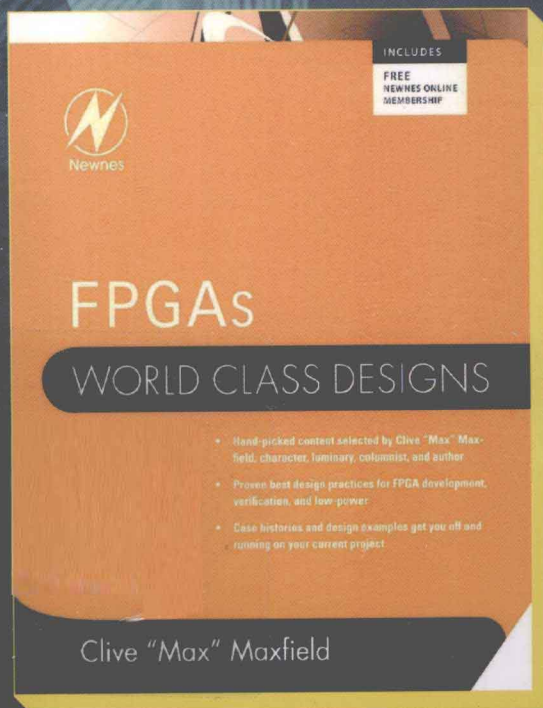


FPGA权威指南

FPGAs World Class Designs

[英] Clive “Max” Maxfield 编著
杜生海 译



TURING

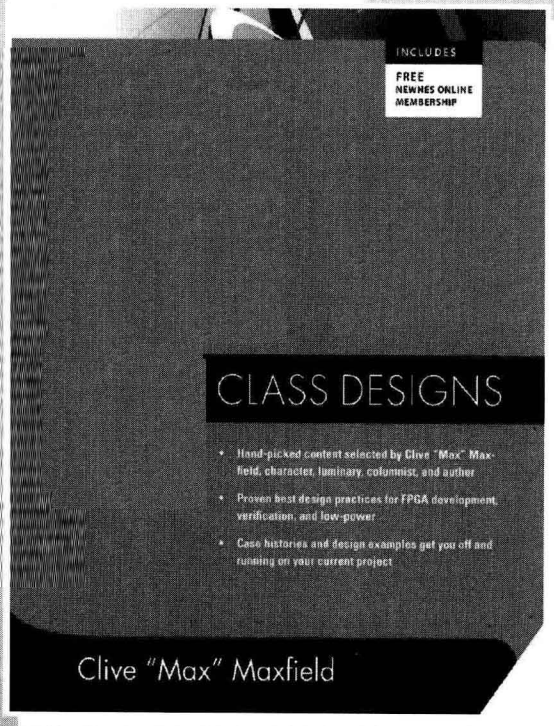
图灵电子与电气工程丛书

FPGA权威指南

FPGAs

World Class Designs

[英] Clive “Max” Maxfield 编著
杜生海 译



人民邮电出版社
北京

图书在版编目(CIP)数据

FPGA权威指南 / (英) 马克斯菲尔德
(Maxfield, C. M.) 编著 ; 杜生海译. — 北京 : 人民邮
电出版社, 2012. 1

(图灵电子与电气工程丛书)

书名原文: FPGAs: World Class Designs

ISBN 978-7-115-26741-2

I. ①F… II. ①马… ②杜… 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(2011)第237993号

内 容 提 要

本书是目前最实用的FPGA类图书的精华集粹, 全书共12章, 内容涵盖FPGA设计基础知识以及FPGA设计全流程。本书重点突出, 实用性强, 所列实例均经过验证。

本书适合电子工程师阅读, 也可作为高等院校相关专业师生的参考指南。

图灵电子与电气工程丛书

FPGA权威指南

-
- ◆ 编 著 [英] Clive “Max” Maxfield
译 杜生海
责任编辑 朱 巍
执行编辑 罗词亮
 - ◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街14号
邮编 100061 电子邮件 315@ptpress.com.cn
网址 <http://www.ptpress.com.cn>
大厂聚鑫印刷有限责任公司印刷
 - ◆ 开本: 787×1092 1/16
印张: 21
字数: 537千字 2012年1月第1版
印数: 1-3 000册 2012年1月河北第1次印刷
- 著作权合同登记号 图字: 01-2010-4014号

ISBN 978-7-115-26741-2

定价: 69.00元

读者服务热线: (010)51095186转604 印装质量热线: (010)67129223

反盗版热线: (010)67171154

版权声明

FPGAs: World Class Designs by Clive “Max” Maxfield, ISBN: 978-1-85617-621-7.

Copyright © 2009 by Elsevier. All rights reserved.

Authorized Simplified Chinese translation edition published by the Proprietor.

ISBN: 978-981-272-452-6.

Copyright © 2011 by Elsevier (Singapore) Pte Ltd. All rights reserved.

Elsevier (Singapore) Pte Ltd.

3 Killiney Road

#08-01 Winsland House I

Singapore 239519

Tel: (65)6349-0200

Fax: (65)6733-1817

First Published 2012

2012年初版

Printed in China by POSTS & TELECOM PRESS under special arrangement with Elsevier (Singapore) Pte Ltd. This edition is authorized for sale in China only, excluding Hong Kong SAR and Taiwan. Unauthorized export of this edition is a violation of the Copyright Act. Violation of this Law is subject to Civil and Criminal Penalties.

