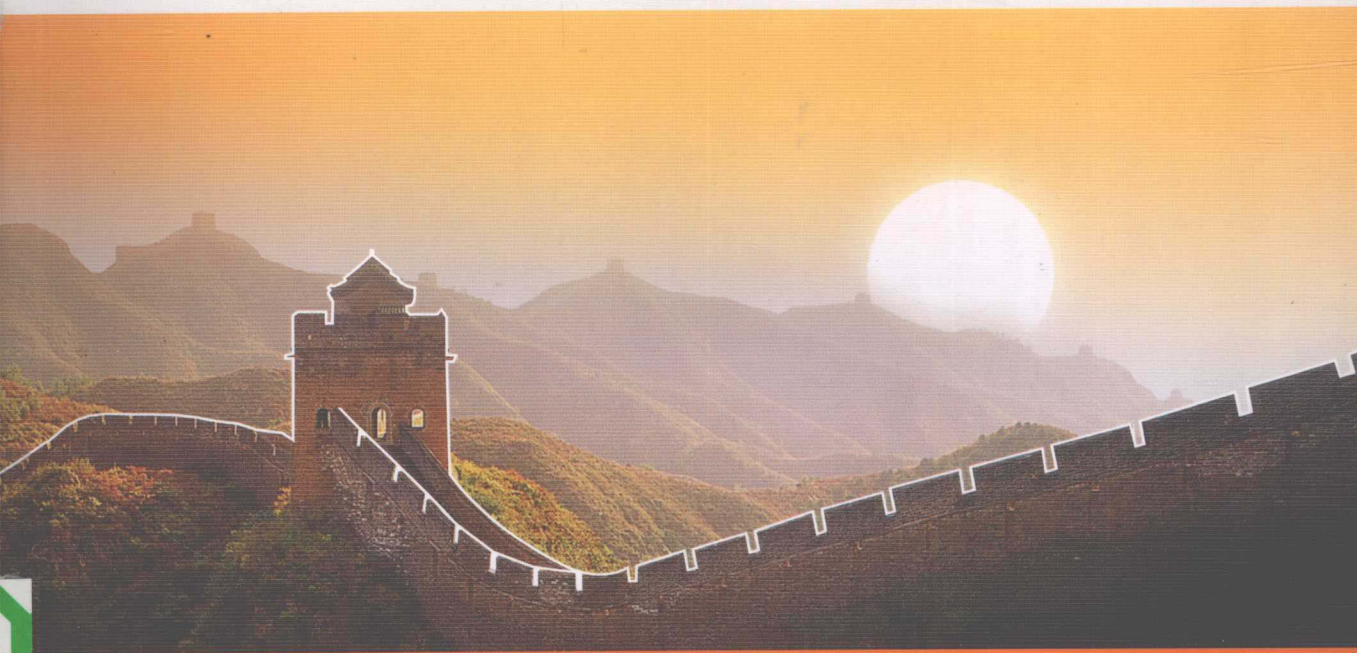


FPGA 数字逻辑设计教程 —— Verilog

Digital Design Using Digilent FPGA Boards
— Verilog/Active-HDL Edition



[美]Richard E. Haskell Darrin M. Hanna 著

郑利浩 王荃 陈华锋 译



电子工业出版社

PUBLISHING HOUSE OF ELECTRONICS INDUSTRY

<http://www.phei.com.cn>

Digital Design Using Digilent FPGA Boards Verilog/Active-HDL Edition

FPGA 数字逻辑设计教程 ——Verilog

電子工業出版社

Publishing House of Electronics Industry

北京 • BEIJING

内 容 简 介

本书根据课堂教学的要求和实际操作的需要,以培养实际工程设计能力为目的,系统地介绍了利用 Verilog 硬件描述语言进行数字电路设计和 FPGA 开发的方法。其中,第 1~4 章系统地介绍了数字电路设计的背景及基础知识;第 5~8 章详细地叙述了数字电路的设计方法及其 Verilog 代码的实现;第 9~13 章对存储器、VGA、PS/2 等常见硬件系统的实现做了介绍。

本书简明扼要、通俗易懂,大量的实例贯穿始终,书中对每一个实例均给出了在 FPGA 上的具体实现方法。在内容编排上,由浅入深,将数字电路设计和 Verilog 语言的学习有机结合在一起,抛弃了传统的烦琐单调的语法讲解篇章,使读者在感受数字电路设计乐趣的同时,轻松地掌握 Verilog 语言。此外,通过本书的学习读者还能够了解代表当今数字电路设计前沿的 FPGA 开发的设计流程。

本书可作为大学本科相关专业教材和 FPGA 培训教材,也可供电子电路工程师和爱好者自学参考。

Translation from the English language edition:

Digital Design Using Digilent FPGA Boards - Verilog/Active-HDL Edition By Richard E. Haskell and Darrin M. Hanna

