

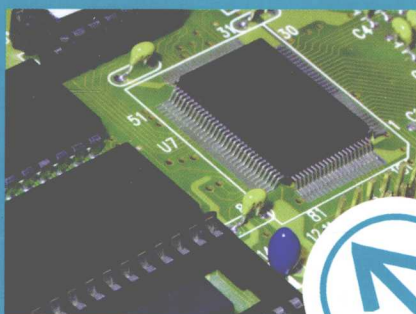
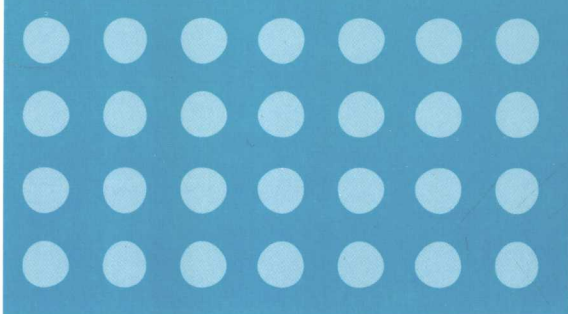


高等学校“十二五”重点规划教材
通信与电子信息工程系列

Shuzi Jicheng Dianlu Yu
EDA Sheji Jichu Jiaocheng

数字集成电路与 EDA设计基础教程

陶佰睿 李静辉 苗凤娟 马 丽 主编



HEUP 哈尔滨工程大学出版社
Harbin Engineering University Press

内 容 简 介

本书共分九章。以基于 Altera 公司 CyconeII 芯片的 DE2 开发板为平台,以 QuartusII, Active - HDL 和 Modelsim EDA 软件为设计工具,全面阐述了 EDA 技术概况、数字集成电路设计基本过程、Verilog 语法规则及数字电路设计案例。

本书以有序启动项目教学法为指导,以工程训练为核心,以培养学生实践能力为目标,着眼提高学生对社会生产实践的适应能力,夯实基础,发展学生创新精神。

本书可作为电子信息类专业的高年级本科生和硕士研究生的阅读材料,也可作为其他工程技术人员和教师系统学习 EDA 技术和 IC 设计的参考书。

图书在版编目(CIP)数据

数字集成电路与 EDA 设计基础教程 / 陶佰睿等主编.
—哈尔滨: 哈尔滨工程大学出版社, 2012.7
ISBN 978 - 7 - 5661 - 0409 - 0

I. ①数… II. ①陶… III. ①数字集成电路 - 电路设计 - 计算机辅助设计 - 高等学校 - 教材 IV. ①TN431.202

中国版本图书馆 CIP 数据核字(2012)第 165134 号

出版发行	哈尔滨工程大学出版社
社 址	哈尔滨市南岗区东大直街 124 号
邮政编码	150001
发行电话	0451 - 82519328
传 真	0451 - 82519699
经 销	新华书店
印 刷	肇东市一兴印刷有限公司
开 本	787mm × 1 092mm 1/16
印 张	15.25
字 数	372 千字
版 次	2012 年 7 月第 1 版
印 次	2012 年 7 月第 1 次印刷
定 价	32.00 元

<http://press.hrbeu.edu.cn>
E-mail: heupress@hrbeu.edu.cn

前 言

全书内容按照数字集成电路及 EDA 的层次化设计方法和知识模块组织,分为两大部分:第一部分“理论与实践”主要介绍数字集成电路电子设计自动化(EDA)技术基础、电子系统设计与电子组装、系统设计与仿真、电路级设计与仿真、Verilog HDL, PLD, FPGA 与 SOPC 的 IC 设计技术等;第二部分以生产实践中的项目案例为主要内容,主要以 Active - HDL, QuartusII 和 ModelSim 工具软件为核心,介绍工程实践中如组合电路、时序电路、交通灯控制器、有限状态机、快速乘法器、FIR 滤波器以及典型数字系统的设计等。

本书内容的编排充分地考虑了高校的教学需求、平台成本和学生的层次,整合了 EDA 和 IC 设计的教学体系,有利于教师主导作用和学生主体作用的发挥,可灵活进行项目有序启动案例教学。

本书第 1,2,6,7 章由陶佰睿编写,第 3,4,5 和 9 章由李静辉编写,第 9 章案例由马丽完成,第 8 章和附录部分由苗凤娟编写。在此要特别感谢姚仲敏教授和刘景军教授的鼓励和热心帮助,他们对本书讲义在教学应用中出现问题提出了很多宝贵意见,对书稿的完成帮助巨大。同时感谢孙艳梅老师、董亮老师和刘道森老师,以及王锐和高万峰同学对本书校对和图形绘制等工作的大力支持,本书在编写过程中,参考或引用了一些专家学者的论著,在此一并表示感谢。

由于编者水平和实践经验还有待进一步提高,书中错误和疏漏在所难免,恳请广大读者批评指正。

作者电子邮箱:Tbr sir@163. com

编 者
2011.09

目 录

第 1 章 绪论	1
1.1 EDA 技术概述	1
1.2 数字 IC 的 EDA 设计流程	3
1.3 硬件描述语言	6
1.4 Verilog HDL 的主要功能	8
1.5 Verilog HDL 的设计方法	9
1.6 可编程器件概述	11
第 2 章 Verilog HDL 基础知识	13
2.1 模块的基本结构及其描述方式	13
2.2 Verilog HDL 设计流程	23
2.3 Verilog HDL 语法	23
2.4 数据类型	28
2.5 运算符	34
2.6 系统任务与系统函数	43
2.7 编译预处理	49
第 3 章 Verilog HDL 行为建模	55
3.1 模块的行为描述	55
3.2 赋值语句	56
3.3 语句块	57
3.4 条件语句	60
3.5 循环语句	64
3.6 过程说明语句	68
3.7 时间控制	71
3.8 过程连续赋值语句	89
3.9 任务和函数说明语句	94
3.10 行为建模实例	99
第 4 章 Verilog HDL 数据流建模	102
4.1 连续赋值语句	102
4.2 数据流建模具体实例	106
第 5 章 Verilog HDL 结构建模	108

