



普通高等教育“十二五”电子信息类规划教材

基于Quartus II的CPLD的 数字系统设计与实现

王忠林 曹献美 编著

免费
电子课件

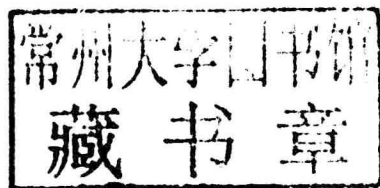


机械工业出版社
CHINA MACHINE PRESS

普通高等教育“十二五”电子信息类规划教材

基于 Quartus II 的 CPLD 的 数字系统设计与实现

王忠林 曹献美 编著



机械工业出版社

本书是山东省精品课程“电子技术”的配套教材之一，第1章介绍了 Quartus II 9. X 的开发流程，第2章介绍了 CPLD 的相关知识，第3章介绍门电路的实现方法，第4章介绍组合逻辑电路，第5章介绍了组合逻辑电路的应用，第6章介绍了触发器的知识，第7章介绍了时序逻辑电路，第8章介绍了时序逻辑电路的设计及应用，第9章是基于原理图的综合设计实例，第10章是基于混合输入的综合设计。

本书可作为高等院校自动化、电子信息、计算机及其他相关专业的教材，也可供从事电子设计的工程技术人员参考。

本书配有免费电子课件，欢迎选用本书作教材的老师发邮件到 jinacmp@163.com 索取，或登录 www.cmpedu.com 注册下载；书中的例程和配套的实验视频索取邮箱是：bzcong@126.com。

图书在版编目 (CIP) 数据

基于 Quartus II 的 CPLD 的数字系统设计与实现/王忠林等编著.
—北京：机械工业出版社，2012.4
普通高等教育“十二五”电子信息类规划教材
ISBN 978-7-111-37582-1

I. ①基… II. ①王… III. ①可程序逻辑器件-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 (2012) 第 032059 号

机械工业出版社 (北京市百万庄大街 22 号 邮政编码 100037)
策划编辑：吉玲 责任编辑：吉玲 王寅生 刘丽敏
版式设计：霍永明 责任校对：李锦莉
封面设计：张静 责任印制：杨曦
北京京丰印刷厂印刷
2012 年 6 月第 1 版·第 1 次印刷
184mm×260mm·17 印张·531 千字
标准书号：ISBN 978-7-111-37582-1
定价：35.00 元

凡购本书，如有缺页、倒页、脱页，由本社发行部调换

电话服务

网络服务

社服务中心：(010)88361066

门户网：<http://www.cmpbook.com>

销售一部：(010)68326294

教材网：<http://www.cmpedu.com>

销售二部：(010)88379649

读者购书热线：(010)88379203

封面无防伪标均为盗版

前 言

科学实验是科学技术发展的重要手段之一，许多科学上的发明和创造都是从实验过程中获得启示，许多科学理论的建立都是从实验中见到先兆，并通过实验得到验证。因此实验技能是科技工作者的一项基本功，实验教学是高等院校的重要教学环节之一。

随着 EDA 技术的发展，其在电子信息、通信、自动控制及计算机等领域的重要性日益突出。与此同时，随着技术市场与人才市场对 EDA 技术需求的不断提高，产品的市场和技术要求也必然反映到教学和科研领域中来。

本书将 EDA 技术引入到数字电路实验的教学中来，将数字电路中的所有实验都采用 EDA 技术来实现，为方便初学者，第 1 章介绍了 Quartus II 9. X 的开发流程，第 2 章介绍了 CPLD 的相关知识，第 3 章介绍门电路的实现方法，第 4 章介绍组合逻辑电路，第 5 章介绍了组合逻辑电路的应用，第 6 章介绍了触发器的知识，第 7 章介绍了时序逻辑电路，第 8 章介绍了时序逻辑电路的设计及应用，第 9 章是基于原理图的综合设计实例，第 10 章是基于混合输入的综合设计。本书为传统数字电路和 VHDL 语言之间架起了桥梁，既有利于用先进技术来处理传统的数字电路问题，又便于读者后续学习 VHDL 语言等课程。

本书所有的实验均已在书中讲述的实验系统上进行了验证，对其他的实验系统，在教学上只要修改相应的引脚设置，也可以通用。书中的所有例程可以和作者联系索取，电子邮箱是：bzcong@126.com。为便于读者观察实验现象，本书还配有实验的视频，也可联系作者索取。

本书是山东省精品课程“电子技术”的配套教材之一，也是编者近年来从事教学、实验及科研工作的总结。本书受以下项目和机构资助：滨州学院自制实验仪器设备项目（BZXYZZSBXM201105），滨州学院教材出版基金项目（BZXYJC200903），滨州学院教学研究重点项目（BZXYJY200806），山东省高校航空信息与控制技术重点实验室（滨州学院）。本书的出版得到滨州学院教务处和物理与电子科学系的大力支持，张士国处长和时宝国书记对本书的出版给予了极大的支持，高平东、马建等同学对书中的部分电路进行了实验验证，在此一并表示衷心的感谢！