本书简体中文版由 Elsevier (Singapore) Pte Ltd. 授权人民邮电出版社在中华人民共和国境内（不包括香港特别行政区和台湾地区）出版与销售。未经许可之出口，视为违反著作权法，将受法律之制裁。

撰稿人简介

Bill Bolton (第 11 章和第 12 章, 可在网站 www.elsevierdirect.com/companions/9781856176217 找到) 是 *Programmable Logic Controllers* 一书的作者。他曾经是英国商业与技术教育委员会 (BTEC) 研究开发与监管部门的主管, 也是联合国教科文组织 (UNESCO) 在阿根廷与泰国的顾问, 英国对巴西技术教育协助项目的成员, 高级物理学纳菲尔德项目成员, 以及大学高级讲师。Bill Bolton 还著有许多成功的工程学教科书。

Robert W. Brodersen (第 10 章) 于 1966 年获得加州州立理工大学波莫纳分校电子工程和数学双学士学位, 随后又分别于 1968 年和 1972 年获得麻省理工学院 (MIT) 电气工程学硕士和博士学位。在达拉斯的德州仪器 (TI) 工作 3 年之后, 他于 1976 年加入加州大学伯克利分校电气工程与计算机科学 (EECS) 系, 在那里从事射频 (RF) 及数字无线通信设计、信号处理应用及设计方法学等研究工作。1994 年, 他成为电气工程与计算机科学系首位 John R. Whinnery 讲座教授。1998 年, 他为建立伯克利无线研究中心 (BWRC) 发挥了重要作用, 这个中心是一个联合体, 其中包括大学研究机构、工业合作伙伴以及政府相关机构, 主要进行高集成度 CMOS 无线系统的设计和开发工作。他于 2006 年退休, 成为名誉教授, 但继续留在伯克利无线研究中心, 作为一名合作科研的领导者, 在 Donald O. Pederson 中心从事电子系统设计工作。

R. C. Cofer (第 7 章) 是 *Rapid System Prototyping with FPGAs* 一书的合著者。

Chen Chang (第 10 章) 是 *Reconfigurable Computing* 的撰稿人。

Rick Gentile (第 8 章和第 9 章) 于 2000 年进入模拟器件公司 (ADI) 担任高级 DSP 应用工程师。现在他是应用工程组的主管, 主要负责在 Blackfin、SHARC 和 TigerSHARC 处理器上的应用工程开发工作。在加入 ADI 之前, Rick 是麻省理工学院林肯实验室技术组成员, 他设计的几个信号处理器被各种各样的雷达传感器所使用。Rick 于 1987 年在马萨诸塞大学阿默斯特分校获得学士学位, 1994 年获得美国东北大学电子与计算机工程双硕士。

Ben Harding (第 7 章) 是 *Rapid System Prototyping with FPGAs* 一书的合著者。

Scott Hauck (第 10 章) 是华盛顿大学电子工程系的教授, 同时也是计算机科学与工程系的兼职教授。他的工作主要集中于 FPGA、实现复杂数字逻辑的可编程和重编程芯片。兴趣爱好是将 FPGA 技术应用于各种能够利用 FPGA 的新奇特性的情况, 例如高性能计算、片上系统 (SOC) 的可重配置子系统、计算机体系结构教育、高光谱图像压缩以及其他领域。Scott Hauck 的工作主要涉及超大规模集成电路 (VLSI)、计算机辅助设计 (CAD) 和 FPGA 及芯片的应用。

David Katz (第 8 章和第 9 章) 有 15 余年电路与系统设计方面的经验。现在, 他是 ADI 公司 Blackfin 应用经理, 主要负责新处理器的详细描述。他在国内和国际刊物上发表过 100 多

篇有关嵌入式处理的文章,还有几篇会议论文。此外,他还是 *Embedded Media Processing* (Newnes 2005) 一书的合著者。之前,他是摩托罗拉公司的一名高级设计工程师,任职于有线调制解调器 (Cable Modem) 和自动化组。David 拥有康奈尔大学电子工程系的学士和工程硕士学位。

Richard Munden (第 4 章) 是 Free Model Foundry 的创始人和 CEO, 于 1987 开始使用和管理计算机辅助工程 (CAE) 系统。从那以后,他就一直从事于模拟与建模方面。Richard 著有一书,该书由 Elsevier 出版。他于 1995 年与人共同创立了 Free Model Foundry, 并担任总裁及 CEO。他在 2006 年全力投入到 Free Model Foundry 的工作中之前曾是位于加州山景城的西门子超声 (以前的 Acuson) 公司的 CAE/PCB 经理。在加入 Acuson 之前, Richard 是位于雷东多海滩的 TRW 公司的一名 CAE 经理。他是多个 EDA 用户组和行业会议的活跃且知名的撰稿人。Richard 专注于板级设计的验证, 读者可以读一读他的博客 blog.FreeModelFoundry.com。

Brian C. Richards (第 10 章) 是 *Reconfigurable Computing* 一书的撰稿人。

Richard Tinder (第 5 章) 在加州大学伯克利分校获得了学士、硕士和博士学位。在 20 世纪 70 年代初期,他在加州大学戴维斯分校做了一年的访问教员,他所在的就是当时的机械工程与材料科学系。在那里,他继续教授材料科学,包括固态热力学和高级反应动力学。接着,他又回到华盛顿州立大学 (WSU), 教授逻辑设计并领导该领域相关的研究,直到 2004 年退休为止。现在,他是华盛顿州立大学电子工程与计算机科学学院的名誉教授。