5.1 模块定义结构	108
5.2 模块端口	108
5.3 实例化语句	109
5.4 结构化建模具体实例	112
第6章 数字 IC 单元设计举例	114
6.1 基于 Verilog HDL 组合逻辑电路设计	114
6.2 基于 Verilog HDL 时序逻辑电路设计	126
6.3 基于 Verilog HDL 移位寄存器设计	131
6.4 基于 Verilog HDL 数字模块设计	135
6.5 利用有限状态机进行复杂时序逻辑的设计	137
6.6 利用状态机的嵌套实现层次结构化设计	140
6.7 通过模块之间的调用实现自顶向下的设计	144
第7章 程序验证	149
7.1 测试验证程序的编写	149
7.2 波形产生	149
7.3 测试验证程序实例	154
7.4 从文本文件中读取向量	157
7.5 向文本文件中写入向量	159
7.6 时序检测器	160
第8章 综合设计实例	162
8.1 有限状态机的概念及其设计实例	162
8.2 RISC 中央处理单元(CPU)的顶层设计	168
第9章 FPGA 设计实例	178
9.1 Altera DE2 开发板的结构	178
9.2 Altera DE2 开发板接口定义	180
9.3 FPGA 芯片配置	180
9.4 基于 DE2 开发板交通灯设计案例	182
附录 A Verilog HDL 设计规范	198
附录 B Active HDL	202
附录 C Quartus II	213
附录 D Modelsim	231
参考文献	238

绪 论

随着 IT 产业日新月异的发展,集成电路(IC,Integrate Circuit)工艺特征线宽已经从微米发展到深亚微米。同时,电子产品对低功耗、微型化、可集成一体化的要求日趋强烈,促进人类科技进入纳米时代。当前,32 nm 制程的 IC 芯片已经开始市场供应,22 nm 工艺节点已经启动,14 nm 工艺正在研发,预计到 2020 年,4 nm IC 工艺将出现。传统的电子线路设计方法远远不能满足科研及社会发展的需要。电子设计已经从分立元件、集成电路、中大规模集成电路、超大规模集成电路以及专用集成电路的设计向电子系统可集成设计的方向发展,新方法、新器件、新工具不断涌现。数字集成电路的设计也越来越依靠电子设计自动化(Electronic Design Automation,EDA)技术的进步。本章将从数字集成电路的设计方法、IC 领域 EDA 技术的发展展开论述。

1.1 EDA 技术概述

EDA(Electronic Design Automation)即“电子设计自动化”,是指以计算机为工作平台,以 EDA 软件为开发环境,以硬件描述语言为设计语言,以可编程器件 PLD 为实验载体(包括 CPLD,FPGA,EPLD 等),以集成电路芯片为目标器件的电子产品自动化设计过程。

1.1.1 电子系统设计方法的演变过程

信息社会的发展以信息产业的进步为动力,而信息产业的核心是集成电路技术。现代集成电路在性能提高、复杂度增加的同时,不仅价格呈下降趋势,而且产品更新换代的频率也越来越快。实现这种进步的主要原因是集成电路的生产制造技术和设计技术的快速发展。制作工艺的进步和 EDA 技术的发展是相辅相成的,没有 EDA 技术的支持,要完成上述超大规模集成电路的设计制造是不可想象的;反之,生产制造技术的不断进步又必将对 EDA 技术提出更高的要求。从历史上看,EDA 技术分为以下三个阶段。

1. AD 阶段(20 世纪 60 年代~20 世纪 80 年代初期)

计算机辅助设计(Computer Aided Design,CAD)阶段主要是一些单机工具软件,如:完成 PCB(Printed Circuit Board)布线设计、电路模拟、逻辑模拟及版图的绘制等功能的软件包。这一阶段人们已经开始用计算机辅助进行 IC 版图编辑、PCB 布局布线等来取代了手工操作,产生了计算机辅助设计的概念。

20 世纪 80 年代初,由于集成电路规模越来越大,制作也趋于复杂,EDA 技术有了较快的发展,出现了以 Mento 为代表的专业 EDA 软件公司。这个时期的软件主要针对产品开发过程的辅助设计,分为设计、分析、生产、测试等多个独立的软件包,每个软件包一般只能完成其中的一项工作。其主要缺点是:第一,各软件工具是由多个公司独立开发的,只能解决某个领域中的局部问题,影响设计的跨平台性和兼容性;第二,对于复杂电子系统的设计,当时的 EDA 工具不能够提供系统级的仿真与综合,产品研发速度受到很大影响。

2. E 阶段(20 世纪 80 年代初期~20 世纪 90 年代初期)

计算机辅助工程设计(Computer Aided Engineering, CAE)阶段主要解决在集成电路与电子系统设计方法学以及设计工具集成化方面的问题。期间,各种设计工具,如原理图输入、编译与链接、逻辑模拟、测试码生成、版图自动布局以及各种单元库均得到丰富。由于采用了统一的数据管理技术,因而能够将各个工具集成为一个 CAE 系统。运用这种系统,按照设计方法学制定的某种设计流程,可以由寄存器传输级(Register Transfer Level, RTL)开始,实现从设计输入到版图输出的全过程设计自动化。CAE 阶段中主要采用基于单元库的半定制设计方法。采用门阵列和标准单元设计方法设计的专用集成电路(Application Specific Integrated Circuits, ASIC)得到了极大的发展,将集成电路工业推进到了 ASIC 时代。其中部分 CAE 系统中还集成了 PCB 自动布局布线软件以及热特性、噪声、可靠性等分析软件,来促进电子系统设计自动化水平。

