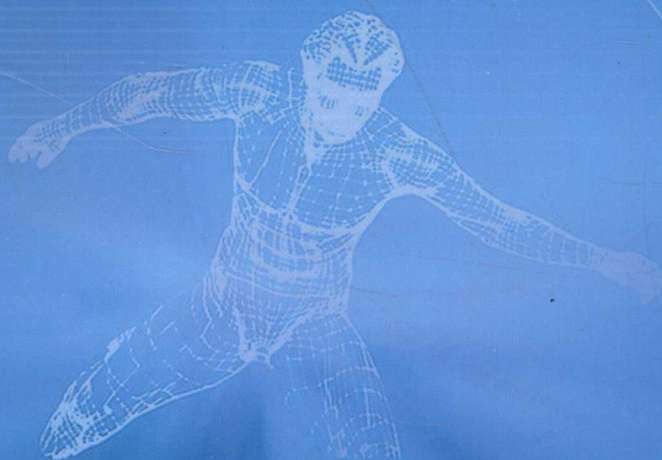


数字系统设计

方法与amp;实践

万国春 童美松◎编著



1,256,45 2,258,40 6,300,00
1,245,00 1,245,23 1,000,11
2,300,00 3,031,40 2,450,00
2,704,00 3,000,00



同济大学出版社
TONGJI UNIVERSITY PRESS

数字系统设计方法与实践

万国春 童美松 编著

 同济大学出版社
TONGJI UNIVERSITY PRESS

内 容 提 要

本书主要介绍了数字系统设计方面的设计方法及其工程应用,以工程实践为导向,借鉴国际知名大学电气电子专业的培养方法,并结合编者的教学和项目实践及多年积累的资料编写而成。参考 Xilinx 公司的官方文档,并吸收国内外相关专业技术文献的精华,提炼其核心知识体系,让每一个读者都能在本书中有所了解、掌握与提高,为培养卓越工程师奠定坚实基础。

本书根据数字系统设计课程教学的要求,以提高数字电路设计能力和创新能力为目的,主要阐述了:①基于 Xilinx FPGA 的数字系统开发相关知识;②运用 VHDL 硬件描述语言进行设计的要求并辅于设计案例;③不同难易程度的综合设计实例以培养综合设计创新能力。书中内容紧紧围绕教学与实践创新,实验设计案例具有实用性和层次化递增的特点。实验基于 Xilinx 公司的 XUP Virtex-II Pro 硬件开发平台和 ISE 10.1 软件开发平台,提供 VHDL/Verilog HDL 参考源码。

图书在版编目(CIP)数据

数字系统设计方法与实践/万国春,童美松编著.--上海:同济大学出版社,2015.10

ISBN 978-7-5608-5948-4

I. ①数… II. ①万…②童… III. ①数字系统—系统设计 IV. ①TP271

中国版本图书馆 CIP 数据核字(2015)第 193178 号

数字系统设计方法与实践

万国春 童美松 编著

责任编辑 张智中 责任校对 徐春莲 封面设计 吴炳锋

出版发行 同济大学出版社 www.tongjipress.com.cn
(地址:上海市四平路 1239 号 邮编:200092 电话:021-65985622)

经 销 全国各地新华书店

印 刷 大丰科星印刷有限责任公司

开 本 787mm×1092mm 1/16

印 张 16.5

印 数 1-3100

字 数 411000

版 次 2015 年 10 月第 1 版 2015 年 10 月第 1 次印刷

书 号 ISBN 978-7-5608-5948-4

定 价 39.00 元

前言

大规模可编程逻辑器件 FPGA/CPLD 的迅猛发展和更加广泛的应用,以及 EDA 技术在电子信息、通信、自动控制及计算机等领域的重要性与日俱增,对数字系统设计技术领域的理论教学和工程设计提出了更高的要求。为了适应 EDA 技术在高新技术行业的需求和高层次人才培养的要求,同时面向工程实践的特点,本书借鉴国内外电气电子专业的培养方法,吸收了国内各电子专业强校所编教材的精华,提炼了核心知识体系,致力于把最新的相关科学技术引入本书中,为培养卓越工程师奠定实践基础。

本书根据数字系统设计专业人才培养的要求,以提高数字电路设计能力和创新能力为目的,主要介绍了:(1)基于 Xilinx FPGA 的数字系统开发相关知识;(2)运用 VHDL 硬件描述语言进行设计的要点,并辅于设计案例;(3)不同难易程度的综合设计实例以培养综合设计创新能力。书中内容紧紧围绕教学与实践创新,实验设计案例具有实用性和层次化递增的特点。实验基于 Xilinx 公司的 XUP Virtex-II Pro 硬件开发平台和 ISE 10.1 软件开发平台,提供 VHDL/Verilog HDL 参考源码。

