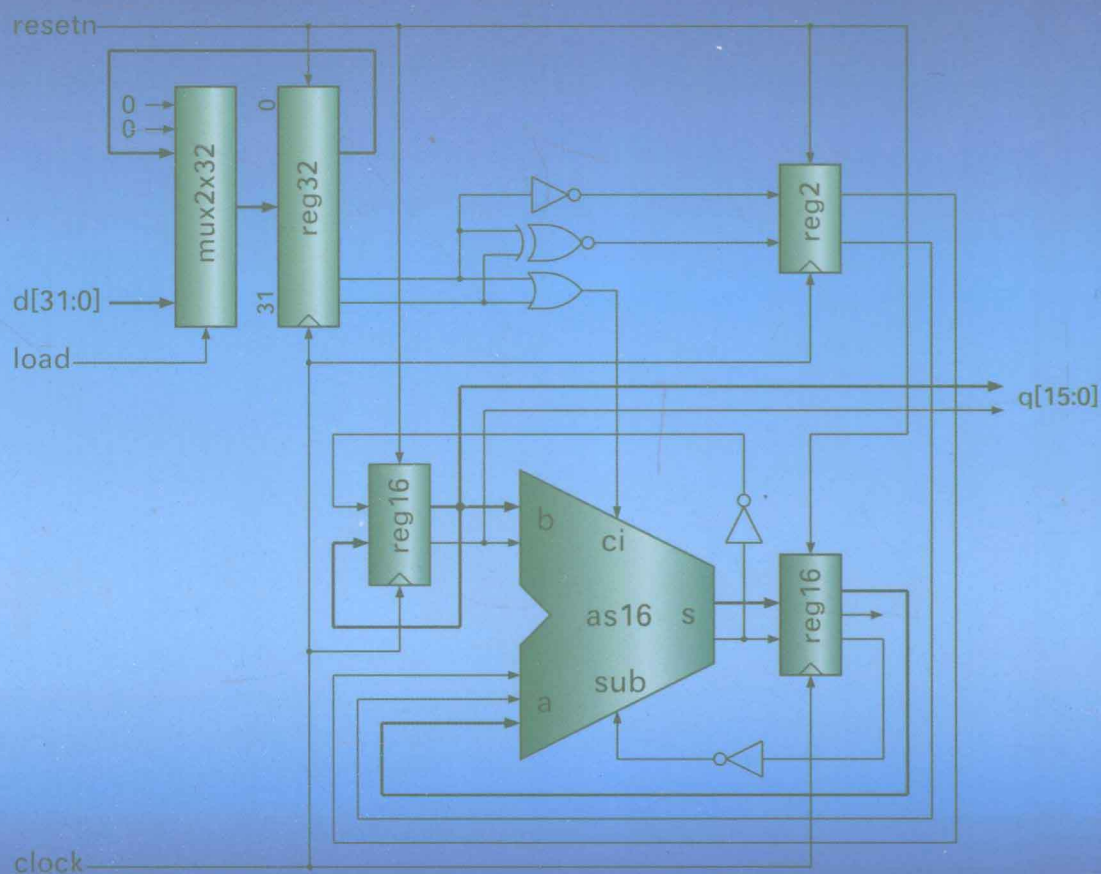


Computer Principles and Design in Verilog HDL

计算机原理与设计

—— Verilog HDL 版

李亚民 著



清华大学出版社

Computer Principles and Design in Verilog HDL

计算机原理与设计

—— Verilog HDL 版

李亚民 著

清华大学出版社

北京

内 容 简 介

本教科书讲述计算机原理、计算机设计以及如何用 Verilog HDL 实现设计。主要包括：计算机基础知识及性能评价方法；数字电路及 Verilog HDL 简介；计算机加、减、乘、除及开方的各种算法（包括 Wallace Tree 快速乘法器和 Newton-Raphson 及 Goldschmidt 除法和开方算法）及其 Verilog HDL 实现；指令系统结构和 ALU 及多端口寄存器堆的 Verilog HDL 设计；单周期、多周期和流水线 CPU 的 Verilog HDL 设计；精确中断和异常处理及其电路实现；浮点算法及带有浮点部件 FPU 的流水线 CPU 的 Verilog HDL 设计；多线程 CPU 的 Verilog HDL 设计；存储器、Cache 和虚拟存储器管理以及带有 Cache、TLB 和 FPU 的 CPU 设计；多核 CPU 的 Verilog HDL 设计；异步通信接口 UART、PS/2 键盘与鼠标接口、视频图像阵列 VGA 接口、I2C 串行总线接口和 PCI 并行总线接口的 Verilog HDL 设计；高性能计算机及互联网络设计。书中的 Verilog HDL 源代码基本上都附有功能仿真波形，以便加深对计算机原理的理解和对计算机设计方法的掌握。

