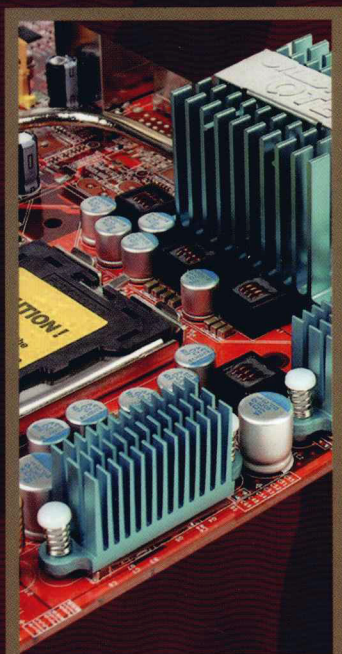


<http://www.phei.com.cn>

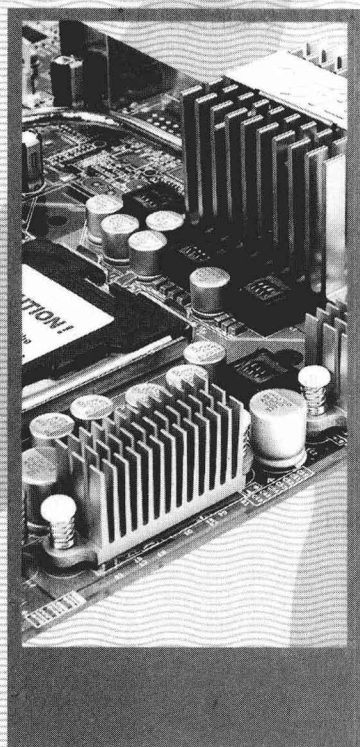


基于 Quartus II 的 FPGA/CPLD 设计实例精解

■ 李大社 王 彬 刘淑娥 等编著



电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY



基于 Quartus II 的 FPGA/CPLD 设计实例精解

■ 李大社 王 彬 刘淑娥 等编著

电子工业出版社

Publishing House of Electronics Industry

北京·BEIJING

内 容 简 介

本书以实例精解的方式讲述基于 Quartus II 的 FPGA/CPLD 设计方法和技巧, 主要包括 EDA 与可编程器件概述、Quartus II 开发环境、硬件设计语言、基本组合逻辑电路设计实例、基本时序逻辑电路设计实例、综合典型实例、小型应用和教学系统的开发, 以及设计系统时应注意的问题。书中所介绍的设计实例均从原理叙述和逻辑分析出发, 采用 EDA 方法进行设计输入、仿真及实现。

本书适合从事 FPGA/CPLD 设计开发的技术人员阅读, 也可作为高等学校相关专业的教学用书。

未经许可, 不得以任何方式复制或抄袭本书之部分或全部内容。

版权所有, 侵权必究。

图书在版编目(CIP)数据

基于 Quartus II 的 FPGA/CPLD 设计实例精解/李大社等编著. —北京: 电子工业出版社, 2010.11
(职场金钥匙)

ISBN 978-7-121-12024-4

I. ①基… II. ①李… III. ①可编程序逻辑器件—系统设计 IV. ①TP332.1

中国版本图书馆 CIP 数据核字(2010)第 201357 号

策划编辑: 张 剑

责任编辑: 雷洪勤

印 刷: 北京市李史山胶印厂

装 订:

出版发行: 电子工业出版社

北京市海淀区万寿路 173 信箱 邮编 100036

开 本: 787×1 092 1/16 印张: 15.75 字数: 403 千字

印 次: 2010 年 11 月第 1 次印刷

印 数: 4 000 册 定价: 35.00 元

凡所购买电子工业出版社图书有缺损问题, 请向购买书店调换。若书店售缺, 请与本社发行部联系, 联系及邮购电话: (010) 88254888。

质量投诉请发邮件至 zltz@phei.com.cn, 盗版侵权举报请发邮件至 dbqq@phei.com.cn。

服务热线: (010) 88258888。

前 言

在现代电子系统设计领域,EDA (Electronic Design Automation, 电子设计自动化) 已经逐渐成为电子系统的重要设计手段。无论是设计逻辑器件还是数字系统,其设计作业的复杂程度都在不断增加,仅仅依靠手工进行数字系统设计已经不能满足要求,所有的设计工作都需要在计算机上借助 EDA 工具进行。EDA 代替人工完成数字系统的逻辑综合、布局布线和设计仿真,而可编程器件可以按照 EDA 的设计结果快速形成一个实际的硬件系统,从而可以真实地验证用户的设计。用户可以反复地,如同修改软件一样来修改系统参数与描述,然后载入可编程器件中。

