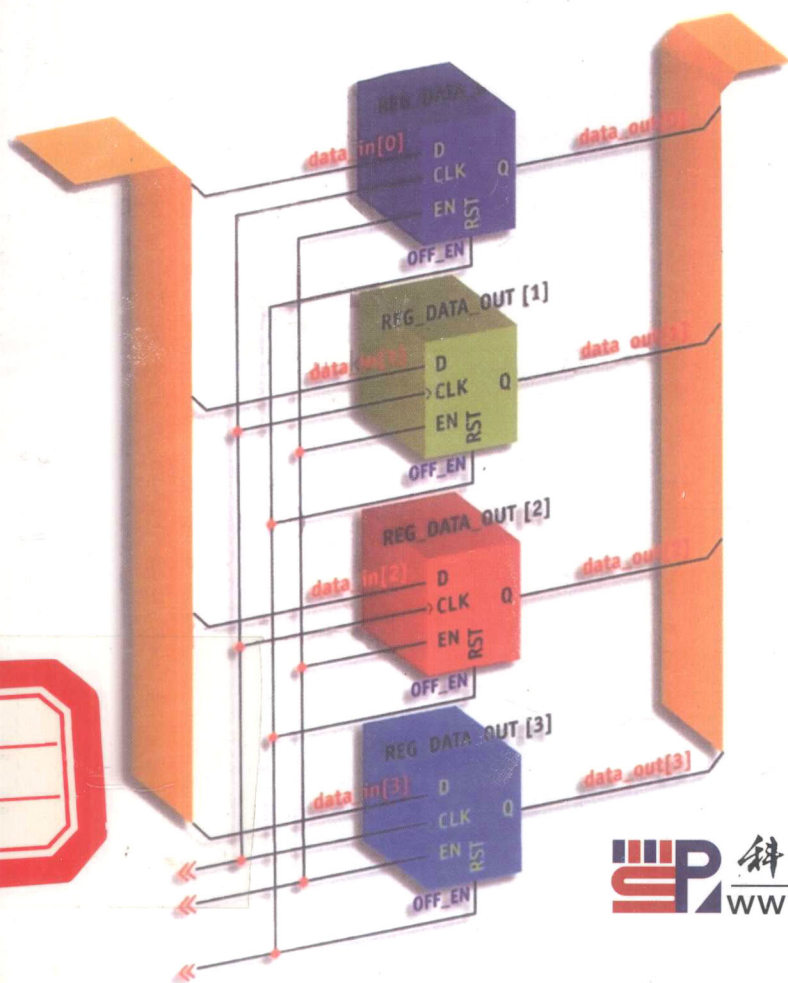


现代超大规模集成电路设计丛书

基于Verilog语言的 实用FPGA设计

〔美〕 K. 科夫曼 著
沈树群 张艳 吴京松 译
刘丽华 审



 科学出版社
www.sciencep.com

现代超大规模集成电路设计丛书

基于 Verilog 语言的实用 FPGA 设计

〔美〕 K. 科夫曼 著
沈树群 张 艳 吴京松 译
刘丽华 审

科 学 出 版 社

北 京

内 容 简 介

本书中, K. 科夫曼结合其工作实践, 讲解了利用 Verilog 进行 FPGA 设计的方法和技巧。内容涉及 Verilog 语言设计实践、数字设计的策略与技巧、数字电路工具箱、Verilog 测试、面向 ASIC 转化的设计等 9 章。学习本书有助于读者快速提高用 Verilog 进行 FPGA 设计的水平。

本书可供从事计算机设计等方面的工作者, 以及高等院校相关专业师生阅读、参考。
Simplified Chinese edition copyright ©2003 by Science Press and Pearson Education North Asia Limited.

Real World FPGA Design with Verilog, 1st ed. by Ken Coffman, Copyright ©2000

ISBN 0-13-099851-6

All Rights Reserved.

Published by arrangement with the original publisher, Pearson Education, Inc., publishing as PRENTICE HALL, INC.

For sale and distribution in the People's Republic of China exclusively (except Taiwan, Hong Kong SAR and Macau SAR).

