



应用型本科 电子及通信工程专业“十三五”规划教材



硬件描述语言 与FPGA设计技术

肖闽进 主编 ◆

董德存 主审 ◆



西安电子科技大学出版社
<http://www.xduph.com>

应用型本科 电子及通信工程专业“十三五”规划教材

硬件描述语言与 FPGA 设计技术

主 编 肖闽进

副主编 赵泓扬 邹 全

主 审 董德存

西安电子科技大学出版社

内 容 简 介

硬件描述语言与 FPGA 设计技术是电子电气及通信信息类专业的一门重要的基础课程。本书主要介绍了在 EDA 软件平台上,用硬件描述语言 VHDL 完成设计文件,然后由计算机自动地完成逻辑编译、化简、分割、综合、优化、布局、布线,之后再通过仿真验证设计项目完成情况,直至掌握对特定目标芯片的适配编译、逻辑映射和编程下载等的工作流程和实现方法。

本书以工程实例为导向,按照理论与实践相结合的要求,对 FPGA 应用中的理论、方法和设计技巧作了全面深入的讨论,大部分实例在软件设计环境 Quartus II 和 Cyclone III 系列 FPGA 器件上实现。

本书可以作为电子电气及通信信息类专业本科学生的教科书,也可以作为相关领域的工程技术人员的参考书。

图书在版编目(CIP)数据

硬件描述语言与 FPGA 设计技术/肖闽进主编. —西安:西安电子科技大学出版社, 2015. 8

应用型本科电子及通信工程专业“十三五”规划教材

ISBN 978 - 7 - 5606 - 3744 - 0

I. ① 硬… II. ① 肖… III. ① 硬件描述语言—高等学校—教材 ② 可编程逻辑器件—系统设计—高等学校—教材 IV. ① TP312 ② TP332.1

中国版本图书馆 CIP 数据核字(2015)第 165282 号

策划编辑 高 樱

责任编辑 许青青 亢列梅

出版发行 西安电子科技大学出版社(西安市太白南路 2 号)

电 话 (029)88242885 88201467 邮 编 710071

网 址 www.xduph.com 电子邮箱 xdupfxb001@163.com

经 销 新华书店

印刷单位 陕西华沐印刷科技有限责任公司

版 次 2015 年 8 月第 1 版 2015 年 8 月第 1 次印刷

开 本 787 毫米×1092 毫米 1/16 印张 14.5

字 数 333 千字

印 数 1~3000 册

定 价 25.00 元

ISBN 978 - 7 - 5606 - 3744 - 0/TP

XDUP 4036001 - 1

*** 如有印装问题可调换 ***

本社图书封面为激光防伪覆膜,谨防盗版。

应用型本科信息工程类专业系列教材

编审专家委员名单

主 任:沈卫康(南京工程学院 通信工程学院 院长/教授)

副主任:张士兵(南通大学 电子信息学院 副院长/教授)

陈 岚(上海应用技术学院 电气与电子工程学院 副院长/教授)

宋依青(常州工学院 计算机科学与工程学院 副院长/教授)

张明新(常熟理工学院 计算机科学与工程学院 副院长/教授)

成 员:(按姓氏拼音排列)

陈 奎(徐州工程学院 信电工程学院 教研室主任/副教授)

陈美君(金陵科技学院 网络与通信工程学院 副院长/副教授)

高 尚(江苏科技大学 计算机科学与工程学院 副院长/教授)

李文举(上海应用技术学院 计算机科学学院 副院长/教授)

梁 军(三江学院 电子信息工程学院 副院长/副教授)

潘启勇(常熟理工学院 物理与电子工程学院 副院长/副教授)

温宏愿(南京理工大学泰州科技学院 电子电气工程学院 副院长)

任建平(苏州科技学院 电子与信息工程学院 副院长/教授)

孙霓刚(常州大学 信息科学与工程学院 副院长/副教授)

盛党红(南京工程学院 自动化学院 书记/教授)

王杰华(南通大学 计算机科学与技术学院 副院长/副教授)

王章权(浙江树人大学 信息科技学院 副院长/副教授)

严云洋(淮阴工学院 计算机工程学院 院长/教授)

杨俊杰(上海电力学院 电子与信息工程学院 副院长/教授)

于继明(金陵科技学院 智能科学与控制工程学院 副院长/副教授)

