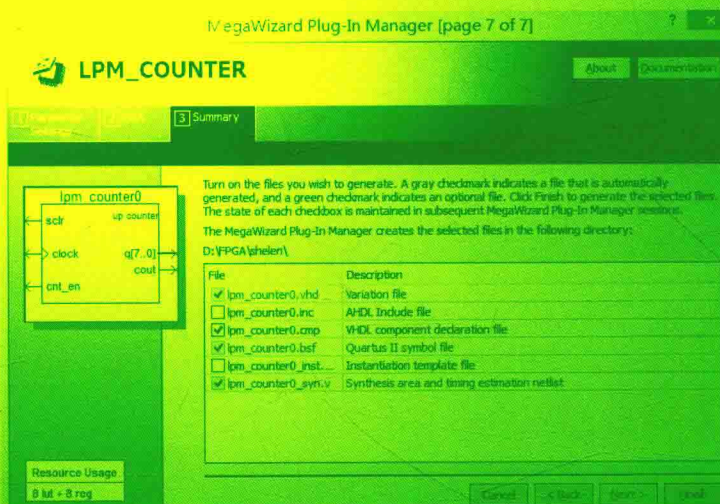


FPGA

数字系统设计与应用

- 郭明良 主 编
- 常国祥 王 欢 副主编



普通高等教育电气类规划教材 <<<

FPGA

数字系统设计与应用

- 郭明良 主 编
- 常国祥 王 欢 副主编



化学工业出版社

· 北京 ·

图书在版编目 (CIP) 数据

FPGA 数字系统设计与应用/郭明良主编. —北京:
化学工业出版社, 2017.8
普通高等教育电气类规划教材
ISBN 978-7-122-29843-0

I. ①F… II. ①郭… III. ①可编程序逻辑器件-
系统设计-高等学校-教材 IV. ①TP332.1

中国版本图书馆 CIP 数据核字 (2017) 第 126401 号

责任编辑: 高墨荣
责任校对: 宋 玮

文字编辑: 孙凤英
装帧设计: 刘丽华

出版发行: 化学工业出版社 (北京市东城区青年湖南街 13 号 邮政编码 100011)

印 装: 三河市延风印装有限公司

787mm×1092mm 1/16 印张 11 $\frac{3}{4}$ 字数 284 千字 2017 年 9 月北京第 1 版第 1 次印刷

购书咨询: 010-64518888 (传真: 010-64519686) 售后服务: 010-64518899

网 址: <http://www.cip.com.cn>

凡购买本书, 如有缺损质量问题, 本社销售中心负责调换。

定 价: 39.00 元

版权所有 违者必究

近年来,随着电子技术、计算机应用技术和 EDA 技术的不断发展,利用 FPGA/CPLD 进行数字系统的开发已被广泛应用于交通运输、航空航天、精密仪器、医疗电子、工业控制等各领域。FPGA/CPLD 具有功能强大、开发周期短以及产品集成度高、易于修改等特点,并且 FPGA/CPLD 本身发展非常迅速,高密度、高性能、低成本的 FPGA 器件推陈出新,也促进了 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。

VHDL 语言作为国际标准的硬件描述语言,已经成为工程技术人员和高校学生的必备技能。本书例子中的文本编辑均采用 VHDL 语言编写,书中所有实例都已通过仿真和硬件测试。

全书根据 EDA 应用技术,特别是结合 FPGA 应用领域包含的主要内容编写,并以实例的形式介绍了 Altera 公司推出的 Quartus II 13.1 设计平台以及 Modelsim 10.0c 仿真平台。本书由浅入深,先介绍 FPGA 设计流程,然后介绍开发平台、仿真平台、VHDL 语言,最后讲解了典型的应用实例。

本书分 7 章,其中第 1 章介绍了什么是 FPGA, FPGA 设计的环境,设计流程,所使用的语言等;第 2 章介绍了 FPGA 的开发平台,主要讲解 Altera 的 Quartus II 13.1,本章图文并茂,以简单的实例为主,重点讲述使用软件的完整开发流程,包括输入、编译、仿真以及下载;第 3 章介绍了 Modelsim 10.0c 仿真平台,介绍了使用 Modelsim 10.0c 进行功能仿真与时序仿真,学完本章即可完成基本的仿真,Modelsim 功能十分强大,由于篇幅的限制,不做进一步讲解;第 4 章介绍了 VHDL 的语言基础,以简单的实例说明 VHDL 程序的基本组成;第 5 章介绍了 VHDL 的数据类型与运算操作符,为后续程序奠定基础;第 6 章介绍了 VHDL 的主要描述语句;第 7 章是实例,详细介绍了如何利用 FPGA 进行程序设计,既有设计原理,也有程序说明,并在最后给出仿真结果。