Peter R. Wilson (第 3 章和第 6 章) 是南安普敦大学的电子学高级讲师。他在英国赫瑞·瓦特大学获得学士学位,在爱丁堡商学院获得 MBA 学位,在南安普敦大学获得博士学位。在重新回到学校之前,曾在 Ferranti、GEC Marconi 与 Analogy 公司工作多年,主要研究领域为航空电子学和电子设计自动化。他出版过多部著作,主要涉及 FPGA 设计、建模与模拟、VHDL、VHDL-AMS、电磁学与电力电子学等。他是 IEEE 的高级会员、IET 成员,还是一位特许工程师。他是 *Design Recipes for FPGAs*^① 一书的作者,并且已经捐献给了 *Circuit Design: Know It All and FPGAs: World Class Designs* 一书。

John Wawrzynek (第 10 章) 是 *Reconfigurable Computing* 一书的撰稿人。

Bob Zeidman (第 2 章) 是 *Designing with FPGAs and CPLDs* 一书的作者,是 Zeidman 咨询公司 (www.ZeidmanConsulting.com) 的总裁,这是硅谷的一家优秀的合同研究和开发公司,他还是软件分析与法医工程公司 (www.SAFE-corp.biz) 的总裁,该公司是软件知识产权分析工具的主要提供者。Bob 被公认为是软件源代码分析与综合领域的先驱,他为综合操作系统编写了 SynthOS™ 程序,还编写了 CodeSuite 程序用来检测软件知识产权被盗用的情况。Bob 同时也是 Verilog 硬件描述语言以及 ASIC 和 FPGA 设计方面的顶级专家。Bob 写过 3 本电子工程教科书 *Verilog Designer's Library*、*Introduction to Verilog* 和 *Designing with FPGAs and CPLDs*, 另外还著有大量的文章与论文。他还在全世界范围内以会议的形式教授工程与商业课程。Bob 是一名在解决知识产权纠纷方面富有经验的、受人尊敬的专家,主要关注于软件源代码、微处理器设计、网络以及其他方面的知识产权纠纷。Bob 拥有 4 项专利,获得了康奈尔大学物理学和电子工程双学士学位,并获得斯坦福大学电子工程硕士学位。

① 中文版:《FPGA 设计实战》,人民邮电出版社,2009。——编者注

前 言

我知道，我对与电子和计算技术有关的所有事情都很着迷，但不论从哪个角度看，今天的现场可编程门阵列（FPGA）都显得“鹤立鸡群”，真是非常棒的器件。

还记得在 20 世纪 60 年代末到 70 年代初，我正在英国约克郡谢菲尔德市上中学。那时，我热衷于订阅电子爱好者杂志《应用电子学》（*Practical Electronics*）。每月月初，在每天上学和放学的路上我都会绕道去报刊店跟那个卖报刊的老人家纠缠一番：“我的杂志到了吗？”当将最近一期杂志牢牢地捧在我那满是粘乎乎热汗的双手中时，我感觉那一天真是幸福极了。

那时候，我对每期杂志中都有的文章“Take Twenty”特别感兴趣。这一文章主要关注的是那些小的电子制作，需要的元器件最多不超过 20 个，成本也不超过 20 先令（见下面的框注）。关于成本这一点，对于一个预算不多的年轻人来说是一个实实在在的优势。

英镑、先令和便士

在我小的时候，英国的货币系统是以英镑（pounds）、先令（shillings）和便士（pence）为基础的，也称作 LSD，其中 L 来源于拉丁语 libra，即镑；D 来源于拉丁语 denarius，这是古罗马时期的一种银币。

这些货币单位的兑换方式为：12 便士兑换 1 先令（也称为 bob），20 先令（即 240 便士）兑换 1 英镑。于是，后来就有了面值 3 便士的硬币，称作 threpeny，还有 6 便士硬币（值 6 便士），也称作 tanner。此外，还有佛罗林（florin）、半克朗（half-crown）、双佛罗林（double-florin）、克朗（crown）、半金镑（half-sovereign）、金镑（sovereign）以及几尼（guinea）等，后来一切都变得越来越复杂。

我们很容易就能从中看出问题。例如，假定某个物品的价格是 1 英镑 8 先令 9 便士，然后让我们算算它值多少个 3 便士硬币（不太容易，是吗？）。因此，英国政府于 1971 年 2 月废除了英镑、先令和便士这些货币单位，官方采用了一种十进制的系统，即 1 英镑等于 100 便士（那时称为“新便士”），就像 1 美元等于 100 美分。然而，似乎很奇怪，大多数英国人拼命抵制货币系统向十进制过渡，并宣称新的货币系统更复杂，永远也不会流行起来！

现在请诚实地回答我，你最近一次在一本关于 FPGA 的书中看到这些有趣的琐事是在什么时候？

不管书中介绍的项目是什么，我都打算努力完成它！只要放学铃声一响，我就立刻飞奔上公交车去最近的电子爱好者商店 Bardwells 购买必需的元器件，然后快速赶回家开始幸福的焊接工作。（有趣的是，尽管那些遥远的日子，已经过去了将近 40 年，Bardwells 商店仍然还在（www.bardwells.co.uk），只不过很久以前就从 Abbeydale 电影院后面的后街角落搬到了 Abbeydale 大街上更加显眼的地方。只要回到英国看望我的母亲和弟弟，我都会顺便拜访一位这位“老朋友”。）

