Xilinx大学计划指定教材

高等院校信息技术规划教材

片上可编程系统原理及应用

何宾 编著

清华大学出版社

Xilinx大学计划指定教材

高等院校信息技术规划教材

片上可编程系统原理及应用

何宾 编著

清华大学出版社
北京

内 容 简 介

本书系统全面介绍了Xilinx公司的片上可编程系统的原理及一些典型应用。全书共分7章，内容包括SOPC设计导论、MicroBlaze处理器原理、PowerPC处理器原理、SOPC开发平台结构、SOPC描述规范、操作系统及板级支持包和基于EDK的设计流程。本书所有资料来自Xilinx公司的技术手册、文献和应用案例，充分反映了Xilinx公司片上可编程系统的最新技术和应用成果，可以帮助读者尽快掌握这一最新技术。本书将片上可编程系统的基本原理和应用相结合，易于读者理解与自学。

本书可作为信息类专业大学本科高年级学生和研究生的教学参考用书，也可作为从事片上可编程系统设计的工程技术人员参考用书。

图书在版编目（CIP）数据

片上可编程系统原理及应用 / 何宾编著. —北京：清华大学出版社，2010.1
（高等院校信息技术规划教材）
ISBN 978-7-302-21456-4

Ⅰ. 片…　Ⅱ. 何…　Ⅲ. 单片微型计算机－高等学校－教材　Ⅳ. TP368.1

中国版本图书馆CIP数据核字(2009)第207696号

责任编辑：战晓雷　林都嘉
责任校对：焦丽丽
责任印制：王秀菊

出版发行：清华大学出版社　　**地　　址**：北京清华大学学研大厦A座
http://www.tup.com.cn　　**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
印 刷 者：北京四季青印刷厂
装 订 者：三河市溧源装订厂
经　　销：全国新华书店
开　　本：185×260　　**印　张**：20.75　　**字　数**：489千字
版　　次：2010年1月第1版　　**印　次**：2010年1月第1次印刷
印　　数：1～3000
定　　价：32.00元

序言 foreword

本书是作者在《EDA 原理及应用》教材的基础上编写的针对 Xilinx 可编程逻辑器件高级应用的教材。

现在 FPGA 越来越被广泛地应用在各个领域中。Xilinx 公司将专用的嵌入式处理器 PowerPC 硬核和 MicroBlaze 嵌入式处理器软核嵌入到了 FPGA 芯片中。这种集成了嵌入式处理器的 FPGA 芯片被定义成 FPGA 的平台。这种基于 FPGA 的嵌入式平台提供了一个灵活的解决方案。在这个解决方案中，一个单 FPGA 芯片上提供了大量不同的 IP 软核和硬核资源。这些固件和硬件可以在任何时间进行升级。这种可编程的结构特点，大大缩短了系统的开发时间，而同一平台能应用在很多领域，提高了平台的资源复用率。

片上可编程系统(System-On-a-Programmable Chip，SOPC)技术是 Xilinx 公司在继 FPGA 设计技术后，又一重要的技术应用成果。SOPC 技术的推出对嵌入式系统设计技术产生了深远的影响。可以预见的是，当 SOPC 芯片成本进一步降低，产品性能进一步提高，软件平台操作更进一步简化后，SOPC 技术将广泛地应用在嵌入式系统的设计中。

本书系统全面介绍了 Xilinx 公司的片上可编程系统的原理及一些典型应用。全书共分 7 章，内容包括 SOPC 设计导论、MicroBlaze 处理器原理、PowerPC 处理器原理、SOPC 开发平台结构、SOPC 描述规范、操作系统及板级支持包和基于 EDK 的设计流程。由于该技术在不断地发展，因此今后必要时，会将一些最新的 SOPC 设计技术编入本书。为了使读者进一步掌握 SOPC 的设计技术，作者计划近期内编写并出版《片上可编程系统原理及应用实验教程》一书。

掌握 SOPC 设计技术，重要的是在学习本书基本设计方法的基础上，多在硬件平台上进行实际练习和操作，并完成一个完整 SOPC 系统的设计。这样读者就能够独立地从事片上可编程系统的设计和开发工作。

在本书的编写过程中，使用了 XIlinx 公司的一些技术文档及资

料，在案例部分还参考了 Xilinx 公司技术人员的设计资料，正是由于这些宝贵的设计资料，才使作者能顺利地完成该书的编写工作。作者的本科生和研究生在此期间试读本书，为该书的编写提供了宝贵的意见。

