅速地发现 Zynq All Programmable SoC 的价值,并率先出版拥有大量实例的、理论和实践相结合的中文设计指南。这些图书将加速中国工程师借助 Zynq 平台把创意变成现实,推动中国的创新事业。

更智能的专业化软硬件平台,将是嵌入式系统的未来。而 Zynq All Programmable SoC,就是这样一个专业的、跨学科、跨产业、跨时空和跨职业生涯的强大软硬件平台,不仅将加速中国学术界创新人才培养,而且还将在推动工业界的创新方面扮演重要的角色!

最后,我再次对何教授及参编作者在该书出版上所作的努力表示诚挚的谢意!

杨 飞

Xilinx 亚太地区销售与市场副总裁

2013 年 1 月

紧密结合 ARM 与 FPGA 技术,致力中国智造

首先,恭喜北京化工大学何宾教授写作完成的《Xilinx Zynq-7000 All Programmable SoC 设计指南》图书即将出版。

ARM 公司自 1990 年诞生以来,一直致力于为芯片的设计开发提供领先的半导体知识产权(IP),全球有超过 250 家公司在 ARM 处理器 IP 的基础上开发出了数以百计的各类芯片。ARM 在全球合作伙伴数量(包括世界领先的半导体公司和系统公司)也已经超过了 1000 个。

2012 年,全球领先 FPGA 厂商赛灵思(Xilinx)公司与 ARM 合作,推出的 Zynq-7000 系列产品,完美地将 ARM 双核 Cortex-A9 MPCore 处理系统与赛灵思可扩展的 28nm 可编程逻辑构架结合在一起。该系列采用了业界事实标准的 AMBA 总线作为片上系统的互连总线,可支持双核 Cortex-A9 处理器系统以及可编程逻辑中定制加速器和外设的并行开发。

作为行业内第一款集成了 Cortex-A 系列内核的软、硬件全面可编程的平台,我们非常自豪能够参与和见证这一技术上的创新。同时,我们也非常高兴地看到 FPGA 的用户将可以从 ARM 生态中丰富的软、硬件资源中获益。相信 Zynq-7000 系列的推出,一定会给中国的工程师们提供一个更强大的工作和创新平台。

何宾老师基于该产品撰写的《Xilinx All Programmable SoC 设计指南》,从片上系统(SoC)设计导论开始,继之以 AMBA 片上总线规范以及 Cortex-A9 处理器内核的说明,之后在详细描述 Zynq-7000 系列芯片特点的基础上系统地介绍了如何基于 Zynq 的丰富资源与特性进行软硬件的协调开发。下册则提供了非常丰富的设计实例来说明设计流程和方法。

作为 ARM 大学计划的一员,我非常高兴地看到 Zynq-7000 系列开发平台已经走入了大学,成为大学生们学习和研究的工具。在校大学生是未来优秀工程师的主力军,何宾老师撰写的可编程平台设计指南,把行业内领先的技术带进了大学的课堂,这对优秀人才的培养以及国内半导体技术的发展都有着重大的意义。

相信本书的出版发行定能够对芯片设计、嵌入式系统乃至计算机软硬相关学科的教学与科研提供很好的平台和参考。

时 昕

ARM 中国区大学计划经理

2013 年 1 月

本书是作者历时近一年撰写的反映 Xilinx 最新可编程技术的著作。编写过程中感触颇多,愿与广大读者一起分享这些心得:

(1) 当 Xilinx 将 ARM 公司的双核 Cortex-A9 处理器嵌入到 FPGA 芯片内,并结合最新的 28nm 工艺,制造出全新一代的可编程 SoC 平台后,取名叫 EPP(Extensible Processing Platform,可扩展的处理平台),后来又改成 All Programmable 平台。在这个名字变化的过程中,反映了 Xilinx 给这个最新 Zynq 设计平台的定位——侧重于嵌入式系统的应用,未来的可编程逻辑器件向着嵌入式处理方向发展,未来的嵌入式系统“硬件”和“软件”将根据应用的要求,真正变成 All Programmable(全可编程),即可以在单芯片内设计满足特定要求的硬件平台和相应的软件应用。在这个全可编程的实现过程中,体现着软件和硬件协同设计、软件和硬件协同调试、软件的串行执行和硬件逻辑的并行执行完美结合、未来的嵌入式系统是“积木块”的设计风格等设计思想。这些设计理念将在 Zynq-7000 平台上由理想变成实现。