仅限于中华人民共和国境内(不包括中国香港、澳门特别行政区和中国台湾省)销售发行。

本书封面贴有 Pearson Education(培生教育出版集团)激光防伪标签。无标签者不得销售。

图书在版编目(CIP)数据

基于 Verilog 语言的实用 FPGA 设计/(美)K. 科夫曼(Ken Coffman)著;沈树群,张艳,吴京松译.—北京:科学出版社,2004

(现代超大规模集成电路设计丛书)

ISBN 7-03-012836-2

I. 基… II. ①K…②沈…③张…④吴… III. ①硬件描述语言-程序设计②可编程逻辑器件 IV. ①TP312②TP332.1

中国版本图书馆 CIP 数据核字(2004)第 004857 号

策划编辑:匡敏钟 谊/文案编辑:邱璐 贾瑞娜/责任校对:刘小梅

责任印制:安春生/封面设计:陈敬

科学出版社 出版

北京东黄城根北街16号

邮政编码:100717

<http://www.sciencep.com>

新蕾印刷厂 印刷

科学出版社发行 各地新华书店经销

*

2004 年 6 月第 一 版 开本: B5(720 × 1000)

2004 年 6 月第一次印刷 印张: 18

印数: 1—3 000 字数: 330 000

定价: 30.00 元(含光盘)

(如有印装质量问题, 我社负责调换(路通))

译者的话

20 世纪 90 年代以来,采用 Verilog 硬件描述语言和 FPGA 进行硬件设计,已经成为最为成熟的大规模数字系统硬件设计的通用手段。

与本书作者的观点相同,我们也认为 Verilog 有很强的生命力,之所以这样说是因为很多工程师都认为 Verilog 比 VHDL 容易学,并且 Verilog 比 VHDL 也更贴近于数字电路本身。因此,熟悉 Verilog 的人没理由再去学习 VHDL,而新入此道的人从 Verilog 入手比从 VHDL 开始学要容易得多。

至于采用 FPGA 来进行硬件实现,我们想目前大多数的设计人员都会采用这种方法,至少首先会利用 FPGA 来进行系统验证,然后为提高性能、降低成本,采用 ASIC 投入量产。

基于以上原因,我们想为大家推荐这本非常实用的书籍。虽然目前有大量关于 Verilog 或 FPGA 的书,但它们都偏重于 Verilog 的语法或 FPGA 本身的芯片结构,缺乏对于实际设计经验性的指导。本书作者从大量实际的设计经验出发,采用 Verilog 针对 FPGA,以及从 FPGA 到 ASIC 的设计过程进行了充分的剖析。作者本人是个兴趣十分广泛的人,书中不乏调侃之处,而且由于他的阅历,此书的写作不太像我们熟悉的教科书,反过来正说明它是一本在设计过程中随时可以翻一翻的书。

本书是一本有关 FPGA 综合的书,它试图使读者变成 ASIC 设计者,尽管它不指望让每一个从事 FPGA 设计的人都变成一个 ASIC 设计的专家。这本书是为那些初入此道的人写的,目的是引导他们能够很快上手。这是一本适合从事 FPGA 技术开发人员阅读的书,也可以作为本科生、研究生和教师的参考书。

参与本书翻译工作的有沈树群、张艳、吴京松。由刘丽华审校。该书的翻译出版还得到了科学出版社的大力支持,在此一并表示感谢。译者水平有限,错误和不妥之处在所难免,恳请读者批评指正。

作者的话

这是我出版的第一本书,我把它献给我的妻子 Judy Coffman。我们是在 1972 年 5 月结的婚。从那以后,不论是我在摇滚乐队演奏,还是在夜校攻读我的学士学位;不论和朋友去听音乐会,还是做数不清的兼职,以及写小说(我的合作人是 Mark Bothum),或筹划摇滚乐演出(我的搭档是 Craig Ranta),直到最近编写这本技术方面的书,她自始至终对我都是那样的耐心,一如既往地支持着我。当然,这些年来我也一直在努力工作以养活这个家。在此期间,我们的花园因为常年疏于修整而杂草丛生。

谢谢你,我的爱妻。我发誓,在完成下一本书后,我就去收拾我们的小花园!