本书可用作高等院校计算机及信息专业本科生和研究生教材，也可供自学者阅读。

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

计算机原理与设计：Verilog HDL 版/李亚民著. —北京：清华大学出版社，2011.6
ISBN 978-7-302-25109-5

I. ①计… II. ①李… III. ①电子计算机—基础理论 ②硬件描述语言，Verilog HDL—程序设计 IV. ①TP3

中国版本图书馆 CIP 数据核字(2011)第 049729 号

责任编辑：薛 慧
责任校对：刘玉霞
责任印制：何 芊

出版发行：清华大学出版社
<http://www.tup.com.cn>
社 总 机：010-62770175
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

地 址：北京清华大学学研大厦 A 座
邮 编：100084
邮 购：010-62786544

印 刷 者：清华大学印刷厂

装 订 者：三河市金元印装有限公司

经 销：全国新华书店

开 本：185×260 印张：33.75

字 数：812 千字

版 次：2011 年 6 月第 1 版

印 次：2011 年 6 月第 1 次印刷

印 数：1~5000

定 价：49.00 元

产品编号：035242-01

序 言

李亚民教授长期从事计算机原理和体系结构的教学与研究，他出版过三本教科书^[35, 36, 37]，发表过大量的杂志与会议文章。本书是他近年来执教和科研的实践总结。他的《计算机组成与系统结构》教科书在 2000 年由清华大学出版社出版；2005 年清华大学出版社出版的《CPU 芯片逻辑设计技术》是基于李亚民教授在清华大学系列讲座的内容而整理出版的。

李亚民教授的这本新书和上述两本书相比，有很多新的发展和不同点。统观全书，它贯穿了用 Verilog HDL 来描述计算机的各种部件和用 Verilog HDL 技术来设计 CPU 各部件。这是本书一个很重要的特点。这种描述和设计技术是要作者花费很大工夫自己去理解和实践的。单从这点而言，可见本书作者在著写本书时，要投入比著写一般“计算机原理”书籍更多的时间和心血。这个特点带给读者的好处是：学完本书后，不仅懂了计算机原理，也可以基本掌握计算机的一种重要的设计技术。

此外，这本书的内容也增加了一些重要的环节，如 CPU 中的多核和多线程技术。大家知道，多核和多线程是今后计算机较长时间的发展方向，尤其是近年来高性能计算进入千万亿次时代以后，多核和多线程技术会有更蓬勃的发展。本书这方面的内容也是计算机学生应该掌握的新内容。本书还增加了浮点部件 FPU 设计、浮点和整数部件结合成整体的完整的流水线 CPU 设计、带有 Cache 和 TLB 的多核 CPU 的设计、常用 I/O 接口的设计。本书中所讲授的计算机各部件，都尽量给出了 Verilog HDL 源代码。这些 Verilog HDL 源代码都经过 Quartus II 或者 Iverilog 的验证。这也体现了作者写书的严谨性和所花费的精力。

中国的计算机事业在改革开放以来，取得了飞速的发展。在高性能计算、网络技术、软件、计算机产业等方面都取得了举世瞩目的成就。然而，我国必须继续重视计算机发展的基础，这就是计算机的设计技术，尤其是计算机芯片设计技术。这方面的不足是和计算机教育分不开的。我国很多大学的“计算机原理”、“计算机组成与结构”的课程讲授和计算机设计技术基础脱离，学生学完计算机原理与组成却一点未掌握计算机设计技术。

本书特点是理论和实践的密切结合。希望本书的出版和推广，能加强大学计算机教学领域内的理论与实践的结合。很多工程领域的结构都好像是个金字塔，只有基础更广泛和更结实了，金字塔的塔尖才可以构建得更高。我们寄希望于广大的大学生和研究生！只有更多的年轻人掌握了计算机和 CPU 的设计技术，我国的计算机和 CPU 设计技术才会攀登新的高峰！

李 三 立
中国工程院院士 清华大学教授
2011 年 5 月 1 日

前 言

欢迎阅读本书。首先郑重声明：本书所载的 Verilog HDL 源代码可以随便使用、随便更改。但本书作者对任何大学、研究所、公司或者个人由于使用本书所载的 Verilog HDL 源代码设计 CPU 或者任何其他电路所造成的任何损失不承担任何责任。

也许有读者会问：“现在出版的计算机原理、计算机组成与结构的教科书已经很多了，你为什么还要写这本书？”诚如本书书名《计算机原理与设计——Verilog HDL 版》所提示的那样，本书不光讲计算机原理，也讲计算机设计，而且是用 Verilog HDL 讲计算机设计。如果只看教科书讲的原理而不自己动手设计，就如同有人向你描述有一种水果是什么什么味道，听了半天也只能感觉和想象。还不如花一分钟亲口尝一尝：噢，原来是这个味道，这就是传说中的梨子吗？