3. EDA 阶段(20 世纪 90 年代后期)

20 世纪 90 年代开始,微电子技术迅猛发展,其工艺水平达到深亚微米级,在一个芯片上已经可集成数百万到上千万只晶体管,其工作速度可以达到 Gb/s 水平,这为制造出规模更大、速度和信息容量更高的芯片系统提供了基础条件,同时也对 EDA 系统提出了更高的要求。尽管 CAD/CAE 技术取得了巨大的成功,但在整个设计过程中,自动化和智能化程度还不高,各种 EDA 软件操作界面千差万别,学习使用困难,并且它们的互不兼容性直接影响到设计环节间的衔接。基于以上不足,人们开始追求贯彻整个设计过程的自动化,这就是电子系统设计自动化(Electronic System Design Automation, ESDA)。它代表了当今电子设计技术的最新发展方向,设计人员按照“自顶向下”的设计方法,对整个系统进行方案设计和功能划分,系统的关键电路系统用一片或几片 ASIC 实现,然后采用硬件描述语言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重复性工作,使他们能将精力集中于创造性方案和创新概念的构思上。

在第三代 EDA 系统中,引入硬件描述语言和行为综合与逻辑综合工具,并采用较高的抽象层次进行设计。按层次式方法进行管理,提高了处理复杂设计的能力,并大幅度缩短了设计周期。另外,采用专用的综合优化工具,使芯片的品质如面积、速度和功耗等获得了优化。其发展趋势是从物理级、电路级和系统级的设计向单芯片系统级设计方向过渡。

1.1.2 EDA 工具的分类

FPGA, PLD, CPLD 及 IC 设计开发领域的 EDA 工具,通常包含仿真器(Simulator)、综合

器(Synthesizer)和配置器(Place and Routing, P&R)等专用软件包,狭义上来讲这一领域的EDA工具就不包括Protel, PSpice, Ewb等原理图和PCB版图设计及电路仿真软件。在数字IC设计领域,目前流行的EDA工具软件有两种分类方法:一种是按公司类别进行分类,另一种是按功能进行划分。

按公司类别分,大体可分两类:一类是EDA专业软件公司,业内最著名的三家公司是Cadence, Synopsys和Mentor Graphics;另一类是PLD器件厂商为了销售其产品而开发的EDA工具,较著名的公司有Altera, Xilinx, Lattice等。前者独立于半导体器件厂商,具有良好的标准化和兼容性,适合于学术研究单位使用,但系统复杂、难于掌握且价格昂贵;后者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,比较适合产品开发单位和初学者使用。

按功能分,EDA工具大体可分为以下三类。

1. 集成的PLD/FPGA开发环境

由半导体公司提供,可以完成设计输入(原理图或HDL)→仿真→综合→布线→下载到器件等所有PLD开发流程所需工作。如Altera公司的Maxplus II, Quartus II, Xilinx公司的ISE, Lattice公司的ispDesignEXPERT等。其优势是功能丰富,软件包全集成化,可以加快动态调试,缩短开发周期;缺点是在综合和仿真环节与专业的软件相比,在性能上稍逊一筹。

2. 综合类

这类软件的功能是对设计输入进行逻辑分析、综合和优化,将硬件描述语句(通常是系统级的行为描述语句)翻译成最基本的与或非门的连接关系(网表),导出给PLD/FPGA厂家的软件进行布局和布线。为了优化结果,在进行较复杂的设计时,基本上都使用这些专业的逻辑综合软件,而不采用厂家提供的集成PLD/FPGA开发工具。如Synplify公司的Synplify, Synopsys公司的FPGAexpress, FPGA Compiler II等。

3. 仿真类

这类软件的功能是对设计进行模拟仿真,包括布局布线前的“功能仿真”(“前仿真”)和包含门延时和线延时等的“时序仿真”(“后仿真”)。相对同样的设计输入,专业软件的仿真速度比集成环境的速度快得多。此类软件最知名的有Model Technology公司的Modelsim, Cadence公司的NC-Verilog/NC-VHDL/NC-SIM等。

上述典型的EDA工具在性能上各有所长,有的综合优化能力突出,有的仿真模拟功能强,多数工具能相互兼容,具有互操作性。如Altera公司的QuartusII集成开发工具,就支持多种第三方的EDA软件,用户可以在QuartusII软件中通过设置直接调用Modelsim和Synplify进行仿真和综合。如果设计的硬件系统不是很大,对综合和仿真的要求不是很高,那么可以在一个集成的开发环境中完成整个设计流程。如果要进行复杂系统的设计,则常规的方法是多种EDA工具协调工作,集各家之所长来完成设计流程。

1.2 数字IC的EDA设计流程

基于FPGA/CPLD的数字IC设计一般流程如图1.1所示。主要包含设计输入、逻辑功

能仿真(前仿真)、逻辑综合、时序功能仿真(后仿真)以及布局布线等。这里逻辑仿真器主要指 modelsim, Verilog-XL 等,逻辑综合器主要指 LeonardoSpectrum, Synplify, FPGA Express/FPGA Compiler 等,FPGA 厂家工具指 Altera 的 Max + PlusII, QuartusII, Xilinx 的 Foundation, Alliance, ISE 等。

