

数字系统设计实验教程

(第二版)

屈民军 唐 奕 马洪庆 主编



科学出版社

数字系统设计实验教程

(第二版)

屈民军 唐 奕 马洪庆 主编



科学出版社

北 京

内 容 简 介

本书是电子类专业核心课程的教材之一,由多年从事数字电子技术理论和实验教学的教师合作完成。书中以 Xilinx 公司的 Vivado FPGA 设计套件为基础,硬件平台以 Xilinx 的 Nexys Video Artix-7 FPGA 多媒体音视频智能互联系统为主,并辅以 Basys 3 FPGA 口袋开发板;软件平台采用 ModelSim、Vivado 等专用开发工具,循序渐进地介绍数字系统设计的原理和方法。

本书内容包括数字系统设计方法介绍、Verilog HDL 介绍、仿真及测试代码的编写、实验软/硬件平台的使用、数字系统基础实验、数字系统综合实验、MIPS 微处理器的设计和应用等 7 章。作者本着培养学生综合设计和创新能力的原则精心设计了 29 个实验项目,以设计性、综合性和探索性实验为主,强调多学科知识的交叉应用,以达到培养和提高学生实验的综合技能的目的。

本书可作为高等院校电子信息类、通信类、自动化类、计算机类、测控技术与仪器等专业的教材,也可供其他从事电子技术工作的技术人员参考。

图书在版编目(CIP)数据

数字系统设计实验教程/屈民军,唐奕,马洪庆主编.—2 版.—北京:科学出版社,2018.6

ISBN 978-7-03-057749-8

I. ①数… II. ①屈… ②唐… ③马… III. ①数字系统—系统设计—实验—教材 IV. ①TP271-33

中国版本图书馆 CIP 数据核字(2018)第 123557 号

责任编辑:刘 博/责任校对:郭瑞芝

责任印制:吴兆东/封面设计:迷底书装

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>

北京九州迅驰传媒文化有限公司 印刷

科学出版社发行 各地新华书店经销

*

2018 年 6 月第 一 版 开本:787×1092 1/16

2018 年 6 月第一次印刷 印张:21 1/4

字数:490 000

定价:69.00 元

(如有印装质量问题,我社负责调换)

前 言

随着集成电路深亚微米工艺技术的发展,EDA 技术和 FPGA 器件获得了长足的发展,七年前出版的《数字系统设计实验教程》在内容方面就显得有些落后了。为了顺应技术潮流,必须对实验的软/硬件平台进行升级,因此对原书的修订就显得势在必行。与此同时,作者经过这七年的实验教学,积累了大量的实验教学心得,获得了新的教学思路,并累积了大量教学素材,为原书的修订打下坚实的基础。这次教程修订的主要思想体现在以下几个方面。

(1) 实验设置更加重视由浅入深,循序渐进,并注重实验的趣味性,便于学生接受的同时激发学生学习兴趣,提高实验教学质量。另外,在实验内容设置上,更加强调设计性、综合性、探索性和多学科知识交叉应用。

(2) 优化了“自顶而下”的数字系统层次化结构设计方法,使层次结构和模块划分更为合理。

(3) 第二版强调数字系统的仿真方法和技巧,使学生不在实验室就可完成实验的大部分内容。

(4) 引进“口袋实验室”,第二版中有相当一部分实验可在 Basys 3 口袋实验开发板进行,因此使每位学生都可以拥有一套低成本的 FPGA 开发板卡,可随时随地进行实验。

鉴于上述教程的修订思想,在第二版修订过程中,各章节工作主要体现在以下几方面。

将原书的第 1、2 章精简为第 1 章,主要介绍 FPGA 器件原理和数字系统设计方法。

第 2 章“Verilog HDL”为新增加内容,使教材内容更加完整。本章系统介绍有关硬件描述语言 Verilog HDL 的语法知识和用 Verilog HDL 描述数字系统设计方法。

第二版更加强调仿真的重要性,增加了部分仿真难点的设计和仿真技巧的介绍。例如,在第 3 章中增加 PS2 鼠标的仿真测试实例,介绍双向端口的仿真。又如,在实验 15 介绍正弦波等模拟信号序列的观察方法。

第 4 章主要介绍软硬件平台使用,由于实验平台均已升级到 2017 年的最新版本,因此,与第一版相比,本章可视为全新的章节。

第二版更加强调循序渐进的教学模式,因此第 5 章增加常用组合模块设计、常用时序模块设计两个数字电路级别的实验,并大幅改写浮点加法器设计实验,目的是强调系统层次化结构设计方法,易于学生从数字电路设计转为数字系统设计。另外,本章还增加学号滚动实验以提高学生的实验兴趣。

第 6 章增加若干音频接口处理实验和游戏实验,更强调多学科知识的交叉应用,更凸显

实验内容的综合性和趣味性。另外，与第一版相比，在许多实验中，作者优化了“自顶向下”的数字系统层次化结构设计方法，使层次结构和模块划分更为合理。

相对第一版，第7章改动较少，只是将CPU设计更新到新的FPGA平台。

另外，请读者注意：本教程配套相关程序代码等电子资料，请登录<http://www.ecsponline.com/>，检索图书名称，在图书详情页“资源下载”栏目中获取。