让读者知道如何自己动手用 Verilog HDL 设计 CPU 及 I/O 接口和总线是作者要写本书的原因之一。本书的一个特点是所有 CPU 和 I/O 接口及总线的 Verilog HDL 源代码全部由作者本人写出并经过 CAD/CAE 软件验证。作者在 2000 年由清华大学出版社出版过一本《计算机组成与系统结构》的教科书^[36]，得到广大读者和网友的好评及推荐，有些大学也选其作为课程教材或参考书使用。与那本书不同，这本《计算机原理与设计——Verilog HDL 版》给出具体的 CPU、I/O 接口和总线设计的 Verilog HDL 源代码并附有仿真波形，以使读者能加深理解和供读者在设计自己的电路时参考，同时也增加了多线程和多核技术等内容。另一个原因是现在用硬件描述语言来设计电路变得很流行，已成主流，用逻辑图的方法直接设计电路的人越来越少了。但大部分介绍硬件描述语言的书都只停留在简单的数字电路的设计上，因此作者想把它引入到计算机原理的教科书中。

也许有读者还会问：“那你为什么选用 Verilog HDL，而不是其他的硬件描述语言呢？”回答这个问题其实有些难。目前比较流行的硬件描述语言主要有 Verilog HDL 和 VHDL 两种。大家都说 Verilog HDL 像 C 语言，而 VHDL 像 C++（实际上像 Ada 语言）。由于作者对 C 比对 C++ 更熟悉一些，所以开始自学硬件描述语言时就自然而然选了 Verilog HDL。据作者后来有限的调查得知，一些大公司也都使用 Verilog HDL。调查的对象仅限作者指导过的本科生和研究生，他们毕业之后有些人目前在著名的大公司从事集成电路的设计工作。看来作者当初的选择是对的。作者不是说选择 VHDL 就不对，其实熟练了都一样，都是很好的设计硬件的语言工具。

考虑到初学者的需要，本书从最基本的二进制数的运算和逻辑电路的设计讲起，顺便介绍 Verilog HDL。因此学习本书时不需要任何先修科目，不懂 C 语言也没有关系。“那是不是这本书的内容太过简单了呢？”不是。本书要讲到如何设计一个带有整数部件 (Integer Unit, IU)、浮点部件 (Floating Point Unit, FPU)、指令 Cache (一种特殊的小容量的快速存储器)、数据 Cache、指令 TLB (Translation Lookaside Buffer, 一种用于地址转换的快速缓冲区) 和数据 TLB 的多核 CPU。其中的 Cache 完全由硬件控制；TLB 不命中时产生异常，在异常处理程序中访问用于存储器管理的页表 (Page Table)，然后填充 TLB；FPU 能对 IEEE 754 单精度浮点数进行加、减、

乘、除和开方运算：加减法用先行进位加减法器实现、乘法用 Wallace Tree 实现、除法和开方用 Newton-Raphson 算法实现。这些都是实际的 CPU 设计中常用的部件和算法。一般的教科书都是分开讲它们的原理。本书也讲原理，但更重要的是把它们有机地结合在一起，设计出一个完整的 CPU。当然，Verilog HDL 源代码和仿真波形也在书中给出。

稍微讲具体一些。本书的内容包括：计算机基础知识及性能评价；逻辑电路及 Verilog HDL 简介；计算机算法及 Verilog HDL 实现；指令系统及 ALU (Arithmetic Logic Unit) 设计；单周期、多周期和流水线 CPU Verilog HDL 设计；精确中断和异常处理及电路实现；浮点算法及 FPU Verilog HDL 设计；带有 FPU 的流水线 CPU Verilog HDL 设计；多线程 CPU Verilog HDL 设计；存储器和虚拟存储器管理；带有 Cache、TLB 及 FPU 的 CPU 设计；多核 CPU Verilog HDL 设计；输入/输出接口和总线设计；高性能计算机及互连网络设计。更详细的内容就只有看目录了。

除了 Newton-Raphson 算法，本书也介绍 Goldschmidt 除法和开方算法。这两种算法都被著名的公司 (比如 Intel 和 IBM) 在设计 CPU 时使用。输入/输出接口部分除了介绍基本的概念和原理之外，还给出了常用 I/O 接口及总线的具体的设计方法，比如异步通信接口 UART、PS/2 鼠标和键盘接口、VGA 接口、I2C 串行总线和 PCI 并行总线等。除了 PS/2 鼠标，其他的接口和总线都给出了具体的 Verilog HDL 设计实例。PS/2 鼠标讲了原理，设计部分当做习题留给了读者。高性能计算机及互连网络设计部分除了讲述一般的设计方法之外，也描述了由作者提出的新型互连网络，包括 Dual-Cube^[15]、Metacube^[20] 和 RDN (Recursive Dual-Net)^[19]。这些互连网络具有用较少的端口连接更多的 CPU 和存储器板并且保证它们之间的通信距离比较短的优点，非常适合用来构建下一代超大规模的超高性能计算机。