呀，跑题了，言归正传。我早期所有的项目都是用分立（即独立封装的）元件搭成的，这些元件包括晶体管、电阻、电容和电感等。后来，我开始使用简单的 jelly-bean 集成电路^①。数字项目中使用的几乎都是德州仪器（TI）的 14 引脚或 16 引脚双列直插式（Dual-In-Line, DIL）TTL 型 SN74xx 系列器件。一个常见的芯片可能包含 4 个 2 输入与门。在那时，我们都觉得这些器件简直聪明得让人不可思议。

年轻的时候，我最大的一个爱好是机器人。我非常渴望制作一种既能够走路又能够说话的机器，但在当时，这几乎是不可能实现的梦想。在数字设计的早期，控制系统基本上是“硬连线的”（hard-wired）：最初用分立晶体管，后来用简单的集成电路。除了在功能上大多都固定外，这一类型的设计还严重地限制了以快速和节约成本方式所能达到的设计复杂度。再到后来，就出现了微处理器。

世界上第一个商用微处理器是因特尔（Intel）的 4004，它于 1971 年首次亮相。紧接着在 1972 年 11 月，英特尔推出 8008，这本质上是 4004 的 8 位版本。随后，又推出了 4040，主要是扩展了 4004 的加法逻辑和比较指令以增强性能，同时使用了一个很小的内部栈支持子程序嵌套。

实际上，4004、4040 和 8008 都是为专门的应用而设计的，直到 1974 年 4 月，英特尔才推出第一个真正的通用微处理器 8080。这是一个 8 位的微处理器，包含大约 4 500 个晶体管，每秒钟可以执行 20 万次操作，因作为许多早期家庭计算机的中央处理器而名声大噪。

8080 以后，微处理器领域出现了爆炸式的发展，例如摩托罗拉（Motorola）于 1974 年 8 月推出了 6800，MOS 技术公司于 1975 年推出 6502，Zilog 于 1976 年推出了 Z80。

你可能猜到了，20 世纪 70 年代中后期对于数字系统设计者而言是一个非常令人激动的时期。一个微处理器加上少量支持芯片就可以构造出极其复杂的控制系统。更美妙的一点是，如果你发现一个错误或者突然想到一个巧妙的增强方法，那么你所要做的事情仅仅是稍微调整一下运行在微处理器上的程序而已。但是，就像所发生的这一切是那么地令人惊奇一样，我们对这些变化的毫无知觉也让我们惊讶不已。

大约在 20 世纪 80 年代初期，我正负责设计专用集成电路（ASIC）。拿今天的标准来衡量，我当时设计的这些电路简直太简单了——我记得，这些电路都是用 5 微米工艺制造的，只包含大约 2000 个等效逻辑门，但在当时，这却是非常让人惊奇的事情。

1984 年，Xilinx 公司发布了第一个 FPGA（但直到 1985 年，这些器件才真正发货）。尽管这些器件比当时那些简单的可编程逻辑器件（PLD）复杂得多，但大多数数字设计工程师却仅仅用这些器件去实现粘合逻辑（glue-logic）、非常简单的状态机和相当有限的数据处理等。

事情的发展真是太快了。今天，FPGA 已经是最令人激动的器件类型之一。除了具有可编程的体系结构外（正是这种体系结构让 FPGA 可以实现从微处理器软核到硬件加速器的任何东西），它们还包含大量的存储单元和成百上千的硬宏（hard-macro），例如乘法器、加法器和数字信号处理（DSP）模块等。另外，除了数千个可编程的通用输入输出（GPIO）引脚外，它们还支持多个高速串行互连通道，等等，这里就不一一列出了。

各种类型的 FPGA 器件的应用范围之广让人难以置信，从电池供电的便携式手持设备，到自动控制和娱乐系统，再到 SETI（搜寻地外文明）计划中用于搜寻外太空生命的每秒万亿次的计算引擎等。

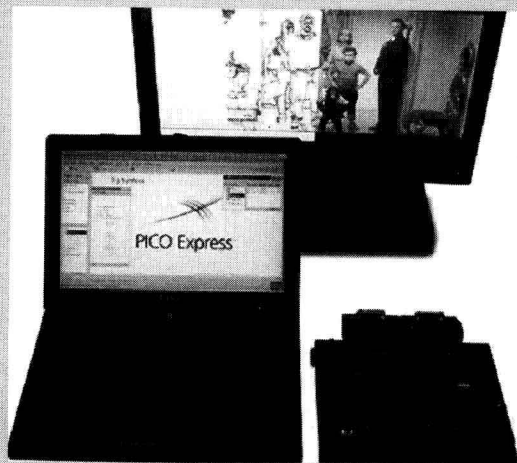
^① 指由一些简单的、固定的逻辑功能组成的电路。——编者注

这真是一个令人振奋的 FPGA 时代。我几乎每天都能看见一些新发布的通告，它们或者关于一系列新的器件，或者关于一种新型的设计工具，或者关于一种可以提高处理性能或 I/O 带宽的新技术。例如，我突然想起，过去几个星期我就了解到以下一些有趣的信息。