序

数字电路设计实践

目前数字电路设计日新月异。器件总是变得更快、更小、集成度更高。10年前大多数人只是偶尔使用 ABEL-HDL 模块化语言进行数字电路设计,且不过几千门而已。到了现在,我们已经能在一个很小的器件里进行百万门级的编程,甚至数百万门级的编程也出现了。对于大多数设计者来说,用原理图输入方式在一个芯片上做系统设计是不实际的。想像一下你怎么处理 1000 页电路图?因此,VHDL 和 Verilog 硬件描述语言应运而生了,尽管人们对孰优孰劣各执一词,但目前它们都有各自的用户群。如果你将用于硬件描述的 C 语言,或者更高级的混合语言也包括进去,那肯定会出现程序语言的多元化。

我个人认为 Verilog 有很强的生命力,之所以这样说是因为很多工程师在决定放弃 VHDL 时,已经知道了 Verilog,也知道了 Verilog 比 VHDL 容易学。因此,熟悉 Verilog 的人没理由再去学习 VHDL,而新入此道的人从 Verilog 入手比从 VHDL 入手要容易得多。

作者进言

如果不考虑个人的观点,实际上设计人员总会在他们的简历里写进他们的 VHDL 和 Verilog 语言技巧。眼下,半数的电子信息技术只有大约 4 年时间的生命周期,而且每过一天就会缩短一天。这就意味着你今天掌握的知识差不多一半在 4 年后将会过时,这真够伤脑筋的。为了不被淘汰,从现在起我们不得不做两件事:

1. 掌握那些永远不会过期的技能,其中包括物理知识(数字电路设计中的模拟技术、传输线理论、能量守恒理论、天线理论和电源管理理论)和诸如同步、亚稳态和传输时延的设计概念。

2. 与时俱进。多参加学术讨论,多读一些有关的杂志,多买一些你认为很有用的书,多关注人们都在谈论些什么。

数字电路设计的水平有高有低。大约 10% 从事 ASIC 设计的精英们在使用价值成千上万的硬件和软件开发工具。他们靠专业性设计谋生。如果 FPGA 设计者使用 5 万门,那么 ASIC 设计者就要使用 50 万门。如果一个 FPGA 设计者使用陈旧

方法,只能做到 4ns(纳秒)的时延,那么一个 ASIC 设计者则能做到小于 1ns 的时延。那些成熟的 ASIC 设计者都是非常谨慎的、有条理的人,他们的计划周密,他们知道一个错误往往是致命的,要花费不知多少金钱才能纠正过来,所以 ASIC 设计者在真正动手之前总是不厌其烦地一遍又一遍地做仿真。

恰恰相反,有 90% 的 FPGA 设计者,他们不够耐心,也很粗心,总想以很小的花费来编辑和调试程序。他们使用廉价的工具,使用 Windows 操作系统的 PC 机。在 ASIC 设计者看来,他们是芸芸众生。这本书就是为那些大多数人写的。

本书是一本有关 FPGA 综合的书,它试图使读者变成 ASIC 设计者,尽管它不指望让每一个从事 FPGA 设计的人都能变成一个 ASIC 设计专家。这本书是为那些初入此道的人而写的,其目的是引导他们很快上手,使他们能够生存。

本书所附的光盘中包括了评估版的 Silos III 仿真器。这个软件包括一个项目/文件管理系统,波形显示工具和完全的 Verilog 语言支持。光盘中还有 David Murray 的漂亮的棱镜编辑器演示版。编辑器具有自动标注反标注特性,诸如 Verilog 关键字高亮显示,关键字加粗打印,行编辑和 Verilog 模板处理。对于此书的购买者来说,花费 40 美元的特价然后注册就可以得到 Prism 编辑器。另外还包括了基于 Windows 的 Emath-Pro 完全功能评估版,它具有 19 个条目,包括有 300 个有用的电子公式工具。

我努力工作试图使这本书更加完美,但是它总是会有纰漏的。如果您发现任何错误或者有争议的问题,期望能随时告诉我!