屈民军、唐奕和马洪庆等参与了本书的修订工作。另外，在本书修订过程中，我们参考了大量书籍、论文以及网络文献，在此向各位作者表示深深的感谢。

由于作者水平有限，书中难免还有不足之处，恳请读者给予批评指正，以便于本书的修订和完善。

编 者

2018年4月于浙江大学

目 录

第 1 章	FPGA 与数字系统设计	1
1.1	现场可编程逻辑器件	1
1.1.1	概述	1
1.1.2	FPGA 发展历史	1
1.1.3	FPGA 芯片的结构	3
1.2	数字系统设计方法	6
1.2.1	数字系统的基本组成	6
1.2.2	数字系统的结构化设计方法	7
1.2.3	数字系统设计实例	8
1.3	基于 FPGA 的数字系统设计流程	12
1.4	基于 FPGA 的数字系统的调试	15
1.4.1	数字系统的调试	15
1.4.2	选择合适的 FPGA 调试方法	18
第 2 章	Verilog HDL	20
2.1	初识 Verilog HDL	20
2.1.1	概述	20
2.1.2	Verilog HDL 的基本结构	20
2.2	Verilog HDL 的基础知识	22
2.2.1	词法	22
2.2.2	常量	24
2.2.3	数据类型和变量	25
2.2.4	参数	26
2.2.5	模块端口类型	27
2.2.6	运算符及优先级	27
2.3	Verilog HDL 的描述语句	31
2.3.1	数据流描述语句	31
2.3.2	行为描述语句	32
2.3.3	Verilog 描述风格及层次化设计	39
2.3.4	编译预处理指令	43
2.4	有限状态机的描述	44
2.4.1	状态机的结构	44
2.4.2	状态机的 Verilog HDL 描述方法	45

2.5 设计举例与技巧	48
2.5.1 常用组合电路的设计	48
2.5.2 常用时序电路的设计	51
2.5.3 数字系统设计实例	54
第3章 testbench 的编写	61
3.1 概述	61
3.2 testbench 的结构形式	61
3.2.1 testbench 的基本结构	61
3.2.2 testbench 结构实例详解	62
3.3 常用的系统任务和系统函数	65
3.4 testbench 的激励和响应	67
3.4.1 testbench 的激励方式	67
3.4.2 仿真结果分析方式	69
3.5 常用激励信号的一些描述形式	69
3.6 testbench 实例	72
3.6.1 组合乘法器实例	72
3.6.2 视频显示接口仿真实例	74
3.6.3 PS2 键盘接口电路实例	77
3.6.4 PS2 鼠标接口电路实例	80
第4章 数字系统实验平台的使用	84
实验1 ModelSim 仿真软件的使用	84
实验2 Vivado 软件的使用	97
实验3 IP 内核的使用与仿真	109
实验4 ILA 的逻辑分析仪实验	123
第5章 数字系统设计的基础实验	130
实验5 常用组合电路模块的设计和应用	130
实验6 浮点数加法器的设计	136
实验7 常用时序电路模块的设计和应用	141
实验8 快速加法器的设计	146
实验9 快速乘法器的设计	150
实验10 学号滚动显示实验	156
实验11 异步输入的同步器和开关防颤动电路的设计	160
第6章 数字系统综合设计实验	166
实验12 数字式秒表	166
实验13 低频数字式相位测量仪的设计	170
实验14 全数字锁相环的设计	176
实验15 直接数字频率合成技术(DDS)的设计与实现	181
实验16 基于FPGA的FIR数字滤波器的设计	186
实验17 数字下变频器(DDC)的设计	191

实验 18 音频编解码芯片接口设计	195
实验 19 音乐播放实验	209
实验 20 基于 FPGA 的实时语音变声系统的设计	221
实验 21 HDMI 显示器接口设计实验	230
实验 22 键盘接口实验	239
实验 23 鼠标接口实验	248
实验 24 文本输入与显示实验	254
实验 25 动态显示实验	260
实验 26 点灯游戏的设计	266
实验 27 推箱子游戏的设计	272
第 7 章 CPU 设计	279
实验 28 多周期 MIPS 微处理器设计	279
实验 29 流水线 MIPS 微处理器设计	299
附录 A Basys3 开发板的使用	314
A.1 FPGA 主芯片介绍	315
A.2 电源电路	315
A.3 时钟电路	315
A.4 基本 I/O 接口	316
A.5 数码管电路	316
A.6 I/O 扩展电路	317
A.7 USB-UART 桥接电路	318
A.8 USB HID Host	318
A.9 VGA 接口	318
A.10 FPGA 调试及配置电路	319
附录 B Nexys Video 开发板的使用	320
B.1 FPGA 主芯片介绍	321
B.2 电源电路	321
B.3 时钟电路	322
B.4 基本 I/O 接口	322
B.5 I/O 扩展电路	323
B.6 音频编解码(CODEC)接口电路	323
B.7 USB-UART 桥接电路	324
B.8 USB HID Host	324
B.9 HDMI 接口	325
B.10 FPGA 调试及配置电路	325
附录 C ASCII 码表	326
附录 D 仿真环境的建立	328
参考文献	332

