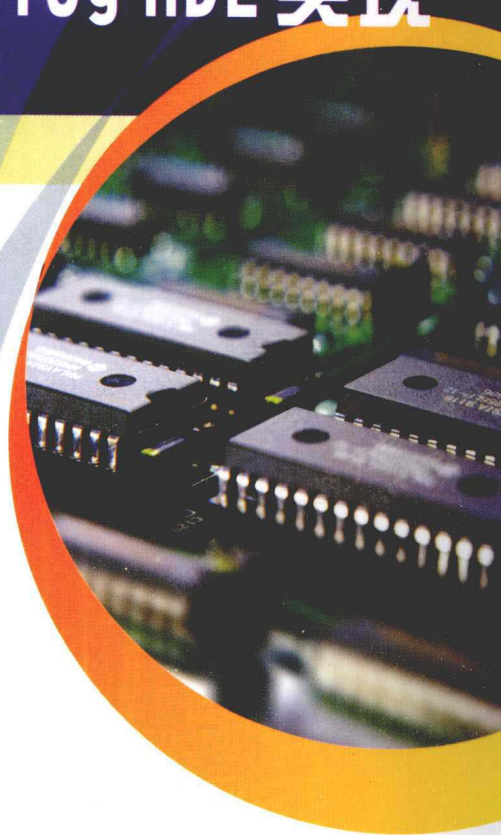
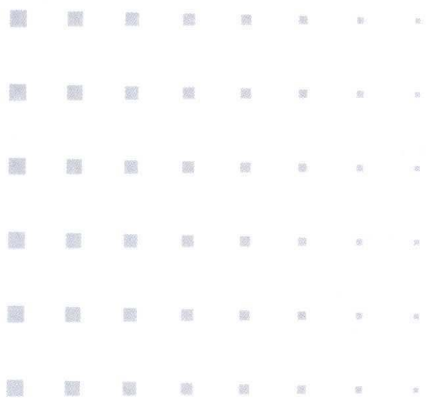




高等学校本科规划教材

# 数字电路设计及Verilog HDL实现

康磊 宋彩利 李润洲 编著



西安电子科技大学出版社  
<http://www.xduph.com>

面向 21 世纪高等学校本科规划教材

# 数字电路设计及 Verilog HDL 实现

康 磊 宋彩利 李润洲 编著

西安电子科技大学出版社

## 内 容 简 介

本书结合现代数字系统设计技术的发展,从教学和实际应用的角度出发,在系统地介绍数字电路分析和设计基本理论、基本方法的基础上,着重分析和说明采用 Verilog HDL 进行数字系统设计和实现的方法。本书主要内容涉及数字系统设计概述、数字技术基础、Verilog 基本语法、组合逻辑电路的分析和设计、时序逻辑电路的分析和设计、可编程逻辑器件原理、Verilog HDL 综合设计实例、Quartus II 开发平台简介等,并将 Verilog HDL 的介绍渗透于各个章节。

本书在内容上由浅入深,实用性强,既可以作为高等院校通信与电子类专业本科生的教材,也可以作为从事各类电子系统设计的科研人员和硬件工程师的应用参考书。

★ 本书配有电子教案,需要者可登录出版社网站下载。

### 图书在版编目(CIP)数据

数字电路设计及 Verilog HDL 实现/康磊,宋彩利,李润洲编著.

—西安:西安电子科技大学出版社,2010.12

面向 21 世纪高等学校本科规则教材

ISBN 978-7-5606-2482-2

I. ①数… II. ①康… ②宋… ③李… III. ①数字电路—电路设计—高等学校—教材  
②硬件描述语言, Verilog HDL—程序设计—高等学校—教材 IV. ①TN79 ②TP312

中国版本图书馆 CIP 数据核字(2010)第 190353 号

策 划 云立实

责任编辑 马晓娟 云立实

出版发行 西安电子科技大学出版社(西安市太白南路 2 号)

电 话 (029)88242885 88201467 邮 编 710071

网 址 www.xduph.com 电子邮箱 xdupfb001@163.com

经 销 新华书店

印刷单位 西安文化彩印厂

版 次 2010 年 12 月第 1 版 2010 年 12 月第 1 次印刷

开 本 787 毫米×1092 毫米 1/16 印张 25.375

字 数 605 千字

印 数 1~3000 册

定 价 36.00 元

ISBN 978-7-5606-2482-2/TN·0576

**XDUP 2774001-1**

\*\*\* 如有印装问题可调换 \*\*\*

本社图书封面为激光防伪覆膜,谨防盗版。

# 前 言

作为一名长期从事数字电路教学的教师，笔者一直希望在教学中能够将最新的专业理论和技术讲授给学生，但是教材总是滞后的事实使教师在教学中不得不面对诸多问题。学生学习完数字电路设计课程后，只能用传统电路设计方法实现系统，如果在其他教学环节，如课外电子大赛、课程设计、毕业设计等中涉及到复杂系统、EDA 设计、可编程器件等问题，则往往需要进行额外的教学或让学生自学。学生毕业后在实际工作中也经常面临同样的问题。因此，笔者萌生了根据自己的教学和实践经验编写一本教材的想法，希望在教材中能够融入可编程逻辑器件、硬件描述语言、EDA 开发技术等内容。