在教学领域,以前涉及可编程器件方面的内容比较少,而 EDA 方面也只是处于电子辅助设计阶段。随着器件技术与 EDA 技术的开发,使得 EDA 在大学本科、研究生教学中成为迫切的需要,目前已经成为电子技术教学中的大纲要求。但是却没有合适的教材,许多有关书籍不是太老就是内容不全,许多书仅仅是技术资料的翻译,或只是大量深入介绍 VHDL 语言,而没有把 EDA 作为学习数字电路或模拟电路的重要工具来讲解。

本书共分为 8 章,第 1 章介绍了 EDA 的发展、CPLD/FPGA 的基础知识;第 2 章介绍了利用 Quartus II 进行 FPGA/CPLD 设计的开发流程,包括设计输入、综合、布局布线、时序分析、仿真、编程和配置等;第 3 章主要介绍当前流行的 VHDL 语言、Verilog 语言的应用;第 4~7 章列出了大量的实例,如组合逻辑电路、时序逻辑电路等逻辑电路的设计方法;第 8 章主要介绍了在开发过程和实际应用中遇到的问题,这部分对于需要实际开发产品的读者和初学者有着非常重要的价值。

本书由于篇幅的限制,没有全部列出源代码,其中的绝大部分程序源代码对 EDA 开发人员具有很好的参考价值。

本书第 1~3 章由李大社编写,第 4 章和第 5 章由王彬和刘淑娥编写,第 6~8 章由管绍朋、邹德平和孙文燕编写。全书由李大社统稿、定稿。李军政和常彬参与了本书例子的录入和校对工作。

在本书的编写过程中,得到了许多朋友和专家的大力帮助,在此表示由衷的感谢,特别要感谢北京华晟公司的周德兴总工程师。

由于编者的能力有限,书中难免存在错误和不足之处,敬请广大读者和专家批评指正。

编者

目 录

第 1 章 EDA 与可编程器件概述	1
1.1 可编程芯片技术的发展	1
1.2 可编程器件开发	2
1.3 可编程器件厂商概述	4
1.4 开发环境与硬件描述语言	6
1.5 CPLD/FPGA 的基本结构	7
1.5.1 乘积项结构	8
1.5.2 查找表结构	10
第 2 章 Quartus II 开发指南	14
2.1 Quartus II 简介	14
2.2 Quartus II 的设计流程	15
2.2.1 设计输入	15
2.2.2 综合	16
2.2.3 布局布线	17
2.2.4 时序分析	18
2.2.5 仿真	19
2.2.6 编程与配置	20
2.2.7 工程更改管理	21
2.3 基于 Quartus II 的设计实例	22
第 3 章 硬件设计语言	31
3.1 硬件描述语言	31
3.2 VHDL 语言	31
3.2.1 VHDL 语言的基本结构	32
3.2.2 标志符	38
3.2.3 数据对象	39
3.2.4 数据类型	39
3.2.5 属性	42
3.2.6 运算符	43
3.2.7 VHDL 的语句和结构体	43
3.2.8 结构体的子结构描述	46
3.3 Verilog HDL	47
3.3.1 Verilog HDL 的数据类型	49
3.3.2 语言要素	57