ISBN:978-0-9801337-7-6

Copyright©2009 by LBE Books, LLC. All rights reserved

本书中文简体专有翻译出版版权由美国 DIGILENT 科技有限公司授予电子工业出版社。该专有出版版权受法律保护。

版权贸易合同登记号 图字:01-2010-5760

图书在版编目(CIP)数据

FPGA 数字逻辑设计教程: Verilog / (美) 哈斯凯尔 (Haskell, R.E.), (美) 汉纳 (Hanna, D.M.) 著; 郑利浩, 王荃, 陈华锋译. —北京: 电子工业出版社, 2010.9

书名原文: Digital Design Using Digilent FPGA Boards - Verilog/Active-HDL Edition

ISBN 978-7-121-11852-4

I. ①F… II. ①哈… ②汉… ③郑… ④王… ⑤陈… III. ①数字逻辑—逻辑设计 ②硬件描述语言, VHDL—程序设计 IV. ①TP302.2

中国版本图书馆 CIP 数据核字 (2010) 第 182649 号

责任编辑: 王敬栋 (wangjd@phei.com.cn)

印 刷: 北京东光印刷厂

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: 787×1 092 1/16 印张: 21 字数: 538 千字

印 次: 2010 年 9 月第 1 次印刷

印 数: 6 000 册 定价: 48.00 元

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 zlts@phei.com.cn, 盗版侵权举报请发邮件至 dbqq@phei.com.cn。

服务热线: (010) 88258888。

Preface

译者序

从美国 DIGILENT 科技有限公司中国区总裁赵峰处拿到《Digital Design Using Digilent FPGA Boards Verilog/Active-HDL Edition》英文原版教材后,仔细看了教材,感觉非常兴奋。该书的内容将数字逻辑设计和 Verilog 语言有机地结合在一起,能够使没有一点基础的读者快速进入现代数字逻辑设计的世界,它非常适合国内当前数字逻辑设计和 EDA 课程教学改革的趋势,译稿的书名为了体现这一特色,命名为《FPGA 数字逻辑设计教程——Verilog》。

翻译过程是漫长和艰苦的,整个翻译过程,加上程序的验证仿真、校稿等持续了将近半年多时间,书中的每一程序,我们都经过逐一验证。全书内容安排合理,循序渐进,前 4 章结合 Verilog 语言讲述了基本逻辑门、逻辑表达式、布尔代数及卡诺图等基础知识,第 5~8 章通过 Verilog 语言的例程讲述了组合逻辑、运算电路、时序电路及状态机等典型数字电路的设计,之后章节也是通过大量的例程讲述了存储器、VGA 显示和 PS/2 接口等的设计。

书中的 76 个例程都可以用美国 DIGILENT 科技有限公司的 BASYS2 和 NEXYS2 开发板实现,对于例程的仿真和综合,考虑到国内使用的习惯,我们将原版中的 Active-HDL 设计工具替换为 Xilinx ISE 10.1。关于开发板的相关资料及书中用到的一些其他资源,读者可以在 <http://www.digilent.cn> 或 www.winfpga.com 网站上找到,并免费下载,书中所使用的设计工具(仿真及综合)Xilinx ISE 10.1,读者可以在 Xilinx 官方网站(<http://www.xilinx.com>)上下载该软件。

本书主要由郑利浩、王荃和陈华锋完成翻译工作,郑利浩最后对全书进行了统稿。在本书的翻译校稿过程中,得到了杭州文源通信技术有限公司的大力支持和帮助,公司的王伟捷对全书的所有例程进行了仿真验证,在此表示衷心的感谢。

由于译者水平有限,加上时间比较仓促,书中不妥之处在所难免,敬请读者批评指正,读者可通过电子邮件(zlh3065@163.com)与本书译者进行交流。

译者

Preface

原著前言

在过去的几十年中，数字电路设计经历了革命性的发展。现场可编程门阵列（FPGAs）现在可以达到上百万门，同时可以拥有上万个翻转触发器。这意味着，用传统的逻辑设计方法分析电路不再可行。比如用传统画逻辑图的方法，分析包含上千个门的数字电路，是不可行的。实际上，现在的工程师都是通过编写硬件描述语言来设计数字系统。目前最常用的硬件描述语言是 VHDL 和 Verilog。两者的应用都很广泛。工程师们在应用这些硬件描述语言的时候，通常更倾向于对逻辑电路的行为级描述，而不是写传统的布尔逻辑方程。计算机辅助设计工具也不仅用来仿真 VHDL 或 Verilog 语言的设计，还用来实现从设计到实际硬件的综合。

