

手把手教你学 FPGA设计

——基于大道至简的至简设计法

潘文明
[网名 大道至简]

易文兵 编著



北京航空航天大学出版社
BEIHANG UNIVERSITY PRESS

手把手教你学 FPGA 设计

——基于大道至简的至简设计法

潘文明

易文兵 编著

[网名 大道至简]

北京航空航天大学出版社

内 容 简 介

本书介绍利用硬件描述语言(Verilog HDL)设计数字系统的方法,提出计数器、状态机、FIFO的三大架构,采用理论与实践相结合的方式,详细介绍各个架构的实现步骤——“八步法”,并配以相应的项目练习。在书的后半部分,介绍模块划分的基本方法,总结并提出了模块间的交互架构,在项目实践中给出了温度监控系统、边缘检测系统等综合性项目的模式。

本书可作为高等院校电子工程类、自动控制类、计算机类等专业的教材,亦可供 FPGA 相关行业的爱好者和从业人员自学与参考。

图书在版编目(CIP)数据

手把手教你学 FPGA 设计:基于大道至简的至简设计
法 / 潘文明,易文兵编著. -- 北京:北京航空航天大学
学出版社,2017.4

ISBN 978-7-5124-2401-2

I. ①手… II. ①潘… ②易… III. ①可编程序逻辑
器件 IV. ①TP332.1

中国版本图书馆 CIP 数据核字(2017)第 086191 号

版权所有,侵权必究。

手把手教你学 FPGA 设计 ——基于大道至简的至简设计法

潘文明 易文兵 编著
[网名 大道至简]
责任编辑 王 实

*

北京航空航天大学出版社出版发行

北京市海淀区学院路 37 号(邮编 100191) <http://www.buaapress.com.cn>

发行部电话:(010)82317024 传真:(010)82328026

读者信箱:emsbook@buaacm.com.cn 邮购电话:(010)82316936

北京市同江印刷有限公司印装 各地书店经销

*

开本:710×1 000 1/16 印张:19 字数:405 千字

2017 年 5 月第 1 版 2017 年 5 月第 1 次印刷 印数:3 000 册

ISBN 978-7-5124-2401-2 定价:45.00 元

若本书有倒页、脱页、缺页等印装质量问题,请与本社发行部联系调换。联系电话:(010)82317024

序

随着微电子技术的发展,可编程门阵列(FPGA)技术已经成为信息产业最热门的技术之一,其应用范围遍及电子、通信、自动化、医疗、军事和航空航天等多个热门领域。在数字集成电路设计领域,前端验证工作通常依托FPGA,并由那些精通硬件构架的FPGA系统工程师来完成。因此,FPGA技术的设计和开发已成为当前发展迅速的朝阳行业之一。

作者曾在华为海思、展讯通信等集成电路设计企业参与了FPGA大型项目的开发设计,近年来又在明德扬科技教育公司从事FPGA的教学培训,从而有机会深入研究和采用多种教学方法进行试验,以便把那些构成FPGA设计基础的知识模块和工具平台,变成FPGA学习者能够实际运用的方法,而更重要的是能够帮助学习者消化核心功能及其设计流程。于是,本书便应运而生,旨在为FPGA学习者提供一种FPGA的至简设计方法。

与一些将FPGA设计的理论知识和软件工具作为叙述重点的教程不同,本书的重点不在于介绍FPGA设计“是什么”,而侧重于引导学习者“怎么做”。经过这样设计后的内容可以引起学习者浓厚的兴趣,并使学习者在短时间内快速掌握FPGA设计方法,正如诺贝尔文学奖获得者、法国作家Anatole France的精辟格言:“教学的全部艺术就是唤醒年轻心灵的天然好奇心。”