前 言

目前,在电子技术设计领域,大规模可编程逻辑器件(FPGA/CPLD)已得到广泛的普及应用,这些器件为数字系统的设计带来了极大的灵活性。这些器件可以通过软件编程对其硬件结构和工作方式进行重构,从而使得硬件的设计可以如同软件设计那样方便快捷。这一切极大地改变了传统的数字系统设计方法、设计过程和设计观念,促进了 EDA 技术的迅速发展。

EDA 技术与应用是电子电气及通信信息类专业的一门重要的专业基础课程,涉及数字逻辑电路的基本理论、硬件描述语言、FPGA/CPLD 应用原理与技术。

本书在内容上突出基础性、实用性和先进性,注重理论与实践结合,培养读者的知识运用及创新能力。

全书共分 8 章。第 1 章 EDA 技术概述,主要介绍硬件描述语言,EDA 设计流程及 Quartus II 设计环境。第 2 章可编程逻辑器件硬件基础,主要讨论可编程器件结构与分类,CPLD/FPGA 的结构原理,重点介绍 CPLD/FPGA 编程与配置方法。第 3 章 VHDL 组合电路设计,通过常用组合电路的 VHDL 描述实例,介绍了 VHDL 基本语法现象,讨论了数据类型、实体与结构、端口、变量、进程及操作符等用法。第 4 章 VHDL 时序电路设计,介绍了基本时序电路的 VHDL 语言描述,讨论了不完整条件语句与时序电路的实现关系,介绍了常用时序电路部件计数器、移位寄存器和移位相加乘法器的设计,同时对属性描述与定义语句的语法进行了详细讨论。第 5 章 FPGA 开发环境 Quartus II 应用基础,以计数器和移位全加器实例为导向,详细介绍了 Quartus II 文本输入设计方法及原理图输入方法,讨论了 JTAG 间接模式编程配置器件、使用 Signal-Tap II 进行采样与分析的方法。第 6 章宏功能模块与 IP 应用,讨论了使用 LPM 模块设计 ROM、RAM 和 FIFO 存储器的方法,以实例为导向,介绍了 DSP 模块构建乘法器、LPM 嵌入式锁相环设计、IP 核 NCO 数控振荡器设计及 8051 单片机 IP 软核应用。第 7 章状态机设计,介绍了数字系统中的状态机应用特性,以 AD0809 采样控制和数字序列检测为例,讨论了状态机的设计方法。第 8 章 FPGA 设计工程实践,以 12 个工程设计专题为例,介绍了 FPGA 工程设计实践应用技术,最后讨论了工程设计中的原则与方法以及 VHDL 工程设

计风格与优化。

限于 EDA 软件，书中一些电路连接图中使用的元器件并未采用国标中规定的形式，请读者注意。

本书第 1、4、5、6、7、8 章由肖闽进编写，第 2 章由赵泓扬编写，第 3 章由邹全编写。本系黄文生、蒋小燕、鲍玉军、郑中桥、胡圣尧老师和实验室许清泉老师参加了部分内容的编写及实验工作，全书由肖闽进统稿。同济大学交通运输学院副院长、博士生导师董德存教授审阅了本书，在此表示感谢。

限于编者水平，书中难免有不当之处，诚请读者批评、指正。

编 者

2015 年 3 月

目 录

第 1 章 EDA 技术概述	(1)
1.1 硬件描述语言	(1)
1.2 EDA 设计流程	(2)
1.2.1 设计输入	(2)
1.2.2 综合与适配	(3)
1.2.3 时序仿真与功能仿真	(4)
1.2.4 编程下载与器件测试	(5)
1.3 Quartus II 设计环境	(5)
1.4 EDA 技术开发中的 IP	(6)
习题与思考题	(7)
第 2 章 可编程逻辑器件硬件基础	(8)
2.1 可编程器件结构	(8)
2.2 可编程逻辑器件的分类	(8)
2.2.1 按集成度分类	(9)
2.2.2 按组成结构分类	(9)
2.2.3 按编程工艺分类	(9)
2.3 PLD 电路表示	(10)
2.4 GAL 器件结构原理	(10)
2.5 CPLD 的结构原理	(13)
2.6 FPGA 的结构原理	(16)
2.6.1 查找表逻辑结构	(16)
2.6.2 Cyclone III 系列器件的结构原理	(17)
2.7 CPLD/FPGA 的编程与配置	(21)
2.7.1 CPLD 在系统编程	(22)
2.7.2 FPGA 配置方式	(23)
2.7.3 用 JTAG 进行 FPGA 的配置	(23)
2.7.4 FPGA 专用配置器件	(24)
习题与思考题	(25)
第 3 章 VHDL 组合电路设计	(26)
3.1 2 选 1 多路选择器及其 VHDL 描述	(26)
3.2 IEEE 库预定义标准逻辑位与矢量数据类型	(29)
3.3 设计库和标准程序包	(30)
3.4 ENTITY(实体)语句结构和语法含义	(30)