硬件描述语言在数字系统设计中扮演着极其重要的角色，有两种硬件描述语言先后被 IEEE 采纳，成为了标准的硬件描述语言，它们分别是 VHDL 和 Verilog。目前，国内有许多关于硬件描述语言的教材，其中大部分都是用 VHDL 语言来实现的。本书选用 Verilog 硬件编程语言的主要原因是：Verilog 与 C 语言有很多相同点，并且大专院校都开设 C 语言课程，这给教学带来很大方便。

为了保持数字电路内容的完整性和理论的系统性，本书以数字电路系统的设计原理为主线，同时系统地阐述了采用 Verilog 作为硬件编程语言进行 EDA 设计的方法。本书包括数制与编码、逻辑代数、组合逻辑电路、触发器、时序逻辑电路等基本内容；为了使读者掌握 EDA 技术，书中增加了 Verilog 语言规则、可编程逻辑电路等有关内容，以目前广泛使用的 Altera 公司的 Quartus II 7.2 EDA 开发平台为例，详细说明了进行 EDA 开发的流程；为了使读者全面系统地掌握 Verilog HDL 设计方法，书中还以几种典型数字系统设计为实例，详细说明了 EDA 设计实现的全过程。书中对由分立元件、中规模逻辑电路和可编程逻辑器件完成电路设计和实现的方法进行了分析和对比，由浅入深，层层深入。

本书共由 8 章组成，第 1 章介绍数字系统的基本概念和数字系统的设计方法，并对电子设计自动化(EDA)流程做了简要说明。

第 2 章首先说明数字信号的表示方法，然后介绍逻辑代数的基本定理、公式、逻辑函数及逻辑函数的化简，最后说明集成逻辑门电路的工作原理和外部特性。

第 3 章介绍硬件描述语言 Verilog 的基本语法规则和常用语句。

第 4 章详细阐述组合逻辑电路的分析、设计方法，对常用组合逻辑单元电路的功能和应用进行说明，并给出每种单元电路的 Verilog 描述。

第 5 章详细阐述时序逻辑电路的基本理论、分析和设计以及如何用 Verilog HDL 实现典型单元电路。

第 6 章介绍可编程逻辑器 SPLD、复杂可编程逻辑器件 CPLD 和现场可编程门阵列 FPGA 的原理与结构，并对 PLD 器件的编程方法进行简要说明。

第 7 章以综合实例的形式给出了采用 Verilog HDL 进行数字系统设计和实现的全

过程。

第 8 章简要介绍 Altera 公司的 Quartus II 集成开发环境的使用方法。

本书由康磊、宋彩利、李润洲合作编写。康磊负责编写第 4、5、7 章，宋彩利负责编写第 3、8 章，李润洲负责编写第 1、2、6 章。康磊负责全书大部分 Verilog 源代码的编写、调试和仿真，田超、卓涛进行了部分代码的编写和调试工作。

本书从教学和工程角度出发，力图做到理论严谨、内容新颖、实用性较强，将 EDA 设计技术应用于数字电路及计算机硬件的教学中，缩小高校教学与实际应用的距离，为今后的项目开发打下良好的基础。希望此书能够对电子工程人员和高校相关专业师生有所帮助。

由于作者水平有限，加之时间仓促，本书难免存在不足之处，敬请广大读者批评指正。作者的电子邮箱为 kangl@263.com。

编 者

2010 年 4 月 11 日

# 目 录

<b>第 1 章 数字系统设计概述</b> .....	1
1.1 数字系统的基本概念 .....	1
1.2 数字系统的设计方法 .....	4
1.2.1 三类常用芯片 .....	4
1.2.2 数字系统的设计过程 .....	5
1.3 EDA 技术基础 .....	7
1.3.1 硬件描述语言 HDL .....	8
1.3.2 EDA 软件开发工具 .....	10
1.3.3 EDA 芯片的设计开发流程 .....	11
习题 .....	12
<b>第 2 章 数字技术基础</b> .....	13
2.1 数制与编码 .....	13
2.1.1 进位计数制 .....	13
2.1.2 数制转换 .....	15
2.1.3 几种常用的编码 .....	17
2.2 逻辑代数 .....	20
2.2.1 基本逻辑运算 .....	20
2.2.2 复合逻辑运算 .....	23
2.2.3 逻辑函数 .....	27
2.2.4 逻辑代数的基本定律、公式和规则 .....	30
2.2.5 逻辑函数的标准形式 .....	32
2.3 逻辑函数的化简 .....	35
2.3.1 代数法化简逻辑函数 .....	36
2.3.2 卡诺图法(图解法)化简逻辑函数 .....	38
2.3.3 含有任意项的逻辑函数化简 .....	46
2.4 逻辑门电路 .....	48
2.4.1 概述 .....	48
2.4.2 TTL 集成逻辑门 .....	52
2.4.3 CMOS 电路 .....	60
习题 .....	66
<b>第 3 章 Verilog HDL 的基本语法</b> .....	70
3.1 Verilog HDL 程序的基本结构 .....	70
3.1.1 模块端口定义 .....	70
3.1.2 模块内容 .....	71
3.2 Verilog HDL 的数据类型 .....	72