因编者水平所限，错误和疏漏在所难免，敬请同仁和读者不吝指正。

编 者

目 录

前言

第1章 Quartus II 9. X 开发流程 1

1.1 Quartus II 概述 1

1.2 面向FPGA/CPLD的开发流程 3

1.2.1 设计输入 3

1.2.2 综合 4

1.2.3 布线布局(适配) 4

1.2.4 仿真 5

1.2.5 下载和硬件测试 5

1.3 Quartus II 操作流程 5

1.3.1 创建工程 5

1.3.2 设计文件输入 7

1.3.3 编译前设置 9

1.3.4 全程编译 10

1.3.5 时序仿真 11

1.3.6 RTL级电路 15

1.3.7 引脚锁定 16

1.3.8 编程下载 17

1.3.9 其他下载方式 18

1.4 Quartus II 9. X 的使用 19

1.4.1 原理图电路设计方法 19

1.4.2 层次化设计流程 22

第2章 CPLD 简介 25

2.1 基本PLD的原理及分类 25

2.1.1 PLD的原理概述 25

2.1.2 PLD的分类 25

2.2 CPLD的结构与原理 26

2.2.1 逻辑阵列块 27

2.2.2 宏单元 27

2.2.3 扩展乘积项 28

2.2.4 可编程连线阵列 28

2.2.5 I/O控制模块 29

2.3 数字电路设计的常用外围模块电路 29

2.3.1 秒信号发生电路 29

2.3.2 单脉冲产生电路 29

2.3.3 电平保持产生电路 30

2.3.4 指示灯电路 30

2.3.5 数码管显示电路 31

2.3.6 矩阵键盘 32

2.3.7 发声电路 33

2.4 数字电路设计的其他辅助电路 33

2.4.1 电源电路 33

2.4.2 复位电路 34

2.4.3 时钟电路 34

第3章 门电路 35

3.1 基本门电路 35

3.1.1 与逻辑 35

3.1.2 或逻辑 37

3.1.3 非逻辑 39

3.2 复合逻辑电路 41

3.2.1 与非逻辑电路 41

3.2.2 或非逻辑电路 43

3.2.3 异或逻辑电路 44

3.2.4 同或逻辑电路 46

3.2.5 与或非逻辑电路 47

第4章 组合逻辑电路 50

4.1 加法器 50

4.1.1 半加法器 50

4.1.2 全加法器 54

4.1.3 4位串行进位加法器 57

4.1.4 4位超前进位加法器 59

4.2 比较器 63

4.2.1 1位比较器 63

4.2.2 4位扩展比较器 66

4.3 编码器 69

4.3.1 二进制编码器 69

4.3.2 8线-3线扩展优先编码器 72

4.3.3 其他编码器 76

4.4 译码器 80

4.4.1 二进制译码器 80

4.4.2 3位二进制扩展译码器 84

4.4.3 显示译码器 88

4.5 数据选择器 93

4.5.1 4选1数据选择器 93

4.5.2 字扩展数据选择器 96

4.5.3 位扩展数据选择器 98

4.6 数据分配器 101

4.6.1 1路-4路数据分配器 101

4.6.2 数据分配器的扩展 104

第5章 组合逻辑电路应用 107

5.1 组合逻辑电路的设计与测试方法·····	107	7.4 基本寄存器·····	181
5.1.1 组合逻辑电路的一般设计 方法·····	107	7.5 移位寄存器·····	185
5.1.2 组合逻辑电路的测试方法·····	107	7.5.1 单向移位寄存器·····	185
5.2 加法器应用·····	108	7.5.2 双向移位寄存器·····	189
5.3 比较器应用·····	109	7.5.3 8位移位寄存器·····	192
5.4 编码器应用·····	111	7.6 移位寄存器型计数器·····	195
5.4.1 16线-4线优先编码器·····	111	7.6.1 环形计数器·····	195
5.4.2 医院呼叫灯的控制电路·····	115	7.6.2 扭环形计数器·····	198
5.5 译码器应用·····	117	第8章 时序逻辑电路的设计及 应用 ·····	202
5.5.1 4线-16线二进制译码器·····	117	8.1 时序逻辑电路的设计与测试方法·····	202
5.5.2 设计一个编码信号显示电路·····	119	8.1.1 时序逻辑电路的一般设计方法·····	202
5.6 数据选择器应用·····	121	8.1.2 同步时序逻辑电路的设计步骤和 方法·····	202
5.7 竞争冒险·····	123	8.1.3 时序逻辑电路的测试方法·····	203
5.7.1 险象的判断·····	124	8.2 <i>N</i> 进制计数器·····	205
5.7.2 险象的解决方法·····	125	8.2.1 二十四进制计数器·····	205
第6章 触发器 ·····	127	8.2.2 六十进制计数器·····	209
6.1 基本触发单元·····	127	8.3 时序逻辑电路的应用·····	212
6.1.1 基本触发器·····	127	8.3.1 计数器的应用·····	212
6.1.2 同步触发器·····	130	8.3.2 异步计数器的应用·····	215
6.1.3 边沿触发器·····	133	8.3.3 简易数字秒表的设计·····	217
6.2 触发器之间的转换·····	137	8.3.4 简易数字钟的设计·····	220
6.2.1 JK触发器转换成其他触发器·····	137	第9章 基于原理图的综合设计 实例 ·····	225
6.2.2 D触发器转换成其他触发器·····	139	9.1 数字频率计·····	225
第7章 时序逻辑电路 ·····	142	9.2 交通灯控制电路·····	230
7.1 同步二进制计数器·····	142	9.3 电风扇控制电路·····	234
7.1.1 同步3位二进制加/减法计 数器·····	142	9.4 跑马灯设计·····	238
7.1.2 同步3位二进制可逆计数器·····	147	9.5 洗衣机控制器·····	241
7.1.3 带控制同步4位二进制加法/可逆 计数器·····	150	第10章 基于原理图和VHDL混合输入 的综合设计实例 ·····	246
7.2 异步二进制计数器·····	155	10.1 篮球比赛进攻24s计时器·····	246
7.2.1 二进制异步加/减法计数器·····	155	10.2 出租车自动计费器·····	251
7.2.2 二进制扩展异步计数器·····	160	10.3 电梯控制器·····	254
7.3 十进制计数器·····	164	10.4 4×4键盘扫描电路设计·····	258
7.3.1 同步十进制加/减法计数器·····	164	附录 实验系统原理图和实验系统 实物图 ·····	263
7.3.2 同步十进制可逆计数器·····	168	参考文献 ·····	265
7.3.3 带控制1位同步十进制加法 计数器·····	171		
7.3.4 单/双时钟1位同步十进制可逆 计数器·····	175		

第 1 章 Quartus II 9. X 开发流程

电子设计自动化 (Electronic Design Automation, EDA) 是发展迅速的新技术, 应用范围日益扩大, 已经成为电子设计工程师们必备的开发手段之一。从可编程逻辑器件的应用角度来理解 EDA 技术, 可以认为 EDA 技术是以可编程逻辑器件为设计载体, 以计算机为设计工具, 以硬件描述语言为主要设计描述方式, 利用软件开发工具完成电子系统设计的全过程, 最终将设计系统集成到特定的芯片中, 完成电子系统芯片化的设计。

EDA 技术以计算机为工具, 硬件设计者只需用硬件描述语言完成对系统的描述, 其他工作都交给计算机及软件开发工具处理。因此硬件设计与传统的手持电烙铁, 面对电路图、元器件的方式已经完全不同, 今天的硬件设计如同软件设计, 面对的是计算机屏幕和程序, 硬件方案的修改如同软件方案的修改一样方便快捷。