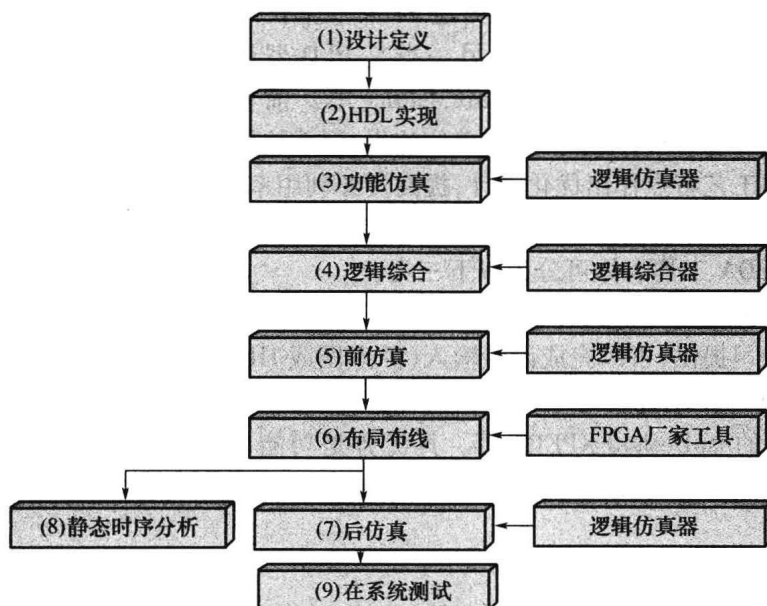


图 1.1 基于 FPGA/CPLD 的数字 IC 设计一般流程

1. 逻辑功能仿真

对完成的 IC 设计文档的逻辑功能仿真如图 1.2 所示。主要包含:调用模块的行为仿真模型、RTL 代码、测试程序(test bench)以及仿真后的测试数据包等。这里“调用模块的行为仿真模型”指的是 RTL 代码中引用的由厂家提供的宏模块或 IP 核,如 Altera 提供的 LPM 库中的乘法器、存储器等部件的行为模型。

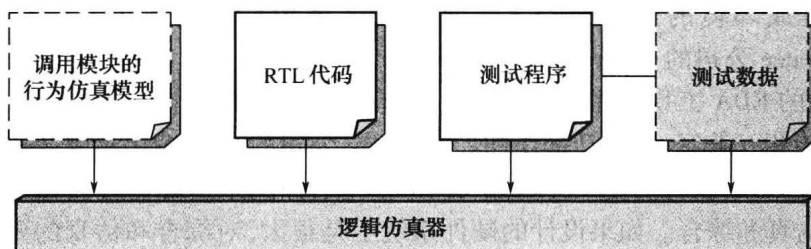


图 1.2 逻辑功能仿真

2. 逻辑综合

逻辑综合是在标准单元库和特定的设计约束的基础上,把设计的高层次描述转换成优化的门级网表的过程。标准单元库可以包含简单的单元,例如,与门、或门和或非门等基本逻辑门,或者宏单元;例如,加法器、多路选择器和特殊的触发器。标准单元库一般指 IC 制

造厂商的工艺库。逻辑综合基本流程如图 1.3 所示。这里“调用模块的黑盒子接口”的导入,是由于 RTL 代码调用了一些外部模块,而这些外部模块不能被综合或无需综合,但逻辑综合器需要其接口的定义来检查逻辑并保留这些模块的接口。

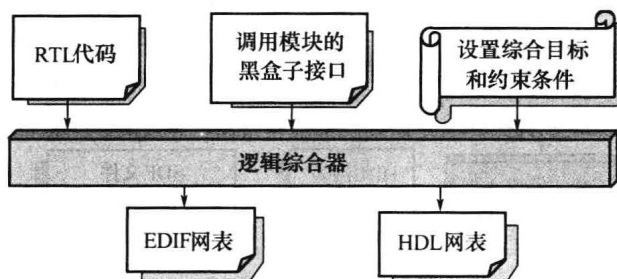


图 1.3 逻辑综合基本流程

3. 前仿真(功能仿真)

前仿真指功能仿真,主旨在于验证电路的功能是否符合设计要求,其特点是不考虑电路门延迟与线延迟。基本与设计器件的物理参数无关。一般来说,对 FPGA 设计这一步可以跳过不做,但可用于 debug 综合有无问题。前仿真大多使用 Modelsim 平台来完成。其一般流程如图 1.4 所示。

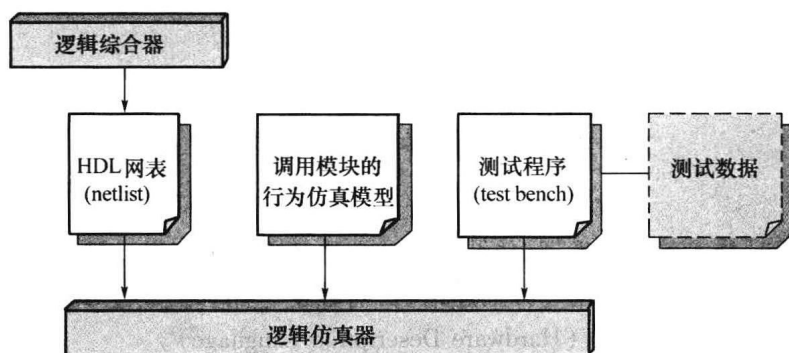


图 1.4 前仿真一般流程

4. 布局布线

布局布线指根据用户约束和物理约束,对设计模块进行实际的布局,并根据设计连接,对布局后的模块进行布线,产生 FPGA/CPLD 配置文件。一般的布局布线流程如图 1.5 所示。

5. 后仿真(时序仿真)

