

EDA技术实用丛书

VHDL与EDA技术 入门速成

本书打破常规、别出心裁地换个角度，以VHDL程序标准模板为“葫芦”，以“照葫芦画瓢”为手段来告诉读者，针对一个具体的数字电路该如何去编写VHDL程序，以期把读者迅速领进EDA技术的大门。

■ 周金富 编著 ■

书中实例的源文件可到人民邮电出版社网站下载

 **人民邮电出版社**
POSTS & TELECOM PRESS

EDA技术实用丛书

VHDL与EDA技术 入门速成

■ 周金富 编著 ■

人民邮电出版社
北京

图书在版编目 (C I P) 数据

VHDL与EDA技术入门速成 / 周金富编著. —北京: 人民邮电出版社, 2009. 6
(EDA技术实用丛书)
ISBN 978-7-115-20753-1

I. V… II. 周… III. ①电子电路—电路设计: 计算机辅助设计②硬件描述语言, VHDL—程序设计 IV. TN702 TP312

中国版本图书馆CIP数据核字 (2009) 第051165号

内 容 提 要

本书主要讲述了 VHDL (硬件描述语言) 和 EDA 设计的内容。本书打破常规, 别出心裁地换个角度, 以 VHDL 程序标准模板为“葫芦”, 以“照葫芦画瓢”为手段来告诉读者针对一个具体的数字电路该如何去编写 VHDL 程序, 以期把读者迅速领进 EDA 技术的大门。

本书形式新颖, 讲解透彻, 语言通俗易懂, 非常适合 VHDL 与 EDA 技术的初学者和自学者使用。本书对有一定经验的电子技术人员也有借鉴参考价值。本书也可以作为高校电子信息类专业的教学参考书。

EDA 技术实用丛书

VHDL 与 EDA 技术入门速成

-
- ◆ 编 著 周金富
责任编辑 姚予疆
执行编辑 刘 洋
 - ◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街 14 号
邮编 100061 电子函件 315@ptpress.com.cn
网址 <http://www.ptpress.com.cn>
北京鸿佳印刷厂印刷
 - ◆ 开本: 787×1092 1/16
印张: 10.5
字数: 254 千字
印数: 1—4 000 册
- 2009 年 6 月第 1 版
2009 年 6 月北京第 1 次印刷

ISBN 978-7-115-20753-1/TN

定价: 22.00 元

读者服务热线: (010)67129264 印装质量热线: (010)67129223
反盗版热线: (010)67171154

前 言

这是一本在内容安排和编写方法上与众不同的书，花点时间来读一读，一定不会让广大读者失望！

学习 EDA 技术已经在国内电子界成为一种热潮，但如果我们用冷静的眼光去透视这股学习 EDA 技术的热情，就会发现其中存在两个问题。

问题之一：授人鱼者多，授人渔者少。

浏览目前已出版的 EDA 书籍，我们就会发现大多数 EDA 书籍几乎都保持着同一副面孔——千篇一律地将 VHDL 当做一门语言课来教，总是让读者按部就班地去学习程序结构、语言要素、顺序语句、并行语句、子程序、库和程序包等基础知识，并且无一例外地都列出一些别人编写好的程序，而对于这些具体的数字电路，它们的程序到底是如何编写出来的，或者说到底该怎样编写程序，却从未见到过有一本书详细讲解过。因此，很多时候，一本书看到最后，除了得到一些已编写好的程序外，读者对如何编写程序仍是一脸茫然，致使读者始终跨不进 EDA 技术的大门。

问题之二：移植程序者多，自编程序者少。

作者曾与多位学过 EDA 专业课程且已进入企业从事 EDA 工作的大学生做过对话，“企业给你一个电路设计课题，你自己会编程吗？”“不会。”“那你怎么完成工作任务呢？”“到新华书店抄一些现成的程序回去就行了。”“假设这个电路设计课题是个新的电路，又找不到可供移植的程序实例，你怎么办呢？”“向教我的老师求援。”“……”作者无语。

这两个问题说明了什么呢？

大家知道，学习 EDA 技术主要应掌握硬件描述语言、软件转换工具、实验开发系统和可重构器件这 4 方面的内容，而如何用硬件描述语言描述数字电路的功能则是学习的重中之重，因此，会不会编写 VHDL 程序直接决定着一个人能否跨进 EDA 技术的大门。既然当前的 EDA 书没有教会读者如何编程，那么怎能不出现“抄袭”一族呢？

其实，VHDL 是一种非常优秀的通用的工业标准语言，用它描述出来的程序，格式是固定的，语句是规范的，也就是说，编写 VHDL 程序是有一个标准模式的，并且这个标准模式已经被作为一个标准模板放进了 EDA 软件中。既然如此，能否让 EDA 书籍换一副面孔？能否让读者换一种学习方法呢？——作者苦苦思索。

于是乎，作者打破常规，剑走偏锋、别出心裁地换个角度，尝试着以 VHDL 程序标准模板为“葫芦”、以“照葫芦画瓢”为手段来告诉读者针对一个具体的数字电路该如何去编写 VHDL 程序，以期把读者迅速领进 EDA 技术的大门。

本书有哪些地方与众不同呢？

第一，侧重点不同。作者既没有把 VHDL 当成一门程序语言课，也没有把 VHDL 语法作为学习重点，而是把不同程序中该使用哪种语句、不同语句中该如何填写标识符作为重点，换句话说，就是把针对一个具体的数字电路应该怎样去编写程序作为重中之重。