本书内容的安排是在假定读者没有任何的数字电路知识的基础上进行的。本书用 76 个例程来告诉读者，数字电路的设计应该从何处着手。这些例程都可以在 Spartan3E 系列的 FPGA 上实现。本书主要采用美国 DIGILENT 科技有限公司的 BASYSTM 开发板和 NEXYS2 开发板，这两个开发板中的任何一个都可以完全实现书中的所有例程，两套开发板都可以在 <http://www.digilentinc.com> 上购买。本书使用 Aldec 公司的 Active-HDL 软件，完成数字系统的设计、仿真、综合和实现。读者可以在 Aldec 的官网（<http://www.aldec.com>）上下载 Active-HDL 软件的学生免费版本。为了能将读者自己的设计综合到 Spartan 3E 的 FPGA 上，读者还需要从 Xilinx 的官网（<http://www.xilinx.com>）上下载免费的 ISE WebPACK 软件。Xilinx 的综合工具可以在 Active-HDL 软件中调用，具体方法在 Aldec 公司 Active-HDL 软件的用户指导手册中有详细说明。读者可以使用 ExPort 工具将综合后的设计下载到 Spartan 3E 的 FPGA 上。ExPort 是 Adept 软件套件的一部分，这套软件可以在 Digilent 的官网（<http://www.digilentinc.com>）上免费下载。

在本书的编写过程中，许多同行和学生都给予了帮助。对他们的探讨精神、对问题的钻研精神表示敬佩。同时，感谢他们对我们提出的宝贵意见。

Richard E. Haskell
Darrin M. Hanna

目 录

Chapter 01 概述

1.1 背景	2
1.2 数字逻辑	5
1.3 Verilog	7

Chapter 02 基本逻辑门

2.1 真值表和逻辑表达式	9
2.1.1 3种基本逻辑门	9
2.1.2 4种常用逻辑门	10
2.2 正逻辑和负逻辑：德摩根定律	12
2.3 基于乘积和的设计	14
2.4 基于和项积的设计	15
习题	21

Chapter 03 布尔代数和逻辑表达式

3.1 布尔定律 (Boolean Theorems)	26
3.1.1 单变量布尔定律	26
3.1.2 双变量和三变量的布尔定律	27
3.2 卡诺图 (Karnaugh Maps)	30
3.2.1 2变量卡诺图	31
3.2.2 3变量卡诺图	32
3.2.3 4变量卡诺图	34
3.3 计算机化简方法	35
3.3.1 乘积项的表格形式	35
3.3.2 素蕴含项	36
3.3.3 本质素蕴含项	38
习题	43

Chapter 04 实现数字电路

4.1 实现门	46
4.2 晶体管—晶体管逻辑 (TTL)	48
4.3 可编程逻辑器件 (PLD 和 CPLD)	49
4.3.1 一个2输入1输出的PLD	49
4.3.2 GAL 16V8	50

4.3.3 CPLD	52
4.4 现场可编程门阵列 (FPGA)	52
习题	55

Chapter 05 组合逻辑

5.1 多路选择器 (Multiplexer)	58
5.1.1 2 选 1 多路选择器	58
5.1.2 4 选 1 多路选择器	59
5.1.3 4 位 2 选 1 多路选择器	60
5.1.4 4 选 1 多路选择器的逻辑方程	65
5.2 7 段显示管	73
5.3 比较器	86
5.3.1 级联比较器	86
5.3.2 TTL 比较器	87
5.4 译码器和编码器	92
5.4.1 译码器 (Decoders)	92
5.4.2 TTL 译码器	93
5.4.3 编码器 (Encoder)	94
5.4.4 优先编码器 (Priority Encoder)	94
5.4.5 TTL 编码器	95
5.5 编码转换器	101
5.5.1 二进制—BCD 码转换器	101
5.5.2 移位加 3 算法 (Shift and Add 3 Algorithm)	101
5.5.3 格雷码转换器 (Gray Code Converters)	103
习题	108

Chapter 06 运算电路

6.1 加法器	112
6.1.1 半加器	112
6.1.2 全加器	112
6.1.3 进位和溢出	114
6.1.4 TTL 加法器	116
6.2 减法器	120
6.2.1 半减器	120
6.2.2 全减器	121
6.2.3 加/减法电路	122
6.3 移位器	125
6.4 乘法运算	126
6.4.1 二进制乘法	126
6.4.2 有符号乘法	129

6.5 除法运算	132
6.6 算术逻辑单元 (ALU)	135
习题	138

Chapter 07 时序电路

7.1 锁存器和触发器	142
7.1.1 SR 锁存器	142
7.1.2 时钟触发 SR 锁存器	143
7.1.3 D 锁存器	143
7.1.4 边沿触发的 D 触发器	144
7.2 寄存器 (Registers)	151
7.3 移位寄存器	156
7.4 计数器 (Counters)	162
7.4.1 任意波形的实现	164
7.4.2 3 位计数器的 Verilog 行为描述	166
7.5 脉冲宽度调制器 (PWM)	175
7.5.1 使用 PWM 控制一个直流电动机的速度	176
7.5.2 使用 PWM 控制一个伺服电动机的位置	177
7.5.3 产生一个频率为 2 kHz 的 PWM 信号	179
7.6 BASYS2/ NEXYS2 板例程	181
习题	192

