



“十三五”普通高等教育本科规划教材

FPGA/CPLD 设计与实践教程

沈莉丽 卢家凰 张志立 编

- 通过大量实例分析，由浅入深讲解FPGA设计理念
- 实例与基本理论紧密结合，提高学生的实践能力
- 教材内容全面，深入浅出，循序渐进，易学易懂



中国电力出版社
CHINA ELECTRIC POWER PRESS

“十三五”普通高等教育本科规划教材

FPGA/CPLD 设计与实践教程

沈莉丽 卢家凰 张志立 编
周 骅 主审



中国电力出版社
CHINA ELECTRIC POWER PRESS

2016年12月



内 容 提 要

本书为“十三五”普通高等教育本科规划教材。

本书是可编程逻辑器件理论课程的配套实验和实践教材,以“可编程逻辑器件及应用”课程的理论教学大纲为基础,结合现代先进的实验教学方法,精心设计了10个基础实验和5个综合实验,较全面地涵盖了可编程逻辑器件理论知识的重点和难点。本书共分4章,第1章为FPGA/CPLD概述,主要介绍了FPGA/CPLD的结构原理及开发应用选择;第2章介绍项目开发环境,主要对软件设计平台——Quartus II及硬件实验平台——KH-310进行介绍;第3章为基础实验,精选的10个实验项目内容涵盖了多个知识点;第4章为综合实验,包括数字钟、简易电子奏乐系统、数字频率计、出租车计费系统及交通灯控制系统的设计。

本书可作为高等院校自动化、电气工程及其自动化、测控技术与仪器等专业可编程逻辑器件课程的实验与实践教材,也可作为电子设计竞赛、FPGA开发应用的自学参考教材。

图书在版编目(CIP)数据

FPGA/CPLD 设计与实践教程 / 沈莉丽, 卢家凰, 张志立编. — 北京: 中国电力出版社, 2017.1

“十三五”普通高等教育本科规划教材

ISBN 978-7-5198-0146-5

I. ①F… II. ①沈… ②卢… ③张… 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 (2016) 第 308061 号

中国电力出版社出版、发行

(北京市东城区北京站西街19号 100005 <http://www.cepp.sgcc.com.cn>)

北京市同江印刷厂印刷

各地新华书店经售

*

2017年1月第一版 2017年1月北京第一次印刷

787毫米×1092毫米 16开本 11.75印张 283千字

定价 26.00 元

敬告读者

本书封底贴有防伪标签,刮开涂层可查询真伪
本书如有印装质量问题,我社发行部负责退换

版权专有 翻印必究

前言

随着信息技术的不断发展,“可编程逻辑器件及应用”已成为电子信息类学生的一门重要专业基础课程,并且在教学、科研及各类大学生电子设计竞赛中起着越来越重要的作用。为了适应现代电子技术的发展和高等院校的教学要求,我们编写了本教程。教程突出了实用性及工程应用的实际性,有助于培养学生工程实践能力、实际问题解决能力、综合应用能力及创新能力。

本书是可编程逻辑器件理论课程的配套实验和实践教材,以“可编程逻辑器件及应用”课程的理论教学大纲为基础,结合现代先进的实验教学方法,精心设计了10个基础性实验和5个综合系统设计项目,较全面地涵盖了可编程逻辑器件理论知识的重点和难点。

本书共分为4章,第1章为FPGA/CPLD概述,主要介绍了FPGA/CPLD的结构原理及开发应用选择;第2章介绍项目开发环境,主要对软件设计平台——Quartus II及硬件实验平台——KH-310进行介绍;第3章为基础实验,为了更好地与数字电路衔接,精选的10个实验项目包含了组合逻辑和时序逻辑电路中典型电路的设计,考虑到理论课程选用VHDL语言作为硬件描述语言的教学内容,每个实验都给出了完整的VHDL参考程序及实验步骤,通过实验学生能够掌握VHDL语言的一般编程方法、硬件描述语言程序设计的基本思想和方法,熟悉开发工具和相关软硬件,激发学生学习的热情,尽快进入FPGA设计实践阶段;在每个实验后还增加了相关知识及实验拓展部分,以供学生进行更为深入的研究;第4章为综合实验,包括数字钟、简易电子奏乐系统、数字频率计、出租车计费系统及交通灯控制系统的设计。每个实验项目都给出了一个设计方案,以供参考,学生可以根据设计要求自行设计其他方案。通过综合系统设计,学生能够掌握模块化程序设计思想,提高分析问题和解决问题的能力。

本书设计的10个基础实验项目和5个综合实验项目,都通过Quartus II软件进行仿真测试,并通过FPGA实验平台进行硬件验证。教师可以根据教学课时及教学实验要求等,选择相应的实验项目。

本书由沈莉丽、卢家凰、张志立共同编写,其中第1章,第2章的2.2节,第3章的3.4、3.6、3.10节,第4章的4.3节和附录A由沈莉丽编写;第2章的2.1节,第3章的3.3、3.5、3.9节和第4章的4.2、4.5节由卢家凰编写;第3章的3.1、3.2、3.7、3.8节和第4章的4.1、4.4节由张志立编写;全书由沈莉丽统稿。本书由周骅主审。

本书在编写过程中得到了许多专家和老师的的大力支持与帮助,他们对教材的编写提出了宝贵的意见,在此表示衷心的感谢。

限于编者水平,书中疏漏及不足之处在所难免,恳请读者批评指正,并请于编者联系。联系邮箱为ivy_shen@nuaa.edu.cn。

目 录

前言

第1章	FPGA/CPLD 概述	1
1.1	PLD 概述	1
1.2	FPGA/CPLD 结构原理	2
1.3	FPGA/CPLD 的开发应用选择	8
第2章	集成开发环境使用介绍	10
2.1	软件平台——Quartus II	10
2.2	实验平台——KH-310	17
第3章	基础实验	28
3.1	半加器的设计	28
3.2	全加器的设计	44
3.3	4选1数据选择器的设计	61
3.4	七段码译码器的设计	70
3.5	七人表决器的设计	78
3.6	分频器的设计	86
3.7	异步十进制计数器的设计	92
3.8	六十进制计数器的设计	100
3.9	Quartus II 调用宏功能模块输入设计方法	111
3.10	跑马灯的设计	118
第4章	综合实验	127
4.1	数字钟的设计	127
4.2	简易电子奏乐系统的设计	135
4.3	数字频率计的设计	144
4.4	出租车计费系统的设计	156
4.5	交通灯控制系统的设计	165
附录A	I/O 对照表	177
参考文献		180