本书是编者在 FPGA 学习和实践中的经验点滴,书中既有日常的学习笔记,对一些常用设计技巧和方法进行深入探讨,也有很多生动的实例,这些实例大都是以特定的工程项目为依托,具有一定的借鉴价值,还有多个完整的项目工程实例,让读者从系统角度理解 FPGA 的开发流程。

本书从工程实践出发,旨在引领读者学会如何在FPGA的开发设计过程中发现问题、分析问题并解决问题。本书所有程序都是测试过的,读者如果需要可发电子邮件至yj74615@163.com联系索取相应资料。

本书可用作高等院校计算机专业本、专科生的教材或教学参考书,也可以作为电子技术课程设计、电子设计大赛或数字系统设计工程技术人员学习EDA技术的参考书。

本书由郭明良任主编,常国祥、王欢任副主编,书中第1章、第2章、第4章、第5章由王欢编写;第6章以及第7章的7.8~7.10节由杨立新编写,第7章的示例7.4~7.7节、7.11节由郭明良编写,第3章与第7章7.1~7.3节由常国祥编写,全书由常国祥主审。

由于水平有限,书中难免有不足之处,敬请读者批评指正。

编者

第 1 章 概述 / 001

1.1 可编程逻辑器件	001
1.2 FPGA/CPLD 的开发工具	002
1.3 硬件描述语言	002
1.3.1 AHDL 语言	002
1.3.2 Verilog HDL 语言	002
1.3.3 VHDL 语言	003
1.4 FPGA/CPLD 的开发流程	003

第 2 章 Quartus II 集成开发环境 / 005

2.1 Quartus II 界面概况	005
2.2 Quartus II 软件开发流程	006
2.2.1 创建工程	006
2.2.2 原理图输入方式	008
2.2.3 文本输入方式	015
2.2.4 波形仿真	017
2.2.5 引脚分配	021
2.2.6 编程下载	021
2.3 LPM 参数化宏功能模块	024
2.3.1 LPM 参数化宏功能模块定制管理器	024
2.3.2 LPM 参数化宏功能模块的应用	030

第 3 章 仿真 / 034

3.1 Modelsim 简介	034
-----------------------	-----

3.2	安装	034
3.3	Modelsim 仿真方法	035
3.3.1	前仿真	035
3.3.2	后仿真	035
3.3.3	Modelsim 仿真的基本步骤	036
3.3.4	Modelsim 的运行方式	036
3.4	Modelsim 功能仿真	036
3.4.1	建立仿真工程	038
3.4.2	Altera 仿真库的编译与映射	039
3.4.3	编译 HDL 源代码和 Testbench	041
3.4.4	启动仿真器并加载设计顶层	042
3.4.5	打开观察窗口, 添加信号	044
3.4.6	执行仿真	045
3.5	Modelsim 时序仿真	046
3.5.1	仿真路径设置	046
3.5.2	Quartus II 仿真环境设置	047
3.5.3	利用 Quartus II 编译源文件	048
3.5.4	生成测试模板并编写测试程序	048
3.5.5	执行仿真	051

第 4 章 VHDL 语言基础 / 054

4.1	VHDL 语言的特点	054
4.2	VHDL 语言的程序结构	055
4.3	VHDL 语言的库	055
4.4	VHDL 语言的程序包	057
4.5	VHDL 语言的实体	058
4.5.1	实体说明	058
4.5.2	实体的类属说明	058
4.5.3	实体的端口说明	059
4.6	VHDL 语言的结构体	060
4.7	VHDL 语言的配置	062

第 5 章 VHDL 数据类型与运算操作符 / 065

5.1	VHDL 的基本语法规则	065
-----	--------------	-----

5.2	VHDL 语言的数据对象	066
5.2.1	常量 (CONSTANT)	066
5.2.2	变量 (VARIABLE)	066
5.2.3	信号 (SIGNAL)	067
5.2.4	文件 (FILES)	068
5.3	VHDL 语言的数据类型	068
5.3.1	预定义的数据类型	068
5.3.2	用户自定义数据类型	071
5.3.3	数据类型的转换	073
5.4	VHDL 语言的操作符	075
5.4.1	逻辑操作符	076
5.4.2	算术操作符	076
5.4.3	关系操作符	077

