

EDA技术实用丛书

数字电路**EDA**技术 入门与实战

■ 罗朝霞 赫建国 编著 ■

书中实例的源文件可到人民邮电出版社网站下载

 **人民邮电出版社**
POSTS & TELECOM PRESS

EDA技术实用丛书

数字电路**EDA**技术 入门与实战

■ 罗朝霞 赫建国 编著 ■

人民邮电出版社
北 京

图书在版编目 (CIP) 数据

数字电路EDA技术入门与实战 / 罗朝霞, 赫建国编著.
北京: 人民邮电出版社, 2009. 4
(EDA技术实用丛书)
ISBN 978-7-115-19676-7

I. 数… II. ①罗…②赫… III. ①数字电路—电路设计:
计算机辅助设计②硬件描述语言, VHDL—程序设计
IV. TN790. 2

中国版本图书馆CIP数据核字 (2009) 第009400号

内 容 提 要

本书从实际应用角度出发, 以帮助读者轻松地从事数字电路的传统分析设计方法过渡到采用 EDA 技术分析和设计数字电路为目的, 用与传统数字电路分析和设计类比的方法, 详细地介绍了基于 EDA 技术的现代数字电路的分析设计流程; 穿插介绍了 VHDL 硬件描述语言和 Quartus II 7.2 开发软件; 同时介绍了最普通的描述数字电路与系统的模型——有限状态机的基本概念、一般描述方法和设计过程; 通过两个数字电路与系统设计的实例, 介绍了应用可编程器件进行设计的方法; 最后讨论了可编程逻辑器件的编程与配置。

本书可供从事数字电路硬件设计的工程师阅读和参考, 可作为 EDA 技术短训班的教材, 也可作为高等院校通信工程、电子工程、计算机应用技术、数字信号处理等专业专科生、本科生或研究生的教材。

EDA 技术实用丛书

数字电路 EDA 技术入门与实战

-
- ◆ 编 著 罗朝霞 赫建国
责任编辑 陈万寿
 - ◆ 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 电子函件 315@ptpress.com.cn
网址 <http://www.ptpress.com.cn>
中国铁道出版社印刷厂印刷
 - ◆ 开本: 787×1092 1/16
印张: 13.5
字数: 329 千字
印数: 1—3 500 册
 - 2009 年 4 月第 1 版
2009 年 4 月北京第 1 次印刷

ISBN 978-7-115-19676-7/TN

定价: 29.00 元

读者服务热线: (010) 67129264 印装质量热线: (010) 67129223

反盗版热线: (010) 67171154

前 言

数字电子技术在各个领域的广泛应用,使其成为电子技术未来发展的趋势,而电子设计自动化(EDA, Electronic Design Automation)技术的不断发展与进步,也使得基于 EDA 技术的分析和设计方法正在成为数字电路与系统设计的主流。对于从事这方面工作的工程技术人员而言,掌握并具备应用这两种技术的技能,无疑是非常重要的。本书所要阐述的主要目的是:帮助那些掌握了传统数字电子技术的基本理论、方法和技能的读者,将学习的重点转移到以 EDA 技术为主要内容的现代数字电路与系统的分析与设计上来。

本书内容共分 8 章。第 1 章介绍数字电路与系统传统设计方法和 EDA 设计方法的主要内容,并对两者进行对比;第 2 章介绍数字电路不同描述方法的 EDA 实现,并对 VHDL 硬件描述语言的基本结构和主要的语法现象作了较详细的说明;第 3 章简要介绍组合逻辑电路的基本概念,具体说明如何用不同的方法分析和设计组合电路,并将一些常用 VHDL 语句的介绍穿插在组合电路的设计中;第 4 章简要介绍时序逻辑电路的基本概念,具体说明如何用不同的方法分析和设计时序逻辑电路,详细介绍一些常用时序逻辑电路模块的 VHDL 设计;第 5 章通过几个典型实例说明如何采用状态机模型进行数字电路系统设计,对设计中需要注意的问题进行了讨论;第 6 章介绍基于复杂可编程逻辑器件的交通灯控制电路的设计;第 7 章通过基于现场可编程阵列的信号产生器的设计,说明采用 EDA 技术设计数字电路与系统的完整过程;第 8 章介绍可编程逻辑器件编程/配置的相关知识。通过这些内容的学习,读者能够掌握应用 EDA 技术进行数字电路分析设计的方法和技巧。

本书第 1 至 5 章由罗朝霞编写,第 6 至 8 章由赫建国编写,全书由罗朝霞统稿。为了便于读者学习,本书配有实例和完整设计项目的代码,以及各章小结、思考与练习题。读者可登录人民邮电出版社的官方网站下载。

感谢为本书编写提供宝贵意见、建议和帮助的人们,本书在编写过程中还参考了不少专家和学者的著作,在此也表示深深的谢意。

EDA 技术的发展非常迅速,并逐渐向数字电子技术渗透,如何将这两者有机地融为一体,我们也一直处于不断地学习和探索之中。限于编者的理论水平和实践经验,书中难免有错误之处,恳请读者批评指正(本书责任编辑电子邮件地址:chenwanshou@ptpress.com.cn)。