3.3.3	表达式	61
3.3.4	赋值语句	67
3.3.5	条件语句	69
3.3.6	循环语句	70
第4章	基本组合逻辑电路设计实例	72
4.1	基本组合逻辑器件	72
4.1.1	三态器件	72
4.1.2	多任务器	73
4.1.3	全加器设计	76
4.2	复杂组合逻辑电路	79
4.2.1	带符号乘法器设计	79
4.2.2	编码器设计	82
4.2.3	加法器	86
4.2.4	表决器	88
第5章	基本时序逻辑电路设计实例	91
5.1	基本时序逻辑器件	91
5.1.1	寄存器	91
5.1.2	触发器	93
5.2	简单时序逻辑电路	96
5.2.1	计数器与分频器设计	96
5.2.2	状态机与通信编码/解码电路	100
5.2.3	动态扫描电路实现	107
5.2.4	发光管点阵实验	110
5.3	复杂时序逻辑电路	112
5.3.1	信号频率测量	112
5.3.2	可控脉冲发生器	113
5.3.3	数控脉宽调制 PWM	115
第6章	综合典型实例	119
6.1	处理器外设	119
6.1.1	Intel 8255 并行接口电路	119
6.1.2	8251 设计	126
6.1.3	A6850 异步通信接口	130
6.2	存储器接口	141
6.2.1	存储器	141
6.2.2	先进先出队列	145
6.3	数字信号处理	151
6.3.1	FFT 变换	151
6.3.2	CRC 校验	155

6.3.3	曼彻斯特编解码	159
6.4	控制类	165
6.4.1	数码锁设计	165
6.4.2	步进电动机的控制	170
6.4.3	电子手表 IC 设计	174
6.5	其他实验	174
6.5.1	键盘接口设计	174
6.5.2	抢答器	183
6.5.3	交通灯控制实验	186
第 7 章	小型应用和教学系统的开发	188
7.1	教学系统介绍	188
7.1.1	功能模块	188
7.1.2	系统功能	189
7.1.3	器件选择	190
7.2	电路原理图设计	192
7.2.1	CPLD 单元	192
7.2.2	单片机和通信单元	194
7.2.3	数码管发光管单元	195
7.2.4	输入单元	196
7.3	单片机程序设计	197
7.3.1	主程序	198
7.3.2	定时器和延时函数	202
7.3.3	获取命令和处理命令	203
7.3.4	获取配置文件	207
7.3.5	单片机更新	211
7.3.6	CPLD 配置	212
7.4	可编程逻辑器件设计	216
7.4.1	动态扫描显示	216
7.4.2	综合	217
第 8 章	系统设计中应注意的问题	220
8.1	工作条件	220
8.2	引脚电压	221
8.3	闭锁(次序上电问题)	221
8.4	高速设计	226
附录 A	本书涉及的专用简写词汇	229
附录 B	关于 Altera 公司下载电缆安装出现问题的解决	230
附录 C	开发实验装置介绍	235

第1章 EDA与可编程器件概述

1.1 可编程芯片技术的发展

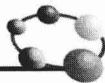
当今社会是数字化的社会,是数字集成电路广泛应用的社会。计算机、通信网络设备、移动电话、数字电视,各种自动化设备都要用到数字集成电路。与此同时,数字集成电路本身也在不断地进行更新换代。它由早期的小规模集成电路发展到超大规模集成电路,而按照摩尔效应的规律,超大规模集成电路的集成度正在不断刷新。目前正向着低电压、低功耗、高速度迈进。

在集成电路开发的过程中,设计与制造集成电路的任务已不完全由半导体厂商来独立承担。系统设计师们更愿意自己设计专用集成电路(ASIC)芯片,而且希望ASIC的设计周期尽可能短,最好是在实验室里就能设计出合适的ASIC芯片,并且立即投入到实际应用中,于是,出现了现场可编程逻辑器件(FPLD)。其中应用最广泛的当属现场可编程门阵列(FPGA)和复杂可编程逻辑器件(CPLD)。

可编程逻辑器件(PLD)能够完成各种数字逻辑功能。典型的PLD由一个“与”门和一个“或”门阵列组成,而任意一个组合逻辑都可以用“与-或”表达式来描述,所以,PLD能以乘积、和的形式完成大量的组合逻辑功能。

早期的可编程逻辑器件规模非常小,只有用熔丝方式才能得到可编程性能,且只能编程一次,后来开发出了紫外线可擦除只读存储器(EPROM)和电可擦除只读存储器(EEPROM)。这一阶段的产品主要有PAL(可编程阵列逻辑)和GAL(通用阵列逻辑)。PAL由一个可编程的“与”平面和一个固定的“或”平面构成,或门的输出可以通过触发器有选择地被置为寄存状态。PAL器件是现场可编程的,它的实现工艺有反熔丝技术、EPROM技术和EEPROM技术等。还有一类结构更为灵活的逻辑器件是可编程逻辑阵列(PLA),它也由一个“与”平面和一个“或”平面构成,但是这两个平面的连接关系是可编程的。PLA器件既有现场可编程的,又有掩膜可编程的。在PAL的基础上,又开发了一种通用阵列逻辑(Generic Array Logic, GAL),它采用EEPROM工艺,实现了电可擦除、电可改写,其输出结构是可编程的逻辑宏单元,因此它的设计具有很强的灵活性。