第1章 FPGA/CPLD 概述

1.1 PLD 概述

1.1.1 PLD 发展历程

可编程逻辑器件 (Programmable Logic Device), 简称 PLD, 是一种由用户编程以实现某种逻辑功能的新型逻辑器件, 诞生于 20 世纪 70 年代。PLD 是大规模集成电路技术发展的产物, 是一种半定制的集成电路, 结合 EDA 技术可以灵活方便地构建数字电子系统。

很早以前, 电子工程师们就曾设想设计一种逻辑可再编程的器件, 但由于集成电路规模的限制, 难以实现。20 世纪 70 年代, 集成电路技术迅猛发展, 随着集成电路规模的增大, 才使得可编程逻辑器件得以诞生和迅速发展。

随着大规模集成电路、超大规模集成电路技术的发展, 可编程逻辑器件发展迅速, 从 20 世纪 70 年代以来, 可编程逻辑器件经历了 PROM (Programmable Read Only Memory)、PLA (Programmable Logic Array)、PAL (Programmable Array Logic)、GAL (Generic Array Logic) 等低密度 PLD 到 CPLD (Complex Programmable Logic Device) 和 FPGA (Field Programmable Gate Array) 高密度 PLD 的发展过程, PLD 集成度、速度不断提高, 功能不断增强, 结构趋于更合理, 使用变得更灵活。

可编程逻辑器件的演变过程大致如下:

- (1) 20 世纪 70 年代, 熔丝编程的 PROM 和 PLA 器件是最早的可编程逻辑器件。
- (2) 20 世纪 70 年代末, 对 PLA 进行了改进, AMD 公司推出 PAL 器件。
- (3) 20 世纪 80 年代初, Lattice 公司发明电可擦写的、比 PAL 使用更灵活的 GAL 器件。
- (4) 20 世纪 80 年代中期, Xilinx 公司提出现场可编程概念, 同时生产出世界上第一片 FPGA 器件。同一时期, Altera 公司推出 EPLD 器件, 比 GAL 器件有更高的集成度, 可以用紫外线或电擦除。
- (5) 20 世纪 80 年代末, Lattice 公司又提出在系统可编程技术, 并且推出了一系列具备在系统可编程能力的 CPLD 器件, 将可编程逻辑器件的性能和应用技术推向了一个全新的高度。
- (6) 20 世纪 90 年代后, 可编程逻辑集成电路技术进入飞速发展时期。器件的可用逻辑门数超过了百万门, 并出现了内嵌复杂功能模块的 SoPC (System on a Programmable Chip)。
- (7) 进入 21 世纪以来, FPGA 在逻辑规模、适用领域、工作速度及成本功能方面的进步变得更加瞩目。

1.1.2 PLD 分类

常见的 PLD 有 PROM、PAL、GAL、PLA、FPGA 等。目前对 PLD 的分类没有统一的标准, 一种器件往往具有多种特征, 并没有严格分类。一般可按以下几种方法进行分类。

1. 按集成度来分

- (1) 简单 PLD。一般是芯片集成度较低的, 早期出现的 PROM、PLA、PAL 及 GAL 都

属于简单 PLD，可用的逻辑门数大概在 500 门以下。

(2) 复杂 PLD。一般是集成度较高的，如现在大量使用的 CPLD、FPGA 器件等。

2. 按编程结构来分

(1) 乘积项结构 PLD。其基本结构是与—或阵列，大部分简单 PLD 和 CPLD 都属于这一类，包括 PROM、PLA、PAL、GAL、CPLD 等器件。

(2) 查找表结构 PLD。由简单的查找表组成可编程门，再构成阵列形式，多数 FPGA 属于这一类。

3. 按互连结构来分

(1) 确定型 PLD。确定型 PLD 提供的互连结构，每次用相同的互连线布线，其时间特性可以确定预知，是固定的，如 CPLD 器件。

(2) 统计型 PLD。统计型结构是指设计系统时，其时间特性是不可以预知的，每次执行相同的功能时，却有不同的布线模式，因而无法预知线路的延时，如 FPGA 器件。

4. 按编程工艺来分

(1) 熔丝型 PLD。早期的 PROM 器件就是采用熔丝结构的，编程过程是根据设计的熔丝图文件来烧断对应的熔丝，达到编程的目的。

(2) 反熔丝型 PLD。这是对熔丝技术的改进，在编程处通过击穿漏层使得两点之间获得导通，与熔丝烧断获得开路正好相反。

(3) EPROM 型 PLD。即紫外线擦除电可编程逻辑器件，是用较高的编程电压进行编程，当需要再次编程时，用紫外线进行擦除。EPROM 可多次编程。

(4) EEPROM 型 PLD。即电可擦写编程器件，与 EPROM 型 PLD 相比，不用紫外线进行擦除，可直接用电擦除，使用更加方便。GAL 器件和部分 CPLD 是 EEPROM 型 PLD。

(5) SRAM 型 PLD。即 SRAM 查找表结构的器件，目前大部分 FPGA 器件都是 SRAM 型 PLD，可方便快速的配置，但是掉电后，其内容丢失，再次上电需要重新配置，因而需要专用器件来完成这类配置操作。

(6) Flash 型 PLD。现在很多 CPLD 器件采用 Flash 工艺，采用此工艺的器件的编程次数可达万次以上，且掉电后不需要重新配置。

1.2 FPGA/CPLD 结构原理

简单 PLD 在实用中已经被淘汰，目前，PLD 的主流产品全部是以超大规模集成电路工艺制造的 CPLD 器件和 FPGA 器件。CPLD 是复杂可编程逻辑器件 (Complex Programmable Logic Device) 的简称，FPGA 是现场可编程门阵列 (Field Programmable Gate Array) 的简称，两者的功能基本相同，编程过程也基本相同，只是芯片内部的实现原理和结构略有不同。下面对两种器件的基本结构和工作原理分别进行介绍，在介绍之前，先对描述 PLD 内部结构的专用电路符号做一个简单的说明。

接入 PLD 内部的与—或阵列输入缓冲器电路一般采用互补结构，电路符号如图 1-1 所示，等效于图 1-2 所示的逻辑结构，即当信号输入 PLD 后，分别以其同相和反相信号接入。

PLD 内部的与阵列用如图 1-3 所示的简化电路符号来描述，表示可以选择 A、B、C 和 D 四个信号中的任一组或全部输入与门。在这里用于形象地表示与阵列，这是在原理上的等效。

同样,或阵列也用类似的方式表示,用如图 1-4 所示的简化电路符号来描述。阵列线连接关系用如图 1-5 所示的简化电路符号来描述,十字交叉线表示两条线未连接;交叉线的交点上打黑点,表示固定连接,即在 PLD 出厂时已连接;交叉线的交点上打叉;表示该点可编程(可改变),在 PLD 出厂后通过编程,其连接可根据需要随时改变。