(2) Zynq-7000 器件是最新半导体技术、计算机技术和电子技术的一个结合体。在一个小小的半导体硅片上却集成了当今最新的信息技术。基于 Zynq-7000 平台进行高性能的嵌入式实现,需要微电子、数字逻辑、嵌入式处理器、计算机接口、计算机体系结构、数字信号处理等相关的知识。Zynq-7000 是一个比较复杂的系统,是对一个设计者的基础理论知识和系统级设计能力的一个真正的考查。在这个平台上实现嵌入式系统的应用,体现着自顶向下的一体化设计理念。

(3) Zynq-7000 平台是非常好的教学平台、科研平台和应用平台。作为教学平台,可以在这个平台上实现全过程的计算机相关课程的教学,使学生可以清楚地看到每个实现的具体过程。这样,学生就可以真正地理解嵌入式系统的内涵;作为科研平台,从事嵌入式相关技术研究人员,可以在这个全开放的平台上,将算法进行高性能的实现。并且,可以在这个平台上实现设计性能分析等研究;作为应用平台,该平台的应用将进一步提高嵌入式系统的灵活性和可靠性、大大降低设计成本,提高产品的市场竞争力。

全书共分 23 章,为了更好地帮助读者学习和掌握 Zynq 平台的设计原理和实现方法,按照 Zynq-7000 基础理论、Zynq-7000 体系结构和 Zynq-7000 设计实践进行了详细的介绍。

(1) Zynq-7000 基础理论篇详细介绍了学习 Zynq-7000 平台需要的基础理论知识。

(2) Zynq-7000 体系结构篇详细介绍了 Zynq-7000 内的处理器系统、可编程逻辑系统、互联结构和外设模块等。

(3) Zynq-7000 设计实践篇,详细介绍了基于 Zynq 全可编程平台的不同设计实例。

本书所给出的设计实例代表着 Zynq 的应用方向,在介绍这些设计实例的过程中,贯穿了很多重要的设计方法和设计思路,这些设计方法和设计思路比设计案例本身更加重要。为了便于读者学习,本书还配套提供了相关设计的完整工程文件及教学课件等资源。

前言

在本书的编写过程中,得到了诸多人士的帮助。彭勃、李宝隆和孙丹阳同学帮助验证了本书的部分案例,并完成了书中部分表格和插图的绘制工作。中科院南京天文光学技术研究所的陆启帅参与了第9章和第10章内容的编写。还要感谢 Xilinx 大学计划、ARM 大学计划、Digilent 公司、Avent 公司和北京麦克泰软件技术有限公司在软件和硬件平台方面给予的大力支持和帮助。

虽然作者花费了大量的精力和时间用于该书的编写,但是由于作者的知识水平有限,书中一定会存在诸多不足之处。在此,也恳请广大读者、同仁对本书提出宝贵的修改意见。

作者联系方式: hebin@mail.buct.edu.cn

编辑联系方式: shengdl@tup.tsinghua.edu.cn

编著者

2013年1月于北京

推荐序(一)	1
推荐序(二)	3
前言	5

第一篇 Zynq-7000 基础理论

第 1 章 可编程 SoC 设计	3
1.1 可编程 SoC 系统	3
1.1.1 软核及硬核处理器	3
1.1.2 可编程 SoC 技术的发展	4
1.1.3 可编程 SoC 技术的特点	5
1.1.4 可编程 SoC 的设计流程	6
1.1.5 可编程 SoC 的开发工具	7
1.2 Xilinx Zynq 平台	13
1.2.1 Zynq 平台的功能	13
1.2.2 处理系统 PS 特性	16
1.2.3 可编程逻辑 PL 特性	21
1.2.4 互联特性	22
1.2.5 Zynq 信号、接口和引脚	23
1.3 Zynq 平台设计方法学	31
1.3.1 使用 PL 实现软件算法的优势	31
1.3.2 设计 PL 加速器	32
1.3.3 PL 加速限制	32
1.3.4 降低功耗	33
1.3.5 实时减负	34
1.3.6 可重配置计算	35

第 2 章 AMBA 协议规范	36
2.1 AMBA 规范导论	36
2.2 AMBA APB 规范	37
2.2.1 AMBA APB 写传输	37
2.2.2 AMBA APB 读传输	39
2.2.3 AMBA APB 错误响应	40
2.2.4 操作状态	41