作 者

2008 年 11 月

目 录

第 1 章 概述	1	方法	48
1.1 数字电路与系统设计分类	1	3.3.2 利用 EDA 工具设计组合	
1.2 传统数字电路与系统设计的		电路	50
主要内容	2	3.4 常用组合电路的 VHDL 设计	64
1.3 数字电路与系统 EDA 设计的		3.4.1 门电路	64
主要内容	2	3.4.2 编码器和译码器	67
1.3.1 可编程逻辑器件	3	3.4.3 数据选择器	75
1.3.2 软件开发工具	3	3.4.4 三态门电路	79
1.3.3 硬件描述语言	4	第 4 章 时序逻辑电路	84
1.4 数字电路与系统 EDA 设计的		4.1 时序电路概述	84
流程	5	4.1.1 时序电路的电路结构	84
1.5 数字电路与系统传统设计方法和		4.1.2 时序电路的描述方法	84
EDA 设计方法的比较	7	4.1.3 时序电路分类	85
第 2 章 数字电路不同描述方法的		4.2 时序电路的分析	85
EDA 实现	10	4.2.1 时序电路的传统分析	
2.1 数字电路原理图输入设计		方法	85
方法的 EDA 实现	10	4.2.2 利用 EDA 工具分析时序	
2.1.1 用原理图描述数字电路	10	电路	89
2.1.2 原理图输入设计方法的		4.3 时序电路的传统设计方法	91
EDA 实现流程	11	4.3.1 用触发器设计时序电路	91
2.2 数字电路文本输入设计方法的		4.3.2 用集成移位寄存器设计	
EDA 实现	21	时序电路	94
2.2.1 用 VHDL 描述数字电路	21	4.3.3 用集成同步计数器设计	
2.2.2 文本输入设计方法的		时序电路	96
EDA 实现流程	33	4.3.4 用集成异步计数器设计	
第 3 章 组合逻辑电路	35	时序电路	100
3.1 组合电路概述	35	4.4 常用时序电路的 VHDL 设计	102
3.2 组合电路的分析	35	4.4.1 触发器	102
3.2.1 组合电路的传统分析		4.4.2 寄存器	109
方法	35	4.4.3 计数器	114
3.2.2 利用 EDA 工具分析组合		4.4.4 分频器	118
电路	39	4.4.5 存储器及其应用	124
3.3 组合电路的设计	48	第 5 章 状态机设计	133
3.3.1 组合电路的传统设计		5.1 状态机的 VHDL 描述	133

5.2 状态机编码	138	7.3 方案论证	174
5.2.1 顺序编码	138	7.3.1 DDS 信号产生器芯片	174
5.2.2 一位热码编码	138	7.3.2 利用微处理器实现 DDS 信号产生器	176
5.2.3 状态位直接输出型编码	138	7.3.3 利用可编程逻辑器件 实现 DDS 信号产生器	177
5.3 状态机设计举例	141	7.4 系统设计	179
5.3.1 状态机设计流程	141	7.5 单元电路设计	180
5.3.2 序列信号检测器的设计	142	7.5.1 显示电路	180
5.3.3 A/D 采样控制器的设计	144	7.5.2 键盘电路	181
5.3.4 自动交通控制系统的 设计	148	7.5.3 数/模转换电路	185
5.4 状态机设计中需要注意的 问题	153	7.5.4 波形数据表	190
5.4.1 相位累加器	191	7.5.5 相位累加器	191
第 6 章 基于复杂可编程逻辑器件的 交通灯控制电路设计	154	7.6 系统连调	191
6.1 数字电路与系统设计的过程	154	7.6.1 各种时钟的产生	191
6.2 方案论证	155	7.6.2 单元电路的组合	193
6.2.1 设计要求	155	7.6.3 系统测试	196
6.2.2 设计要求的真值表描述	155	7.7 设计总结	196
6.2.3 基于微控制器的交通灯 控制电路设计	157	第 8 章 可编程逻辑器件的编程/配置	197
6.2.4 基于可编程逻辑器件的 交通灯控制电路设计	158	8.1 编程/配置模式	197
6.3 系统设计	159	8.2 ByteBlaster 下载电缆	198
6.4 单元电路设计	159	8.2.1 ByteBlasterMV 下载 电缆	198
6.4.1 显示电路	159	8.2.2 JTAG 编程/配置模式	200
6.4.2 时钟产生电路	161	8.2.3 被动串行方式 (PS) 模式	201
6.4.3 状态产生电路	165	8.3 可编程逻辑器件的编程/配置	202
6.4.4 控制信号产生电路	166	8.3.1 编程器的设置	202
6.5 系统连调	169	8.3.2 可编程逻辑器件管脚的 设置	203
6.6 设计总结	171	8.4 配置芯片	204
第 7 章 基于现场可编程阵列的信号 产生器的设计	172	8.4.1 配置芯片的类型	205
7.1 引言	172	8.4.2 配置电路	205
7.2 直接数字合成	173	8.4.3 配置芯片 EPC2 的编程	206
		参考文献	210

第 1 章 概 述

在自然界中,存在着许许多多的物理量。其中有这样一类物理量,它们在时间和数量上是不连续的,它们的变化总是发生在一系列离散的瞬间,它们的数量大小和每次的增减变化都是某一个最小单位的整数倍,而小于这个最小单位的数值是没有物理意义的。该类型的物理量称为数字量。表示数字量的信号称为数字信号。工作在数字信号下的电路称为数字电路。

