



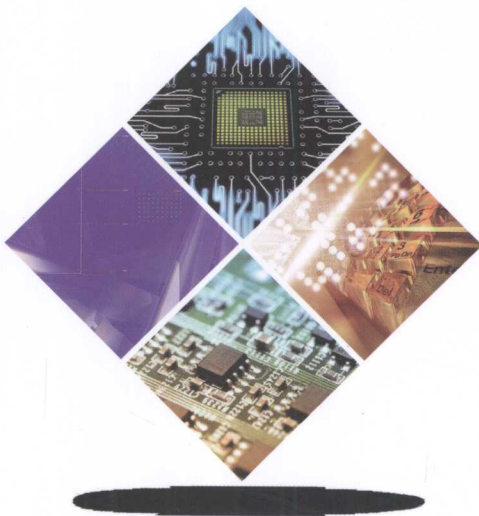
自动化类专业系列实验教材
AUTOMATION

EXPERIMENT OF CPLD AND ITS APPLICATION

可编程器件及应用 实验技术

于强◎主编

胡文彬 薛冰 韦金辰◎编著



清华大学出版社



自动化类专业系列实验教材
AUTOMATION

EXPERIMENT OF CPLD AND ITS APPLICATION

可编程器件及应用 实验技术

于强◎主编

胡文彬 薛冰 韦金辰◎编著

清华大学出版社
北京

内 容 简 介

本书在介绍 EDA 设计开发技术的基础上,着重对基于可编程器件(CPLD)的实践应用技术进行说明,通过设置具有代表性的基础实验和综合实验项目,使读者理解和掌握数字系统设计基本原理和方法。

全书内容分为 5 章。第 1 章对 EDA 技术与应用及可编程器件设计方法与流程等内容进行介绍,帮助读者建立基本的专业知识背景。第 2 章为可编程器件实验系统的硬件平台介绍,主要对书中实验项目所使用的硬件资源以及硬件平台使用注意事项进行说明,为后续实验做硬件方面准备。第 3 章是实验系统软件环境的介绍,包括 Quartus II 集成开发环境的使用方法、在 Quartus II 中调用 ModelSim 进行仿真的方法等,为后续实验做软件方面准备。第 4 章、第 5 章是本书的主要内容,分别设置了 7 个基础性和 5 个综合性可编程器件设计与应用实验项目,由浅入深地指导读者掌握可编程器件应用技术,锻炼读者独立开发具有一定复杂程度的数字系统的能力。

本书适合作为高等院校可编程器件设计应用及其相关专业的高年级本科生、研究生的教材,同时可供相关科研人员、学者、工程技术人员参考。

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

版权所有,侵权必究。侵权举报电话:010-62782989 13701121933

图书在版编目(CIP)数据

可编程器件及应用实验技术/于强主编. --北京:清华大学出版社,2016

自动化类专业系列实验教材

ISBN 978-7-302-44421-3

I. ①可… II. ①于… III. ①可编程序逻辑阵列—高等学校—教材 IV. ①TP303

中国版本图书馆 CIP 数据核字(2016)第 168701 号

责任编辑:文 怡 王 芳

封面设计:李召霞

责任校对:李建庄

责任印制:宋 林

出版发行:清华大学出版社

网 址: <http://www.tup.com.cn>, <http://www.wqbook.com>

地 址:北京清华大学学研大厦 A 座 邮 编:100084

社 总 机:010-62770175 邮 购:010-62786544

投稿与读者服务:010-62776969, c-service@tup.tsinghua.edu.cn

质量反馈:010-62772015, zhiliang@tup.tsinghua.edu.cn

课件下载: <http://www.tup.com.cn>, 010-62795954

印 装 者:三河市中晟雅豪印务有限公司

经 销:全国新华书店

开 本:185mm×230mm 印 张:10.25

字 数:219 千字

版 次:2016 年 9 月第 1 版

印 次:2016 年 9 月第 1 次印刷

印 数:1~2000

定 价:29.00 元

产品编号:059012-01

FOREWORD

可编程器件技术是当今信息领域发展速度最快、应用最广泛的技术之一。

本书以可编程器件实验为主要内容,将专业知识和应用技术相结合,帮助读者快速掌握可编程器件应用基本技术,并具有初步的数字逻辑电路设计能力。

读者通过本书的学习,能够了解 PLD 技术的发展及硬件描述语言的广泛应用,掌握 VHDL 语言的基本用法及 Quartus II 工具,最终能够运用 VHDL 语言独立设计具有一定功能的数字逻辑电路,具备应用可编程器件编程能力。