感谢李保敏、朱红林、何军阅读并校对了书稿，并帮助完成了书中一部分表格和插图的绘制工作。李保敏负责第 1 章的编写，朱红林负责第 4 章的编写，何军负责第 7 章的编写，同时，还要感谢 Xilinx 大学计划在软件和硬件上的大力支持。最后，也要对清华大学出版社的编辑和领导的辛勤工作表示感谢。正是由于他们的支持和帮助，使作者能在短时间内完成该书。

虽然作者在翻译部分资料时，尽力保证原文的意思，但是由于作者的能力有限，书中一定会存在不足之处。在此，也恳请广大读者、同仁对本书提出宝贵的修改意见。

何　宾

2009 年 4 月

目录 contents

第 1 章 SOPC 设计导论 …… 1

1.1 SOPC 概述 …… 1
1.1.1 软核及硬核处理器 …… 1
1.1.2 SOPC 技术的发展 …… 2
1.1.3 SOPC 技术特点 …… 3
1.2 SOPC 设计与优化技术 …… 4
1.2.1 SOPC 设计技术 …… 4
1.2.2 通用 SOPC 优化技术 …… 6
1.2.3 专用 SOPC 优化技术 …… 8
1.3 Xilinx 的 SOPC 芯片 …… 10
1.3.1 Spartan-3 系列 FPGA …… 10
1.3.2 Virtex-Ⅱ Pro 系列 FPGA …… 12
1.3.3 Virtex-4 系列 FPGA …… 12
1.3.4 Virtex-5 系列 FPGA …… 13

第 2 章 MicroBlaze 处理器原理 …… 15

2.1 MicroBlaze 处理器结构 …… 15
2.1.1 MicroBlaze 处理器结构概述 …… 15
2.1.2 MicroBlaze 处理器的寄存器 …… 20
2.1.3 MicroBlaze 处理器的虚拟存储器管理 …… 28
2.1.4 MicroBlaze 处理器的事件及处理 …… 35
2.1.5 MicroBlaze 处理器的指令和数据缓存 …… 39
2.1.6 MicroBlaze 处理器的调试和跟踪 …… 41
2.2 MicroBlaze 信号接口 …… 42
2.2.1 PLB 总线接口 …… 42
2.2.2 OPB 总线接口 …… 44
2.2.3 LMB 总线接口 …… 45

2.2.4 FSL 接口 …… 46
2.2.5 XCL 接口 …… 48
2.2.6 调试接口 …… 51
2.2.7 跟踪接口 …… 51
2.3 MicroBlaze 应用二进制接口 …… 53
2.3.1 堆栈规约 …… 53
2.3.2 存储器模型 …… 54
2.3.3 中断和异常句柄 …… 54
2.4 MicroBlaze 指令集结构 …… 55
2.4.1 MicroBlaze 指令类型 …… 55
2.4.2 MicroBlaze 指令集 …… 56

第 3 章 PowerPC 处理器原理 …… 64

3.1 PowerPC 处理器结构 …… 64
3.1.1 PowerPC 处理器体系结构概述 …… 64
3.1.2 PowerPC 软件结构概述 …… 67
3.1.3 PowerPC 寄存器 …… 68
3.2 PowerPC 处理器 I/O 接口 …… 69
3.2.1 时钟和电源管理接口 …… 70
3.2.2 CPU 控制接口 …… 72
3.2.3 复位接口 …… 72
3.2.4 指令侧的 PLB 接口 …… 73
3.2.5 数据侧的 PLB 接口 …… 74
3.2.6 设备控制寄存器接口 …… 76
3.2.7 外部中断控制器接口 …… 78
3.2.8 PPC405 JTAG 调试端口 …… 79
3.2.9 调试接口 …… 80
3.2.10 跟踪接口 …… 81
3.2.11 处理器版本寄存器接口 …… 82
3.2.12 额外的 FPGA 指定信号 …… 83
3.3 PowerPC 处理器 OCM 控制器 …… 83
3.3.1 OCM 控制器特点 …… 84
3.3.2 OCM 控制器的操作 …… 85
3.3.3 OCM 的编程模型 …… 86
3.4 PowerPC 处理器 APU 控制器 …… 88
3.4.1 FCM 指令处理 …… 89
3.4.2 APU 控制器配置 …… 92