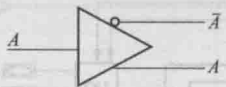


图 1-1 PLD 的互补缓冲器

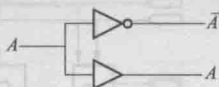


图 1-2 PLD 的互补输入

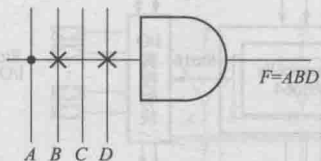


图 1-3 PLD 中与阵列表示

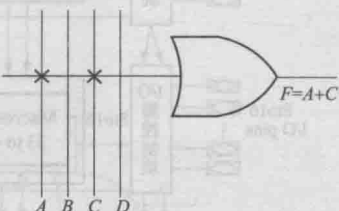


图 1-4 PLD 中或阵列表示



图 1-5 阵列线连接表示

1.2.1 基于乘积项的 PLD 结构

采用这种结构的 PLD 芯片有 Altera 的 MAX7000、MAX3000 系列 (EEPROM 工艺), Xilinx 的 XC9500 系列 (Flash 工艺) 和 Lattice、Cypress 的大部分产品 (EEPROM 工艺)。下面先看一下这种 PLD 的总体结构 (以 MAX7000 为例, 其他型号的结构与此都非常相似), 如图 1-6 所示。

这种 PLD 可分为宏单元、可编程连线 (PIA) 和 I/O 控制块。每个阵列逻辑块 LAB 由 16 个宏单元组成, 宏单元是 PLD 的基本结构, 由它来实现基本的逻辑功能。宏单元的结构如图 1-7 所示, 它是由一些与—或阵列加上触发器构成的, 其中与—或阵列用以完成组合逻辑功能, 触发器用以完成时序逻辑功能。可编程连线负责信号传递, 连接所有的宏单元、布线池、布线矩阵。CPLD 中的布线资源比 FPGA 的要简单得多, 布线资源也相对有限, 一般采用集中式布线池结构。所谓布线池, 其本质就是一个开关矩阵, 通过打结点可以完成不同宏单元的输入与输出项之间的连接。由于 CPLD 的布线池结构固定, 因此 CPLD 的输入管脚到输出管脚的标准延时固定, 被称为 pin to pin 延时, 用 T_{pd} 表示。 T_{pd} 延时反映了 CPLD 器件可以实现的最高频率, 也清晰地表明了 CPLD 器件的速度等级。I/O 控制块是 CPLD 外部封装引脚和内部逻辑间的接口。I/O 控制块负责输入、输出的电气特性控制, 如可以设定集电极开路输出、摆率控制、三态输出等。每个 I/O 单元对应一个封装引脚, 对 I/O 单元编程, 可将引脚定义

为输入、输出和双向功能。图 1-6 中的 INPUT/GCLK1、INPUT/GCLRn、INPUT/OE1、INPUT/OE2 是全局时钟、清零和输出使能信号，这几个信号由专用连线与 PLD 中每个宏单元相连，信号到每个宏单元的延时相同并且延时最短。

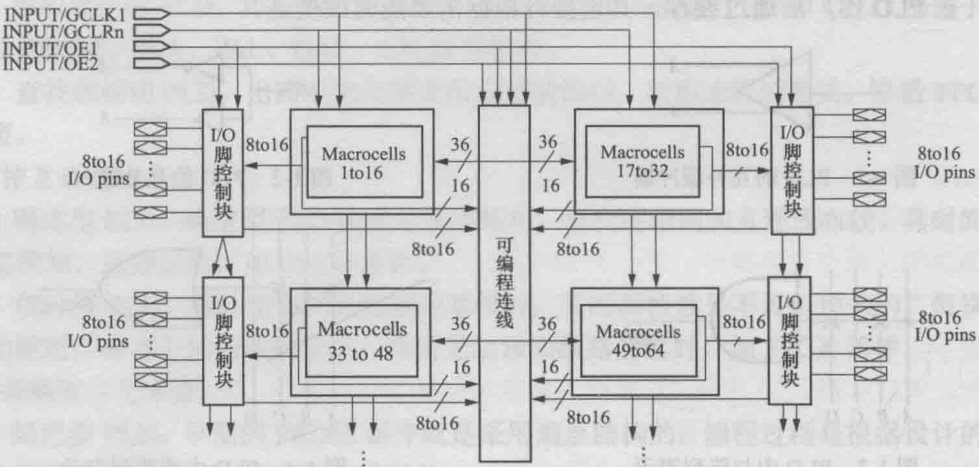


图 1-6 基于乘积项的 PLD 结构

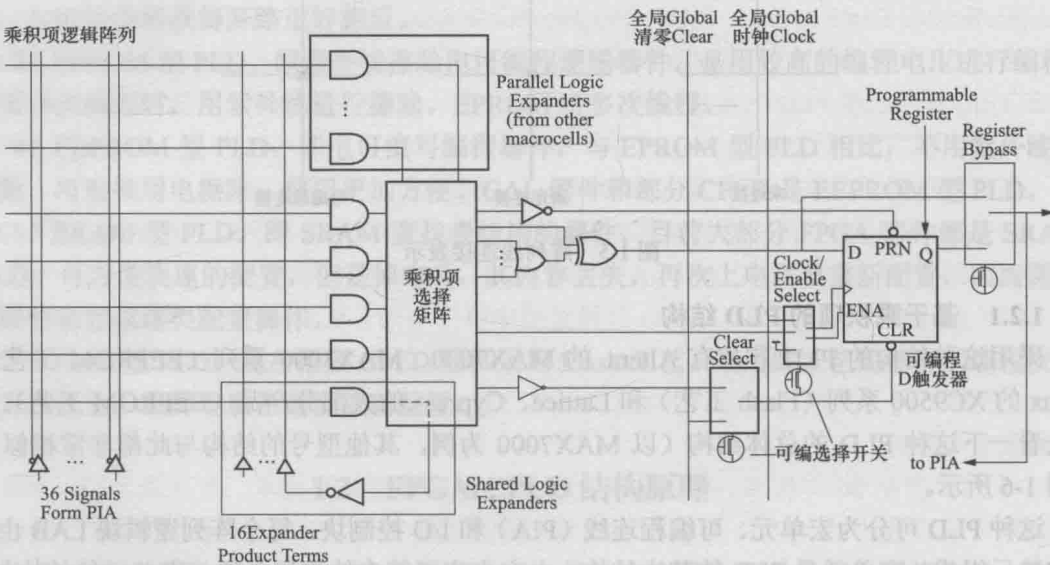


图 1-7 宏单元结构

图 1-7 中左侧是乘积项阵列，实际就是一个与—或阵列，每一个交叉点都是一个可编程熔丝，如果导通就是实现与逻辑；后面的乘积项选择矩阵是一个或阵列。两者一起完成组合逻辑。右侧是一个可编程 D 触发器，它的时钟、清零输入都可以编程选择，可以使用专用的全局清零和全局时钟，也可以使用内部逻辑（乘积项阵列）产生的时钟和清零。如果不需要触发器，也可以将此触发器旁路，信号直接输给 PIA 或输出到 I/O 脚。