后仿真也就是综合后的功能仿真和布局布线后的时序仿真,是指电路已经映射到特定的工艺环境下,综合考虑电路的路径延迟与门延迟的影响,与使用器件的物理特性密切相关。后仿真是关于验证电路是否在一定时序条件下满足设计要求、是否存在时序违规的过程,不能被省略。一般后仿真多采用 Modelsim 平台进行,其一般流程如图 1.6 所示。

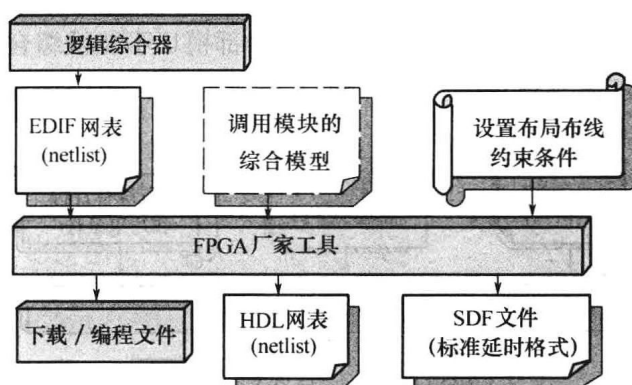


图 1.5 布局布线流程

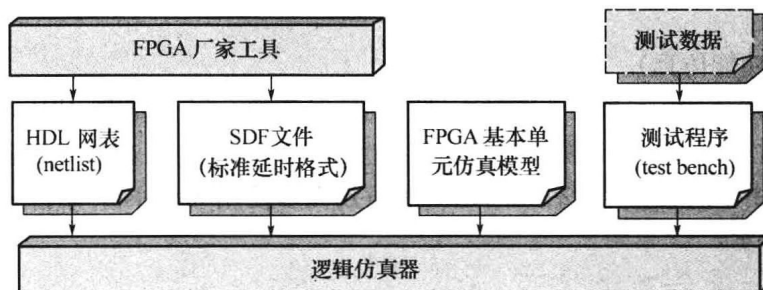


图 1.6 后仿真一般流程

1.3 硬件描述语言

随着电子设计技术的飞速发展,传统的设计方法已满足不了设计的要求,因此要求能够借助当今先进的 EDA 工具,使用一种描述语言,对数字电路和数字逻辑系统能够进行形式化的描述,这就是硬件描述语言(Hardware Description Language)。

1.3.1 HDL 概述

目前,电子设计的规模越来越大,复杂度越来越高,20 世纪 90 年代末普通设计的规模已经达到百万门的数量级。当前 Intel 公司的商用 Core i7 CPU 集成了 7.31 亿只晶体管。为使如此复杂的芯片变得易于被人脑理解,很有必要用一种高级语言来表达其功能,隐藏其具体实现的细节。这也就是在大系统程序编写中高级程序设计语言取代汇编语言的原因。同样,在芯片设计中也不得不使用硬件描述语言,而具体实现工作交给逻辑综合工具完成。

硬件描述语言(HDL)是一种用形式化方法来描述数字电路和设计数字逻辑系统的语言。是硬件设计人员和 EDA 工具之间的桥梁,它主要用于从算法级、门级到开关级的多种抽象设计层次的数字系统建模。被建模的数字系统对象的复杂性可以介于简单的门和完整的电子数字系统之间。它可以使数字逻辑电路设计者利用这种语言来描述自己的设计思

想,然后利用电子设计自动化(在下面简称为 EDA)工具进行仿真,再自动综合到门级电路,再用 ASIC 或 FPGA 实现其功能。目前,这种称之为高层次设计(High Level Design)的方法已被广泛采用。

硬件描述语言发展至今已有 20 多年历史,当今业界的标准中(IEEE 标准)主要有 VHDL 和 Verilog HDL 这两种硬件描述语言。

1.3.2 HDL 的发展历史

1962 年,Iverson 公司提出 HDL 概念,到目前为止已经出现了多种 HDL 产品,它们中的一部分主要是从 C 语言发展而来的,而另一部分是以 PASCAL 语言为基础的。当前,以 C 语言为基础的 Verilog HDL 和 VHDL 成为业界应用的主流。

Verilog HDL 语言是一种专用硬件建模语言,是由 Gateway Design Automation 公司于 1983 年为其模拟器产品开发的。由于 GDA 公司的模拟、仿真器产品的广泛使用,Verilog HDL 作为一种易于使用且实用的硬件描述语言逐渐为众多设计者所接受。1989 年,GDA 公司被 Cadence 公司并购。1990 年,Cadence 公司正式发布 Verilog HDL 语言,并成立了 Open Verilog International (OVI)这一促进 Verilog 发展的国际性组织。1995 年,Verilog HDL 语言成为 IEEE 标准,称为 IEEE Std 1364—1995。

1.3.3 Verilog HDL 与 VHDL 的比较

Verilog HDL 和 VHDL 都是用于逻辑设计的硬件描述语言,并且都已成为 IEEE 标准。VHDL 在 1987 年成为 IEEE 标准,Verilog HDL 在 1995 年才正式成为 IEEE 标准。VHDL 比 Verilog HDL 早成为 IEEE 标准是因为 VHDL 是美国军方组织开发的,而 Verilog HDL 则是从一个普通公司的私有财产转化而来,基于 Verilog HDL 的优越性,才成为 IEEE 标准,因而有更强的生命力。

VHDL 其英文全名为 VHSIC Hardware Description Language,而 VHSIC 则是 Very High Speed Integrated Circuit 的缩写词,意为甚高速集成电路,故 VHDL 其准确的中文译名为甚高速集成电路的硬件描述语言。