第 1 章

FPGA 与数字系统设计

1.1 现场可编程逻辑器件

1.1.1 概述

随着集成电路深亚微米工艺技术的发展,可编程逻辑器件(FPGA)及其应用获得了长足的发展,FPGA 器件的单片规模大大扩展,系统运行速度不断提高,功耗不断下降,价格大幅度降低。因此,与传统电路设计方法相比,利用 FPGA/CPLD 进行数字系统的开发具有功能强大、开发过程投资小、周期短、便于修改及开发工具智能化等特点。并且随着电子工艺不断改进,低成本高性能的 FPGA/CPLD 器件推陈出新,促使 FPGA/CPLD 成为当今硬件设计的首选方式之一。熟练掌握 FPGA/CPLD 设计技术已经成为电子设计工程师的基本要求。

电子设计自动化(Electronic Design Automation, EDA)技术是以计算机为工作平台,融合了应用电子技术、计算机技术、智能化技术最新成果而开发出来的一套先进的电子系统设计的软件工具。集成电路设计技术的进步也对 EDA 技术提出了更高的要求,大大地促进了 EDA 技术的发展。以高级语言描述、系统仿真和综合技术为特征的 EDA 技术,代表了当今电子设计技术的最新发展方向。EDA 设计技术的基本流程是设计者按照“自顶而下”的设计方法,对整个系统进行方案设计和功能划分。电子系统的关键电路一般用一片或几片专用集成电路(ASIC)实现,采用硬件描述语言(HDL)完成系统行为级设计,最后通过综合器和适配器生成最终的目标器件。这种被称为高层次的电子设计方法,不仅极大地提高了系统的设计效率,而且使设计者摆脱了大量的辅助性工作,将精力集中于创造性的方案与概念的构思上。近年来的 EDA 技术主要有以下特点:

- (1)采用行为级综合工具,设计层次由 RTL 级上升到了系统级;
- (2)采用硬件描述语言描述大规模系统,使数字系统的描述进入抽象层次;
- (3)采用布局规划(floor planning)技术,即在布局布线前对设计进行平面规划,使得复杂 IC 的描述规范化,做到在逻辑综合早期设计阶段就考虑到物理设计的影响。

从某种意义上讲,FPGA 和 EDA 技术的发展,将会进一步引起数字系统设计思想和方法的革命。正是在这样的技术发展背景下,为了配合数字系统设计课程教学,本书主要讨论基于 FPGA 器件来实现数字系统。

1.1.2 FPGA 发展历史

Xilinx 公司于 1984 年发明了世界首款 FPGA,在接下来的 30 多年里,在应用需求和工艺技术发展的驱动下,FPGA 的器件得到了迅速发展,容量提升了一万多倍,速度提升了 100 多倍,每单位功能的成本和能耗降低到原来的万分之一。FPGA 经历了如下几个阶段。

1. 发明阶段(1984~1992 年)

首款 FPGA, 即 Xilinx 公司基于 SRAM 技术可重复编程的 XC2064, 只包含 64 个逻辑模块, 每个模块含有两个 3 输入查找表(LUT)和一个寄存器。尽管容量很小, XC2064 晶片的尺寸却非常大, 因此价格昂贵。

在成本压力下, FPGA 架构师寻求通过架构和工艺创新来尽可能地提高 FPGA 的设计效率。为提高效率, 架构经历了从复杂的 LUT 结构到 NAND 门再到单个晶体管的演变。另外, 各厂商也在探索不同新工艺, Actel 公司以牺牲可重复编程能力为代价, 采用反熔丝工艺, 在 1990 年推出当时最大容量的 FPGA (Actel 1280)。

在这一阶段, 尚无自动布局布线。另外, 完全不同的 FPGA 架构排除了通用设计工具的可能。因此 FPGA 厂商就担负起了为各自器件开发电子设计自动化(EDA)的任务。

2. 扩展阶段(1992~1999 年)

FPGA 初创公司都是无晶圆厂的公司, 在当时属于新鲜事物。由于没有晶圆厂, 他们在 20 世纪 90 年代初期通常无法获得领先的芯片技术。到 20 世纪 90 年代后期, IC 代工厂意识到 FPGA 是推动工艺发展的理想因素, 只要能新工艺产出晶体管和导线, 就能制造基于 SRAM 的 FPGA。因此, 每一代新工艺的出现都会将晶体管数量增加一倍, FPGA 成本减半, 并将 FPGA 的尺寸增加一倍。

在这一阶段, 基于 SRAM 工艺的 FPGA 体现了明显的产品优势, 因为它们可以率先采用每种新工艺。而反熔丝在新节点上的验证工作则需要额外数月甚至数年时间。基于反熔丝的 FPGA 丧失了竞争优势。为获得上市速度和成本优势, 架构创新与工艺改进相比就要退居其次。

在这一阶段, EDA 有了长足发展, 到 20 世纪 90 年代末, EDA 能自动完成综合、布局和布线等设计流程。

3. 积累阶段(2000~2007 年)

2000 年以后, FPGA 已成为数字系统中的通用组件。容量和设计尺寸快速增加, FPGA 在数据通信领域开辟了巨大市场。21 世纪初期互联网泡沫破灭之后, 迫切需要降低成本, 这也减少了很多“临时”ASIC 用户。定制芯片对小的研发团队来说风险太大。当他们发现 FPGA 可以解决他们的问题时, 自然就变成了 FPGA 用户。