1.2.2 乘积项结构 PLD 的逻辑实现原理

下面以一个简单的电路为例，具体说明 PLD 是如何利用以上结构实现逻辑的，电路如图 1-8 所示。

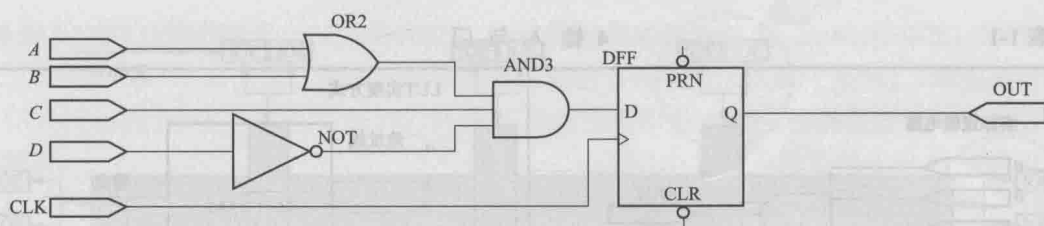


图 1-8 电路图 1

假设组合逻辑的输出 (AND3 的输出) 为 f , 则 $f = (A + B) \cdot C \cdot \bar{D} = AC\bar{D} + BC\bar{D}$ 。PLD 将以图 1-9 所示的方式来实现组合逻辑 f 。

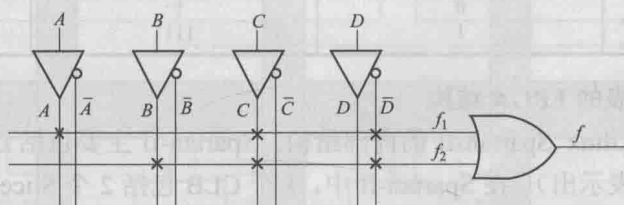


图 1-9 电路图 2

A 、 B 、 C 、 D 由 PLD 芯片的管脚输入后进入可编程连线阵列 (PIA), 在内部会产生 A 、 $\bar{A}$ 、 B 、 $\bar{B}$ 、 C 、 $\bar{C}$ 、 D 、 $\bar{D}$ 8 个输出。图 1-9 中每一个叉表示相连, 即可编程熔丝导通, 所以可得 $f = f_1 + f_2 = (A + B) \cdot C \cdot \bar{D} = AC\bar{D} + BC\bar{D}$ 。这样组合逻辑就实现了。电路中 D 触发器的实现比较简单, 直接利用宏单元中的可编程 D 触发器来实现。时钟信号 CLK 由 I/O 脚输入后进入芯片内部的全局时钟专用通道, 直接连接到可编程触发器的时钟端。可编程触发器的输出与 I/O 脚相连, 把结果输出到芯片管脚。这样 PLD 就完成了电路的功能。以上这些步骤都是由软件自动完成的, 不需要人为干预。

图 1-8 的电路是一个很简单的例子, 只需要一个宏单元就可以完成。但对于复杂的电路, 一个宏单元是不能实现的, 这时就需要通过并联扩展项和共享扩展项将多个宏单元相连, 宏单元的输出也可以连接到可编程连线阵列, 再作为另一个宏单元的输入。这样 PLD 就可以实现更复杂的逻辑功能。

这种基于乘积项的 PLD 基本都是由 EEPROM 和 Flash 工艺制造的, 一上电就可以工作, 无须其他芯片配合。

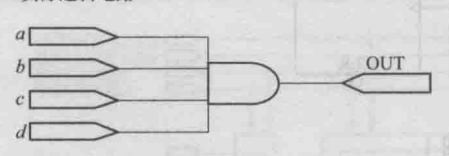
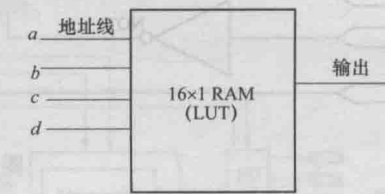
1.2.3 查找表的原理与结构

采用这种结构的 PLD 芯片称为 FPGA, 如 Altera 的 ACEX、APEX 系列、Xilinx 的 Spartan、Virtex 系列等。

查找表 (Look-Up-Table) 简称为 LUT, LUT 本质上就是一个 RAM。目前 FPGA 中多使用 4 输入的 LUT, 所以每一个 LUT 可以看成是一个有 4 位地址线的 16×1 的 RAM。当用户通过原理图或 HDL 语言描述了一个逻辑电路以后, PLD/FPGA 开发软件会自动计算逻辑电路的所有可能的结果, 并把结果事先写入 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。

表 1-1 所示是一个 4 输入与门的例子。

表 1-1 4 输入 与 门

实际逻辑电路		LUT实现方式	
			
a、b、c、d 输入	逻辑输出	地址	RAM 中存储内容
0000	0	0000	0
0001	0	0001	0
...	0	...	0
1111	1	1111	1

1.2.4 基于查找表的 FPGA 结构

图 1-10 所示为 Xilinx Spartan-II 的内部结构。Spartan-II 主要包括 CLB_s、I/O 块、RAM 块和可编程连线（未表示出）。在 Spartan-II 中，1 个 CLB 包括 2 个 Slices，每个 Slices 包括 2 个 LUT、2 个触发器和相关逻辑。Slices 可以看成是 Spartan-II 实现逻辑的最基本结构(Xilinx 其他系列，如 SpartanXL、Virtex 的结构与此稍有不同，具体请参阅数据手册)。