Chapter 08 有限状态机

8.1 米里状态机和摩尔状态机	196
8.2 摩尔状态机序列检测器	196
8.3 米里状态机序列检测器	198
习题	213

Chapter 09 数据通道和控制单元

9.1 Verilog 中的 while 语句	215
9.2 数据通道和控制单元	216

Chapter 10 整合数据通道和控制单元

10.1 改进的 GCD 算法	238
10.2 改进的整数平方根算法	243

Chapter 11 存储器 (Memory)

11.1 只读存储器	249
11.2 分布式的存储器	253

Chapter 12 VGA 控制器

12.1 时序	263
---------	-----

12.2 其他标准图像模式	265
Chapter 13 PS/2 接口	
13.1 概述	289
13.2 键盘	291
13.3 鼠标	295
附录 A 代码仿真及设计实现	305
附录 B 数字系统	314
附录 C Verilog 快速参考指南	324

1 Chapter 01

概 述



在本章你将学习：

- ◎ 计算机和数字逻辑发展背景
- ◎ 摩尔定律
- ◎ 集成电路的发展和数字电路的设计
- ◎ 什么是 Verilog

1.1 背 景

在过去的 60 年中，数字逻辑改变了世界。众所周知的计算机起源于“二战”时期，1944 年宾夕法尼亚大学摩尔电子工程学院设计建造了第一台电子计算机 ENIAC，它首先应用在军方炮兵部队进行的弹道模拟运算上。具体的计算机发展史我们可以从表 1.1 中看到。

表 1.1 部分计算机和数字逻辑的重大事件

日 期	事 件
3000BC	古巴比伦人用金属丝或杆串扁珠子来代表数字，直到今天它还被远东的一些地方所沿用
1614-1617	John Napier，苏格兰数学家，主要发明了对数，它可以通过指数相加来完成乘法运算，通过指数相减来完成除法运算。他还发明了纳皮尔尺，可以通过特殊的方式移动尺子进行乘除运算
1623	Wilhelm Schickard，德国教授，发明了第一台机械计算器，被称为计算钟
1630	William Oughtred，英国数学家和教士，发明了计算尺
1642-1644	Blaise Pascal，法国数学家，物理学家，宗教哲学家，发明了 Pascaline，第一个机械式的计算器
1672-1674	Gottfried Wilhelm Von Leibniz，德国数学家，外交家，历史学家，法理学家，以及不同微积分的发明家，发明了被称为步进计算器的机械计算器。该计算器用特殊齿轮——Leibniz 轮来实现乘法。虽然计算器从未成功工作过，但它的设计对以后的机械计算机有重大的影响
1823-1839	Charles Babbage，英国数学家和发明家，为自动计算对数表而开始差分机的研究。尽管政府投入了大量资金，这台差分机最终未能完成。1834 年，Babbage 开始从事一种更强大的计算分析机的研究，分析机被认为是第一台通用计算机。该设计超前了当时时代 100 年，那时还不能提供设计要求的精密齿轮，所以分析机未能成功运作。Babbage 被公认为是计算机之父
1854	George Boole，英国逻辑学家和数学家，出版了《思维规律的研究》，是逻辑数学的基石
1890	Herman Hollerith，美国发明家，用穿孔卡片实现了 19 世纪 90 年代的统计表。1896 年他建立了制表机公司，最终于 1924 年成为 IBM 公司
1906	Lee De Forest，美国物理学家，发明了由三个电极真空管组成的三极管。这些管子直到 20 世纪 40 年代才应用到计算机领域
1936	Alan M. Turing，英国逻辑学家，他发表了名为《On Computer Numbers》的论文，论证了有限状态机（自动机）可以完成任意的计算。在“二战”后的英国早期计算机发展过程中，Turing 扮演着举足轻重的角色
1937	George Stibitz，是贝尔电话实验室的物理学家。他用继电器构造二进制电路，可以进行加减乘除运算
1938	Konrad Zuse，德国工程师，建立了第一台二进制计算器 Z1。1941 年，他完成了通用的电子机械计算机 Z3
1938	Claude Shannon，在其美国麻省理工大学硕士论文的基础上出版了《A Symbolic Analysis of Relay and Switching Circuits》一书，书中展示了符号逻辑和二进制运算可以运用于继电器的电路设计
1942	John V. Atanasoff，美国爱荷华州大学教授，完成了简单的电子计算器
1943	IBM-Harvard Mark I，大型电子机械计算器成功运行
1944-1945	J. Presper Eckert 和 John W. Mauchly 在宾夕法尼亚大学摩尔电子工程学院设计并制造了 ENIAC，这是第一台全功能电子计算器