- Align Engineering (www.aligneng.com) 提出了 Align 锁相环 (Align Locked Loop, ALL) 的概念，这种锁相环可以让每一个 FPGA 的 LVDS 对成为一个完整的 SERDES (高速串行互连) 解决方案。
- MagnaLynx (www.magnalynx.com) 发布了一种超高性能、少引脚数量的互连技术，可以用于高速嵌入式应用中所使用的 FPGA 和其他器件。
- Actel (www.actel.com) 推出了一系列低成本的 FPGA 器件，内部仅包含大约 15 000 逻辑门，只要 99 美分。
- Altera (www.altera.com) 推出了大量 65 nm 工艺的 Stratix III 器件，其内部含有 340 000 个逻辑单元。
- Lattice Semiconductor (www.latticesemi.com) 致力于具有高端特性的低成本、非易失性 FPGA 器件；同时，一系列达到汽车级别，即获得 AEC-Q100 认证的器件也丰富了其产品类别。
- QuickLogic (www.quicklogic.com) 发布了其超低功耗客户定制标准产品 (Customer-Specific Standard Product, CSSP) 的更复杂的版本，这种产品的特点是具有 FPGA 类型的可编程结构，同时还有专门的硬宏单元。
- Xilinx (www.xilinx.com) 和 Synfora (www.synfora.com) 发布了一种高级算法综合技术 PICO Express FPGA。通过这一技术，可以将无时序特性的 ANSI C 写成的算法转换为相应的 RTL 代码，并进行传统的综合与布局布线 (见下面的注释)。

Pico Express FPGA 算法综合技术

为了演示 PICO Express FPGA 技术的强大，Synfora 的技术人员开发了一套相当酷的 demo 版，可以完成实时色彩转换和 Sobel 边缘检测。如下图所示，用 C 语言写成的无时序算法首先将 RGB 视频流转换为 YUV 格式。



接下来，使用 Sobel 算法检测图像中的边缘。Sobel 算法是一种计算量大、卷积类型的算法，可以计算图像中的梯度，并基于相邻像素之间的梯度差异来识别边缘。

最后，再将处理后的视频流转换为 RGB 格式。如上图所示，图像中一半显示的是边缘，另一半显示的是正常视频，以便作对比。这一切都是在运行频率仅 133 MHz、开箱即用的 Xilinx Spartan 3E 开发板上实时对高清视频进行的。这真是不错，我真想找一个这种小玩意儿自己玩玩！

那么，未来会怎样呢？当我还在写这些文字的时候^①，FPGA 的最新技术水平处于 65 nm 工艺。但是，到 2008 年夏季，至少有一个 FPGA 供应商（Altera）将发布使用 45 nm^②制程的新的系列器件。这些新的 FPGA 将为容量和性能树立全新的标准。例如，据我了解，单个 FPGA 器件将能够实现等价于 1 200 万门 ASIC（专用集成电路）的逻辑，这绝对是不可思议的。

每一代新的 FPGA 器件都会打开一些全新的应用领域，而这些领域在之前是完全不使用可编程逻辑器件的。我们只能去想象这些新一代的器件能够带来什么样的设计方法，我们也只能去想象未来几年会有什么样的新奇与惊喜。我已经等不及了！同时……

在这本书中，为了给读者介绍比较广泛的技术概况，我选了各种各样的主题，主要涉及用 FPGA 实现的项目中用到的设计方法，也覆盖了相关的设计工具以及方法学。我真的希望你喜欢读这些内容，或许有一天，你也会将自己的一部分设计贡献给这本书，到时候我们出下一版。

Clive “Max” Maxfield

^① 那是 2008 年 3 月 1 日，也就是说我必须要说 White Rabbits，因为在英国，每个人在这一天都要互相说这句话，但不要问我为什么。（据说在英国，如果你在每月的第一天说的第一句话是 White Rabbits，那么将会获得好运。

^② 本书出版时，FPGA 的工艺已达到 28 nm 级。——编者注

——译者注

目 录

第 1 章 FPGA 结构比较	1	1.10.2 微处理器软核	18
1.1 一点提醒	1	1.11 时钟树和时间管理器	19
1.2 一些背景信息	2	1.11.1 时钟树	19
1.3 反熔丝、SRAM与其他	3	1.11.2 时钟管理器	19
1.3.1 基于SRAM的器件	3	1.12 通用I/O	22
1.3.2 基于SRAM器件的安全问题和解决方案	3	1.12.1 可配置I/O标准	22
1.3.3 基于反熔丝的器件	4	1.12.2 可配置I/O阻抗	23
1.3.4 基于EPROM的器件	5	1.12.3 核电压与I/O电压	23
1.3.5 基于E ² PROM/FLASH的器件	5	1.13 吉比特收发器	24
1.3.6 FLASH-SRAM混合器件	6	1.14 IP硬核、IP软核与IP固核	25
1.3.7 小结	6	1.15 系统门与实际门	25
1.4 细粒度、中等粒度和粗粒度体系结构	6	1.16 FPGA年龄	27
1.5 基于MUX与基于LUT的逻辑块	8	第 2 章 设计技巧、原则与指导	28
1.5.1 基于MUX的体系结构	8	2.1 硬件描述语言	29
1.5.2 基于LUT的体系结构	8	2.2 自顶向下设计	37
1.5.3 基于MUX还是基于LUT	10	2.2.1 使用HDL	38
1.5.4 3、4、5或6输入LUT	10	2.2.2 书面设计规范	38
1.5.5 LUT、分布式RAM与移位寄存器	10	2.2.3 分配资源	38
1.6 CLB、LAB与slice	11	2.2.4 设计划分	38
1.6.1 Xilinx逻辑单元	12	2.2.5 设计灵活性与优化	38
1.6.2 Altera逻辑部件	12	2.2.6 可重用性	38
1.6.3 slicing和dicing	12	2.2.7 布局规划	39
1.6.4 CLB和LAB	13	2.2.8 验证	39
1.6.5 分布式RAM和移位寄存器	14	2.2.9 了解体系结构	39
1.7 快速进位链	14	2.3 同步设计	40
1.8 嵌入式RAM	15	2.3.1 同步设计五原则	40
1.9 嵌入式乘法器、加法器和MAC等	15	2.3.2 竞争条件	41
1.10 嵌入式处理器核(硬核与软核)	16	2.3.3 延迟相关逻辑	43
1.10.1 微处理器硬核	17	2.3.4 保持时间违例	44
		2.3.5 毛刺	45
		2.3.6 门控时钟	47
		2.3.7 异步信号与亚稳态	47