FPGA 不局限于典型问题, 单纯提高容量不足以保证市场增长。FPGA 厂商通过如下两种方式解决了这一挑战。针对低端市场, 厂商再度关注效率问题, 并生产低容量、低性能、低成本的 FPGA 系列, 例如, Xilinx Spartan 系列 FPGA。针对高端市场, FPGA 厂商通过开发各种重要功能的 IP 内核, 努力让客户更方便地使用 FPGA。这些 IP 内核中最值得注意的是存储器、控制器、各种通信协议模块(包括以太网 MAC), 甚至微处理器等。

设计特点在 2000 年以后发生了改变。大型 FPGA 容纳超大型设计(完整子系统)。FPGA 用户不再只是实现逻辑; 他们需要使 FPGA 设计符合系统标准要求。积累阶段末期, FPGA 已不仅是门阵列, 而且还是集成有可编程逻辑的复杂功能集。FPGA 俨然变成了一个系统。

4. 系统阶段(2008 年以后)

这个阶段出现了片上可编程系统(SoPC 技术)和系统级芯片(SoC 技术),是 FPGA 和 ASIC 技术融合的结果。FPGA 越来越多地整合系统模块:高速数据收发器、存储器、DSP 处理单元和嵌入式处理器内核等。同时还进一步集成了重要控制功能:比特流加密与验证、混合信号处理、电源与温度监控以及电源管理等。

在系统阶段,器件发展同时也推动了 EDA 开发工具的发展。系统 FPGA 需要高效的系统编程语言,目前可利用 OpenC、System Verilog 和 C 语言等软件来开发 FPGA 系统。

目前,基于 FPGA 的片上可编程的概念仍在进一步向前发展,如动态可重构 FPGA 技术等。

1.1.3 FPGA 芯片的结构

如前所述,FPGA 是在 PAL、GAL、CPLD 等可编程器件的基础上进一步发展的产物。它是作为 ASIC 领域中的一种半定制电路而出现的,既解决了定制电路的不足,又克服了原有可编程器件门电路的缺点。

目前主流的 FPGA 仍是基于查找表(LUT)技术,并且整合了常用功能(如 DRAM、时钟管理和 DSP 等)的硬核模块,其内部结构示意图如图 1.1 所示。FPGA 芯片主要由以下 6 部分组成。

1. 基本可编程逻辑单元(CLB)

CLB 是 FPGA 内的基本逻辑单元,CLB 的实际数量和特性依器件的不同而不同。在 Xilinx 公司的 FPGA 器件中,CLB 是基于查找表结构的,每个 CLB 由多个(一般为 4 个或 2 个)相同的 Slice 和附加逻辑构成,如图 1.2 所示。

Slice 是 Xilinx 公司定义的基本逻辑单元,其内部结构如图 1.3 所示,一个 Slice 由两个 4 输入的函数发生器、进位逻辑、算术逻辑、存储逻辑和函数复用器组成。