早期的PLD器件可以实现速度特性较好的逻辑功能,但其过于简单的结构也使它们只能实现规模较小的电路。20世纪80年代中期,Altera和Xilinx分别推出了类似于PAL结构的扩展型EPLD(Erasable Programmable Logic Device)和与标准门阵列类似的FPGA(Field Programmable Gate Array),它们都具有体系结构和逻辑单元灵活、集成度高以及适用范围宽等特点。与门阵列等其他ASIC相比,它们又具有设计开发周期短、设计制造成本低、开发工具先进、标准产品无须测试、质量稳定以及可实时在线检验等优点,因此被广泛应用于产品的原型设计。



1.2 可编程器件开发

PLD 是可编程逻辑器件 (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, 只是实现原理略有不同, 所以我们有时可以忽略这两者的区别, 统称为可编程逻辑器件或 PLD/FPGA。

PLD 是电子设计领域中最具活力和发展前途的一项技术, 它的影响丝毫不亚于 20 世纪 70 年代单片机的出现和使用。可以毫不夸张地说, PLD 能完成任何数字器件的功能, 上至高性能 CPU, 下至简单的 74 系列电路, 都可以用 PLD 来实现。PLD 如同一张白纸, 工程师可以通过传统的原理图输入法, 或是硬件描述语言, 自由地设计一个数字系统。通过软件仿真, 我们可以事先验证设计的正确性。在 PCB 完成以后, 还可以利用 PLD 的在线修改能力, 随时修改设计而不必改动硬件电路。使用 PLD 来开发数字电路, 可以大大缩短设计时间, 减少 PCB 面积, 提高系统的可靠性。PLD 的这些优点使得 PLD 技术在 20 世纪 90 年代以后得到飞速的发展, 同时也大大推动了 EDA 软件和硬件描述语言 (HDL) 的进步。

如何使用 PLD 呢? 其实 PLD 的使用很简单, 学习 PLD 比学习单片机要简单得多, 有数字电路基础, 会使用计算机, 就可以进行 PLD 的开发。不熟悉 PLD 的朋友, 可以先看一看可编程逻辑器件的发展历程。开发 PLD 需要 PLD 开发软件和 PLD 芯片。

1. PLD 开发软件

由于 PLD 软件已经发展得相当完善, 用户甚至可以不用详细了解 PLD 的内部结构, 就能用自己熟悉的方法, 如原理图输入或 HDL 语言来完成相当优秀的 PLD 设计。所以对初学者来说, 首先应了解 PLD 开发软件和开发流程。了解 PLD 的内部结构, 将有助于提高我们设计的效率和可靠性。

获得 PLD 开发软件的途径非常多。许多 PLD 公司都提供免费试用版或演示版 (当然商业版大都是收费的), 例如可以免费从 <http://www.altera.com/> 上下载 Altera 公司的 Maxplus II (Baseline 版或 E+MAX 版), 或向其代理商索取这套软件。Xilinx 公司也提供免费软件: WebPack 可以从 Xilinx 网站下载。Lattice、Actel 等公司也都有类似的免费软件提供。以上免费软件都需要在网上注册申请 License 文件。通常这些免费软件已经能够满足一般设计的需要, 要想软件功能更强大一些, 只能购买商业版软件。

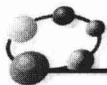
如果打算使用 VHDL 或 Verilog HDL 硬件描述语言来开发 PLD/FPGA, 通常还需要使用一些专业的 HDL 开发软件, 这是因为 FPGA 厂商提供的软件的 HDL 综合能力一般都不是很强, 需要其他软件来配合使用。

对于 PLD 产品, 一般分为两类:

(1) 基于乘积项 (Product-Term) 技术, 一般用于 EEPROM (或 Flash) 工艺的中小规模 PLD。

(2) 基于查找表 (Look-Up Table) 技术, 一般用于 SRAM 工艺的大规模 PLD/FPGA。

EEPROM 工艺的 PLD 密度小, 多用于 5000 门以下的小规模设计, 适合做复杂的组合逻辑, 如译码。SRAM 工艺的 PLD (FPGA), 密度高, 触发器多, 多用于 10000 门以上的大规模设计, 适合做复杂的时序逻辑, 如数字信号处理和各种算法。



2. PLD 开发

在 PLD/FPGA 开发软件中完成设计以后, 软件会产生一个最终的编程文件 (如 .pof)。应如何将编程文件烧到 PLD 芯片中去呢?

对于基于乘积项 (Product-Term) 技术, EEPROM (或 Flash) 工艺的 PLD (如 Altera 的 MAX 系列, Xilinx 的 XC9500 系列), 厂家提供编程电缆, 如 Altera 公司的叫做 Byteblaster。电缆一端装在计算机的并行打印口上, 另一端接在 PCB 板上的一个十芯插头上, PLD 芯片有 4 个引脚 (编程脚) 与插头相连, 如图 1-1 所示。

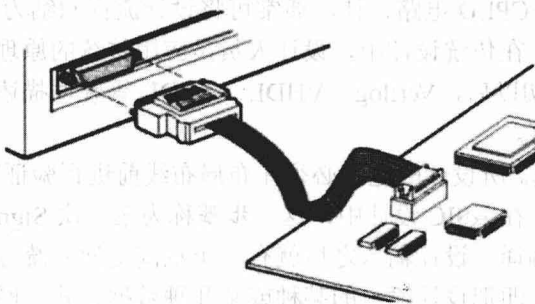


图 1-1 PLD 芯片与插头相连

下载电缆向系统板上的器件提供配置或编程数据, 这就是所谓的在线可编程 (ISP)。下载电缆使用户能够独立地配置 PLD 器件, 而不需要编程器或任何其他编程硬件。编程电缆可以向代理商购买, 也可以根据厂家提供的编程电缆的原理图自己制作。早期的 PLD 是不支持 ISP 的, 它们需要用编程器烧写。目前的 PLD 都可以用 ISP 在线编程, 也可用编程器编程。这种 PLD 可以加密, 并且很难解密。具体应用步骤如图 1-2 所示。

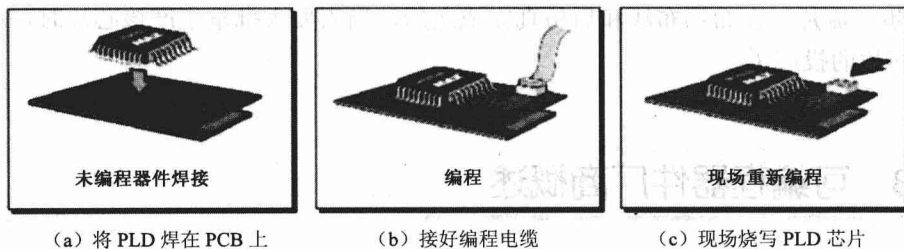


图 1-2 应用步骤

对于基于查找表技术 (Look-Up Table) 的 SRAM 工艺的 FPGA (如 Altera 的所有 FLEX、ACEX、APEX 系列, Xilinx 的 Spartan、Virtex), 由于 SRAM 工艺的特点, 断电后数据会消失, 因此调试期间可以用下载电缆配置 PLD 器件, 调试完成后, 需要将数据固化在一个专用的 EEPROM 中 (用通用编程器烧写), 上电时, 由这片配置 EEPROM 先对 PLD 加载数据, 十几毫秒后, PLD 即可正常工作 (也可由 CPU 配置 PLD)。但 SRAM 工艺的 PLD 一般不可以加密。

对于另一种反熔丝 (Anti-fuse) 技术的 FPGA, 如 Actel、Quicklogic 的部分产品就采用这种工艺, 其用法与 EEPROM 的 PLD 一样, 但这种 PLD 不能重复擦写, 所以初期开发过程比较麻烦, 费用也比较高昂。但反熔丝技术也有许多优点, 例如, 布线能力更强, 系统速度