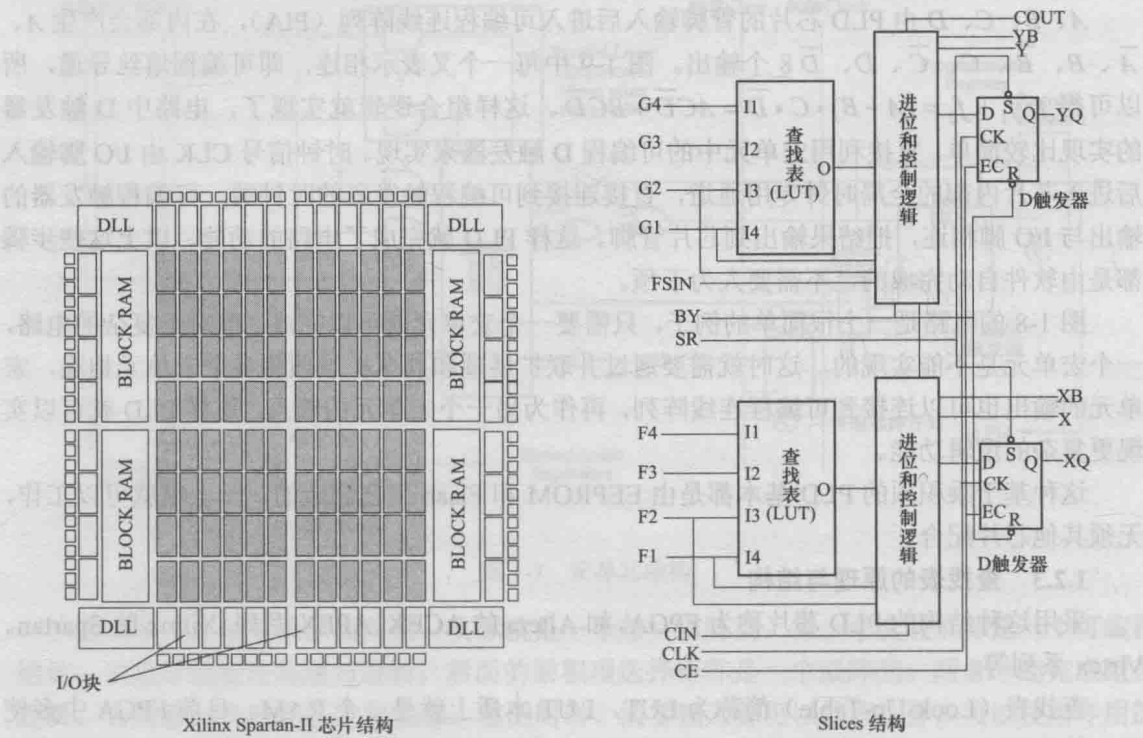


图 1-10 Xilinx Spartan-II 的内部结构

Altera 的 FLEX/ACEX 芯片的内部结构如图 1-11 所示。
逻辑单元 (LE) 的内部结构如图 1-12 所示。
FLEX/ACEX 的结构主要包括 LAB、I/O 块、RAM 块（未表示出）和可编程行/列连线。