第一章介绍了数字系统设计自动化技术的发展历史和基于 FPGA 的数字系统设计方法流程;第二章中,对 FPGA 的结构和工作原理建立大致的了解,并简单介绍了 Xilinx 公司推出的两个 FPGA 系列以及本书所用 FPGA 的型号选择;第三章着重讲解 Xilinx ISE 软件开发平台的使用,简要介绍 ISE 中高级组件的功能概况;第五章通过常用数字电路的设计回顾运用 VHDL 语言实现组合逻辑电路、时序逻辑电路和有限状态机的设计;第五章的内容集中于 FPGA 开发设计中涉及到的设计原则和设计技巧;第七、八章递进式的介绍多个综合数字系统的设计。此外,针对 Xilinx 近几年大力推广新一代设计套件——Vivado 的情况,本书在附录一中对 Vivado 设计套件的突出特点进行了必要的介绍,并与 ISE 设计套件进行比较。附录二和附录三分别是书中所用硬件平台详细的展示。

本书的定位目标是,通过给出完整的软硬件开发流程,结合从简到难的数字系统设计案例和设计实践,让读者能够比较全面的掌握基于 FPGA 的数字系统设计方法,设计能力与创新能力得到大幅提高。

本书在编写过程中,得到了同济大学研究生院在职教育管理处、电子科学与技术系教师的关心和帮助,并提出了许多宝贵的修改意见,对于他们的帮助和支持,编者在此表示衷心感谢!

特别感谢课题组研究生谢鹏飞、尹桂珠和薛柯,他们在专业技术资料收集、整理、工程实践案例分析和验证等方面做了大量、卓有成效的专业技术工作。

由于编者水平有限,书中疏漏和不足之处在所难免,恳请广大读者批评指正。

编者
2015 年 10 月

目录

前 言

第一章 数字系统设计与 FPGA	001
1.1 数字系统设计自动化技术的发展历程	001
1.2 数字系统的设计流程	002
1.3 基于 FPGA 的数字系统设计	005
1.3.1 可编程逻辑器件的发展历史	005
1.3.2 基于 FPGA 的数字系统设计流程	006
第二章 现场可编程门阵列 FPGA	009
2.1 FPGA 的结构和工作原理	009
2.2 Xilinx 产品概述	012
2.2.1 Spartan 系列	012
2.2.2 Virtex 系列	013
2.3 FPGA 的配置	015
2.4 实验平台的选择	017
第三章 Xilinx ISE 开发套件	019
3.1 ISE 10.1 开发流程	019
3.1.1 设计输入	020
3.1.2 仿真	026
3.1.3 添加约束	028
3.1.4 综合	038
3.1.5 实现	043
3.1.6 iMPACT 编程与配置	046
3.2 ISE 高级组件	054
3.2.1 在线逻辑分析仪(ChipScope Pro)	054
3.2.2 平面布局规划器(PlanAhead)	055
3.2.3 时序分析器(Timing Analyzer)	056
3.2.4 布局规划器(Floorplanner)	061
3.2.5 功耗分析工具(XPower)	062
【设计实践】	065
3-1 ChipScope Pro 的逻辑分析实验	065

第四章 基本数字电路的 VHDL 设计	075
4.1 组合逻辑电路的 VHDL 设计	075
4.1.1 加法器	075
4.1.2 多路选择器	079
4.1.3 编码器与译码器	082
【设计实践】	085
4-1 快速加法器的设计	085
4-2 4×4 乘法器的设计	087
4-3 ChipScope Pro 的 VIO 实验	089
4.2 时序电路的 VHDL 设计	093
4.2.1 基础时序元件	095
4.2.2 计数器的 VHDL 设计	100
4.2.3 堆栈与 FIFO	103
4.2.4 多边沿触发问题	107
【设计实践】	110
4-4 奇数与半整数分频器设计	110
4-5 DCM 模块设计实例	112
4.3 有限状态机的 VHDL 设计	120
4.3.1 VHDL 状态机的一般形式	120
4.3.2 有限状态机的一般设计方法	127
4.3.3 有限状态机的 VHDL 描述	129
【设计实践】	135
4-6 交通灯控制器	135
4-7 乒乓游戏设计	138
第五章 FPGA 开发设计方法	142
5.1 FPGA 系统设计的基本原则	142
5.1.1 面积与速度的平衡互换原则	142
5.1.2 硬件可实现原则	143
5.2 FPGA 中的同步设计	144
5.3 FPGA 中的时钟设计	145
5.3.1 全局时钟	145
5.3.2 门控制时钟	146
5.3.3 多级逻辑时钟	149
5.3.4 行波时钟	150
5.3.5 多时钟系统	151
5.4 FPGA 系统设计的常用技巧	153
5.4.1 乒乓操作	153
5.4.2 串并/并串转换	155

5.4.3 流水线设计	155
【设计实践】.....	158
5-1 32 位流水线加法器的设计	158
第六章 综合设计实例	161
6.1 数码管扫描显示电路	161
6.2 八位除法器的设计	166
6.3 Virtex-II Pro 的 SVGA 显示控制器设计	171
第七章 数字系统综合实验	182
7.1 数字时钟设计	182
7.2 直接数字频率合成技术(DDS)的设计与实现	186
7.3 音乐播放器实验	191
7.4 基于FPGA的FIR数字滤波器的设计	202
7.5 数字下变频器(DDC)的设计	207
第八章 CPU 设计.....	211
附录一 Vivado 设计套件	232
附录 1.1 单一的、共享的、可扩展的数据模型	232
附录 1.2 标准化 XDC 约束文件——SDC	233
附录 1.3 多维度解析布局器	234
附录 1.4 IP 封装器、集成器和目录	235
附录 1.5 Vivado HLS 把 ESL 带入主流	237
附录 1.6 其他特性	238
附录二 XUP Virtex-II Pro 开发系统的使用	240
附录 2.1 Virtex-II Pro FPGA 主芯片介绍	241
附录 2.2 电源供电模块	241
附录 2.3 时钟电路	242
附录 2.4 SVGA 视频模块	243
附录 2.5 AC97 音频解码模块	244
附录 2.6 RS232 串行接口模块	245
附录 2.7 PS2 接口模块	246
附录 2.8 开关、按键和 LED 指示灯	247
附录 2.9 下载配置模块	247
附录 2.10 高速和低速的扩展连接器	248
附录三 通用型开发板底板普及板 V11.0.1 的使用.....	251
参考文献	253