3.2.1	常量 .....	72
3.2.2	变量 .....	74
3.3	Verilog HDL 的运算符 .....	76
3.4	Verilog HDL 的基本语句 .....	80
3.4.1	赋值语句 .....	80
3.4.2	条件语句 .....	82
3.4.3	循环语句 .....	83
3.4.4	结构声明语句 .....	85
3.4.5	编译预处理语句 .....	89
3.5	模块化程序设计 .....	91
	习题 .....	93
<b>第 4 章</b>	<b>组合逻辑电路</b> .....	<b>94</b>
4.1	组合逻辑电路概述 .....	94
4.2	组合逻辑电路分析 .....	94
4.2.1	组合逻辑电路分析方法 .....	94
4.2.2	简单组合逻辑电路分析举例 .....	95
4.3	组合逻辑电路设计 .....	97
4.3.1	用中小规模集成电路设计组合逻辑电路 .....	98
4.3.2	用 Verilog HDL 设计组合逻辑电路的方法 .....	98
4.3.3	组合逻辑电路设计举例 .....	100
4.4	常用组合逻辑电路 .....	108
4.4.1	加法器 .....	108
4.4.2	编码器 .....	113
4.4.3	译码器 .....	119
4.4.4	数据选择器和数据分配器 .....	131
4.4.5	数值比较器 .....	140
4.4.6	奇偶产生/校验电路 .....	144
4.5	组合电路中的竞争与险象 .....	148
4.5.1	竞争与险象的概念 .....	148
4.5.2	险象分类 .....	149
4.5.3	险象的判别 .....	151
4.5.4	险象的消除 .....	152
	习题 .....	153
<b>第 5 章</b>	<b>时序逻辑电路</b> .....	<b>157</b>
5.1	概述 .....	157
5.1.1	时序逻辑电路的特点 .....	157
5.1.2	时序逻辑电路的分类 .....	158
5.2	集成触发器 .....	158
5.2.1	触发器的工作原理 .....	159
5.2.2	常用触发器 .....	163
5.2.3	各种类型触发器的相互转换 .....	176
5.3	时序逻辑电路分析 .....	178
5.3.1	同步时序逻辑电路分析 .....	179

5.3.2	异步时序逻辑电路分析 .....	182
5.4	时序逻辑电路的设计方法 .....	185
5.4.1	同步时序逻辑电路的传统设计方法 .....	186
5.4.2	异步时序逻辑电路的传统设计方法 .....	200
5.4.3	用 Verilog HDL 描述时序逻辑电路 .....	203
5.5	常用时序电路及其应用 .....	209
5.5.1	计数器 .....	209
5.5.2	寄存器 .....	230
	习题 .....	249
<b>第 6 章</b>	<b>可编程逻辑器件 .....</b>	<b>254</b>
6.1	概述 .....	254
6.1.1	可编程逻辑器件的概念 .....	254
6.1.2	可编程逻辑器件的发展历程 .....	255
6.1.3	可编程逻辑器件的分类 .....	256
6.2	PLD 的编程元件 .....	257
6.2.1	熔丝型开关 .....	257
6.2.2	浮栅型编程元件 .....	258
6.2.3	SRAM 编程元件 .....	261
6.3	简单 PLD 的原理与结构 .....	262
6.3.1	PLD 的阵列图符号 .....	262
6.3.2	可编程逻辑阵列 PLA .....	264
6.3.3	可编程阵列逻辑 PAL .....	265
6.3.4	通用阵列逻辑 GAL .....	267
6.4	复杂可编程逻辑器件 CPLD .....	269
6.4.1	CPLD 的原理与结构 .....	269
6.4.2	CPLD 器件实例 .....	270
6.5	现场可编程门阵列 FPGA .....	275
6.5.1	FPGA 的原理与结构 .....	275
6.5.2	FPGA 器件实例 .....	277
6.6	CPLD 和 FPGA 的编程 .....	286
6.6.1	在系统可编程技术 .....	286
6.6.2	JTAG 边界扫描测试技术 .....	288
	习题 .....	292
<b>第 7 章</b>	<b>Verilog 综合设计实例 .....</b>	<b>294</b>
7.1	分频器的设计 .....	294
7.1.1	偶数分频器 .....	294
7.1.2	奇数分频器 .....	298
7.1.3	半整数分频器 .....	302
7.2	乐曲播放器 .....	304
7.2.1	时钟信号发生器模块 .....	305
7.2.2	音频产生器模块 .....	306
7.2.3	乐曲存储模块 .....	309
7.2.4	乐曲控制模块 .....	317



7.2.5 乐曲播放器顶层模块 .....	318
7.3 电子表 .....	319
7.3.1 时钟调校及计时模块 .....	319
7.3.2 整数分频模块 .....	322
7.3.3 时钟信号选择模块 .....	323
7.3.4 七段显示模块 .....	324
7.3.5 顶层模块的实现 .....	326
7.4 VGA 控制器 .....	326
7.4.1 VGA 显示原理 .....	326
7.4.2 VGA 控制信号发生器 .....	329
7.4.3 像素点 RGB 数据输出模块 .....	341
7.4.4 顶层模块的设计与实现 .....	342
7.4.5 RGB 模拟信号的产生 .....	344
7.5 简单模型机设计 .....	344
7.5.1 指令系统设计 .....	344
7.5.2 数据通路设计 .....	349
7.5.3 系统各功能模块设计 .....	351
7.5.4 指令时序设计 .....	361
7.5.5 控制器设计 .....	365
习题 .....	378
<b>第 8 章 Quartus II 开发平台简介 .....</b>	<b>379</b>
8.1 Quartus II 开发环境的建立 .....	379
8.1.1 软件的安装 .....	379
8.1.2 驱动程序安装 .....	380
8.2 Quartus II 设计流程与步骤 .....	381
8.2.1 创建工程 .....	381
8.2.2 设计输入 .....	385
8.2.3 编译设计文件 .....	389
8.2.4 仿真 .....	390
8.2.5 引脚分配 .....	394
8.2.6 下载验证 .....	395
8.2.7 对配置器件编程 .....	396
习题 .....	397
<b>参考文献 .....</b>	<b>398</b>