数字电路的输入和输出信号都是数字信号。数字电路所做的工作就是对数字信号进行存储、传输和处理。数字电路既可以是一个逻辑部件,也可以是一个独立的实用系统。

1.1 数字电路与系统设计分类

自计算机诞生以来,数字电路与系统设计就分成了数字电路与系统硬件设计和数字电路与系统软件设计两大类。本书介绍的数字电路与系统设计指的是数字电路与系统硬件设计。

数字电路与系统需用数字器件来实现,数字器件不同则设计方法不同。根据所用数字器件的不同,数字电路与系统硬件设计分为 3 类:选用通用器件设计、选用可编程逻辑器件设计、设计专用集成电路。其中后两个任务都可借助 EDA 工具完成。设计好的数字系统可能是一块由许多种实现不同功能的通用器件组成的印制电路板,也可能是一块由少量可编程器件组成的印制电路板,还有可能仅仅是一个芯片。

通用集成电路芯片生产厂商提供的通用器件逻辑功能简单固定,通常可用它们组成复杂的数字电路系统。设计者在选择通用器件时,需要考虑所选的器件是否满足所需功能,如何将器件连接组成复杂的数字与电路系统。

虽然选用通用器件可以组成多种复杂的数字电路与系统,但为了减少系统电路的体积、重量、功耗和提高可靠性和效率,设计者往往把所设计的系统直接做成一片大规模或超大规模集成电路。这种为某种专门用途设计的集成电路被称作专用集成电路(ASIC, Application Specific Integrated Circuit)。这类集成电路的通用性差、设计制造成本高、周期长,并且风险较大。为了克服这些缺点,出现了可编程逻辑器件(PLD, Programmable Logic Device)。

可编程逻辑器件是 ASIC 的一个重要分支,它是厂家作为通用器件生产的集成电路。但是它的逻辑功能是由设计者根据系统设计的具体要求通过相应的器件编程来实现的。此外,由于可编程逻辑器件的集成度高,因此它可以满足大多数数字电路与系统设计的需要。设计者通过相应的器件编程就可以把设计的系统集成在一片可编程器件上,而不再需由厂商来设计和制造相应的集成电路,这样便解决了专用集成电路的专用性、成本高、开发周期较长及风险较大的主要矛盾。

1.2 传统数字电路与系统设计的主要内容

传统数字电路与系统设计包括电路设计、电路制作和电路调测 3 方面内容。

电路设计是指在分析了系统要求之后, 根据所选用的通用集成器件, 在进行方案对比分析并确定所选用方案之后, 设计各个模块电路以及总体电路的电路原理图及电路接线图。电路制作是选用面包板或者印制电路板, 根据接线图制作符合设计要求的电路。电路调测是在设计制作好的电路板上验证所设计电路的功能是否符合设计要求, 若不符合要求, 需要检查错误是由电路制作还是设计方案引起的, 并进行相应的处理, 直至符合要求为止。

采用传统方法设计数字电路与系统, 如果设计的是一个不太复杂的数字电路与系统, 选用中、小规模集成电路, 依靠手工设计还可勉强实现。若是一个选用超大规模集成电路实现的复杂系统, 特别是当电路系统非常庞大时, 仅仅依靠人力进行研制已无法完成, 即使是历经千辛万苦最终研制成功, 但由于周期过长, 早已落后, 因此失去了市场竞争力而惨遭淘汰。

1.3 数字电路与系统 EDA 设计的主要内容

由于可编程器件和计算机的出现, 人们发现硬件设计的许多事情可以交给计算机完成, 于是出现了电子设计自动化技术, 即 EDA (EDA, Electronic Design Automation) 技术。

EDA 技术发展迅速、涉及面广、内容丰富, 现在人们对其理解各异且尚未统一。通常情况下, EDA 技术可分为狭义 EDA 技术和广义 EDA 技术。狭义 EDA 技术是指以大规模可编程逻辑器件为载体, 以硬件描述语言 (HDL, Hardware Description Language) 为系统逻辑描述的主要表达方式, 借助计算机和可编程逻辑器件的软、硬件开发系统等工具, 通过相关的开发软件, 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、逻辑简化、逻辑分割、逻辑综合及优化、逻辑布局布线、逻辑仿真, 直至对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 ASIC 的一门新技术。数字电路与系统设计中提到的 EDA 技术指的是狭义的 EDA 技术。广义 EDA 技术是指以计算机和微电子技术为先导, 汇集了计算机图形学、数据库管理、图论和拓扑逻辑、编译原理、微电子工艺与结构学和计算数学等多种计算机应用学科最新成果的先进技术。也就是说, 广义 EDA 技术除了包括狭义 EDA 技术内容外, 还包括计算机辅助分析技术 (如 PSPICE、EWB、MATLAB 等), 印制电路板计算机辅助设计 PCB-CAD 技术 (如 PROTEL、ORCAD 等) 等内容。

数字电路与系统 EDA 设计框图如图 1-1 所示。

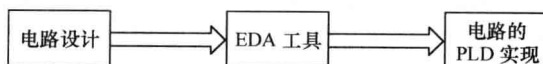


图 1-1 数字电路与系统 EDA 设计框图

图 1-1 中的电路设计是设计者在分析了设计要求之后, 用原理图或硬件描述语言来描述

符合要求的设计。在电路设计中常用和推荐的描述方式是硬件描述语言,因为它是利用 EDA 技术进行数字电路与系统设计的主要表达手段。EDA 工具是利用 EDA 技术进行数字电路与系统设计的自动化设计软件开发工具,它的任务是将设计者所做的设计最合理地映射到某个可编程逻辑器件的特定结构中去。没有良好的 EDA 工具的配合,可编程逻辑器件是无法应用的。电路的 PLD 实现是将所做的设计采用大规模可编程逻辑器件完成,大规模可编程逻辑器件是利用 EDA 技术进行数字电路与系统设计的载体,即它是 EDA 技术的物质基础。

