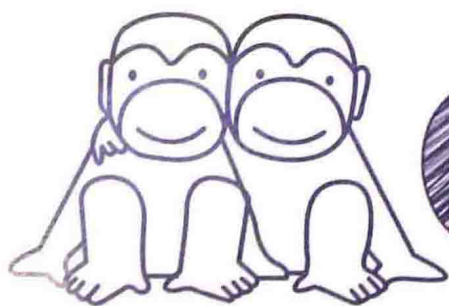


自己动手写 CPU

流水线、指令集、总线……阅读本书，那都不是事儿！

本书手把手地教您从零实现一个具有五级流水线、兼容MIPS32体系结构、带有Wishbone总线接口的处理器，并为其移植了嵌入式操作系统 μ C/OS-II，打造属于您自己的独一无二的计算机系统。

雷思磊 著



CPU



电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
http://www.phei.com.cn

自己动手写 CPU

雷思磊 著



電子工業出版社
Publishing House of Electronics Industry
北京·BEIJING

内 容 简 介

本书使用 Verilog HDL 设计实现了一款兼容 MIPS32 指令集架构的处理器——OpenMIPS。OpenMIPS 处理器具有两个版本，分别是教学版和实践版。教学版的主要设计思想是尽量简单，处理器的运行情况比较理想化，与教科书相似，便于使用其进行教学、学术研究和讨论，也有助于学生理解课堂上讲授的知识。实践版的设计目标是能完成特定功能，发挥实际作用。

全书分为三篇。第一篇是理论篇，介绍了指令集架构、Verilog HDL 的相关知识。第二篇是基础篇，采用增量模型，实现了教学版 OpenMIPS 处理器。首先实现了仅能执行一条指令的处理器，从这个最简单的情况出发，通过依次添加，实现逻辑操作指令、移位操作指令、空指令、移动操作指令、算术操作指令、转移指令、加载存储指令、协处理器访问指令、异常相关指令，最终实现了教学版 OpenMIPS 处理器。第三篇是进阶篇，通过为教学版 OpenMIPS 添加 Wishbone 总线接口，从而实现了实践版 OpenMIPS 处理器，并与 SDRAM 控制器、GPIO 模块、Flash 控制器、UART 控制器、Wishbone 总线互联矩阵等模块组成一个小型 SOPC，然后下载到 FPGA 芯片以验证实现效果，最后为实践版 OpenMIPS 处理器移植了嵌入式实时操作系统 $\mu\text{C}/\text{OS-II}$ 。

本书适合计算机专业的学生、FPGA 开发人员、处理器设计者、嵌入式系统应用开发工程师、MIPS 平台开发人员以及对处理器内部的实现感兴趣的读者阅读，也可以作为高等院校计算机原理、计算机体系结构等课程的实践参考书。