第 6 章 VHDL 的主要描述语句 / 079

6.1	顺序描述语句	079
6.1.1	变量赋值语句	079
6.1.2	信号赋值语句	080
6.1.3	WAIT 语句	080
6.1.4	IF 语句	082
6.1.5	CASE 语句	085
6.1.6	NULL 语句	087
6.1.7	断言 (ASSERT) 语句	087
6.1.8	LOOP 语句	088
6.1.9	NEXT 语句	090
6.1.10	EXIT 语句	091
6.2	并发描述语句	092
6.2.1	进程语句	092
6.2.2	并发信号赋值语句	093
6.2.3	并发过程调用语句	095
6.2.4	块 (BLOCK) 语句	097
6.2.5	元件例化语句	098
6.2.6	生成语句	100
6.3	属性描述与定义语句	102

7.1 自动邮票售票机设计	111
7.1.1 自动邮票售票系统总体模块图的设计	111
7.1.2 票价设定模块的设计	112
7.1.3 邮票类型选择模块的设计	113
7.1.4 点阵票型显示模块的设计	116
7.1.5 邮票类型选择与票价设定模块的设计	120
7.1.6 邮票数量设定模块的设计	121
7.1.7 邮票购买模块的设计	123
7.1.8 数据转换模块的设计	126
7.1.9 动态扫描模块的设计	127
7.1.10 数码管显示模块的设计	128
7.1.11 综合设计	129
7.2 交通灯控制系统的设计	130
7.2.1 交通灯控制系统模块图	130
7.2.2 控制模块设计	130
7.2.3 显示模块设计	135
7.2.4 综合设计	139
7.3 八路抢答器的设计	140
7.3.1 主持人控制模块	141
7.3.2 抢答信号锁存模块	142
7.3.3 倒计时模块	144
7.3.4 二进制编码转 BCD 码模块	146
7.3.5 扫描信号产生模块	147
7.3.6 数码管位信号与段信号匹配模块	147
7.3.7 BCD 码转七段码模块	148
7.3.8 报警模块	149
7.3.9 综合设计	150
7.4 数字频率计 VHDL 程序与仿真	151
7.5 乐曲硬件演奏电路设计	155
7.5.1 顶层设计	156
7.5.2 音调产生模块	157
7.5.3 音调查询	158
7.5.4 节拍和音符数据发生器模块	159

7.5.5 “梁祝”乐曲演奏数据	160
7.6 数控分频器的设计	161
7.7 状态机 A/D 采样控制电路实现	162
7.8 比较器和 D/A 器件	164
7.9 ASK 调制解调 VHDL 程序及仿真	165
7.9.1 ASK 调制 VHDL 程序及仿真	165
7.9.2 ASK 解调 VHDL 程序及仿真	166
7.10 FSK 调制与解调 VHDL 程序及仿真	167
7.10.1 FSK 调制 VHDL 程序及仿真	167
7.10.2 FSK 解调方框图及电路符号	168
7.10.3 FSK 解调 VHDL 程序及仿真	169
7.11 多功能波形发生器 VHDL 程序与仿真	171

第1章

概述

随着电子技术、芯片集成技术、计算机及其软件技术的飞速发展，现场可编程逻辑门阵列 FPGA 和复杂可编程逻辑器件 CPLD 在电子设计领域的应用越来越广泛。它们以其高集成度、高速度和高可靠性及其延时可小至纳秒级的特点，并结合其并行工作方式，在超高速领域和实时测控方面有着非常广阔的应用前景。

1.1 可编程逻辑器件

可编程逻辑器件 (Programmable Logic Device, PLD) 是一种通过用户编程来实现某种逻辑功能的新型逻辑器件，经过近几十年的发展，可编程逻辑器件已经从最初简单的 PLA、PAL、GAL 发展到了目前应用广泛的 CPLD (Complex Programmable Logic Device, 复杂的可编程逻辑器件)，FPGA (Field Programmable Gate Array)。

FPGA 采用互补金属氧化物半导体工艺制成，是一种基于查找表的可编程逻辑器件，在结构上主要分为可编程逻辑单元、可编程输入/输出单元和可编程连线三部分。FPGA 内部阵列块之间采用分段式进行互连，结构比较灵活，但是延时不可预测，比较适合于触发器多的逻辑相对简单的数据型系统。FPGA 保存逻辑功能的物理结构多为 SRAM 型，即掉电后将丢失原有的逻辑信息，所以在使用中需要为 FPGA 芯片配置一个专用 ROM，将设计好的逻辑信息烧录到此配置芯片中。系统上电时，FPGA 就能自动从配置的芯片中读取逻辑信息。FPGA 可实时地对外围或内置的 RAM 或 ROM 编程，实时配置器件功能，可进行现场编程或在线配置。