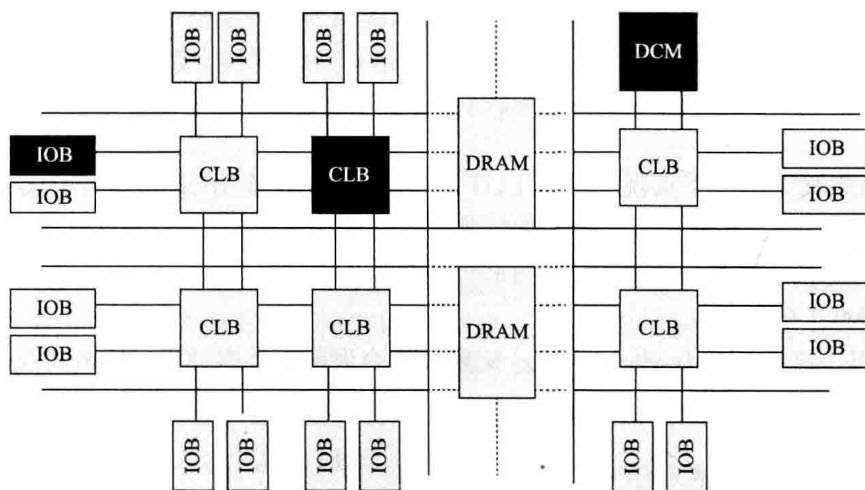


图 1.1 FPGA 芯片的内部结构

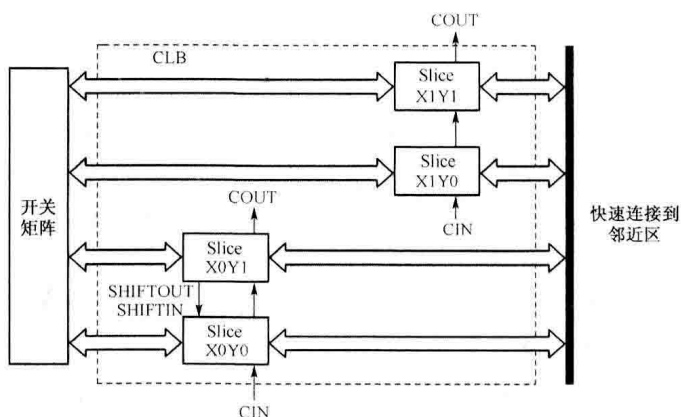


图 1.2 典型的 CLB 结构示意图

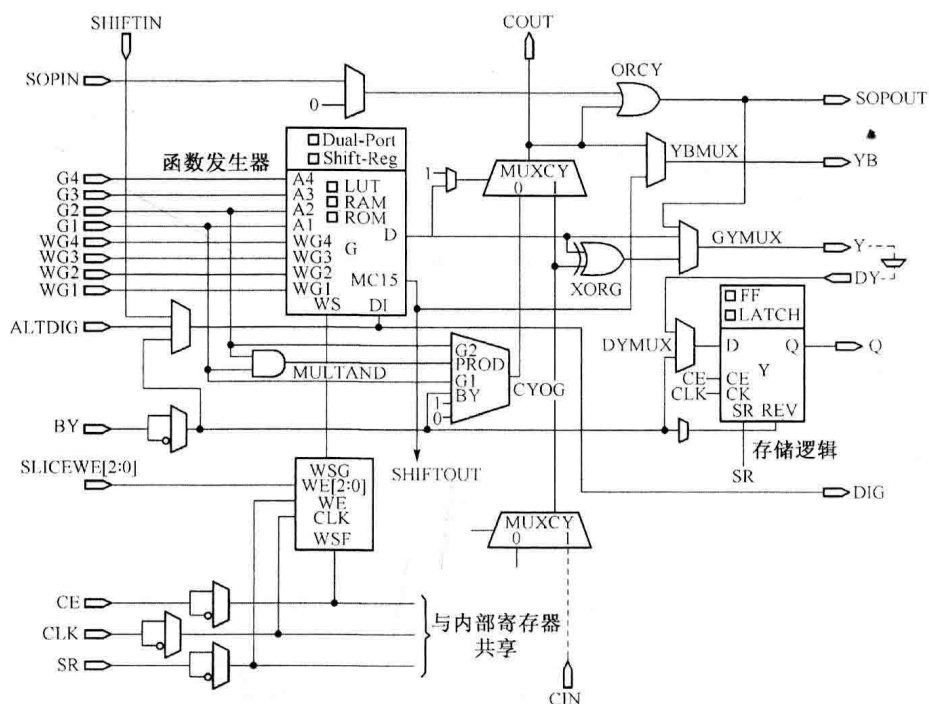


图 1.3 典型的 4 输入 Slice 结构示意图

4 输入函数发生器用于实现 4 输入 LUT、分布式 RAM 或 16bit 移位寄存器；存储逻辑可配置为 D 触发器或锁存器；进位逻辑由专用进位信号和函数复用器 (MUXCY) 组成，用于实现快速的算术加减法操作；算术逻辑包括一个异或门 (XORG) 和一个用于提高乘法运算效率的专用与门 (MULTAND)。

CLB 可以由开关矩阵 (switch matrix) 配置成组合逻辑、时序逻辑、分布式 RAM 和分布式 ROM。

2. 可编程输入/输出单元 (IOB)

可编程输入/输出单元简称 I/O 单元，是芯片与外界电路的接口部分，完成不同电气特性下对输入/输出信号的驱动与匹配要求，其结构示意图如图 1.4 所示。

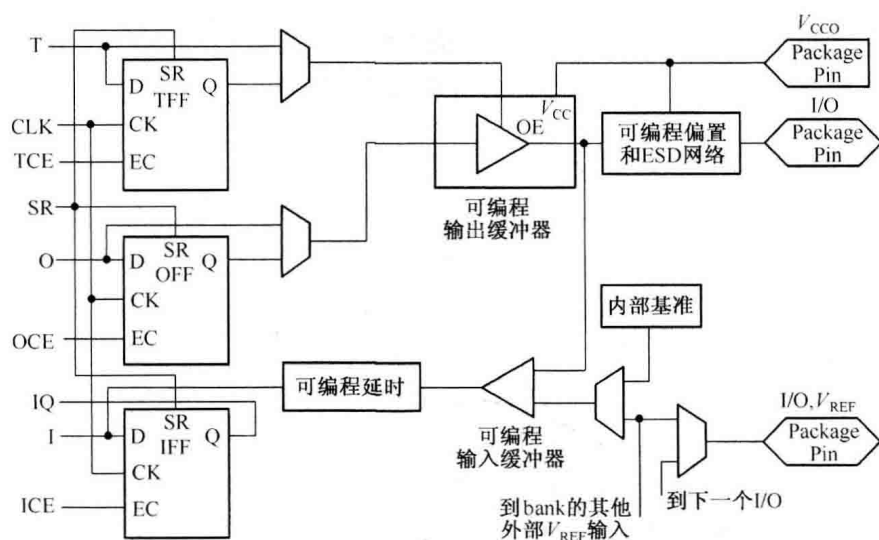


图 1.4 典型的 IOB 内部结构示意图

FPGA 内的 I/O 按组分类,每组都能够独立地支持不同的 I/O 标准。通过软件的灵活配置,可适配不同的电气标准与 I/O 物理特性,可以调整驱动电流的大小,可以改变上拉、下拉电阻。目前, I/O 口的频率也越来越高,一些高端的 FPGA 通过 DDR 寄存器技术可以支持高达 2Gbit/s 的数据速率。