未经许可，不得以任何方式复制或抄袭本书之部分或全部内容。
版权所有，侵权必究。

图书在版编目 (CIP) 数据

自己动手写 CPU / 雷思磊著. —北京：电子工业出版社，2014.9
ISBN 978-7-121-23950-2

I. ①自… II. ①雷… III. ①微处理器—系统设计 IV. ①TP332

中国版本图书馆 CIP 数据核字 (2014) 第 173014 号

策划编辑：孙学瑛

责任编辑：徐津平

特约编辑：顾慧芳

印 刷：北京中新伟业印刷有限公司

装 订：三河市鹏成印业有限公司

出版发行：电子工业出版社

北京市海淀区万寿路 173 信箱 邮编 100036

开 本：787×980 1/16 印张：34.75 字数：823 千字

版 次：2014 年 9 月第 1 版

印 次：2014 年 9 月第 1 次印刷

印 数：3000 册 定价：99.00 元（含光盘 1 张）

凡所购买电子工业出版社图书有缺损问题，请向购买书店调换。若书店售缺，请与本社发行部联系，联系及邮购电话：(010) 88254888。

质量投诉请发邮件至 zltz@phei.com.cn，盗版侵权举报请发邮件至 dbqq@phei.com.cn。

服务热线：(010) 88258888。

自己动手写处理器？

自己动手写处理器！

没错，您手上拿着的就是一本介绍如何实现处理器的书，通过阅读本书，您可以实现世界上独一无二、独属于您的处理器。

吹牛？

噢，No，理工科学生不打诳语。

不信？

……那就请您阅读本书。

写作背景

自 1971 年世界上第一款单芯片微处理器 4004 诞生已逾 40 多年，使用“日新月异”来形容这 40 多年处理器的发展变化亦不为过，无论是速度、集成度，还是架构等许多方面都有了前人难以想象的变化。不过可惜的是，处理器设计制造一直都是高科技行业，轻易无法涉足。大多数人对处理器的直观印象就是一个银白色的小芯片，有许多管脚，至于里面是如何工作的，则不甚了解，更不用说自己制作处理器了。

幸运的是，在处理器发展的同时，可编程逻辑器件也在持续发展。可编程逻辑器件不仅是技术的革新，也带来了观念的革新、设计流程的革新。如今通过编写代码可以在可编程逻辑器件上实现十分复杂的电路设计，比如处理器。于是，普通大众也能有机会了解处理器内部的实现原理，甚至参与处理器的设计、研发。

实际上，目前已经有很多可以下载到可编程逻辑器件上运行的处理器，这些处理器称为软核处理器，比如：NiosII、OR1200、LEON3、OpenSparc 等，这些软核处理器有的是开源的，有的不是开源的。笔者在前期深入阅读了几款开源软核处理器的代码，包括：OC8051、OR1200、LEON3。其中，OC8051 是 OpenCores 提供的一款 8 位两级流水线处理器，与 Intel 8051 兼容，是 CISC (Complex Instruction Set Computer) 类型，采用 Verilog HDL 编写代码。OR1200 是 OpenCores 提供的一款 32 位五级流水线处理器，是 RISC (Reduced Instruction Set Computer) 类型，也采用 Verilog HDL 编写代码。LEON3 是由 Gaisler Research 公司设计发布的一款 32 位七级流水线处理器，也是 RISC 类型，但采用的是 VHDL 编写代码。

通过阅读上述开源软核处理器的代码，一方面消除了笔者对处理器的神秘印象，另一方面也激发了笔者强烈的创作冲动，陆游曾言：纸上得来终觉浅，绝知此事要躬行。是啊！为

何不自己也写一个处理器？于是世间又多了一款开源处理器 OpenMIPS。OpenMIPS 是具有哈佛结构的 32 位五级流水线标量处理器，兼容 MIPS32 体系结构，这样可以使用现有的 MIPS 编译开发环境。它分为教学版和实践版两个版本，每个版本都使用 VHDL 和 Verilog HDL 两种语言编写。

本书以 Verilog HDL 编写的版本为例，详细介绍了 OpenMIPS 从无到有、从小到大、一步一步成长完善的过程。

写作目的

● 撕掉处理器贴着的“高大上”标签

处理器一贯被人们贴上“高大上”、“高科技”诸如此类的标签，贴标签的原因就在于人们对此不了解，越不了解，越觉得神秘，就越不想了解，如此，“高大上”的标签算是贴牢了。本书的目的之一就是帮助读者改变这种固有的偏见，从本质上认识处理器，撕掉这些标签。简单来说，处理器的作用就是识别 0、1 编码，也就是识别指令，据此进行各种运算和数据处理。处理器只能计算小学数学课堂上讲授的四则运算，再加上一些并不复杂的与、或、非等逻辑运算，其余诸如平方、开方、微分、积分等都是做不了的。是不是很简单？

● 对现有处理器相关书籍的补充

翻看现有的处理器相关书籍，会有两个体会：一个体会是大多数书籍讲授的理论部分太多，实践部分太少。过多的理论、过少的实践，会使得读者知其然，不知其所以然；第二个体会是有少部分书籍讲授处理器的实现，但是介绍的方式不太易懂，读者需要对处理器有一定了解之后才能阅读此类书籍。因此，笔者想结合 OpenMIPS 介绍处理器实现，本书完全按照 OpenMIPS 的实现过程讲解，从零起步，遇到什么问题就解决什么问题，笔者认为这样易于理解。

● 抛砖引玉

高手在民间，此言不虚，笔者写作此书的第三个目的就是抛砖引玉，希望有更多人士能够参与维护和改进 OpenMIPS，为其添加更多的功能，或者改善性能。当然，也希望出现更多类似的软核处理器，这样，大家可以相互学习、相互探讨、取长补短、共同进步。