CPLD 是一种基于乘积项的可编程逻辑器件，主要由可编程逻辑宏单元、可编程输入/输出单元和可编程内部连线组成。CPLD 内部采用固定长度的线进行各逻辑块的互连，因此引脚和引脚的延迟时间几乎是固定的，与逻辑设计无关，设计调试比较简单，毛刺比较容易处理，性价比较高。CPLD 具有很宽的输入结构，适合逻辑复杂、输入变量多、对触发器的需求量相对较少的逻辑型系统。CPLD 结构大多为 EEPROM 或 Flash ROM 形式，具有编程后即可固定下载的逻辑功能，掉电不丢失原有的信息。

高集成度、高速度和高可靠性是 FPGA/CPLD 最显著的特点，其时钟延时可小至纳秒数量级，结合其并行工作方式，在超高速领域和实时测控方面有着广阔的应用前景。FPGA/CPLD 的集成规模非常大，可利用先进的 EDA 工具进行电子系统设计和产品开发。由于开发工具的通用性、设计语言的标准化及设计过程几乎与所用器件的硬件结构没有关系，因而设计开发成功的各类逻辑功能块软件有很好的兼容性和可移植性。它几乎可用于任何型号和规模的 FPGA/CPLD，从而使得产品设计效率大幅度提高。

国际上生产 FPGA/CPLD 的主流公司并在国内市场占有较大份额的是 Xilinx、Altera、

Lattice 三家公司。典型的 FPGA 产品有：Lattice 公司的 MachXO、ispXPGA、EC/ECP、ECP2/M、ECP3、FPSC 等系列；Altera 公司的 MAX II、STRATIX、ACEX1K 等系列；Xilinx 公司的 XC3000、XC4000、Spartan-6、Virtex-6 等系列。典型的 CPLD 产品有：Lattice 公司的 ispMACH4A5、ispMACH4000、ispXPLD5000 等系列；Altera 公司的 MAX3000A、MAX7000 等系列；Xilinx 公司的 CoolRunner-II、CoolRunnerXPLA3、XC9500/XL/XV 等系列。

对于一个开发项目，究竟是选择 FPGA 还是 CPLD，主要取决于开发项目本身的需要。对于大规模的 ASIC 设计或单片机系统设计，则多采用 FPGA，而普通规模且产量不是很大的项目，通常使用 CPLD。

1.2 FPGA/CPLD 的软件开发工具

目前，比较流行的、主流厂家的开发环境有 Altera 公司的 Quartus II、Xilinx 公司的 ISE/ISE-WebPACK series 及 Lattice 公司的 ispLEVER，这些软件的基本功能相同，主要的区别是面向的目标器件不一样，性能各有优劣。这些集成开发软件给 FPGA 带来了方便，同时为第三方 EDA 工具提供了接口。

常用的 FPGA/CPLD 开发工具大致包含设计输入编辑器、HDL 综合器、仿真器、适配器、下载编程等模块。Quartus II 开发环境软件是 Altera 公司在 21 世纪初推出的第四代 FPGA/CPLD 开发环境，它支持原理图、VHDL 和 Verilog 语言文本文件，以及以波形与 EDIF 等格式文件作为设计输入，并支持这些文件的任意混合设计。它具有门级仿真器，可以进行功能仿真和时序仿真，能够产生精确的仿真结果。适配之后所生成的供时序仿真用的 EDIF、VHDL 和 Verilog 这三种不同格式的网表文件界面友好、使用便捷并支持第三方 EDA 工具。Quartus II 提供了完整的多平台设计环境，可满足各种特定设计需要，同时集成了单芯片可编程 SoPC 开发环境，支持层次化设计等，为用户提供了优越的性能与无法比拟的系统级设计效率，缩短了产品开发周期，降低了开发成本，深受广大用户的欢迎。

1.3 硬件描述语言

硬件描述语言（HDL）是一种国际上流行的描述数字电路和系统的语言，可以在 EDA 工具的支持下，快速实现设计者的意图。在实际应用中最常见的 HDL 语言有 AHDL、Verilog HDL 和 VHDL 三种。