“好像内容太多了，一个学期学不完。”以上内容可分成两部分，一部分放在本科讲，另一部分给研究生讲。举例来讲，TLB 设计、FPU 设计、多线程和多核 CPU 设计、高性能计算机及互连网络设计等内容可在研究生课程中讲授，其余部分在本科课程中教授。如果在本科阶段能设计出流水线 CPU 就已经很好了，课时充裕的话再加上中断和 Cache 设计。存储器管理要讲，但不一定要设计 TLB。以上只是举例，大家可以根据学校的具体情况做出不同的安排。

“那么需要什么样的环境呢？”如果只想验证用 Verilog HDL 设计的电路的逻辑功能是否正确，只要有一台普通的 PC 就行，所需的软件可以从网上下载，免费的，而且也不用注册，安装后马上就能用，而且相当好用。比如作者使用的 Icarus Verilog (Iverilog) 和 Altera 公司的 Quartus II Web Edition 就是这样的软件。如果想要用 FPGA 实现，买些 FPGA 实验板，即可实现诸如“计算机原理与设计实验”等课程所要求的内容。Altera 公司也与国内很多高校合作建立了硬件实验室，免费提供 FPGA 实验板。其实买也不贵。如果使用 Xilinx FPGA 板，则部分代码需要修改。

“如果想自学，读这本书合适吗？”自学很重要，不管是出于工作需要还是个人兴趣。对已经参加工作的技术人员或在校的非计算机专业的学生来讲，如果想自学，可以慢慢读、慢慢做，因为咱没有期末考试。本书的写作风格就是试图把复杂

的概念通俗易懂地讲出来，让人容易理解，所以本书非常适合自学者阅读。

“如果只想学原理不想学设计，读这本书合适吗？”对每个基本概念，本书都是先讲它的原理，再讲设计。如果只想学原理，只要不看设计部分就行了。不过还是都学为好，因为设计的过程也是学习的过程。如果读者已经懂了原理，直接看设计部分就行了，这样效率会更高。

“每章最后的习题有没有标准答案？”没有。有些题目尤其是设计类型的题目的做法有很多种，只要到了罗马(到了北京也行)，都得 100 分，都是标准答案。有新意的，加 10 分。有些题目附加了提示，仅供读者参考。作者鼓励大家发挥自己的创造性，找一些书上没有的题目来做。

作者欣喜地看到，计算机的硬件实验教学终于开始受到重视，从 2009 年开始在南京大学召开的全国计算机组成与结构课程群实验教学研讨会就是一个证明。作者有幸受邀和参加会议的各位老师进行了交流。大家一致认为，计算机偏硬件类课程在整个计算机学科的教学居于非常重要的位置，应该从现在起加强对这方面的投入，包括教师队伍的建设、教材的更新以及实验环境的准备等。本书的出版，也算是响应这个会议的号召吧。

希望读者在学完本书之后，在对计算机原理与设计和 Verilog HDL 的理解和使用上，比没有阅读之前有一定程度的提高。从个人的角度讲，熟练地掌握了 CPU 设计技术，至少会增加一种就业机会，而且一不小心就成了专家。从大局着眼，相信读者会为我国的 CPU 设计水平的提高做出自己的贡献。另外，本教科书的这种写法是一种新的尝试，因此作者也期待有更多的与本书风格类似的教科书出现，大家一起来繁荣我国的计算机基础教育事业——也就是清华大学教授李三立院士在“序言”中提到的——把金字塔的基础打牢。

本书得以出版，作者感谢李三立教授。李教授为本书的写作和出版提供了很好的建议和帮助。另外，大约从十年前开始，李教授就为作者联系了很多高校，在暑假期间开设 CPU 设计系列讲座，比如清华大学、东南大学和浙江大学等。作者也感谢清华大学出版社对本书的支持。

本书中所有的 Verilog HDL 源代码均经过 Quartus II Web Edition 或者 Iverilog 的验证，C 语言程序由 gcc 编译。本书初稿在 Linux/CentOS 环境下完成：中文输入用 emacs + chinput，除了从屏幕截取的仿真波形图，其他所有的图几乎都是用 tgif 或 gnuplot 画成，PDF 文件用 VTeX/Linux + CJK 生成。以上均为免费软件，作者一并向这些软件开发者和相关公司表示感谢。

由于作者水平有限，本书所载的 Verilog HDL 源代码写得不太漂亮，而且只是一家之言。希望读者能写出质量更高的代码并对本书的错误之处加以指正。来信请寄：yamin@ieee.org 或者 yamin@computer.org。