适合谁读

适合对处理器内部的实现有着强烈好奇心的朋友阅读，通过本书介绍的 OpenMIPS 处理器的实现过程，您将全方位地了解 32 位 RISC 处理器的内部设计。

适合不满足于教科书的同学阅读，本书可以作为您的实践参考书，帮助您理解书本上抽象的概念，同时培养动手能力。

适合正在从事软核处理器开发、设计的朋友阅读，本书将给您一些经验、一些好的方法，帮助您事半功倍。

适合正在从事嵌入式开发的朋友阅读，本书对处理器的一些介绍，将有助于嵌入式开发。

内容安排

全书共 15 章，分为三篇，第一篇是理论篇，包含第 1、2 章，介绍了指令集架构、Verilog HDL 的相关知识。第二篇是基础篇，包含第 3~11 章，采用增量模型，实现了教学版 OpenMIPS 处理器。第三篇是进阶篇，包含第 12~15 章，实现了实践版 OpenMIPS 处理器，并为其移植了嵌入式实时操作系统 $\mu\text{C}/\text{OS-II}$ 。每章的主要内容如下。

● 第一篇 理论篇

第 1 章给出了计算机的简单组成模型、简单使用模型，对比了 RISC 与 CISC，说明了指令集架构的作用，并列举了目前几种主要的指令集架构，由于 OpenMIPS 采用的是 MIPS32 指令集架构，所以本章后半部分重点介绍了 MIPS32 指令集架构。

第 2 章介绍了 FPGA、Verilog HDL 的基础知识，FPGA 是可编程逻辑器件的一种，本书的实践版 OpenMIPS 处理器就将在 FPGA 上运行。

● 第二篇 基础篇

第 3 章介绍了教学版 OpenMIPS 处理器的设计蓝图，包括设计目标、处理器接口，以及最终完成时组成 OpenMIPS 的各个模块的作用，力图使读者有一个整体认识。另外，本章还详述了 OpenMIPS 处理器的实现方法。

第 4 章实现了 OpenMIPS 处理器的第一条指令 ori，之所以选择这条指令作为我们实现的第一条指令，就是因为它足够简单，指令 ori 用来实现逻辑“或”运算，通过这条简单指令的实现，初步建立了 OpenMIPS 的五级流水线结构，后续章节实现其余指令的时候，都是在这个初步建立的流水线结构基础上进行扩充的。本章还建立了用于测试的小型 SOPC，并通过 ModelSim 仿真验证 ori 指令、五级流水线实现的正确与否。

第 5 章讨论并解决了流水线数据的相关问题，然后修改第 4 章的 OpenMIPS，添加实现了 MIPS32 指令集架构中定义的逻辑、移位操作与空指令。

第 6 章介绍添加实现了 MIPS32 指令集架构中定义的移动操作指令。

第 7 章介绍添加实现了 MIPS32 指令集架构中定义的算术操作指令。

第 8 章介绍添加实现了 MIPS32 指令集架构中定义的转移指令，OpenMIPS 支持延迟转移。

第 9 章介绍添加实现了 MIPS32 指令集架构中定义的加载存储指令。

第 10 章介绍添加实现了 MIPS32 指令集架构中定义的协处理器 CP0，以及协处理器访问指令。

第 11 章介绍添加实现了 MIPS32 指令集架构中定义的异常相关指令，并实现了异常处理。

在每一类指令的实现过程中，都是先介绍该类指令的格式、作用和用法，然后介绍实现思路，接着通过修改代码实现该类指令，最后，编写测试程序，使用仿真的方式验证实现的正确性。

● 第三篇 进阶篇

第 12 章介绍在教学版 OpenMIPS 处理器的基础上，通过添加 Wishbone 总线接口模块，实现了实践版 OpenMIPS 处理器。

第 13 章讲述设计实现了基于实践版 OpenMIPS 处理器的小型可编程片上系统 SOPC 的整个过程。该 SOPC 包括 GPIO、UART 控制器、Flash 控制器、SDRAM 控制器等模块，这些模块都具有 Wishbone 总线接口，与 OpenMIPS 处理器一起挂接在 Wishbone 总线互联矩阵上。