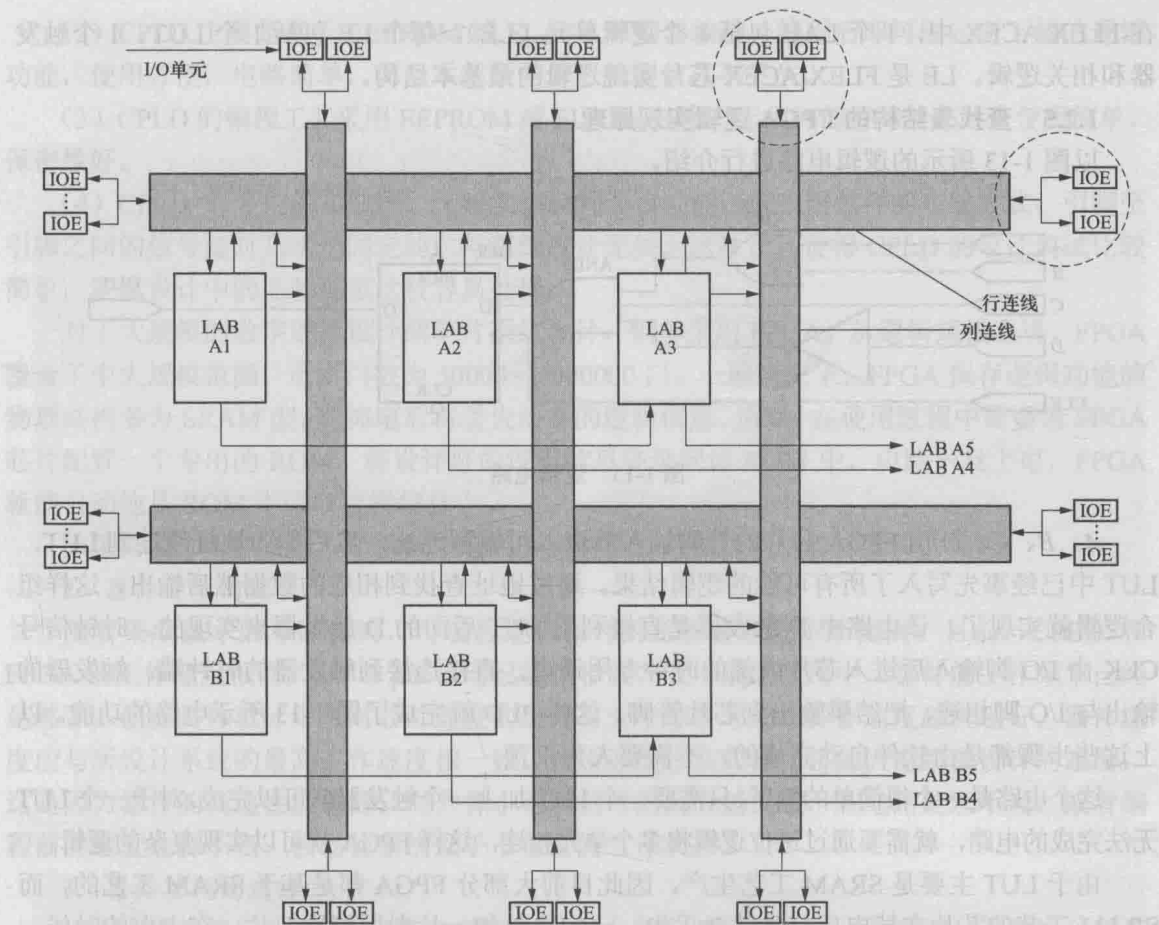


图 1-11 Altera 的 FLEX/ACEX 芯片的内部结构

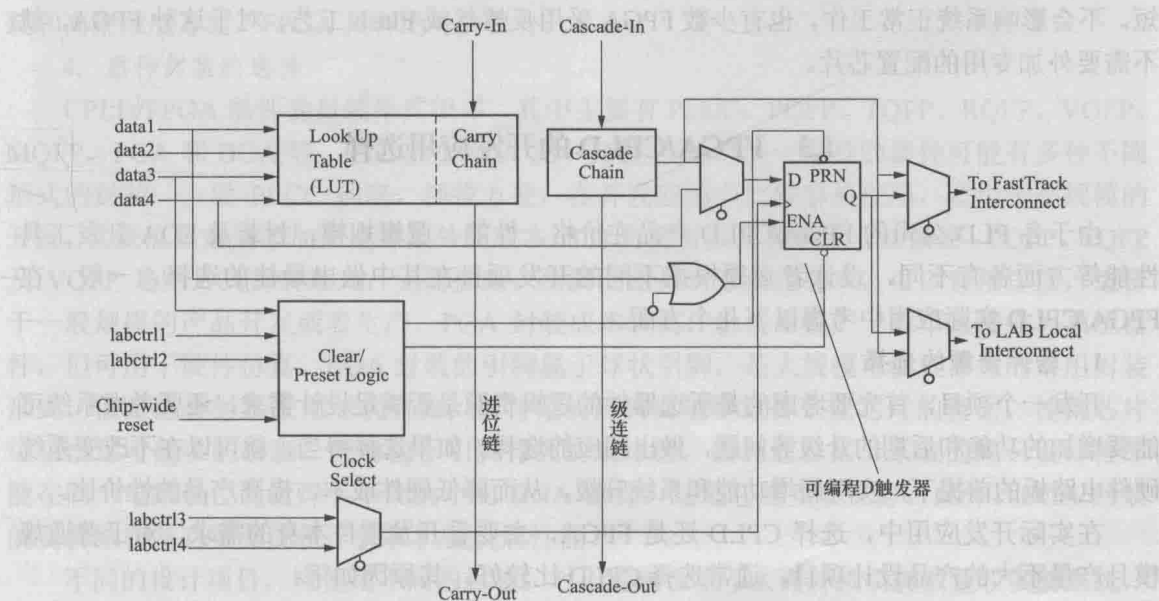


图 1-12 逻辑单元 (LE) 的内部结构

在 FLEX/ACEX 中, 1 个 LAB 包括 8 个逻辑单元 (LE), 每个 LE 包括 1 个 LUT、1 个触发器和相关逻辑。LE 是 FLEX/ACEX 芯片实现逻辑的最基本结构。

1.2.5 查找表结构的 FPGA 逻辑实现原理

以图 1-13 所示的逻辑电路进行介绍。

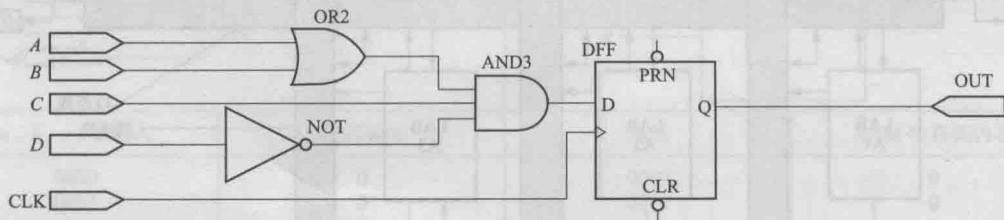


图 1-13 逻辑电路

A、B、C、D 由 FPGA 芯片的管脚输入后进入可编程连线, 然后作为地址线连到 LUT, LUT 中已经事先写入了所有可能的逻辑结果, 通过地址查找到相应的数据然后输出, 这样组合逻辑就实现了。该电路中 D 触发器是直接利用 LUT 后面的 D 触发器来实现的。时钟信号 CLK 由 I/O 脚输入后进入芯片内部的时钟专用通道, 直接连接到触发器的时钟端。触发器的输出与 I/O 脚相连, 把结果输出到芯片管脚。这样 PLD 就完成了图 1-13 所示电路的功能。以上这些步骤都是由软件自动完成的, 不需要人为干预。

这个电路是一个很简单的例子, 只需要一个 LUT 加上一个触发器就可以完成。对于一个 LUT 无法完成的电路, 就需要通过进位逻辑将多个单元相连, 这样 FPGA 就可以实现复杂的逻辑。

由于 LUT 主要是 SRAM 工艺生产, 因此目前大部分 FPGA 都是基于 SRAM 工艺的, 而 SRAM 工艺的芯片在掉电后信息就会丢失, 一定要外加一片专用配置芯片, 在上电的时候, 由这个专用配置芯片把数据加载到 FPGA 中, 然后 FPGA 就可以正常工作, 由于配置时间很短, 不会影响系统正常工作。也有少数 FPGA 采用反熔丝或 Flash 工艺, 对于这种 FPGA, 就不需要外加专用的配置芯片。

1.3 FPGA/CPLD 的开发应用选择

由于各 PLD 公司的 FPGA/CPLD 产品在价格、性能、逻辑规模、封装及 EDA 开发工具性能等方面各有不同, 设计者必须根据不同的开发项目在其中做出最佳的选择。一般, 在 FPGA/CPLD 实际应用中考虑以下几个方面。

1. 器件资源的选择

开发一个项目, 首先要考虑的是所选器件的逻辑资源是否满足设计需求, 还要考虑系统可能要增加的功能和后期的升级等问题, 做出相应的选择。如果选择得当, 就可以在不改变系统硬件电路板的前提下, 实现新增功能和系统升级, 从而降低硬件成本, 提高产品的性价比。

在实际开发应用中, 选择 CPLD 还是 FPGA, 主要看开发项目本身的需求, 对于普通规模且产量不大的产品设计项目, 通常选择 CPLD 比较好, 其原因如下:

(1) 在中小规模范围, CPLD 价格较便宜, 能直接用于系统。CPLD 器件的逻辑规模覆盖面属于中小规模 (1000~5000 门), 有很宽的选择范围, 上市速度快, 市场风险小。

(2) CPLD 主要是基于 EEPROM 或 FLASH 存储器编程, 编程后即可固定所下载的逻辑功能, 使用方便, 电路简单, 而且系统断电后, 编程信息不丢失。

(3) CPLD 的编程工艺采用 EEPROM 或 FLASH 技术, 无须外部存储器芯片, 使用简单, 保密性好。

(4) CPLD 有专门的布线区, 无论实现何种逻辑功能, 或采用怎样的布线方式, 引脚至引脚之间的信号延时几乎是固定的, 与逻辑设计无关。这种设计使得 CPLD 的设计调试比较简单, 逻辑设计中的毛刺现象比较容易处理。

对于大规模的数字逻辑设计或单片系统设计, 则多采用 FPGA。从逻辑规模上讲, FPGA 覆盖了中大规模范围, 逻辑门数为 50000~2000000 门。一般情况下, 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, 在使用过程中需要为 FPGA 芯片配置一个专用的 ROM, 将设计好的逻辑信息烧录到该 ROM 中, 电路一旦上电, FPGA 就能自动地从 ROM 中读取逻辑信息。

如果需要, 可以在同一个系统中选用不同的器件, 充分利用各种器件的优势。

2. 器件速度的选择

随着集成技术的不断提高, 可编程逻辑器件的工作速度也不断提高。目前, CPLD 和 FPGA 的工作速度很高, pin to pin 延时已达纳秒级, 在一般的应用中, 器件的工作频率已经足够了。但在具体设计中应对芯片速度的选择有一个综合考虑, 并非速度越快越好, 器件速度应与所设计系统的最高工作速度相一致。使用速度过高的器件将加大电路板设计的难度, 这是因为器件的高速性能越好, 对外界小毛刺信号的反应越灵敏, 若电路处理不当, 或者编程前的配置选择不当, 极易使系统处于不稳定的工作状态。