Verilog HDL 和 VHDL 作为描述硬件电路设计的语言,其共同的特点在于:能形式化地抽象表示电路的结构和行为,支持逻辑设计中层次与领域的描述,可借用高级语言的精巧结构来简化电路的描述,具有电路仿真与验证机制以保证设计的正确性、支持电路描述由高层到低层的综合转换,硬件描述与实现工艺无关(有关工艺参数可通过语言提供的属性包括进去),便于文档管理、易于理解和设计重用。但是 Verilog HDL 和 VHDL 又各有其自己的特点。它们主要有以下几点不同。

(1)从推出过程来看,VHDL 偏重于标准化的考虑,而 Verilog HDL 与 EDA 工具的结合更为紧密。VHDL 是国际上第一个标准化的 HDL 语言(IEEE—1076),是为了实现美国国防部 VHSIC 计划所推出的各个电子部件供应商具有统一数据交换格式的要求。相比之下,Verilog HDL 则是在全球最大的 EDA/ESDA 供应商 Cadence 公司的扶持下针对 EDA 工具开发的 HDL 语言。

(2)与 VHDL 相比,Verilog HDL 的编程风格更加简洁高效。如果单纯从描述结构上看,

两者的代码之比为 3:1。

(3) 目前市场上所有 EDA/ESDA 工具都同时支持这两种语言,而在 ASIC 设计领域,Verilog HDL 由于历史原因在 IC 设计领域具有较多的设计资源。

一般认为 Verilog HDL 在系统级抽象方面比 VHDL 略差一些,而在门级开关电路描述方面比 VHDL 强得多。Verilog HDL 和 VHDL 建模能力的比较如图 1.7 所示。

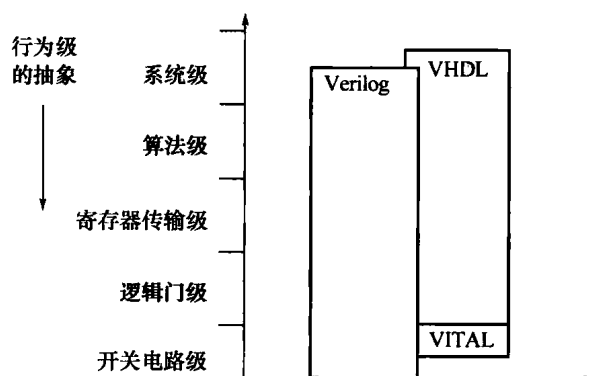


图 1.7 Verilog HDL 和 VHDL 建模能力比较

Verilog HDL 较为适合系统级 (System)、算法级 (Alogrithem)、寄存器传输级 (RTL)、逻辑级 (Logic)、门级 (Gate)、电路开关级 (Switch) 设计,而对于特大型 (几百万门级以上) 的系统级 (System) 设计,则 VHDL 更为适合,由于这两种 HDL 语言还在不断地发展过程中,它们都会逐步地完善自己。

1.4 Verilog HDL 的主要功能

用 Verilog HDL 描述的电路设计就是该电路的 Verilog HDL 模型。Verilog 模型可以是实际电路的不同级别的抽象。这些抽象的级别和它们对应的模型类型共有以下五种。

- (1) 系统级 (System): 用高级语言结构实现设计模块的外部性能的模型。
- (2) 算法级 (Algorithm): 用高级语言结构实现设计算法的模型。
- (3) RTL 级 (Register Transfer Level): 描述数据在寄存器之间流动和如何处理这些数据的模型。
- (4) 门级 (Gate-level): 描述逻辑门以及逻辑门之间连接的模型。
- (5) 开关级 (Switch-level): 描述器件中三极管和储存节点以及它们之间连接的模型。

一个复杂电路系统的完整 Verilog HDL 模型是由若干个 Verilog HDL 模块构成的,每一个模块又可以由若干个子模块构成。其中有些模块需要综合成具体电路,而有些模块只是与用户所设计的模块交互的现存电路或激励信号源。利用 Verilog HDL 语言结构所提供的这种功能就可以构造一个模块间的清晰层次结构来描述极其复杂的大型设计,并对所作设计的逻辑电路进行严格的验证。

Verilog HDL 作为一种结构化和过程性的行为描述语言,它的语法结构极其适合于算法

级和 RTL 级的模型设计。这种描述语言具有以下功能。

- (1) 可描述顺序执行或并行执行的程序结构。
- (2) 用延迟表达式或事件表达式来明确地控制过程的启动时间。
- (3) 通过命名的事件来触发其他过程里的激活行为或停止行为。
- (4) 提供了条件、if-else、case、循环程序结构。
- (5) 提供了可带参数且非零延续时间的任务(task)程序结构。
- (6) 提供了可定义新的操作符的函数结构(function)。
- (7) 提供了用于建立表达式的算术运算符、逻辑运算符、位运算符。

Verilog HDL 语言作为一种结构化的语言也非常适合于门级和开关级的模型设计。因其结构化的特点又使它具有以下功能:

- (1) 提供了完整的一套组合型原语(primitive)。
- (2) 提供了双向通路和电阻器件的原语。
- (3) 可建立 MOS 器件的电荷分享和电荷衰减动态模型。

Verilog HDL 的构造性语句可以精确地建立信号的模型。这是因为在 Verilog HDL 中, 提供了延迟和输出强度的原语来建立精确程度很高的信号模型。信号值可以有不同的强度, 可以通过设定宽范围的模糊值来降低不确定条件的影响。