第二，着眼点不同。作者没有把读者对象完全定位在有教师释疑解惑的大学生群体上，

而是兼顾到自学的初学者群体；没有把别人编写好的程序作为范例供读者去模仿或移植，而是对这些程序进行逐句剖析，告诉读者程序为什么要如此编写。

第三，领路方法不同。作者对于实践课题不是简单地列出 VHDL 程序，然后让读者去体会，而是先以两个课题进行示范，手把手教初学者如何一步一步地编写出程序，然后给出两个课题供初学者自己练习，最后才允许初学者与附录中的答案进行比对。

编写本书的指导思想定位于“入门”和“速成”4 字：不求一网打尽天下鱼，也不求一口吃成个大胖子，但求简单易懂、即学即用、顺利入门、短期速成，因此本书仅有短短的 5 章。希望初学者在学习本书时，把重点放在第 2 章上，因为会编写 VHDL 程序是进入 EDA 技术大门的关键性一步！

作者在编写本书的过程中，得到了许多专家学者的帮助，许多思想都是受到了专家学者们观点的启发，在此特由衷地向他们表示深深的敬意和感谢！

参与本书编写工作的还有周秀明、薛仕凤、季如林、许俊、征亚兰、陈宏景、万海峰、周祥磊、周秀珍、沈国荣。在本书的编写出版过程中，作者得到了人民邮电出版社陈万寿编审和刘洋编辑给予的悉心指导和全力支持，在此特别致以衷心的感谢！

另外，由于本书采取了打破常规的写作方法，书中错误在所难免，恳望专家学者及广大读者给予批评指正。大家可以通过电子邮件（liuyang@ptpress.com.cn）与我们交流。

作者 周金富

目 录

第 1 章 可重构器件入门..... 1	
1.1 可重构器件的基本构成 和功能 2	
1.1.1 可重构器件的分类 2	
1.1.2 PLD 的基本构成和功能 2	
1.1.3 PAD 的基本构成和功能 2	
1.2 可重构器件的结构和 重构原理 3	
1.2.1 PLD 的结构 3	
1.2.2 PAD 的结构 5	
1.2.3 PLD 的重构原理 5	
1.2.4 PAD 的重构原理 10	
1.3 可重构器件的比较和选用 10	
第 2 章 硬件描述语言入门..... 13	
2.1 VHDL 编程入门..... 13	
2.1.1 VHDL 程序设计模板 14	
2.1.2 程序语句运用规则 17	
2.1.3 语句标识符填写方法 41	
2.2 VHDL 编程示范..... 55	
2.2.1 1 位全加器的分析 55	
2.2.2 1 位全加器的设计 56	
2.2.3 电子烟花的分析 59	
2.2.4 电子烟花的设计 61	
第 3 章 VHDL 编程的实践与提高 70	
3.1 VHDL 编程实践..... 70	
3.1.1 实践课题 70	
3.1.2 实践课题答案 73	
3.2 VHDL 编程进阶..... 73	
3.2.1 关于设计模块、实体 和结构体 73	
3.2.2 关于库和程序包 74	
3.2.3 关于程序语句中的 保留字 76	
3.2.4 关于程序语句中的 表达式 77	
3.2.5 关于程序中的 并行语句 78	
3.2.6 关于程序中的顺序语句 82	
3.2.7 关于程序包的自行 设计方法 84	
3.2.8 关于子程序及 子程序调用 88	
3.2.9 关于库元件和参数化 模块的调用 96	
3.2.10 关于状态机的设计 98	
3.2.11 关于寄存器的 7 种 设计 104	
3.2.12 关于计数器的 14 种 设计 107	
3.3 VHDL 程序的优化 113	
3.3.1 一切都应面向综合 114	
3.3.2 程序都应设法优化 114	
3.3.3 表述注意避免混乱 120	
3.3.4 尽量遵守业界的 习惯规定 120	
第 4 章 软件转换工具入门..... 122	
4.1 MAX+PLUS II 软件功能简介 .. 122	
4.1.1 软件基本功能 122	
4.1.2 主要菜单功能 123	
4.1.3 工具按钮功能 124	
4.1.4 输入方法简介 125	
4.2 MAX+PLUS II 软件操作方法 .. 126	
4.2.1 项目的建立 127	
4.2.2 项目的编译 128	
4.2.3 项目的校验 129	
4.3 PAC_Designer 软件操作方法 ... 131	
4.3.1 常规使用 131	

4.3.2	非常规增益设置	133	5.2.2	已重构芯片的 功能验证	148
4.3.3	宏库的宏调用	135	5.3	自制 EDA 硬件实验板	149
第 5 章	实验开发系统入门	136	5.3.1	实验板的基本 硬件配置	149
5.1	实验开发系统的 作用与原理	136	5.3.2	基本硬件电路原理图	150
5.1.1	GW48 实验开发 系统简介	136	附录	实践课题答案	155
5.1.2	实验开发系统的作用	145	4—10 线译码器 VHDL 程序		155
5.1.3	实验开发系统的原理	145	4 位加法器 VHDL 程序		157
5.2	实验开发系统的操作方法	148	参考文献		160
5.2.1	逻辑重构文件的下载	148			