更快, 功耗更低, 同时抗辐射能力强, 耐高低温, 可以加密, 所以在一些有特殊要求的领域中运用较多, 如军事及航空航天。

3. ISP 技术与开发流程

20 世纪 90 年代初, 在系统可编程 (In-System Programmability, ISP) 技术首先应用于可编程逻辑器件, 它改变了数字电子系统的设计和实现方法, 为人们提供了更为方便的 EDA 手段。所谓“在系统可编程”, 是指可编程器件在不脱离所在应用系统的情况下, 能够通过计算机对其进行编程, 而不需要专用的编程器。

进行 ASIC 与 FPGA/CPLD 电路设计, 通常可将设计流程归纳为以下 7 个步骤。

第 1 步: 设计输入。在传统设计中, 设计人员是应用传统的原理图输入方法来开始设计的。自 20 世纪 90 年代初以后, Verilog、VHDL、AHDL 等硬件描述语言的输入方法得到了广大工程设计人员的认可。

第 2 步: 布线前仿真。所设计的电路必须在布局布线前进行验证, 主要目的是在仿真时, 验证电路功能是否有效。在 ASIC 设计中, 这一步骤称为第一次 Sign-off。

第 3 步: 设计输入编译。设计输入之后就有一个从高层次系统行为设计向低层次门级逻辑电路的转化翻译过程, 即把设计输入的某种或某几种数据格式 (网表) 转化为底层软件能够识别的某种数据格式 (网表), 以求达到与其工艺无关。

第 4 步: 设计输入的优化。对于上述综合生成的网表, 根据布尔方程功能等效的原则, 用更小更快的综合结果替代一些复杂的单元, 并与指定的库映射生成新的网表。

第 5 步: 布局布线。当初步的仿真被验证后, 就开始布局布线。这一步可规划出 ASIC 和 FPGA/CPLD 设计。

第 6 步: 后仿真。设计人员需要利用在布局布线中获得的更精确的 RC 参数再次验证电路的功能和时序。在 ASIC 设计中, 这一步骤称为第二次 Sign-off。

第 7 步: 流片。在布局布线和后仿真完成之后, 当需要大批量生产该芯片时, 就可以开始 ASIC 芯片的投产了。

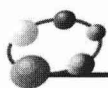


1.3 可编程器件厂商概述

随着可编程逻辑器件应用的日益广泛, 许多 IC 制造厂家涉足 PLD/FPGA 领域。目前世界上有十几家生产 CPLD/FPGA 的公司, 最大的三家是: Altera、Xilinx、Lattice Vantis, 其中 Altera 和 Xilinx 占有了 60% 以上的市场份额。

1. Altera

Altera 于 20 世纪 90 年代以后发展很快, 是最大可编程逻辑器件供应商之一, 主要产品有: MAX3000/7000、FELX6K/10K、APEX20K、ACEX1K 等。普遍认为其开发工具——MAX Plus II 是较成功的 PLD 开发平台。但 MAX Plus II 的 HDL 综合能力不够强, 一般建议使用 Altera 公司提供的免费 VHDL 综合工具。



2. Xilinx

它是 FPGA 的发明者, 老牌的 PLD 公司, 是最大的可编程逻辑器件供应商之一。产品种类较全, 主要有: XC9500/4000, Coolrunner (XPLA3), Spartan, Vertex。开发软件为 Foundation 和 ISE。通常来说, 在欧洲用 Xilinx 的人较多, 在亚太地区用 Altera 的人较多, 在美国则是平分秋色。全球 PLD/FPGA 产品 60% 以上是由 Altera 和 Xilinx 提供的。可以说, Altera 和 Xilinx 共同决定了 PLD 技术的开发方向。

3. Lattice Vantis

Lattice 是 ISP 技术的发明者, ISP 技术极大地促进了 PLD 产品的开发, 与 Altera 和 Xilinx 相比, 其开发工具比 Altera 和 Xilinx 略逊一筹。中小规模 PLD 比较有特色, 而且参考书较多, 不过其大规模 PLD 的竞争力还不够强 (Lattice 没有基于查找表技术的大规模 FPGA)。Lattice 1999 年推出可编程模拟器件; 1999 年收购 Vantis (原 AMD 子公司), 成为第三大可编程逻辑器件供应商; 2001 年 12 月收购 Agere 公司 (原 Lucent 微电子部) 的 FPGA 部门; 主要产品有 ISPLSI 2000/5000/8000, MACH4/5。