续表

日 期	事 件
1946	John Von Neumann, 他曾是 ENIAC 项目顾问, 并且写了一篇对接下来的 EDVAC 项目较有影响的报告, 在美国普林斯顿大学的高级研究所开始自己的计算机研究工作
1947	Walter Brattain、John Bardeen 和 Willam Schockley 在贝尔实验室发明了晶体管
1948	在英国, 第一个存储程序在 Manchester Mark I 电子计算机上运行
1951	第一台商业化计算机 Ferranti Mark I 和 UNIVAC 交付
1953	IMB 的第一台电子计算机 701 交付
1958	Jack Kilby, 美国得州仪器的工程师, 建立了移相振荡器, 这是第一个集成电路芯片 (IC)
1959	Robert Noyce——Fairchild Semiconductor 创始人之一——生产出第一个平面工艺集成电路。它使得大规模集成电路的生产成为可能。1968 年, Noyce 创办 Intel
1963	美国数字设备公司 (Digital Equipment Corporation) 引进第一台微型计算机
1964	IBM 引进 system/360 为大型计算机的一员
1965	Gordon Moore 在《Electronics》杂志中预测到: 每年的集成芯片中元件数目都会翻倍, 即著名的摩尔定律。1975 年他将摩尔定律更正为集成度每两年翻一番
1969	IBM 研发出第一个片上可编程逻辑阵列 PLA
1971	Marcian E. Hoff, Intel 工程师, 发明了第一款微型处理器
1975	Intersil, 发明第一款可编程逻辑阵列 (FPLA)
1978	Monolithic Memories, 发明可编程阵列逻辑 (PAL) 设备
1981	IBM 发明个人计算机; VHDL 的发展由美国国防部开始。VHDL 中的 V 代表 VHSIC (较高速集成电路), HDL 代表硬件描述语言
1983	Intermetrics、IBM 和 Texas Instrument 获得研发 VHDL 的合同
1984	Xilinx 公司成立并设计销售 FPGAs。 Gateway Design Automation 公司引入硬件描述语言 Verilog
1987	VHDL 成为 IEEE 标准 (IEEE 1076)
1990	Cadence Design System 采用 Verilog
1995	Verilog 成为 IEEE 标准

算盘作为辅助运算设备已被使用了几千年, 但这需要人们实实在在地拨动算珠完成运算。用机器进行计算的想法源于 17 世纪早期。其中, 最著名的 Pascaline 是布莱士·帕斯卡 (Blaise Pascal, 1623—1662) 在 19 岁时为帮助身为上诺曼底税务专员的父亲计算税收而设计的。Pascal 设计了 5 位和 8 位版本的 Pascaline, 将拨号盘、轮轴、齿轮和数字相关联。这一计算器可做加法运算, 也可以通过补码转换完成减法运算, 还可以通过重复的加减运算进行乘除运算。然而, 由于 Pascaline 存在机械故障频出、计算能力有限的问题, 最终还是沦为商业上的失败品。

查尔斯·巴贝奇 (Charles Babbage, 1792—1871) 被认为是计算机之父, 他于 1822 年建造了差分机的工作模型, 可以进行 6 位数字计算并完成数表编制。此后, 查尔斯·巴贝奇做了详细的计划, 准备设计建造更强大的、全尺度的差分机, 并可在金属板上打印出数学表格。计划中机器的长和高均为 10 英尺, 宽为 5 英尺, 由蒸汽驱动。此想法获得英国政府的资助, 巴贝奇和他的首席机械工程师约瑟夫·克莱门特 (Joseph Clement) 在接下来的

12年中努力地建造差分机,在此期间出现了诸多的技术和个人问题(当时的机器工具无法满足巴贝奇的精度要求,妻子去世,与克莱门特之间的严重分歧)。1834年,巴贝奇设想出一种更强大的分析机,可以解决任何数学计算问题,而不仅仅是基于差分的方法。与分析机相比,差分机显得相形见绌,因此他转而专注于分析机的开发。1842年政府最终因不满而取消了他的项目。在生命余下的日子里,尽管巴贝奇知道有生之年无法完成分析机的设计,但他还是继续工作。他记录了上千页的关于分析机的笔记,其中包含了数以百计的轴轮,成千上万的齿轮之类的记录。笔记中有许多现代计算机的元素,包括存储(Memory)、引擎(CPU)和穿孔卡片。其中,穿孔卡片被用于机器的外部编程。