K. 科夫曼

华盛顿芒特弗农

Kcoffman@sos.net

前言

关于技术的现状

我从加拿大魁北克大学希库蒂米分校毕业并得到工程学士学位后,又在加拿大滑铁卢大学得到了硕士学位。其实那时我根本不知道实验室做出的东西会与现实中的设计有着天壤之别。我曾花费数小时试图弄懂具有异步控制逻辑的行波传送 FIFO,在实际设计中却发现它并没有什么用处。行波二进制计数器是通过将前一位“Q”输出作为后一位的时钟输入来实现的。亚稳定性是听说过,但并没有人教过我应该在哪些地方注意它和怎样减少它带来的一些问题。我需要学习怎样正确地实现一个边沿探测器。我认为我知道什么是瞬态脉冲,但实际上我并不清楚什么时候这样的脉冲会带来问题和如何消除它们。我天真地以为只要遵循完美的制造流程就一定能完成一个设计,而结果却是我最初的设计在功能上都是不完善的。

逻辑综合能力在学校里是学不到的,只有通过实践才能真正得到提高。实际设计时若采用卡诺图来简化逻辑,就好像退回到电子设计历程中的石器时代。在实际设计时,更多的是选择 JK 触发器或者是 T 触发器来简化译码逻辑。

在我的工程课程中所未学到的但后来被证明为最有用的一些技术有:打字(自我高中时代,我母亲就强迫我在 IBM 电动打字机上进行练习)和计算机编程(自学的)。另外,做工程师的经历赋予我如何学习新技巧和如何着手工作的能力。

本书就是关于我在硬件设计中屡次碰壁后所总结出的经验教训。如果我当时有这样一本书并读懂了它,许多错误是可以避免的。此书并不是只有 Verilog 语言的相关知识,如前所述,实际设计中可能遇到的问题及其解决方法才是本书最大的贡献。已经有很多关于 Verilog 语言细节的书籍。本书注重的是实用的硬件设计,在这里 Verilog 只是一种工具。我希望我能再出版一本类似关于如何在实际中进行设计的 VHDL 的书。本书的另一大特点体现在从 HDL 编程验证、综合、器件选择到布局布线等 FPGA 的整个设计流程。很多书都是只对 HDL 的编程做了介绍。

本书中最重要的建议就是在开始编程前弄清到底需要做什么,还要强调的一点是,设计过程 70% 的时间要用来编写测试程序和进行调试。尽管本书对这两点没有给予更多的笔墨。硬件设计的效果如何不是说你到底做了多少,而是要看你设计的硬件是否做到接近生产的程度。

优秀的和可靠的 IP 技术和设计实践,包括 FPGA 和 ASIC 方面的实践的可再用

性是当今的热门课题。本书的内容就是有关设计可再用方面的,虽然在此之前我从未与作者合作过,但我对本书提供的可再用性设计很有信心,它值得信赖。因此我认为本书是 FPGA 和 ASIC 设计新手的必读书籍。

J. 伯杰龙

Qualis 设计公司

janick@qualis.com

感 谢

我很荣幸能与许多有智慧的人一起工作,他们给予了我大量的帮助。阅读了我的手稿并提出许多出色建议的人有:Janick Bergeron, Dr. Sajjan G. Shiva, David Graf。David Pellerin 将我介绍给了编辑 Bernard Goodwin,而 Bernard Goodwin 对我极具耐心。没有 Dave 及时的指导和鼓励,我也不会有这本书。Joan McNamara 则耐心地帮助我通过了作品编辑的过程,Bob Lentz 则为编辑我支离破碎的文章做了大量的工作。

