



卓越工程师培养计划

▪ EDA ▪

<http://www.phei.com.cn>

张鹏南 孙 宇 夏洪洋 编著



基于 Quartus II

的 VHDL 数字系统 设计入门与应用实例



电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY

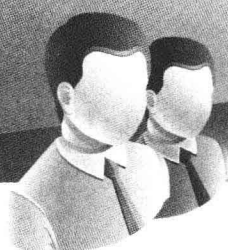


卓越工程师培养计划

▪ EDA ▪

<http://www.phei.com.cn>

张鹏南 孙 宇 夏洪洋 编著



基于Quartus II

的VHDL数字系统 设计入门与应用实例

电子工业出版社

Publishing House of Electronics Industry

北京·BEIJING

内 容 简 介

本书从实际角度出发,系统介绍了数字系统设计的相关知识,其主要内容包括 EDA 技术、FPGA/CPLD 器件、硬件描述语言 VHDL 设计基础及实例说明、Quartus II 数字集成软件和 ModelSim 仿真软件的基本操作。通过对常用数字系统功能模块设计以及典型数字系统分析和验证,来提高读者的实践能力和工程设计能力。

本书突出实用性、强调实践性,书中的实例丰富,具有代表性,而且这些实例基本都经过了实践检验。在内容上,深入浅出、语言流畅、图文并茂、通俗易懂。本书适于电子信息工程、电子科学与技术、自动化等相关领域的科研和工程技术人员阅读,也可作为高等学校相关专业的教学用书。

未经许可,不得以任何方式复制或抄袭本书之部分或全部内容。

版权所有,侵权必究。

图书在版编目(CIP)数据

基于 Quartus II 的 VHDL 数字系统设计入门与应用实例/张鹏南,孙宇,夏洪洋编著. —北京:电子工业出版社,2012.5

(卓越工程师培养计划)

ISBN 978-7-121-17011-9

I. ①基… II. ①张… ②孙… ③夏… III. ①可编程序逻辑器件-系统设计②硬件描述语言-程序设计
IV. ①TP332.1②TP312

中国版本图书馆 CIP 数据核字(2012)第 093760 号

策划编辑:张 剑(zhang@phei.com.cn)

责任编辑:刘真平

印 刷:

装 订:北京中新伟业印刷有限公司

出版发行:电子工业出版社

北京市海淀区万寿路 173 信箱 邮编 100036

开 本:787×1092 1/16 印张:15 字数:384 千字

印 次:2012 年 5 月第 1 次印刷

印 数:4 000 册 定价:39.00 元

凡所购买电子工业出版社图书有缺损问题,请向购买书店调换。若书店售缺,请与本社发行部联系,联系及邮购电话:(010) 88254888。

质量投诉请发邮件至 zlt@phei.com.cn,盗版侵权举报请发邮件至 dbqq@phei.com.cn。

服务热线:(010) 88258888。

前 言

随着微电子技术和计算机技术的发展,数字系统的设计方法发生了深刻的变化。传统的搭积木式的方法已经不能适用于现代数字系统设计的要求,这就迫切需要一种新的设计方法以适应现代数字系统的设计。

EDA 技术本身在不断发展,功能不断强大,不仅电路设计仿真需要借助 EDA 技术,系统级设计、综合、芯片版图验证等都离不开 EDA 技术。因此,EDA 技术在现代数字系统设计中起到举足轻重的作用。而应用最为广泛的复杂可编程逻辑器件 CPLD 和现场可编程门阵列 FPGA 来实现数字系统设计,也是目前应用 EDA 技术设计数字系统的潮流。因此可以基于高性能的 CPLD 和 FPGA 器件,通过硬件描述语言 VHDL 描述数字系统,再应用主流 EDA 软件工具来实现现代数字系统,这样才能使电子系统向速度更快、体积更小、稳定性更高的方向发展。

全书共分为 9 章,其中第 1 章介绍了现代数字系统、EDA 技术及可编程逻辑器件的基础知识。第 2 章主要介绍了数字集成开发软件 Quartus II 的设计流程、设计方法,并结合实例介绍了 Quartus II 的基本操作。第 3 章主要介绍了硬件描述语言 VHDL 的编程基础,包括 VHDL 程序的基本结构、VHDL 的基本语言要素、VHDL 的顺序语句和并行语句,以及子程序和 VHDL 的描述风格等。第 4 章结合仿真工具 ModelSim 的具体实例来介绍 VHDL 数字电路的仿真。第 5 章和第 6 章分别介绍了组合逻辑电路 VHDL 的程序设计和时序逻辑电路程序设计。第 7 章介绍了有限状态机的设计与实现。第 8 章主要介绍了基于 VHDL 常见功能模块的实现,包括按键消抖电路、循环冗余校验模块、序列检测器及 LED 数码管显示控制器设计。第 9 章介绍了基于自顶向下设计方法的几个典型数字系统的实例,包括数字时钟、数字频率计、简易多功能信号发生器、洗衣机控制系统及二进制振幅键控(ASK)调制器与解调器。

本书在教学实践的基础上编写,突出实用性、强调实践性。在内容的前后安排上由浅入深,由易到难。读者不仅可以在理论上掌握现代数字系统的设计流程和方法,而且可以结合具体实例利用 EDA 工具进行数字系统设计。在每章后面均附有思考与练习,以便于读者更好地理解 and 消化所学知识,相信会为读者的学习和工作带来一定的帮助。

本书可以作为高等院校电子信息工程、电子科学与技术、自动化等电子类相关领域的本科生和研究生的参考书,同时也适用于从事数字电路系统设计的科研工程技术人员。

本书由张鹏南主持编写并制定大纲。其中第 1、2、3、6 章及附录由张鹏南编写,第 4、5 章由夏洪洋编写,第 7 章由梁燕华编写,第 8、9 章由孙宇编写。全书由张鹏南统稿。参加本书编写的还有宋一兵、王献红、李文秋、张洪信、张轩、管殿柱等。

在本书的编写过程中,黑龙江科技学院的穆秀春老师对书稿提出了宝贵的建议和意见,在此表示由衷的感谢!