黄君凯

2016年10月于广州暨南园

前言

随着我国工业智能化的发展,工业企业的个性化需求剧增。相应的,对 FPGA/IC 行业人才的需求,特别是高端人才的需求也随之剧增。然而,目前国内的 FPGA/IC 教育相对落后:高等院校缺乏项目经验丰富的教师;现有教材陈旧,无法跟上最新技术的发展;学习资料缺乏,大都重视软件操作示范,不重视设计理念传授等。这些客观因素使学生对 FPGA/IC 望而生畏,即使有大量学生立志学好 FPGA/IC 设计,但多数不得要领,最终放弃学习。由于失去了群众基础,导致国内缺乏高素质的 FPGA 人才,从而对我国工业智能化的发展产生了一定的影响。为了扩大 FPGA/IC 设计的群众基础,降低 FPGA/IC 设计的学习门槛,明德扬科技教育公司(简称明德扬)结合培训班、企业内训,以及多年国内顶尖企业的工作经验,研发出一套 FPGA/IC 设计流程——“至简设计法”。“至简设计法”是明德扬丰富项目经验的结晶,它将高级的设计技巧转化成几条规则,掌握了这些规则就相当于拥有多年的经验。“至简设计法”流程中的每一步都简单易操作,容易上手,便于掌握。掌握“至简设计法”,可以实现 0 仿真、0 调试、一次成功的设计,极大地提高了工作效率。明德扬致力于打造“最简单、最实用、最高效”的 FPGA/IC 设计法,不遗余力地为普及 FPGA/IC 设计,提升客户竞争力,提高我国该行业整体的设计能力做贡献。

很多同学在刚开始学习 FPGA/IC 设计时都会走弯路,下面是两位学员的学习经历与心得。

学员一:

- “① 买一本书,花大量的时间学习语法;
- ② 到网上搜一些资料,类似“大全”、“玩转”之类的速成之法;
- ③ 买一个开发板,看代码、下载例程看现象。”

学员一的情况具有普遍性,作为一个刚接触 FPGA 的初学者,这似乎是必经之路,很多学员在来明德扬学习之前就以这样的方式学了大约半年的时间。当然,这样的学习方法不是说学不好,而是很难学好,或者需花费很长的时间。以看书学习为例,可以说书是根源,学习任何知识都离不开书,书作为一个知识体系,要求全面、系统,但作为初学者就没有必要学得那么全面、系统了。很多学员在学 Verilog 时,就是拿一本书,仔仔细细地看,花费了不少时间,但效果却不理想。我们都知道,Verilog 的很多语法是不能综合的,而这些语法也没有实际的电路与之对应,当然在



具体的逻辑设计时也是不能用的,但书上并不会很明确地指出。再举一个例子,也是很多人都犯过的一个错误:以普通信号作为 always 的敏感信号列表。当我们这样写的时候,根本不会意识到这是错误的,因为书上就是这么教的。有经验的工程师都知道,对于 Verilog 语法,书上是一个全集,很多用法作为知识点是对的,但在实际应用中就是错误的。因此,太依赖书本的学习,反而会学得全而不好,还浪费时间。

学员二:

“一开学我立马就买了当时很多人推荐的某一品牌的一款开发板,但却不知道怎么学,上来就死抠代码,感觉好费劲啊!一方面我当时的基础确实很差,本科时候就没写过任何代码;另一方面看别人写的代码本来就费解,而卖家的视频也是蜻蜓点水一带而过,视频上提到的问题不是我的问题,我想解决的问题视频中又没说,反正我的困惑与卖家的视频就不在一个频道上,可能我的问题人家压根儿就觉得不是问题,因为我当时就是不折不扣的零基础。就这样,我混混沌沌地看了好多天代码,最终还是云里雾里。当时有人跟我讲过要去仿真看波形,而不是看代码,但是装什么软件,怎么装,软件怎么用,我一点都不知道。确实蛮痛苦的。”

以上两位同学遇到的问题,相信大多数初学者也遇到过,也很受困扰。明德扬在培训学员的过程中发现了一个普遍存在的问题:不同的人,代码风格往往也是不一样的。而一套优秀的代码规范,对团队、企业有非常强的指导意义,可以给他人提供一个可读性强、便于维护纠错的环境,这可以大大提高团队、企业的工作效率。