第 4 章 SOPC 开发平台结构 …… 94

4.1 设计流程及 EDK 工具概述 …… 94
4.1.1 设计流程概述 …… 94
4.1.2 EDK 工具概述 …… 95
4.1.3 工程建立和管理 …… 102
4.2 平台产生器 …… 103
4.2.1 Platgen 工具的命令格式 …… 103
4.2.2 加载路径 …… 103
4.2.3 输出文件 …… 104
4.2.4 存储器的产生 …… 104
4.3 仿真模型产生器 …… 106
4.3.1 仿真库 …… 106
4.3.2 CompXLib/CompEDKLib 工具 …… 107
4.3.3 仿真模型 …… 107
4.3.4 Simgen 命令格式 …… 108
4.3.5 输出文件 …… 109
4.3.6 存储器初始化 …… 109
4.4 库产生器 …… 110
4.4.1 库产生器命令的选择项 …… 110
4.4.2 加载路径 …… 110
4.4.3 输出文件 …… 112
4.4.4 生成库和驱动 …… 113
4.4.5 中断和中断控制器 …… 115
4.4.6 XMDStub 外设(MicroBlaze 指定) …… 115
4.4.7 STDIN 和 STDOUT 外设 …… 115
4.5 虚拟平台产生器 …… 115
4.5.1 VPgen 命令选项 …… 116
4.5.2 输出文件 …… 116
4.5.3 可使用的模型 …… 116
4.6 平台规范工具 …… 117
4.6.1 PsfUtility 命令选项 …… 117
4.6.2 MPD 的创建 …… 117
4.6.3 PsfUtility 的 DRC 检查 …… 118
4.6.4 HDL 外设定义 …… 119
4.7 版本管理工具 …… 131
4.8 比特流初始化软件 …… 131
4.9 Flash 存储器编程 …… 132

4.9.1 支持的 Flash 硬件 …… 132
4.9.2 编程的先决条件 …… 133
4.9.3 编程对话框 …… 133
4.9.4 定制 Flash 编程 …… 134
4.9.5 可操作的特点和方法 …… 136
4.9.6 使用 Flash 存储器 …… 137
4.10 GNU 编译器工具 …… 137
4.10.1 编译器框架 …… 138
4.10.2 编译器使用及选项 …… 138
4.10.3 MicroBlaze 编译器 …… 145
4.10.4 PowerPC 编译器 …… 151
4.11 GNU 调试器 …… 154
4.11.1 GNU 选项 …… 154
4.11.2 GDB 调试流程 …… 154
4.11.3 MicroBlaze 的 GDB 目标 …… 154
4.11.4 PowerPC 的 GDB 目标 …… 155
4.11.5 控制台模式 …… 155
4.12 Xilinx 微处理器调试器 …… 156
4.12.1 XMD 的使用 …… 157
4.12.2 连接命令选项 …… 160
4.12.3 XMD 内部 TCL 命令 …… 166
4.13 系统 ACE 文件产生器 …… 167
4.13.1 GenACE 模型 …… 167
4.13.2 产生 ACE 文件 …… 169

第 5 章 SOPC 描述规范 …… 173

5.1 微处理器硬件规范 …… 173
5.2 微处理器外设规范 …… 177
5.2.1 MPD 语法 …… 177
5.2.2 总线接口 …… 181
5.2.3 IO 接口 …… 181
5.2.4 选项 …… 181
5.2.5 参数 …… 184
5.2.6 端口 …… 185
5.2.7 保留参数 …… 186
5.2.8 保留的端口连接 …… 186
5.2.9 设计考虑 …… 189
5.3 外设分析命令 …… 191

5.4 黑盒定义 …… 192
5.5 微处理器软件规范 …… 193
5.5.1 MSS 关键字 …… 194
5.5.2 全局参数 …… 195
5.5.3 实例特定参数 …… 196
5.5.4 MDD/MLD 特定参数 …… 197
5.5.5 OS 特定参数 …… 197
5.5.6 处理器特定参数 …… 197
5.6 微处理器库定义 …… 198
5.6.1 库定义文件 …… 198
5.6.2 MLD 格式规范 …… 199
5.6.3 MLD 参数描述 …… 202
5.6.4 设计规则检查 …… 204
5.6.5 库产生 …… 204
5.7 微处理器驱动定义 …… 204
5.8 Xilinx 板描述格式 …… 207
5.8.1 XBD 格式 …… 207
5.8.2 属性命令 …… 208
5.8.3 本地参数命令及子属性 …… 209
5.8.4 本地端口命令及子属性 …… 209
5.8.5 使用 IO_INTERFACE 关联 IP …… 210
5.8.6 使用 IO_INTERFACE 桥接 IP …… 211