第一章 数字系统设计与 FPGA

1.1 数字系统设计自动化技术的发展历程

在数字系统设计自动化技术发展之初(20世纪70年代),出现了借助电子计算机辅助设计(Computer Aided Design, CAD)集成电路和数字系统的软件产品,其功能主要是大规模集成电路(Large Scale Integration, LSI)布线设计和印刷电路板(Printed Circuit Board, PCB)布线设计,它使用二维图形编辑和分析工具代替传统的手工布图线方法,将设计人员从重复性的繁杂劳动中解放出来,使工作效率和产品设计的复杂程度大大提高。人们把这种技术称之为计算机辅助设计技术。

20世纪80年代,出现了第二代电路CAD软件,其产品主要是交互式逻辑图编辑工具、逻辑模拟工具、LSI和PCB自动布局布线工具,它可以使设计人员在产品的设计阶段对产品的性能进行分析,验证产品的功能,并且生成产品制造文件。这一时期的电路CAD工具已不单单是代替设计工作中绘图的重度劳动,而是具有一定的设计功能,可以代替设计人员的部分设计工作,人们称之为计算机辅助工程(Computer Aided Engineering, CAE)技术。

20世纪80年代末至20世纪90年代初,随着电路CAD技术的不断发展,融合了计算机辅助制造(Computer Aided Manufacturing, CAM)、计算机辅助测试(Computer Aided Translation, CAT)和计算机辅助工程等概念,形成了第三代电路CAD系统,也就是电子设计自动化(Electronic Design Automation, EDA)这一概念。这一时期EDA工具的主要功能是以逻辑综合、硬件行为仿真、参数分析和测试为重点。

数字系统设计自动化技术的发展历程如图1.1所示。

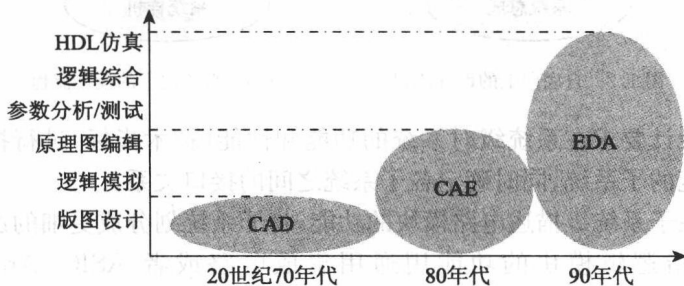


图 1.1 20 世纪数字系统设计自动化技术的发展历程

目前流行的 EDA 工具门类齐全、种类繁多,主要构成为:设计输入模块、设计数据库模块、综合模块、分析验证模块和布局布线模块,它能够在算法级、寄存器传输级(RTL)、门级

和电路级进行设计描述、综合与仿真。

另外,EDA 工具与前两代电路 CAD 产品的重要差别之一是,不仅可以用逻辑图进行设计描述,还可以用文字硬件描述语言进行设计描述,以及用图文混合方式进行设计描述。

1.2 数字系统的设计流程

数字系统的设计从设计方法学角度来讲,有自顶向下(TOP-DOWN)和自底向上(DOWN-UP)两种方法。

由于 EDA 工具首先是在低层次上得到发展的,所以 DOWN-UP 设计方法曾经被广泛应用,这种方法以门级单元库和基于门级单元库的宏单元库为基础,从小模块逐级构造大模块以至整个电路,其设计流程如图 1.2 所示。在设计过程中,任何一级发生错误,往往都会使得设计重新返工。因此,自底向上的设计方法效率和可靠性低、设计成本高。

随着 EDA 技术的不断发展, TOP-DOWN 设计方法目前得到越来越广泛的应用。按照 TOP-DOWN 设计思路,数字系统的设计流程可分为这样几个层次:系统设计、模块设计、器件设计和版图设计,如图 1.3 所示。