而市面上大多数 FPGA 教材往往并不重视代码规范,大篇幅的软件使用、语法知识和大量理论知识令初学者望而生畏,其实很多语法在实际工程中并没有机会用到。另外,很多 FPGA 书籍都会讲一个个独立的项目如何实现,以及针对不同项目的相应设计思路,但却缺少一套通用的设计方法,这样同学们最多只能学一个项目会一个项目,而不能举一反三。也正是因为如此,才有这本书创作的初衷。

为了帮助更多的 FPGA 学习者更好地理解 FPGA 设计的精髓,提高学习效率,明德扬提出了“至简”的设计思想,并创作了这本书。

大道至简与至简设计法

“大道至简”出自老子的《道德经》:“万物之始,大道至简,衍化至繁。”大道至简是指大道理(基本原理、方法和规律)是极其简单的,简单到一两句话就能说明白,把复杂冗繁的表象层层剥离之后就是事物最本质的东西。所谓“真传一句话,假传万卷书”。一门技术一门学问,弄得很深奥是因为没有看穿实质,就像 FPGA 的学习,搞得很复杂是因为没有抓住代码的关键。

1. 大道至简,抓住根本

大道理是极其简单的,简单到一两句话就能说明白。事情难就难在简单,简单不是敷衍了事,也不是单纯幼稚,而是最高级的智慧和成熟睿智的表现。

至简设计法是明德扬通过多年的教学经验找到的学习 FPGA 的关键和根本方法。FPGA 学习既不能花大量时间在软件工具/语法上,也不能花精力在独立的知识点上,而应花费 80% 的时间在 20% 的核心功能上。

什么是核心功能? 至简设计中一个完整的设计流程就是核心功能。注意,这里说的是“设计”流程,而非项目流程。市面上的 FPGA 设计教程大部分讲的都是项目流程,即介绍功能→新建项目→加入设计文件→编译工程→配置引脚→上板。这些流程都是介绍软件是怎么用的,而最重要的——设计文件是如何设计的,都没有详细介绍。

至简设计法的设计流程,讲的是实现一个功能的过程,是将功能转化成代码的过程。不仅简单的 UART/SPI/VAG/IIC/摄像头采集,甚至复杂的边缘检测/SDRAM 接口/软件无线电等,都可以采用这个设计流程将功能转化成代码。

2. 大道至简,衍化至繁

大道至简是一种哲学,无论是做人还是做事它都会给人以指导。在当今这个大千世界中,我们要学会把复杂变成简单,用智慧创造“简单”,在变迁中不断地升华。

至简设计法是先将复杂的模块划分成简单的模块,再用模块设计流程设计出来。FPGA 学习的过程,就是不断地将复杂的模块,划分成简单模块,再用至简设计法设计的过程。不断地重复这个过程,像搭积木一样一层层地堆砌,最终就能通过简单的规则做出复杂的设计。