全书从内容结构上分为三部分。第一部分为可编程器件实验技术基础,共包括 3 章内容。第 1 章绪论介绍可编程器件及 EDA 技术的基本概念、设计方法和流程;第 2 章对实验中用到的硬件资源进行介绍和说明,包括开发板核心电路和外设电路,以及开发板使用注意事项;第 3 章对实验用到的软件工具进行描述,主要包括 Quartus II 集成开发环境的使用方法和 ModelSim 仿真软件的使用方法。

第二部分和第三部分分别从基础性实验和综合设计性实验两个层次,从简单到复杂,循序渐进引导读者学习和实践可编程器件设计应用技术。第二部分为可编程器件基础性实验篇,包括 7 个基础性实验。其中,实验一为组合逻辑全加器设计,实验二为组合逻辑 3-8 译码器设计,实验三为四位数据比较器设计,实验四为多路数据选择器设计,实验五为编码器设计,实验六为计数器设计,实验七为七段数码管控制设计。第三部分为可编程器件设计和技术提高篇,设置了五个综合设计性实验。其中,实验一为组合逻辑电路设计,实验二为计数器及时序电路设计,实验三为数字钟设计,实验四为函数信号发生器设计,实验五为模拟信号检测电路设计。本书末尾为每个实验设计了相应的实验报告,可以作为活页供实验者直接填写,方便快捷。

本书融合了作者多年来的教学积累和工程应用实践经验,力图展现以下特点:

第一,简单明了地介绍可编程器件应用技术的知识要点,并将知识点融入实验项目中,在由浅入深的实践中逐步建立完善的设计思路,达到迅速掌握设计工具的使用方法、形成初步设计能力的目的;

第二,选取具有一定代表性和通用性的软硬件资源作为开发工具,配以典型的设计实

例,并附有源代码,使得实验项目具有真实可操作性和检验性;

第三,力求内容丰富、图文并茂、文字流畅,力图使本书成为一本学习和使用可编程器件应用技术方面有价值的参考书。

在本书编写过程中为了更好地反映新技术的发展,参考和引用了多人的研究成果和书籍,在此对有关作者一并表示衷心感谢。

由于时间仓促,加上作者水平所限,疏漏之处在所难免,敬请读者批评指正。

作 者

2016年6月

第 1 章 绪论	1
1.1 EDA 技术及其应用介绍	1
1.2 可编程逻辑器件概述	3
1.3 PLD 设计开发简介	6
1.3.1 PLD 设计方法	6
1.3.2 PLD 设计流程	7
1.4 本章小结	12
第 2 章 实验系统硬件平台介绍	13
2.1 CPLD 开发板介绍	13
2.2 开发板核心电路详解	15
2.2.1 CPLD 主芯片	15
2.2.2 电源电路	17
2.2.3 时钟和复位电路	18
2.2.4 JTAG 电路	20
2.2.5 扩展 I/O 分配电路	20
2.3 板载外设电路详解	21
2.3.1 按键及 LED	21
2.3.2 拨码开关电路	21
2.3.3 蜂鸣器电路	21
2.3.4 七段数码管显示电路	22
2.3.5 液晶 1602 与 12864 接口电路	24
2.3.6 RS232 串口电路	24
2.3.7 VGA 接口电路	24