为了完成 EDA 课程设计的系统设计要求, 本书将从课程设计开发工具的使用、硬件描述语言、可编程逻辑器件原理等多个方面, 全面介绍 EDA 课程设计所需的基础知识, 使读者对 EDA 技术有较为全面的了解, 为完成 EDA 课程设计提供必要的技术支持。本书中的硬件描述语言主要以原理图的形式输入, 使读者把精力更专注于系统设计上, 而把软件本身的难度降到最低。本章主要介绍课程设计开发工具 Quartus II 的使用以及操作流程。

1.1 Quartus II 概述

本书给出的所有示例和实验都是基于 Quartus II 9.0 的, 其应用方法和设计流程对于其他流行 EDA 工具的使用具有一定的典型性和普遍性, 所以在此对它作一些介绍。

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 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 上可以完成面向 FPGA/CPLD 的整个开发流程, 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 VHDL、Verilog 的设计流程, 其内部嵌有 VHDL、Verilog 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、FPGA Compiler II, 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 系统开发和数字通信模块的开发。

Quartus II 包括模块化的编译器。编译器包括的功能模块有分析与综合器 (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)、编辑数据接口 (Compiler Database Interface) 等。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 窗口中运行该模块来启动编译器模块。在 Compiler Tool 窗口中, 可以打开该模块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), 它们是复杂或高级系统构建的重要组成部分, 在 SOPC 设计中被大量使用, 也可与 Quartus II 普通设计文件一起使用。Altera 提供的 LPM 函数均基于 Altera 器件的结构做了优化设计。在许多实用情况中, 必须使用宏功能模块

才可以使用一些 Altera 特定器件的硬件功能，如各类片上存储器、DSP 模块、LVDS 驱动器、PLL 以及 SERDES 和 DDIO 电路模块等。

图 1-1 所示的下排是 Quartus II 编译设计主控界面，它显示了 Quartus II 自动设计的各主要处理环节和设计流程，包括设计输入编辑、设计分析与综合、适配器、编程文件汇编（装配）、时序参数提取以及编程下载几个步骤。在图 1-1 所示上排的流程框图是与下面的 Quartus II 设计流程相对照的标准的 EDA 开发流程。

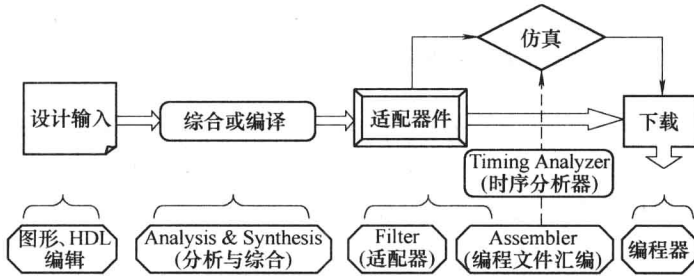


图 1-1 Quartus II 设计流程

Quartus II 编译器支持的硬件描述语言有 VHDL（支持 VHDL/87 及 VHDL/97 标准）、Verilog HDL 及 AHDL（Altera HDL）。

Quartus II 支持层次化设计，可以在一个新的编辑输入环境中对使用不同输入设计方式完成的模块（元件）进行调用，从而解决了原理图与 HDL 混合输入设计的问题。在设计输入之后，Quartus II 的编译器将给出设计输入的错误报告。可以使用 Quartus II 带有的 RTL Viewer 观察综合后的 RTL 图。

运行 Quartus II 9.0，可以看到 Quartus II 的管理器窗口，如图 1-2 所示。管理器窗口主要包含项目导航窗口、任务窗口、消息窗口，可以通过 View→Utility Windows 菜单下的选项添加或隐藏这些窗口。

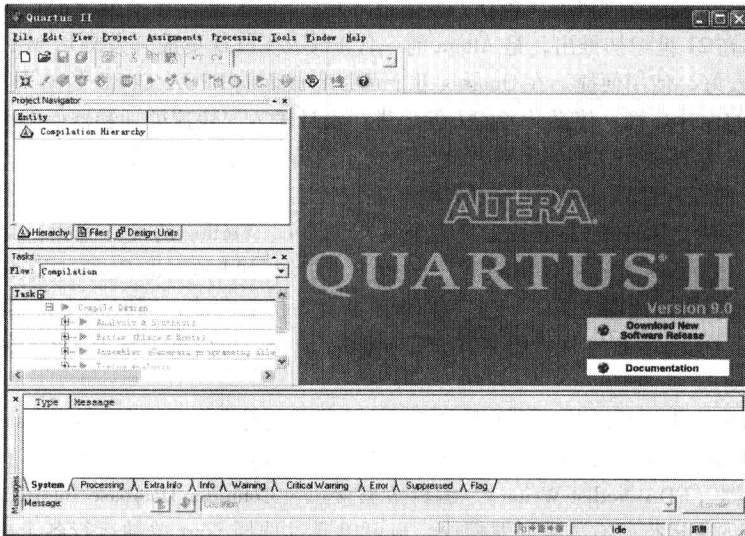


图 1-2 Quartus II 9.0 管理器窗口

为了保证 Quartus II 的正常运行，第一次运行软件，需要设置 license.dat 文件，否则工具的许多功能将被禁用。在 Quartus II 管理器窗口选择 Tools→License Setup…，单击 License file 的“…”按钮，在出现的对话框中选择 license.dat 文件或直接输入具有完整路径的文件名，如图 1-3 所示。