穿孔卡片在编程方面的运用可追溯到1801年Joseph-Marie Jacquard发明的自动换梭织机。在1880年,Herman Hollerith成为美国人口普查特别专员,当年的人口普查数据要花费几年的时间制表。1882年Hollerith成为麻省理工一名机械工程讲师,他发明了一种电子机械系统,可以计算并整理穿孔卡片上记载的统计数据。Hollerith设计的制表机被用在1890年的人口普查中,并在六周之内就完成了数据制表工作。1896年,Hollerith成立了制表机公司,也就是之后的IBM。

接下来,在计算机的发展过程中的主要推动力是“二战”期间在宾夕法尼亚大学电子工程摩尔电子工程学院的ENIAC的开发(电子数值集成和计算)。J.Presper Eckert和John W.Mauchly是这个大型电子计算机主要的设计人。1944年,因为陆军兵工署的合同,他们开始从事之后的计算机开发工作——EDVAC(电子离散变量计算机),这是第一台存储编程计算机。然而,专利权的分歧导致Eckert和Mauchly于1946年从摩尔电子工程学院辞职并开始他们自己的公司——电子控制公司,旨在生产万能的自动计算机——UNIVAC。1948年,由于经济问题迫使他们重组为Eckert—Mauchly计算机公司,并最终在1950年出售给Remington Rand。第一台UNIVAC是在1951年诞生的。

IBM行动迅速并于1953年生产出自己的第一台计算机,此后的一切众所周知,不必多说。在贝尔实验室里发明的晶体管对数字逻辑和计算机产生了重大的影响。巨大并极其耗能的真空管的工作原理,即用电子来控制半导体中电流和电压大小的原理,在后来也彻底并长远地改变了电子行业的发展。固态技术的发展引发了20世纪60年代集成电路的诞生、70年代微处理器和80年代可编程逻辑设备的问世。

1984年,Xilinx公司成立。Xilinx不是第一个生产可编程逻辑设备的公司,但是它代表了当时在集成电路技术飞速发展的背景下应运而生的数百家小公司。Xilinx生产的Spartan 3E现场可编程门阵列(FPGA),我们将在书中的实验例程中讲述并运用。

表1.1中一个重要的记录是1965年Goordon Moore预测集成电路中晶体管的数量会每年翻一番,在1975年时达65000个,如图1.1所示。1975年之后他修改了他的“定律”并预测晶体管集成数量会以每两年翻一番的速度增长。回望过来的这些年,应该说摩尔定律还是相当准确的,如图1.2所示。

微处理器的发展彻底改变了我们的生活方式。今天,从电话到汽车,微处理器几乎被用于我们所使用的各种产品中。通用微处理器通过执行存储在内存中的指令,实现着特定的算法。随着FPGA的发展,很多算法可以不用写入存储设备而直接由硬件实现,这样可以使算法的执行速度大幅提高。本书的第9章和第10章将会介绍怎样直接在硬件中实现算法。