1.3.1 AHDL 语言

AHDL 语言是 Altera 公司发明的 HDL 语言，其优点是易学易用，学过高级语言的人可以在很短的时间内掌握，但缺点是其移植性不好，一般只用于 Altera 公司自己的开发系统。

1.3.2 Verilog HDL 语言

Verilog HDL 语言是由 GDA（Gatewag Design Automation）公司的 Philio P.Moorby 在 1983 年末首创的，最初只设计了一个仿真与验证工具，之后又陆续开发了相关的故障模拟与时序分析工具。1985 年 Moorby 推出商用仿真器 Verilog-XL，获得了巨大成功，从而使得 Verilog HDL 迅速得到推广应用。1989 年 Cadence 公司收购了 GDA 公司，Verilog HDL

成为了该公司的独家专利。1990年, Cadence公司公开发表了 Verilog HDL, 成立 OVI (Open Verilog International)组织, 并推动了 Verilog HDL 的发展。IEEE 于 1995 年制定了 Verilog HDL 的 IEEE 标准, 即 Verilog HDL1364—1995, 2001 年发布了 Verilog HDL1364—2001。

1.3.3 VHDL 语言

VHDL (Very-High-Speed Integrated Circuit Hardware Description Language) 是在 ADA 语言基础上发展起来的硬件描述语言, 起源于美国政府于 1980 年开始启动的超高速集成电路计划, 根据集成电路结构和功能描述需要, 于 1983 年由美国国防部发起创建, 并于 1987 年成为 IEEE 标准即 IEEE std 1076—1987, 后来又进行一些修改, 称为新的标准版本, 即 IEEE std 1076—1993。

VHDL 作为一种通用的硬件描述语言, 支持结构化和自顶向下的设计方法, 有助于设计的模块化; 可以支持各种不同类型的数字电路和系统设计; 既支持传输延时也支持惯性延时, 不仅可以很好地描述系统和电路的逻辑功能, 也可以真实地反映系统和电路的时间特性; 具有多层次描述和仿真系统硬件功能的能力, 可以从系统级到门级电路不同层次对数字进行建模和描述且不同的描述可混合使用, 简化硬件设计任务, 提高设计效率和可靠性, 缩短产品开发周期; 可以从一个模拟工具移植到另一个模拟工具, 从一个综合工具移植到另一个综合工具, 从一个工作平台移植到另一个工作平台上执行; 采用 VHDL 描述硬件电路时, 设计者不需要了解器件内部结构, 也与器件内部结构无关, 使得 VHDL 设计程序的硬件实现目标器件有广阔的选择范围。

目前, VHDL 成为硬件描述语言标准之一, 得到了众多的 FPGA/CPLD 开发平台的支持, 广泛应用于电子工程设计领域。

1.4 FPGA/CPLD 的开发流程

一般来说, 使用 VHDL 语言进行数字系统的开发要和具体的集成开发环境结合起来, VHDL 语言的设计文件程序需要依靠集成开发环境转换为实际可用的电路网表, 最后生成用于 IC 生产的版图, 或者由适配软件用此网表对 FPGA/CPLD 进行布线。采用 FPGA/CPLD 开发工具进行硬件电路设计大多采用自顶向下的设计方法, 即从系统总体要求出发, 自顶向下将系统逐步分解为各个子系统和模块, 直到整个系统中各子系统关系合理, 便于逻辑级的设计和实现为止。

FPGA/CPLD 开发流程主要包括设计准备、设计输入、综合、仿真、适配、下载及硬件测试等步骤。

(1) 设计准备

系统设计之前, 首先要进行方案论证、系统设计和芯片选择等设计准备工作。设计者首先根据任务要求, 如系统完成的功能及复杂程度, 对工作速度和器件本身的资源、成本及连线的可行性等方面进行权衡, 选择合适的设计方案和合适的器件类型, 通常采用自顶向下的设计方法。

(2) 设计输入

设计输入常用的方法是硬件描述语言和原理图输入方式, 硬件描述语言是用文本方式设计输入, 分为普通硬件描述语言和行为描述语言。普通语言支持逻辑方程、真值表、状态机

等简单的设计输入;行为描述语言是常用的高层硬件描述语言,主要有 VHDL 和 Verilog HDL 两个 IEEE 标准,这种方式具有很强的逻辑描述和仿真功能,输入效率高,不需对底层电路和 FPGA/CPLD 结构很熟悉。原理图输入方式是一种最直接的描述方式,使用 FPGA/CPLD 开发工具提供的元器件库及各种符号和连线画出原理图,形成原理图输入文件。这种输入方式容易实现仿真,便于信号的观察和电路的调整,但这种方式要求设计者有丰富的电路知识并对器件结构比较熟悉。