3. 器件功耗的选择

由于在系统编程的需要, CPLD 的工作电压大多为 5V 和 3.3V, 而 FPGA 工作电压的流行趋势是越来越低, 3.3、2.5、1.8V 等低工作电压的 FPGA 应用已十分普遍。因此, 就低功耗和高集成度方面, FPGA 具有绝对的优势。

4. 器件封装的选择

CPLD/FPGA 器件的封装形式很多, 其中主要有 PLCC、PQFP、TQFP、RQFP、VQFP、MQFP、PGA 和 BGA 等。芯片的引脚从 28~1517 不等, 同一型号的器件可能有多种不同形式的封装。一般 PLCC 插座, 插拔方便, 在开发应用中比较容易使用, 适用于小规模的开发, 其缺点是需要添加插座等额外的成本, I/O 资源有限及易被人非法解密。PQFP、TQFP 及 VQFP 是贴片式封装形式, 无需插座, 引脚间距只有零点几毫米, 可以直接焊接, 适合于一般规模的产品开发或者生产。PGA 封装成本较高, 价格昂贵, 一般不直接作为系统器件, 但可用于硬件仿真。BGA 封装的引脚属于球状引脚, 是大规模可编程器件的常用封装形式, 但是这种封装采用球状引脚, 以特定的阵列有规律地排列在芯片的背面, 使得芯片可引出尽可能多的引脚, 同时由于引脚排列的规律性, 因而适合某一系统的同一设计程序能在同一电路板位置上焊上不同大小的 BGA 器件, 这是它的重要优势。此外, BGA 封装的引脚结构具有更强的抗干扰和机械抗震性能。

不同的设计项目, 应使用不同的封装。对于逻辑含量不大而外接引脚数量较多的系统, 需要大量的 I/O 资源才能以单片形式将外围器件的工作系统协调起来, 选用贴片封装形式的器件比较好。

第2章 集成开发环境使用介绍

2.1 软件平台——Quartus II

2.1.1 Quartus II 软件简介

Quartus II 是由 Altera 公司开发的 EDA 集成开发工具,属于第四代 PLD 开发平台,是 MAX+Plus II 的升级版本。目前 Altera 已经停止了对 MAX+Plus II 的更新支持。Quartus II 是 Altera 公司针对的 FPGA/CPLD 系列器件的综合性开发软件,它的版本不断升级,从 4.0 版到 16.0 版,这里介绍的是 Quartus II 8.0 版。

1. Quartus II 的优点

该软件界面友好、使用便捷、功能强大,有一个完全集成化的可编程逻辑设计环境,是先进的 EDA 工具软件。该软件具有开放性、与结构无关、多平台、完全集成化、丰富的设计库、模块化工具等特点,支持原理图、VHDL、Verilog HDL 以及 AHDL (Altera Hardware Description Language) 等多种设计输入形式,内嵌自有的综合器以及仿真器,可以完成从设计输入到硬件配置的完整 PLD 设计流程。

Quartus II 可以在 XP、Linux 以及 Unix 上使用,除了可以使用 Tcl 脚本完成设计流程外,提供了完善的用户图形界面设计方式。它具有运行速度快、界面统一、功能集中、易学易用等特点。

2. Quartus II 对器件的支持

Quartus II 支持 Altera 公司的 MAX 3000A 系列、MAX 7000 系列、MAX 9000 系列、ACEX 1K 系列、APEX 20K 系列、APEX II 系列、FLEX 6000 系列、FLEX 10K 系列,支持 MAX7000/MAX3000 等乘积项器件;支持 MAX II CPLD 系列、Cyclone 系列、Cyclone II、Stratix II 系列、Stratix GX 系列等;支持 IP 核,包含了 LPM/Mega Function 宏功能模块库,用户可以充分利用成熟的模块,简化了设计的复杂性,加快了设计速度。此外,Quartus II 通过和 DSP Builder 工具与 Matlab/Simulink 相结合,可以方便地实现各种 DSP 应用系统;支持 Altera 的片上可编程系统(SOPC)开发,集系统级设计、嵌入式软件开发、可编程逻辑设计于一体,是一种综合性的开发平台。

3. Quartus II 对第三方 EDA 工具的支持

对第三方 EDA 工具的良好支持也使用户可以在设计流程的各个阶段使用熟悉的 EDA 工具。Altera 的 Quartus II 可编程逻辑软件属于第四代 PLD 开发平台。该平台支持一个工作组环境下的设计要求,其中包括支持基于 Internet 的协作设计。Quartus 平台与 Cadence、Exemplar Logic、Mentor Graphics、Synopsys 和 Synplicity 等 EDA 供应商的开发工具相兼容,改进了软件的 Logic Lock 模块设计功能,增添了 Fast Fit 编译选项,推进了网络编辑性能,而且提升了调试能力。

2.1.2 Quartus II 软件的安装

Quartus II 软件的安装流程分两步:安装 Quartus II 软件和注册 Quartus II 软件。有注册册

Quartus II 软件部分参考软件注册说明书，下面介绍 Quartus II 软件的安装。

(1) 双击“setup.exe”，运行安装文件，进入如图 2-1 所示的安装界面。

(2) 单击“Next”，进入如图 2-2 所示的“License Agreement”界面，选择“I accept the terms of the license agreement（我接受许可协议的条款）”。