2.3.8	PS/2 键盘、鼠标接口电路	24
2.3.9	串行 DAC、ADC 电路	25
2.3.10	I2C 接口电路	27
2.3.11	实时时钟电路	28
2.3.12	温度传感器电路	28
2.3.13	红外线接收电路	29
2.3.14	步进电机控制电路	29
2.4	板载跳线说明	29
2.5	开发板使用注意事项	31
2.5.1	供电安全	31
2.5.2	JTAG 的插拔安全	31
2.6	本章小结	32
2.7	习题	32
第 3 章	实验系统软件环境说明	33
3.1	Quartus II 集成开发环境简介	33
3.1.1	概述	33
3.1.2	Quartus II 的安装	35
3.1.3	Quartus II 图形用户界面介绍	38
3.2	Quartus II 使用方法	42
3.2.1	创建工程	42
3.2.2	建立顶层图	44
3.2.3	仿真	48
3.2.4	生成顶层原理图	51
3.2.5	分配管脚	51
3.2.6	编程下载	52
3.2.7	连线	52
3.3	ModelSim 使用方法	53
3.3.1	ModelSim 软件安装	53
3.3.2	在 Quartus II 中调用 ModelSim-Altera	55
3.4	本章小结	60
3.5	习题	60

第 4 章 可编程器件实验——基础篇	61
4.1 实验一 组合逻辑全加器设计	61
4.2 实验二 组合逻辑 3-8 译码器设计	70
4.3 实验三 四位数据比较器设计	75
4.4 实验四 多路数据选择器设计	78
4.5 实验五 编码器设计	83
4.6 实验六 计数器设计	88
4.7 实验七 七段数码管控制设计	92
4.8 本章小结	96
4.9 习题	97
第 5 章 可编程器件实验——提高篇	98
5.1 实验一 组合逻辑电路设计	98
5.2 实验二 计数器及时序电路设计	102
5.3 实验三 数字钟设计	110
5.4 实验四 函数信号发生器设计	118
5.5 实验五 模拟信号检测电路设计	120
5.6 本章小结	124
参考文献	125
实验报告一	127
实验报告二	129
实验报告三	131
实验报告四	133
实验报告五	137
实验报告六	141
实验报告七	143
实验报告八	145
实验报告九	147
实验报告十	151
实验报告十一	153
实验报告十二	155



绪 论

本章学习目标

- 了解 EDA 技术及其应用。
- 学习可编程器件基本结构、分类及原理。
- 掌握 PLD 的设计方法及流程。

本章旨在介绍现代数字系统设计概念、方法等基本常识,使读者对可编程器件设计与应用有一个概略认识。

1.1 EDA 技术及其应用介绍

微电子技术的进步主要表现在大规模集成电路加工技术即半导体工艺技术的发展上,表征半导体工艺水平的线宽已经达到了 45nm,并还在不断地缩小,而在硅片单位面积上,集成了更多的晶体管。集成电路设计正在不断地向超大规模、极低功耗和超高速的方向发展,专用集成电路(Application Specific Integrated Circuit,ASIC)的设计成本不断降低,在功能上,现代的集成电路已能够实现单片电子系统(System on a Chip,SoC)。

现代电子设计技术的核心已日趋转向基于计算机的电子设计自动化技术(Electronic Design Automation,EDA)。EDA 技术就是依赖功能强大的计算机,在 EDA 工具软件平台上,对以硬件描述语言(Hardware Description Language,HDL)为系统逻辑描述手段完成的设计文件,自动地完成逻辑编译、化简、分割、综合、布局布线以及逻辑优化和仿真测试,直至实现既定的电子线路系统功能。EDA 技术使得设计者的工作仅限于利用软件的方式,即利用硬件描述语言和 EDA 软件来完成对系统硬件功能的实现,这是电子设计技术的一个巨大进步。

EDA 技术在硬件实现方面融合了大规模集成电路制造技术、集成电路版图设计、

ASIC 测试和封装、现场可编程门阵列(Field Programmable Gate Array,FPGA)/复杂可编程逻辑器件(Complex Programmable Logic Device,CPLD)编程下载和自动测试等技术;在计算机辅助工程方面融合了计算机辅助设计(CAD)、计算机辅助制造(CAM)、计算机辅助测试(CAT)、计算机辅助工程(CAE)技术以及多种计算机语言的设计概念;而在现代电子学方面则包含了更多的内容,如电子线路设计理论、数字信号处理技术、数字系统建模和优化技术及其高频的长线技术理论等。因此,EDA 技术为现代电子理论和设计的表达与实现提供了可能性。正因为 EDA 技术丰富的内容以及与电子技术各学科领域的相关性,其发展的历程与大规模集成电路设计技术、计算机辅助工程、可编程逻辑器件,以及电子设计技术和工艺的发展是同步的。