李亚民

2011 年 5 月 5 日

目 录

第 1 章	计算机基础知识及性能评价	1
1.1	计算机系统概述	1
1.1.1	计算机系统的组成	1
1.1.2	计算机发展简史	2
1.1.3	计算机指令结构	4
1.1.4	CISC 和 RISC	7
1.1.5	一些基本单位的意义	9
1.2	计算机的基本结构	10
1.2.1	RISC CPU 的基本结构	10
1.2.2	多线程 CPU 和多核 CPU	12
1.2.3	存储层次和虚拟存储器管理	13
1.2.4	I/O 接口和总线	14
1.3	如何提高计算机的性能	15
1.3.1	计算机性能和性能评价	15
1.3.2	踪迹驱动模拟和执行驱动模拟	16
1.3.3	高性能计算机和互联网络	18
1.4	硬件描述语言	18
1.5	习题	21
第 2 章	逻辑电路及 Verilog HDL 简介	22
2.1	基本逻辑门和常用逻辑门	22
2.2	用 Verilog HDL 实现基本的逻辑操作	24
2.3	逻辑门的 CMOS 晶体管实现以及晶体管级的 Verilog HDL	29
2.3.1	CMOS 反向器	29
2.3.2	CMOS 与非门和或非门	31
2.4	四种风格的 Verilog HDL 描述	33
2.4.1	晶体管开关级的 Verilog HDL	33
2.4.2	逻辑门级的 Verilog HDL	35
2.4.3	数据流风格的 Verilog HDL	36
2.4.4	功能描述风格的 Verilog HDL	37
2.5	常用的组合电路及其设计	40
2.5.1	多路选择器设计	40
2.5.2	译码器设计	41
2.5.3	32 位移位器设计	43
2.6	时序电路的设计方法	47

2.6.1	D 锁存器	47
2.6.2	D 触发器	49
2.6.3	状态转移图及时序电路设计	52
2.7	习题	58
第 3 章	计算机算法及其 Verilog HDL 实现	61
3.1	二进制整数	61
3.1.1	无符号二进制整数	62
3.1.2	补码表示的带符号二进制整数	62
3.2	加减法算法及 Verilog HDL 实现	63
3.2.1	加法和减法器设计	63
3.2.2	先行进位加法器设计	69
3.3	乘法算法及 Verilog HDL 实现	73
3.3.1	无符号数乘法器设计	73
3.3.2	带符号数乘法器设计	74
3.3.3	无符号数 Wallace 树型乘法器设计	77
3.3.4	带符号数 Wallace 树型乘法器设计	82
3.4	除法算法及 Verilog HDL 实现	84
3.4.1	恢复余数除法器设计	84
3.4.2	不恢复余数除法器设计	86
3.4.3	带符号数不恢复余数除法器设计	89
3.4.4	Goldschmidt 除法算法	91
3.4.5	Newton-Raphson 除法算法	94
3.5	开方算法及 Verilog HDL 实现	97
3.5.1	恢复余数开方算法	97
3.5.2	不恢复余数开方算法	100
3.5.3	Goldschmidt 开方算法	105
3.5.4	Newton-Raphson 开方算法	108
3.6	习题	111
第 4 章	指令系统及 ALU 设计	113
4.1	指令系统结构	113
4.1.1	操作数类型	113
4.1.2	数据在存储器中的存放方法	114
4.1.3	指令类型	115
4.1.4	指令结构	117
4.1.5	寻址方式	118
4.2	MIPS 指令格式和通用寄存器定义	119

4.2.1	MIPS 指令格式	119
4.2.2	MIPS 通用寄存器	120
4.3	MIPS 指令和 ALU 设计	120
4.3.1	本书 CPU 可执行的 MIPS 指令	120
4.3.2	ALU 设计	123
4.4	习题	125
第 5 章	单周期 CPU 及其 Verilog HDL 设计	127
5.1	执行一条指令所需的硬件电路	127
5.1.1	与取指令有关的电路	127
5.1.2	寄存器计算类型指令执行时所需电路	128
5.1.3	立即数计算类型指令执行时所需电路	130
5.1.4	访问存储器类型指令执行时所需电路	130
5.1.5	条件转移类型指令执行时所需电路	131
5.1.6	跳转和子程序调用及返回类型指令执行时所需电路	132
5.2	寄存器堆设计	133
5.2.1	寄存器堆的硬件电路设计	133
5.2.2	结构描述风格的寄存器堆 Verilog HDL 代码	135
5.2.3	功能描述风格的寄存器堆 Verilog HDL 代码	138
5.3	数据路径设计	139
5.3.1	多路选择器的使用	139
5.3.2	单周期 CPU 的总体电路	142
5.3.3	单周期 CPU 的 Verilog HDL 代码	143
5.4	控制部件设计	144
5.4.1	控制部件的逻辑设计	145
5.4.2	控制部件的 Verilog HDL 代码	147
5.5	存储器及测试程序设计	149
5.5.1	数据存储器设计	149
5.5.2	指令存储器及测试程序设计	150
5.5.3	单周期 CPU 测试结果及说明	151
5.6	习题	156
第 6 章	异常和中断处理及其电路实现	158
6.1	异常和中断	158
6.1.1	异常和中断的定义与类型	158
6.1.2	查询中断和向量中断	159
6.1.3	中断屏蔽和中断嵌套	162
6.1.4	中断优先级	163