第 14 章将第 13 章实现的小型 SOPC 下载到实际的硬件平台上，编写测试程序，验证实践版 OpenMIPS 处理器实现的正确性。

第 15 章介绍了嵌入式实时操作系统 $\mu\text{C}/\text{OS-II}$ ，并将其移植到本书设计的 OpenMIPS 处理器上，进一步验证了实践版 OpenMIPS 处理器实现的正确性，也为 OpenMIPS 处理器发挥实际作用奠定了基础。

本书特色

● 从无到有、从小到大，介绍一款处理器的成长过程

在本书之前已有介绍软核处理器实现的书籍，这些书在介绍实现方法时有一个共同点：一次考虑所有的指令、所有的情况，然后给出代码。笔者认为这不是读者易于接受的一种方法，而且这也可能不是作者实现处理器时采用的方法。在本书中，笔者借鉴了软件开发中的“增量模型”的概念，使用了一种完全不同的实现方法：先考虑最简单的情况，给出代码，然后考虑稍微多一点的情况，修改、补充代码，随着考虑情况的增多，不停地修改、补充代码，最终，实现需求的两个版本。

● 教学、实践兼顾

OpenMIPS 处理器分为教学版和实践版两个版本。

教学版的主要设想是尽量简单，比如：在一个时钟周期内可以取到指令，完成存储、加载数据，这样处理器的运行情况（比如：流水线的运行）就比较理想化，与教科书相似，代码也很清晰简单，便于使用其进行教学、学术研究和讨论，也有助于读者理解教科书上讲授的计算机的原理、计算机体系结构等知识。

实践版的主要设计思想是使 OpenMIPS 成为一个实际可用的处理器，能够下载到可编程逻辑器件上，运行实际有用的程序。为此，添加了 Wishbone 总线接口，这样就能方便地利用各种已有的 SDRAM、Flash、GPIO、UART、LCD 等模块控制器，组成一个 SOPC，完成特定功能，进一步还可为其移植操作系统。

光盘内容

本书附带光盘提供了 OpenMIPS 的所有源代码，以及一些开发工具，详情如下。

- Code 文件夹

提供了本书每一章涉及的 OpenMIPS 源代码、测试程序。

- Tools 文件夹

提供了 GNU 工具链的安装文件，以及一个小工具 Bin2Mem.exe，该工具用来将二进制数文件转化为可以用于 ModelSim 仿真的格式。

- Doc 文件夹

提供了本书使用的一些 IP 核的说明手册，包括 UART 控制器、SDRAM 控制器、GPIO 模块等。还提供了 FPGA 开发平台 DE2 的说明手册。

- DE2 文件夹

提供了用来将程序写入 DE2 上 Flash 芯片的工具，在第 14、15 章会用到。

致谢

感谢 OC8051、OR1200、LEON3 的开发者，正是你们的辛苦工作、无私奉献，使得我们有机会领略、学习这些优秀的作品，向你们致敬！

笔者是第二次与博文视点合作，一如既往的敏捷、迅速、干练，在此特别感谢孙学瑛老师，孙老师以专业的眼光审阅了全书，提出了许多宝贵意见，为本书的顺利出版耗费了不少心力。

感谢我的好友张世伟老师，为我提供了 DE2 开发平台。

最后，感谢我的爸爸、妈妈、姐姐、姐夫，以及可爱的外甥女，任何成绩的取得都离不开家人的身影，谢谢你们！

笔者学识有限，尽管对本书已通读数次，但仍不能保证书中无一纰漏，欢迎读者朋友对本书提出批评、建议，可以通过邮箱 leishangwen@163.com 与笔者交流。

写作体会

在实现 OpenMIPS 处理器的过程中，笔者深刻体会到“罗马非一日建成”这句话，外表看起来巨大、庞杂的罗马，也是通过人们一步一步、一天一天、一点一点建成的，处理器也是如此，读者首先不要被处理器的神秘吓到，从最简单的地方入手，逐步增加功能、完善设计，一行代码一行代码地书写，不仅要有实现处理器的远大目标，还要确立切实可操作的短期目标，比如本周实现除法指令，下周实现转移指令，诸如此类，等有一天你突然回头，会发现，原来已经走了那么远，实现了那么多功能。李白有诗云：两岸猿声啼不住，轻舟已过万重山。当是此意。