第 1 章 可重构器件入门

EDA 技术是电子设计自动化技术的简称，是现代电子信息工程领域的一门新技术。它把电子设计技术与电路制造技术有机地融合在了一起，目前它不仅在向 ESDA（电子系统设计自动化）这个深度方向发展，而且还在向超越电子设计范畴进入其他领域这个广度方向发展。因此，EDA 技术现在已成了电子产品研制开发过程中不可或缺的一门重要技术。

EDA 技术所包含的内容十分丰富，涉及面也非常广泛，但就目前情况来看，应用最为广泛的主要是基于可重构器件的 EDA 技术。所以，现代电子工程技术人员在学习 EDA 技术时，应把重点放在关注基于可重构器件的 EDA 技术上，并把掌握基于可重构器件的 EDA 技术作为自己的一项必备技能。

那么，什么是基于可重构器件的 EDA 技术呢？

所谓基于可重构器件的 EDA 技术，就是在 PC 上使用 EDA 软件把自行设计的电子电路制作到购买来的大规模可重构器件（PLD 和 PAD）内部，研制出专用集成电路（ASIC）的技术。很显然，基于可重构器件的 EDA 技术，可以使电子工程技术人员在拥有一台计算机、一套 EDA 软件和一片（或几片）大规模可重构器件的条件下，就能够自行设计制造出具有自主知识产权的属于设计者自己的专用集成电路。

在这种设计者自己的专用集成电路内部，可能做成的是一个单元电路，也可能做成的是由若干个单元电路组合起来的复杂电路，它们已经能够完成一个电子产品中电路所具有的控制处理功能，因此，本书约定：把设计者自己的专用集成电路的内部电路，统称为一个电路模块（注意：这只是一个设计者自己制作的电路模块，仅仅相当于市售的一块普通的集成电路，这个电路模块配上外围电路才能构成电路系统）。

下面我们看看设计开发这种电路模块的工作流程是怎样的。

在基于可重构器件的 EDA 技术中，设计开发电路模块的工作流程是：首先使用硬件描述语言或图形来对电路模块的功能、行为、结构和接口进行描述，得到程序设计文件（这一步称为编程），再利用软件转换工具将程序设计文件编译转换成逻辑重构文件（这一步称为编译），最后通过实验开发系统将逻辑重构文件下载到大规模可重构器件中并进行功能验证（这一步称为下载）。

从这个“编程→编译→下载”的工作流程中可以看出：硬件描述语言是进行电路模块设计的一种主要表达手段，软件转换工具是进行电路模块设计的一种自动化转换工具，实验开发系统是进行电路模块设计的下载及验证工具，而大规模可重构器件则是电路模块设计的目标器件。因此，学习基于可重构器件的 EDA 技术，主要应掌握硬件描述语言、软件转换工具、实验开发系统和大规模可重构器件这 4 方面的内容。

下面我们就从认识可重构器件入手，开始我们的基于可重构器件的 EDA 技术的自学课程。

可重构器件，在当前的 EDA 书籍中都被称为可编程器件，但一些专家和学者已经提出，可编程器件改称为可重构器件更为合适，作者也认同这种观点，所以本书把通常所说的可编程器件改称为可重构器件。

可重构器件是这样进行定义的：功能可以重新构造的半导体器件，称为可重构器件。

1.1 可重构器件的基本构成和功能

1.1.1 可重构器件的分类

可重构器件可分为逻辑功能可重新构造的可重构逻辑器件（PLD）和电路功能可重新构造的可重构模拟器件（PAD）两大类，其中 PLD 又可分为乘积项结构的 CPLD 和查找表结构的 FPGA 两类。

1.1.2 PLD 的基本构成和功能

PLD 主要由可重构通用逻辑块、可重构 I/O 控制块和可重构内部连线三部分构成。

可重构通用逻辑块是可重构器件的主要部分，它的作用是用来实现最基本的逻辑功能，由于通用逻辑块是由与/或阵列加寄存器或者是由查找表加寄存器组成的，故通过重构，既能实现组合逻辑功能，又能实现时序逻辑功能。

可重构内部连线的作用是提供通用逻辑块与通用逻辑块之间、通用逻辑块与 I/O 控制块之间的信息传递通道，通过重构，把只具有简单逻辑功能的通用逻辑块级联成具有复杂逻辑功能的完整数字电路。

可重构 I/O 控制块是一种受控三态缓冲器，它的作用是提供器件内部电路与器件外部引脚之间的信息传递通道，通过重构，可以把这些信息通道配置成输入、输出或双向工作模式。

1.1.3 PAD 的基本构成和功能

PAD 主要由参数可重构基本电路单元、可重构布线池和一些辅助电路构成。

参数可重构基本电路单元是一种由两个仪用放大器和一个输出放大器配以电阻电容构成的差分输入差分输出的放大电路，通过参数重构可以使电路的增益、带宽和阈值满足多种不同的电路设计要求。

可重构布线池的作用是提供放大电路与放大电路之间、放大电路与器件引脚之间的信息传递通道，通过重构，可以把放大电路级联成具有放大、运算、滤波或 D/A 转换等功能的各种模拟电路。

辅助电路的作用是提供一些参考电压、自动校正单元等。

1.2 可重构器件的结构和重构原理

1.2.1 PLD 的结构

CPLD 和 FPGA 在基本构成上是完全相同的，都是由可重构通用逻辑块、可重构 I/O 控制块和可重构内部连线三部分构成的，它们的主要区别就是在结构上的排列不同和通用逻辑块的构成不同。