由于编者水平有限,书中难免有纰漏和欠妥之处,恳请广大读者批评指正。

编 著 者

· III ·

目 录

第 1 章 绪论	1
1.1 现代数字系统	1
1.2 EDA 技术	4
1.3 CPLD 与 FPGA 器件	6
1.4 思考与练习	19
第 2 章 Quartus II 开发软件	20
2.1 Quartus II 概述	20
2.2 Quartus II 设计流程	20
2.3 Quartus II 软件的使用	21
2.4 Quartus II 设计实例	26
2.5 思考与练习	50
第 3 章 硬件描述语言 VHDL 编程基础	52
3.1 VHDL 概述	52
3.2 VHDL 程序的基本结构	54
3.3 VHDL 的基本要素	61
3.4 顺序语句	73
3.5 并行语句	86
3.6 子程序	97
3.7 VHDL 的描述风格	99
3.8 思考与练习	101
第 4 章 VHDL 数字电路的仿真	103
4.1 VHDL 仿真概述	103
4.2 VHDL 测试平台	103
4.3 ModelSim 简介	107
4.4 思考与练习	117
第 5 章 组合逻辑电路 VHDL 程序设计	118
5.1 组合逻辑电路概述	118
5.2 基本门电路	118

5.3	编码器和译码器	121
5.4	多路选择器与多路分配器	126
5.5	数值比较器	130
5.6	加法器	132
5.7	乘法器	137
5.8	其他组合逻辑电路	139
5.9	综合实例	141
5.10	思考与练习	142
第 6 章	时序逻辑电路 VHDL 程序设计	144
6.1	时序逻辑电路概述	144
6.2	触发器	145
6.3	寄存器	152
6.4	计数器	158
6.5	分频器	163
6.6	其他时序逻辑电路	169
6.7	综合实例	174
6.8	思考与练习	178
第 7 章	有限状态机的设计与实现	180
7.1	有限状态机的概述	180
7.2	一般有限状态机的设计	182
7.3	Moore 型有限状态机的设计	183
7.4	Mealy 型有限状态机的设计	185
7.5	思考与练习	187
第 8 章	常用功能模块的 VHDL 实现	188
8.1	按键消抖电路的设计	188
8.2	循环冗余校验 (CRC) 模块的设计	189
8.3	序列检测器的设计	191
8.4	LED 数码管显示控制器的设计	193
8.5	思考与练习	199
第 9 章	数字系统的设计实例	200
9.1	简易数字钟的设计	200
9.2	数字频率计的设计	206
9.3	简易多功能信号发生器的设计	210
9.4	洗衣机控制系统的设计	214

9.5 二进制振幅键控 (ASK) 调制器与解调器的设计	223
9.6 思考与练习	227
附录 A VHDL 关键字及标准库文件	228
附录 B 常用的 FPGA 开发工具	229
参考文献	231

第1章 绪论

随着计算机技术、通信技术及微电子技术的飞速发展，现代电子产品已经渗透到生产和生活的各个领域，其中，EDA 技术发挥了举足轻重的作用。应用最为广泛的复杂可编程逻辑器件 CPLD 和现场可编程门阵列 FPGA 来实现数字系统设计，也是目前应用 EDA 技术设计数字系统的潮流。本章将介绍现代数字系统的基本理论、EDA 技术的基本内容，以及 CPLD/FPGA 器件的结构、主流产品和特点，以便使读者对数字系统、EDA 技术及 CPLD/FPGA 器件有更加系统的认识。



1.1 现代数字系统

随着数字集成技术和计算机技术的迅速发展，数字系统设计的理论和方法也在不断地发展和变化。数字系统的实现方法经历了由分立元件、小规模集成电路（SSI）、中规模集成电路（MSI）、大规模集成电路（LSI）到超大规模集成电路（VLSI）的过程。本节将介绍数字系统的概念、传统和现代数字系统的设计方法，以及基于 FPGA/CPLD 数字系统设计流程。

1. 数字系统的概念

数字系统是指用来处理逻辑信息，并且对数字信号进行加工、传输和存储的电路实体。最常见的数字系统包括计算机、计算器、数字音像设备、数字电话系统等。通常，数字系统主要由数据处理器和控制器两部分构成。数据处理器由寄存器和组合逻辑电路组成。寄存器用于暂时存储信息，组合逻辑电路实现对数据的加工和处理。控制器接收外部输入信号和各个逻辑功能部件、子系统的反馈输入信号，进行处理后发出各种控制命令，从而控制各个逻辑功能部件协调工作，如果大型的数字系统包括若干个子系统，则控制各个子系统协调工作，实现整个数字系统的复杂功能。