# 第1章 数字系统设计概述

数字系统在日常生活中扮演着越来越重要的角色，其例子不胜枚举，如手机、数字电话、数字电视、数码相机、手持设备、互联网络以及最典型的数字计算机等。除了在日常生活中的应用，数字系统也已广泛应用于通信、航天、自动控制、医疗、教育、气象等工业、商业、科学研究诸多领域。同时，从21世纪开始人类就进入了数字时代。

本章介绍关于数字系统、数字系统的设计方法和目前广泛采用的电子设计自动化(EDA)技术的相关基础知识。

## 1.1 数字系统的基本概念

### 1. 数字信号

数字系统的一个典型特征就是能够表示并处理离散的信息量。离散的信息量是非连续物理量的数值表述，这样的物理量称为数字量。例如，0~9十个阿拉伯数字、26个英文字母、工厂生产的产品个数、某一年级的学生人数等。数字量在时间和数值上都不连续，其变化总是发生在一系列离散的瞬间，如工厂生产的产品只能在一些离散的时间点完成，且产品个数只能一件一件地增加。

数字量在数字系统中只采用0、1两种数码表示，因为只有两种取值，所以称为二进制。二进制的0、1数码是构成数字信息的基本元素，不论所表示的是1位或多位的数值信息，或是存储设备的地址信息，或是计算机的指令信息，都用由这两个数码组合形成的二进制形式代码表示。采用二进制的另一个主要原因就是只需要两种不同的电路状态就可分别表示出0和1，不但电路易于实现，而且易于分辨。

表示0、1两种数码的两种不同状态，可以是低电平和高电平，也可以是无脉冲和有脉冲等。这类用于表示数字量，且参数具有离散特征的电信号称为数字信号。图1.1.1给出了数字量110100010的两种数字信号波形。图(a)中高电平表示“1”，低电平表示“0”；图(b)中有脉冲表示“1”，无脉冲表示“0”。

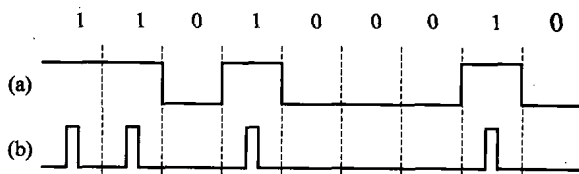


图 1.1.1 数字信号

数字系统可以对数字量进行描述和处理。但有些时候,人们也可能面对一些具有连续变化特性的物理量,如正弦交流信号、压力、温度、速度等,这样的物理量称为模拟量。模拟量在时间和数值上都是连续的,也就是说,在某一个瞬间,它可以是取值范围内某一区间的任意值。模拟量不能直接送入数字系统,需要经过采样、量化、编码,将之转化为数字量后才可由数字系统表示并处理。比如,采集正弦交流信号在某些时间点的电压值,并将其划分到某个数值等级内(量化),然后将每一个等级转化为一组对应的二进制编码。为处理模拟量,需要在数字系统中加入模/数(Analog/Digital, A/D)转换器来完成模拟量到数字量的转化。

## 2. 数字电路

工作于数字信号的电路称为数字电路。数字电路采用二进制,1、0两种数码正好对应于两种逻辑状态:真和假,因此可方便地进行逻辑运算和逻辑处理。逻辑运算是数字电路最基本的运算形式,因此数字电路也称为数字逻辑电路。

数字电路的发展经历了电子管、晶体管分立器件电路,直到现在广泛应用的集成电路。经过半个多世纪的发展,数字电路的应用几乎延伸到了所有的领域。

数字电路中,实现基本逻辑运算的电子电路称为门电路,如用于实现“与”运算的与门、实现“或”运算的或门等。采用少量的几个门电路还可以很容易地构成一种能够存储并记忆1位二进制信息的逻辑部件,称为触发器。门电路和触发器是数字电路最基本的电路单元。早期的门电路是由导线将晶体管、电阻等独立的元件连接形成的,称为分立元件门电路。从20世纪60年代开始,构成门电路或触发器的电子电路可以被制作于一块半导体芯片中,形成了目前广泛应用的集成电路IC(Integrated Circuit)。

集成电路的一个重要指标是集成度。集成度指每块芯片或芯片每单位面积中包含的晶体管的数量,通常用于表示集成电路的规模。集成门电路和集成触发器的结构简单,只包含几个少量的晶体管,集成度低,规模小,因此被称为小规模集成电路SSIC(Small Scale Integrated Circuit)。但从此后,随着集成电路技术的迅速发展,更多的晶体管,甚至是整个电路都能够被制作于一块芯片中,形成了规模更大的集成电路。按照摩尔定律,芯片的集成度每1~2年就能提高一倍,有望在2012年达到 $9718 \times 10^6$ 晶体管/片。