对我产生影响的人还有:Craig Ranta, Rick Penn, G. Scott Bright, Jerrold Gray, Bruce Dippie, Paul Swanson, Dock Brown, Jim Neumiller, Larry Liu, Gary Croft, Mike Kahn, Hal Bridges, Jeff Sanders, Paul Maltseff, Tom Dickens, Michael Irvine, Steve Swedenburg, Tom Dillon, Donn Gabrielson 和 Ed Millett。感谢:Lisa Vartanian——metric.c(包括在光盘中)的作者。

Usenet 新闻组上的质疑有助于我对 FPGA 进行综合的思考并取得进展。这些人包括:Paul Menchini, Peter Alfke, Ray Andraka, Edward Arthur, Rajesh Bawankule, Stuart Sutherland, Tom Coonan, Ben Cohen, Steven Knapp, Austin Franklin, Utku Ozcan, John Cooley。

非常感谢提供软件和支持我的朋友:Dave Pfof, John Bennett; Xilinx 的提供者 Patrick Kane; Exemplar Logic 的提供者 Tom Feist; Simucad 的提供者 Richard Jones; 模型技术的提供者 Dennis Reynolds 和 Dave Kresta。

特别感谢 William M. McDonald 和 Robert Craig (Coolbob) Slaten。RIP 兄弟的指导。

目 录

第一章 Verilog 语言设计实践	1
1.1 小型过热探测器	2
1.2 可综合的 Verilog 要素	6
1.3 Verilog 的层次	10
1.4 内建逻辑原语	12
1.5 锁存器和触发器	14
1.6 阻塞性赋值与非阻塞性赋值	20
1.7 Verilog 语法	24
第二章 数字设计的策略与技巧	37
2.1 设计步骤	37
2.2 数字原语模拟模块的建立	37
2.3 使用 LUT 来实现逻辑功能	39
2.4 关于设计步骤	42
2.5 同步逻辑规则	51
2.6 时钟策略	57
2.7 逻辑化简	60
2.8 综合器做些什么	62
2.9 面积/延时优化	65
第三章 数字电路工具箱	66
3.1 Verilog 层次回顾	66
3.2 三态信号和总线	67
3.3 双向总线	71
3.4 优先编码器	72
3.5 综合中面积/速度的优化	76
3.6 在运行速度和级联时间之间折中	80
3.7 FPGA 逻辑单元的延时	81
3.8 状态机	84
3.9 加法器	94
3.10 减法器	103
3.11 乘法器	104

第四章 更多的数字电路:计数器、只读存储器及随机存储器	108
4.1 行波计数器	108
4.2 约翰逊计数器	109
4.3 线性反馈移位寄存器	111
4.4 循环冗余校验	121
4.5 只读存储器(ROM)	123
4.6 随机存储器(RAM)	125
4.7 先入先出存储器(FIFO)介绍	145
第五章 Verilog 测试	146
5.1 编译指令	147
5.2 自动测试	160
第六章 实用设计:工具、技术及权衡策略	168
6.1 使用 LeonardoSpectrum 进行编译	169
6.2 完整的设计流程,8 位相等比较器	182
6.3 使用层次设计法设计 8 位相等比较器	188
6.4 Xilinx 环境下的优化选项	196
6.5 映射选项	196
6.6 布局/布线选项	198
6.7 逻辑级时序分析报表/版图设计后的时序分析报告	200
6.8 接口选项	202
6.9 VHDL/VERILOG 仿真选项	203
6.10 其他的设计管理器工具	205
第七章 几种架构的比较	212
7.1 决定集成电路价格的因素	212
7.2 FPGA 器件设计	213
7.3 在选择 FPGA 器件时需要考虑的问题	213
7.4 Xilinx 公司 FPGA 器件的架构	215
7.5 Altera 公司 CPLD 器件架构	220
第八章 元件库、可再用模块及 IP	225
8.1 生产率提高的关键	225
8.2 库单元	227
8.3 结构化编程模式	231
8.4 原理图设计和 Verilog 语言设计的比较	233
8.5 使用 LogiBLOX 模块生成器	236
8.6 另一种模块生成器:CORE Generator 工具	237