数字系统可以认为是一种层次的结构，任何复杂的数字系统都是由底层的基本电路开始，逐步向上构建起来的。数字系统的层次结构如图 1-1 所示，从底层向上，复杂的程度

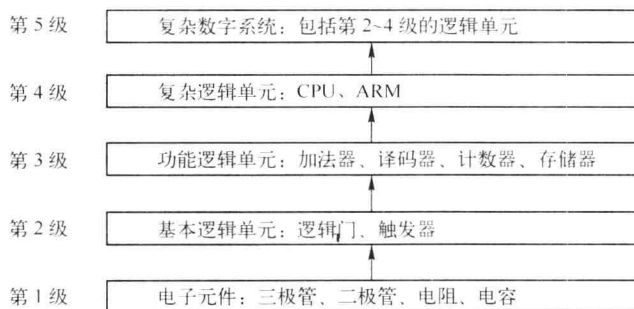


图 1-1 数字系统的层次结构

逐层增加, 功能也不断增强。基本电路由单独的原件组成, 能执行特定的功能。

数字系统设计需要考虑的是基本逻辑单元, 这样不必过多地关心门电路内部电子线路的细节, 而是更多地关注它们的外部特性及用途, 以便实现更高一级的逻辑功能。

2. 传统数字系统设计方法

传统的数字系统设计方法是自下而上 (Bottom-Up) 的设计方法, 这是一种传统的设计思想。所谓自下而上设计方法指的是首先需要进行门级单元库的设计, 然后在门级单元库的基础上建立单元库 (如加法器、译码器、寄存器等), 从小规模逐级构造成大规模以致整个数字系统, 如图 1-2 所示。自下而上设计方法的具体实施步骤是首先选择元器件, 然后用所选元器件进行逻辑电路设计, 完成系统各个独立功能模块设计, 再将各个功能模块连接起来, 完成整个系统的硬件设计。可以看出, 上述过程是从底层开始设计, 直至最高层设计完毕。

传统的数字系统设计方法依赖于设计师的经验, 注重细节, 而对整个系统缺乏规划, 当设计出现问题时, 如果要修改就会比较麻烦, 甚至前功尽弃, 设计周期长, 灵活性差, 效率低。设计系统规模越大, 其缺点就越突出。

3. 现代数字系统设计方法

现代数字系统采用的是自上而下 (Top-Down) 的设计方法, 所谓自上而下是指将数字系统的整体分成若干个子系统和模块, 若子系统的规模较大, 则还需要将子系统进一步分解为更小的子系统和模块, 每个模块也可以进一步细化, 并借助 EDA 技术完成从工艺到物理实现的映射。

在 Top-Down 设计中, 将设计分成不同的层次: 系统级、功能级、门级、开关级等。按照自上而下的顺序, 在不同层次上对系统进行描述和仿真。Top-Down 设计方法如图 1-3 所示。在 Top-Down 设计过程中, 需要有 EDA 工具的支持, 设计需要经过“设计—验证—修改设计—再验证”的过程, 不断反复, 直到得到的结果能够完全实现所要求的逻辑功能, 并且在速度、功耗、价格和可靠性方面实现较为合理的平衡。

现代数字系统设计是从顶层设计开始的, 设计中可以逐层描述, 逐层仿真, 保证满足系统指标, 克服了大规模电子系统高复杂度所带来的问题, 系统可以层次式地规划分为易于处理的子系统, 再层次式地求精。结合应用领域的具体要求, 及时调整设计方案, 进行性能优化, 从而保证设计的正确性, 缩短设计周期。

传统数字系统和现代数字系统设计方法的比较如表 1-1 所示。

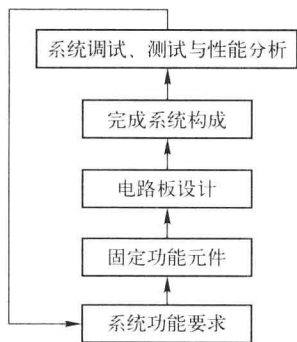


图 1-2 自下而上的设计方法

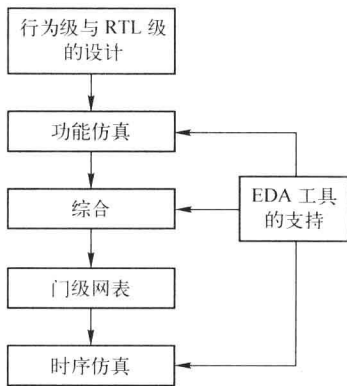


图 1-3 Top-Down 设计方法

表 1-1 传统数字系统和现代数字系统设计方法的比较

特 征	传统数字系统设计方法	现代数字系统设计方法
设计方法	自下而上	自上而下
采用器件	通用的逻辑器件	可编程逻辑器件
设计文件	电路原理图	HDL 语言编写源程序
设计对象	电路板	芯片
仿真时期	系统硬件设计后期	系统硬件设计早期

4. 基于 FPGA/CPLD 器件的数字系统设计流程

现代数字系统的实现主要可选择两类器件，一类是可编程逻辑器件（PLD），另一类是专用集成电路（ASIC）。本节主要讨论基于 FPGA/CPLD 器件的数字系统设计流程，如图 1-4 所示。