1. CPLD 的结构

CPLD 的结构排列示意图如图 1.1 所示，内部连线位于芯片的中心部位形成一个矩阵，通用逻辑块则围绕着内部连线矩阵排列，I/O 控制块排列在芯片的四周。

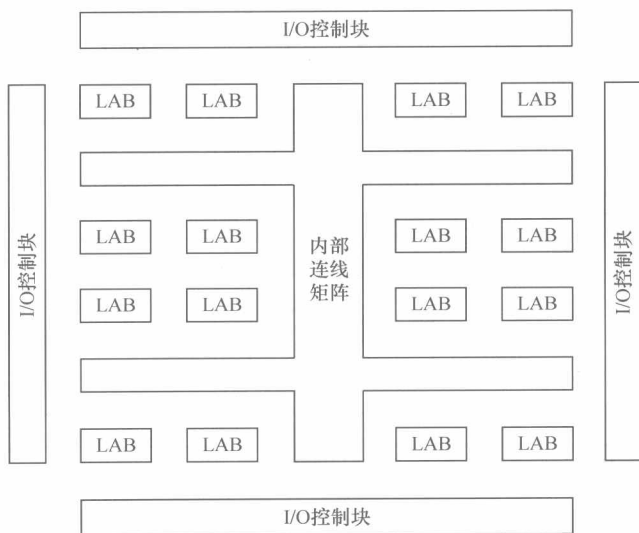


图 1.1 CPLD 结构排列示意图

CPLD 的通用逻辑块是一种被称为 LAB 的逻辑阵列块，LAB 是一种大单元，实际是由宏单元组成的阵列，每个 LAB 拥有 16 个宏单元。宏单元的结构如图 1.2 所示，主要由逻辑阵列、乘积项选择矩阵和可重构寄存器构成。逻辑阵列和乘积项选择矩阵实际是一种与阵列可重构/或阵列固定的与或阵列，通过重构可实现各种各样的组合逻辑；可重构寄存器通过重构可配置成带时钟端的 D、T、JK、SR 4 种触发器，它们和逻辑阵列、乘积项选择矩阵配合，则可实现各种各样的时序逻辑（实现组合逻辑时寄存器则被配置成旁路）。

2. FPGA 的结构

FPGA 的结构排列示意图如图 1.3 所示，通用逻辑块均匀分布于芯片中排列成矩阵，内部连线则穿行于通用逻辑块的行列之间，I/O 控制块同样排列在芯片的四周。

FPGA 的通用逻辑块是一种被称为 CLB 的可配置逻辑块，CLB 是一种小单元。可配置逻辑块 CLB 的结构如图 1.4 所示，主要由查找表式逻辑函数发生器和可重构寄存器构成。逻辑

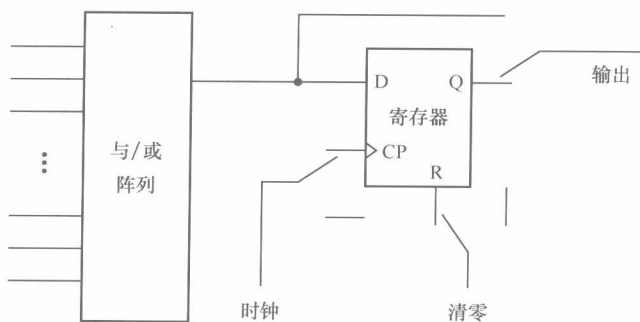


图 1.2 CPLD 中宏单元的结构框图

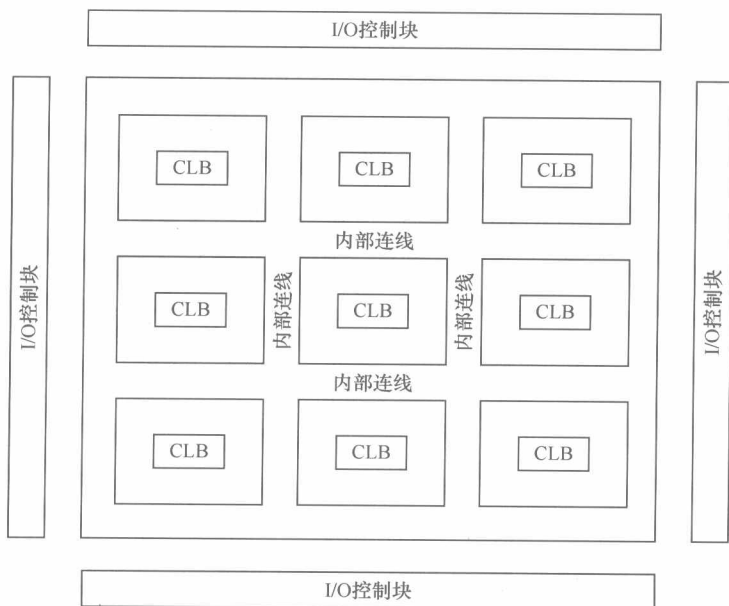


图 1.3 FPGA 结构排列示意图

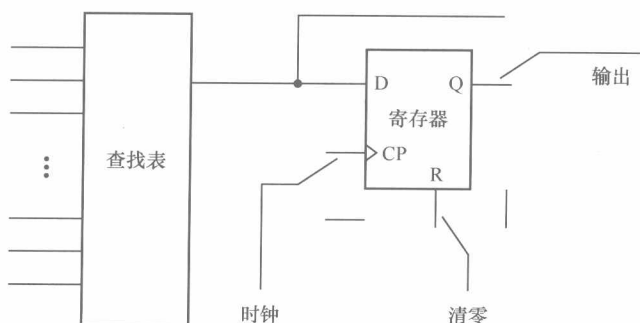


图 1.4 FPGA 中逻辑块的结构框图

函数发生器实际是一种静态随机存取存储器 (SRAM)，存放着各种组合逻辑函数，通过重构可实现各种各样的组合逻辑；可重构寄存器通过重构可配置成带时钟端的 D、T、JK、SR 4