3.4.1	实体名	(31)
3.4.2	端口语句和端口信号名	(31)
3.4.3	端口模式	(31)
3.4.4	数据类型	(31)
3.5	ARCHITECTURE(结构体)语句结构和语法含义	(32)
3.5.1	赋值符号和数据比较符号	(32)
3.5.2	逻辑操作符	(33)
3.5.3	条件语句	(33)
3.5.4	进程语句和顺序语句	(34)
3.5.5	IF 语句	(37)
3.6	8-3 线优先编码器 VHDL 设计	(38)
3.7	文件保存与文件名	(40)
3.8	1 位二进制全加器的 VHDL 描述	(40)
3.8.1	半加器的 VHDL 描述	(40)
3.8.2	CASE 语句	(41)
3.8.3	并置操作符	(42)
3.8.4	全加器的 VHDL 描述	(42)
3.8.5	元件例化语句	(43)
3.9	硬件乘法器 VHDL 设计	(44)
3.9.1	变量	(45)
3.9.2	信号	(46)
3.9.3	进程中的信号与变量赋值特点	(47)
3.9.4	FOR_LOOP 循环语句	(48)
3.10	参数传递说明语句	(49)
3.11	整数类型	(49)
3.12	VHDL 操作符	(50)
3.12.1	移位操作符	(51)
3.12.2	省略赋值操作符	(51)
3.12.3	逻辑操作符	(52)
3.12.4	关系操作符	(53)
3.12.5	求和操作符	(54)
3.12.6	求积操作符	(55)
3.12.7	符号操作符	(55)
3.12.8	混合操作符	(55)
3.13	数据类型转换函数	(55)
3.14	参数传递映射语句	(57)
	习题与思考题	(58)
第 4 章	VHDL 时序电路设计	(59)
4.1	基本时序电路 D 触发器的 VHDL 描述	(59)

4.1.1	上升沿检测表达式和信号属性函数	(60)
4.1.2	不完整条件语句与时序电路	(60)
4.1.3	VHDL 实现时序电路的其他表述方法	(61)
4.2	计数器的 VHDL 设计	(63)
4.2.1	十六进制加法计数器	(63)
4.2.2	十进制加法计数器	(64)
4.3	移位寄存器的 VHDL 设计	(66)
4.4	移位相加乘法器的 VHDL 设计	(67)
4.5	属性描述与定义语句	(72)
4.5.1	信号类属性	(73)
4.5.2	数据区间类属性	(73)
4.5.3	数值类属性	(73)
4.5.4	数组属性	(74)
4.5.5	用户定义属性	(74)
	习题与思考题	(75)
第 5 章	FPGA 开发环境 Quartus II 应用基础	(76)
5.1	Quartus II 文本输入设计方法	(76)
5.1.1	建立工作库文件夹和编辑设计文件	(76)
5.1.2	创建工程	(77)
5.1.3	编译前设置	(80)
5.1.4	全程编译	(81)
5.1.5	时序仿真	(83)
5.1.6	寄存器传输级(RTL)电路图	(88)
5.2	器件引脚锁定与程序下载	(89)
5.2.1	引脚锁定	(89)
5.2.2	配置文件下载	(91)
5.3	JTAG 间接模式编程配置器件	(92)
5.3.1	将 SOF 文件转化为 JTAG 间接配置文件	(93)
5.3.2	下载 JTAG 间接配置文件	(94)
5.3.3	USB Blaster 编程配置器件使用方法	(95)
5.4	嵌入式逻辑分析仪使用方法	(95)
5.4.1	打开 SignalTap II 编辑窗口	(96)
5.4.2	调入待测信号	(96)
5.4.3	SignalTap II 参数设置	(97)
5.4.4	文件存盘与综合适配	(98)
5.4.5	编译下载	(99)
5.4.6	启动 SignalTap II 进行采样与分析	(99)
5.4.7	SignalTap II 的其他设置和控制方法	(100)
5.5	Quartus II 原理图输入设计方法	(101)