6.2	带有异常和中断处理功能的 CPU 的设计	164
6.2.1	异常和中断的处理过程以及相关的寄存器	164
6.2.2	与异常和中断有关的指令	165
6.2.3	带有异常和中断处理功能的 CPU 总体结构	166
6.2.4	带有异常和中断处理功能的 CPU Verilog HDL 代码	168
6.3	CPU 的异常与中断测试	173
6.3.1	测试程序和测试数据	173
6.3.2	CPU 异常及中断处理测试结果及说明	176
6.4	习题	181
第 7 章	多周期 CPU 及其 Verilog HDL 设计	182
7.1	把一条指令的执行分成若干个周期	182
7.1.1	取指令周期 IF	183
7.1.2	指令译码周期 ID	184
7.1.3	指令执行周期 EXE	185
7.1.4	存储器访问周期 MEM	188
7.1.5	结果写回周期 WB	189
7.2	多周期 CPU 的总体电路及 Verilog HDL 代码	189
7.2.1	多周期 CPU 的总体电路	189
7.2.2	多周期 CPU 的 Verilog HDL 代码	189
7.3	用有限状态机实现多周期 CPU 的控制部件	191
7.3.1	多周期 CPU 的控制部件的状态转移图	191
7.3.2	多周期 CPU 的控制部件的总体结构	191
7.3.3	下一状态函数	192
7.3.4	控制信号的产生	193
7.3.5	控制部件的 Verilog HDL 代码	193
7.4	存储器及测试程序设计	198
7.4.1	存储器设计	198
7.4.2	测试程序代码	198
7.4.3	多周期 CPU 测试结果	199
7.5	习题	203
第 8 章	流水线 CPU 及其 Verilog HDL 设计	204
8.1	流水线技术的基本概念	204
8.1.1	取指令 IF 级的电路	206
8.1.2	指令译码 ID 级的电路	207
8.1.3	指令执行 EXE 级的电路	208
8.1.4	存储器访问 MEM 级的电路	208

8.1.5	结果写回 WB 级的电路	208
8.2	流水线 CPU 的相关问题及解决对策	210
8.2.1	数据相关及解决对策	210
8.2.2	控制相关及解决对策	214
8.3	流水线 CPU 的整体设计及 Verilog HDL 代码	215
8.3.1	流水线 CPU 的整体电路	215
8.3.2	流水线 CPU 的 Verilog HDL 代码	216
8.4	流水线 CPU 的测试	224
8.4.1	流水线 CPU 的测试程序	224
8.4.2	流水线 CPU 的仿真波形	225
8.5	精确中断和异常事件处理	228
8.5.1	异常事件和中断的种类以及相关的寄存器	228
8.5.2	流水线 CPU 的中断响应过程	229
8.5.3	流水线 CPU 处理异常事件	232
8.6	带有处理异常和中断功能的流水线 CPU 的设计	235
8.6.1	流水线 CPU 的总体结构	235
8.6.2	流水线 CPU 的 Verilog HDL 代码	236
8.6.3	异常和中断的测试程序与仿真波形	242
8.7	习题	245
第 9 章	浮点算法及 FPU Verilog HDL 设计	246
9.1	IEEE 754 浮点数格式	246
9.2	单精度浮点数与整数之间的转换	247
9.2.1	浮点数转换成整数	247
9.2.2	整数转换成浮点数	251
9.3	浮点加法器 FADD 设计	253
9.3.1	浮点加法算法	253
9.3.2	浮点加法器 Verilog HDL 代码	255
9.3.3	流水线浮点加法器设计	262
9.4	浮点乘法器 FMUL 设计	269
9.4.1	浮点乘法算法	269
9.4.2	Wallace 树型浮点乘法器 Verilog HDL 代码	271
9.4.3	流水线 Wallace 树型浮点乘法器设计	276
9.5	浮点除法器 FDIV 设计	282
9.5.1	浮点除法算法	282
9.5.2	Newton-Raphson 浮点除法器 Verilog HDL 代码	283
9.6	浮点开方器 FSQRT 设计	292
9.6.1	浮点开方算法	292