依据集成度的高低,数字电路分为小规模集成电路SSIC、中规模集成电路MSIC(Middle Scale Integrated Circuit)、大规模集成电路LSIC(Large Scale Integrated Circuit)、超大规模集成电路VLSIC(Very Large Scale Integrated Circuit)、甚大规模集成电路ULSIC(Ultra Large Scale Integrated Circuit)和巨大规模集成电路GSIC(Giga Scale Integration Circuit)等类别。按照制造工艺,数字电路又可分为TTL型和CMOS型。TTL电路采用双极型晶体管作为主要电子器件,问世较早,在长期的使用过程中逐渐演化成为一种电路标准。CMOS电路采用NMOS和PMOS两种互补的金属-氧化物半导体场效应晶体管作为主要电子器件,具有显著的低功耗、高密度等特性,这些特性对大规模集成电路的设计与制造非常重要。目前CMOS电路逐渐取代了TTL电路的主导地位,发展成为主流的电路形式。

数字电路的主要研究对象是电路的输入与输出之间的逻辑关系,所采用的主要分析工具是逻辑代数。逻辑代数是描述客观事物逻辑关系的数学方法,由英国数学家乔治·布尔(George Boole)于1849年创立,所以也称为布尔代数。逻辑代数广泛应用于线路设计和自

动化系统中,是分析和设计数字电路的数学基础。对数字电路逻辑功能的描述,通常采用的方法有:真值表、逻辑表达式、逻辑电路图、波形图、状态转移图等。近年来,使用硬件描述语言以文本的方式来描述电路的结构与功能,逐渐成为设计复杂数字逻辑电路的一种主要手段。

### 3. 数字系统

一直以来,数字电路的一项主要研究内容就是如何实现对数字信息的可靠存储、方便快速的运算及满足应用需求的各种操作处理。为达到这一目标,通常需要将多个数字电路的功能模块有机地组织成一个电子系统,在控制电路的统一协调指挥下,完成对数字信息的存储、传输和处理等操作,这样的系统称为数字系统。数字系统的实现基于数字电路技术,处理的是以二进制形式表示的具有离散特征的数据。从这个角度看,数字系统就是能够存储、传输、处理以二进制形式表示的离散数据的逻辑模块/子系统的集合。

数字系统的组成框图如图 1.1.2 所示,通常由控制电路、输入电路、输出电路、功能单元电路和时基电路组成。输入电路引入外部信号,如开关、按键的状态等。输出电路送出数字系统的处理结果,如将处理结果在发光二极管、七段数码管或液晶显示器上输出显示。功能单元电路按系统设计要求完成对数据信息的加工处理,通常包括存储电路和运算电路。不同应用目的的数字系统对数据有不同的处理操作要求,功能单元电路的结构与功能也不尽相同,复杂程度也可能有较大差异。有些系统的功能单元电路本身可能又由多个电路模块构成,因此在图中用虚线框表示。输入电路、输出电路和功能单元电路在数字信息的处理过程中执行具体的任务,它们需要在控制电路的统一调度指挥下,协调有序地动作,才能保证处理任务的正确执行。时基电路为所有的电路模块提供所需的定时信号。

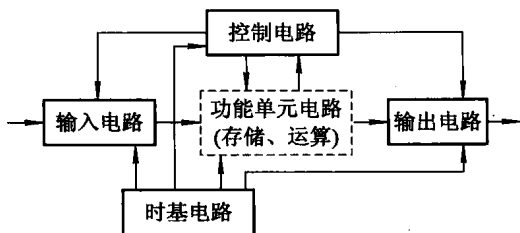


图 1.1.2 数字系统组成框图

数字系统区别于功能模块电路的一个典型特点就是在其组成结构中包含了控制电路。控制电路在时基电路产生的定时信号的作用下,按照数字系统设计的算法流程进行状态转移,在不同的状态条件下产生不同的用于控制其他各部件的控制信号,协调各部件的动作,实现自动连续的处理过程。一个典型的例子就是数字计算机。计算机由运算器、控制器、存储器、输入设备、输出设备五部分组成。运算器对数据进行算术和逻辑运算处理;存储器负责存储程序和数据;输入设备接收外部的信息,并将之转换为二进制代码存入存储器;输出设备将运算结果表达给最终用户。相对于控制器,这些部件都属于执行单元。控制器按照存储的程序,自动连续地逐条解释程序的每条指令,产生相应的控制命令以控制其他各部件的动作,实现复杂任务的处理。

数字系统区别于功能模块电路的一个典型特点就是在其组成结构中包含了控制电路。控制电路在时基电路产生的定时信号的作用下,按照数字系统设计的算法流程进行状态转移,在不同的状态条件下产生不同的用于控制其他各部件的控制信号,协调各部件的动作,实现自动连续的处理过程。一个典型的例子就是数字计算机。计算机由运算器、控制器、存储器、输入设备、输出设备五部分组成。运算器对数据进行算术和逻辑运算处理;存储器负责存储程序和数据;输入设备接收外部的信息,并将之转换为二进制代码存入存储器;输出设备将运算结果表达给最终用户。相对于控制器,这些部件都属于执行单元。控制器按照存储的程序,自动连续地逐条解释程序的每条指令,产生相应的控制命令以控制其他各部件的动作,实现复杂任务的处理。

数字计算机是一个典型的例子,也是一个复杂的例子,它的一个组成模块往往都比一些简单数字系统的规模更大。下面介绍数字系统的设计方法以及在数字系统设计过程中涉及的相关知识,目的是使读者学习完本书后,掌握与数字电路相关的基础知识,具备一些系统设计的基本技能。本书作为辅助说明而选用的例子,都是一些简单且易于实现的电路。关于计算机的组成结构、工作原理及实现,则需要在掌握了数字电路基础知识之后,通过一些专门的课程深入了解。