(3) 功能仿真

用户所设计的电路必须在编译之前进行逻辑功能验证,即功能仿真,也叫前仿真,此时的仿真没有延时信息,对于初步的功能检测来说非常方便。仿真前需建立波形文件和测试向量,仿真结果会生成报告文件并输出仿真波形,从中可以观察到各节点的信号变化,发现错误,可返回设计输入中修改逻辑设计。

(4) 逻辑综合和优化

在设计过程中,编译器对设计输入文件进行逻辑化简、综合优化,并适当地用一片或多片器件自动进行适配,最后产生编程用的编程文件。如果要把 VHDL 的软件设计与硬件的可实现性挂钩,需要利用 EDA 软件系统的综合器进行逻辑综合。而所谓的逻辑综合就是将电路的高级语言转换成版图表示,或转换到 FPGA/CPLD 的配置网表文件,有了版图信息就可以把芯片生产出来了。有了对应的配置文件,就可以使对应的 FPGA/CPLD 变成具有专门功能的电路器件。

(5) 目标器件的布线/适配

逻辑综合通过后必须利用适配器将综合后的网表文件针对某一具体的目标器进行逻辑映射操作,其中包括底层器件配置、逻辑分割、逻辑优化、布线与操作,适配完成后可以利用适配所产生的仿真文件做精确的时序仿真。

适配器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,产生最终的下载文件,如 JEDEC 格式的文件。适配所选定的目标器件(FPGA/CPLD)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结构相对应。

(6) 时序仿真

时序仿真也叫延时仿真,是在选择了具体器件并完成布局、布线后进行的时序关系仿真。由于不同器件的内部延时不一样,不同的布局布线方案也给延时造成不同的影响,因此适配后,对系统和各模块进行时序仿真,分析其时序关系,估计设计的性能及检查和消除竞争冒险等是非常有必要的。

(7) 目标器件的编程/下载

时序仿真完成后,软件就可以产生供器件编程使用的数据文件,然后下载到对应的具体 FPGA/CPLD 芯片中去。

器件编程需要满足一定的条件,如编程电压、编程时序和编程算法等。普通的 CPLD 和一次性编程的 FPGA 需要专用的编程器完成器件的编程工作。基于 SRAM 的 FPGA 可以由 EPROM 或其他存储体进行配置。在线可编程器件则不需要专门的编程器,只需一根编程下载连接线即可。器件在编程完毕后,可以用编译时产生的文件对器件进行校验、加密等工作。对于具有边界扫描测试能力、支持 JTAG 技术及在线编程能力的器件来说,测试起来会更加方便。

第2章

Quartus II 集成开发环境

Quartus II 是 Altera 公司在 2001 年推出的第四代开发工具,是一个集成化的多平台设计环境,能够直接满足特定设计需要,在 FPGA 和 CPLD 设计各个阶段都提供了工具支持,并为可编程片上系统 (SoPC) 提供全面的设计环境,是一个系统级的高效的 EDA 设计工具。Quartus II 集成开发环境包括设计输入、综合、布局/布线、时序分析、仿真和编程/配置等,具有功能强大、界面友好、易于掌握等特点。本章主要介绍 Quartus II 集成开发环境的使用方法。

2.1 Quartus II 界面概况

Quartus II 由不同的窗口构成,本节结合 Quartus II 13.1 版本软件的图形界面,介绍 Quartus II 软件的应用开发环境。

在桌面上双击 Quartus II 13.1 图标,启动 Quartus II 13.1 应用程序,进入如图 2-1 所示的主界面。该界面主要由以下 5 部分构成:

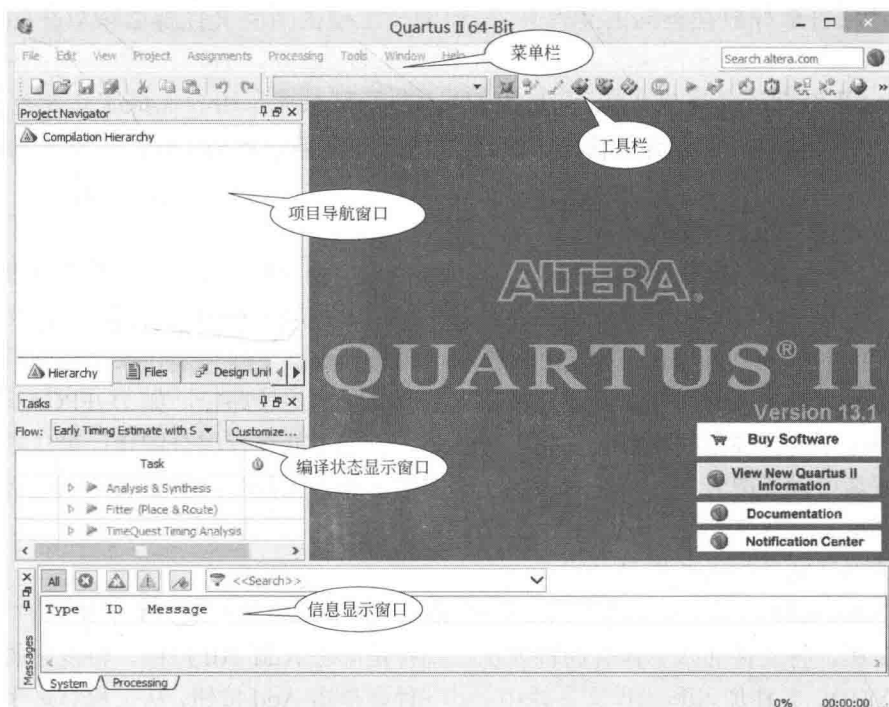


图 2-1 Quartus II 13.1 主界面

(1) 菜单栏

Quartus II 的菜单栏是设计过程中所需要的功能命名, 以完成相应的操作, 主要包括文件菜单 File、编辑菜单 Edit、视图菜单 View、工程菜单 Project、工程配置菜单 Assignments、操作菜单 Processing、工具菜单 Tool 等。

(2) 工具栏

工具栏中是常用命令的快捷图标, 这些图标在菜单栏内可以找到相应的命令。

(3) 项目导航窗口

该窗口由三个标签页构成, 分别为层次体系标签页 Hierarchy、工程文件标签页 Files、设计单元标签页 Design Units, 用户可以在该窗口管理工程项目, 给工程项目添加、移除设计资源或者根据设计需要调整文件之间的层次关系。

(4) 编译状态显示窗口

该窗口提示编译的过程步骤、编译进度及编译步骤所耗费的时间信息。

(5) 信息显示窗口

该窗口实时提供系统信息、定时、警告及相关错误信息等。

2.2 Quartus II 软件开发流程

2.2.1 创建工程

任何一项设计都是一个工程, 在开始设计一个具体的项目之前, 首先要建立一个工作文件夹, 以便存储工程项目文件, 此文件夹被 Quartus II 软件默认为工作库。一般而言, 不同的设计项目最好放在不同的文件夹中, 而同一工程的所有文件都必须放在同一个文件夹中。

Quartus II 为设计者提供了工程设计向导, 向导可以提示用户完成工作文件夹设置、工程名设置、目标器件的指定、仿真器和综合器的选择等一系列工作, 其具体的设计过程如下:

(1) 打开建立新工程管理窗口

在菜单栏中, 选择 File→New Project Wizard 命令, 弹出如图 2-2 所示的对话框。从上到下依次指定工程目录、工程名及顶层文件名。工程目录存放工程项目所有文件的文件夹, 默认路径为 D:/alter/13.1/quartus。为方便管理, 一般另外设置文件夹来存放工程项目, 单击后面的浏览按钮, 选择新的文件夹, 或直接输入工程目录的地址路径, 如 D:/FPGA/shelen, 这里要注意的是不能将硬盘的根目录作为工程目录, 否则不能进行综合编译; 第二项设置工程名, 此工程名可以任取名字, 一般用顶层文件实体名作为工程名; 第三项是顶层文件的实体名, 这里都用 shelen 来命名。

(2) 将设计文件加入工程

单击 Next 按钮, 弹出如图 2-3 所示的对话框, 在该框中将工程有关的所有 VHDL 文件加入此工程。将文件加入工程有两种方法: 一种是单击 Add All 按钮, 将设定的工程目录中的所有 VHDL 文件加入到工程文件栏中; 另一种是单击 Add 按钮, 从工程目录中选出相关的 VHDL 文件。由于是新建文件, 所以此处不需要设置。