5.5.1	输入设计项目和文件保存	(101)
5.5.2	将设计项目设置成可调用的元件	(102)
5.5.3	设计全加器顶层文件	(103)
5.5.4	将设计项目设置成工程和时序仿真	(103)
	习题与思考题	(104)
第 6 章	宏功能模块与 IP 应用	(105)
6.1	LPM 存储器设计	(105)
6.1.1	ROM 模块设计	(105)
6.1.2	RAM 模块设计	(111)
6.1.3	FIFO(先进先出)存储器模块设计	(113)
6.2	DSP 模块构建乘法器	(114)
6.3	正弦信号发生器设计	(115)
6.3.1	正弦信号发生器模块设计	(115)
6.3.2	使用嵌入式逻辑分析仪测试数据波形	(116)
6.4	在系统存储器数据读写编辑器应用	(117)
6.4.1	打开在系统存储单元编辑窗口	(117)
6.4.2	读取 ROM 中的波形数据	(117)
6.4.3	ROM 写入数据	(118)
6.4.4	输入/输出数据文件	(118)
6.5	IP 核 NCO 数控振荡器设计	(119)
6.5.1	选择 IP 核 NCO 并确定文件名	(119)
6.5.2	进入文件生成设置窗口	(119)
6.5.3	NCO 参数设置	(120)
6.5.4	生成仿真文件	(121)
6.5.5	加入 IP 授权文件	(121)
6.5.6	编译与功能检测	(122)
6.6	LPM 嵌入式锁相环设计	(122)
6.6.1	建立嵌入式锁相环元件	(122)
6.6.2	锁相环仿真测试和硬件特性	(124)
6.7	8051 单片机 IP 软核应用	(125)
6.8	LPM 模块的文本调用	(129)
6.8.1	在 Quartus II 中对宏功能模块进行例化	(129)
6.8.2	LPM 模块的 VHDL 文本方式调用	(129)
	习题与思考题	(132)
第 7 章	状态机设计	(133)
7.1	数字系统中的状态机应用特性	(133)
7.2	有限状态机的设计	(134)
7.2.1	数据类型定义语句	(134)
7.2.2	有限状态机的设计	(136)

7.3 状态机的编码方式	(144)
7.3.1 状态位直接输出型编码	(144)
7.3.2 顺序编码	(146)
7.3.3 一位热码编码	(146)
7.3.4 剩余状态处理	(146)
习题与思考题	(147)
第8章 FPGA 设计工程实践	(149)
8.1 LED 数码管显示控制	(149)
8.1.1 LED 数码管的工作原理	(149)
8.1.2 采用 FPGA 实现 LED 数据管静态显示控制	(149)
8.1.3 采用 FPGA 实现 LED 数码管动态显示控制	(150)
8.1.4 LED 显示程序设计与仿真测试	(150)
8.2 液晶显示器(LCD)控制	(152)
8.2.1 LCD 显示的原理	(152)
8.2.2 MDLS 字符型液晶显示模块 FPGA 驱动控制电路	(154)
8.2.3 液晶显示器驱动模块的 VHDL 控制程序	(154)
8.3 TLC5510 采样控制设计	(156)
8.3.1 TLC5510 与 FPGA 接口电路原理	(156)
8.3.2 TLC5510 VHDL 采样控制程序设计	(157)
8.4 TLC7524 数/模转换控制设计	(158)
8.4.1 TLC7524 接口电路原理	(158)
8.4.2 TLC7524 数/模转换控制程序设计	(159)
8.5 2FSK 调制与解调设计	(161)
8.5.1 2FSK 信号调制原理和 VHDL 模块设计	(161)
8.5.2 2FSK 信号解调原理和 VHDL 模块设计	(165)
8.6 4PSK 调制与解调设计	(168)
8.6.1 4PSK 信号调制原理和 VHDL 模块设计	(168)
8.6.2 4PSK 信号解调原理和 VHDL 模块设计	(172)
8.7 交织编码器设计	(175)
8.7.1 交织编码原理	(175)
8.7.2 交织编码 FPGA 模块设计与仿真测试	(176)
8.8 数字音乐播放控制器设计	(180)
8.8.1 数字音乐播放控制原理	(180)
8.8.2 数字音乐播放控制器 VHDL 模块设计	(180)
8.9 步进电机驱动控制设计	(185)
8.9.1 步进电机驱动原理	(185)
8.9.2 步进电机步距八细分系统设计	(187)
8.9.3 细分驱动的 VHDL 程序和仿真测试	(189)
8.10 直流电机控制设计	(192)