20 世纪 70 年代,在集成电路制作方面,MOS 工艺得到广泛的应用,可编程逻辑技术及其器件问世,计算机作为一种运算工具在科研领域得到广泛应用。而在后期,CAD 的概念已见雏形,这一阶段人们开始利用计算机取代手工劳动,辅助进行集成电路版图编辑、PCB 布局布线等工作。

20 世纪 80 年代,集成电路设计进入了互补场效应管(Complementary Metal Oxide Semiconductor,CMOS)时代,CPLD 进入商业应用,相应的辅助设计软件投入使用。在 80 年代末,出现了 FPGA。CAE 和 CAD 技术的应用更为广泛,它们在印制电路板(Printed Circuit Board,PCB)设计的原理图输入、自动布局布线及 PCB 分析,以及逻辑设计、逻辑仿真、布尔函数综合和化简等方面担任了重要的角色。特别是各种硬件描述语言的出现、应用以及标准化方面的重大进步,为电子设计自动化必须解决的电路建模、标准文档及仿真测试奠定了基础。

进入 20 世纪 90 年代,随着硬件描述语言标准化的进一步确立,计算机辅助工程、辅助分析和辅助设计在电子技术领域获得更加广泛的应用。与此同时,电子技术在通信、计算机及家电产品生产中的市场需求和技术需求,极大地推动了全新的电子设计自动化技术的应用和发展。特别是集成电路设计工艺步入了超深亚微米阶段,百万门以上的大规模可编程逻辑器件的陆续面世,以及基于计算机技术的面向用户的低成本大规模 ASIC 设计技术的应用,促进了 EDA 技术的成形。更为重要的是各 EDA 公司致力于推出兼容各种硬件实现方案和支持标准硬件描述语言的 EDA 工具软件,更有效地将 EDA 技术推向成熟和实用。

EDA 技术在进入 21 世纪后,得到了更大的发展,突出表现在以下方面:

- 在 FPGA 上实现数字信号处理(Digital Signal Processing,DSP)应用成为可能,用纯数字逻辑进行 DSP 模块的设计,使得高速 DSP 实现成为现实,并有力地推动了软件无线电技术的实用化和发展。基于 FPGA 的 DSP 技术,为高速数字信号处理算法提供了实现途径。
- 嵌入式处理器软核的成熟,使得可编程片上系统(System On a Programmable

Chip, SOPC)步入大规模应用阶段,在一片 FPGA 上实现一个完备的数字处理系统成为可能。

- 在仿真和设计两方面支持标准硬件描述语言的功能强大的 EDA 软件不断推出。
- 电子技术领域全方位融入 EDA 技术,除了日益成熟的数字技术外,传统的电路系统设计建模理念发生了重大的变化,如软件无线电技术的崛起、模拟电路系统硬件描述语言的表达和设计的标准化、系统可编程模拟器件的出现、数字信号处理和图像处理的全硬件实现方案的普遍接受以及软硬件技术的进一步融合等。
- EDA 使得电子领域各学科的界限更加模糊,更加互相包容,如模拟与数字、软件与硬件、系统与器件、ASIC 与 FPGA、行为与结构等。
- 基于 EDA 的用于 ASIC 设计的标准单元已涵盖大规模电子系统及复杂 IP 核模块。
- 软硬 IP(Intellectual Property)核在电子行业的产业领域广泛应用。
- SoC 高效低成本设计技术的成熟。系统级、行为验证级硬件描述语言的出现(如 System C),使复杂电子系统的设计和验证趋于简单。

1.2 可编程逻辑器件概述

可编程逻辑器件(Programmable Logic Device, PLD)起源于 20 世纪 70 年代,是在专用集成电路的基础上发展起来的一种新型逻辑器件,是当今数字系统设计的主要平台,其主要特点是完全由用户通过软件进行配置和编程,从而完成特定的功能,且可以反复擦写。

PLD 的基本组成部分是一个“与门”阵列和一个“或门”阵列,电路的每个输出都是输入的“与-或”函数。阵列中输入线和输出线的交点通过逻辑元件相连接。这些元件是接通还是断开,可由厂家根据器件的结构特征决定或由用户根据要求编程决定。PLD 的基本结构如图 1.1 所示。