目录

2.2.5	AMBA3 APB 信号	42
2.3	AMBA AHB 规范	43
2.3.1	AMBA AHB 结构	43
2.3.2	AMBA AHB 操作	44
2.3.3	AMBA AHB 传输类型	46
2.3.4	AMBA AHB 猝发操作	47
2.3.5	AMBA AHB 传输控制信号	53
2.3.6	AMBA AHB 地址译码	54
2.3.7	AMBA AHB 从设备传输响应	55
2.3.8	AMBA AHB 数据总线	58
2.3.9	AMBA AHB 传输仲裁	59
2.3.10	AMBA AHB 分割传输	64
2.3.11	AMBA AHB 复位	67
2.3.12	AHB 数据总线的位宽	67
2.3.13	AMBA AHB 接口设备	69
2.4	AMBA AXI 4 规范	70
2.4.1	AMBA AXI 4 功能	71
2.4.2	AMBA AXI 4 互联结构	79
2.4.3	AXI 4-Lite 功能	80
2.4.4	AXI 4-Stream 功能	81

第二篇 Zynq-7000 体系结构

第 3 章	Zynq-7000 应用处理单元	87
3.1	应用处理单元	87
3.1.1	基本功能	87
3.1.2	系统级视图	89
3.2	Cortex-A9 处理器	90
3.2.1	中央处理器	90
3.2.2	L1 高速缓存	93
3.2.3	存储器管理单元	94
3.2.4	接口	97
3.2.5	NEON	97
3.2.6	性能监视单元	98

3.3	侦听控制单元	98
3.3.1	地址过滤	98
3.3.2	SCU 主设备端口	99
3.4	L2 高速缓存	99
3.4.1	互斥 L2-L1 高速缓存配置	101
3.4.2	高速缓存替换策略	101
3.4.3	高速缓存锁定	102
3.4.4	使能/禁止 L2 高速缓存控制器	103
3.4.5	RAM 访问延迟控制	103
3.4.6	保存缓冲区操作	104
3.4.7	在 Cortex-A9 和 L2 控制器之间的优化	104
3.4.8	预取操作	106
3.4.9	编程模型	106
3.5	片上存储器	107
3.5.1	片上存储器结构	107
3.5.2	片上存储器功能	109
3.6	APU 接口	114
3.6.1	PL 协处理接口	114
3.6.2	中断接口	117
3.7	APU 内的 TrustZone	117
3.7.1	CPU 安全过渡	118
3.7.2	CP15 寄存器访问控制	118
3.7.3	MMU 安全性	119
3.7.4	L1 缓存安全性	120
3.7.5	安全异常控制	120
3.7.6	CPU 调试 TrustZone 访问控制	120
3.7.7	SCU 寄存器访问控制	120
3.7.8	L2 缓存中的 TrustZone 支持	121
3.8	应用处理单元复位	121
3.8.1	复位功能	121
3.8.2	复位后的 APU 状态	122
3.9	功耗考虑	122
3.9.1	待机模式	123
3.9.2	在 L2 控制器内的动态时钟门控	123

目录

3.10	系统地址分配	124
3.10.1	地址映射	124
3.10.2	系统总线主设备	125
3.10.3	I/O 外设	125
3.10.4	SMC 存储器	126
3.10.5	SLCR 寄存器	126
3.10.6	杂项 PS 寄存器	127
3.10.7	CPU 私有总线寄存器	127
3.11	中断	127
3.11.1	中断环境	128
3.11.2	中断控制器的功能	129
3.11.3	编程模型	133
3.12	定时器	134
3.12.1	CPU 私有定时器和看门狗定时器	134
3.12.2	全局定时器	135
3.12.3	系统看门狗定时器	136
3.12.4	三重定时器/计数器	137
3.12.5	I/O 信号	140
3.13	DMA 控制器	141
3.13.1	DMA 控制器结构及特性	141
3.13.2	DMA 控制器功能	144
3.13.3	外部信号	153
3.13.4	寄存器描述	155
3.13.5	用于管理器和命令的指令集参考	156
3.13.6	编程模型参考	157
3.13.7	编程限制	163
3.13.8	DMAC IP 配置选项	164
第 4 章	Zynq-7000 可编程逻辑资源	165
4.1	Zynq-7000 可编程逻辑资源特性	165
4.2	可编程逻辑资源功能	167
4.2.1	CLB、Slice 和 LUT	167
4.2.2	时钟管理	167
4.2.3	块 RAM	168