## 1.2 数字系统的设计方法

### 1.2.1 三类常用芯片

选择不同的器件设计开发数字系统,需采用不同的设计方法与流程。下面,首先介绍三类常用来实现数字系统的芯片。

#### 1. 标准芯片

设计数字系统,可以选用具有通用、固定逻辑功能的集成电路器件,如集成门电路、集成触发器、加法器、译码器、计数器等。有大量的这一类集成电路产品可供选择使用,虽然具体产品可能来自于不同的厂商,但一般都遵循统一的命名规则,相同编号的芯片具有相同的逻辑功能和引脚排列。这一类集成电路器件称为标准芯片。

标准芯片集成度通常都较低(一般低于 100 晶体管/片),只能实现一些简单、固定的逻辑功能。使用标准芯片设计数字系统时,需要先选择合适的芯片,利用芯片实现基本的逻辑功能模块,然后再根据系统逻辑功能需求,决定各模块之间的连接方式。多个具有不同逻辑功能的模块相互连接,可搭建成更大的逻辑电路。

采用标准芯片的设计方法主要用于 20 世纪 80 年代之前,其缺点是:

(1) 所需要的芯片个数多,占用电路板体积大,功耗大,可靠性差,难于实现复杂的逻辑功能。

(2) 逻辑功能固定,一旦完成设计,就很难再进行更改。

#### 2. 可编程逻辑器件 PLD

可编程逻辑器件 PLD(Programmable Logic Device)是 20 世纪 70 年代开始发展起来的一类集成电路器件。与标准芯片类似,PLD 具有通用的逻辑结构,可以按通用的集成电路器件进行批量生产。不同的是,PLD 内部包含大量的可编程开关,用户编程配置这些开关为不同的状态,就能实现不同的逻辑功能,而且这样的编程配置过程可以由最终的电路产品用户借助编程工具实现,而不必由芯片制造厂商来完成。

PLD 的优点主要表现在以下几个方面:

(1) PLD 作为通用芯片,可批量生产,成本低,但又可编程配置实现不同的电路,设计后能实现专用集成电路 ASIC(Application Specific Integrated Circuit)的功能。

(2) 大多数的 PLD 器件允许多次编程,便于系统的修改、升级和维护。

(3) PLD 的集成度高,与标准芯片相比,可以实现更复杂的逻辑电路。目前应用最广泛的一类 PLD 器件是现场可编程门阵列 FPGA(Field Programmable Gate Array),其集成度可达到千万级以上晶体管/片,甚至可以将整个系统制作于一个芯片内。由于大部分电路都可以在芯片内实现,因此,相对于标准芯片,使用 PLD 设计的电路具有功耗低、体积小、可靠性高等优点。

基于以上这些优势,PLD 器件获得了广泛的应用,成为了设计数字系统的一类主流器件。

### 3. 定制芯片

使用 PLD 设计数字系统,能够满足大多数应用的需求。但其内部的可编程开关在带来可编程定制、便于修改升级等优势的同时,也带来了一些缺点,如可编程开关耗费了芯片空间,限制了可实现电路的规模;增加了器件的成本;降低了速度性能;增大了功耗。

在一些对集成度、速度、功耗等性能要求较高的系统中,可以将设计好的电路交付半导体器件制造厂商,由厂商选择合适的技术来生产满足特定性能指标的芯片。这样的芯片可依据用户的要求定制生产,因此称为定制芯片。由于生产的芯片主要用于一些特定的应用场合,因此也称为专用集成电路 ASIC。

定制芯片按照其设计与生产的方式,可分为全定制芯片和半定制芯片两类。全定制芯片由设计者完全决定芯片内的晶体管数量、晶体管的放置位置、相互之间的连接方式等。半定制芯片是在厂商预构建的一些电路的基础上由用户设计版图,再交付生产厂家进行生产的。比如厂商可预构建一些标准单元或门阵列,用户基于这些标准单元或门阵列设计电路,然后由厂商根据用户的需求布线连接各单元,生产出满足功能与性能需求的芯片。相对于全定制芯片,半定制芯片可以减少设计的复杂性,缩短设计开发周期,但性能要差一些。

不论是全定制芯片,还是半定制芯片,它们的主要优点在于:针对特定的应用需求生产,能够根据特定的任务进行优化。相对于标准芯片和 PLD,定制芯片具有更好的性能,能够实现更大规模的电路。

定制芯片的缺点是:

- (1) 设计和开发周期长,产品投放市场时间长。
- (2) 生产过程中可能要经过多次反复的尝试,成本高,风险大。为降低成本,通常需要生产足够的数量,以降低每片芯片的平均价格。

定制芯片通常用于微处理器、信号处理等大规模专用集成电路。

## 1.2.2 数字系统的设计过程

### 1. 设计方法

数字系统的设计通常有两种方法:一种是自底向上(Bottom-Up)的设计方法;另一种是自顶向下(Up-Bottom)的设计方法。

#### 1) 自底向上

自底向上的设计方法是传统的使用标准芯片设计数字系统时所采用的主要方法。

自底向上的设计过程从底层设计开始。首先根据需求选择元器件;然后依据各个元器件的功能,设计实现各个独立的电路模块;最后将各个模块连接起来,组成完整的数字系统。

这种设计方法的优点是符合硬件工程师的设计习惯,缺点是:由于从底层独立模块的设计开始,系统的整体性能不易把握,而且只有在系统设计完成后,才能进行整体测试,一旦发现错误或系统不能满足某些指标要求,修改起来就比较困难。