外部输入信号可以通过 IOB 模块的存储单元输入到 FPGA 的内部,也可以直接输入至 FPGA 内部。当外部输入信号经过 IOB 模块的存储单元输入到 FPGA 内部时,其保持时间(hold time)的要求可以降低,通常默认为 0。

为了便于管理和适应多种电气标准, FPGA 的 IOB 被划分为若干个组(bank),每个 bank 的接口标准由其接口电压 V_{CCO} 决定,一个 bank 只能有一种 V_{CCO} ,但不同 bank 的 V_{CCO} 可以不同。只有相同电气标准的端口才能连接在一起, V_{CCO} 电压相同是接口标准的基本条件。

3. 数字时钟管理模块(DCM)

业内大多数 FPGA 均提供数字时钟管理模块(Digital Clock Manager, DCM), Xilinx 公司最先进的 FPGA 提供了数字时钟管理和相位环路锁定。DCM 可实现时钟频率合成、时钟相位调整和消除时钟信号畸变三大功能。

4. 嵌入式块式存储器(DRAM)

大多数 FPGA 都具有内嵌的块 RAM,这大大拓展了 FPGA 的应用范围和灵活性。块 RAM 可被配置为单端口 RAM、双端口 RAM、内容地址存储器(CAM)和 FIFO 等常用存储结构。RAM、FIFO 是比较普及的概念,不再赘述。CAM 在其内部的每个存储单元中都有一个比较逻辑,写入 CAM 的数据会和内部的每一个数据进行比较,并返回与端口数据相同的所有数据的地址,因而在路由的地址交换器中有广泛的应用。除了块 RAM,还可以将 FPGA 中的 LUT 灵活地配置成 RAM、ROM 和 FIFO 等结构。

单片块 RAM 的容量为 18Kbit,即位宽为 18 位、深度为 1024。可以根据需要改变其位宽和深度,但要满足两个原则:首先,修改后的容量(位宽 $\times$ 深度)不能大于 18Kbit;其次,

位宽最大不能超过 36 位。当然, 可以将多片块 RAM 级联起来形成更大的 RAM, 此时只受限于芯片内块 RAM 的数量, 而不再受上面两条原则的约束。

5. 丰富的布线资源

布线资源连通 FPGA 内部的所有单元, 而连线的长度与工艺决定着信号在连线上的驱动能力和传输速度。FPGA 芯片内部有着丰富的布线资源, 根据工艺、长度、宽度和分布位置的不同而划分为 4 个类别。第一类是全局布线资源, 用于芯片内部全局时钟和全局复位/置位信号的布线; 第二类是长线资源, 用于完成芯片 bank 间的高速信号和第一全局时钟信号的布线; 第三类是短线资源, 用于完成基本逻辑单元之间的逻辑互连和布线; 第四类是分布式的布线资源, 用于专有时钟、复位等控制信号线。

在实际中, 设计者不需要直接选择布线资源, 布局布线器可自动根据输入逻辑网表的拓扑结构和约束条件选择布线资源来连通各个模块单元。从本质上讲, 布线资源的使用方法和设计的结果有密切、直接的关系。

6. 底层内嵌功能单元和内嵌专用硬核

内嵌功能模块主要指延迟锁相环(Delay Locked Loop, DLL)、锁相环(Phase Locked Loop, PLL)、数字信号处理(Digital Signal Processing, DSP)和中央处理器(Central Processing Unit)。现在越来越丰富的内嵌功能单元使得单片 FPGA 成为系统级的设计工具, 并具备了软硬件联合设计的能力, 逐步向 SoC 平台过渡。

内嵌专用硬核是相对底层嵌入的软核而言的, 指 FPGA 处理能力强大的硬核(hard core), 等效于 ASIC 电路。为了提高 FPGA 性能, 芯片生产商在芯片内部集成了一些专用的硬核。例如, 为了提高 FPGA 的乘法速度, 主流的 FPGA 中都集成了专用乘法器; 为了适用通信总线与接口标准, 很多高端的 FPGA 内部都集成了串并收发器(SERDES), 可以达到数十 Gbit/s 的收发速度。

Xilinx 公司的高端产品不仅集成了 Power PC 系列 CPU, 还内嵌了 DSP Core 模块, 其相应的系统级设计工具是 EDK 和 Platform Studio, 并依此提出了片上系统(System on Chip, SoC)的概念。通过 PowerPC、MicroBlaze、PicoBlaze 等平台, 能够开发标准的 DSP 处理器及其相关应用, 达到 SoC 的开发目的。

1.2 数字系统设计方法

1.2.1 数字系统的基本组成

数字系统的结构框图如图 1.5 所示, 数字系统一般可划分为数据通道子系统和控制器子系统两大部分。数据通道子系统主要完成数据采集、存储、运算处理和数据传输等功能; 控制器单元是执行算法的核心电路, 根据外部控制输入和数据通道单元的响应信号, 依照设计方案中的既定算法, 按序控制系统内各模块进行工作。