处理器实现了，但要把它的实现过程明白地表达出来，让读者理解，则又是一件难事。笔者从开始写作到最终完稿的这一过程中，一直承受着巨大的煎熬，几度欲放弃写作，所幸的是最终坚持了下来。

最后，我想对各位读者说：

一个人的旅行是孤单的
一个人的冬季是寒冷的
但是
一个人的处理器是骄傲的
让我们骄傲一次

雷思磊

2014年8月

目 录

第一篇 理论篇

第 1 章 处理器与 MIPS	2	2.2.3 布局布线	20
1.1 计算机的简单模型	2	2.2.4 下载	20
1.1.1 计算机的简单组成模型	2	2.2.5 仿真	20
1.1.2 计算机的简单使用模型	3	2.2.6 工具介绍	21
1.2 架构与指令集	4	2.3 Verilog HDL 简介	21
1.2.1 CISC 与 RISC	4	2.4 Verilog HDL 中模块的结构	22
1.2.2 主要的几种 ISA	5	2.5 Verilog HDL 基本要素	24
1.3 MIPS 指令集架构的演变	6	2.5.1 常量	24
1.4 MIPS32 指令集架构简介	9	2.5.2 变量声明与数据类型	24
1.4.1 数据类型	9	2.5.3 向量	26
1.4.2 寄存器	9	2.5.4 运算符	26
1.4.3 字节次序	10	2.6 Verilog HDL 行为语句	29
1.4.4 指令格式	11	2.6.1 过程语句	29
1.4.5 指令集	11	2.6.2 赋值语句	31
1.4.6 寻址方式	13	2.6.3 条件语句	32
1.4.7 协处理器 CP0	13	2.6.4 循环语句	34
1.4.8 异常	14	2.6.5 编译指示语句	35
1.5 本书的目标与组织方式	14	2.6.6 行为语句的可综合性	37
第 2 章 可编程逻辑器件与 Verilog HDL	16	2.7 电路设计举例	38
2.1 可编程逻辑器件概述	16	2.8 仿真	41
2.2 基于 PLD 的数字系统设计流程	18	2.8.1 系统函数	42
2.2.1 设计输入	19	2.8.2 Test Bench	43
2.2.2 综合	20	2.8.3 ModelSim 仿真	45
		2.9 本章小结	48

第二篇 基础篇

第 3 章 教学版 OpenMIPS 处理器		3.1.1 设计目标	50
蓝图	50	3.1.2 五级流水线	50
3.1 系统设计目标	50	3.1.3 指令执行周期	52

3.2	教学版 OpenMIPS 处理器接口	53	5.2	OpenMIPS 对数据相关问题的 解决措施	111
3.3	文件说明	54	5.3	测试数据相关问题的解决效果	115
3.4	实现方法	55	5.4	逻辑、移位操作与空指令说明	115
第 4 章	第一条指令 ori 的实现	58	5.5	修改 OpenMIPS 以实现逻辑、 移位操作与空指令	119
4.1	ori 指令说明	58	5.5.1	修改译码阶段的 ID 模块	120
4.2	流水线结构的建立	59	5.5.2	修改执行阶段的 EX 模块	127
4.2.1	流水线的简单模型	59	5.6	测试程序 1——测试逻辑 操作实现效果	129
4.2.2	原始的 OpenMIPS 五级 流水线结构	60	5.7	测试程序 2——测试移位 操作与空指令实现效果	130
4.2.3	一些宏定义	62	5.8	小结	131
4.2.4	取指阶段的实现	63	第 6 章	移动操作指令的实现	132
4.2.5	译码阶段的实现	65	6.1	移动操作指令说明	132
4.2.6	执行阶段的实现	74	6.2	移动操作指令实现思路	133
4.2.7	访存阶段的实现	78	6.2.1	新的数据相关情况的 解决	135
4.2.8	回写阶段的实现	81	6.2.2	系统结构的修改	136
4.2.9	顶层模块 OpenMIPS 的实现	81	6.3	修改 OpenMIPS 以实现移动 操作指令	137
4.3	验证 OpenMIPS 实现效果	85	6.3.1	HI、LO 寄存器的实现	137
4.3.1	指令存储器 ROM 的实现	85	6.3.2	修改译码阶段的 ID 模块	138
4.3.2	最小 SOPC 的实现	87	6.3.3	修改执行阶段	141
4.3.3	编写测试程序	88	6.3.4	修改访存阶段	146
4.3.4	建立 Test Bench 文件	89	6.3.5	修改回写阶段	149
4.3.5	使用 ModelSim 检验 OpenMIPS 实现效果	90	6.3.6	修改 OpenMIPS 顶层 模块	149
4.4	MIPS 编译环境的建立	92	6.4	测试程序	150
4.4.1	VisualBox 的安装与设置	93	第 7 章	算术操作指令的实现	152
4.4.2	GNU 工具链的安装	96	7.1	简单算术操作指令说明	153
4.4.3	使用 GNU 工具进行编译	97	7.2	简单算术操作指令实现思路	156
4.4.4	使用 GNU 工具进行链接	99	7.3	修改 OpenMIPS 以实现简单 算术操作指令	157
4.4.5	得到 ROM 初始化文件	102	7.3.1	修改译码阶段的 ID 模块	157
4.4.6	编写 Makefile 文件	103			
4.5	第一条指令实现小结	105			
第 5 章	逻辑、移位操作与空指令 的实现	107			
5.1	流水线数据相关问题	107			