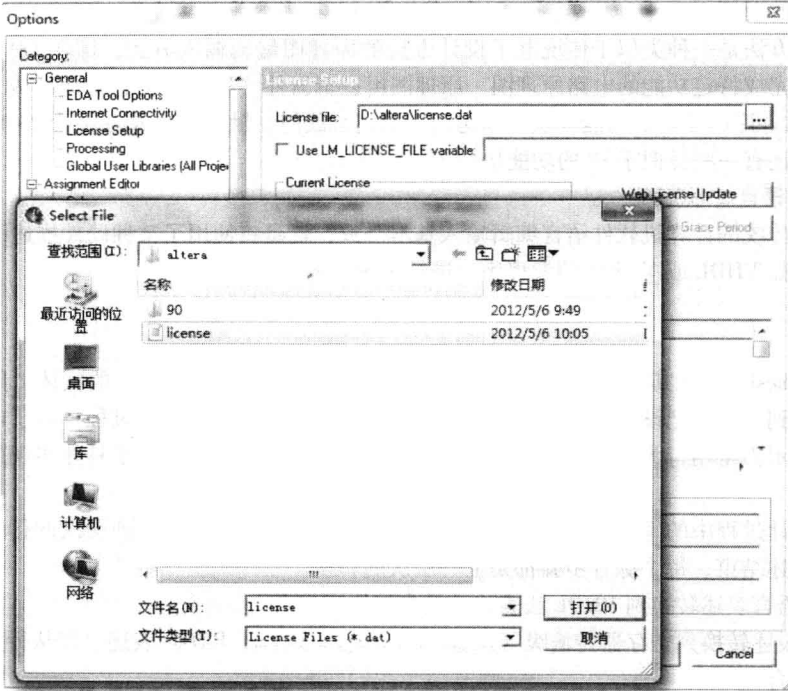


图 1-3 设置 license.dat 文件

1.2 面向 FPGA/CPLD 的开发流程

完整地理解利用 EDA 技术进行设计开发的流程对于正确地选择和使用 EDA 软件、优化设计项目、提高设计效率十分有益。一个完整的、典型的 EDA 设计流程既是自上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。

1.2.1 设计输入

图 1-4 所示是基于 EDA 软件的 FPGA 开发流程框图，下面将分别介绍各个设计模块的功能特点。对于目前流行的用于 FPGA/CPLD 开发的 EDA 软件，图 1-4 所示的设计流程具有普遍性。

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以下两种类型。

1. 图形输入

图形输入通常包括状态图输入、波形图输入和原理图输入等方法。

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。

波形图输入方法则是将待设计的电路看成一个黑盒子，只需告诉 EDA 工具该黑盒子电路的输入和输出时序波形图，EDA 工具即能据此

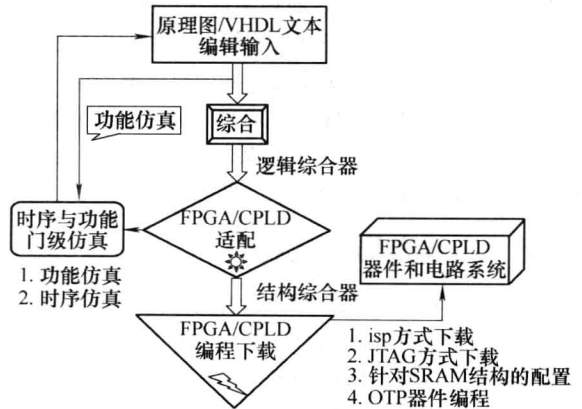


图 1-4 FPGA/CPLD 的 EDA 开发流程

完成黑盒子电路的设计。

原理图输入方法是一种类似于传统电子设计方法的原理图编辑输入方式，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原理图。原理图由逻辑器件（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，甚至还有一些类似于 IP 的功能块。

2. 硬件描述语言文本输入

这种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（HDL）的电路设计文本，如 VHDL 或 Verilog 的源程序，进行编辑输入。

1.2.2 综合

综合（Synthesis），就其字面含义应该为把抽象的实体结合成单个或统一的实体。因此，综合就是把某些东西结合到一起，把设计抽象层次中的一种表述转化成另一种表述的过程。对于电子设计领域的综合概念可以表示为将用行为和功能层次表达的电子系统转换为低层次的便于具体实现的模块组合装配而成的过程。

事实上，设计过程中的每一步都可称为一个综合环节。设计过程通常从高层次的行为描述开始，以最底层的结构描述结束，每个综合步骤都是上一层级的转换。

1) 从自然语言表述转换到 VHDL 法表述，是自然语言综合。

2) 从算法表述转换到寄存器传输级（Register Transport Level, RTL）表述，即从行为域到结构域的综合，是行为综合。

3) 从 RTL 表述转换到逻辑门（包括触发器）的表述，即逻辑综合。

4) 从逻辑门表述转换到版图表述（ASIC 设计），或转换到 FPGA 的配置网表文件，可称为版图综合或结构综合。

一般地，综合是仅对应于 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 源文件中的逻辑描述语句都是可综合的。

在综合之后，HDL 综合器一般都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，在这种网表文件中用各自的格式描述电路的结构，如在 VHDL 网表文件中采用 VHDL 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低层次的电路描述网表文件。

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，它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，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纯的或一个方向的。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。

1.2.3 布线布局（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 格式的文件。适配器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。

适配器就是将综合后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、优化、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，同时产生可用于编程的文件。

1.2.4 仿真

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，就是所谓的仿真。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，以验证设计，排除错误。仿真是在 EDA 设计过程中的重要步骤。图 1-4 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（当然也可以选用第三方的专业仿真工具），它可以完成两种不同级别的仿真测试。

1) 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适配器。综合后所得的 EDIF 等网表文件通常作为 FPGA/CPLD 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。

2) 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要求的。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，在设计项目编辑编译（或综合）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。

1.2.5 下载和硬件测试

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/CPLD 进行下载，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。

最后是将含有载入了设计的 FPGA/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，以排除错误，改进设计。

1.3 Quartus II 操作流程

完成电路设计后，必须借助于 EDA 工具中的综合器、适配器、时序仿真器和编程器等工具进行相应的处理，才能使此项设计在 FPGA/CPLD 上完成硬件实现，并得到硬件测试。在 EDA 工具的设计环境中，有多种途径来完成目标电路系统的表达和输入方式，如 HDL 的文本输入方式、原理图输入方式、状态图输入方式以及混合输入方式等。相比之下，HDL 文本输入方式最基本、最直接。本节将通过实例详细介绍基于 Quartus II 的 VHDL 文本输入设计流程，包括设计输入、综合、适配、仿真测试和编程下载等方法。在设计任何阶段出现错误，都需要进行修改，纠正错误，重复上述过程，直至每个阶段都正确为止。

下面以一个 8 位二进制计数器 CNT8.vhd 的设计为例，介绍如何经过设计的各个阶段，最终将生成的下载文件下载到 FPGA/CPLD 芯片中，完成 8 位二进制计数器设计的完整过程，从而学习 Quartus II 的操作流程。

1.3.1 创建工程

任何一项设计都是一项工程（Project），都必须首先为此工程建立一个放置与此工程相关的所有设计文件的文件夹。此文件夹将被 EDA 软件默认为工作库（Work Library）。Quartus II 编辑器的工作对象是项目，项目用来管理所有设计文件以及编辑设计文件过程中产生的中间文档。为方便项目的管理，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文件夹中。还应该特别注意，不要将文件夹设在计算机已有的安装目录中，也不要将工程文件夹建立在“桌面”上，更不要将工程文件直接放在安装目录中。在一个项目下，可以有多个设计文件，这些设计文件的格式可以是原理