种触发器，它们和逻辑函数发生器配合，则可实现各种各样的时序逻辑（实现组合逻辑时寄存器则被配置成旁路）。

1.2.2 PAD 的结构

PAD 器件中，各品种的基本结构比较类似，较具代表性的有：ispPAC10、ispPAC20 和 ispPAC80，其结构图分别如图 1.5、图 1.6 和图 1.7 所示。



图 1.5 ispPAC10 结构框图



图 1.6 ispPAC20 结构框图



图 1.7 ispPAC80 结构框图

1.2.3 PLD 的重构原理

本书所说的重构，其实就是对可重构器件内部的通用逻辑块、I/O 控制块、内部连线进行重新连接。后面将谈到，逻辑重构文件，实际上是由地址数据和电平数据组合起来的一长串数据，其中的电平数据用来“接通或断开”由 E²PROM 或 SRAM 构成的“开关”，地址数据则用来“指定”接通或断开的开关的“位置”。这样在逻辑重构文件传送到（即下载到）可重构器件的过程中，可重构器件中通用逻辑块、I/O 控制块、内部连线三部分的“开关”便按电路功能被一一接通或断开，于是，可重构器件内部便被重新连接成了一个具备某种功能的电路，这块可重构器件也就被重新构造造成了一块新的 ASIC，这就是本书所说的重构。

1. CPLD 的重构原理

我们知道，任何一种组合逻辑都可以用“与—或”表达式来描述，例如图 1.8 所示的 1 位半加器的逻辑表达式可表示为 $so = (a+b) \overline{ab}$ 和 $co = ab$ ，经化简可得最简“与—或”表达式为 $so = a\overline{b} + \overline{a}b$ 和 $co = ab$ 。当软件处理到 $so \leq (a \text{ OR } b) \text{ AND } (a \text{ NAND } b)$ 和 $co \leq \text{NOT } (a \text{ NAND } b)$ 这两条 VHDL 语句时，软件会自动地判断（识别）出这是一个半加器，于是把半加器的最简表达式转换成一组逻辑重构数据，然后下载到 CPLD 中，在与/或阵列中和寄存器上进行相关连接，如图 1.9 所示（注意与/或阵列后边的寄存器是旁路掉的），这样就在 CPLD

中做出了一个半加器，这是一个组合电路，如果要做的是时序电路，则软件会自动地把与/或阵列后边的寄存器配置成相应的触发器。

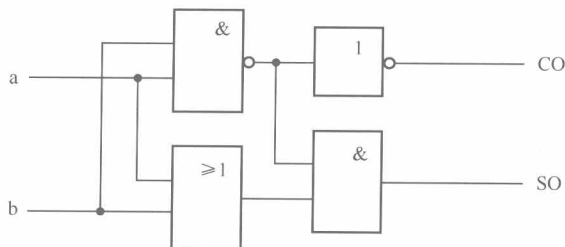


图 1.8 1 位半加器原理图

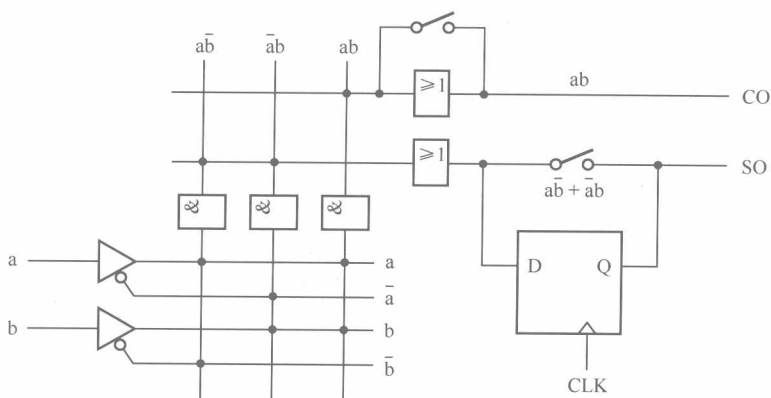


图 1.9 CPLD 编程为半加器的示意图

2. FPGA 的重构原理

我们知道，各种组合电路的输入变量可能是互不相同的，输出变量也可能是互不相同的，但每一个组合电路的输入变量与输出变量之间的因果关系却是固定不变的，这种因果关系如果用真值表表达出来，那可是最清楚不过的——各种组合电路的真值表是互不相同的，每种组合电路的真值表是唯一的。FPGA 正是基于这一点来实现组合逻辑的。FPGA 中的逻辑函数发生器实际是一个 SRAM，预先存放着各种组合逻辑电路的真值表，这样当软件处理到例如 $so \leq (a \text{ OR } b) \text{ AND } (a \text{ NAND } b)$ 和 $co \leq \text{NOT } (a \text{ NAND } b)$ 这两条 VHDL 语句时，软件会自动地判断（识别）出这是一个半加器，于是便产生一组逻辑重构数据，然后下载到 FPGA 中，在 SRAM 中查找半加器的真值表，并把真值表指定的输出变量通过连线开关进行相关连接（注意寄存器是旁路掉的），这样就在 FPGA 中做出了一个半加器，这是一个组合电路，如果要做的是时序电路，则软件会自动将查找表后边的寄存器配置成相应的触发器。