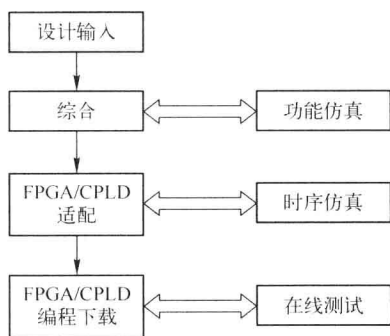


图 1-4 基于 FPGA/CPLD 器件的数字系统设计流程

1) 设计输入 设计输入是将设计者所设计的工程利用 EDA 工具以某种形式表达出来，并输入到相应开发软件中的过程。它通常包括两种输入方式：图形输入和文本输入。

(1) 图形输入：图形输入包括原理图输入、状态图输入和波形图输入等方法。最常用的是原理图输入方法。

原理图输入方法是图形化的表达方式，使用元件符号和连线来描述设计。其特点是适于描述连接关系和接口关系。对用户来讲，原理图输入设计过程形象直观，由于设计方式接近于底层电路布局，因此易于控制逻辑资源的耗用，节省面积。由于它要求设计工具提供必要的元件库或逻辑宏单元，如果输入的是较为复杂的逻辑或元件库中不存在的模型，采用原理图输入方式往往很不方便，易读性下降，错误排查困难，整体调整和结构升级困难。

(2) 文本输入：文本输入采用硬件描述语言，这是高层次设计中最为普遍的输入方式。硬件描述语言是一种用文本形式来描述和设计电路的语言，常用的语言有 AHDL、VHDL 及 Verilog HDL。设计者可以利用 EDA 工具对文本输入进行编辑和编译，最后形成某些目标文件，然后用 FPGA/CPLD 器件实现。实际上，文本输入可以采用任何文本编辑器，但是通常需要在专用的硬件描述语言编辑环境中进行。

2) 综合 所谓综合，就是将电路的高级语言描述（如 HDL、原理图或状态图的描述）转换成低级的、可与 FPGA/CPLD 或构成 ASIC 的门阵列基本结构相映射的网表文件。这是高层次描述转换为硬件电路的关键步骤，所以说综合在可编程逻辑器件开发过程中最为关

键。综合器就是能够将原理图或 HDL 语言表达或描述的电路功能转化为具体的电路结构网表的工具。一般 EDA 软件的综合器可由专业的第三方 EDA 公司提供, 当输入的 HDL 文件在 EDA 工具中检测无误后, 利用综合器来生成网表文件。

3) 适配 适配也称为结构综合, 是由综合器产生网表文件配置于指定的目标器件中, 并产生最终的可下载文件。利用适配器将综合后的网表文件针对某一具体的目标器件进行逻辑映射操作, 包括底层器件配置、逻辑分割、逻辑优化、布局布线等。一般适配器需由 FPGA/CPLD 供应商提供。

4) 仿真 仿真也称为模拟, 是对所设计电路功能的验证。用户可以在设计过程中对整个系统和各个模块进行仿真, 即在计算机上用软件验证功能是否正确, 各部分的实现配合是否准确。如果有问题可以随时进行修改, 从而避免了逻辑错误。仿真包括三种, 即行为仿真、功能仿真和时序仿真。

所谓行为仿真, 就是将 VHDL 设计源程序直接送到 VHDL 仿真器中进行仿真。该仿真只是根据 VHDL 语义进行的, 与具体电路没有关系。

所谓功能仿真, 是直接对所设计的电路进行功能仿真。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, 主要是验证电路与理想情况是否一致。仿真过程不涉及任何具体器件的硬件特性, 不经历适配阶段, 是将综合后的网表文件送到仿真器进行仿真。

所谓时序仿真, 是在选择了具体器件并完成了布局布线后进行的包含延时的仿真。它接近真实器件的运行特性, 仿真文件中已经包含了器件硬件特性参数, 因此, 仿真精度高。

通常情况下, 首先进行功能仿真, 待确认设计文件所表达的功能满足设计者原有意图时, 即逻辑功能满足要求后, 再进行综合、适配和时序仿真, 以便把握设计项目在硬件条件下的运行。

5) 编程 编程是将适配后生成的编程文件装入到可编程逻辑器件中的过程, 也称为下载。把对 FPGA 中的 SRAM 进行直接下载的方式称为配置, 但对反熔丝结构和 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。对于 CPLD 来说, 是将熔丝图文件, 即 JEDEC 文件下载到 CPLD 中去; 对于 FPGA 来说, 是将生成的数据文件, 即 BG 文件配置到 FPGA 中。

6) 在线测试 在线测试是将含有下载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。



1.2 EDA 技术

电子设计自动化 (Electronic Design Automation, EDA) 是指利用计算机完成电子系统的设计。它经历了计算机辅助设计 (Computer Assist Design, CAD)、计算机辅助工程设计 (Computer Assist Engineering, CAE) 和电子设计自动化 (Electronic Design Automation, EDA) 3 个阶段。

EDA 技术在硬件方面融合了集成电路制造技术、IC 版图设计技术、ASIC 测试、封装技术及 CPLD/FPGA 技术等; 在计算机辅助工程方面融合了计算机辅助设计 CAD 技术、计算机辅助制造 CAM 技术、计算机辅助测试 CAT 技术及多种计算机语言的设计概念; 而在现代

电子学方面则容纳了更多的内容,如数字电路设计理论、数字信号处理技术、系统建模和优化技术等。EDA 技术涉及面广,内容丰富,已经成为电子技术领域中极其重要的组成部分。

1. EDA 技术的主要内容

EDA 技术主要包括 3 个方面内容:可编程逻辑器件、硬件描述语言和软件开发工具。

1) 可编程逻辑器件 可编程逻辑器件(PLD)是一种由用户编程以实现某些逻辑功能的新型器件。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、速度及性能方面不断地改进和提高,已由最初的简单低密度 PLD 发展到如今的复杂高密度 PLD,目前已经能够支持在一个芯片上集成一个完整的系统。本书将在 1.3 节详细介绍。