图文件、文本文件（如 AHDL、VHDL、Verilog HDL 等文件）、符号文件、底层输入文件以及第三方 EDA 工具提供的多种文件格式，如 EDIF、HDL、VQM 等。

在此要利用 New Project Wizard 工具选项创建此设计工程，即令顶层设计 CNT8.vhd 为工程，并设定此工程的一些相关信息，如工程名、目标器件、综合器、仿真器等。

1) 打开建立新工程管理窗口。在 Quartus II 管理器窗口中选择菜单 File→New Project Wizard...，出现新建项目向导 New Project Wizard 对话框的第一页，如图 1-5 所示。其中第一行的 D:\MyProject\CNT8B 表示工程所在的工作库文件夹；第二行的 CNT8 表示此项工程的工程名，工程名可以取任何其他的名，也可直接用顶层文件的实体名作为工程名，在此就是按这种方式取的名；第三行是当前工程顶层文件的实体名，这里即为 CNT8。

2) 添加或删除与该工程有关的文件。单击下方的 Next 按钮，将弹出如图 1-6 所示的对话框。在弹出的对话框中单击 Filename 栏后的按钮“...”可浏览文件选项，将与工程相关的所有 VHDL 文件（如果有）加入此工程。此工程文件加入的方法有两种：第一种是单击 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；第二种是单击 Add 按钮，从工程目录中选出相关的 VHDL 文件。也可以选中不相关的文件单击 Remove 按钮移除工程。初学者还没有建立文件，可以先跳过该页。

3) 选择目标芯片。单击下方的 Next 按钮，选择芯片。根据器件的封装形式、引脚数目和速度级别，选择目标器件。读者可以根据具备的实验条件进行选择，这里选择的芯片是 MAX II 系列中的 EPM240T100C5 芯片，如图 1-7 所示。

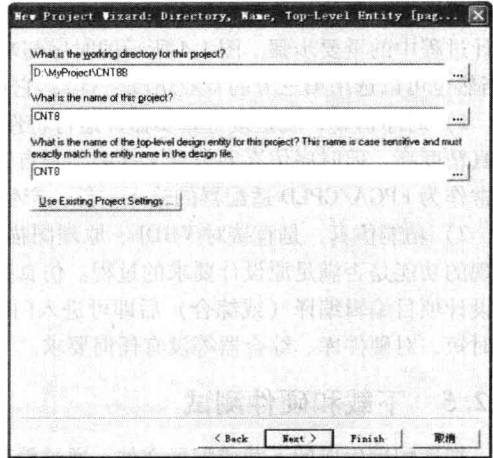


图 1-5 New Project Wizard 对话框的第一页

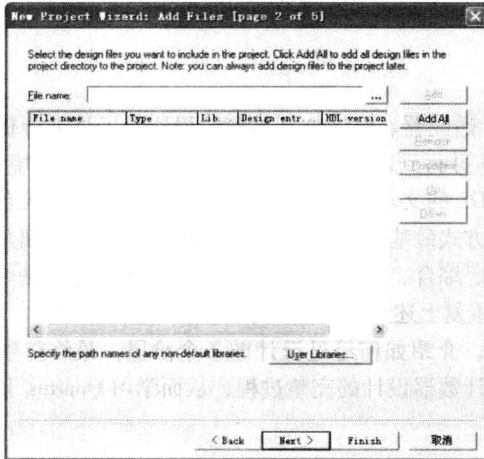


图 1-6 New Project Wizard 对话框的第二页

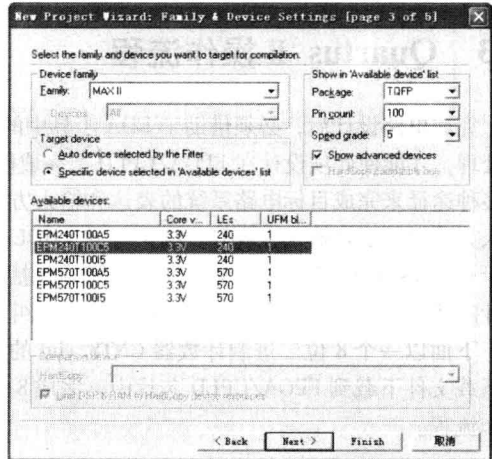


图 1-7 New Project Wizard 对话框的第三页

4) EDA 工具设置。单击 Next 按钮后，弹出的下一个对话框是 EDA 工具设置对话框——EDA Tool Settings，如图 1-8 所示。此窗口有 3 项选择：Design Entry/Synthesis 用于选择输入的 HDL 类型和综合工具；Simulation 用于选择仿真工具；Timing Analysis 用于选择时序分析工具，这是除 Quartus II 自含的所有设计工具以外，还包括外加的工具，因此，如果都不作选择，表示仅选择 Quartus II 自含的所有设计工具。对开发工具不熟悉的读者，建议采用系统默认选项。

5) 结束设置。再单击 Next 按钮后即弹出工程设置统计对话框, 上面列出了此项工程相关设置情况, 如图 1-9 所示。最后单击 Finish 按钮, 即已设定好此工程, 并出现 CNT8 的工程管理窗口, 或称 Compilation Hierarchies 窗口, 主要显示本工程项目的层次结构和各层次的实体名。