3. 大道至简,凝聚智慧

大道至简,经过整合创新,跳出原来的框框,去粗取精,抓住关键和根本,挥动奥卡姆剃刀,剔除无效的、可有可无的、非本质的东西,提炼出精华。

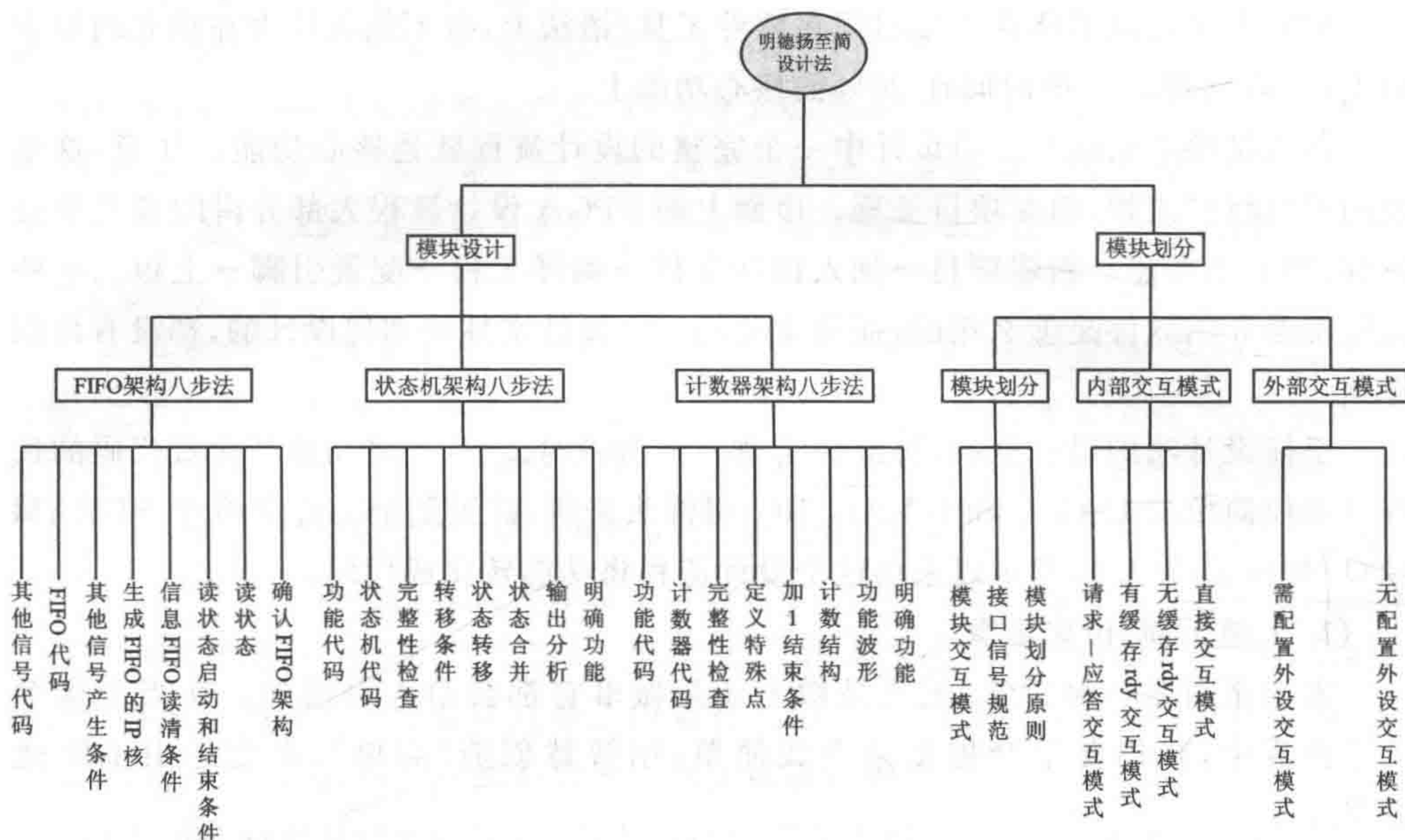
至简设计法,看似简单的几个规则,然而总结归纳出这些规则,却需要相当大的智慧:

- 必须要有众多领域的项目经验,才能获知各领域的设计需求和要点;
- 必须要有丰富的工作经验,才能拥有高超的设计技巧;
- 必须要有丰富的教学经验,明白学习的难点和重点,明白学生思考的方式,才能将高超的设计技巧转成可学习和操作的流程;
- 必须要有很大的智慧,不断地归纳和总结,才能将这些流程和步骤变得简单易用。

至简设计法,是结合以上要点,经过明德扬多年探索的智慧结晶。在此,我希望与更多 FPGA 学习者分享,让更多的人应用此设计方法,在 FPGA 设计学习和做实际项目中得心应手,为推进国家产业智能化发展贡献绵薄之力。



至简设计法内容体系



本书共 8 章,分为三篇,其中每篇对应的章和具体内容介绍如下:

第一篇包括第 1~4 章,主要介绍 FPGA 设计核心——模块。本书提出把模块分为三大类型,即计数器架构、状态机架构及 FIFO 架构,并配有项目设计练习。

第二篇包括第 5、6 章,主要介绍模块的由来——模块划分。本书总结了两大类常见的模块划分应用场合,即 FPGA 内部模块之间的交互和 FPGA 与外设的交互,并配有实践练习。