3. PLD 的重构过程

图 1.10 所示为 PAL12L6 的内部逻辑图，图中，标有“×”号处表示该处的横竖线之间由熔丝连接着（熔丝可以保留或熔断，即可以重新构造），标有“·”号处表示该处的横竖线之间由导线连接着（导线是固定连接，即不可以重新构造）。

现在要用 PAL12L6 实现图 1.11 所示的逻辑功能，其重构过程如下。

首先，工程设计人员用 ABEL 语言把图 1.11 所示的电路描述成如下的 ABEL 程序文件。

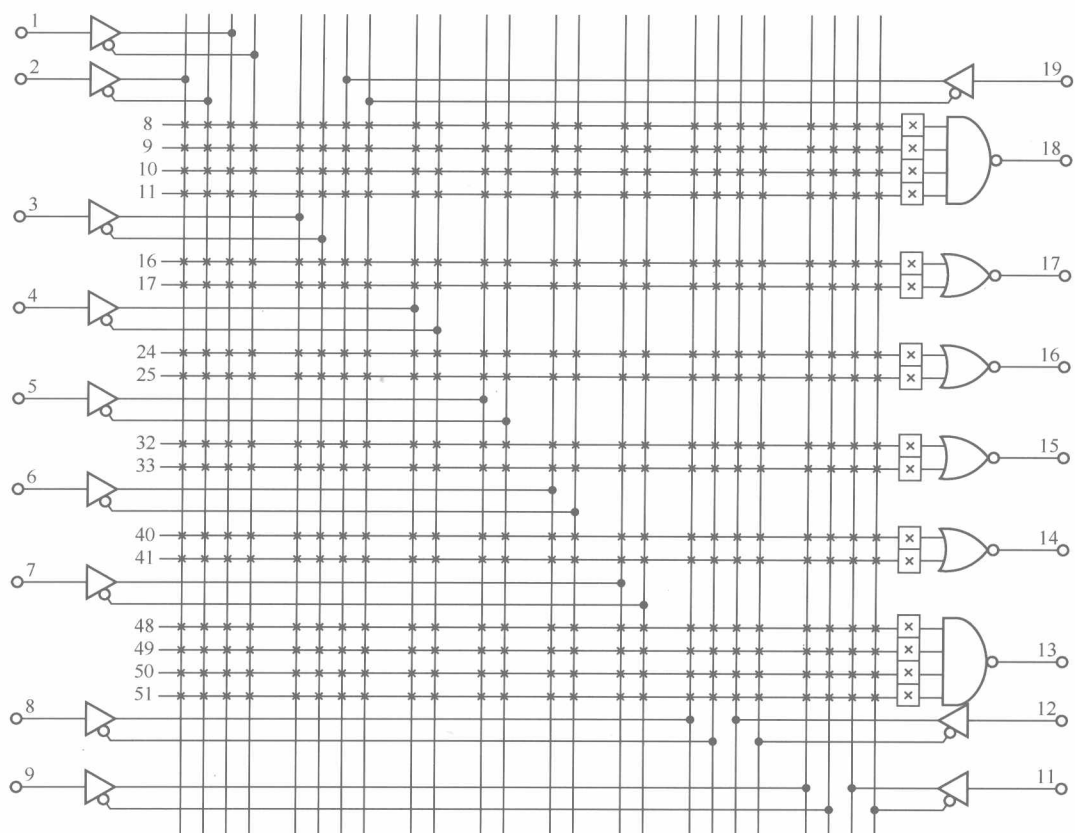


图 1.10 PAL12L6 内部逻辑图

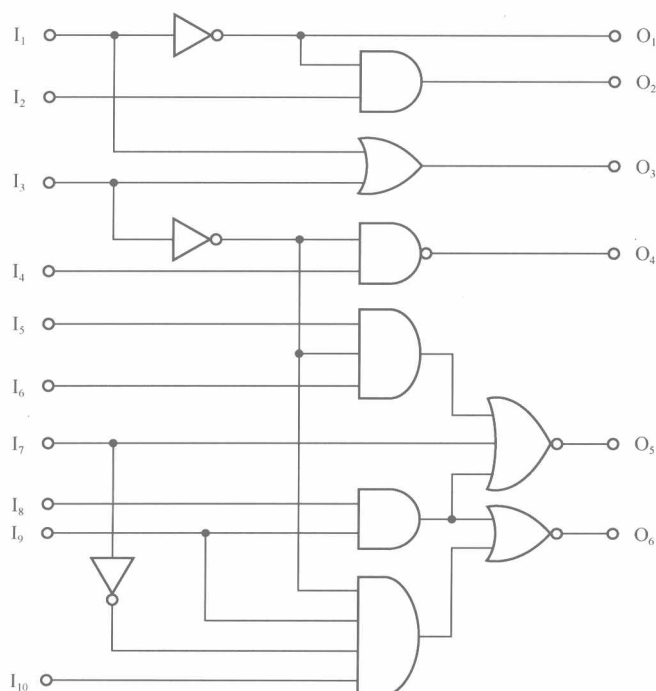


图 1.11 重构实验用的逻辑电路图

```

MODULE m002
TITLE logic gates
    designer
    date 3.9.1994'
VO2 DEVICE 'P12L6'
I1, I2, I3, I4, I5, I6, I7, I8, I9, I10  PIN  1, 2, 3, 4, 5, 6, 7, 8, 9, 11;
O1, O2, O3, O4, O6, O5  PIN  18, 17, 16, 15, 14, 13;
EQUATIONS
!O1=I1;           --!表示非 即 “-”
!O2=I1#I2;        --#表示或 即 “+”
!O3=!I1&!I3;       --&表示与 即 “.”
!O4=!I3&I4;
!O5=!I3&I5&I6#I7#I8&I9;
!O6=I8&I9#!I3&!I7&I9&I10;
END m002