第 6 章 操作系统及板级支持包 …… 212

6.1 Xilinx 的微核 …… 212
6.1.1 标准 C 库 …… 213
6.1.2 板级支持包 …… 215
6.1.3 Xilkernel 核 …… 221
6.1.4 LibXil 库 …… 233
6.2 lwIP 库 …… 237
6.2.1 建立硬件系统 …… 238
6.2.2 建立软件系统 …… 238
6.2.3 软件 API …… 242
6.3 VxWorks 操作系统的板级支持包 …… 244
6.3.1 概述 …… 244
6.3.2 使用 XPS 产生 VxWorks6.5 BSP …… 245
6.3.3 VxWorks6.5 BSP …… 246
6.3.4 引导 VxWorks …… 250

6.3.5 缓存、MMU 和 FPU …… 256
6.4 Linux 操作系统下的板级支持包 …… 257
6.4.1 概述 …… 257
6.4.2 开始 Linux …… 258
6.4.3 从 XPS 创建 BSP …… 259
6.4.4 Linux 核配置 …… 261
6.4.5 Linux 设备参考 …… 270

第 7 章 基于 EDK 的设计流程 …… 273

7.1 工程的建立 …… 273
7.1.1 使用 BSP 向导 …… 273
7.1.2 新建工程的结构分析 …… 277
7.1.3 工程的下载 …… 279
7.2 添加 IP 到硬件设计 …… 279
7.2.1 打开工程 …… 279
7.2.2 添加和配置 GPIO 外设 …… 280
7.2.3 产生外部 GPIO 连接 …… 282
7.2.4 添加软件程序并编译 …… 283
7.2.5 设计验证 …… 283
7.3 添加定制的 IP 到系统 …… 285
7.3.1 打开工程 …… 285
7.3.2 产生外设模板 …… 285
7.3.3 创建外设 …… 289
7.3.4 添加和连接外设 …… 291
7.3.5 设计验证 …… 293
7.4 编写应用程序 …… 294
7.4.1 添加 BRAM 控制器和 BRAM …… 294
7.4.2 更新软件应用程序 …… 295
7.4.3 分析目标文件 …… 296
7.4.4 设计验证 …… 298
7.5 使用 SDK 工具 …… 299
7.5.1 添加定时器和中断控制器 …… 299
7.5.2 创建 SDK 软件工程 …… 301
7.5.3 编写中断句柄 …… 303
7.5.4 添加连接脚本 …… 304
7.5.5 验证操作 …… 304

7.6 设计的软件和硬件调试 …… 307
7.6.1 打开工程 …… 308
7.6.2 例化 ChipScope 核 …… 308
7.6.3 启动软件调试器 …… 309
7.6.4 启动 ChipScope Pro 硬件调试器 …… 311
7.6.5 执行 H/S 验证 …… 312

第1章 chapter 1

SOPC设计导论

本章主要对SOPC(System-On-a-Programmable Chip)设计技术进行简要的介绍，在SOPC概述部分介绍软核和硬核处理器，以及SOPC的发展背景和SOPC技术的特点；在SOPC设计方法部分介绍SOPC设计流程、通用SOPC优化技术和专用SOPC优化技术；在SOPC芯片部分介绍Xilinx公司支持SOPC设计的主要芯片的种类和性能。

本章从总体上介绍SOPC设计技术，本章内容对学习后续章节非常重要，读者要掌握本章的内容，以便更好地学习后续章节。

1.1 SOPC概述

基于现场可编程门阵列(Field Programmable Gate Array，FPGA)的SOPC，包含嵌入式的软核或硬核处理器、存储器和硬件加速器。SOPC的出现为设计者提供了设计高性能嵌入式系统和优化系统的条件。