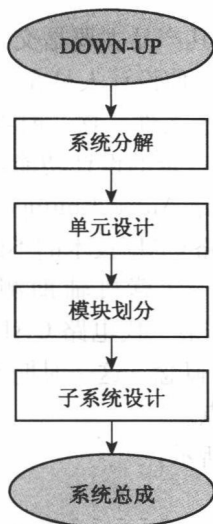


图 1.2 自底向上的设计流程

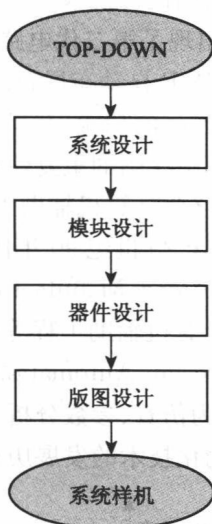


图 1.3 自顶向下的设计流程

系统设计将设计要求在系统级对系统的功能和性能(技术指标)进行描述,并将系统划分成实现不同功能的子系统,同时确定各子系统之间的接口关系。

模块设计是在子系统级描述电路模块的功能,将子系统划分成更细的逻辑模块。

器件设计是指逻辑模块的功能用通用集成电路或者 ASIC (Application Specific Integrated Circuit)来实现,如果采用 ASIC 实现方案,则还需对 ASIC 的内部逻辑和外接引脚的功能进行定义。在以通用集成电路作为主要硬件构成的设计中,器件设计主要解决元器件的选用问题,因而模块设计所占比例很大,器件设计工作相对较少。而在以 ASIC 作为主要硬件构成的设计中,器件设计也就是 ASIC 设计,因此器件设计占了很大的比例。ASIC

的采用使得模块设计工作大部分是在器件设计中完成的,即模块设计延伸到器件设计工作中,使得这两部分设计工作的分界线不那么明显了。

版图设计包含 ASIC 芯片版图设计和 PCB 版图设计。

ASIC 版图设计包括芯片物理结构分析、逻辑分析、建立后端设计流程、版图布局布线、版图编辑、版图物理验证等设计工作,这些工作可以融合到上一层次的器件设计中。如果采用 CPLD (Complex Programmable Logic Device) 和 FPGA (Field Programmable Gate Array) 作为 ASIC 设计的实现手段,则芯片版图设计工作将可大为简化。

PCB 版图设计则是按照系统设计要求,确定电路板的物理尺寸,并进行元器件的布局和布线,从而完成系统样机的整体功能。无论采用通用集成电路还是 ASIC 作为主要硬件构成,PCB 版图设计工作都是不可或缺的,特别是高速数字系统,更是决定系统设计成败的重要一环。

在这里要特别提出的是 CPLD/FPGA 的系统编程功能,它可以在完成 PCB 设计和焊接工作之后,重新修改可编程逻辑器件(Programmable Logic Device, PLD)的内部逻辑,这使得数字系统设计更为灵活和方便。

在上述各层次的设计中,主要有描述、划分、综合和验证 4 种类型的工作,这些工作贯穿于整个设计的各个层次。首先在高级别层次进行描述、验证,然后经过划分和综合,将高级别的描述转换至第一级别的描述,在经过验证、划分和综合,将设计工作向更低级别延伸。

下面分别介绍这 4 种类型的设计工作。

1. 描述

指用文字(例如硬件描述语言 VHDL、Verilog HDL 等)、图形(例如真值表、状态图、逻辑电路图、PCB 或芯片版图)或者二者结合来描述不同设计层次的功能,主要有几何描述、结构描述、RTL 描述和行为描述 4 种描述方法。

(1) 几何描述

几何描述主要是指集成电路芯片版图后者 PCB 版图的几何信息。这些信息可以用物理尺寸表达,也可以用符号来表达;可以用图形方式描述,也可以掩膜网表文件的形式存在。

(2) 结构描述

结构描述表示一个电路的基本元件构成以及这些基本元件之间的相互连接关系,它可以用文字表达,也可以用图形来表达;可以在电路级,也可以在门级进行结构描述。电路级的基本元件是晶体管、电阻、电容等,电路级描述表达了这些基本元件的互连关系;门级的基本元件是各种逻辑门和触发器,门级描述表达了这些基本元件之间的互连关系,即逻辑电路的结构信息。除了使用图形描述方式之外,结构描述的信息还可以用门级网表的形式存放在网表(Net List)文件中。

(3) RTL 描述

RTL 描述表示信息在一个电路中的流向,即信息是如何从电路的输入端,经过何种变换,最终流向输出端的。RTL 的基本元件是寄存器、计数器、多路选择器、存储器、算术逻辑单元(ALU)和总线等宏单元,RTL 描述表达了数据流在宏单元之间的流向,因此也称为数据流描述。与此同时,RTL 描述还隐含了宏单元之间的结构信息,所以一个正确的 RTL 描述可以被直接转换或综合为结构描述(即门级)网表的形式。

(4) 行为描述

行为描述表示一个电路模块输入信号和输出信号之间的相互关系,也可以用文字或者图形两种形式来表达。算法级描述是对 RTL 之上的模块电路的行为描述,行为描述不包含模块电路的结构信息,所以不能用以模块电路为基本元件的图形来表达。通常采用真值表、状态图或硬件描述语言等形式来描述模块电路的输入信号与输出信号之同的对应关系。即使是一个正确的行为描述,也不一定能够被转换或综合为可以用逻辑门电路实现的形式。也就是说,不一定能够被综合成一个正确的 RTL 描述。

2. 划分

划分是在不同的设计层次,将大模块逐级划分成小模块的过程,它可以有效降低设计的复杂性、增强可读性。在划分模块时应注意以下几点:

(1) 在同一层次的模块之间,尽量使模块的结构匀称,这样可以减少在资源分配上的差异,从而有效避免系统在性能上的瓶颈。

(2) 尽量减少模块之间的接口信号线。在信号连接最少的地方划分模块,使模块之间用最少的信号线相连,以减少由于接口信号复杂而引起的设计错误和布线困难。

(3) 划分模块的细度应适合描述。如果用硬件描述语言 HDL 描述模块的行为,可以划分到算法一级;用逻辑图来描述模块,则需要划分到门、触发器和宏模块一级。

(4) 对于功能相似的逻辑模块,应尽量设计成共享模块。这样可以改善设计的结构化特性,减少需要设计的模块数量,提高模块设计的可重用性。

(5) 划分时尽量避免考虑与器件有关的特性,使设计具有可移植性,即可以在不同的器件上实现(例如采用不同制造工艺,或者不同制造方法等)设计。

3. 综合

综合是将高层次的描述转换至低层次描述的过程。综合可以在不同的层次上进行,通常分为 3 个层次:行为综合、逻辑综合、版图综合。

(1) 行为综合

行为综合是将算法级的行为描述转换为寄存器传输级描述的过程,这样不必通过人工改写就可以较快地得到 RTL 描述。因此可以缩短设计周期,提高设计速度,并且可以在不同的设计方案中,寻求满足目标集合和约束条件但花费最少的设计方案。

(2) 逻辑综合

逻辑综合是在标准单元库和特定设计约束(例如面积、速度、功耗、可测性等)的基础上,把 RTL 描述转换成优化的门级网表的过程。首先将 RTL 描述转换成由各种逻辑门(反相器、与非门、触发器或锁存器等)组成的结构描述,然后对其进行逻辑优化,再依照所选工艺的工艺库参数,将优化后的结构描述映射到实际的逻辑门电路——门级网表文件中。逻辑综合将给出满足 RTL 描述的逻辑电路(门级网表),它可以分为组合逻辑电路综合和时序逻辑电路综合两大类。

(3) 版图综合

版图综合是将门级网表转换为 ASIC 或者 PCB 版图的布局布线表述,并生成版图文件的过程。

4. 验证

验证是对功能描述和综合的结果是否能够满足设计功能要求进行模拟分析的过程。如果验证的结果不能满足要求,则必须对该层次的功能描述进行修正,甚至可能需要修改更高

层次的功能描述和划分,直到验证的结果满足设计功能的要求为止。

验证的目的主要有以下 3 个方面:

- 验证原始描述的正确性。
- 验证综合结果的逻辑功能是否符合原始描述。
- 验证综合结果中是否含有违反设计规则的错误。

验证方法通常有 3 种:逻辑模拟(也称仿真)、规则检查和形式验证。

1.3 基于 FPGA 的数字系统设计

随着集成电路深亚微米工艺技术的发展,FPGA 器件及其应用获得了长足的发展,FPGA 器件的单片规模大大扩展,系统运行速度不断提高,功耗不断下降,价格大幅度调低。因此,与传统电路设计方法相比,利用 FPGA/CPLD 进行数字系统的开发具有功能强大、投资小、周期短、便于修改及开发工具智能化等特点。并且随着电子工艺不断改进,低成本高性能的 FPGA/CPLD 器件推陈出新,促使了 FPGA/CPLD 成为当今硬件设计的首选方式之一。熟练掌握 FPGA/CPLD 设计技术已经是电子设计工程师的基本要求。

电子设计自动化(EDA)技术是以计算机为工作平台,融合了应用电子技术、计算机技术、智能化技术最新成果而开发出来的一套先进的电子系统设计的软件工具。集成电路设计技术的进步也对 EDA 技术提出了更高的要求,大大地促进了 EDA 技术的发展。以高级语言描述、系统仿真和综合技术为特征的 EDA 技术,代表了当今电子设计技术的最新发展方向。EDA 设计技术的基本流程是设计者按照“自顶而下”的设计方法,对整个系统进行方案设计和功能划分。电子系统的关键电路一般用一片或几片专用集成电路(ASIC)实现,采用硬件描述语言(HDL)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。这种被称为高层次的电子设计方法,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辅助性工作,将精力集中于创造性的方案与概念的构思上。近年来的 EDA 技术主要有以下特点:

- (1) 采用行为级综合工具,设计层次由 RTL 级上升到了系统级;
- (2) 采用硬件描述语言描述大规模系统,使数字系统的描述进入抽象层次;
- (3) 采用布局规划(planning)技术,即在布局布线前对设计进行平面规划,使得复杂 IC 的描述规范化,做到在逻辑综合早期设计阶段就考虑到物理设计的影响。

从某种意义上讲,FPGA 和 EDA 技术的发展,将会进一步引起数字系统设计思想和方法的革命。正是在这样的技术发展背景下,为了配合数字系统设计课程教学,本书主要讨论基于 FPGA 器件来实现数字系统,所有设计实验课题在 Xilinx 公司的 XUP Virtex-II Pro 开发平台上实现,不过少量实验课题还需要读者自己制作扩展板。

1.3.1 可编程逻辑器件的发展历史

可编程逻辑器件(PLD)是 20 世纪 70 年代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硬件平台。PLD 的应用和发展简化了电路设计,缩短了系统设计的周期,提高了系统的可靠性并降低了成本,因此获得了广大硬件工程师的青睐,形成了巨大的 PLD