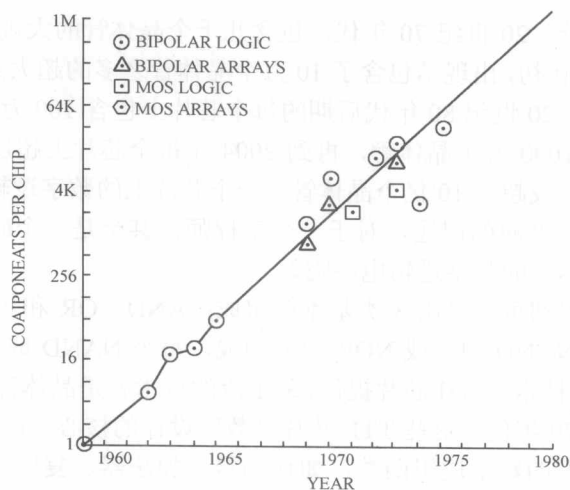


图 1.1 1965 年的原始版摩尔定律

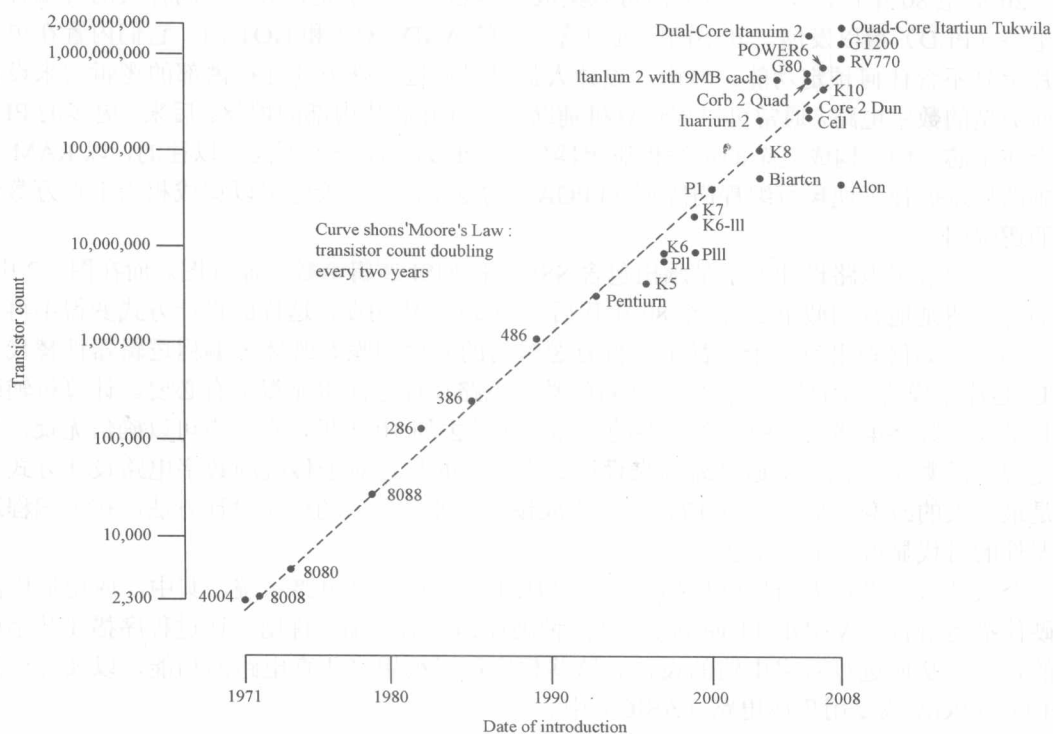


图 1.2 1975 年的更正版摩尔定律

1.2 数字逻辑

20 世纪 60 年代，第一个集成电路芯片问世，该芯片上集成了不到 100 个晶体管，被称为小规模集成电路（SSI）。中规模集成电路（MSI）在 20 世纪 60 年代得到发展，一个芯片

上集成了几百个晶体管。20 世纪 70 年代, 包含几千个晶体管的大规模集成电路 (LSI) 得以发展。20 世纪 80 年代初, 出现了包含了 10 万个晶体管之多的超大规模集成电路 (VLSI)。

如图 1.2 所示, 从 20 世纪 80 年代后期的每个芯片上包含 100 万个晶体管, 到 90 年代中期每个芯片上包含 1000 万个晶体管, 再到 2004 年每个芯片上超过 1 亿个晶体管, 今天一个芯片上已经可以集成超过 10 亿个晶体管。一个芯片上的数字逻辑成倍地增长对数字逻辑设计人员来说是一个严峻的问题。对于一个工程师, 甚至是一个研发团队来说, 如何设计一个包含上百万晶体管的数字逻辑电路呢?

任何数字逻辑电路都可以只由 3 类基本门组成: AND、OR 和 NOT。事实上, 任何数字逻辑电路都可以由 NAND 门 (或 NOR 门) 构成, 每个 NAND 或 NOR 门包含 4 个晶体管。这些基础门由多种技术的 SSI 芯片提供, 最主流的技术就是晶体管—晶体管逻辑 (TTL)。从 20 世纪 60 年代到 70 年代, 这些 TTL 芯片是数字设计的核心。而多种 MSI TTL 芯片又可以用于实现不同类型的数字逻辑函数, 如译码器、加法器、复用器、比较器及其他逻辑函数。

20 世纪 80 年代, 成千上万的门可以集成在单芯片上。于是, 很多不同种类的可编程逻辑电路 (PLD) 得以发展。其中的阵列包含大量的 AND、OR 和 NOT 门, 它们内置在单个芯片上且不含任何预定功能。然而, 设计人员可以通过特殊方式连接内部的逻辑门来设计任何类型的数字电路, 通常可以用计算机辅助工具打开芯片内部的熔丝。后来, 更多的 PLD 整合在单芯片中, 构成复杂的可编程逻辑器件 (CPLD)。而一个不同于以往的, 以 RAM 为基础的技术促使了现场可编程门阵列 (FPGA) 的诞生, 在今天它可以集成相当于百万数量级的逻辑门。

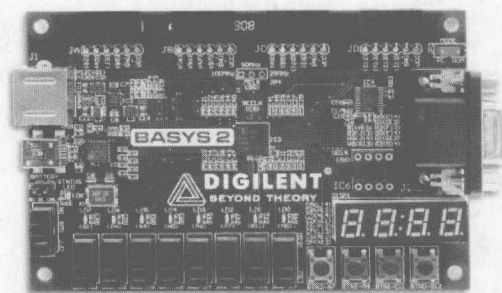
传统的数字电路设计方法是画出包含 SSI 门和 MSI 逻辑函数的原理图。而在图 1.2 中, 我们可以清楚地看到截至 20 世纪 80 年代后期和 90 年代初期, 这样的设计方式变得不再可行。那么, 如何画出包含十万甚至上百万逻辑门的原理图呢? 既然可编程逻辑器件替代了 TTL 芯片, 成为新的设计主流, 那么新的数字电路设计方法将显得很有必要。计算机辅助工具是今天数字电路设计中至关重要的一环。在过去的 10 年里, 有一点可以确信无疑, 那就是现在的数字工程师是通过编程来设计数字电路的! 这对于传统的数字电路设计方式来说是最重大的改变。很多在用 TTL 芯片完成设计时非常重要的传统设计方法, 在可编程逻辑器件的时代显得不那么重要。