从以上分析描述可知,数字电路与系统 EDA 设计主要包括这几方面的内容:可编程逻辑器件,EDA 软件开发工具,硬件描述语言。

1.3.1 可编程逻辑器件

可编程逻辑器件是 ASIC 的一个重要分支,它是一种由用户编程以实现某种逻辑功能的新器件,故可编程逻辑器件也称为可编程 ASIC。可编程逻辑器件是 EDA 技术的物质基础。

可编程逻辑器件从 20 世纪 70 年代发展到现在,在结构、工艺、集成度、速度和性能等方面都不断改进和提高,已由最初的简单低密度 PLD 发展到如今的复杂高密度 PLD。目前提到的可编程逻辑器件,一般是指复杂高密度 PLD 中的复杂可编程逻辑器件(CPLD, Complex Programmable Logic Device)和现场可编程门阵列(FPGA, Field Programmable Gate Array)。本书中提到的可编程逻辑器件也是指高密度可编程逻辑器件:CPLD 和 FPGA。

不同厂家对 CPLD 和 FPGA 的定义有所不同。通常,根据结构特点和工作原理,CPLD 和 FPGA 的分类方法是:以乘积项结构方式构成逻辑行为的器件称为 CPLD,如 Lattice 的 ispLSI 系列、Xilinx 的 XC9500 系列和 Altera 的 MAX7000S 系列等;以查找表法结构方式构成逻辑行为的器件称为 FPGA,如 Xilinx 的 SPARTAN 系列、Altera 的 FLEX10K、ACEX1K 和 Cyclone 系列等。FPGA 和 CPLD 都是可编程 ASIC 器件,有很多共同特点,但由于 CPLD 和 FPGA 硬件结构上的差异,又使得它们具有各自的特点。尽管如此,对用户而言,CPLD 和 FPGA 的设计流程是相似的,使用 EDA 软件的设计方法也没有太大的差别。设计时,只需用根据所需选器件型号充分发挥器件的特性就可以了。

如今,FPGA 和 CPLD 器件的应用已十分广泛,它们将随着 EDA 技术的发展而成为电子设计领域的重要角色。

1.3.2 软件开发工具

EDA 软件开发工具是 EDA 技术的强有力支持。EDA 软件开发工具是指以工作站或者高档计算机为基本工作平台,利用计算机图形学、拓扑逻辑学、计算数学和人工智能等多种应用学科的最新成果而开发出来的一套软件工具,它是一种帮助设计工程师进行电子系统设计的辅助工具。一般的 EDA 软件开发工具包括:编辑器、仿真工具、检查/分析工具、优化/综合工具和布局布线工具等。

编辑器的主要功能是用来对设计输入进行图形或文本等方面的编辑操作,它通常包括图形编辑器和文本编辑器。图形编辑器用来编辑表示器件的几何图形、电子系统的框图以及原理图等;文本编辑器在系统级上用来编辑电子系统的自然描述语言,在其他层次上主要用来编辑电路的硬件描述语言文本。

仿真器是用来设计仿真操作的一种工具,它是用来评价 EDA 工具的一项重要指标,其

功能是用来验证设计的正确性和准确度。仿真是 EDA 技术中的一项非常重要的技术,采用仿真便于在开发设计的早期阶段发现设计中的错误,这样做可以大大减少设计重复和修改的次数及时间,从而提高了设计者的工作效率。

检查/分析工具是用来对设计者的具体设计进行编译、检查和分析的,目的是发现设计中的错误和对可能的结果进行分析。通过分析检查/分析工具产生的各种报表文件,设计者可以对设计系统中的错误、系统性能以及相应的各种时序关系等有一个清楚地认识,从而能及时发现设计中的错误和检查系统性能能否满足设计的需求。

优化/综合工具是用来完成优化功能和逻辑综合功能的一种开发工具。其中,优化功能是指根据布尔方程功能等效的原则,采用不同的优化方法来对设计进行优化操作,从而提高设计系统的性能和占用较少资源;逻辑综合功能是将抽象描述转化成电路网表或者是一组逻辑方程的形式,目的是方便设计系统的具体实现。

布局布线工具实现由逻辑设计到物理实现的映射。最终物理实现对应的器件不同,则各自的布局布线工具也会有较大差异。

EDA 软件工具开发厂商大体可分两类:一类是 EDA 专业软件公司,较著名的有 Mentor Graphics、Cadence Design Systems、Synopsys 和 Viewlogic Systems 等;另一类是半导体器件厂商,为了销售他们的产品而开发 EDA 工具,较著名的公司有 Altera、Xilinx 和 Lattice 等。EDA 专业软件公司独立于半导体器件厂商,推出的 EDA 系统具有较好的标准化和兼容性,也比较注意追求技术上的先进性,适合于从事学术性基础研究的单位使用。而半导体厂商开发的 EDA 工具,能针对自己器件的工艺特点作出优化设计,提高资源利用率,降低功耗,改善性能,比较适合于产品开发单位使用。在 EDA 技术发展策略上,EDA 专业软件公司面向应用,提供 IP 模块和相应的设计服务;而半导体厂商则采取三位一体的战略,注重器件生产、设计服务和 IP 模块的提供。

1.3.3 硬件描述语言

硬件描述语言是硬件设计者和 EDA 工具之间的界面,设计者使用硬件描述语言描述自己的设计方案,并把这个描述告诉 EDA 工具,最后在 EDA 工具的帮助下进行详细设计及验证。也就是说 EDA 技术的设计语言是硬件描述语言。