1.1.1 软核及硬核处理器

SOPC嵌入式处理器分为软核和硬核处理器两类。Xilinx和Altera公司都提供了将物理的处理器核集成到FPGA硅片上的硬核处理器产品。一个处理器使用专门的硅片实现称为硬核处理器，比如Altera将ARM922T的硬核集成到Excalibur系列的FPGA芯片中，Xilinx将PowerPC 405硬核集成到Virtex-Ⅱ Pro系列的FPGA芯片中。

软核处理器是通过使用FPGA的通用逻辑实现的。软核处理器通过HDL语言或网表进行描述的。软核处理器必须进行综合才能使用。

在基于软核和硬核处理器的SOPC系统中，本地存储器、处理器总线、内部外设、外设控制器和存储器控制器必须使用FPGA的通用逻辑实现。

表1.1给出了Xilinx公司的软核和硬核处理器的性能。

表 1.1 软核和硬核处理器的性能

处理器	处理器类型	器件类型	速度/MHz	DMIPs
PowerPC405	硬核	Virtex-4	450	680
MicroBlaze	软核	Virtex-Ⅱ Pro	150	123
MicroBlaze	软核	Spartan-3	85	65

1.1.2 SOPC 技术的发展

由于程序地要求嵌入式系统具有更多的功能、更好的性能和灵活性，因此传统上的设计方法已经不适应这种要求。当设计人员试图通过高性能的嵌入式处理器得到更高的性能时，遇到了吞吐量和性能方面的限制，而这种限制源于系统和结构的瓶颈，以及存储器带宽的限制。现在解决问题的方法是“专用”，即对某个嵌入式系统的应用使用专门的解决方法。比如，数字信号处理器 DSP 用于解决某一类专门的数字信号处理。对于一些高容量的应用，设计人员可能还需要专门开发 ASIC 芯片。

现在 FPGA 广泛地应用在各个领域中。因此，很多 FPGA 厂商将专用的嵌入式处理器 PowerPC、ARM 等嵌入到了 FPGA 芯片中。这种集成了嵌入式处理器的 FPGA 芯片被定义成 FPGA 的平台。这种基于 FPGA 的嵌入式平台提供了一个灵活的解决方案。在这个解决方案中，一个单 FPGA 芯片上提供了大量不同的 IP 软核和硬核资源。这些固件和硬件可以在任何时间进行升级。这种可编程的结构特点，大大缩短了系统的开发时间，而同一平台能应用在很多领域，提高了平台的资源复用率。

这种结构同时还使设计人员可以优化系统吞吐量和开发周期，提供前所未有的软件和硬件协同设计的灵活性，这种灵活性主要体现在设计人员能够权衡软件和硬件设计的实现方法。这种协同性不同于传统的嵌入式系统的协同设计，虽然以前也使用软件和硬件的协同设计，但是在实现级别上基本上还是使用大量的分离的设计流程。比如，硬件设计人员制定硬件设计规范，软件设计人员制定软件设计规范。这样就导致对问题截然不同的理解，而且对设计团队提出了很高的要求。

更进一步说，FPGA 平台，即 SOPC 集成了传统的软核和硬核处理器、片上总线、大量不同的 I/O 设备和接口标准、定制的硬件加速处理器，以及混合的定制的总线或点对点的拓扑结构，以提高系统的性能。

在 SOPC 的层次上，FPGA 的应用领域已经大大拓宽了，它不再是传统意义上用于连接不同接口设备的“连接逻辑”。由于 FPGA 的容量和性能不断提高，因此它就逐步地变成嵌入式系统的中心。FPGA 容量不断提高，已经将嵌入式处理器和大量 I/O 集成在 FPGA 芯片内。当 FPGA 发展到 SOPC 的阶段后，设计的复杂度也不断地提高，硬件和软件设计在 FPGA 平台上都显得十分重要。而且由于 FPGA 集成了片上总线和存储器，因此也需要系统设计和系统结构方面的经验。

在 SOPC 阶段，设计已经从以硬件描述语言 HDL 为中心的硬件设计，转换到了以 C 语言进行功能描述为中心。所以就形成了以 C 语言描述 SOPC 的功能，而用 HDL 语言