2) 硬件描述语言 硬件描述语言(HDL)是一种用文本形式来描述和设计电路的语言。设计者可以利用硬件描述语言来描述自己的设计,然后利用 EDA 工具进行综合和仿真,最后变成某种目标文件,再利用 ASIC 或 FPGA 具体实现。

利用硬件描述语言进行电子系统设计可以提高效率,增加开发成果的可继承性并缩短开发时间。采用硬件描述语言来设计硬件电路既适用于小批量产品生产开发,也适用于大批量产品的研制,因而得到了广泛的应用。Verilog HDL 和 VHDL 都是应用于电路设计的硬件描述语言,并且都已经成为 IEEE 的标准,两者各自都有自己的优点和特点。本书主要介绍 VHDL 语言。

3) 软件开发工具 EDA 软件开发工具是 EDA 技术强有力的支持。现代数字系统的方案设计与验证、系统逻辑综合、布局布线、性能仿真、器件编程等均由 EDA 工具一体化来完成。

2. EDA 软件开发工具

所谓 EDA 软件开发工具是指以工作站或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。一般 EDA 开发软件包括集成的 FPGA/CPLD 开发工具、设计输入工具、逻辑综合器、仿真器、检查/分析工具及布局布线工具等。

1) 集成的 FPGA/CPLD 开发工具 集成的 FPGA/CPLD 开发工具是由 FPGA/CPLD 芯片生产厂家提供的,这些工具可以完成从设计输入、逻辑综合、模拟仿真到适配下载等全部工作。如 Altera 公司的 MAX + PLUS II、Quartus II 软件及 Xilinx 公司的 ISE 工具。

2) 设计输入工具 设计输入工具的主要功能是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、电子系统的框图及原理图等;文本编辑器在系统上用来编辑电子系统的描述语言,在其他层次上用来编辑电路的硬件描述语言文本。

3) 逻辑综合器 逻辑综合器能够自动完成将设计者在 EDA 平台上编辑输入的 HDL 文本、原理图或状态图描述,依据给定的硬件结构和约束控制条件进行编译、优化和转换,最终获得门级电路甚至更低层的电路描述网表文件的过程。

4) 仿真器 仿真器提供了对设计进行模拟仿真的手段,包括布线以前的功能仿真和布线以后的包括延时的时序仿真。在一些复杂的设计中,仿真比设计本身还要艰巨。仿真器的仿真速度、仿真的准确性、易用性等成为衡量仿真器性能的重要指标。按照处理 HDL 语言的类型,仿真器可以分为 Verilog HDL 仿真器、VHDL 仿真器和混合仿真器,混合仿真器能

够同时处理 Verilog HDL 和 VHDL, 如 Mentor 公司的 ModelSim、Cadence、Synopsys 软件。

5) 检查/分析工具 用来对设计者的具体设计进行编译、检查和分析, 发现设计中的错误和对可能产生的结果进行分析。它可以产生各种报表文件, 设计者可以对设计系统中的错误、系统性能及相应各种时序关系等方面有清楚的了解, 从而能及时发现设计中的错误和检查系统性能是否满足设计的需求。

6) 布局布线工具 它用来实现从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的差异。

EDA 软件工具开发厂商大体可分为两类: 一类是 EDA 专业软件公司, 较著名的有 Mentor Graphics、Cadence、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, 适合高学术性基础研究的单位使用。



1.3 CPLD 与 FPGA 器件

可编程逻辑器件 (Programmable Logic Device, PLD) 是近年发展起来的一种新型器件, 是当前数字系统设计的主要硬件基础, 是硬件描述语言 VHDL 的物理实现工具。它的应用和发展不仅简化了电路设计, 降低了开发成本, 提高了系统可靠性, 而且对数字系统设计自动化起到了极大的推动作用。

1. 可编程逻辑器件概述

自 20 世纪 60 年代集成电路诞生以来, 数字集成电路经历了 SSI、MSI、LSI、VLSI 甚至 ULSI 的阶段, 可以将它们分为两大类: 一类是标准逻辑器件, 另一类是专用集成电路 (Application Specific Integrated Circuit, ASIC)。

1) 标准逻辑器件 标准逻辑器件是具有标准逻辑功能的通用 SSI、MSI 集成电路, 如 TTL 工艺的 54/74 系列和随后发展起来的 CMOS 工艺的 CD4000 系列中的各种基本逻辑门、触发器、选择器、分配器、计数器和寄存器。

标准逻辑器件是传统“自下而上”设计方法的主要器件, 生产批量大而且成本低。但是由标准逻辑器件和其他元件组成的电子系统, 往往需要元件的种类、数量很多, 连线也复杂, 因而所设计的系统体积大、功耗大、可靠性差。

2) 专用集成电路 专用集成电路是相对于通用集成电路而言的, 它是专门针对某一领域或某一用户需要而设计制造的集成电路器件, 它可以将某些专用电路或系统设计在一个芯片上, 从而使系统的体积小、重量轻、功耗低, 而且具有高性能、高可靠性及保密性好等优点。专用集成电路设计方法按照 ASIC 版图结构和制造工厂的不同, 可以分为全定制和半定制两种。