7.3.2 修改执行阶段的 EX 模块 ..	164	8.3 转移指令实现思路	210
7.4 测试简单算术操作指令		8.3.1 实现思路	210
实现效果	171	8.3.2 数据流图的修改	210
7.5 流水线暂停机制的设计		8.3.3 系统结构的修改	211
与实现	174	8.4 修改 OpenMIPS 以实现	
7.5.1 流水线暂停机制的设计	174	转移指令	212
7.5.2 流水线暂停机制的实现	175	8.4.1 修改取指阶段的 PC 模块 ..	212
7.6 乘累加、乘累减指令说明	180	8.4.2 修改译码阶段	213
7.7 乘累加、乘累减指令实现思路	181	8.4.3 修改执行阶段的 EX 模块 ..	223
7.8 修改 OpenMIPS 以实现乘		8.4.4 修改 OpenMIPS 模块	225
累加、乘累减指令	182	8.5 测试转移指令的实现效果	225
7.8.1 修改译码阶段的 ID 模块 ..	182	8.5.1 测试跳转指令	225
7.8.2 修改执行阶段的 EX 模块 ..	184	8.5.2 测试分支指令	227
7.8.3 修改 EX/MEM 模块	189	第 9 章 加载存储指令的实现	230
7.8.4 修改 OpenMIPS 模块	190	9.1 加载存储指令说明	230
7.9 测试乘累加、乘累减指令		9.1.1 加载指令 lb、lbu、lh、	
实现效果	190	lhu、lw 说明	230
7.10 除法指令说明	191	9.1.2 存储指令 sb、sh、sw	
7.11 除法指令实现思路	192	说明	231
7.11.1 试商法	192	9.1.3 加载存储指令用法示例	232
7.11.2 实现思路	193	9.1.4 加载指令 lwl、lwr 说明	233
7.11.3 系统结构的修改	193	9.1.5 存储指令 swl、swr 说明	235
7.12 修改 OpenMIPS 以实现		9.2 加载存储指令实现思路	238
除法指令	194	9.2.1 数据流图的修改	239
7.12.1 增加 DIV 模块	194	9.2.2 系统结构的修改	240
7.12.2 修改译码阶段的 ID 模块 ..	199	9.3 修改 OpenMIPS 以实现加载	
7.12.3 修改执行阶段的 EX		存储指令	240
模块	200	9.3.1 修改译码阶段	240
7.12.4 修改 OpenMIPS 模块	204	9.3.2 修改执行阶段	247
7.13 测试除法指令实现效果	204	9.3.3 修改访存阶段	249
7.14 数据流图的修改	205	9.3.4 修改 OpenMIPS 顶层模块 ..	260
第 8 章 转移指令的实现	206	9.4 修改最小 SOPC	261
8.1 延迟槽	206	9.4.1 添加数据存储器 RAM	262
8.2 转移指令说明	207	9.4.2 修改最小 SOPC	264
		9.5 测试程序	265