8.7 设计的再用,重新使用你自己的代码	241
8.8 购买 IP 设计	242
8.9 总结	244
第九章 面向 ASIC 转化的设计	245
9.1 半定制器件	246
9.2 ASIC 转换的设计准则	247
9.3 同步设计规则	248
9.4 延迟线	251
9.5 测试用语	253
9.6 POC 测试向量	255
参考文献	256
光盘使用说明	257
术语表	259
资料索引	266
后记	268
作者介绍	270

第一章 Verilog 语言设计实践

随着科技的快速发展,数字电路设计工程师所面临的挑战发生了戏剧性的变化。设计过程变得更快,门电路的数量在不断增加,物理体积却在减小,芯片的引脚更多了,间距也更密了。然而,底层的设计要素却没有改变,将来也不会改变。所以设计者做出的设计必须遵循以下要求:

- 清晰并易于理解。
- 逻辑的正确性。即设计必须准确无误地实现指定的逻辑功能。设计者应收集用户需求、器件参数并设计输入准则,然后做出满足最终用户需求的设计。
- 能在最恶劣的环境温度和工艺变化中正常运行。随着元器件的老化和温度的变化,电路元件的性能也会发生改变。温度的变化是由元器件自身或外界热源引起的。没有任意两个元器件是完全相同的,特别是在不同时间,不同的制造平台,或是在不同的设计规则下生产的元器件更是如此。器件与时序相关的诸变量,包括时钟偏移、寄存器建立和保持时间、传输时延和输出上升/下降时间等都必须给予考虑,做出说明。
- 可靠性。最终设计不能超出组件功耗的限制。每个元器件都有其工作的温度范围。例如,一个商业用途的器件其温度额定值是 $0 \sim 70^{\circ}\text{C}$ ($32 \sim 160^{\circ}\text{F}$)。器件温度包括环境温度(即产品使用时周围空气的温度)、产品运行时因为内部热源的增加而产生温度和器件自身产生的温度。需注意内部温度的升高与门电路的数量及其逻辑状态改变的速度是成比例的。
- 不会产生满足工作需要及规定以外的 EMI/RFI(中文解释见书后附录,以下同)。
- 是可测试的且被证实满足用户需要。
- 不超出能量消耗的范围(例如,对电池组驱动的电路上)。

无论设计的最终模式如何,也无论使用何种设计和测试工具,以上的要求都必须遵守。

综 合

综合是高级设计描述向目标硬件的转换过程。在本书中,综合代表将 Verilog 代码转换为硬件可识别的设计网表的过程。

数字电路设计者的工作包括编写用于综合的 HDL 代码。这些 HDL 代码将在目标硬件中运行并规定产品的各项操作。设计者还需要编写激励代码用于测试系统输出。设计者编写的代码可读性强,而且要经过编译器的编译才能被硬件设备

最终识别。

为什么选用 HDL?

进行数字设计的方法可谓不少,例如:使用电路示意图。使用电路示意图容易实现更紧凑、更快速和更简明的 FPGA 设计。但是电路示意图方式不具备可移植性,而且在包含 10 或 20 层以上的设计中会变得难以处理。所以,对于大型可移植性系统而言,HDL 是最佳选择。

作为对比,程序列表 1-1 和程序列表 1-2 体现了其他书与本书 Verilog 设计的不同。

程序列表 1-1 其他书的设计

```
//将寄存器 b 的内容传送到寄存器 a  
a <= b;
```

程序列表 1-2 本书的设计

```
/* 作为在大规模设计中以 -3 速度级运行的器件部分,信号 b 需在 7.3ns  
的时间内变换为信号 a。此设计在旁置时电流应小于 80 $\mu$ A,在运行状态  
下应小于 800 $\mu$ A。整个设计在经过 CE 测试和近两个月的编写、调试、  
整合、编制文档并交付给用户以后,其成本应少于 1.47 美元。信号 a  
必须与 75MHz 的系统时钟同步,可随全局系统复位而恢复。输入信号 b  
应被设置在具有 208 条引脚封装结构的第 79 引脚或靠近第 79 引脚处,  
这有助于满足寄存器 a 建立和保持时间的要求。*/
```

```
a <= b;
```