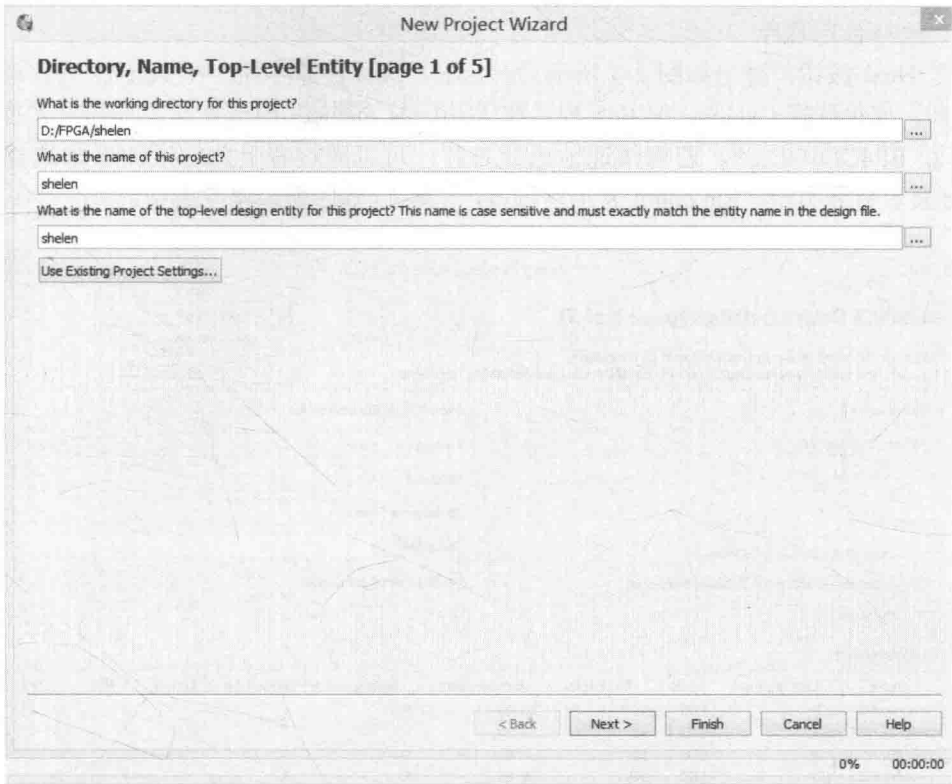


图 2-2 项目目录及名称

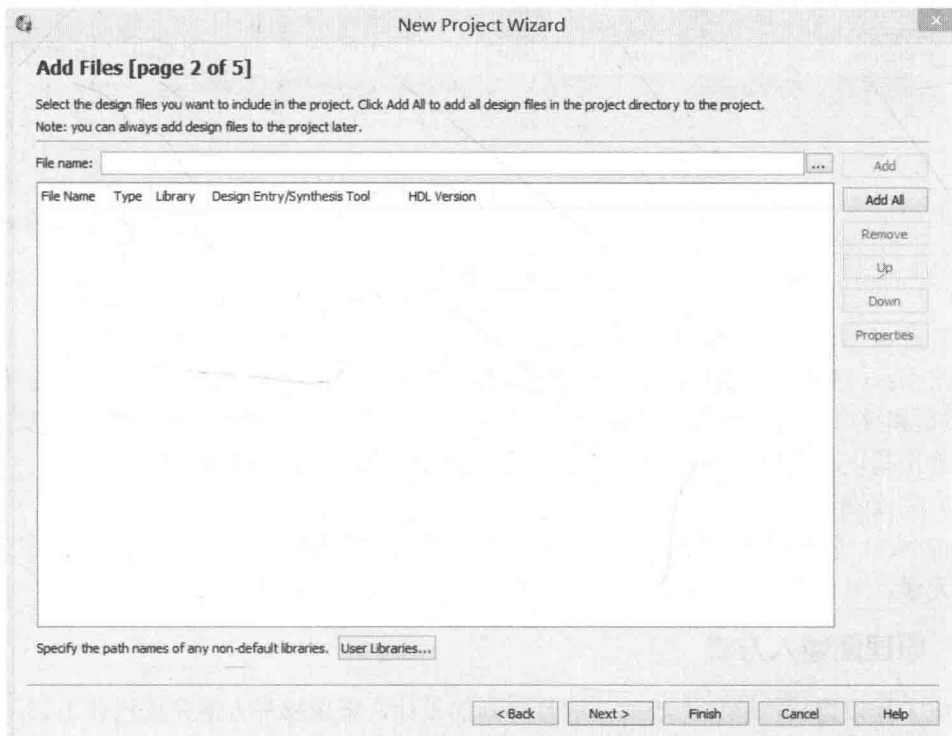


图 2-3 将文件加入工程