产业规模。

20 世纪 70 年代初到 70 年代中期为 PLD 的第一阶段,这个阶段只有简单的可编程只读存储器(PROM)、紫外线可擦除只读存储器(EPROM)和电可擦除只读存储器(EEPROM)3 种。由于结构的限制,它们只能完成简单的数字逻辑功能。

20 世纪 70 年代中期到 80 年代中期为 PLD 的第二阶段,这个阶段出现了结构上稍微复杂的可编程阵列逻辑(Programmable Array Logic, PAL)和通用阵列逻辑(Generic Array Logic, GAL)器件,正式被称为 PLD,能够完成各种逻辑运算功能。典型的 PLD 由“与”、“或”阵列组成,用“与或”表达式来实现任意组合逻辑,所以 PLD 能以乘积和形式完成大量的逻辑组合。

20 世纪 80 年代中期到 90 年代末为 PLD 的第三阶段,这个阶段 Xilinx 和 Altera 分别推出了与标准门阵列类似的 FPGA 和类似于 PAL 结构的扩展型 CPLD,提高了逻辑运算的速度,具有体系结构和逻辑单元灵活、集成度高和适用范围宽等特点,兼容了 PLD 和通用门阵列的优点,能够实现超大规模的电路,编程方式也很灵活,成为产品原型设计和中小规模(一般小于 10 000 门)产品生产的首选。在这个阶段,CPLD、FPGA 器件在制造工艺和产品性能上都获得长足的发展,达到了 0.18 μm 工艺和百万门的规模。

20 世纪 90 年代末到目前为 PLD 的第四阶段,这个阶段出现了 SoPC(System on Programmable Chip)和 SoC(System on Chip)技术,是 PLD 和 ASIC 技术融合的结果,涵盖了实时化数字信号处理技术、高速数据收发器、复杂计算和嵌入式系统设计技术的全部内容。Xilinx 和 Altera 也推出了相应的 SoC FPGA 产品,制造工艺达到 45 nm 水平,系统门数也超过千万门。并且,这一阶段的逻辑器件内嵌了硬核高速乘法器、高速串行接口,时钟频率高达 500 MHz 的 PowerPC 微处理器与软核 MicroBlaze/PicoBlaze 相结合。它已超越了 ASIC 器件的性能和规模,也超越了传统意义上 FPGA 的概念,使 PLD 的应用范围从单片扩展到系统级。目前,基于 PLD 片上可编程的概念仍在进一步向前发展。

1.3.2 基于 FPGA 的数字系统设计流程

数字系统设计发展至今天,需要利用多种 EDA 工具进行设计,了解并熟悉其设计流程应成为当今电子工程师的必备知识。FPGA 是在 PAL、GAL、CPLD 等可编程器件的基础上进一步发展的产物。它是作为 ASIC 领域中的一种半定制电路而出现的,既解决了定制电路的不足,又克服了原有可编程器件门电路的缺点。FPGA 开发的一般流程如图 1.4 所示,包括电路设计、设计输入、功能仿真、综合、综合后仿真、实现与布局布线、时序仿真、芯片编程与调试等主要步骤。

1. 电路设计

在系统设计之前,首先要进行方案论证、系统设计和 FPGA 芯片选择等准备工作。系统设计工程师根据任务要求,如系统的指标和复杂度,对工作速度和芯片本身各种资源、成本等方面的要求进行权衡,选择合理的设计方案和合适的器件类型。一般都采用自顶而下的设计方法,把系统分成若干个子系统,再把每个子系统划分为若干个功能模块,直至分成基本模块单元电路为止。

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)与原理图输入等。

原理图输入方式是一种最直接的描述方式,这种方法虽然直观并易于仿真,但效率很低,且不易维护,不利于模块构造和重用。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。

HDL 设计方式是目设计大规模数字系统的最好形式,其主流语言有 IEEE 标准中的 VHDL 与 VerilogHDL。HDL 语言在描述状态机、控制逻辑、总线功能方面较强,用其描述的电路能在特定综合器作用下以具体硬件单元较好地实现。HDL 主要特点有:语言与芯片工艺无关,利于自顶向下设计,便于模块的划分与移植,可移植性好,具有很强的逻辑描述和仿真功能,而且输入效率很高。

近年来出现的图形化 HDL 设计工具,可以接收逻辑结构图、状态转换图、数据流图、控制流程图及真值表等输入方式,并通过配置的翻译器将这些图形格式转化为 HDL 文件,如 Mentor Graphics 公司的 Renoir, Xilinx 公司的 Foundation Series 都带有将状态转换图翻译成 HDL 文本的设计工具。

另外,FPGA 厂商软件与第三方软件设有接口,可以把第三方设计文件导入进行处理。如 Foundation 与 Quartus 都可以把 EDIF 网表作为输入网表而直接进行布局布线,布局布线后,可再将生成的相应文件交给第三方进行后续处理。

3. 功能仿真

功能仿真也称为前仿真,是在编译之前对用户所设计的电路进行逻辑功能验证,此时的仿真没有延迟信息,仅对初步的功能进行检测。仿真前,要先利用波形编辑器或 HDL 等工具建立波形文件和测试向量(即将所关心的输入信号组合成序列),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。常用的工具有 ModelTech 公司的 ModelSim、Synopsys 公司的 VCS 和 Cadence 公司的 NC-Verilog、NC-VHDL 等软件。