为了说明设计过程,现再举一个常见的例子。此例说明了设计者在进行设计时所面临的问题。即使你对 Verilog 语言不熟悉,也无需担心,在本章稍后的篇章里对此将有详细说明。

1.1 小型过热探测器

以下是 Sarah(工程经理)写给 Sam(硬件设计师)的一封信:

收:sam@engineering

发:sarah@management

主题:热设计项目

一位客户要求设计一个装置:若按住开关或机器过热则红灯变亮,此装置由电池驱动。其最终成本不超过 0.02 美元,这样公司在以每个 9.95 美元出售时才能赢利。

首先, Sam 估计了设计的范围。根据以往的经验,他认为此系统与他去年所设计的电路极其相似。他计算了先前设计所用的门电路数目,并比较了这两个设计之间的差异,确定了这个系统需要 20 个门电路。他还考虑了系统应有的运行速度和其他重要因素,包括他在估计以前设计的复杂程度上可能出现的误差,甚至还考虑了他一周的假期和已定购的机票。他清楚地知道整个工作要包括设计、测试、整合和编制文档。他计算了所需的引脚数,包括一个按键输入、一个过热输入、一个过热输出、至少一个复位和一个时钟输入。根据他所估计的门电路数和引脚数, Sam 选择了一个拥有更多引脚数的设备和运行速度更快的 FPGA 器件以更好服务于用户需求的变化。Sam 把初步的时间计划和所选择的部件上报给老板后就开始工作了。老板对他的工作给予了肯定,并要求他尽快地完成设计工作,而且成本要足够低廉。

现在 Sam 已经完整地考虑了进行设计前的各个事项,可以开始进行设计了。他认为完成整个功能需要一个小型触发器电路,应该采用同步数字电路设计。Sam 编写的 Verilog 程序如下。

程序列表 1-3 过热探测器设计举例

```
module  overheat (clock, reset, overheat _ in, pushbutton _ in , overheat _
    out);
input  clock, reset, ocerheat _ in, pushbutton _ in;
output  overheat _ out;
reg  overheat _ out;
reg  pushbutton _ sync1, pushbutton _ sync2;
reg  ocerheat _ in _ sync1, ocerheat _ in _ sync2;
// 通常情况下,同步输入与系统时钟的相位无关。
// 外部信号使用双同步触发器以最大程度地减小系统的亚稳定性。
// 对于抖动和具有较慢的上升/下降时间的外部信号,进行过滤和锁存,
    效果会更好。

always @ (posedge clock or posedge reset )
begin
    if (reset)
```

```

begin
    pushbutton _ sync1 <= 1'b0;
    pushbutton _ sync2 <= 1'b0;
    overhear _ in _ sync1 <= 1'b0;
    ocerheat _ in _ sync2 <= 1'b0;
end

else begin
    pushbutton _ sync1 <= pushbutton _ in;
    pushbutton _ sync2 <= pushbutton _ sync1;
    overhear _ in _ sync1 <= overhear _ in;
    ocerheat _ in _ sync2 <= overhear _ in _ sync1;
end

end

// 当用户发现过热并按下按钮时,锁存过热输出信号。

always @ (posedge clock or posedge reset )
begin
    if(reset)
        overhear _ out <= 1'b0;

    // overhear _ out 将一直保持(除非重启)。

    else if(overheat _ in _ sync2 && pushbutton _ sync2)
        overhear _ out <= 1'b1;
end

endmodule

```

第一个 always 部分看上去没有任何用处,好像应该可以删除。但是在以前的设计中, Sam 遇到过一些错误的逻辑行为(将在第二章中讨论),所以在实际应用中,他通常都采用双同步输入的方式。第二个 always 部分的作用是在 overhear _ in _ sync 和 pushbutton _ sync 确定后再确定 pushbutton _ out。

Sam 完成了设计中最有趣的部分:代码的实际设计。他很快就运行了编译器、仿真器以确定自己没有拼写错误和语法错误;然后他进行了测试,以检验自己的设计。其测试程序如程序列表 1-4。