4.2.4	数字信号处理	169
4.2.5	输入/输出	170
4.2.6	低功耗串行收发器	171
4.2.7	PCI-E 模块	172
4.2.8	模拟/数字转换	173
4.2.9	配置	173
第 5 章	系统互联结构	175
5.1	系统互联功能及特性	175
5.1.1	数据路径	177
5.1.2	时钟域	178
5.1.3	连接性	179
5.1.4	AXI ID	180
5.1.5	寄存器	180
5.2	服务质量	181
5.2.1	基本仲裁	181
5.2.2	高级 QoS	181
5.2.3	DDR 端口仲裁	182
5.3	AXI_HP 接口	182
5.3.1	AXI_HP 接口结构及特点	183
5.3.2	接口数据宽度	186
5.3.3	交易类型	187
5.3.4	命令交替和重新排序	187
5.3.5	性能优化总结	188
5.4	AXI_ACP 接口	188
5.5	AXI_GP 接口	189
5.6	AXI 信号总结	190
5.7	PL 接口选择	194
5.7.1	使用通用主设备端口的 Cortex-A9	194
5.7.2	通过通用主设备的 PS DMA 控制器(DMAC)	194
5.7.3	通过高性能接口的 PL DMA	197
5.7.4	通过 AXI ACP 的 PL DMA	197
5.7.5	通过通用 AXI 从(GP)的 PL DMA	200

目录

第 6 章	系统公共资源特性及功能	201
6.1	时钟子系统	201
6.1.1	时钟系统结构及功能	201
6.1.2	CPU 时钟域	202
6.1.3	时钟编程实例	204
6.1.4	时钟系统内生成电路结构	205
6.2	复位子系统	209
6.2.1	复位系统结构和层次	209
6.2.2	启动流程	210
6.2.3	复位的结果	212
第 7 章	Zynq 调试和测试子系统	213
7.1	JTAG 和 DAP 子系统	213
7.1.1	JTAG 和 DAP 系统功能	215
7.1.2	JTAG 和 DAP 系统 I/O 信号	217
7.1.3	编程模型	217
7.1.4	ARM DAP 控制器	219
7.1.5	跟踪端口接口单元 TPIU	220
7.1.6	Xilinx TAP 控制器	220
7.2	CoreSight 系统结构及功能	221
7.2.1	CoreSight 结构	221
7.2.2	CoreSight 功能	222
第 8 章	Zynq 平台的启动和配置	226
8.1	Zynq 平台启动和配置功能	226
8.2	外部启动要求	227
8.3	BootROM	229
8.3.1	BootROM 功能	229
8.3.2	BootROM 头部	230
8.3.3	启动设备	233
8.3.4	BootROM 多启动和启动分区查找	237
8.3.5	调试状态	238
8.3.6	BootROM 后状态	239
8.4	器件配置接口	242

8.4.1	器件配置接口功能	243
8.4.2	器件配置流程	245
8.4.3	PL 配置	248
8.4.4	寄存器集合	249
第 9 章	Zynq 平台主要外设模块	250
9.1	DDR 存储器控制器	250
9.1.1	DDR 存储器控制器接口及功能	251
9.1.2	AXI 存储器端口接口	253
9.1.3	DDR 核交易调度器	254
9.1.4	DDRC 仲裁	255
9.1.5	DDR 控制器 PHY	256
9.1.6	DDR 初始化和标定	256
9.1.7	纠错码	258
9.2	静态存储器控制器	258
9.2.1	静态存储器控制器接口及功能	259
9.2.2	静态存储器控制器和存储器的信号连接	261
9.3	四-SPI Flash 控制器	262
9.3.1	四-SPI Flash 控制器功能	263
9.3.2	四-SPI 控制器反馈时钟	265
9.3.3	四-SPI Flash 控制器接口	266
9.4	SD/SDIO 外设控制器	267
9.4.1	SD/SDIO 控制器功能	268
9.4.2	SD/SDIO 控制器传输协议	270
9.4.3	SD/SDIO 控制器接口信号连接	273
9.5	通用输入/输出控制器	273
9.5.1	通用输入/输出 GPIO 接口及功能	274
9.5.2	通用输入/输出 GPIO 中断功能	276
9.6	USB 主机、设备和 OTG 控制器	276
9.6.1	USB 控制器接口及功能	277
9.6.2	USB 主机操作模式	281
9.6.3	USB 设备操作模式	282
9.6.4	USB OTG 操作模式	285
9.7	吉比特以太网控制器	285