第三篇包括第 7、8 章,主要介绍基于 OV7670 的图像采集,以及基于 SDRAM 的 VGA 显示控制器。应用前两篇提到的三大架构及模块划分来完成这个工程,进一步深化对前面内容的理解。

本书特点

1. 注重设计思路

本书重点讲项目实现的整体思路,而并非讲某一具体项目的实现。

2. 设计理念

本书提出了“至简”的设计理念,一次只考虑一个问题。

3. 优秀的代码风格及规范

本书提供的代码风格统一,代码的规范是资深从业人员从多年工作经验中总结、提炼,并结合明德扬就业培训班培训成果所得到的。优秀的代码风格对团队、企业都有非常大的指导意义。

4. 注重实践

本书选择以案例的形式进行编写,所有方法均用例子阐述,并配有丰富的项目实践练习。

5. 不罗列知识点

坚持项目中遇到什么问题就解决(讲解)什么问题,避免罗列知识点,让 FPGA 学习者集中精力学习 FPGA 设计的核心。

如何学习本书

至简设计法的目的,是对程序编写工作中易出现的混乱状态进行规范化,以达到简便、有序、快捷的效果。本书通过例题引出概念,并详举大量实际案例进行分析、对比,帮助读者理解至简设计法的设计思想,清晰直观地学习和掌握该方法。此外,作者对于“二八定律”在 FPGA 设计方面的应用有独到见解:采取“权值阶梯模式”,即强调核心功能的学习、巩固和使用,以及一般功能在实际中边做边学的模式,来达到在最短的时间能独立完成项目的效果。另外,希望读者在完成项目时,先根据功能要求自己试着构思、编写代码,再与给出的思路、代码相比较,这样读者才能深刻理解至简设计法,并把这套设计思想真正运用到实践中去。

从明德扬就业培训班的培训成果来看,只要耐心学完整套设计思想,就能独立完成项目设计。设计者拿到一个新项目,知道从哪里下手,且编写的代码可综合,逻辑缜密,在设计过程中对逻辑变化考虑周全,就可在短时间内调试成功并上板实现。

本书虽然没有语法篇,但通篇用到的语法都是比较简单、常用的,有一定 Verilog 基础的读者读起来必定毫无障碍,而初学者也不必担心看不懂,明德扬会推出配套视频,介绍用到的语法知识点。另外,软件使用篇也包括在配套视频中,主要包括 Quartus II, Modelsim, Gvim 的使用。

本书可作为高等院校电子工程类、自动控制类、计算机类等专业的教材,亦可供 FPGA 相关行业的爱好者和从业人员自学与参考。

感谢参与本书部分内容编写及在编写过程中提供宝贵建议的黄君凯、崔苗、李国庆和陈泽熹。还要感谢北京航空航天大学出版社的编辑老师们,在他们的大力支持和帮助下本书才得以与广大读者见面。

由于作者水平有限,难免存在错误和遗漏之处,恳请各位读者批评指正。同时,也欢迎大家就 FPGA 开发相关技术与作者交流,联系邮箱:book@mdy-edu.com。

配套视频、开发板购买地址以及例程源码和相关资料下载地址:<http://www.mdy-edu.com/>。

潘文明 易文兵

2016 年 12 月于深圳明德扬科技教育有限公司

目 录

第一篇 模块设计

第 1 章 模 块	2
1.1 模块结构	2
1.2 模块的三大架构概述	4
第 2 章 计数器架构	6
2.1 计数器架构八步法	6
2.1.1 计数器规则	6
2.1.2 第一步:明确功能	9
2.1.3 第二步:功能波形	10
2.1.4 第三步:计数结构	10
2.1.5 第四步:加 1 和结束条件	11
2.1.6 第五步:定义特殊点	12
2.1.7 第六步:完整性检查	12
2.1.8 第七步:计数器代码	13
2.1.9 第八步:功能代码	14
2.1.10 总 结	16
2.2 计数器项目实践	16
2.2.1 PWM 流水灯项目	16
2.2.2 UART 接口项目	22
2.2.3 VGA 接口项目	34
2.2.4 SCCB(IIC)接口项目	42
第 3 章 状态机架构	54
3.1 状态机架构八步法	54
3.1.1 状态机规则	54
3.1.2 第一步:明确功能	58
3.1.3 第二步:输出分析	58
3.1.4 第三步:状态合并	59
3.1.5 第四步:状态转移	60