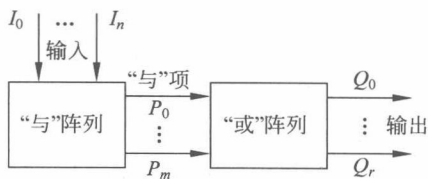


图 1.1 PLD 的基本结构

PLD“与门”阵列的输入为外部输入原变量及在阵列中经过反相后的反变量。它们按所要求的规律连接到各个“与门”的输入端,并在各“与门”的输出端产生某些输入变量的“与”项作为“或门”阵列的输入,这些“与”项按一定的要求连接到相应或门的输入端,在每个或门的输出端产生输入变量的“与-或”函数表达式。

在上述基本结构的基础上,附加一些其他逻辑元件,如输入缓冲器、输出寄存器、内部反馈、输出宏单元等,便可构成各种不同的 PLD。

可编程逻辑器件根据集成密度可分为低密度可编程逻辑器件(LDPLD)和高密度可编程逻辑器件(HDPLD)两类。LDPLD 主要是指早期发展起来的 PLD,它包括 PROM、可编程逻辑阵列(Programmable Logic Array, PLA)、可编程阵列逻辑(Programmable Array Logic, PAL)和通用阵列逻辑(Generic Array Logic, GAL)四种,其集成密度一般小于 700 门/片,这里的门是指 PLD 等效门。HDPLD 包括 EPLD、CPLD 和 FPGA 三种,其集成密度大于 700 门/片。随着集成工艺的发展,HDPLD 的集成密度不断增加,性能不断提高。目前集成度最高的 HDPLD 集成密度每片可达数百万门。

可编程逻辑器件根据编程方式分为两类:一类是一次性编程(One Time Programmable, OTP)器件;另一类是可多次编程器件。OTP 器件只允许对器件编程一次,编程后不能修改,其优点是集成度高、工作频率和可靠性高、抗干扰性强。可多次编程器件的优点是可多次修改设计,特别适合于系统样机的研制。

目前,常用的可编程逻辑器件都是从与或阵列和门阵列发展起来的,所以可以从结构上将其分为两大类:①阵列型 PLD;②现场可编程门阵列 FPGA。阵列型 PLD 的基本结构由“与门”阵列和“或门”阵列组成。简单 PLD(PROM、PLA、PAL 和 GAL)、EPLD 和 CPLD 都属于阵列型 PLD。

1. EPLD 和 CPLD

可擦除可编程逻辑器件(Erasable PLD, EPLD)是 Altera 公司推出的基于电可擦除 CMOS(Electronically Erasable CMOS, E²CMOS)编程工艺的 PLD,其基本逻辑单元是宏单元。宏单元由可编程的与或阵列、可编程寄存器和可编程 I/O 三部分组成。从某种意义上讲, EPLD 是改进的 GAL,它在 GAL 的基础上大量增加宏单元的数目,提供更大的与阵列,灵活性比 GAL 有较大改善,集成密度大幅度提高,内部连线相对固定,延时小,有利于器件在高频下工作,但内部互连能力相对较弱。

复杂可编程逻辑器件(Complex PLD, CPLD)是在 20 世纪 80 年代末 Lattice 公司提出系统可编程(In System Programmability, ISP)技术以后,于 90 年代初出现的。它是在 EPLD 的基础上发展起来的,采用 E²CMOS 工艺制作。与 EPLD 相比,增加了内部连线,对逻辑宏单元和 I/O 单元也有重大的改进。其典型器件有 Altera 公司的 MAX7000 系列, Xilinx 公司的 XC7000 和 XC9500 系列, Lattice 公司的 ispLSI 系列和 AMD 公司的

MACH 系列。

2. FPGA

现场可编程门阵列(Field Programmable Gate Array,FPGA)是 Xilinx 公司于 1985 年推出的,采用 CMOS-SRAM 编程工艺制作。其内部由可构造逻辑模块(CLB)、可构造输入输出块(IOB)和可编程互连资源(IR)组成。FPGA 具有集成密度高、编程速度快、设计灵活及可再配置等特点。FPGA 采用查找表(Look-Up-Table)结构,如 Altera 公司的 ACEX、APEX 系列,Xilinx 公司的 Spartan、Virtex 系列等。