4. 综合

综合(Synthesis)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,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:将设计输入编译成由门电路、RAM、触发器等基本逻辑单元组成的逻辑连接网表。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,最后获得的结构与综合器的工作性能有关。

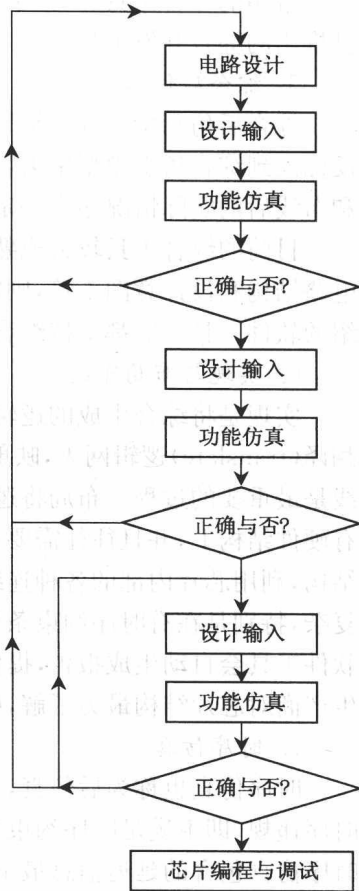


图 1.4 FPGA 一般开发流程

常用的综合工具有 Synplicity 公司的 Synplify、Synplify Pro 软件和各个 FPGA 厂家自己推出的综合开发工具。

5. 综合后仿真

综合后仿真检查综合结果是否与原设计一致。在仿真时,把综合生成的标准延时文件反标注到综合仿真模型中去,可估计门延时带来的影响。但这一步骤不能估计线延时,因此和布线后的实际情况还有一定的差距,并不十分准确。

目前的综合工具较为成熟,对于一般的设计可以省略这一步,但如果在布局布线后发现电路结构和设计意图不符,则需要回溯到综合后仿真来确认问题的来源。在功能仿真中介绍的软件工具一般都支持综合后仿真。

6. 实现与布局布线

实现是将综合生成的逻辑网表配置到具体的 FPGA 芯片上。实现主要分为 3 个步骤:翻译(translate)逻辑网表,映射(map)到器件单元,布局布线(place & route)。其中,布局布线是最重要的过程。布局将逻辑网表中的硬件原语和底层单元合理地配置到芯片内部的固有硬件结构上,并且往往需要在速度最优和面积最优之间作出选择。布线根据布局的拓扑结构,利用芯片内部的各种连线资源,合理正确地连接各个元件。目前,FPGA 的结构非常复杂,特别是在有时序约束条件时,需要利用时序驱动的引擎进行布局布线。布线结束后,软件工具会自动生成报告,提供有关设计中各部分资源的使用情况。由于只有 FPGA 芯片生产商对芯片结构最为了解,所以布局布线必须选择芯片开发商提供的工具。

7. 时序仿真

时序仿真也称为后仿真,是指将布局布线的延时信息反标注到设计网表中来检测有无时序违规(即不满足时序约束条件或器件固有的时序规则,如建立时间、保持时间等)现象。时序仿真包含的延迟信息最全,也最精确,能较好地反映芯片的实际工作情况。由于不同芯片的内部延时不一样,不同的布局布线方案也给延时带来不同的影响。因此在布局布线后,通过对系统和各个模块进行时序仿真,分析其时序关系,估计系统性能,以及检查和消除竞争冒险是非常有必要的。

8. 芯片编程与调试

设计开发的最后步骤就是在线调试或者将生成的配置文件写入芯片中进行测试。芯片编程配置是在功能仿真与时序仿真正确的前提下,将实现与布局布线后形成的位流数据文件(bitstream generation)下载到具体的 FPGA 芯片中。FPGA 设计有两种配置形式:一种是直接由计算机经过专用下载电缆进行配置,另一种则是由外围配置芯片进行上电时自动配置。因为 FPGA 具有掉电信息丢失的性质,所以可在验证初期使用电缆直接下载位流文件,如有必要再将文件烧录配置于芯片中(如 Xilinx 的 XC18V 系列,Altera 的 EPC2 系列)。

将位流文件下载到 FPGA 器件内部后进行实际器件的物理测试即为电路验证,当得到正确的验证结果后就证明了设计的正确性。

第二章 现场可编程门阵列 FPGA

2.1 FPGA 的结构和工作原理

可编程逻辑器件(PLD)可由用户根据自己需求来构造逻辑功能,具有并行处理和在线系统编程的灵活性,已成为数字系统设计的主流平台之一,同时是实现 ASIC 逻辑的一种主要方式。PLD 由早期的低密度 PAL 和 GAL 器件发展到目前 CPLD 和 FPGA 产品。

CPLD 采用可编程与阵列和固定或阵列结构,基于 EEPROM 编程技术,产品具有高密度、高速度、低功耗、低价格等优点;下载代码烧写到 CPLD 内部后可永久保持,即使系统掉电也可保持相应的逻辑功能,硬件电路简单;CPLD 内部逻辑的单元间的连线为连续式布线,信号延迟时间可预测。