描述硬件的具体实现方法。这也是和传统的FPGA设计和嵌入式系统设计最大的区别，即软件和硬件的真正的协同设计。

1.1.3 SOPC技术特点

作为新的嵌入式系统的设计平台，使用SOPC进行嵌入式系统设计具有以下几个方面的优点。

1. 定制

基于FPGA的嵌入式系统的设计人员可以很灵活地选择所要连接的外设和控制器。因此，设计人员可以设计出一个独一无二的外设，这个外设可以直接和总线连接。对于一些非标准的外设，设计人员很容易地使用FPGA嵌入式平台实现。比如，设计人员很容易地在FPGA平台上设计出具有10个UART接口的嵌入式系统。因此，在FPGA系统中，像这样类似的配置是很容易实现的。

2. 延缓过时

一些公司，特别是为军方提供产品的那些公司，它们产品的供货周期常常比标准电子产品的周期要长。电子元器件的过时(停产)是一个非常严重的问题，会导致这些公司无法继续提供其产品。由于软核处理器的HDL源代码可以通过购买得到，因此基于FPGA的软核处理器是一个非常好的解决方案，它可以充分地满足产品长期供货的要求。

3. 降低元件成本

由于基于FPGA平台的嵌入式系统的功能多样性，以前需要用很多元件才能实现的系统，现在可以使用一个FPGA芯片实现。比如，辅助I/O芯片或协处理器与现有的处理器之间的连接。减少在设计中所使用的元件的数量，不但可以降低元件的成本，而且可以大大缩小电路板的尺寸。

4. 硬件加速

选择基于FPGA的SOPC的一个重要的原因就是，SOPC能在硬件和软件之间进行权衡，使嵌入式系统达到最大的效率和性能。比如，当算法是嵌入式系统软件性能的瓶颈时，一个使用FPGA定制的协处理器引擎能用来实现算法，这个协处理器通过专用的，低延迟的通道与嵌入式处理器连接。使用现代的硬件设计工具，很容易地将软件瓶颈转向硬件处理。

SOPC的出现给嵌入式系统设计带来了非常多的优点，但是由于采用基于FPGA的嵌入式平台，这个平台集成了软件和硬件的平台设计工具，因此设计比较复杂。FPGA的嵌入式的软件设计工具比标准的传统的嵌入式系统的软件设计要新，软件设计工具相对来说还不成熟。但是随着SOPC技术的进一步发展，这个问题将会解决。芯片的成本也是一个问题，采用专用的嵌入式平台比采用基于FPGA的嵌入式平台成本要低，但是

随着制造工艺的不断更新和 SOPC 芯片的成本的降低，相信在不久的将来，SOPC 成本甚至还会低于专用的嵌入式平台。

1.2 SOPC 设计与优化技术

SOPC 设计与优化技术在很大程度上影响着 SOPC 的性能，因此，在基于 FPGA 平台进行 SOPC 设计与优化时，必须正确掌握设计方法和优化方法。下面将对其进行详细介绍。

1.2.1 SOPC 设计技术

SOPC 设计技术不同于传统的嵌入式系统设计流程和设计方法，其设计是软件和硬件的协同设计，同时又是基于软件为中心的设计技术。下面首先介绍 Xilinx 的 SOPC 设计流程，然后介绍以软件为中心的 SOPC 设计技术。

1. SOPC 设计流程

正如前面所说，基于 SOPC 的嵌入式系统的设计需要软件和硬件的协同设计。下面给出 Xilinx 公司使用 XPS(Xilinx Platform Studio)进行嵌入式系统设计的流程。