采用硬件描述语言进行设计是当前发展的趋势。硬件描述语言可以对硬件电路进行行为描述、寄存器传输描述或结构化描述。它能比传统的电路原理图更有效地表示硬件电路的特性;它可借鉴高级程序语言设计的具体方法来对硬件电路的行为和功能结构进行高度抽象化的描述,从这点上看,它更适合大规模系统的设计;它可对硬件电路的设计进行不同层次、不同领域的模拟验证和综合优化等处理,从而实现硬件电路设计的高度自动化。采用硬件描述语言设计电路可提高效率,增加开发成果的可继承性,故可大大降低成本、缩短研制周期,它既适用于小批量产品开发,又适用于大批量产品的样品研制,因而得到了广泛地应用。

目前,最有代表性的硬件描述语言是美国国防部开发的 VHDL (VHDL, VHSIC Hardware Description Language) 和 Verilog 公司开发的 Verilog HDL。一般而言,两种语言的侧重点稍有不同,前者非常适合大型电子系统的描述,后者则更加适合硬件细节的描述。本书不是从理论上严格地讨论 VHDL 的正规定义,而是把它作为一种语言和设计工具,在后续章节通过

实例逐步讲述如何正确理解和使用 VHDL 进行数字电路的设计。

由于 VHDL 是一种硬件描述语言,它描述的对象始终是客观电路,因此,在学习 VHDL 语言时,有几个特别需要注意的问题。

VHDL 是一种硬件描述语言,它不同于其他的计算机语言,如 C 语言或汇编语言。采用计算机语言编程的程序设计者的思维模式一般是一维的,因为任何复杂的程序在一个单 CPU 系统的计算机中运行,永远是单向和一维的,即在 CPU 工作的任一时间段内只能完成一种操作。而 VHDL 语言则不同,它描述的对象是电路系统,而电路系统内部子系统的工作方式可以是相互独立的,也可以是互为因果的,它必须适应电路系统的这种实际工作方式,以并行和顺序的多种语句方式来描述在同一时刻中所有可能发生的事件。这就要求设计者摆脱一维的思维模式,以多维并发的思路来进行 VHDL 的程序设计。

VHDL 设计与器件无关,因为在利用 VHDL 进行设计时,可在脱离具体目标器件的情况下进行,但设计者必须清楚软件程序与硬件构成之间的关系。正因为 VHDL 的硬件描述与具体的工艺技术和硬件结构无关,VHDL 设计的硬件实现目标器件可以有宽泛的选择范围,其中包括各种系列的可编程器件。设计与器件无关是和传统硬件电路设计方法相比较而言,因为传统的设计是从选择具体的通用元器件开始的,电路的设计和实现都是围绕这些元器件进行的,这就要求设计者熟悉这些元器件,因为元器件的更换往往会导致设计重新开始,这一点在本书后续章节采用不同中规模器件实现计数器中,得到了充分体现。但设计与器件无关也并非真正无关,只是关系没有传统设计那样紧密而已,比如在设计只读存储 ROM 时,如果采用的是 LPM 宏功能模块,则设计者选用的可编程逻辑器件资源必须有 EAB/ESB 嵌入式模块,否则就会出错。还有设计者如果在设计时需要自行选择可编程逻辑器件,也需要对所选器件的资源、速度等方面有所了解,以便大致确定所选器件是否能满足设计要求。如果设计者将选择器件这一工作交给 EDA 软件工具完成,则设计也就与器件无关了。

设计者应正确理解常用语句的硬件含义。30%的基本 VHDL 语句就可以完成绝大部分的电路设计,很多生僻的语句并不能被所有的 EDA 综合软件所支持,在程序移植或者更换 EDA 软件平台时,很容易产生兼容性问题,也不利于阅读和修改。目前大部分 EDA 软件的综合能力只对 RTL 或更低层次的行为描述才能保证是可综合的,而众多初学者试图做的,却是想让 EDA 软件工具去综合算法级或者更加抽象的硬件行为描述。其实,通常 EDA 软件对 VHDL 代码的综合能力总是比人差,对于一段代码,如果设计者不能想象出一个较直观的硬件实现方法,那 EDA 软件肯定也不行。正确判断 VHDL 代码的可综合性要靠实践来积累经验,当设计者可以较准确判断代码的可综合性的时候,说明设计者对 HDL 的掌握就完全入门了。所以对初学者而言,正确理解常用 VHDL 语句的硬件含义,比多学几个新语法要有用的多。本书将在后续章节通过实例来介绍一些 VHDL 的常用语句,希望读者能认真体会其硬件含义。