目录

9.7.1	吉比特以太网控制器接口及功能	286
9.7.2	吉比特以太网控制器接口编程向导	287
9.7.3	吉比特以太网控制器接口信号连接	292
9.8	SPI 控制器	293
9.8.1	SPI 控制器的接口及功能	294
9.8.2	SPI 控制器时钟设置规则	296
9.9	CAN 控制器	296
9.9.1	CAN 控制器接口及功能	297
9.9.2	CAN 控制器操作模式	299
9.9.3	CAN 控制器消息保存	300
9.9.4	CAN 控制器接收过滤器	300
9.9.5	CAN 控制器编程模型	301
9.10	UART 控制器	303
9.11	I ² C 控制器	307
9.11.1	I ² C 速度控制逻辑	308
9.11.2	I ² C 控制器的功能和工作模式	308
9.12	ADC 转换器接口	310
9.12.1	ADC 转换器功能	311
9.12.2	ADC 命令格式	312
9.12.3	供电传感器报警	312
9.13	PCI-E 接口	313
第 10 章	Zynq 平台描述规范	315
10.1	Zynq 平台文件描述规范功能集	315
10.2	微处理器硬件规范	316
10.2.1	通用微处理器硬件规范	316
10.2.2	AXI 系统微处理器硬件规范	318
10.2.3	Zynq-7000 系统微处理器规范实例	319
10.3	微处理器外设规范	322
10.3.1	微处理器规范框架	322
10.3.2	总线接口规范	324
10.3.3	I/O 接口规范	324
10.3.4	选项规范	325
10.3.5	参数规范	327

10.3.6	端口规范	335
10.3.7	设计考虑	337
10.4	外设分析命令	338
10.5	黑盒定义	340
10.6	微处理器软件规范	341
10.6.1	微处理器软件规范格式	341
10.6.2	全局参数	342
10.6.3	实例指定参数	343
10.6.4	MDD/MLD 指定参数	344
10.6.5	OS 指定参数	344
10.6.6	处理器指定参数	344
10.7	微处理器库定义	345
10.7.1	库定义文件	345
10.7.2	MLD 格式规范	345
10.7.3	MLD 参数描述	349
10.7.4	设计规则检查	350
10.7.5	库产生	350
10.8	微处理器驱动定义	351
10.8.1	驱动定义文件	351
10.8.2	MDD 格式规范	351
10.9	Xilinx 板描述格式	353
10.9.1	XBD 格式	354
10.9.2	属性命令	354
10.9.3	本地参数命令及子属性	355
10.9.4	本地端口命令及子属性	355
10.9.5	使用 IO_INTERFACE 关联 IP	356
10.9.6	AXI 系统 XBD 格式	357
第 11 章	高级综合工具 HLS	361
11.1	高级综合工具结构	361
11.1.1	不同的命令对 HLS 综合结果的影响	362
11.1.2	从 C 中提取硬件结构	363
11.2	高级综合工具调度和绑定	365
11.2.1	高级综合工具调度	365

目录

11.2.2	高级综合工具绑定	365
11.3	Vivado HLS 工具的优势	366
11.4	C 代码的关键属性	367
11.4.1	函数	368
11.4.2	类型	369
11.4.3	循环	369
11.4.4	数组	371
11.4.5	端口	372
11.4.6	操作符	373
11.5	HLS 内提供的用于时钟测量的术语	374

第三篇 Zynq-7000 设计实践

第 12 章	Zynq 基本处理系统的建立和运行	377
12.1	使用 BSB 向导生成 Zynq 基本系统	377
12.1.1	Zynq 硬件系统的生成	377
12.1.2	生成 Hello World 应用工程	384
12.1.3	运行 Hello World 应用工程	387
12.2	生成和运行存储器测试工程	389
12.2.1	导入前面的 XPS 设计到 SDK	389
12.2.2	生成存储器测试工程	390
12.2.3	运行存储器测试工程	391
12.2.4	调试存储器测试工程	392
12.3	生成和运行外设测试工程	394
12.3.1	导入前面的 XPS 设计到 SDK	394
12.3.2	生成外设测试工程	394
12.3.3	运行外设测试工程	398
第 13 章	添加 AXI IP 到设计	400
13.1	设计原理	400
13.2	添加 IP 到系统设计	401
13.2.1	创建设计工程	401
13.2.2	添加 GPIO IP 到设计	402
13.2.3	添加 AXI Timer IP 到设计	403