查找表简称为 LUT,LUT 本质上就是一个 RAM。当用户通过原理图或 HDL 语言描述了一个逻辑电路以后,PLD/FPGA 开发软件会自动计算逻辑电路的所有可能的结果,并把结果事先写入 RAM。这样,每输入一个信号进行逻辑运算就等于输入一个地址进行查表,找出地址对应的内容,然后输出即可。

由于 LUT 主要适合 SRAM 工艺生产,所以目前大部分 FPGA 都是基于 SRAM 工艺的,而 SRAM 工艺的芯片在掉电后信息就会丢失,一定需要外加一片专用配置芯片。在上电的时候,由这个专用配置芯片把数据加载到 FPGA 中,FPGA 就可以正常工作。由于配置时间很短,不会影响系统正常工作。也有少数 FPGA 采用反熔丝或 Flash 工艺,对这种 FPGA,就不需要外加专用的配置芯片。

3. FPGA 与 CPLD 的比较

FPGA 基于 SRAM 的架构,集成度高,以 LE(包括查找表、触发器及其他)为基本单元,有内嵌存储器、DSP 等,支持 I/O 标准丰富。具有易挥发性,需要有上电加载过程。在实现复杂算法、队列调度、数据处理、高性能设计、大容量缓存设计等领域中有广泛应用,如 Altera Stratix 系列。

CPLD 基于 E²CMOS 工艺,集成度低,以 MicroCell(包括组合部分与寄存器)为基本单元,具有非挥发特性,可以重复写入。在黏合逻辑、地址译码、简单控制、FPGA 加载等设计中有广泛应用,如 Altera MAX3000A 系列。

尽管 FPGA 和 CPLD 都是可编程 ASIC 器件,有很多共同特点,但由于 CPLD 和 FPGA 结构上的差异,它们具有各自的特点:

- CPLD 更适合完成各种算法和组合逻辑,FPGA 更适合于完成时序逻辑。
- 在编程上 FPGA 比 CPLD 具有更大的灵活性。CPLD 通过修改具有固定内连电路的逻辑功能来编程,FPGA 主要通过改变内部连线的布线来编程。
- FPGA 的集成度比 CPLD 高,具有更复杂的布线结构和逻辑实现。
- CPLD 的编程采用 E²PROM 或 FAST Flash 技术,无须外部存储器芯片,使用简单。而 FPGA 的编程信息需存放在外部存储器上,使用方法复杂。

- CPLD 的速度比 FPGA 快,并且具有较大的时间可预测性。这是由于 FPGA 是门级编程,并且 CLB 之间采用分布式互联,而 CPLD 是逻辑块级编程,并且其逻辑块之间的互联是集总式的。
- CPLD 编程次数可达 1 万次,系统断电时编程信息也不丢失。FPGA 每次上电时,需从器件外部将编程数据重新写入 SRAM 中,编程信息在系统断电时丢失,其优点是可在工作中快速编程,且次数不限,从而可以实现板级和系统级的动态配置。
- CPLD 保密性好,FPGA 保密性相对较差。
- 一般情况下,CPLD 的功耗要比 FPGA 大,且集成度越高越明显。

1.3 PLD 设计开发简介

PLD 的设计开发流程就是利用 EDA 开发软件和编程工具对 PLD 芯片进行开发的过程。

1.3.1 PLD 设计方法

PLD 是可编程芯片,因此 PLD 的设计方法包括硬件设计和软件设计两部分。PLD 硬件包括 PLD 芯片电路、存储器、输入输出接口电路以及其他设备,软件即相应的 HDL 程序以及嵌入式 C 程序。硬件设计是基础,但其方法比较固定,本书主要介绍软件的设计方法。

总的来说,在 PLD 设计中,一般有两种方法:自上至下设计或自下至上设计。较大规模的设计一般采用前者,首先进行模块分割,把模块分割成子模块,然后再把子模块分割成下一级子模块,依次逐级划分。利用模块分割可以简化设计,提高程序的可读性,提高逻辑综合效率,提高程序的可移植性,从而便于理解设计所完成的功能,便于读懂程序,便于调试,有利于提高设计的性能,设计的可靠性。