1.4 数字电路与系统 EDA 设计的流程

基于 CPLD/FPGA 器件的数字电路与系统 EDA 设计流程如图 1-2 所示。

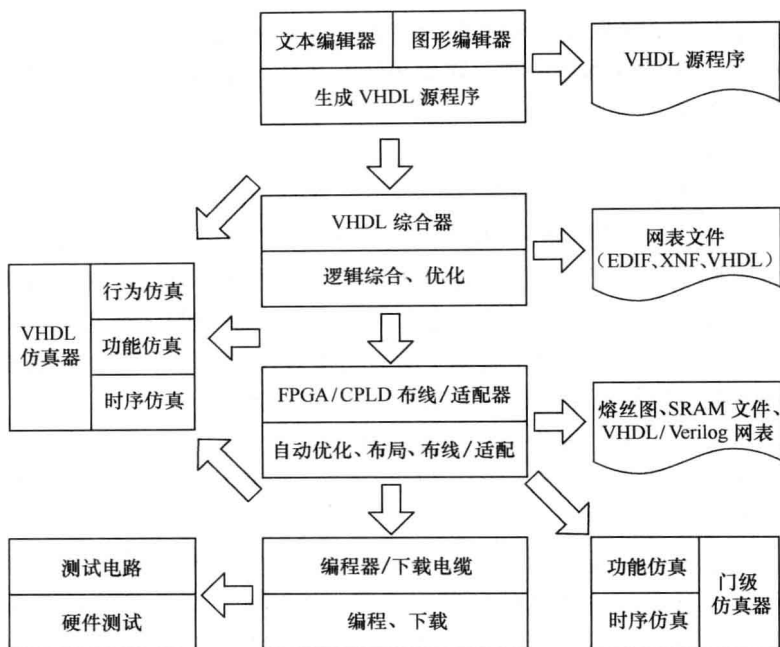


图 1-2 数字电路与系统 EDA 设计流程

1. 源程序的编辑和编译

利用 EDA 技术进行一项工程设计，首先需要将其用一定的逻辑表达手段表达出来，然后进行排错编译，变成 VHDL 文件格式，为进一步的逻辑综合做准备。EDA 工具的文本编辑器或图形编辑器允许设计者将设计用文本或者图形的方式表达出来。

2. 逻辑综合和优化

逻辑综合，就是将电路的高级语言或原理图描述转换成低级的，可与 CPLD/FPGA 或构成 ASIC 的门阵列基本结构相映射的网表文件。而逻辑映射就是针对给定硬件结构组件，将电路的高级描述进行编译、优化、转换和综合得到门级电路甚至更底层的电路描述文件。网表文件就是按照某种规定描述电路的基本组成和如何相互连接的文件。

HDL 综合器对 VHDL 源文件的综合是针对某一 CPLD/FPGA 产品系列的，综合后的结果可以被硬件系统所接受，具有硬件可实现性。需要注意的是：由于 VHDL 仿真器的行为仿真功能是面向高层次的系统仿真，只能对 VHDL 的系统描述作可行性的评估测试，不针对任何硬件系统，因此基于这一仿真层次的许多 VHDL 语句不能被 HDL 综合器所接受。不能被 HDL 综合器接受的语句在现阶段是无法在硬件系统中实现的。

3. 目标器件的布线/布局

目标器件的布线/布局又称为适配。所谓逻辑适配，就是将综合器产生的网表文件针对某一具体的目标器件进行逻辑映射操作，其中包括底层器件配置、逻辑分割、逻辑优化、布线与布局操作等，配置于指定的目标器件中，产生最终的下载文件。适配由适配器来完成，适配器又称为结构综合器。适配完成后可以利用适配所产生的仿真文件做精确的时序仿真，同时产生可用于编程的文件。适配所选定的 CPLD/FPGA 目标器件必须属于原综合器指定的目标器件系列。通常，EDA 软件中的综合器可由专业的第三方 EDA 公司提供，而适配器则需由 CPLD/FPGA 供应商自己提供，因为适配器的适配对象直接与器件结构相对应。

4. 仿真

设计的过程中有3种仿真：行为仿真、功能仿真和时序仿真。

行为仿真只是根据 VHDL 的语义进行的，与具体电路没有关系，它是将 VHDL 设计源程序直接送到 VHDL 仿真器中所进行的仿真。该仿真可以充分发挥 VHDL 中适用于仿真控制的语句及有关的预定义函数和库文件。

功能仿真仅对 VHDL 描述的逻辑功能进行测试模拟，以了解其实现的功能是否满足原设计的要求，仿真过程不涉及具体器件的硬件特性，如延时特性，它是将综合后的 VHDL 网表文件再送到 VHDL 仿真器中所进行的仿真。

时序仿真，就是将布线器/适配器所产生的 VHDL 网表文件送到 VHDL 仿真器中所进行的仿真。该仿真将器件特性加以考虑，故可得到精确的时序仿真结果。经适配处理后生成的 VHDL 网表文件中包含了较为精确的延时信息，网表文件中描述的电路结构与适配后的结果是一致的。

需要注意的是，图 1-2 中有两个仿真器，一个是 VHDL 仿真器，另一个是门级仿真器，它们都能进行功能仿真和时序仿真。所不同的是仿真用的文件格式不同，即网表文件不同。

5. 目标器件的编程/下载

把适配后生成的下载或编程文件，通过编程器或编程电缆下载到 CPLD/FPGA 中完成具体的系统设计。通常，把对 CPLD 的下载称为编程（Program）；把对 FPGA 中的 SRAM 进行直接下载的方式称为配置（Configure），但对于反熔丝结构和 Flash 结构的 FPGA 的下载和对 FPGA 的专用配置 ROM 的下载仍称为编程。

6. 硬件仿真/硬件测试

硬件仿真，就是在 ASIC 设计中，利用 CPLD/FPGA 对系统的设计进行功能检测，通过后再将其 VHDL 设计以 ASIC 形式实现的过程。硬件测试就是将含有载入了设计的硬件系统进行统一测试，根据设计项目在目标系统上的实际工作情况验证所设计的系统是否符合设计要求。

硬件仿真和硬件测试的目的，是为了在更真实的环境中检验 VHDL 设计的运行情况，特别是对于 VHDL 程序设计上不是十分规范、语义上含有一定歧义的程序。由于目标器件功能的可行性约束，综合器对于设计的理解常在一定有限范围内选择；而 VHDL 仿真器的理解是纯软件行为，其理解的选择范围要宽得多。这种理解上的偏差将会导致仿真结果和综合后实现的硬件电路在功能上的不一致。除此之外，还有许多其他的因素也会产生这种不一致。因此，进行硬件仿真和硬件测试是必不可少的。