```

然后，利用编译软件把程序文件转换为如下的熔丝图文件。

```

8  --x-  ----  --  --  --  --  ----  ----
16 --x-  ----  --  --  --  --  ----  ----
17 -x--  ----  --  --  --  --  ----  ----
24 ---x  -x--  --  --  --  --  ----  ----
32 ----  -x--  x-  --  --  --  ----  ----
40 ----  ----  --  --  --  --  x---  x---
41 ----  -x--  --  --  --  --  -x  ---  xx--
48 ----  -x--  --  x-  x-  --  ----  ----
49 ----  ----  --  --  --  x-  ----  ----
50 ----  ----  --  --  --  --  x---  x---

```

熔丝图文件中，“8、16、17、…”表示 PAL12L6 中横线的编号（即行号），“-”表示熔丝熔断，“×”表示熔丝连接状态保留。注意：这里只列出其中的一部分文件，其余的熔丝全熔断的行号在此文件中被省略掉而未画出。

与此同时，编译软件又把熔丝图文件编译成如下的 JEDEC 格式文件。

```

*L08 1101 1111 11 11 11 11 1111 1111
*L16 1101 1111 11 11 11 11 1111 1111
*L17 1011 1111 11 11 11 11 1111 1111
*L24 1110 1011 11 11 11 11 1111 1111
*L32 1111 1011 01 11 11 11 1111 1111
*L40 1111 1111 11 11 11 11 0111 0111
*L41 1111 1011 11 11 11 10 1111 0011
*L48 1111 1011 11 01 01 11 1111 1111
*L49 1111 1111 11 11 11 01 1111 1111
*L50 1111 1111 11 11 11 11 0111 0111

```

--0 电平则熔丝连接状态保留
--1 电平则熔丝熔断

JEDEC 格式文件就是要下载到 PAL12L6 器件中的逻辑重构文件。在 JEDEC 格式文件中，“*L08、*L16、*L17、…”称为地址数据，“0”和“1”称为电平数据。

最后，把逻辑重构文件下载到 PAL12L6 器件中。在下载过程中，逻辑重构文件中的地址

数据“引导”电平数据到达“指定位置”，电平数据则“保留或熔断”指定位置上作为“开关”使用的熔丝，这样就对 PAL12L6 器件的功能进行了重新构造，使 PAL12L6 成为了一块具备图 1.11 所示电路功能的数字电路模块了。