目前微电子技术已经发展到 SoC 阶段,即集成系统阶段,相对于集成电路的设计思想有着革命性的变化。SoC 是一个复杂的系统,它将一个完整产品的功能集成在一个芯片上,包括核心处理器、存储单元、硬件加速单元以及众多的外部设备接口等,具有设计周期长、实现成本高等特点,因此其设计方法必然是自顶向下地从系统级到功能模块的软硬件协同设计,达到软硬件的无缝结合。

这么庞大的工作量显然超出了单个工程师的能力,因此需要按照层次化、结构化的设计方法来实施。首先由总设计师将整个软件开发任务划分为若干个可操作的模块,并对其接口和资源进行评估,编制出相应的行为或结构模型,再将其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,并为自己所设计的模块负责,然后由上层模块设计师对下层模块进行功能验证。

自顶向下的设计流程从系统设计开始,将系统划分为若干个二级单元,然后再把各个二级单元划分为下一层次的基本单元,一直下去,直到能够使用基本模块或者 IP 核直接实现为止。流行的 PLD 开发工具都提供了层次化管理,可以有效地梳理错综复杂的层次,能够方便地查看某一层次模块的源代码以修改错误。

在工程实践中,还存在软件编译时长的问题。由于大型设计包含多个复杂的功能模块,其时序收敛与仿真验证复杂度很高,为了满足时序指标的要求,往往需要反复修改源文件,再对所修改的新版本进行重新编译,直到满足要求为止。这里存在两个问题:首先,软件编译一次需要长达数小时甚至数周的时间,这是开发者所不能容忍的;其次,重新编译和布局布线后结果差异很大,会破坏已满足时序的电路。因此必须提出一种有效提高设计性能,继承已有结果,便于团队化设计的软件工具。PLD 厂商意识到这类需求,由此开发出了相应的逻辑锁定和增量设计的软件工具。

1.3.2 PLD 设计流程

PLD 的开发流程一般包括设计定义、设计输入、功能仿真、逻辑综合、布局布线、时序仿真、时序分析以及芯片下载验证等主要步骤。图 1.2 为 PLD 设计流程图。

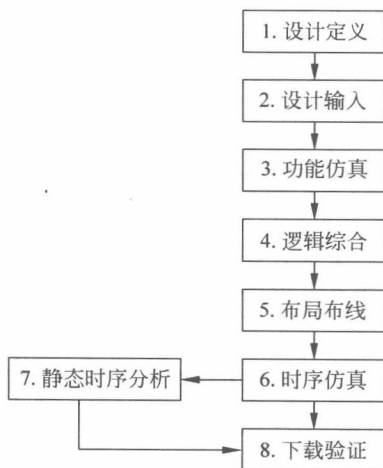


图 1.2 PLD 设计流程

1. 设计定义

依据系统概要设计指导和详细设计具体规定,设计定义规定了 PLD 模块必须完成的

功能以及与外围器件的接口,包括接口信号规格、处理时钟频率、时序要求、管脚分配锁定等,是对 PLD 进行编程设定的依据。对设计定义的要求是合理、清晰、准确。

2. 设计输入

设计输入是将所设计的系统或电路以开发软件要求的某种形式表示出来,并输入给 EDA 工具的过程。常用的方法有硬件描述语言(HDL)与原理图输入两种方式。HDL 设计方式是现今设计大规模数字集成电路的良好形式,除 IEEE 标准中 VHDL 与 Verilog HDL 两种形式外,尚有各 PLD 厂商推出的专用语言,如 Quartus 下的 AHDL。HDL 语言描述在状态机(State Machine)、控制逻辑、总线功能方面较强,可以利用特定综合器(如 Synopsys 公司的 PLD Compiler II 或 PLD Express)以具体硬件单元较好地实现所描述电路。原理图输入方式是一种最直接的描述方式,在可编程芯片发展的早期应用比较广泛,它将所需的器件从元件库中调出来,画出原理图。原理图输入在顶层设计、数据通路逻辑、手工最优化电路等方面具有图形化直观、单元节俭、功能明确等特点,但效率很低,且不易维护,不利于构造和重用。原理图输入方式更主要的缺点是可移植性差,当芯片升级后,所有的原理图都需要作一定的改动。