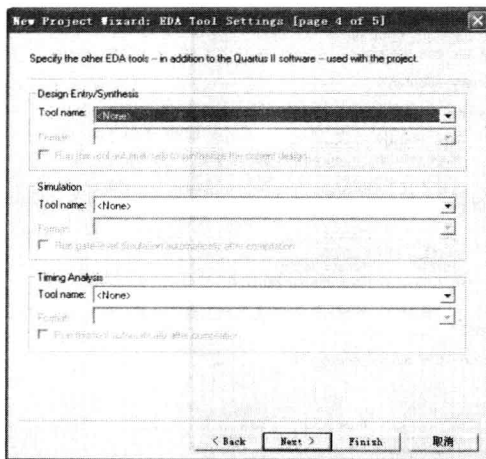


图 1-8 New Project Wizard 对话框的第四页

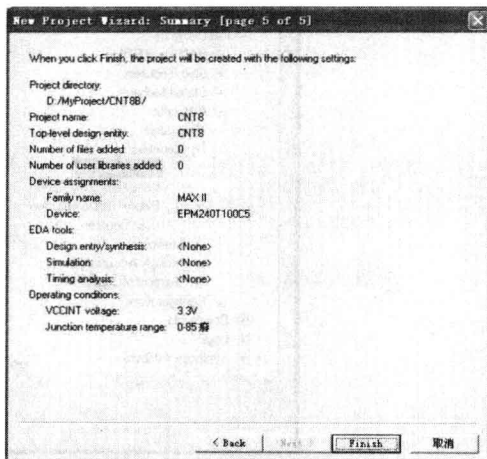


图 1-9 New Project Wizard 对话框的第五页

Quartus II 将工程信息存储在工程配置文件 (quartus) 中。它包含有关 Quartus II 工程的所有信息, 包括设计文件、波形文件、SignalTap II 文件、内存初始化文件等以及构成工程的编译器、仿真器和软件构建设置。

建立工程后, 新建工程向导中的各个选项, 在新建工程结束后, 仍然可以修改或重新进行设置, 通过选择菜单命令 Assignments→Settings…实现。

1.3.2 设计文件输入

Quartus II 支持 AHDL、VHDL 及 Verilog HDL 等硬件描述语言描述的文本文件, 关于如何用 VHDL 描述硬件电路请读者参考其他书籍。这里将结合实例说明如何使用文本编辑器模板输入 VHDL 文本文件。

新建 VHDL 文本文件, 在 Quartus II 管理器界面中选择菜单 File→New…, 或单击新建文件按钮, 出现 New 对话框, 如图 1-10 所示。在对话框 Design Files 中选择 VHDL File, 单击 OK 按钮, 打开文本编辑器。在文本编辑器窗口下, 按照 VHDL 的规则输入设计文件, 并将其保存, VHDL 文件的扩展名为 .vhd。不同的硬件描述语言编写的文件, 其文件扩展名不同, 如 AHDL 文件扩展名为 .tdf, Verilog HDL 文件扩展名为 .v。

Quartus II 提供了文本文件的编辑模板, 使用这些模板可以快速准确地创建 VHDL 文本文件, 避免语法错误, 提高编辑效率。例如, 用 VHDL 模板设计一个 8 位二进制计数器的 VHDL 文本文件。

1) 选择菜单 Edit→Insert Template…, 打开 Insert Template 对话框, 单击右侧 Language Template 栏目打开 VHDL, VHDL 栏目下显示出所有 VHDL 的程序模板, 如图 1-11 所示。

2) 在 VHDL 模板中选择 Full Designs→Arithmetic→Counters→Binary Counter, Insert Template 对话框的右侧会出现计数器模板程序的预览, 这是一个带清零和使能端的计数器模板。单击 Insert, 模板程序出现在文本编辑器中。

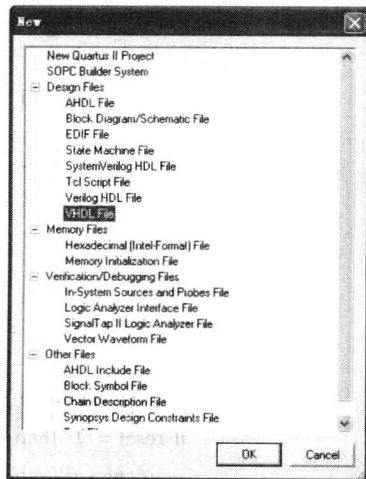


图 1-10 New 对话框

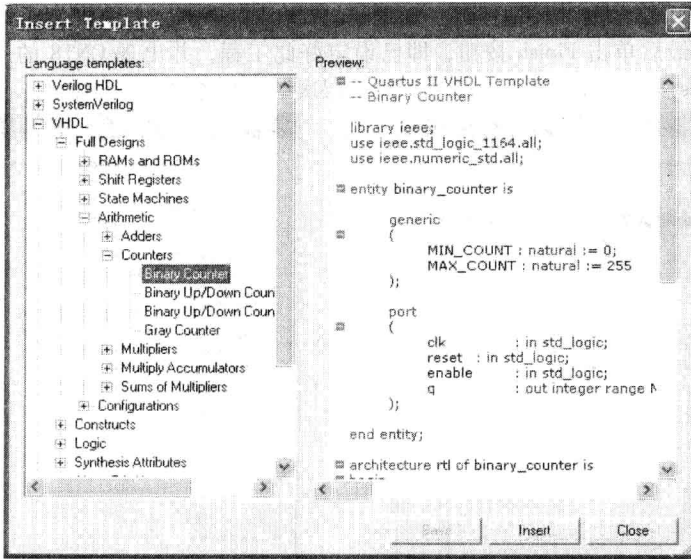


图 1-11 程序模板对话框

3) 根据设计要求, 对模板中的文件名、信号名、变量名等黑色部分的内容进行修改。将实体名 binary_counter 修改为 CNT8。修改后的 VHDL 代码如下:

```
-- Quartus II VHDL Template
-- Binary Counter
library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_unsigned.all;
use ieee.numeric_std.all;
entity CNT8 is
    port
    (
        clk          : in std_logic;
        reset        : in std_logic;
        enable       : in std_logic;
        q            : out std_logic_vector (7 downto 0)
    );
end entity;
architecture rtl of CNT8 is
begin
    process (clk)
        variable cnt : std_logic_vector(7 downto 0);
    begin
        if (rising_edge(clk)) then
            if reset = '1' then
                cnt := (others => '0');
            elsif enable = '1' then
```

```

        cnt := cnt + 1;
    end if;
end if;
end if;
q <= cnt;
end process;
end rtl;

```

1.3.3 编译前设置

对工程进行编译处理前，必须做好必要的设置。步骤如下：

1) 选择目标芯片。目标芯片的选择也可以这样来实现：选择 Assignments→Settings 命令，在弹出的对话框中选择 Category 项下的 Device。这里选择目标芯片为 EPM240T100C5（此芯片已在建立工程时选定了）。在 Settings 对话框下，除了可以进行器件 Device 设置外，还可以进行与设计有关的各种其他功能设置，如库（Libraries）、文件（Files）、EDA 工具（EDA Tool）、编译（Compilation）、分析与综合（Analysis & Synthesis）、定时分析（TimeAnalysis）、仿真（Simulator）、配置（Fitting）等设置。Setting 对话框如图 1-12 所示。