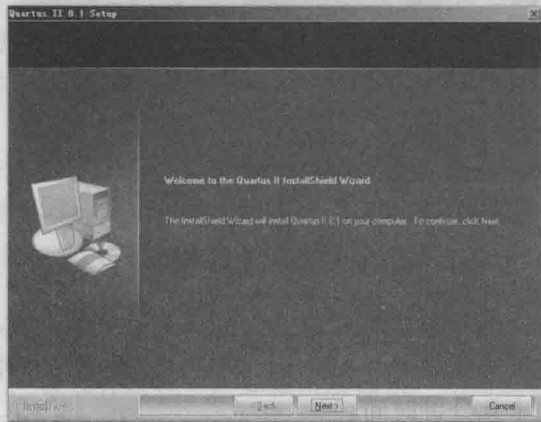


图 2-1 Quartus II 软件的安装界面

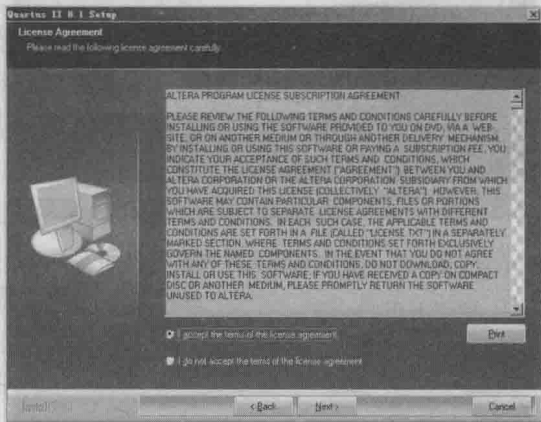


图 2-2 “License Agreement”界面

(3) 继续单击“Next”，进入如图 2-3 所示的用户信息设置界面，填写好用户名及公司名。

(4) 继续单击“Next”，进入如图 2-4 所示的软件安装位置设置界面，可以选择默认的安装位置（C:\altera\81），或者自行选择其他的安装位置。

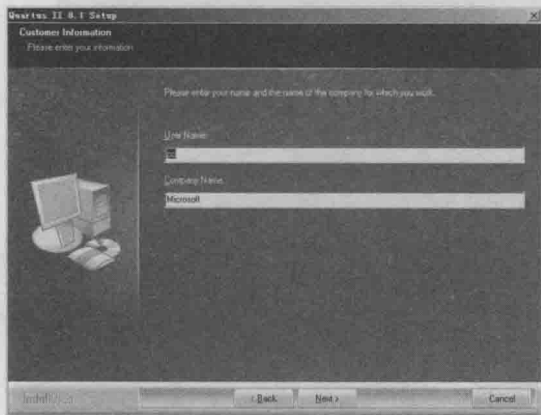


图 2-3 用户信息设置界面

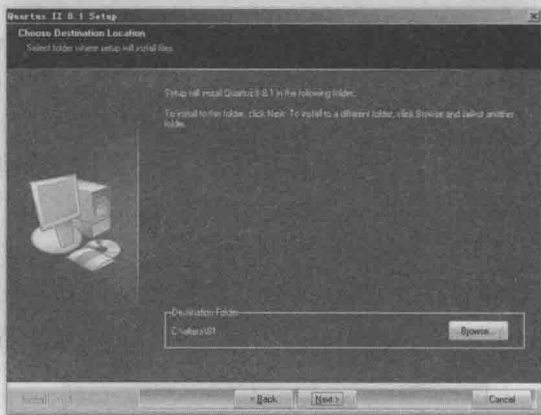


图 2-4 软件安装位置设置界面

(5) 继续单击“Next”，进入如图 2-5 所示的软件安装文件夹设置界面，可以选择默认的安装文件夹（Altera），或者自行设定新的文件夹。

(6) 继续单击“Next”，进入如图 2-6 所示的安装类型选择界面，可以选择完整（Complete）安装，即安装 Quartus II 软件所有的功能；或者选择自定义（Custom）安装，根据自己的选择，安装 Quartus II 软件的部分功能。

(7) 继续单击“Next”，进入如图 2-7 所示的安装信息汇总界面，此界面显示了之前所有的安装设置信息。

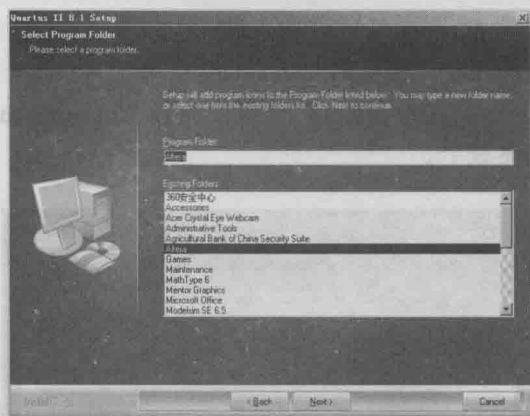


图 2-5 软件安装文件夹设置界面

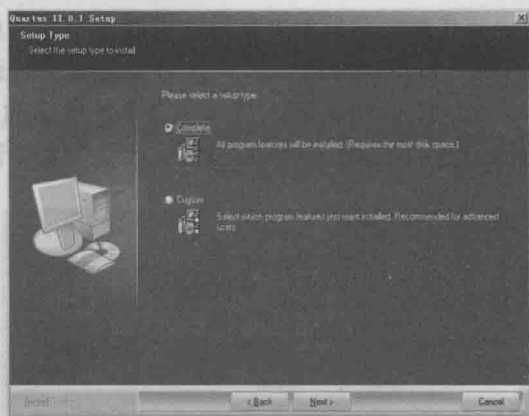


图 2-6 安装类型选择界面

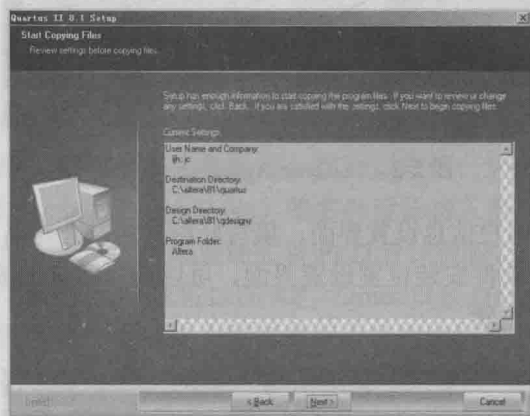


图 2-7 安装信息汇总界面

(8) 继续单击“Next”，进入自动安装阶段，安装进度如图 2-8 所示，安装过程需要几分钟到十几分钟时间。

(9) 安装完成后出现如图 2-9 所示界面，选择“是”，在桌面上将出现 Quartus II 软件的

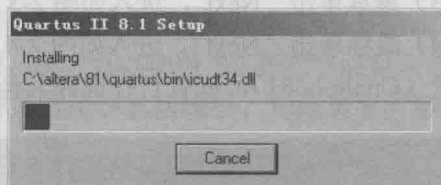


图 2-8 安装进度

快捷方式，随后出现如图 2-10 所示界面，选择“Finish”，至此，Quartus II 软件安装结束。

2.1.3 Quartus II 设计流程

和目前大多数软件开发环境（如 Visual C++）一样，Quartus II 对设计项目也采取工程

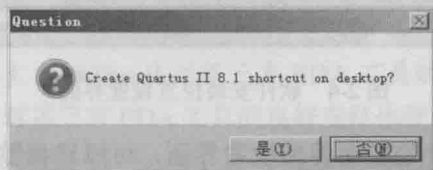


图 2-9 生成桌面快捷方式



图 2-10 安装成功界面

管理模式，即每一个设计项目对应一个工程。在一个工程下，可以包含多个设计文件，可以随时根据设计需要调整各个设计文件之间的层次关系，可以将其他设计资源加入工程，也可以将某些设计文件从本工程中移除。典型的基于 Quartus II 开发平台的设计流程如图 2-11 所示。

(1) 新建工程。新建一个工程，包含所有的设计文件，可以将其他设计资源加入工程，