2.3.8 允许使用异步逻辑的情况	51	3.6.3 for语句	72
2.4 浮动节点	52	3.6.4 while循环	72
2.5 总线竞争	53	3.6.5 exit语句	73
2.6 独热状态编码	54	3.6.6 next语句	73
2.7 可测性设计	55	3.7 层次化设计	73
2.8 测试冗余逻辑	56	3.7.1 函数	73
2.8.1 什么是冗余逻辑	56	3.7.2 包	73
2.8.2 怎样测试冗余逻辑	57	3.7.3 元件	74
2.9 初始化状态机	58	3.7.4 过程	75
2.10 可观测节点	58	3.8 调试模型	75
2.11 扫描技术	59	3.9 基本数据类型	75
2.12 内建自测试	60	3.9.1 基本类型	75
2.13 特征分析	61	3.9.2 数据类型: bit	76
2.14 小结	62	3.9.3 数据类型: Boolean	76
第3章 VHDL 基础	64	3.9.4 数据类型: 整数	76
3.1 引言	64	3.9.5 数据类型: 字符型	77
3.2 实体: 模型接口	65	3.9.6 数据类型: 实数	77
3.2.1 实体定义	65	3.9.7 数据类型: 时间	77
3.2.2 端口	66	3.10 小结	77
3.2.3 通用属性语句	66	第4章 存储器建模	78
3.2.4 常数	66	4.1 存储器阵列	78
3.2.5 实体举例	67	4.1.1 Shelor方法	79
3.3 构造体: 模型行为	67	4.1.2 VITAL_Memory包	80
3.3.1 构造体的基本定义	67	4.2 存储器功能建模	80
3.3.2 构造体声明	67	4.2.1 使用行为模型方法	80
3.3.3 构造体语句	68	4.2.2 使用VITAL2000方法	89
3.4 进程: VHDL中的基本功能单元	68	4.3 VITAL_Memory路径延迟	95
3.5 基本变量类型和操作符	69	4.4 VITAL_Memory时序约束	97
3.5.1 常数	69	4.5 预加载存储器	97
3.5.2 信号	69	4.5.1 行为存储器预加载	98
3.5.3 变量	69	4.5.2 VITAL_Memory预加载	100
3.5.4 布尔操作符	69	4.6 其他类型存储器的建模	100
3.5.5 算术操作符	70	4.6.1 同步静态RAM	100
3.5.6 比较操作符	70	4.6.2 DRAM	103
3.5.7 移位函数	70	4.6.3 SDRAM	104
3.5.8 拼接	70	4.7 小结	109
3.6 判断与循环	71	第5章 同步状态机设计与分析	110
3.6.1 if-then-else语句	71	5.1 引言	110
3.6.2 case语句	72	5.2 时序状态机模型	112

5.3 全记录状态图	115	5.12.2 序列检测器的设计: D到 JK卡诺图的转换	152
5.4 基本记忆单元	118	5.13 简单状态机分析	156
5.4.1 置位优先基本单元	118	5.14 简单状态机的VHDL描述	159
5.4.2 复位优先基本单元	120	5.14.1 上升沿D触发器的VHDL 行为级描述	160
5.4.3 激励表组合形式	121	5.14.2 简单状态机的VHDL行为 级描述	160
5.4.4 基本单元的混合输出	122	参考文献	161
5.4.5 基本单元的混合输出响应	123	第6章 嵌入式处理器	163
5.5 触发器简介	124	6.1 引言	163
5.5.1 触发机制	125	6.2 简单的嵌入式处理器	164
5.5.2 触发器类型	126	6.2.1 嵌入式处理器体系结构	164
5.5.3 触发器设计的层次化流程 图和模型	126	6.2.2 基本指令	165
5.6 FSM (触发器) 设计步骤及映射 算法	127	6.2.3 取指执行周期	166
5.7 D触发器: 通用型	128	6.2.4 嵌入式处理器的寄存器 分配	166
5.7.1 D锁存器	128	6.2.5 基本指令集	167
5.7.2 上升沿触发D触发器	131	6.2.6 结构级还是行为级	168
5.7.3 主从式D触发器	134	6.2.7 机器码指令集	168
5.8 触发器的转换: T、JK触发器以及 其他触发器	136	6.2.8 微处理器的结构单元	169
5.8.1 T触发器及其从D触发器的 转换方法	137	6.2.9 处理器函数包	169
5.8.2 JK触发器及其从D触发器 转换的方法	138	6.2.10 程序计数器	170
5.8.3 用JK触发器设计T触发器 和D触发器	140	6.2.11 指令寄存器	171
5.8.4 激励表回顾	141	6.2.12 算术逻辑单元	172
5.8.5 专用触发器和锁存器的 设计	142	6.2.13 存储器	173
5.9 锁存器和触发器中的严重时序 问题: 警告	145	6.2.14 微控制器	175
5.10 异步预置位和复位	146	6.2.15 简单微处理器总结	178
5.11 触发器的建立时间和保持时间 要求	147	6.3 FPGA中的软核处理器	178
5.12 使用边缘触发器设计简单的 同步状态机: 映射转换	149	6.4 小结	178
5.12.1 三比特二进制加减计数 器设计: D到T的卡诺图 转换	149	第7章 数字信号处理	179
		7.1 概述	179
		7.2 基本DSP系统	180
		7.3 基本DSP术语	181
		7.4 DSP体系结构	182
		7.5 DSP元件中的并行执行	183
		7.6 FPGA中的并行执行	184
		7.7 何时使用FPGA实现DSP功能	185