9.6.2	Newton-Raphson 浮点开方器 Verilog HDL 代码	293
9.7	习题	301
第 10 章	带有 FPU 的流水线 CPU 及其 Verilog HDL 设计	303
10.1	CPU/FPU 流水线模型	303
10.1.1	CPU/FPU 可执行的指令	303
10.1.2	CPU/FPU 基本的流水线模型	304
10.2	带有两个写端口的寄存器堆设计	306
10.3	浮点数据相关以及流水线暂停	307
10.3.1	浮点运算结果的内部前推和流水线暂停	308
10.3.2	lwc1 和 swc1 造成的流水线暂停及内部前推	310
10.3.3	浮点除法和开方指令造成的流水线暂停	315
10.4	带有 FPU 的流水线 CPU 的总体结构及 Verilog HDL 代码	316
10.4.1	带有 FPU 的流水线 CPU 的具体电路	317
10.4.2	浮点部件 FPU 的具体电路	318
10.4.3	整数部件 IU 的具体电路	319
10.4.4	带有 FPU 的流水线 CPU 的 Verilog HDL 代码	320
10.5	存储器模块及 CPU/FPU 的测试	328
10.5.1	指令存储器和数据存储器	328
10.5.2	CPU/FPU 的测试程序	329
10.5.3	CPU/FPU 的仿真波形	331
10.6	习题	335
第 11 章	多线程 CPU 及其 Verilog HDL 设计	337
11.1	多线程 CPU 概述	337
11.1.1	多线程 CPU 的基本概念	337
11.1.2	多线程 CPU 的基本结构	338
11.2	多线程 CPU 设计	338
11.2.1	线程的选择方法	338
11.2.2	多线程 CPU 的详细电路	339
11.2.3	多线程 CPU 的 Verilog HDL 代码	340
11.3	多线程 CPU 的仿真波形	343
11.4	习题	351
第 12 章	存储器和虚拟存储器管理	352
12.1	存储器	352
12.1.1	静态存储器 (SRAM)	352
12.1.2	动态存储器 (DRAM)	354

12.1.3 只读存储器 (ROM)	354
12.1.4 相联存储器 (CAM)	355
12.1.5 存储层次	356
12.2 高速缓存 (Cache)	357
12.2.1 Cache 的映像机制	358
12.2.2 Cache 块的替换算法	360
12.2.3 Cache 写策略	362
12.2.4 数据 Cache 电路设计及 Verilog HDL 代码	362
12.3 虚拟存储器管理及 TLB 设计	365
12.3.1 虚拟存储器与主存的关系	366
12.3.2 分段管理	366
12.3.3 分页管理	367
12.3.4 快速地址转换 TLB 及其电路设计	368
12.3.5 TLB 与 Cache 的并行访问	372
12.4 MIPS 基于 TLB 的虚拟地址转换机制	373
12.4.1 MIPS 的虚拟地址空间	373
12.4.2 MIPS TLB 的构成	374
12.4.3 MIPS 虚拟地址转换	377
12.4.4 MIPS TLB 维护指令	377
12.5 习题	380
第 13 章 带有 Cache 及 TLB 和 FPU 的 CPU 设计	382
13.1 Cache 和 TLB 的总体结构	382
13.2 与 Cache 有关的电路设计	383
13.2.1 指令 Cache 的 Verilog HDL 代码	383
13.2.2 数据 Cache 和指令 Cache 与外部存储器的接口	384
13.2.3 Cache 不命中时流水线暂停的电路	385
13.3 与 TLB 有关的电路设计	385
13.3.1 指令 TLB (ITLB) 和数据 TLB (DTLB)	386
13.3.2 TLB 不命中时异常信号的产生	386
13.3.3 与 TLB 不命中异常有关的寄存器	387
13.3.4 对 TLB 不命中异常的处理	391
13.4 带有 Cache 及 TLB 的 CPU 设计	392
13.4.1 带有 Cache 及 TLB 的 CPU 总体结构	392
13.4.2 带有 Cache 及 TLB 的 CPU 的 Verilog HDL 代码	393
13.5 带有 Cache 及 TLB 的 CPU 的测试程序和仿真波形	406
13.6 习题	414

第 14 章 多核 CPU 及其 Verilog HDL 设计	415
14.1 多核 CPU 概述	415
14.1.1 多核 CPU 的基本概念	415
14.1.2 多核 CPU 的 Cache 一致性问题	417
14.2 多核 CPU 设计	418
14.2.1 多核 CPU 的总体结构	418
14.2.2 多核对外部总线的竞争与仲裁	418
14.2.3 多核 CPU 的 Verilog HDL 代码	419
14.3 多核 CPU 的测试程序及仿真波形	421
14.4 习题	426
第 15 章 输入/输出接口及设计	427
15.1 I/O 接口概述	427
15.1.1 I/O 地址空间和 I/O 指令	427
15.1.2 I/O 查询和中断	427
15.1.3 直接存储器访问 DMA	428
15.1.4 总线和总线的同步方式	429
15.2 数据错误检测及校正	429
15.2.1 奇偶校验	429
15.2.2 错误纠正码 ECC (扩展的海明码)	430
15.2.3 循环冗余校验 CRC	431
15.3 异步通信接口 UART	432
15.4 PS/2 接口	441
15.4.1 PS/2 键盘	442
15.4.2 PS/2 鼠标	444
15.5 视频图像阵列 VGA	446
15.5.1 VGA 及其接口控制器设计	446
15.5.2 VGA 显示键盘字符	456
15.6 I/O 总线	461
15.6.1 I2C 串行总线	461
15.6.2 PCI 并行总线	472
15.7 习题	480
第 16 章 高性能计算机及互连网络设计	482
16.1 高性能计算机的种类	482
16.1.1 集中式共享存储器系统 (SMP)	482
16.1.2 分布式共享存储器系统 (DSM)	483
16.2 互连网络的构成	484