(1) 全定制 (Full-Custom) 设计方法: 所谓全定制设计方法, 是利用人机交互图形系统, 由版图设计者针对具体电路和具体要求, 从每个器件的图形、尺寸开始设计, 直至整个版图的布局布线。它可以获得最佳的电路性能和最小的芯片尺寸, 有利于提高集成度和降低生产成本, 适用于通用芯片和高性能芯片的设计及库单元的设计。其缺点是设计周期长、设

计费用高、风险大,而且容易出错。

(2) 半定制设计方法: 所谓半定制 (Semi-Custom) 设计方法是一种约束型设计方法。按照逻辑实现方式的不同, 半定制设计方法可以分为门阵列、标准单元和可编程逻辑器件3类。

- 门阵列 (Gate Array) 设计方法: 门阵列设计方法是一种较早使用的 ASIC 设计方法。首先将含有固定器件数不含连线的内部相同单元排成一定规模的阵列, 将含有固定器件数不含连线的 I/O 相同单元排在四周, 并留有固定的布线通道, 形成一定规模、一定 I/O 端口数、没有连线 (没有功能) 的芯片版图, 这也称做“门阵列母片”。在固定规模 (器件数)、固定端口数的门阵列母片的基础上, 芯片设计者根据需要将内部单元和 I/O 单元分别进行内部连线构成所需功能的各种单元, 再进行总体布局布线, 构成一定功能的芯片连线版图。按此连线版图进行制版, 再在预先生产出的母片上继续完成后续工序, 制出最终芯片。利用门阵列方法设计的芯片面积、最大规模、最多引脚数、布线通道及单元中的器件数和部分连接是固定的, 利用率不能达到 100%, 灵活性差, 性能不能达到最佳。
- 标准单元 (Standard Cell) 设计方法: 电路的基本单元及各种 I/O 单元都按一定的标准, 依据特定工艺, 由专门人员预先设计好, 存放于一个统一的库中, 称为标准单元库。芯片设计者只要根据电路的逻辑网表及设计约束条件, 应用相关软件调用标准库中的单元进行布局布线, 即可快速形成最终的芯片版图。标准单元可获得较佳的电路性能和较小的芯片尺寸, 有利于缩短芯片设计周期, 降低设计成本, 适用于专用集成电路 (ASIC) 和较高性能的芯片设计。
- 可编程逻辑器件设计方法: 可编程逻辑器件设计方法是用可编程逻辑器件设计用户定制的数字电路系统。可编程逻辑器件的制作采用 CMOS 工艺, 在这些器件的内部, 集成了大量功能独立的分立器件。它们可以使用基本逻辑门以及由基本逻辑门组成的宏单元、与阵列、或阵列等。依据不同需求, 芯片内部元件的种类、数量可以有不同的设置。此外, 芯片内还有大量可配置的连线, 在器件出厂时, 芯片内的各个元件、单元相互间没有连接, 芯片暂时不具备任何逻辑功能。芯片内的各个元件和单元如何连接, 由用户根据自身设计的电路功能要求, 通过计算机编程决定。

2. 可编程逻辑器件分类

可编程逻辑器件的分类方式有很多种, 可以按照集成度分类、编程工艺分类及器件结构分类等, 下面介绍几种常用的分类方式。

1) 按集成度分类 根据集成度的密度不同, PLD 分为低密度 PLD 器件 (LDPLD) 和高密度 PLD 器件 (HDPLD)。常见的低密度 PLD 器件有 PROM、PLA、EPROM、E²PROM、PAL 和 GAL 器件等, 通常简称为 PLD 器件; 常见的高密度 PLD 器件有 EPLD、CPLD 及 FPGA, 如图 1-5 所示。

2) 按编程工艺分类 可编程逻辑器件按编程工艺分类可以分为以下几类。

(1) 采用熔丝 (Fuse) 器件, 早期的 PROM 器件采用的就是此结构, 根据设计熔丝图文件来烧断对应的熔丝从而达到编程的目的。但是用户只能编程一次, 一旦写入信息, 就不能再改写。

(2) 反熔丝 (Antifuse) 器件, 是对熔丝技术的改进, 在编程处通过击穿漏层, 使两点之间导通, 这与熔丝烧断获得开路正好相反。

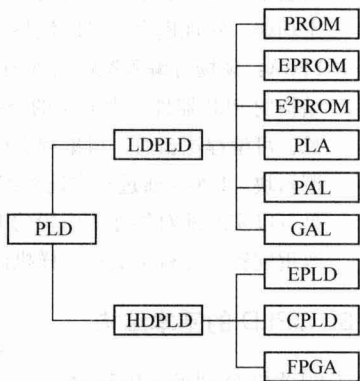


图 1-5 可编程逻辑器件按集成度分类

(3) EPROM 器件, 紫外线可擦除可编程逻辑器件, 通过对存储信息的晶体管施加较高电压, 产生高能电子来实现写入信息。

(4) E²PROM 器件, 电可擦除可编程器件, 是利用低能电子穿过氧化物的方法来实现编程和擦除的。

(5) SRAM 器件, SRAM 是数字系统的重要组成部分, 即使不同的系统也可以使用相同的 SRAM, 因此 SRAM 是一种能大量生产的标准电路, 目前嵌入式 SRAM 也占据重要地位。