7.8	FPGA的DSP设计考虑	186	9.2	广播电视系统——NTSC和PAL制式	220
7.8.1	时钟与信号的布线	186	9.2.1	视频分辨率	220
7.8.2	流水线	187	9.2.2	隔行扫描和逐行扫描	221
7.8.3	算法实现的选择	187	9.3	颜色空间	222
7.8.4	DSP知识产权	187	9.3.1	伽马校正	222
7.9	FIR滤波器概念举例	188	9.3.2	色度下采样	224
7.10	小结	189	9.4	数字视频	226
第8章	嵌入式音频处理基础	191	9.4.1	ITU-R BT.601 (前称为CCIR-601)	226
8.1	引言	191	9.4.2	ITU-R BT.656 (前称为CCIR-656)	227
8.1.1	声音是什么	192	9.5	从系统角度看视频	230
8.1.2	音频信号	193	9.5.1	视频源	230
8.1.3	语音处理	193	9.5.2	视频显示	235
8.2	音频信源与音频信宿	194	9.6	嵌入式视频处理考虑	238
8.2.1	在模拟与数字音频信号之间转换	194	9.6.1	视频端口特性	238
8.2.2	音频转换器背景知识	195	9.6.2	视频ALU	240
8.2.3	连接到音频转换器	196	9.6.3	DMA考虑	242
8.3	互连	199	9.6.4	视频算法分类	244
8.3.1	连接器	199	9.6.5	带宽计算	244
8.3.2	数字连接	200	9.6.6	去隔行处理	246
8.4	动态范围与精度	200	9.6.7	扫描速率转换	246
8.5	音频处理方法	207	9.6.8	像素处理	247
8.5.1	如何将数据输入到处理器内核	207	9.6.9	处理图像边界	248
8.5.2	块处理与采样处理	207	9.6.10	色度重采样、伽马校正和颜色转换	249
8.5.3	双缓存	207	9.6.11	缩放与剪切	249
8.5.4	二维DMA	208	9.6.12	显示处理	250
8.5.5	基本操作	208	9.7	压缩和解压缩	251
8.5.6	信号生成	210	9.7.1	无损和有损压缩	252
8.5.7	滤波与算法	210	9.7.2	图像压缩	253
8.5.8	采样率变换	211	9.7.3	视频压缩	254
8.5.9	音频压缩	212	9.7.4	EMP中的编码与解码	257
8.5.10	语音压缩	214	参考文献	参考文献	259
参考文献	参考文献	216	第10章	利用 Simulink 中的框图设计流式 FPGA 应用	261
第9章	嵌入式视频与图像处理基础	217	10.1	使用基于流的操作符设计高性能数据路径	262
9.1	引言	217			
9.1.1	人类视觉感知	218			
9.1.2	什么是视频信号	219			

10.2 图像处理设计引擎	263	11.2.1 与	282
10.2.1 将RGB视频转换为灰度 视频	263	11.2.2 或	283
10.2.2 二维视频滤波	265	11.2.3 非	283
10.2.3 将视频滤波器映射到BEE2 FPGA开发平台	268	11.2.4 与非	284
10.3 在Simulink中加入控制	269	11.2.5 或非	285
10.3.1 使用Simulink块设计控 制器	270	11.2.6 异或	286
10.3.2 使用Matlab M语言设计 控制器	270	11.3 锁存器	286
10.3.3 使用VHDL或Verilog设计 控制器	272	11.4 多路输出	287
10.3.4 使用嵌入式微处理器设计 控制器	272	11.5 输入程序	289
10.4 组件重用：简单与复杂子系统库	273	11.6 功能框图	290
10.4.1 信号处理元件	273	11.6.1 逻辑门	291
10.4.2 瓦片式子系统	273	11.6.2 布尔代数	293
10.5 小结	275	11.7 编程举例	295
致谢	276	第 12 章 定时器	298
参考文献	276	12.1 定时器类型	298
第 11 章 梯形图与功能框图编程	278	12.2 对定时器编程	299
11.1 梯形图	279	12.2.1 序列	299
11.2 逻辑功能	281	12.2.2 级联定时器	301
		12.2.3 循环开关定时器	302
		12.3 延迟关定时器	302
		12.4 脉冲定时器	303
		12.5 编程实例	305
		索引	306