2) 选择配置器件的工作方式。对于不同的芯片此选项卡中的内容不同。单击 Device and Pin Options 按钮，进入选择对话框，这将弹出 Device and Pin Options 对话框，如图 1-13 所示，首先选择 General 选项卡，在此工程中由于使用的是 CPLD，可以使用默认设置。对于 FPGA 用户，可以在 Options 栏内选中 Auto-restart configuration after error 复选框，使对 FPGA 的配置失败后能自动重新配置，并加入 JTAG 用户编码。注意窗口下方，将随鼠标单击的项目名而显示对应的帮助说明，用户可随时参考。

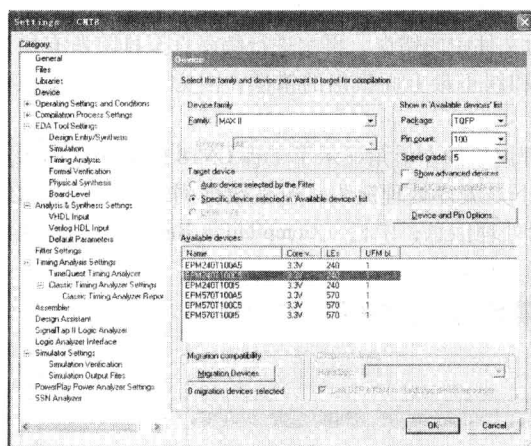


图 1-12 Settings 对话框

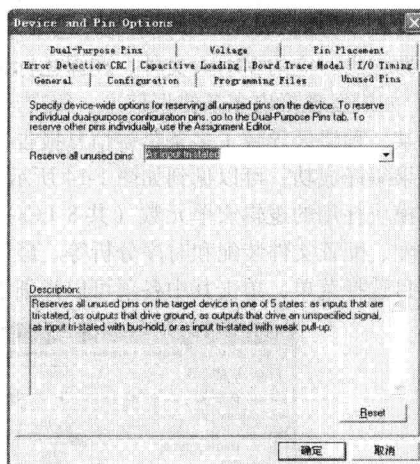


图 1-13 Device and Pin Options 对话框

3) 选择配置器件和编程方式。对于 CPLD 用户在这里不做任何设置。对于 FPGA 用户，如果希望对编程配置文件能在压缩后下载进配置器件中，可在编译前做好设置。在此，在 Configuration 选项卡中，选中 Generate compressed bitstreams 复选框，就能产生用于 EPCS 的 POF 压缩配置文件，选择配置器件为 EPCSX（视开发板上的芯片型号决定），其配置模式可选择 Active Serialo，这种方式只对专用的 Flash 技术的配置器件（专用于 Cyclone/II/III，STRATIX 等系列 FPGA 的 EPCS4、EPCS1、EPCS16 等）进行编程。

4) 选择目标器件引脚端口状态。选择图 1-13 所示对话框中的 Unused Pins 选项卡，可根据实际需要选择目标器件闲置引脚的状态，可选择为输入状态（呈高阻态，推荐此项选择），或输出状态（呈低电平），或输出不定状态，或不作任何选择。


再选择图 1-13 所示对话框中的 Dual-Purpose Pins 选择卡, 同样这里对于 CPLD 用户不做任何设置。而对于 FPGA 用户可以将 nCEO 原来的 Use as programming pin 改为 Use as regular I/O。这样可以将此端口也作为普通 I/O 口来使用。

在其他选项卡中也可作一些选择, 各选项的功能可参考对话框下方的 Description 说明。

1.3.4 全程编译

Quartus II 编译器是由一系列处理模块构成的, 这些模块负责对设计项目的检错、逻辑综合、结构综合、输出结果的编辑配置以及时序分析。在这一过程中, 将设计项目适配到 FPGA/CPLD 目标器中, 同时产生多种用途的输出文件, 如功能和时序信息文件、器件编程的目标文件等。编译器首先检查出工程设计文件中可能的错误信息, 以供设计者排除, 然后产生一个结构化的以网表文件表达的电路原理图文件。

在编译前, 设计者可以通过各种不同的设置, 指导编译器使用各种不同的综合和适配技术(如时序驱动技术等), 以便提高设计项目的工作速度, 优化器件的资源利用率。而且在编译过程中及编译完成后, 可以从编译报告窗口中获得所有相关的详细编译结果, 以利于设计者及时调整设计方案。

编译前首先设置顶层文件, 打开准备进行编译的文件, 如打开前面编辑的文件 CNT8.vhd, 执行菜单命令 Project→Set as Top-Level Entity。后面进行设计处理的各项操作就是针对这一顶层文件 CNT8.vhd 进行的。在本工程中, 由于在建工程时已经设置了, 这里可以不再进行设置。选择 Processing→Start Compilation 命令, 或单击编译器快捷方式按钮  启动全程编译。这里所谓的全程编译(Compilation)包括以上提到的 Quartus II 对设计输入的多项处理操作, 其中包括排错、数据网表文件提取、逻辑综合、适配、装配文件(仿真文件与编程配置文件)生成以及基于目标器件的工程时序分析等。

编译过程中要注意工程管理窗口下方的 Processing 栏中的编译信息。如果工程中的文件有错误, 启动编译后在下方的 Processing 处理栏中会显示出来。对于 Processing 栏显示出的语句格式错误, 可双击此条文, 即弹出对应的 VHDL 文件, 在深色标记条处即为文件中的错误, 再次进行编译直至排除所有错误(注意: 如果发现报出多条错误信息, 每次只需要检查和纠正最上面报出的错误, 因为许多情况下, 都是由于某一种错误导致了多条错误信息报告)。

如果编译成功, 可以见到如图 1-14 所示的工程管理窗口的左上角显示了工程 CNT8 的层次结构和其中结构模块耗用的逻辑宏单元数(共 8 LCs); 在此栏下是编译处理流程; 包括数据网表建立、逻辑综合、适配、配置文件装配和时序分析等。最下栏是编译处理信息; 中栏(Compilation Report 栏)是编译报告项目选择菜单, 单击其中各项可以详细了解编译与分析结果。