1.5 数字电路与系统传统设计方法和 EDA 设计方法的比较

数字电路与系统传统设计方法和 EDA 设计方法的不同主要体现在以下几个方面。

1. 两者采用的设计方法不同

传统数字系统硬件设计采用自底向上的设计方法，即根据系统对硬件的要求，详细编制设计规格书，并画出系统控制流图；然后根据技术规格书和系统控制流图，对系统的功能进行细化，合理地划分功能模块，并画出系统的功能框图；接着进行各功能模块的细化和电路

设计；各功能模块电路设计、调试完成后，将各功能模块的硬件电路连接起来，再进行系统的调试，最后完成整个系统的硬件设计。采用传统方法设计数字系统，特别是当电路系统非常庞大时，设计者必须具备较多的设计经验，而且繁杂多样的原理图阅读和修改也给设计者带来诸多的不便。

数字系统 EDA 设计采用自顶向下的方法。自顶向下设计法是目前最常用的设计方法。数字系统设计分为系统级设计和逻辑级设计两个阶段。采用自顶向下的设计方法，需要先进行系统级设计，再进行逻辑级设计。

系统级设计的过程是：首先在详细了解设计任务的基础上，确定顶层系统的方案。这是设计过程的第一阶段，要求对设计任务做透彻地了解，确定设计任务及系统的整体功能、输入信号及输出信号。其次，描述系统功能，设计算法。描述系统功能就是用符号、图形、文字、表达式等形式来正确描述系统应具有的逻辑功能和应达到的技术指标。设计算法就是寻求一个解决问题的步骤，实质上是把系统要实现的复杂运算分解成一组有序进行的子运算。描述算法的工具：算法流程图、ASM 图、MDS 图等。系统级设计实质上是原理性设计，是数字系统设计的关键步骤，也是最困难的，最具有创造性的一步。

逻辑级设计的过程是：首先根据算法选择电路结构。系统算法决定电路结构。虽然不同的算法可以实现相同的系统功能，但是电路结构是不同的；相同的算法也可能对应不同的电路结构。其次，选择器件并实现电路。根据设计选择适当的器件来实现电路，并导出详细的电路图。

通过对数字系统 EDA 设计的描述可知，该方法首先从系统设计入手，在顶层进行功能划分和结构设计，采用 HDL 对高层次的系统进行描述，然后再逐级设计底层的结构。由于高层次的设计与器件及工艺无关，而且在芯片设计前就可以用软件仿真手段验证系统方案的可行性，因此自顶向下的设计方法有利于在早期发现结构设计中的错误，避免不必要的重复设计，提高设计的一次成功率。

2. 两者适用范围不同

传统数字系统设计方法适用于中、小规模电路系统设计。

数字系统 EDA 设计方法适用于高效率大规模复杂系统设计。

3. 两者设计难度不同

在传统数字系统设计中，设计者在设计电路前，对于组合电路应写出该电路的逻辑表达式或真值表；对于时序电路应写出电路的状态表。而这一工作是相当困难的，尤其是系统比较复杂时更是如此。

数字系统 EDA 设计中，往往采用硬件描述语言设计电路，这就无需写出相关电路的逻辑表达式或状态表，从而降低了设计难度，大大缩短了设计周期。

4. 两者的设计文件不同

传统数字系统硬件设计形成的设计文件是详细标注了各逻辑器件名称和相互间信号连接关系的若干张原理图。原理图的多少根据系统的大小和复杂程度而定。有些大而复杂系统的原理图有几十万张，如此多的原理图给归档、阅读、修改和使用都带来了极大的不便。

数字系统 EDA 设计方法主要的设计文件是 HDL 硬件描述语言编写的源程序。如果需要也可以转换成原理图形式输出。硬件描述语言既是硬件描述语言，同时也是文档型语言，用其编写的源程序作为归档文件的好处是：资料量小，便于保存；移植能力强；阅读方便，阅

读程序比阅读原理图容易。

5. 两者选用的器件不同

在传统数字系统硬件设计中,设计者总是根据系统的具体需要,选择市场上能买到的逻辑元器件,来构成所设计的逻辑电路,从而完成系统的硬件设计。尽管随着微处理器的出现,在由微处理器及其相应硬件构成的系统中,系统的许多硬件功能可用软件系统来完成,从而在较大程度上简化了系统硬件电路的设计,但是,这种选择通用元器件构成系统硬件电路的方法并未改变。传统数字系统硬件设计,所需的元器件种类复杂繁多,如果暂时无法购得,则须重新改动设计。随着可编程器件的广泛应用,通用集成器件逐渐淡出市场,某些十几年前可轻易购得的器件,如今已很难购得且价格提升幅度大。

在数字系统 EDA 设计中,设计者可将设计的硬件电路用可编程逻辑器件实现,或者根据需要设计自用的 ASIC 芯片,而无需受到只能使用通用元器件的限制。

6. 两者电路制作的难度不同

采用传统数字系统硬件设计方法,电路的实现可选用面包板或印制电路板。如果在面包板上实现电路,周期稍短,但只适合复杂程度低的系统,且容易出错。若选用印制电路板实现电路,则难度大且周期长。对于多层板图的电路板制作更是一个复杂费时的过程。