Verilog HDL 有着类似 C 语言的风格。其中有许多语句如高级程序语句: if 语句、case 语句等和 C 语言中的对应语句十分相似。在学习 C 语言编程的基础上进行学习 VerilogHDL 并不困难, 我们只需对 Verilog HDL 某些语句的特殊方面着重理解, 并加强应用练习就能很好地掌握这门技术, 并利用它的强大功能来设计复杂的数字逻辑电路。

1.5 Verilog HDL 的设计方法

Verilog HDL 的设计方法主要有两种: 自底向上(Bottom-Up)的设计方法与自顶向下(Top-Down)的设计方法。另外, 还可根据实际情况, 利用这两种方法的组合进行综合设计, 即综合设计方法。

现代集成电路制造工艺技术的改进, 使得在一个芯片上集成数十乃至数百万个器件成为可能, 但我们很难设想仅由一个设计师独立设计如此大规模的电路而不出现错误。利用层次化、结构化的设计方法, 一个完整的硬件设计任务首先由总设计师划分为若干个可操作的模块, 编制出相应的模型(行为的或结构的), 通过仿真加以验证后, 再把这些模块分配给下一层的设计师, 这就允许多个设计者同时设计一个硬件系统中的不同模块, 其中每个设计者负责自己所承担的部分; 而由上一层设计师对其下层设计者完成的设计用行为级上层模块对其所做的设计进行验证。

1.5.1 自底向上(Bottom-Up)的设计方法

自底向上(Bottom-Up)的设计方法是一种传统的电子系统设计方法, 它的主要步骤是: 首先根据系统要求编制技术规格书, 并画出系统控制流程图; 然后依照技术规格书和系统控制流程图, 对系统的功能进行细化, 合理划分功能模块, 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;当各功能模块电路设计、调试完成后,将各功能模块连接起来进行全系统的调试,其流程如图 1.8 所示。

自底向上的设计方法常用于原理图设计中,其优点是:

- (1)它是分立器件设计人员比较熟悉的设计方法;
- (2)实现各子模块电路所需的时间较短。

但这种设计方法也有许多不足之处,具体表现为:

- (1)容易造成对系统的整体功能把握不足;
- (2)由于设计中必须先设计好各个小模块,所以整个系统的实现需要较长的时间;
- (3)这种方法对设计人员之间相互进行协作有较高的要求。

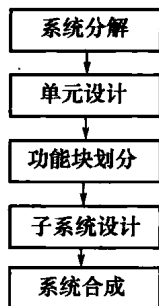


图 1.8 自底向上设计流程

1.5.2 自顶向下(Top-Down)的设计方法

随着硬件技术以及 HDL 语言的发展,传统的自底向上的设计方法已经不能满足复杂度日益增长的系统的要求。目前,在电子系统的设计中已经越来越多地采用自顶向下的设计方法了。自顶向下的设计(即 Top-Down 设计)是从系统级开始,把系统划分为基本单元,然后再把每个基本单元划分为下一层次的基本单元,一直这样做下去,直到可以直接用 EDA 元件库中的元件来实现为止。对于设计开发整机电子产品的单位和个人来说,新产品的开发总是从系统设计入手,先进行方案的总体论证、功能描述、任务和指标的分配。

随着系统变得复杂和庞大,特别需要在样机问世之前,对产品的全貌有一定的预见性。目前,EDA 技术的发展使得设计师有可能实现真正的自顶向下的设计,其流程如图 1.9 所示。

自顶向下的设计方法一般采用 HDL 语言,具有以下一些优点。

- (1)在设计周期开始就做好了系统分析。
- (2)由于设计的主要仿真和调试过程是在高层次完成的,所以能够在早期发现结构设计上的错误,避免设计工作的浪费,同时也减少了逻辑仿真的工作量。
- (3)自顶向下的设计方法方便了从系统划分和管理整个项目,使得几十万门甚至几百万门规模的复杂数字电路的设计成为可能。并可减少设计人员,避免不必要的重复设计,提高了设计的一次成功率。

自顶向下的设计方法有以下缺点。

- (1)因采用的综合工具不一样,得到的最小单元不标准。
- (2)制造成本高。

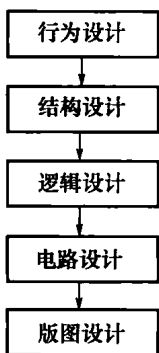


图 1.9 自顶向下设计流程

1.5.3 HDL 综合设计方法

复杂数字逻辑电路和系统的设计过程通常是以上两种设计方法的结合。复杂系统的设计需要综合考虑多个目标的平衡。在高层系统采用自顶向下的设计方法来实现,而在低层

系统采用自底向上的方法从元件库或数据库中调用已有的单元设计。综合设计方法兼有上述两种设计方法的优点,而且可以使用矢量测试库进行测试。

1.6 可编程器件概述

数字电子技术发展日新月异,使集成电路不断发展、更新换代,由早期的电子管、晶体管、中小规模集成电路,发展到超大规模集成电路以及许多实现专门功能的专用集成电路。同时随着微电子技术的发展,设计与制造集成电路的任务已不完全由半导体厂商来独立承担,设计工程师们希望自行开发设计出可用的集成电路芯片。因此出现了各种可编程逻辑器件,至今主要有五种器件可供实现专用集成电路的要求,它们是:

- (1) 简单可编程逻辑器件(Programmable Logic Device, PLD);
- (2) 复杂可编程逻辑器件(Complex Programmable Logic Device, CPLD);
- (3) 现场可编程门阵列(Field Programmable Gate Array, FPGA);
- (4) 标准单元(Standard Cell);
- (5) 门阵列(Gate Array)。

这些器件的出现,使电子系统的设计工程师利用与器件相应的计算机辅助设计(Computer Aided Design, CAD)软件,在实验室里就可以设计出 ASIC 芯片。