9.6	链接加载指令 ll、条件存储指令 sc 说明	267	10.6.2	修改执行阶段	305
9.7	ll、sc 指令实现思路	269	10.6.3	修改访存阶段	311
9.7.1	ll、sc 指令的实现	269	10.6.4	修改 OpenMIPS 模块	314
9.7.2	数据流图的修改	270	10.7	测试程序	314
9.7.3	系统结构的修改	271	第 11 章	异常相关指令的实现	316
9.8	修改 OpenMIPS 以实现 ll、sc 指令	271	11.1	MIPS32 架构中定义的异常类型	316
9.8.1	LLbit 寄存器的实现	271	11.2	精确异常	318
9.8.2	修改译码阶段的 ID 模块	273	11.3	异常处理过程	319
9.8.3	修改访存阶段	275	11.4	异常相关指令介绍	321
9.8.4	修改 OpenMIPS 模块	279	11.4.1	自陷指令	321
9.9	测试 ll、sc 指令实现效果	279	11.4.2	系统调用指令 syscall	324
9.10	load 相关问题	281	11.4.3	异常返回指令 eret	325
9.10.1	load 相关问题介绍	281	11.5	异常处理实现思路	325
9.10.2	解决方法	281	11.5.1	实现思路	325
9.11	修改 OpenMIPS 以解决 load 相关问题	283	11.5.2	修改数据流图	326
9.11.1	修改译码阶段的 ID 模块	283	11.5.3	修改系统结构	326
9.11.2	修改 OpenMIPS 模块	284	11.6	修改 OpenMIPS 以实现异常处理	328
9.12	测试 load 相关问题解决效果	285	11.6.1	修改取指阶段	328
9.13	小结	286	11.6.2	修改译码阶段	330
第 10 章	协处理器访问指令的实现	287	11.6.3	修改执行阶段	339
10.1	协处理器介绍	287	11.6.4	修改访存阶段	346
10.2	协处理器 CP0 中的寄存器	288	11.6.5	修改协处理器 CP0	352
10.3	协处理器 CP0 的实现	295	11.6.6	修改控制模块 CTRL	355
10.4	协处理器访问指令说明	300	11.6.7	修改 OpenMIPS	357
10.5	协处理器访问指令实现思路	300	11.7	再次修改最小 SOPC	358
10.5.1	实现思路	300	11.8	测试程序	359
10.5.2	数据流图的修改	301	11.8.1	测试程序 1——测试系统调用异常	359
10.5.3	系统结构的修改	301	11.8.2	测试程序 2——测试自陷异常	361
10.6	修改 OpenMIPS 以实现协处理器访问指令	303	11.8.3	测试程序 3——测试时钟中断	364
10.6.1	修改译码阶段	303	11.9	教学版 OpenMIPS 处理器实现小结	366

第三篇 进阶篇

第 12 章 实践版 OpenMIPS 处理器设计与实现	368
12.1 实践版 OpenMIPS 处理器的设计目标	368
12.2 Wishbone 总线介绍	370
12.2.1 Wishbone 总线接口说明	370
12.2.2 Wishbone 总线单次读操作的过程	372
12.2.3 Wishbone 总线单次写操作的过程	373
12.2.4 SEL_O/SEL_I 信号说明	374
12.3 实践版 OpenMIPS 处理器接口	375
12.4 实践版 OpenMIPS 处理器的实现思路	376
12.5 从教学版 OpenMIPS 到实践版 OpenMIPS	378
12.5.1 Wishbone 总线接口模块的实现	378
12.5.2 修改 CTRL 模块	384
12.5.3 修改 OpenMIPS 顶层模块	386
12.6 实践版 OpenMIPS 处理器实现小结	386
第 13 章 基于实践版 OpenMIPS 的小型 SOPC	387
13.1 小型 SOPC 的结构	387
13.2 Wishbone 总线互联矩阵 WB_CONMAX	388
13.3 GPIO	390
13.4 UART 控制器	392
13.4.1 UART 简介	392
13.4.2 UART16550 IP 核介绍	394
13.5 Flash 控制器	398
13.5.1 Flash 简介	398
13.5.2 Flash 控制器的设计	399
13.5.3 Flash 控制器的实现	400
13.6 SDRAM 控制器	403
13.6.1 SDRAM 简介	403
13.6.2 SDRAM CONTROLLER IP 核	409
13.7 实现基于实践版 OpenMIPS 的小型 SOPC	412
13.8 本章小结	423
第 14 章 验证实践版 OpenMIPS 处理器	424
14.1 DE2 平台简介	424
14.2 测试需要的硬件连接	425
14.3 QuartusII 工程建立	426
14.4 测试步骤说明	430
14.5 测试——GPIO 实验	431
14.5.1 测试内容	431
14.5.2 测试程序	431
14.5.3 编译测试程序	432
14.5.4 将测试程序写入 Flash 芯片	433
14.5.5 下载小型 SOPC 到 DE2	435
14.5.6 测试效果	435
14.6 测试二——UART 实验	435
14.6.1 测试内容	435
14.6.2 测试程序	436