Xilinx 公司的 XPS 工具包用于开发基于 FPGA 平台的嵌入式系统，从图 1.1 可以看出该工具支持传统的硬件和嵌入式软件的设计流程。

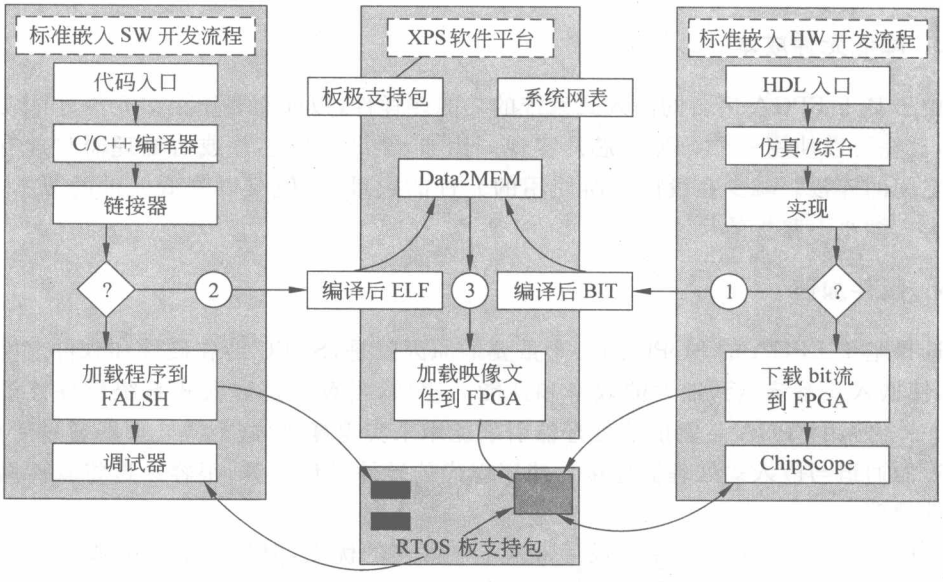

图 1.1 Xilinx 的 SOPC 设计流程

XPS 支持硬核 PowerPC 和软核 Microblaze 处理器。并将设计的导入、创建和 IP 核定制进行了流水化的处理。由于 XPS 知道平台 FPGA 的硅片属性和选项，能自动地为

其外设生成软件驱动、测试代码和创建板级支持包BSP(Board Support Package)。这些BSP是常用的实时操作系统RTOS(Real-Time Operating System),比如VxWorks和嵌入式Linux提供的设备驱动。

XPS的设计流程就是一个软件和硬件协同处理和设计的过程。软件流程完成C语言代码的编写、编译和链接的过程。硬件流程完成HDL设计输入、综合、仿真和实现的过程。XPS提供了一个Data2MEM工具,该工具能将C语言生成的ELF(Executable and Linkable Format)文件代码插入到生成后的FPGA的比特流文件中,将其生成能够下载到FPGA中,并能启动映像文件。通过这个过程设计人员能够使软件开发和调试进行实时处理,而不需要额外的时间开销。

Xilinx的JTAG连接技术,完成FPGA的下载、FPGA的调试、C代码的下载和软件的调试。

XPS集成了软件和硬件调试工具,使它们之间可以相互触发,这使得嵌入式系统内部变成"可见",使嵌入式设计者能很快地找到和发现问题,而无需知道这个问题是软件还是硬件产生的。

2. 基于软件的设计方法

SOPC设计中,一个好的开发工具提供对目标平台合理的抽象,而这个抽象对设计人员来说是比较容易理解的。硬件抽象使得软件开发人员不需要从应用程序的开发转向真实的硬件实现,事实上对于软件人员来说这也是不可能的事情。但是软件开发人员在设计程序的时候开始将并发性和基于消息驱动的硬件概念融入到程序设计中。

这个抽象允许软件设计人员创建、测试和调试应用程序,同时促进开发人员使用程序设计方法使得目标系统达到最大的性能。同时,好的开发工具还提供了将原始的高级描述转换成优化过的低级的目标系统可加载和执行的代码。

为了达到这两个要求,为自动的基于FPGA平台的硬件生成工具主要目的是自动编译和优化问题,提出编程的抽象模型、编程的方法。这些工具目的就是建立面向软件的设计经验。

面向软件的编程、仿真和调试工具提供了对FPGA平台的合理的抽象,允许系统设计人员在原型设计阶段,开始应用的开发、实验,而不需要专门的硬件知识,这一点对于原型设计阶段是非常重要的。如图1.2所示,通过使用软件到硬件的设计方法和工具,传统的软件和硬件设计流程会得到极大地改善。但是并不是说不需要硬件的技巧。事实上,一个完整的和优化过的系统中只使用软件知识是不可能实现的。通过软件和硬件设计技巧和使用现代设计工具就能很快地建立工作原型。

面向软件的设计流程一个非常重要的特点就是在最合理的平台资源中使用软件来描述设计规范。如果最合适的平台资源是微处理器,那么事情就比较简单,只需要针对这个处理器进行交叉编译(交叉编译就是在一个平台上生成另一个平台上的可执行代码),但是如果是FPGA的话,传统的设计流程要求用HDL语言重新书写RTL级的描述,那是一件既耗时,又容易出现设计错误的事情。但是使用面向软件的设计流程,只需要对最初的设计语言进行简单的一些修改,而不需要关心目标系统的资源。