(6) Flash 器件, 可以实现多次编程。

3) 按器件结构分类 目前使用的 PLD 基本上都是由输入缓冲电路、与阵列、或阵列和输出缓冲电路组成的。输入缓冲电路主要用于产生输入变量的原变量和反变量, 并增强输入信号的驱动能力; 与阵列和或阵列是核心, 与阵列用来产生乘积项, 或阵列主要用来产生乘积项之和形式的函数; 输出缓冲电路可以是组合电路输出、时序电路输出或可编程输出结构, 输出信号还可以通过内部通道反馈到输入端, 如图 1-6 所示。根据结构特点, 可以将 PLD 划分为简单 PLD、CPLD 和 FPGA。

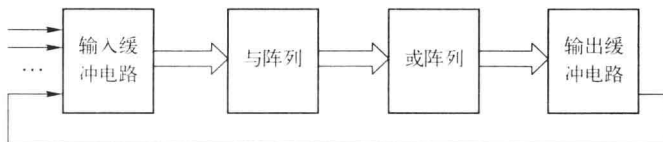


图 1-6 PLD 的基本结构

- 简单 PLD: 简单 PLD 主要是指早期的可编程逻辑器件, 由与阵列和或阵列组成, 任何组合逻辑函数均可化为“与或”表达式并用“与门-或门”二级电路实现, 而任意时序电路又都是由组合逻辑电路加上存储元件组成的。简单 PLD 能够完成大量的组合逻辑功能, 有较高的速度和较好的性能。
- CPLD: 复杂 PLD 编程逻辑器件 (Complex Programmable Logic Device), 简称 CPLD。CPLD 是由 GAL 发展而来的, 是基于乘积项结构的 PLD 器件, 可以看做是对原始可编程逻辑器件的扩充。它通常由大量的可编程逻辑宏单元围绕一个位于中心的、延时固定的可编程互联矩阵组成。其中可编程逻辑宏单元较为复杂, 具有复杂的 I/O 单元互连结构, 可根据用户需要生成特定的电路结构, 完成一定功能。它规模大, 可以代替几十个甚至上百个通用 IC, 具有高密度、高速度和低功耗等特点。
- FPGA: 现场可编程逻辑门阵列 (Field Programmable Gate Array), 简称 FPGA。FPGA 是基于查找表结构的 PLD 器件, 由简单的查找表组成可编程逻辑门, 再构成阵列形式, 通常包含 3 类可编程资源: 可编程逻辑块、可编程 I/O 块、可编程内连线。可编程逻辑块排列成阵列, 可编程内连线围绕逻辑块。FPGA 通过对内连线编程, 将逻辑块有效组合起来, 实现用户要求的特定功能。它的优点是可以实时地对外加或内置的 RAM 或 EPROM 编程, 实时地改变器件功能, 实现现场可编程或在线重配置, 是科学实验、样机研制、小批量产品生产的最佳选择器件。

3. CPLD 的基本结构

不同器件公司的 CPLD 产品结构不同, 但是任何芯片公司的 CPLD 的基本结构都应该包括 3 部分: 可编程逻辑宏单元、可编程 I/O 单元和可编程内连线。CPLD 的基本结构如图 1-7 所示。

1) 可编程逻辑宏单元 可编程逻辑宏单元由一些与、或阵列加上若干个触发器构成, 其中有一个触发器与输出端相连, 其余触发器的输出和输入不相连, 但是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复杂时序电路。CPLD 采用乘积项共享结构, 在 CPLD 的宏单元中, 如果输出表达式的与项较多, 对应的或门输入端不够用, 可以借

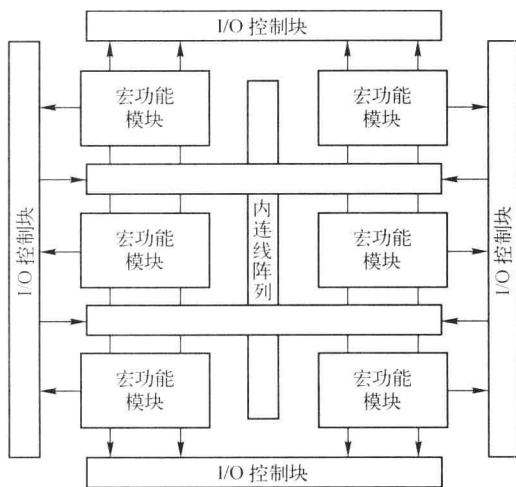


图 1-7 CPLD 的基本结构

助可编程开关将同一个单元（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，从而提高资源利用率，实现快速复杂的逻辑函数。另外，CPLD 触发器的时钟既可以工作在同步，也可以工作在异步，有些器件的时钟还可以通过数据选择器或时钟网络进行选择。

2) 可编程 I/O 单元 CPLD 的可编程 I/O 单元是内部信号 I/O 引脚的接口，它完成不同电气特性下对输入/输出信号的驱动和匹配。

3) 可编程内连线 可编程内连线（Programmable Interconnect Array, PIA），也称布线池。可编程内连线用于从 I/O 到逻辑块输入的信号布线，或者对从逻辑块输出到其自身输入或其他逻辑块的输入信号布线。有的逻辑块具有自己的内部反馈，而不需要把输出反馈到可编程内连线去。大多数 CPLD 都是采用下列两种方式之一来实现其内部可编程连线的。其一是基于阵列的内连线，其二是基于多路选择的内连线。阵列内连线方式允许 PIA 中的任何信号可布线到任何其他的逻辑块，PIA 中的每一项由垂直线来表示且被分配作为给定逻辑块的输入，从而使每个到逻辑块的输入项均为有一个 PI 项。逻辑块的输出可以通过记忆元件连接与 PI 项，器件输入也能够连接与 PI 项。