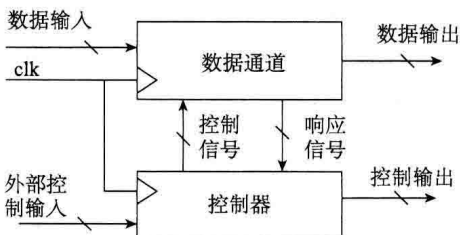


图 1.5 数字系统的结构框图

数据通道通常为设计者所熟悉的各种功能电路, 无论是取用现成模块或自行设计一般不会花很大精

力。因此,系统设计的主要任务是设计控制器。控制器通常为一个有限状态机电路。

1.2.2 数字系统的结构化设计方法

1. 自顶而下的设计方法

从理论上讲,任何数字系统都可以看成一个复杂的时序系统。自顶而下的设计方法是设计者从整个系统逻辑功能出发,先进行最顶层的系统设计,然后按照一定的原则将系统划分成若干子系统,逐级向下,再将每个子系统划分为若干功能模块,模块还可以继续划分为子模块,直至分成最基本模块实现。在自顶而下的划分过程中,最重要的是将系统或子系统划分为控制器单元和若干个数据通道单元。

自顶而下的数字系统设计方法可分为以下几个步骤。

(1)对设计任务进行分析,根据设计任务把所要设计的系统合理地划分成若干子系统,使其分别完成较小的任务。

(2)将系统或子系统划分为控制器单元和若干个数据通道单元。

(3)设计控制器,以控制和协调各数据通道的工作。

(4)对各子系统功能部件进行逻辑设计。

(5)对于复杂的数字系统,还要对各子系统间的连接关系及数据流的传送方式进行设计。

对整个设计要求和任务的良好理解是设计任务能否很好完成的关键。只有仔细分析和明确了总体设计任务后,才有可能合理地进行子系统的划分工作。系统的划分过程,实际上是把总体任务划分成若干个分任务的过程。这项工作完成的好坏可由下列原则进行初步衡量。

(1)对所要解决的总体任务是否已全部清楚地描述出来?

(2)是否有更清楚、更简单的描述可以概括所要解决的问题?

(3)各子系统所承担的分任务是否清楚、明确?是否有更清楚的划分方式?

(4)各子系统之间的相互关系是否明确?它们之间的控制关系是怎样的?

(5)控制部分与被控制部分是否清楚、明确?它们之间的控制关系是怎样的?

2. 用 ASM 图设计控制器

控制器的设计是数字系统硬件设计的中心环节。控制器本身也是一个子系统,它的作用是解释所接收到的各个输入信号,根据输入信号和预定的算法流程图程序控制整个系统按指定的方式工作。

从本质上讲,由硬件直接实现的控制器设计与一般时序电路并无区别,仅仅由于设计着眼点不同,控制器的设计有其独特性。控制器设计的主要特点是不必过分追求状态最简,触发器的数量也不必一味地追求最少。主要理由是控制器的成本只占总成本中很小一部分,但控制器的性能对整个系统的工作有举足轻重的影响。有时,在控制时序中增加一些多余状态,往往会使数字系统工作更加直观,便于监视和检查故障。因此在状态化简时,应首先考虑工作性能的优劣,维修是否方便,工作是否可靠直观,而不必过分追求最简状态。这样虽然增加了一些硬件设备,却换得了设计简便、工作明确、维修方便等好处。

控制器的形式多种多样,但其基本设计方法有较强的规律性。目前,使用算法流程图(ASM)是设计数字系统的最好方法。ASM图将控制器的控制过程用图形语言方式表达出来,

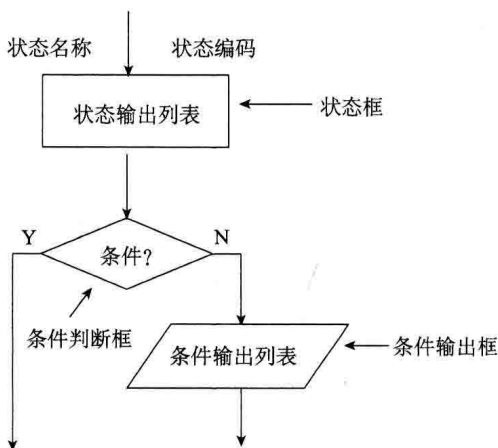


图 1.6 ASM 图的基本图形

类似于描述软件程序的流程图。ASM 图能和实现它的硬件很好地对应起来，显示了软件工程与硬件工程在理论上的相似性和可转换性。ASM 图是设计控制器的重要工具，主要由状态框、条件判断框、条件输出框、开始块和结束块等组成，如图 1.6 所示。下面先介绍 ASM 图的基本图形。

状态框是一个具有进口和出口的矩形框，代表系统的一个状态。状态经历的时间称为状态时间，在同步系统中，状态时间为同步时钟周期的整数倍且至少为一个同步时钟周期。状态名称写在框外左上方，状态编码写在框外右上方，状态输出列表(操作内容)写在矩形框内。

条件判断框简称分支框，用单入口双出口的菱形或单入口多出口的多边形(多个条件)表示。框内写检测条件，出口处注明各分支所满足的条件。

条件输出框由平行四边形组成。它的入口必须来自条件判断框的一分支，当分支条件满足时，给出指定的输出，输出的操作内容写在框内。注意，条件输出框不是控制器的一个状态，它经历的时间取决于状态时间。