8.10.1	直流电机转速 PWM 控制原理	(192)
8.10.2	直流电机 PWM 驱动的 VHDL 模块设计和仿真测试	(193)
8.11	数字频率计设计	(195)
8.11.1	频率计原理和 VHDL 模块设计	(195)
8.11.2	频率计 VHDL 程序及仿真测试	(195)
8.12	LED 显示时钟设计	(199)
8.12.1	数字显示时钟实现原理	(199)
8.12.2	LED 时钟的 VHDL 程序和仿真测试	(200)
8.13	VHDL 工程设计风格与优化	(205)
8.13.1	描述方法与编码风格	(205)
8.13.2	不同的状态机描述	(208)
8.13.3	层次化设计方法	(211)
8.13.4	相似逻辑设计在一个层次	(211)
8.13.5	使用寄存器作为模块的分界线	(212)
8.13.6	优化设计方法	(212)
参考文献		(219)

第 1 章 EDA 技术概述

进入 21 世纪,电子设计自动化(Electronic Design Automation, EDA)技术的发展给数字系统的设计带来了革命性的变化。微电子技术和现代计算机技术的进步,使可编程逻辑器件和设计工具快速发展。利用 EDA 工具,采用可编程逻辑器件,正在成为数字系统设计的主流。采用高级语言描述,具有系统级仿真和综合能力是 EDA 技术的基本特征。

EDA 技术融合了应用电子技术、计算机技术、信息处理及智能化技术的最新成果。利用 EDA 工具,设计者可以从概念、算法、协议等开始设计电子系统,大量工作可以通过计算机完成,并可以将电子产品从电路设计、性能分析到设计出 IC 版图或 PCB 版图的整个过程在计算机上自动处理完成,真正实现电子产品的自动化设计。

EDA 技术采用可编程逻辑器件,通过对器件内部的设计实现系统功能,是一种基于芯片的设计方法。EDA 设计可以根据需要定义器件的内部逻辑和管脚,将电路板设计的大部分工作放在芯片的设计中进行,通过对芯片的设计实现数字系统的逻辑功能。EDA 设计过程依靠 EDA 工具软件平台,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,自动地完成逻辑化简、逻辑分割、逻辑综合、结构综合(布局布线),以及逻辑优化和仿真测试等功能,直至实现既定性能的电子线路系统功能。EDA 技术使得设计工作几乎仅限于利用软件的方式,即利用硬件描述语言(HDL)和 EDA 软件完成对系统硬件功能的实现。将上述目标具体化的途径之一,是采用 VHDL 设计数字系统模块,利用 Quartus II 开发环境,在 FPGA(Field Programmable Gate Array, 现场可编程逻辑阵列)或 CPLD(Complex Programmable Logic Device, 复杂可编程逻辑器件)上实现应用系统硬件。这种基于芯片的设计方法可以减少芯片的使用数量,缩小系统体积,降低功率消耗,提高系统的速度性能和可靠性。

1.1 硬件描述语言

EDA 技术的语言工具是硬件描述语言(HDL)。常用的 HDL 有 VHDL、Verilog HDL、System Verilog 和 System C,其中 VHDL、Verilog HDL 在现在 EDA 设计中使用最多,也得到几乎所有的主流 EDA 工具的支持。而 System Verilog 和 System C 这两种 HDL 语言还处于完善过程中。VHDL 是电子设计主流硬件的描述语言之一,以下将重点介绍它的编程方法和使用技术。

VHDL 的英文全名是 VHSIC(Very High Speed Integrated Circuit)Hardware Description Language,1983 年由美国国防部发起创建,由 IEEE(the Institute of Electrical and Electronics Engineers)进一步发展并在 1987 年作为“IEEE 标准 1076”(IEEE Std 1076)发布。从此,VHDL 成为硬件描述语言的业界标准之一。自 IEEE 公布了 VHDL 的标准版本之后,各 EDA 公司相继推出了自己的 VHDL 设计环境,或宣布自己的设计工具支持 VHDL。此后 VHDL 在电子