1.6.1 CPLD

CPLD 是结构比较复杂的可编程逻辑器件,逻辑宏单元内部主要包括与或阵列、可编程触发器和多路选择器等电路,能独立地配置为时序或组合逻辑工作方式。CPLD 器件与 GAL 器件相似,其逻辑宏单元与 I/O 做在一起,称为输出逻辑宏单元,但其宏单元及其与阵列数目比 GAL 大得多。CPLD 器件的宏单元在内部,称为内部逻辑宏单元。CPLD 除了密度高之外,许多优点都反映在逻辑单元上。它包括以下输出宏单元结构和特点。

(1) 寄存器输出和反馈,可用于实现计数器和移位寄存器等。

(2) 多触发器结构和“隐埋”触发器结构。CPLD 的宏单元内通常含有两个或两个以上的触发器,其中只有一个触发器与输出端相连,其余触发器的输出不与输出端相连,但可以通过相应的缓冲电路反馈到与阵列,从而与其他触发器一起构成较复杂的时序电路。不与输出端相连的触发器称为“隐埋”触发器,这种结构对于引脚数有限的 CPLD 器件来说,可以增加触发器数目。

(3) 可编程 I/O 单元,允许用户对这些引脚编程,作为输入或输出。

(4) 异或门输出结构,可用于一般用户的多功能计数,能十分有效地建立大的计数器。

(5) 乘积项共享结构。如果 CPLD 可以借助可编程开关将同一单元(或其他单元)中的其他或门与之联合起来使用,或者在每个宏单元中提供未使用的乘积项供其他宏单元使用和共享,提高了资源利用率,可以实现复杂的逻辑函数。

(6) 异步时钟和时钟选择。CPLD 器件中各触发器的时钟可以异步工作,有些器件中触发器的时钟还可以通过数据选择器或时钟网络进行选择。此外,逻辑宏单元内触发器的异步清零和异步置位也可以用乘积项进行控制,因而使用时更加灵活。

CPLD 作为可编程逻辑器件的扩展。它通常由可编程逻辑的功能块围绕一个位于中心、时延固定的可编程互连矩阵构成。由于用固定长度的金属线实现逻辑单元之间的互连,而可编程逻辑单元又是类似 PAL 的与阵列,使得 CPLD 与 FPGA 相比很容易计算输入到输出的传输延迟,但是会有灵活性方面的一些限制。不过,CPLD 的设计比 FPGA 简单。

1.6.2 FPGA

FPGA 是 20 世纪 80 年代中期出现的高密度可编程器件,短短十几年来,取得了惊人的发展,其单片集成密度从最初的 1 200 门发展到目前的几百万门,而且时钟频率由最初不到 10 MHz 发展到目前 GHz 以上。它与 CPLD 不同之处在于,FPGA 的结构类似于掩膜可编程门阵列(MPGA),由许多独立的可编程模块组成,用户可以通过编程将这些模块连接起来实现不同的设计。FPGA 兼容了 MPGA 和阵列器 CPLD 两者的优点,因而具有更高的集成度、更强的逻辑实现能力和更好的设计灵活性,在基本结构方面,FPGA 一般的几种分类方法如下。

1. 按逻辑功能块的大小分类

可编程逻辑块(CLB)是基本逻辑构造单元。按照逻辑功能块的大小不同,可将 FPGA 分为细粒度结构和粗粒度结构两类。其中细粒度 FPGA 的逻辑功能块一般较小,仅由很小的几个晶体管组成,非常类似于门阵列的基本单元,其优点是功能块的资源可以在实际工程中被完全利用,缺点是完成复杂的逻辑功能需要大量的连线 and 开关,速度较慢;而粗粒度 FPGA 的逻辑块规模大,功能强,完成复杂逻辑只需要较少的功能块和内部连线,能获得较好性能,缺点是功能块的资源利用不充分。

2. 按互连结构分类

根据 FPGA 内部的连线结构不同,可将其分为分段互连型和连续互连型两类。分段互连型 FPGA 中有不同长度的多种金属线,各金属线段之间通过开关矩阵或反熔丝编程连接。这种连线结构走线灵活,有多种可行方案,但走线延时与布局布线的具体处理过程有关,在设计完成前无法预测,设计修改将引起延时性能发生变化。而连续互连型 FPGA 是利用相同长度的金属线,通常是贯穿于整个芯片的长线来实现逻辑功能块之间的互连,连接与距离远近无关。在这种连线结构中,不同位置逻辑单元的连接线是确定的,因而布线延时是固定和可预测的。

3. 按编程特性分类

根据采用的开关元件的不同,FPGA 可分为一次编程型和可重复编程型两类。一次编程型 FPGA 采用反熔丝开关元件,其工艺技术决定了这种器件具有体积小、集成度高、互连线特性阻抗低、寄生电容小及可获得较高的速度等优点;此外它还有加密位、反拷贝、抗辐射、抗干扰、不需外接 PROM 或 EPROM 等优点。由于只能一次编程,比较适合于定型产品及大批量应用。

可重复编程型 FPGA 采用 SRAM 开关元件或快闪 EPROM 控制的开关元件。每个逻辑块的功能以及它们之间的互连模式由存储在芯片中的 SRAM 或快闪 EPROM 中的数据决定。SRAM 型开关的 FPGA 是易失性的,每次重新加电,FPGA 都要重新装入配置数据,实现系统功能的动态重构。采用快闪 EPROM 控制开关的 FPGA 具有非易失性和可重复编程的双重优点,但不能实现动态重构。此外,其静态功耗较反熔丝型及 SRAM 型的 FPGA 要高。