#### 2) 自顶向下

传统的自底向上的设计方法主要用于数字系统的手工设计阶段,设计调试、错误排查、系统测试和修改都比较困难,难于实现大规模的复杂电路,已不能满足设计者的要求。近年来,电子设计自动化(EDA)技术以计算机为工作平台,以 EDA 软件工具为开发环境,

在多种不同的设计环境中都极大地影响着数字系统的设计过程,使得数字系统自顶向下的这一设计方法成为可能。

自顶向下的设计过程从系统的概念设计开始,描述并定义系统的行为特性,并在系统级进行仿真测试。然后,依据系统的功能需求,将整个系统划分为若干个相对独立的子系统。若子系统规模较大,还可以继续划分,直至划分为便于逻辑设计和实现的基本模块。这一划分过程不必考虑硬件的功能特性,完全可以依据系统的功能需求进行,但划分应遵循以下基本原则:

- (1) 各模块相对独立,功能集中,易于实现。
- (2) 模块间接逻辑关系明确,接口简单,连线少。

划分后的每个子系统/模块可独立进行设计、仿真及测试,设计完成后并入系统整体框架中,构成一个完整的系统。

自顶向下的设计方法从系统的整体结构向下,逐步求精,由高层模块定义下层模块的功能和接口,易于对系统的整体结构和行为特性进行控制。其次,划分后的每个子系统/模块相互独立,一方面便于多个设计者同时进行设计,对设计任务合理分配,用系统工程的方法对设计进行管理;另一方面,当设计不能满足某一方面的要求时,也便于将修改过程定位于某些具体的模块,若保持模块间的接口方式不变,则这种修改不会影响到其他电路模块的设计与实现,因此能够大大地缩短系统设计周期。模块的合理划分是设计的核心所在。

自顶向下设计方法的缺点是划分后的基本模块往往不标准,制造成本可能很高,而自底向上的方法采用标准单元,较为经济,但可能不能满足一些特定指标的要求。复杂数字系统的设计过程常采用这两种方法的结合,以综合平衡多个目标。

## 2. 数字系统设计流程

数字系统产品通常由一块或多块印刷电路板(Printed Circuit Board, PCB)构成。一个典型例子是微型计算机的主机板,它将多个用于实现逻辑功能的集成电路芯片以及一些其他的部件安装于电路板上,通过电路板的布线构成一个完整的系统。

自顶向下的数字系统产品设计的一般流程如下:

- (1) 明确设计要求,确定系统的整体设计方案。
- (2) 将系统划分为多个功能相互独立的子系统/模块。
- (3) 选择芯片,独立设计各个子系统/模块。
- (4) 定义各子系统/模块间的互连线路,将所有模块组合成完整系统。

(5) 对设计完成的电路进行功能仿真,检测其逻辑功能是否正确。早期,只有实际搭建完成电路后,才能验证设计是否正确。目前,许多计算机辅助分析软件都提供仿真功能,可以先对设计进行仿真模拟,尽早发现逻辑设计上的错误,避免不必要的时间和资金的浪费,待仿真正确后再进行实际电路的测试。

(6) 进行电路板的物理设计,包括确定电路板上每个芯片的物理位置、芯片之间的互连接模式等。随着芯片规模不断扩大,器件外围引脚越来越密集,使得电路板正确合理的布局布线成为一项繁重且复杂的工作,手工操作难以胜任。目前,这一阶段的工作多采用 PCB 计算机辅助设计工具软件进行,如 Protel。

(7) 对物理映射后的电路进行时序仿真。第(5)步的仿真过程主要用于检测电路的逻辑设计,确定其是否具有与设计预期相同的功能行为,称为功能仿真。即使一个功能仿真

正确的电路,在物理映射之后,也可能由于电路板物理布线时产生的各种干扰等而导致速度过慢,甚至不能正确操作,因此需要对综合了实际物理特性的电路进一步进行仿真检测。区别于功能仿真,这一时期的仿真称为时序仿真。时序仿真能够反映电路板的一些实质性的性能问题,时序仿真不正确的电路,需要返回电路板的物理设计阶段进行修正,若问题不能通过修改电路板的物理设计解决,就需要返回之前的设计过程进行修改,甚至是重新进行设计。

(8) 制作原型板、测试及投产。在上述的数字系统设计过程中,如果选择的芯片是PLD或定制芯片,那么在进行电路板设计之前,必须首先完成这些芯片的设计。随着集成电路集成度的提高,单个芯片内可以实现越来越多的电路,系统大部分的电路结构都可以移至芯片内实现,甚至可以将一个系统的所有核心电路都集成制造于一个芯片内(称为片上系统或SoC),而只在电路板上布局一些输入、输出等外围电路模块。因此,可以说数字系统的主要设计任务转移到了芯片设计方面。本书在后续章节中主要使用PLD器件阐述数字电路和数字系统的设计方法与过程。采用PLD的原因,一方面是由于它在成本、研发周期、多次编程及便于修改升级等方面的优势而获得的应用广泛性;另一方面在于它的用户可编程特性。终端用户可以自己设计电路和系统,编程下载后即可实现一个集成电子系统或形成一个专用集成芯片,可方便地对所完成的系统进行测试与验证。