设计领域得到了广泛应用,并逐步取代了原有的非标准硬件描述语言。

随着 VHDL 的标准化,出现了一些支持该语言的行为仿真器。创建 VHDL 的最初目标是用于标准文档的建立和电路功能模拟,其基本想法是在高层次上描述系统和元件的行为。但到了 20 世纪 90 年代初,人们发现,VHDL 不仅可以作为系统模拟的建模工具,还可以作为电路系统的设计工具,即可以利用软件工具将 VHDL 源码自动地转化为文本方式表达的基本逻辑元件连接图,即网表文件。这种方法显然对于电路自动设计是一个极大的推进。很快,电子设计领域出现了第一个软件设计工具,即 VHDL 逻辑综合器,它把标准 VHDL 的部分语句描述转化为具体电路实现的网表文件。1993 年,IEEE 对 VHDL 进行了修订,从更高的抽象层次和系统描述能力上扩展了 VHDL 的内容,公布了新版本 VHDL,即 IEEE 1076-1993。现在,VHDL 和 Verilog 作为 IEEE 的工业标准硬件描述语言,得到众多 EDA 公司的支持,在电子工程领域已成为事实上的通用硬件描述语言(目前最新的 VHDL 标准版本是 IEEE 1076-2002)。

VHDL 语言具有很强的电路描述和建模能力,能从多个层次对数字系统进行建模和描述,从而大大简化了硬件设计任务,提高了设计效率和可靠性。VHDL 具有与具体硬件电路无关和与设计平台无关的特性,并且具有良好的电路行为描述和系统描述的能力,在语言易读性和层次化、结构化设计方面表现出了强大的生命力和应用潜力。因此,VHDL 支持各种模式的设计方法:自顶向下与自底向上或混合方法。面对当今许多电子产品生命周期缩短,需要多次重新设计以融入最新技术、改变工艺等问题,VHDL 具有良好的适应性。用 VHDL 进行电子系统设计的一个很大的优点是设计者可以专心致力于其功能的实现,而不需要对不影响功能的、与工艺有关的因素花费过多的时间和精力。

1.2 EDA 设计流程

EDA 技术设计开发流程对于正确选择和使用 EDA 软件、优化设计项目、提高设计效率具有重要意义。EDA 设计流程既是自顶向下设计方法的具体实施途径,也是 EDA 工具软件本身的组成结构。图 1-1 是基于 EDA 软件的 FPGA/CPLD 开发流程框图。对于目前流行的 EDA 工具软件,图 1-1 的设计流程具有一般性。以下分别介绍各设计模块的功能特点。

1.2.1 设计输入

将电路系统以一定的表达方式输入计算机,是在 EDA 软件平台上对 FPGA/CPLD 开发的最初步骤。通常,使用 EDA 工具的设计输入可分为两种类型:图形输入和 HDL 文本输入方式。

1. 图形输入方式

图形输入通常包括原理图输入、状态图输入和波形图输入三种常用方法。

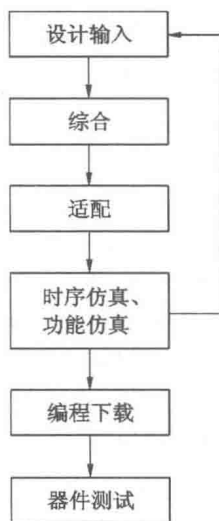


图 1-1 基于 EDA 软件的 FPGA/CPLD 开发流程

状态图输入方法就是根据电路的控制条件和不同的转换方式,用绘图的方法,在 EDA 工具的状态图编辑器上绘出状态图,然后由 EDA 编译器和综合器将此状态变化流程图图形编译综合成电路网表。状态图输入可简化状态机的设计过程。

波形图输入方法则是将待设计的电路看成是一个黑盒子,只需告诉 EDA 工具黑盒子电路的输入和输出时序波形图,EDA 工具即能据此完成黑盒子电路的设计。

图形输入设计方法是一种类似于传统电子设计方法的图形编辑输入方式,即在 EDA 软件的图形编辑界面上绘制能完成特定功能的电路图形。图形由逻辑器件(符号)和连接线构成,图中的逻辑器件可以是 EDA 软件库中预制的功能模块,如与门、非门、或门、触发器以及各种含 74 系列器件功能的宏功能块,也可以是一些类似于 IP 的功能块。当图形编辑绘制完成后,图形编辑器将对输入的图形文件进行排错,之后再将其编译成适用于逻辑综合的网表文件。