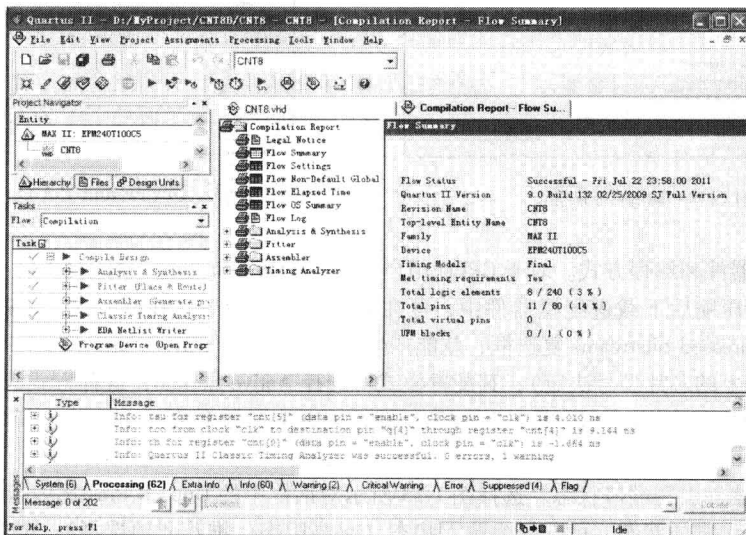


图 1-14 全程编译后的窗口

1.3.5 时序仿真

当一个设计工程的编译通过之后,能否实现预期的逻辑功能,仍需要进一步的检验,波形仿真分析是必不可少的一个环节。波形仿真就是在波形编辑器中将设计的逻辑功能用波形图的形式显示,通过查看波形图,检查设计的逻辑功能是否符合设计要求。以 VWF 文件方式的仿真流程的详细步骤如下:

1) 选择菜单 File→New..., 如图 1-10 所示, 在 New 对话框中选择 Verification/Debugging Files→Vector Waveform File。单击 OK, 出现波形编辑窗口, 显示一个空的波形文件, 如图 1-15 所示, 将文件保存为 CNT8. vwf。

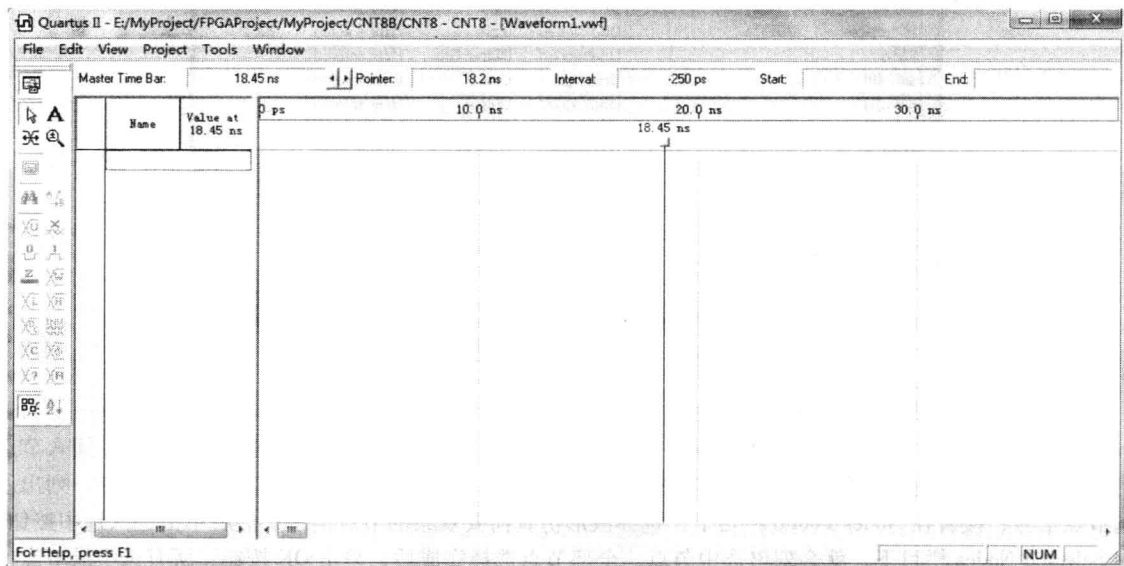


图 1-15 激励信号波形编辑器

2) 设置仿真时间区域和网格大小。对于时序仿真来说,将仿真时间轴设置在一个合理的时间区域上十分重要。通常设置的时间范围在数十微秒间。而网格的大小可以辅助信号的分析。

选择 Edit→End Time 命令, 在弹出的对话框中的 Time 文本框中输入 50, 单位选择 μs , 整个仿真域的时间即设定为 $50\mu\text{s}$, 如图 1-16 所示。单击 OK 按钮, 结束设置。选择菜单 Edit→Grid Size..., 根据需要修改网格大小, 如 20ns, 通常用网格大小表示信号状态的基本维持时间。

3) 插入仿真节点。方法一是首先选择 View→Utility Windows→Node Finder 命令, 弹出的对话框如图 1-17 所示, 在 Filter 下拉列表框中选择 Pins: all (通常已默认选择此项), 然后单击 List 按钮, 于是在下方的 Nodes Found 列表框中出现设计中的 CNT8 工程的所有端口引脚名。如果希望 Node Finder 对话框是浮动的, 可以右击此窗口边框, 在

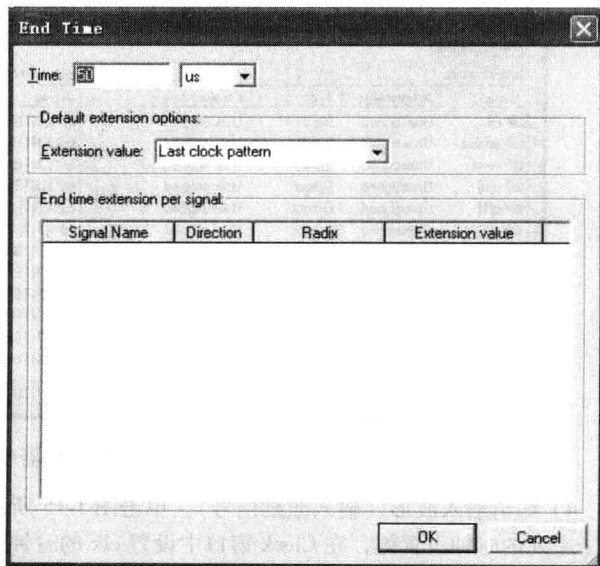


图 1-16 设置仿真时间长度