目前的设计趋势是以 HDL 语言为主,原理图为辅,进行混合设计以发挥二者各自的长处。在 HDL 语言中,VHDL 的数据类型丰富,对大型系统的描述能力强;Verilog HDL 对寄存器传输级(Register Transfer Level,RTL)和门级等底层电路描述能力强,风格类似 C 语言。在使用范围上,VHDL 在欧洲使用较多,Verilog HDL 则在北美和亚洲使用较多。

3. 功能仿真

仿真是指使用设计软件包对已实现的设计进行完整测试,模拟实际物理环境下的工作情况。功能仿真仅对设计的逻辑功能进行模拟与测试以验证其功能是否正确,与具体的芯片器件无关,仿真过程中没有加入任何时序信息;而在布局布线后,提取具体芯片器件的响应延迟、连线延时等时序参数,并在此基础上进行的仿真称为时序仿真,它是接近真实器件运行的仿真。

功能仿真,是仅对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。仿真过程没有加入时序信息,不涉及具体器件的硬件特性,如延时特性。仿真前先利用波形编辑器和 HDL 等建立波形文件和测试向量,仿真结果将会生成报告文件和输出信号波形,从中便可以观察各个节点信号的变化。如果发现错误,则返回设计修改逻辑设计。图 1.3 为功能仿真示意图。

Mentor Graphics 公司的 ModelSim 是业界公认最强大、最方便易用的仿真工具软件。ModelSim 支持 VHDL 和 Verilog HDL 两种语言混编仿真,并有清晰的图形化显示

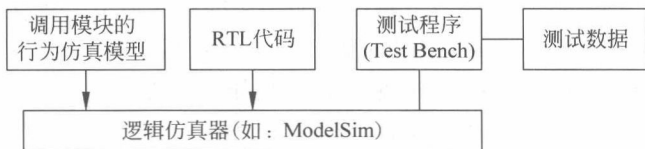


图 1.3 功能仿真示意图

供使用者观察仿真的结果。

4. 逻辑综合

综合,就是针对给定的电路实现功能和实现此电路的约束条件,如速度、功耗、成本及电路类型等,通过计算机对设计进行优化处理,获得一个能满足上述要求的电路设计方案。也就是说,被综合的文件是 HDL 文件(或相应原理图文件等),综合的依据是逻辑设计的描述和各种约束条件,综合的结果则是一个硬件电路的实现方案,该方案必须同时满足预期的功能和约束条件。对于综合来说,满足要求的方案可能有多个,综合器将产生一个最优的或接近最优的结果。因此,综合的过程也就是设计目标的优化过程,最后获得的电路结构与综合器的工作性能有关。图 1.4 为逻辑综合示意图。

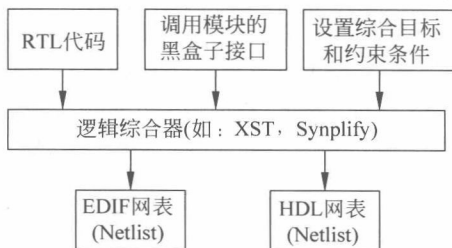


图 1.4 逻辑综合示意图

综合以 HDL 语言或原理图作为输入,输出网表文件供后续的“实现”使用。综合对设计实现后的 PLD 性能有很重要的影响,同时又不与 PLD 的内部硬件结构直接关联。因此不仅 PLD 制造厂商(如 Xilinx 公司、Altera 公司)投入了大量研发力量开发良好的综合工具嵌入自己的设计软件中,还有很多并不出产 PLD 芯片的公司研究综合技术,开发了很强大的第三方综合工具,其性能甚至超过了制造厂商自己的综合工具,如 Synopsys 公司的 Synplify。

5. 布局布线

图 1.5 为布局布线处理示意图。布局布线在 PLD 设计中有时又被称为设计“实现”(Implement)。实现可理解为利用实现工具把逻辑映射到目标器件物理结构的资源中,