重构后的 PAL12L6 的内部逻辑图如图 1.12 所示。

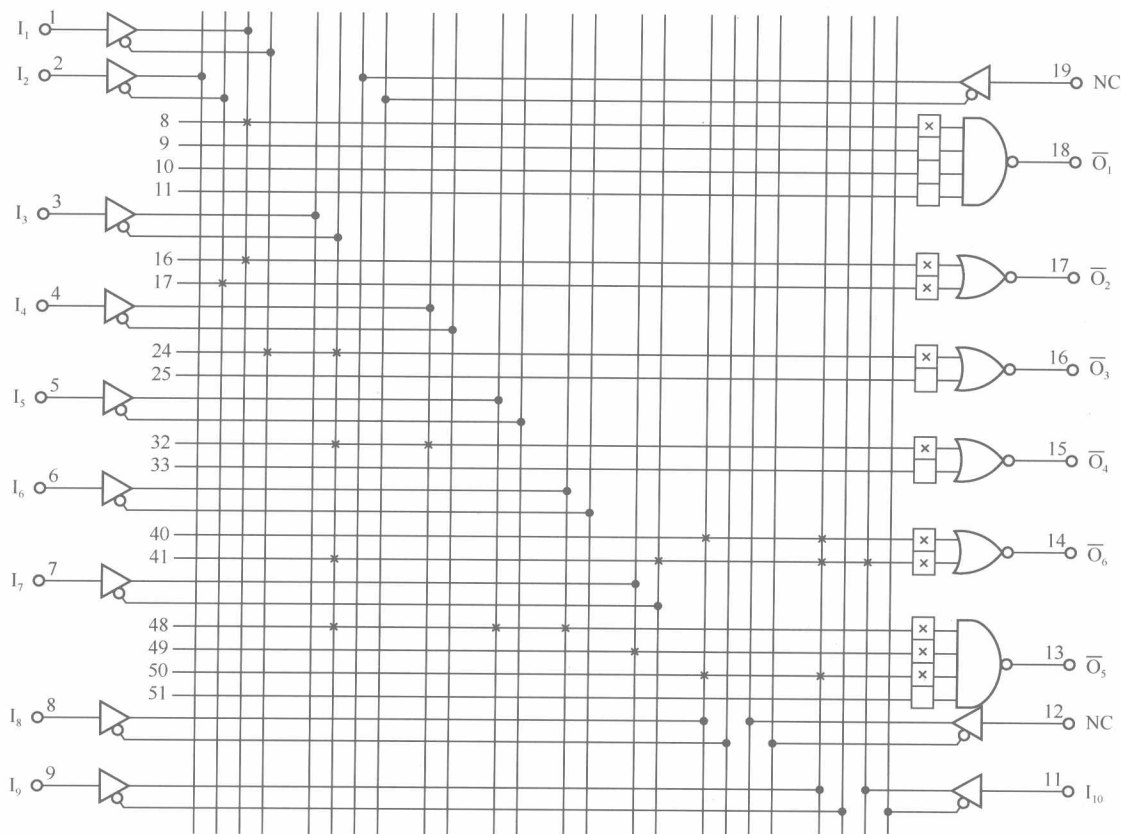


图 1.12 重构后的 PAL12L6 内部逻辑图

从这个重构实例的重构过程可以看出如下内容。

(1) 对于可重构器件来说，PLD 和 PAD 都有着一个共同点，那就是器件内部的逻辑单元或电路单元在未进行功能重构之前是不能实现任何电路功能的，只有经过功能重构之后才能具备某种电路功能。因此，要想使可重构器件成为具备某种电路功能的 ASIC，必须要对可重构器件进行功能重构工作。

(2) 在对可重构器件进行下载工作时，可重构器件能够识别并执行的指令只能是由 0 和 1 等组成的“二进制数字代码”，而不会执行其他形式的任何指令。因此，要想把可重构器件重构成欲设计的电路模块，必须先将欲设计电路的功能转换成二进制数字代码进行描述，否则可重构器件就会拒绝用户下载的任何指令文件。这些二进制数字代码被人们称为“机器码”。机器码的代码结构（即 0 和 1 的排列位置）是随着电路功能的不同而千变万化的，又是随着器件品种的不同而互不相同的，它不易记忆，很难看懂其与电路功能之间的对应关系，而且机器码的编写工作特别烦琐，非常容易出错，只有极其专业的软件工程师才能掌握，一般的

电子技术人员根本无法涉足。所以,要想让普通的电子技术人员也能对可重构器件进行功能重构工作,就必须开发出一种直观易懂、易编易记的硬件描述语言,来替代难编难懂又难记的机器码。

(3) 硬件描述语言虽然能替代机器码既通俗易懂又十分方便地描述出欲设计电路的功能,但是可重构器件却不能“读懂”更不能“执行”硬件描述语言,从而使重构工作仍然难以进行。因此,要想利用硬件描述语言对可重构器件的功能进行重构,就必须有一个软件工具来完成把硬件描述语言翻译转换成机器码的编译工作。

(4) 由此看来,要想把可重构器件开发成某种 ASIC,就必须先用硬件描述语言对电路模块的功能、行为、结构和接口进行描述,得到程序设计文件,再利用软件转换工具将程序设计文件编译转换成逻辑重构文件,最后通过实验开发系统将逻辑重构文件下载到可重构器件中并进行功能验证,才能得到我们自己设计制作的 ASIC。因此,把可重构器件开发成 ASIC 的工作流程分为三步——第一步是编程工作,第二步是编译工作,最后一步是下载工作。在编程工作中,我们必须会要用硬件描述语言来描述欲设计电路的功能;在编译工作中,我们必须会要会操作软件工具;在下载工作中,我们必须会要会使用实验开发系统,这就是我们在学习基于可重构器件的 EDA 技术时,主要应掌握硬件描述语言、软件转换工具、实验开发系统和可重构器件这 4 个方面内容的原因。

1.2.4 PAD 的重构原理

PAD 与 PLD 不同,其内部的放大电路已预先制作好,不需我们再去重新构造放大电路,所需做的重构工作只有电路互连、增益设置、带宽调整、阈值调整这 4 步,所以, PAD 的重构原理很简单,重构工作主要是通过连接布线、通断电阻、选择电容量来实现的。

1.3 可重构器件的比较和选用

CPLD 和 FPGA 虽然都属于可重构器件,在构成上又有许多相同点,但由于结构的不同以及重构原理的不同, CPLD 和 FPGA 还是存在着许多不同之处,这里对它们作一些比较。

CPLD 大都基于乘积项结构,实现工艺多为 E^2 PROM,重构次数可达 1 万次,系统断电时重构信息不会丢失;FPGA 大都基于查找表结构,实现工艺多为 SRAM,可以重构任意次,系统断电时重构信息也会丢失,每次上电时都需从器件外部的存储器中把重构信息调出重新写到 FPGA 里。

CPLD 中可重构通用逻辑块属于大单元,触发器数量较少,故更适合于完成组合逻辑电路;FPGA 中可重构通用逻辑块属于小单元,触发器数量较多,故更适合于完成时序逻辑电路。

CPLD 的集总式连线结构决定了它的时延是均匀和可预测的;FPGA 的分布式连线结构决定了它的时延是不均匀和不可预测的。

CPLD 的速度比 FPGA 快,价格也更便宜,使用起来更方便,保密性更强;FPGA 比 CPLD 在重构上具有更大的灵活性,集成度更高,更适合完成大的电路系统。

表 1.1 列出了 CPLD 与 FPGA 的性能对照。表 1.2 和表 1.3 简要列出了世界 3 大公司典型产品的主要参数,供选用时参考。