算法流程图的开始块和结束块符号如图 1.7 所示，在算法流程图中使用开始块和结束块的目的是提高流程图的可读性。开始块和结束块不进行任何操作。

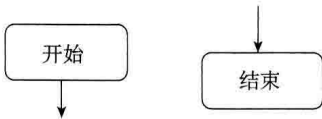


图 1.7 算法流程图的开始块和结束块

1.2.3 节将通过两个实例详细介绍数字系统的设计方法，重点介绍 ASM 图的画法。

1.2.3 数字系统设计实例

1. 频率测量系统控制器的设计

1) 频率测量原理

频率是指周期性信号在单位时间(1s)内变化的次数。若在一定时间间隔 T (即闸门时间)内计得这个周期性信号的重复变化次数 N ，则频率 $f=N/T$ 。根据这个公式可得如图 1.8 所示的数字频率测量系统(频率计)的原理框图。

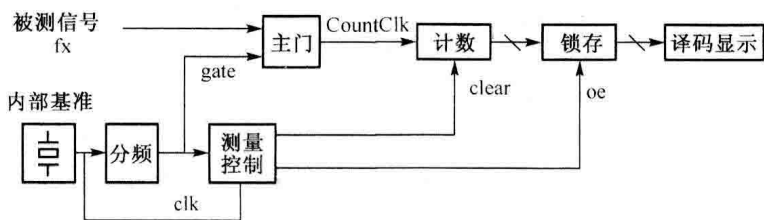


图 1.8 频率测量系统的原理框图

图 1.9 所示为频率测量系统的测量时序图,工作过程为被测信号 f_x 送入主门一个输入端,主门开通与否由主门的另一个输入端门控信号 $gate$ 决定。在主门开通前,先由清零信号 $clear$ 将计数器清零,在主门开通时间 T 内,被测信号 f_x 通过主门送至十进制计数器进行计数。当主门时间 T 结束后,控制电路首先产生数据锁信号 oe 将计数值 N 送至寄存器中锁存并显示,完成一次测量,然后测量控制电路产生一个复位信号,将计数器清零,同时将分频电路复位,等待下一次测量。当 $T=1s$ 时,显示值 N 为频率,而当 $T=1ms$ 、 $10ms$ 、 $100ms$ 时只需重新定位小数点即可。

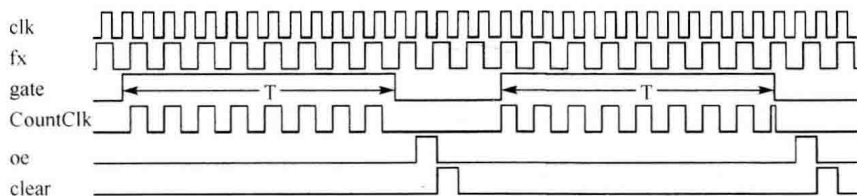


图 1.9 频率测量系统的测量时序图

2) 频率测量控制器的 ASM 图设计

从图 1.8 可看出,数字频率计的数据通道就是计数器、锁存器、显示译码器等较为简单的单元电路。系统设计的核心是控制器的设计,根据图 1.9 的测量时序图,可将频率测量分为 4 个工作阶段。

(1) 测量准备阶段,这一阶段对频率计数器进行清零,为下一次测量做准备。

(2) 测量等待阶段,等待门控信号 $gate$ 高电平到来,在这一阶段,频率计数器处于保持状态。

(3) 频率测量阶段,即门控信号 $gate$ 高电平期间,这期间,频率计数器对被测信号进行计数。

(4) 测量结果输出阶段,锁存最新测量结果,并将结果送入译码显示。

根据频率测量的 4 个工作阶段,控制器相应设置 4 个状态:计数器清零、等待 $gate$ 正脉冲、计数测量和测量结果锁存。根据图 1.9 的频率测量系统的测量时序要求,可以画出控制器的 ASM 图,如图 1.10 所示。

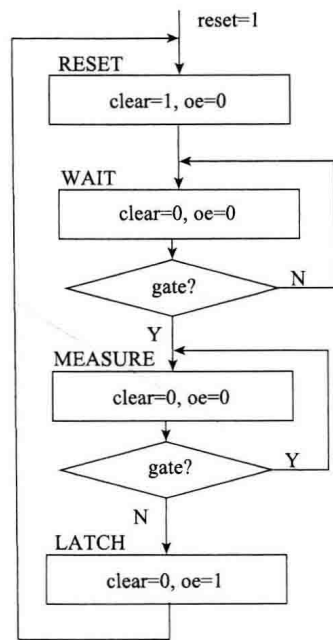


图 1.10 控制器的 ASM 图

从图 1.9 所示的时序图和图 1.10 所示的 ASM 图可看出,频率测量是连续测量。

有两点需要说明一下,一是从 ASM 可看出, WAIT 状态和 MEASURE 状态是可以合并的,但为了使设计更具可读性,一般来说保留两个状态。二是图 1.8 中没有 $reset$ 复位信号,且本系统中不需要该信号,但在控制器设计中,为了便于仿真和调试,复位信号是必不可缺的。因此,在设计时序模块时,一般要加入 $reset$ 复位信号,但在使用该控制器模块实例时,若不需要 $reset$ 复位信号,则可使该信号接低电平,即无效电平。

3) ASM 图与状态机图的转换