4. Actel

Actel 是反熔丝 (一次性烧写) PLD 的领导者。由于反熔丝 PLD 抗辐射、耐高低温、功耗低、速度快, 所以在军品和宇航应用上有较大的优势。Altera 和 Xilinx 则一般不涉足军品和宇航级市场。

5. Cypress

PLD/FPGA 不是 Cypress 的最主要业务, 但有一定的用户群。中国地区代理商有: 富昌电子, 裕利 (科汇二部), 德创电子专业 PLD/FPGA 公司; 以一次性反熔丝工艺为主, 在中国地区销售量不大。

6. Lucent

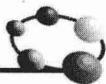
Lucent 的主要特点是有不少用于通信领域的专用 IP 核, 但 PLD/FPGA 不是 Lucent 的主要业务, 在中国地区使用的人很少。2000 年 Lucent 的半导体部独立出来并更名为 Agere。2001 年 12 月 Agere 公司的 FPGA 部门被 Lattice 收购。

7. ATMEL

PLD/FPGA 不是 ATMEL 的主要业务, ATMEL 的中小规模 PLD 做得不错。ATMEL 也做了一些与 Altera 和 Xilinx 兼容的片子, 但在品质上与原厂家还有一些差距。

8. Clear Logic

它生产一些与著名 PLD/FPGA 大公司兼容的芯片, 这种芯片可将用户的设计一次性固化, 不可编程, 批量生产时的成本较低。但由于大部分用户对其品质不放心, 并且担心失去大公司的技术支持, 所以使用者极少。



9. WSI

该公司主要生产 PSD (单片机可编程外围芯片) 产品。这是一种特殊的 PLD, 如最新的 PSD8xx、PSD9xx, 集成了 PLD、EPROM、Flash 并支持 ISP (在线编程), 其价格偏贵, 但集成度高, 主要用于配合单片机工作。目前中国地区代理商是晓龙国际和武汉力源。2000 年 8 月 WSI 被 ST 收购。



1.4 开发环境与硬件描述语言

随着现代材料的开发和工艺水平的提高, 超大规模集成电路已经可以将一个很大的电路系统集成到一个很小的芯片内。随着高层次自动综合技术、混合模拟技术及可测试性技术理论的研究和应用, 诞生了自顶向下的设计方法 (Top-Down Design, TDD), 它直接面向用户需要, 从系统总体出发, 根据电路系统的行为和功能要求, 从上到下逐层完成相应的设计描述。然后经过综合与优化、模拟与验证, 直到生成器件, 完成系统的整体设计, 同时它也推动了电子设计自动化 EDA 进入电子系统设计自动化时代。要进行自动化设计, 就需要用形式化方法来描述电路, 即用 HDL 硬件描述语言进行电路设计。目前, 国际上越来越多的 EDA 工具都接受 HDL 语言作为设计输入, 如 Mentor Graphics 的 Autologic E、Cadence 的 SPW、Synopsys 的 VHDL/Design Compiler 和 Altera 的 Quartus 等, 都可以解决从系统的高层次行为描述直接生成 ASIC 器件的一系列技术问题。

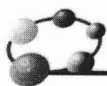
VHDL 的英文全名是 Very-High-Speed Integrated Circuit Hardware Description Language, 诞生于 1982 年。1987 年底, VHDL 被 IEEE 和美国国防部确认为标准硬件描述语言。自 IEEE 公布了 VHDL 的标准版本——IEEE-1076 (简称 87 版) 之后, 各 EDA 公司相继推出了自己的 VHDL 设计环境, 或宣布自己的设计工具可以与 VHDL 接口。此后 VHDL 在电子设计领域得到了广泛的应用, 并逐步取代了原有的非标准的硬件描述语言。1993 年, IEEE 对 VHDL 进行了修订, 从更高的抽象层次和系统描述能力上扩展 VHDL 的内容, 公布了新版本的 VHDL, 即 IEEE 标准的 1076-1993 版本 (简称 93 版)。现在, VHDL 和 Verilog HDL 作为 IEEE 的工业标准硬件描述语言, 又得到众多 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。