14.6.3 测试效果	438	15.10.4 堆栈布局	467
14.7 测试三——模拟操作系统的 加载过程	439	15.10.5 示例	468
14.7.1 测试内容	439	15.11 μ C/OS-II 在 OpenMIPS 处理器上的移植	470
14.7.2 测试程序 BootLoader	439	15.11.1 文件目录的建立	470
14.7.3 测试程序 SimpleOS	443	15.11.2 修改 os_cpu.h 文件	473
14.7.4 将测试程序写入 Flash	446	15.11.3 修改 os_cpu_a.S 文件	474
14.7.5 测试效果	448	15.11.4 修改 os_cpu_c.c 文件	496
14.8 本章小结	449	15.12 测试程序	500
第 15 章 为 OpenMIPS 处理器 移植 μC/OS-II	450	15.12.1 创建 openmips.h 文件	500
15.1 为什么需要操作系统	450	15.12.2 创建 openmips.c 文件	502
15.2 嵌入式实时操作系统介绍	451	15.13 编译指示文件的建立	508
15.3 μ C/OS-II 简介	452	15.14 OpenMIPS 处理器运行 移植后的 μ C/OS-II	515
15.4 μ C/OS-II 特点	452	15.15 本章小结	516
15.5 μ C/OS-II 的几个概念	454	附录 A 教学版 OpenMIPS 各个模块 的接口说明	517
15.5.1 任务	454	A.1 PC 模块接口说明	517
15.5.2 任务调度	456	A.2 IF/ID 模块接口说明	518
15.5.3 任务切换	456	A.3 ID 模块接口说明	518
15.5.4 μ C/OS-II 的中断处理	457	A.4 Regfile 模块接口说明	520
15.5.5 时钟节拍	457	A.5 ID/EX 模块接口说明	520
15.5.6 μ C/OS-II 的初始化	458	A.6 EX 模块接口说明	521
15.5.7 μ C/OS-II 的启动	458	A.7 DIV 模块接口说明	524
15.6 μ C/OS-II 的基本功能	458	A.8 EX/MEM 模块接口说明	525
15.6.1 任务间的通信与同步	459	A.9 MEM 模块接口说明	527
15.6.2 任务管理	459	A.10 MEM/WB 模块接口说明	529
15.6.3 时间管理	459	A.11 CP0 模块接口说明	530
15.6.4 内存管理	460	A.12 LLbit 模块接口说明	532
15.7 μ C/OS-II 的文件体系	460	A.13 HILO 模块接口说明	532
15.8 μ C/OS-II 的移植条件	461	A.14 CTRL 模块接口说明	533
15.9 C 语言中使用汇编代码	463		
15.10 MIPS 函数调用规范	465		
15.10.1 寄存器使用规范	465		
15.10.2 参数传递	466		
15.10.3 函数返回值	466		

附录 B OpenMIPS 实现的所有	
指令及对应的机器码	534
B.1 逻辑操作指令	534
B.2 移位操作指令	534
B.3 移动操作指令	535
B.4 算术操作指令	535
B.5 转移指令	536
B.6 加载存储指令	536
B.7 协处理器访问指令	537
B.8 异常相关指令	537
B.9 空指令及其他指令	537
参考文献	538

第一篇 理论篇

第 1 章 处理器与 MIPS 2

第 2 章 可编程逻辑器件与 Verilog HDL 16