3.1.6	第五步:转移条件	60
3.1.7	第六步:完整性检查	63
3.1.8	第七步:状态机代码	63
3.1.9	第八步:功能代码	67
3.1.10	总 结	68
3.2	状态机项目实践	68
3.2.1	简易报文类型识别器项目	68
3.2.2	SPI 接口项目	78
3.2.3	SDRAM 接口项目	88
第 4 章	FIFO 架构	112
4.1	FIFO 架构八步法	112
4.1.1	概 述	112
4.1.2	第一步:确认 FIFO 架构	123
4.1.3	第二步:读状态	124
4.1.4	第三步:读状态启动和结束条件	124
4.1.5	第四步:信息 FIFO 读清条件	125
4.1.6	第五步:生成 FIFO 的 IP 核	125
4.1.7	第六步:其他信号的产生条件	125
4.1.8	第七步:FIFO 代码	126
4.1.9	第八步:其他信号代码	127
4.1.10	总 结	129
4.2	FIFO 项目实践	129
4.2.1	报文汇聚模块项目	129
4.2.2	以太网切包模块项目	134
4.2.3	UDP 打包模块项目	142

第二篇 模块划分

第 5 章	模块划分核心知识	158
5.1	模块划分概述	158
5.1.1	模块划分原则	158
5.1.2	模块划分端口规范	159
5.2	模块划分过程	159
5.2.1	模块划分思路	159
5.2.2	模块的端口及模块之间的数据流向	162
第 6 章	模块划分常用架构及项目实践	164
6.1	直接交互架构	164

6.2	无缓存 rdy 交互架构	165
6.3	有缓存 rdy 交互架构	165
6.4	请求-应答交互架构	165
6.5	外设交互架构	167
6.6	模块划分项目实践	167
6.6.1	AT93C46 的存储与读取项目	167
6.6.2	温度采集与显示项目	170
6.6.3	OV7670 图像采集项目	172
 第三篇 项目实践 		
第 7 章	基于 FPGA 的温度监控系统	176
7.1	背景介绍	176
7.2	项目要求	176
7.3	基本原理	179
7.3.1	PC 端(串口调试助手)	179
7.3.2	温度传感器 DS18B20	179
7.3.3	数码管显示	181
7.3.4	蜂鸣器	183
7.4	设计方案	183
7.5	FPGA 设计实现	187
7.5.1	硬 件	187
7.5.2	Verilog 代码设计	188
7.6	功能仿真	222
7.7	板级调试	227
7.8	总 结	227
第 8 章	基于 FPGA 的图像边缘检测系统	228
8.1	背景介绍	228
8.2	设计要求	228
8.3	基本原理	229
8.3.1	PLL	229
8.3.2	OV7670 配置与采集	233
8.3.3	灰度转换	234
8.3.4	高斯滤波	235
8.3.5	Sobel 边缘检测	237
8.3.6	乒乓操作	238
8.4	设计方案	239



8.5	FPGA 设计实现	241
8.5.1	硬 件	241
8.5.2	Verilog 代码设计	242
8.6	功能仿真	279
8.7	板级调试	286
8.8	总 结	287
参考文献		288

第一篇 模块设计

- 第 1 章 模 块
- 第 2 章 计数器架构
- 第 3 章 状态机架构
- 第 4 章 FIFO 架构

第 1 章

模 块

1.1 模块结构

采用模块化的设计使系统更有条理且便于仿真和测试,那么整个项目的设计思想就是模块套模块,自顶向下依次展开。

本书主要以 Verilog 硬件描述语言为主,模块是 Verilog 的基本描述单位,用于描述每个设计的功能和结构,以及其他模块通信的外部接口。模块有五个主要部分:端口定义、参数定义(可选)、I/O 说明、内部信号说明、功能定义。模块总是以关键词 module 开始,以关键词 endmodule 结尾。它的一般语法结构如下:

1 module module_name(
2 clk , //端口1, 时钟
3 rst_n, // 端口2, 复位
4 dout // 其他信号,如dout
5);

6
7 parameter DATA_W = 8;

8
9 input clk ; // 输入信号定义
10 input rst_n; // 输入信号定义
11
12 output[DATA_W-1:0] dout; //输出信号定义

13
14 reg [DATA_W-1:0] dou;//信号类型(reg,wire)定义
15
16 reg signal1;//信号类型(reg,wire)定义
17
18 //..... 以下为描述功能部分//

➡ 端口定义

➡ 参数定义(可选)

➡ I/O说明

➡ 内部信号说明

```

19
20 // 组合逻辑写法
21 always@(*)begin
22 end
23
24 // 时序逻辑写法
25 always@(posedge clk or negedge rst_n)begin
26     if(rst_n==1'b0)begin
27     end
28     else begin
29     end
30 end
31
32 endmodule

```

功能定义(功能描述)

下面详细分析这段代码：

1. 模块的端口定义(第1~5行)

模块的端口声明了模块的输入/输出口。其格式如下：

```
module 模块名(端口 1, 端口 2, 端口 3,……);
```

其中,模块是以 module 开始,以 endmodule 结束。模块名是模块唯一的标识符,建议模块名尽量用能够描述其功能的名称来命名。模块的端口表示模块的输入和输出口,也就是它与其他模块联系端口的标识。

2. 参数定义(第7行)

参数定义是将常量用符号代替,以增加代码的可读性和可修改性,是一个可选的语句,不用时可以省略。参数定义的一般格式如下：

```
parameter DATA_W = x;
```

3. I/O 说明(第9~12行)

输入口：

```

input [信号位宽-1:0] 端口名 1;
input [信号位宽-1:0] 端口名 2;
……;

```

输出口：



```
output [信号位宽 - 1:0] 端口名 1;  
output [信号位宽 - 1:0] 端口名 2;  
.....;
```

输入/输出:

```
inout [信号位宽 - 1:0] 端口名 1;  
inout [信号位宽 - 1:0] 端口名 2;  
.....;
```

4. 内部信号说明(第 14~16 行)

在模块内,用到的和与端口有关的 wire 和 reg 类型变量的声明,如:

```
reg [width - 1:0] R 变量 1, R 变量 2 .....;  
wire [width - 1:0] W 变量 1, W 变量 2 .....;  
.....;
```

5. 功能定义(描述功能)(第 21~31 行)

模块中最重要的部分是逻辑功能定义部分。以下三种方法可在模块中产生逻辑:

- ① 用“assign”声明语句,如描述一个两输入“与”门:assign a = b & c;
- ② 模块例化;
- ③ 用“always”块。

看到这里,相信很多读者都会一头雾水,因为整个模块语法结构很复杂,有各种型号定义、类型,以及功能描述。为了避免学员花过多时间在这些固定格式的学习上,明德扬制作了一套 GVIM(Graphical Vi Improved)模板,只需根据端口名和信号位宽进行修改,就可以轻松地写出 1~4 部分。如在 GVIM 输入 Module(首字母大写),便可以得到上述语法结构中的模板。另外,输入一些其他指令可以得到不同的模板,如 Shixu(时序逻辑电路)、Zuhe(组合逻辑电路)、Jsq(计数器)和 Ztj(状态机)等,在后续章节使用时再做说明。

模块设计的难点在于功能定义部分,因此本书将重点介绍该部分的设计。设计模块的功能定义时只要按照明德扬模块架构一步步进行,不管多么复杂的功能,都能设计出条理清晰、高效正确的代码。

1.2 模块的三大架构概述

明德扬根据模块实现的功能,将其分成三种架构——计数器架构、状态机架构和 FIFO 架构。几乎所有模块,都可以应用以上三种架构进行设计。不同架构,设计步骤不同。本书将在第 2 章介绍计数器架构八步法及其项目应用,在第 3 章介绍状态