用图形表达的输入方法的优点主要在于设计者不需要学习新的相关知识(如 HDL 等),设计过程形象直观,适用于初学者或演示等。然而,其缺点同样十分明显:由于图形设计并未标准化,不同的 EDA 图形处理工具对图形的设计规则、存档格式和图形编译方式都不同,因此图形文件兼容性差,难以交换和管理。随着电路设计规模的扩大,原理图输入描述方式必然引起一系列难以克服的困难,如电路功能原理的易读性下降、错误排查困难、整体调整和结构升级困难。例如,将一个 4 位的单片机设计升级为 8 位单片机几乎难以在短时间内实现。由于在原理图中已确定了设计系统的基本电路结构和元件,留给综合器和适配器的优化选择的空间已十分有限,因此难以实现用户所希望的面积、速度以及不同风格的综合优化。显然,原理图的设计方法明显偏离了电子设计自动化最本质的含义。在设计中,由于必须直接面对硬件模块的选用,因此行为模型的建立将无从谈起,从而无法实现真实意义上的自顶向下的设计方案。

2. HDL 文本输入方式

文本输入方式与传统的计算机软件语言编辑输入基本一致,就是将使用了某种硬件描述语言的电路设计文本,如 VHDL 或 Verilog HDL 的源程序,进行编辑输入。应用 HDL 的文本输入方法克服了上述原理图输入法存在的所有弊端,为 EDA 技术的应用和发展打开了一个广阔的天地。当然,在一定的条件下,情况会有所改变。目前有些 EDA 输入工具可以把图形的直观与 HDL 的优势结合起来。如状态图输入的编辑方式,即用图形化状态机输入工具,用图形的方式表示状态图,当填好时钟信号名、状态转换条件、状态机类型等要素后,就可以自动生成 Verilog/VHDL 程序。又如,在原理图输入方式中,连接用 HDL 描述的各个电路模块,直观地表示系统的总体框架,再用自动 HDL 生成工具生成相应的 VHDL 或 Verilog HDL 程序。但总体来看,纯 HDL 输入设计仍然是最基本、最有效和最通用的输入方法。

1.2.2 综合与适配

1. 综合

综合仅是对 HDL 而言的。利用 HDL 综合器对设计进行综合是十分重要的一步,因为综合过程将把软件设计的 HDL 描述与硬件结构挂钩,是将软件转化为硬件电路的关键步

骤,是文字描述与硬件实现的一座桥梁。综合就是将电路的高级语言(如行为描述)转换成低级的、可与 FPGA/CPLD 的基本结构相映射的网表文件或程序。当输入的 HDL 文件在 EDA 工具中检测无误后,首先面临的是逻辑综合,因此要求 HDL 源文件中的语句都是可综合的。

在综合后,综合器一般可以生成一种或多种文件格式网表文件,如 EDIF、VHDL、Verilog、VQM 等标准格式。各种网表文件用各自的格式描述电路的结构,如 Verilog 网表文件采用 Verilog 的语法,用结构描述的风格重新诠释综合后的电路结构。

整个综合过程就是将设计者在 EDA 平台上编辑输入的 HDL 文本、原理图或状态图形描述,依据给定的硬件结构组件和约束控制条件进行编译、优化、转换和综合,最终获得门级电路甚至更底层的电路描述网表文件的过程。由此可见,综合器工作前,必须给定最后实现的硬件结构参数,综合器的功能就是将软件描述与给定的硬件结构用某种网表文件的方式对应起来,使之成为相应的映射关系。如果把综合理解为映射过程,那么显然这种映射不是唯一的,并且综合的优化也不是单方向的。为达到速度、面积、性能的要求,往往需要对综合加以约束,称为综合约束。

2. 适配

适配器也称结构综合器,它的功能是将由综合器产生的网表文件配置于指定的目标器件中,使之产生最终的下载文件,如 JEDEC、JAM、SOF、POF 格式的文件。适配所选定的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 FPGA/CPLD 供应商自己提供,因为适配器的适配对象直接与器件的结构细节相对应。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,其中包括底层器件配置、逻辑分割、逻辑优化、逻辑布局布线操作。适配完成后可以利用适配所产生的仿真文件作精确的时序仿真测试,同时产生可用于编程的文件。