在数字系统 EDA 设计中,设计重点已从如何实现设计转向如何设计,由于分工细化,设计者无需过多地干预实现的细节,如电路板的设计、制作等电路实现方面的问题。

7. 两者调试阶段不同

在传统数字系统硬件设计中,调试只有等到硬件电路组装完成之后才能进行。这样做的结果是,系统设计时存在的问题只有在后期才能较容易发现,一旦考虑不周,系统设计有可能重新开始,使得设计成本和周期都大大增加。

采用数字系统 EDA 设计方法,在系统设计过程中要进行三级仿真,即系统数学模型的仿真、系统数据流的仿真和系统门电路电原理的仿真。并且这三级仿真贯穿系统硬件设计的全过程,这样做,便于在系统设计的早期发现问题,从而大大缩短了设计周期,降低了成本。

8. 两者对设计者的硬件理论知识和经验要求不同

在传统数字系统硬件设计中,对设计者要求较高。设计者在电子技术理论和设计实践方面必须是行家里手。设计者必须熟悉不同器件的性能和使用方法;必须知道器件的封装形式和电气特性;必须懂得不同在线测试仪表的使用方法和性能指标;必须熟练掌握大量与设计理论和优化技术毫无关系的技能、技巧。这些显然既不符合现代电子技术发展的需求,又不能适应快速更新换代的产品市场要求。

采用数字系统 EDA 设计方法时,由于 EDA 技术的标准化和 HDL 设计语言与设计平台对具体硬件设计的无关性,使设计者能更大程度地将自己的才智和创造力集中在设计项目性能的提高和成本的降低上,而将更具体的硬件实现工作交给专门部门完成。

第2章 数字电路不同描述方法的EDA实现

当用 EDA 工具综合一个数字电路时，设计者可以用多种方法提供电路原始描述，如原理图、硬件描述语言、波形图等。

本章举例说明几种数字电路不同描述方法的 EDA 实现，这里的 EDA 工具选用 Altera 公司的 Quartus II 7.2 开发软件。

2.1 数字电路原理图输入设计方法的 EDA 实现

2.1.1 用原理图描述数字电路

数字电路原理图是指用电路元件的图形符号及连线表示电路逻辑功能的图形。设计者可在分析设计要求的基础上，通过设计逐步得到原理图。

这里通过一个简单的例子说明如何用电路原理图描述数字电路。

例 2-1 设计一个半加器，要求能够完成两个一位二进制数的半加运算。

解：第一步，设变量。

设 A、B 为半加器的两个输入变量，代表输入的两个一位二进制数；S 和 C_o 为输出变量，分别代表和以及进位。

第二步，根据题意列真值表。

半加器的真值表如表 2-1 所示。

表 2-1 半加器真值表

A	B	C_o	S
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

第三步，写表达式。

由真值表可得进位及和的表达式： $S = A \oplus B$ $C_o = AB$

第四步，画逻辑电路图。

将表达式中的运算符号用电路符号表示，就可根据表达式画出半加器的电路原理图，如图 2-1 所示。

例 2-2 利用例 2-1 设计的半加器和或门设计一个全加器。

解：第一步，设变量。

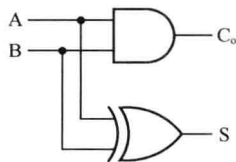


图 2-1 半加器原理图

设 A 、 B 、 C_i 为半加器的 3 个输入变量，代表输入的两个一位二进制数和低位来的进位； S 和 C_o 为输出变量，分别代表和以及向高位的进位。

第二步，根据题意列真值表。

全加器的真值表如表 2-2 所示。

第三步，写表达式。

表 2-2 全加器真值表

A	B	C_i	C_o	S
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

由真值表可得和及进位的表达式： $S = A \oplus B \oplus C_i$ $C_o = AB + C_i(A \oplus B)$

第四步，画逻辑电路图。

将全加器的表达式和半加器的做比较，可得全加器的电路原理图，如图 2-2 所示。

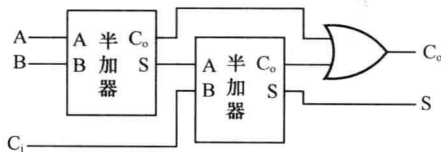


图 2-2 全加器原理图

2.1.2 原理图输入设计方法的 EDA 实现流程

1. Quartus II 软件设计流程

EDA 工具选用 Altera 公司的 Quartus II 7.2 开发软件，如无特殊说明，本书以后都将 Quartus II 7.2 简称为 Quartus II。下面首先对该软件进行简要说明。

Quartus II 是 Altera 公司在 21 世纪初推出的 CPLD/FPGA 集成开发环境，它是该公司前一代 CPLD/FPGA 集成开发环境 MAX+PLUS II 的更新换代产品。Quartus II 提供了一种与结构无关的设计环境，其界面友好，使设计者能方便地进行设计输入、快速处理和器件编程。

Quartus II 提供了完整的多平台设计环境，能满足各种特定设计的需要。Quartus II 是单片可编程系统（SOPC）设计的综合性环境和 SOPC 开发的基本设计工具；Quartus II 与 MATLAB 和 DSP Builder 结合，可以进行基于 FPGA 的 DSP 系统开发，是 DSP 硬件系统实现的关键 EDA 工具。Quartus II 可以直接利用第三方的综合工具，如 Leonardo Spectrum，并能直接调用这些工具。Quartus II 具备仿真功能，同时也支持第三方的仿真工具，如 ModelSim。

Quartus II 包括模块化的编译器。编译器所包含的功能模块有分析/综合器、适配器、装配