FPGA 内部资源丰富,并可嵌入微处理器核,适用于数字信号处理及复杂逻辑控制系统。由于 FPGA 采用 SRAM 技术制造,上电后,配置代码写入 FPGA 片内 SRAM 中,配置 FPGA 内逻辑单元,实现对应的逻辑功能。掉电后,SRAM 存储数据丢失,FPGA 内已配置的逻辑关系消失。所以,需要外加一片专用代码配置存储芯片,每次上电时,FPGA 从存储芯片读取配置数据并写入片内 SRAM 中。

目前主流的 FPGA 仍是基于查找表(Look-Up Table, LUT)技术,且多采用 4 输入查找表。4 输入查找表可视为具有 4 根地址线,容量为 $16 \times 1\text{bit}$ 的 RAM,根据用户 HDL 代码描述,EDA 软件将 4 输入逻辑函数真值表写入 RAM,4 输入变量连接到 RAM 的 4 根地址线,当输入信号取值变化时,只需从地址对应的存储单元中读取数值,无需重构逻辑电路,即可实现任意 4 输入变量函数的逻辑功能。4 输入查找表逻辑符号如图 2.1 所示。若使用 4 输入 LUT 实现函数 $F=ABCD$,地址 A, B, C, D 与存储单元数据关系如表 2.1 所列,LUT 内部存储结构如图 2.2 所示。复杂逻辑函数可使用多个 4 输入查找表和多路选择器设计实现。

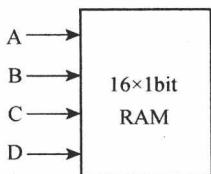


图 2.1 4 输入查找表逻辑符号

表 2.1 地址与存储单元数据关系

地址(ABCD)	RAM 存储单元数据
0 000	0
0 001	0
0 010	0
⋮	0
1 110	0
1 111	1

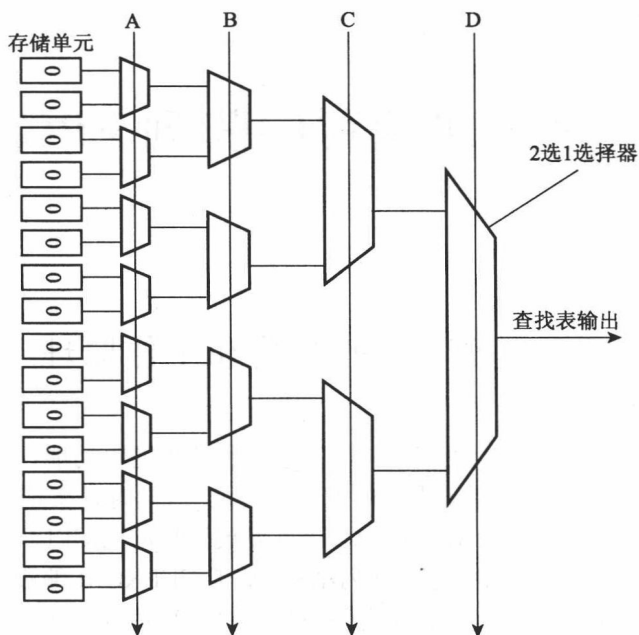


图 2.2 查找表内部结构

一般，FPGA 整合了如 DRAM、时钟管理和 DSP 等常用功能的硬核模块，其内部结构示意图如图 2.3 所示。FPGA 芯片主要由 6 部分组成，分别为可编程输入输出单元(I/O)、基本可编程逻辑单元(CLB)、完整的时钟管理模块(Digital Clock Manager, DCM)、嵌入块式存储器模块(DRAM)、可编程互联资源(IR)、内嵌的底层功能单元和内嵌专用硬件模块。

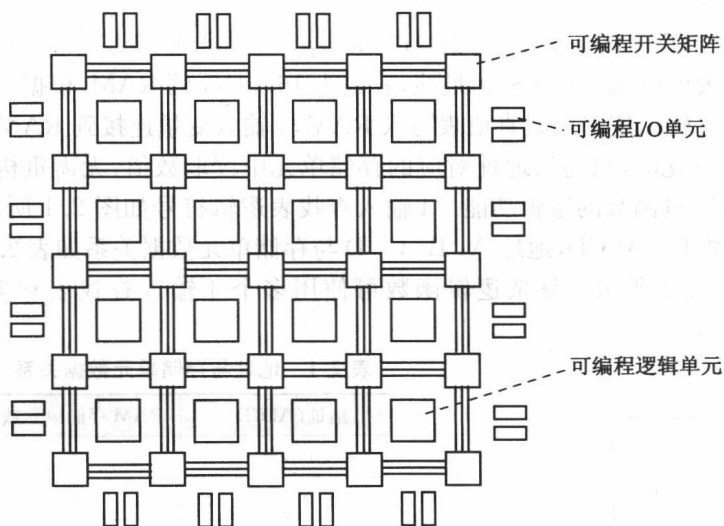


图 2.3 FPGA 内部结构

1. 基本可编程逻辑单元(CLB)

CLB 是 FPGA 内的基本逻辑单元，CLB 的实际数量和特性依器件的不同而不同。在