第 1 章 FPGA 结构比较

Clive Maxfield

本章摘自我的一本书，书名为 *The Design Warrior's Guide to FPGAs: Devices, Tools, and Flows*^①，书号为 9780750676045。实际上，不管书名是什么，我想它只不过是想提醒读者，所谓设计指南其实并不是要教你如何用地 FPGA 实现设计，而是提供一些广泛的一般性概念的详细介绍。例如，FPGA 是什么，FPGA 如何工作，如何对 FPGA 进行编程，基于原理图、基于 HDL、基于 C/C++、基于 DSP、基于嵌入式处理器等各类设计的不同设计流程。

现在看起来，从我自己的书中选择一章作为本书的开篇有点不太谦虚，不过确实有很多不同结构类型的 FPGA 可供我们使用，要弄清楚所有的类型恐怕是一件非常困难的事情。另外，FPGA 厂商的网站和技术支持材料中往往只介绍他们自己的技术，并倾向于贬低其他技术，这就令局面更加混乱。

为了说明这个问题，本章列举了所有可能的结构。首先，我们要了解基于 SRAM、基于反熔丝技术、基于 Flash 以及基于 Flash 与 SRAM 混合技术之间的差异。其次，要了解基于 MUX 和基于 LUT 的逻辑块。最后还有将 LUT 作为分布式 RAM 或移位寄存器、快速进位链、嵌入式块 RAM、嵌入式乘法器、DSP 元件、微处理器软核与硬核、时钟产生器、高速串行收发器等使用。

另外，也解决了一个棘手的问题，那就是“系统门”与实际逻辑门的意义。答案可能会（也可能不会）令你大吃一惊。

——Clive “Max” Maxfield

1

1.1 一点提醒

这一章，我们介绍 FPGA 多种多样的结构特点。某些结构（比如使用反熔丝还是 SRAM 配置单元）是无法共存的。有的 FPGA 厂商专注于其中的一种，还有的厂商提供基于这些不同技术的多个器件系列。（除非另外注明，这里主要讨论基于 SRAM 的器件。）

对于不同的嵌入式模块，比如乘法器、加法器、存储器和微处理器核，不同的厂商有不同原料的“食谱”来满足不同“口味”。（就像不同牌子的巧克力，巧克力块有大有小。某些 FPGA 器件系列也会有大型/中型/小型的嵌入式 RAM 块，其他的可能有更多的乘法器、支持更多的 I/O 标准等。）

^① 中文版：《FPGA 设计指南：器件、工具和流程》，人民邮电出版社，2007。——编者注

问题是每个厂商和每个器件系列所支持的特性几乎每天都在变化。这意味着一旦你决定所需要的特性，就需要做一些研究来看看哪个厂商当前提供的器件最能够满足你的要求。

1.2 一些背景信息

在讲述本章的主要部分之前，我们需要弄清楚一些概念，以便在统一的鼓点下前进。例如，你将看到“结构”（fabric）一词贯穿全书，在讨论硅芯片时，它是指器件的基础结构（有点像短语“文明社会的结构”）。

单词 fabric 来源于中世纪英语 fabryke，意思是“建造好的某些东西”。

当你首次听到有人这样来使用“结构”一词，可能会觉得有点傲气或自负（实际上，有些工程师认为这个词仅仅是一个由 ASIC 和 FPGA 厂商推广的市场概念，目的是使他们的器件听上去比实际更加复杂精密）。不过，说真的，一旦你习惯了之后，就会感到它确实是一个很有用的词汇。

2

当我们谈到 IC 的“几何结构”时，指的是建造在芯片上的独立结构的尺寸，比如场效应管（FET）的沟道。这些结构小到令人难以置信。在 20 世纪 80 年代的早中期，器件是基于 $3\mu\text{m}$ 的几何结构，这意味着它们的最小结构在尺寸上只有 1m 的百万分之三（在与交流时，我们可以说，“这个 IC 基于 $3\mu\text{m}$ 技术”）。

符号 μ 代表 micro，来源于希腊语“micros”，意思是微小（因此 μP 一般作为微处理器的缩写）。

每个新的几何结构被称为“工艺节点”。在 20 世纪 90 年代，基于 $1\mu\text{m}$ 几何结构的器件开始出现，而且特征尺寸在这个十年的过程中持续快速减小。到了 21 世纪，高性能 IC 已经具有 $0.18\mu\text{m}$ 的几何结构了。到了 2002 年，发展到了 $0.13\mu\text{m}$ ；2003 年，基于 $0.09\mu\text{m}$ 的器件开始出现了。

在米制体系中， μ 代表百万分之一，因此 $1\mu\text{m}$ 就是百万分之一米。

所有小于 $0.5\mu\text{m}$ 的几何结构被划分为深亚微米（DSM）。按照某些还未确定的观点（或者那些来自不同谈话者的多重定义），我们在向超深亚微米（UDSM）领域迈进。

当几何结构减小到 $1\mu\text{m}$ 以下时，事情开始变得有些不方便，因为我们不得不比较啰嗦地说“零点一三微米”之类的话。由于这个原因，nm 的使用逐渐流行起来，这里的 nm 是纳米（nano）的缩写， 1nm 等于 $1\mu\text{m}$ 的千分之一，也就是百万分之一米的千分之一。这样，不必那么啰嗦地说“零点零九微米”，可以直接说“90nm”就可以了。当然，这都是在说同一回事，但是如果你想在这些话题上和朋友们聊得很愉快，最好还是使用这些约定俗成的术语，跟上潮流，而不是表现得唠唠叨叨的、好像是来自千禧年之前的老古董。

3