1.2.3 时序仿真与功能仿真

仿真就是让计算机根据一定的算法和一定的仿真库对 EDA 设计进行模拟测试,以验证设计,排除错误。在编程下载前必须利用 EDA 工具对适配生成的结果进行模拟测试。仿真是 EDA 设计过程中的重要步骤。图 1-1 所示的时序与功能门级仿真通常由 PLD 公司的 EDA 开发工具直接提供(也可以选用第三方的专业仿真工具),可以完成两种不同级别的仿真测试。

时序仿真,就是接近真实器件运行特性的仿真,仿真文件中已包含了器件的硬件特性参数,因而仿真精度高。但时序仿真的仿真文件必须来自针对具体器件的综合器与适配器。综合后所得的 EDIF、VQM 等网表文件通常作为 FPGA 适配器的输入文件,产生的仿真网表文件中包含了精确的硬件延迟信息。

功能仿真,是直接对 VHDL、原理图描述或其他描述形式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。仿真过程不涉及任何具体器件的硬件特性,不经历综合与适配阶段,在设计项目编辑、编译后即可进入门级仿真器进行模拟测试。直接进行功能仿真的好处是设计耗时短,对硬件库、综合器等没有任何要求。对于规模比较大的设计项目,综合与适配在计算机上的耗时是十分可观的,如果每一次修改后的模拟都

必须进行时序仿真,显然会极大地降低开发效率。因此,通常的做法是,首先进行功能仿真,待确认设计文件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情况。

1.2.4 编程下载与器件测试

下载与配置是指把适配后生成的下载或配置文件,通过编程器或编程电缆向 FPGA 或 CPLD 下载,以便进行硬件调试和验证(Hardware Debugging)。

通常,将对 CPLD 的下载称为编程(Program),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。

器件测试用于将含有载入了设计的 FPGA 或 CPLD 的硬件系统进行统一测试,以便最终验证设计项目在目标系统上的实际工作情况,排除错误,改进设计。

1.3 Quartus II 设计环境

Quartus II 是 Altera 提供的 FPGA/CPLD 开发集成环境,Altera 是世界上最大的可编程逻辑器件供应商之一。Quartus II 应用方法和设计流程对于其他流行的 EDA 工具而言具有一定的典型性和一般性,本书给出的设计和实践也多是基于 Quartus II 环境的。

Quartus II 在 21 世纪初推出,是 Altera 前一代 FPGA/CPLD 集成开发环境 MAX+plus II 的更新换代产品,其界面友好,使用便捷。在 Quartus II 上可以完成整个 FPGA/CPLD 流程,它提供了一种与结构无关的设计环境,使设计者能方便地进行设计输入、快速处理和器件编程。

Altera 的 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,也是单芯片可编程系统(SOPC)设计的综合性环境和 SOPC 开发的基本设计工具,并为 Altera DSP 开发包进行系统模型设计提供了集成综合环境。

Quartus II 设计工具完全支持 Verilog、VHDL 的设计流程,其内部嵌有 Verilog、VHDL 逻辑综合器。Quartus II 也可以利用第三方的综合工具,如 Leonardo Spectrum、Synplify Pro、DC-FPGA,并能直接调用这些工具。同样,Quartus II 具备仿真功能,同时也支持第三方的仿真工具,如 ModelSim。此外,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 包括模块化的编译器。编译器包括的功能模块有分析/综合器(Analysis & Synthesis)、适配器(Fitter)、装配器(Assembler)、时序分析器(Timing Analyzer)、设计辅助模块(Design Assistant)、EDA 网表文件生成器(EDA Netlist Writer)、编译器数据库接口(Compiler Database Interface)等。编译器可以通过选择 Start Compilation 来运行所有的编译器模块,也可以通过选择 Start 单独运行各个模块,还可以通过选择 Compiler Tool (Tools 菜单),在 Compiler Tool 窗口中运行相应的功能模块。在 Compiler Tool 窗口中,可以打开相应的功能模块所包含的设置文件或报告文件,或打开其他相关窗口。

此外,Quartus II 还包含许多十分有用的 LPM (Library of Parameterized Modules) 模块,它们是复杂或高级系统构建的重要组成部分,可在 Quartus II 中与普通设计文件一