- 16.3 互联网络的拓扑特性 486
 - 16.3.1 节点度 (Degree) 486
 - 16.3.2 直径 (Diameter) 486
 - 16.3.3 平均距离 (Average Distance) 487
 - 16.3.4 对分宽度 (Bisection Bandwidth) 487
- 16.4 常用的互联网络 487
 - 16.4.1 Mesh 487
 - 16.4.2 Torus 487
 - 16.4.3 Hypercube 488
 - 16.4.4 Tree 和 Fat-Tree 488
- 16.5 基本的通信操作 489
 - 16.5.1 一对多广播 490
 - 16.5.2 多对多广播 491
 - 16.5.3 一对多私通 491
 - 16.5.4 多对多私通 492
- 16.6 新型互联网络 492
 - 16.6.1 Dual-Cube 493
 - 16.6.2 Metacube 497
 - 16.6.3 RDN 499
- 16.7 习题 502
- 参考文献 503
- 图索引 505
- 表索引 515
- 术语索引 517

第 1 章 计算机基础知识及性能评价

欢迎阅读这本《计算机原理与设计——Verilog HDL 版》教科书。本书将从最基本的逻辑电路及其 Verilog HDL 设计讲起，由浅入深，逐步讲到如何用 Verilog HDL 设计一个带有分开的指令及数据 TLB/Cache 和能执行浮点加、减、乘、除及开方指令的浮点部件 FPU 的多核 CPU。本书也给出常用的输入/输出接口的设计方法，包括异步通信接口 UART、PS/2 键盘与鼠标接口、视频图像阵列 VGA 接口、I2C 和 PCI 总线接口。以上 CPU 和 I/O 接口基本上都有 Verilog HDL 源代码，供读者在设计自己的电路时参考。另外，我们也将描述高性能计算机及其互连网络的设计方法。

本章主要讲解一些基本的概念，对计算机原理与设计和计算机性能评价做一个简单的介绍，以便为学习以后各章打下一点基础并留下一个整体印象。

1.1 计算机系统概述

假设你有一台 PC，存储器有 16GB，CPU 主频为 3GHz。问：上述的两个 G 是否相同？本节最后将给出这个问题的答案。在此之前，我们简要介绍计算机和计算机系统各自包含的内容、计算机发展简史、计算机的指令结构以及 RISC 和 CISC 的基本概念与不同之处。

1.1.1 计算机系统的组成

读者可能知道什么是单片机，或者至少听说过（没听说过也没关系）。单片机的全称是单片计算机 (Single Chip Computer)，它就是一个芯片，内部集成有 CPU (Central Processing Unit)、存储器 (Memory) 和一些常用的 I/O 接口 (Input/Output Interface)。由此可见，包含了以上三种部件的芯片或电路板就叫计算机。

那么，一般的用户能不能直接使用这个计算机呢？答案是不能。我们常说的计算机实际上是指计算机系统 (Computer Systems)。计算机系统中除了包含有计算机 (CPU、存储器和 I/O 接口) 之外，还有计算机软件 (Software) 和 I/O 设备 (Input/Output Devices)。当然还少不了电源。

最基本的软件是操作系统 (Operating System)。它负责管理计算机系统的所有硬件资源，为用户使用计算机系统提供一个界面。最基本的也是最常用的操作系统有 MS-DOS 和 Linux 核 (Kernel)。微软的各种版本的 Windows 或诸如 Fedora 和 CentOS 等可被认为是在基本的操作系统基础上开发的高级图形界面。

软件是用计算机语言 (高级语言或汇编语言) 编写的程序再经过编译或汇编后生成的可执行的二进制代码。编译器 (Compiler) 和汇编器 (Assembler) 本身也是软件，有时也称为软件工具。其他的软件工具还有编辑器 (Editor) 和软件调试器 (Debugger)。计算机系统还配有常用的实用程序库 (Utilities)。除此之外的其他软件可以统称为应用程序 (Applications)。提问：第一个汇编器是用汇编语言编写的吗？