基于PLD的集成电子系统或专用集成芯片的设计,对于复杂的系统,通常也需要划分为若干个功能相互独立的子系统/模块分别进行设计。电子设计自动(EDA)技术的发展,为芯片的设计与开发提供了许多便利的工具与手段,整个设计与开发过程几乎都可以在EDA软件工具的支持下自动完成。下面就EDA技术的基本概念、主要内容以及EDA技术支持下的集成电路芯片设计流程进行简单的介绍。

### 1.3 EDA 技术基础

EDA技术是一种汇集了计算机图形学、拓扑逻辑学、微电子工艺与结构学、计算数学等多种应用学科最新成果的先进技术,其研究对象是电子设计的全过程,涵盖的范畴相当广泛。目前,EDA还没有一个统一的定义,从集成电子系统/专用集成电路芯片设计的角度看,EDA技术是指:以大规模可编程逻辑器件为设计载体,以硬件描述语言为系统逻辑描述的主要表达方式,以计算机、大规模可编程逻辑器件的开发软件及实验开发系统为设计工具,通过有关的开发软件,自动完成用软件方式设计的电子系统到硬件系统的逻辑编译、逻辑化简、逻辑分割、逻辑综合及优化、逻辑布局布线、逻辑仿真,直至对于特定目标芯片的适配编译、逻辑映射、编程下载等工作,最终形成集成电子系统或专用集成芯片的一门技术<sup>[1]</sup>。

按这一定义,EDA技术的主要内容包括:大规模可编程逻辑器件、硬件描述语言、EDA软件开发工具和实验开发系统。在1.2.1节,对可编程逻辑器件进行了简单介绍,这些器件的具体原理及结构,将在第6章中详细介绍。实验开发系统通常用于电路或系统设计的测试与验证。面向特定应用的设计可能需要选择一些能够满足其需求的实验开发系统,但通常情况下,用于一般电路和系统测试的实验开发系统都会包含这样一些电路模块:可编程逻辑器件;编程/下载电路;常用的输入/输出电路,如按键、开关、发光二极管、7段数码管、液晶显示屏等;各种信号,如时钟、脉冲、高低电平等产生电路;用于连接其他电路模块的接口电路以及开发系统的扩展接口等。不同的实验开发系统有不同的配



置与结构,具体的使用需要参考相关的数据文档,这里就不再赘述。下面对硬件描述语言和 EDA 软件开发工具进行简要介绍。

### 1.3.1 硬件描述语言 HDL

#### 1. 硬件描述语言的概念

硬件描述语言 HDL(Hardware Description Language)是一种以文本形式描述数字电路和数字系统的语言。它类似于典型的计算机编程语言,可以使用计算机软件进行编辑、检索、编译等处理操作,所不同的是,它专门用于描述逻辑电路和系统的硬件结构与行为特性,而不用来编写计算机程序。

HDL 用软件方法描述数字电路和系统,允许设计者从系统设计的整体结构与行为描述开始,逐层向下分解设计和描述自己的设计思想,并能够在每一层次利用 EDA 工具进行相应的仿真验证。这使得电路或系统在实际构建之前就能进行功能测试,能够有效地降低设计成本,并缩短设计周期。

HDL 既可用于数字系统的行为描述,也可用于具体逻辑电路的结构描述。按照描述的层次,由高到底,可粗略地分为行为级、寄存器传输级 RTL(Register Transfer Level)和门电路级。寄存器由触发器构成,能够存储一组二进制信息,是数字系统的一类重要组成部分。寄存器传输级描述就是用数字系统内部的寄存器以及各寄存器(组)间二进制信息传输的数据通路(可以直接传送,或经过数据处理部件的加工)来描述数字系统。门电路级则是用构成数字系统的逻辑门以及逻辑门之间的连接模型来描述数字系统。寄存器传输级和门电路级与逻辑电路都有明确的对应关系,而行为级描述则不考虑硬件的具体结构。

高层次描述的电路和系统要得以实现,需要转化为底层的门电路级,这一转化过程称为综合。综合之后,还需要针对特定的目标器件,利用其内部资源进行合理布局,并布线连接各逻辑模块。这一过程称为适配或布局布线。在 EDA 开发工具的支持下,这些过程都可以自动完成,使得设计者不必太多考虑电路实现的细节,而将设计重心放在系统的行为与结构建模上,这样更有利于设计出正确的大规模复杂电路和系统。

#### 2. VHDL 与 Verilog

HDL 的发展至今已有 20 多年的历史,期间形成了多种不同的硬件描述语言。其中有很多都是专有的,也就是说,这些硬件描述语言由特定的公司或厂家提供,也只能用该公司或厂家提供的技术和产品设计数字系统。在公共设计领域,有两种硬件描述语言先后被 IEEE(the Institute of Electrical and Electronics Engineers)采纳,成为了标准的硬件描述语言,它们分别是 VHDL 和 Verilog。

VHDL 的首字母 V 是英文缩写 VHSIC(Very High Speed Integrated Circuit)的第一个字母,因此,其中文翻译应为甚高速集成电路硬件描述语言(VHSIC Hardware Description Language)。VHDL 由美国军方于 1982 组织开发,在 1987 年底被 IEEE 和美国国防部确认为标准硬件描述语言。

Verilog 于 1983 年初创于 GDA(Gateway Design Automation)公司。1989 年, Cadence 公司收购了 GDA 公司, Verilog 成为了 Cadence 公司专有的 HDL。1990 年, Cadence 公司开放了 Verilog,成立了一个公司和大学的联盟机构 OVI(Open Verilog International),并