4. FPGA 的基本结构

FPGA 的结构与生产厂家有关，通常由基本可编程逻辑单元、可编程输入/输出单元、布线资源、嵌入式块 RAM、底层嵌入功能单元和内嵌专用硬核等组成，其基本结构如图 1-8 所示。

1) 基本可编程逻辑单元 基本可编程逻辑单元（Configurable Logic Block, CLB）是实现用户功能的基本单元，它们通常排列成一个阵列，散布于整个芯片，其内部的连接和配置可以根据设计灵活地改变。CLB 由查找表（Look-Up-Table, LUT）和寄存器（Register）组成。LUT 本质上就是一个 RAM， N 个输入项的逻辑函数可以用一个 2^N 位容量的 RAM 实现，函数值存放在 RAM 中，RAM 的地址线起输入线的作用，地址即输入变量值，RAM 输出为逻辑函数值，由连线开关实现与其他功能块的连接。目前，查找表一般为 4 输入，所以每一个 LUT 可以看成是一个有 4 位地址线的 16×1 的 RAM。当用户通过电路原理图和 HDL 语言描述一个逻辑电路以后，FPGA 开发软件会自动计算逻辑电路的所有可能结果，并把结果事先

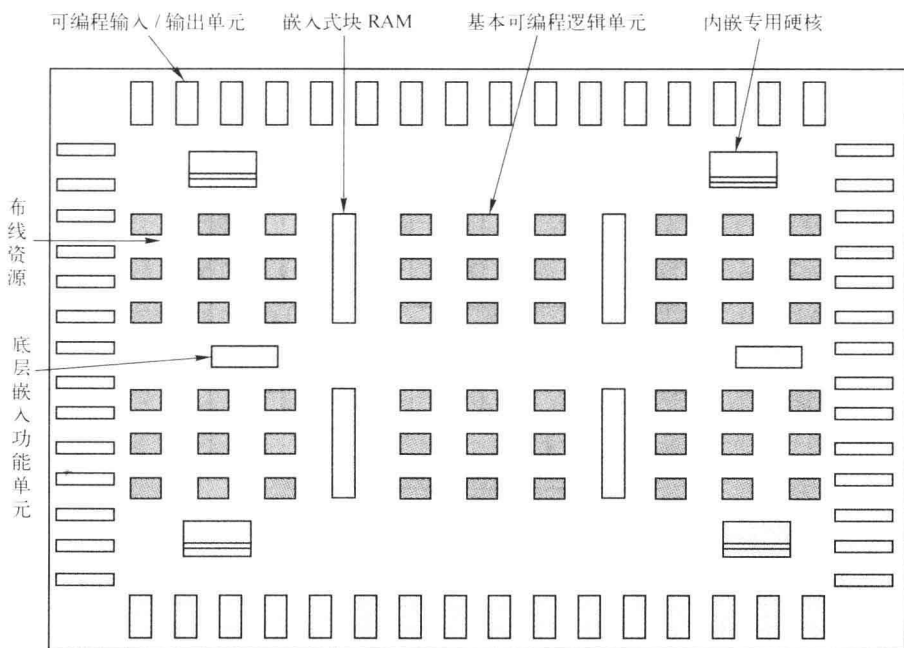


图 1-8 FPGA 的基本结构

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。

2) 可编程输入/输出单元 可编程输入/输出单元即可编程 I/O 块 (I/O Block, IOB)，完成芯片上逻辑与外部封装脚的接口，常围绕在阵列与芯片四周，可编程内部互连包括各种长度的线段和编程连接开关，它们将各个可编程逻辑模块或可编程 I/O 块连接起来，构成特定功能的电路。为了使 FPGA 有更灵活的应用，目前大多数 FPGA 的可编程 I/O 块被设计成可编程模式，即通过软件的配置，可以适配不同的电气标准与 I/O 物理特性，可以调整匹配阻抗的特性和上下拉电阻，可以调整输出驱动电流的大小等。

3) 嵌入式块 RAM (Block RAM) 目前，大多数 FPGA 内部都嵌入了可编程 RAM 模块，该模块大大扩展了 FPGA 的应用范围和使用灵活性。一般，嵌入式块 RAM 可配置单端口 RAM、双端口 RAM、伪双端口 RAM、内容地址存储器 (CAM)、先进先出 (FIFO) 等常用的存储结构。

4) 布线资源 布线资源通过 FPGA 内部所有单元，连线好的长度和工艺决定信号在连线上的驱动能力和传输速度。布线资源可分为全局性的专用布线资源、长线资源和短线资源。全局性的专用布线资源用来完成器件内部的全局时钟和全局复位/置位的布线；长线资源用来完成器件分区 (Bank) 间的一些高速信号和一些第二全局时钟信号的布线；短线资源用来完成基本逻辑单元间的逻辑互连与布线。

5) 底层嵌入功能单元 底层嵌入功能单元指的是通用程度较高的嵌入功能模块，比如 PLL、DLL、DSP、CPU 等。并非所有的 FPGA 都有底层嵌入功能单元，但是随着 FPGA 的发展，这些模块被越来越多地嵌入到其内部，以适应不同场合的需求。

6) 内嵌专用硬核 内嵌专用硬核指的是那些通用性较弱，而针对性较强的专用硬核。但是并非所有的 FPGA 器件都含有内嵌专用硬核。高端通信市场的可编程逻辑器件都有内嵌