今天的数字设计人员用硬件描述语言 (HDL) 设计数字电路系统。其中, 应用最广泛的硬件描述语言是 VHDL 和 Verilog。这两种硬件描述语言都允许用户通过程序描述数字电路的行为, 从而进行数字电路的设计。这些程序可以被用来仿真电路的功能, 以及综合到 CPLD、FPGA 或专用集成电路 (ASIC) 中。

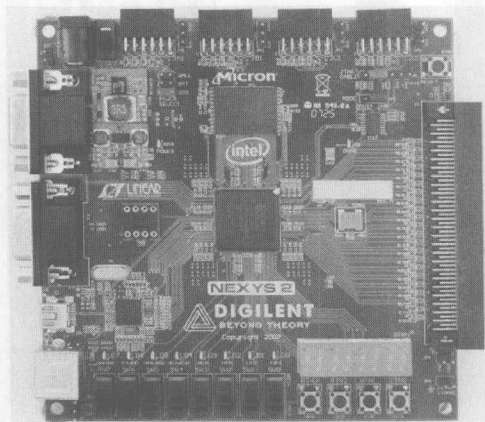
为了让大家更容易地掌握 Verilog 知识, 本书在介绍基础数字电路设计的过程中列举了很多 Verilog 的例程, 并假设读者没有任何的数字逻辑基础知识。

我们将用 Xilinx ISE 来仿真和综合我们的 Verilog 代码。Verilog 程序可以用任何 Verilog 仿真器仿真。在附录 A 中, 我们提供了一个简明的 Xilinx ISE 软件的使用向导。你可以在 Xilinx 网站上下载免费版的 ISE™ WebPACK™。在程序实现过程中, 产生的 “*.bit” 文件可以下载到 BASYS2、NEXYS2 (如图 1.3 所示) 和类似功能的 FPGA 开发板上。Digilent 公司的 BASYS2 开发板包含一块 10 万门的 Xilinx Spartan 3E FPGA (同时提供 25 万门的版本)、

8 个拨位开关、4 个按钮开关、8 个 LED 和 7 段数码管。开发板上时钟频率可采用跳线的方式将其设置为 25MHz、50MHz 或 100MHz。同时，它还提供了外部电路的接口，包含一个 VGA 口和一个 PS/2 口。NEXYS2 开发板和 BASYS2 有些类似，但是其功能更为强大，它包含一块 50 万门或 120 万门的 Spartan 3E FPGA、一个 Hirose FX2 扩展口、16MB 的快速 PSDRAM、16MB 的 Flash、50MHz 时钟和为外部时钟而设的管座。这使得 NEXYS2 非常适用于嵌入式处理。



(a) BASYS2 开发板



(b) NEXYS2 开发板

图 1.3 开发板实物图

本书中所有的 Verilog 程序都可以用 BASYS2、NEXYS2 和类似功能的 FPGA 开发板来实现。唯一的区别在于，不同的开发板需要选用不同的约束文件，文件“nexys2.ucf”约束 NEXYS2 开发板中的引脚，文件“basys2.ucf”约束 BASYS2 开发板中的引脚。相关的配套资料可以到译者序中提供的网址下载。表 1.2 列出了 BASYS2 开发板和 NEXYS2 开发板的跳线设置。

表 1.2 开发板跳线设置

BASYS2 开发板	NEXYS2 开发板
JP3 跳线设置为 JTAG	电源选项跳线设置为 USB
移除 JP4 跳线选择 50MHz 时钟	模式跳线设置为 JTAG

1.3 Verilog

Verilog 基于 C 语言但又不是 C 语言，它是硬件描述语言，用来对数字逻辑电路进行建模。它和 C 语言的语法非常相似，但语言本身的行为却大不相同。在本书中，你可以通过学习其中的例程，以及完成每章之后的习题来学习 Verilog。和任何程序语言一样，你只有通过自己编写程序，然后仿真你的设计并观察输出波形才能学会 Verilog。除此之外，通过这样的方法，你还能学会数字电路的设计方法。

2

Chapter 02

基本逻辑门



在本章你将学习：

- ◎ 以真值表和逻辑表达式的方式定义基本逻辑
- ◎ 迪摩根定律
- ◎ 正逻辑门和负逻辑门有什么样的关联
- ◎ 怎样用真值表写出基于乘积项之和或累加项之积的逻辑表达式