VHDL 主要用于描述数字系统的结构、行为、功能和接口。除了含有许多具有硬件特征的语句外, VHDL 的语言形式和描述风格与句法十分类似于一般的计算机高级语言。VHDL 的程序结构特点是将一项工程设计或设计实体 (可以是一个元件、一个电路模块或一个系统) 分成外部 (或称可视部分及端口) 和内部 (或称不可视部分) 两部分。在对一个设计实体定义了外部界面后, 一旦其内部开发完成后, 其他的设计就可以直接调用这个实体。这种将设计实体分成内、外部分的概念是 VHDL 系统设计的基本点。

应用 VHDL 进行工程设计有以下优点。

(1) 与其他的硬件描述语言相比, VHDL 具有更强的行为描述能力, 从而决定了它成为系统设计领域最佳的硬件描述语言。强大的行为描述能力是避开具体的器件结构, 从逻辑行为上描述和设计大规模电子系统的重要保证。

(2) VHDL 丰富的仿真语句和库函数, 使得在任何大系统的设计早期就能查验设计系统



的功能可行性,随时可对设计进行模拟仿真。

(3) VHDL 语句的行为描述能力和程序结构决定了它具有支持大规模设计的分解和已有设计的再利用功能,符合市场需求的大规模系统高效、高速地完成必须有多人甚至多个组共同并行工作。

(4)对于用 VHDL 完成的设计,可以利用 EDA 工具进行逻辑综合和优化,并自动把 VHDL 描述设计转变成门级网表。

(5) VHDL 对设计的描述具有相对独立性,设计者可以不懂硬件的结构,也不必管最终设计实现的目标器件是什么,而进行独立的设计。

以下程序是一个简单的 VHDL 的例子(12 位寄存器)。

```
-- VHDL Example
-- User-Defined Macrofunction
ENTITY reg12 IS
PORT (
d: IN BIT_VECTOR (11 DOWNT0 0) ;
clk : IN BIT;
q: OUT BIT_VECTOR (11 DOWNT0 0)) ;
END reg12;
ARCHITECTURE a OF reg12 IS
BEGIN
PROCESS
BEGIN
WAIT UNTIL clk = '1';
q <= d;
END PROCESS;
END a;
```



1.5 CPLD/FPGA 的基本结构

CPLD 与 FPGA 都是可编程逻辑器件,它们是在 PAL、GAL 等逻辑器件的基础之上发展起来的。与以往的 PAL、GAL 等相比较,FPGA/CPLD 的规模比较大,它可以替代几十块甚至几千块通用 IC 芯片。这样的 FPGA/CPLD 实际上就是一个子系统部件。这种芯片受到世界范围内电子工程设计人员的广泛关注和普遍欢迎。经过了十几年的发展,许多公司都开发出多种可编程逻辑器件。比较典型的就是 Xilinx 公司的 FPGA 器件系列和 Altera 公司的 CPLD 器件系列,它们开发较早,占有较大的 PLD 市场。通常来说,在欧洲用 Xilinx 的人多,在日本和亚太地区用 Altera 的人多,在美国则是平分秋色。全球 PLD/FPGA 产品 60%以上是由 Altera 和 Xilinx 提供的。可以说,Altera 和 Xilinx 共同决定了 PLD 技术的发展方向。当然还有许多其他类型器件,如: Lattice、Vantis、Actel、Quicklogic、Lucent 等。

尽管 FPGA/CPLD 和其他类型 PLD 的结构各有其特点和长处,但概括起来,它们是由以下三部分组成的。



- 二维的逻辑块阵列，构成了 PLD 器件的逻辑组成核心；
- I/O 块：连接逻辑块的互连资源；
- 连线资源：由各种长度的连线线段组成，其中也有一些可编程的连接开关，它们用于逻辑块之间、逻辑块与输入/输出块之间的连接。

1.5.1 乘积项结构

采用基于乘积项 (Product-Term) 的 PLD 结构的芯片有：Altera 的 MAX7000, MAX3000 系列 (EEPROM 工艺)，Xilinx 的 XC9500 系列 (Flash 工艺)，Lattice、Cypress 的大部分产品 (EEPROM 工艺)。

基于乘积项的 PLD 的内部结构 (以 MAX7000 为例，其他型号的结构与此都非常相似) 如图 1-3 所示。

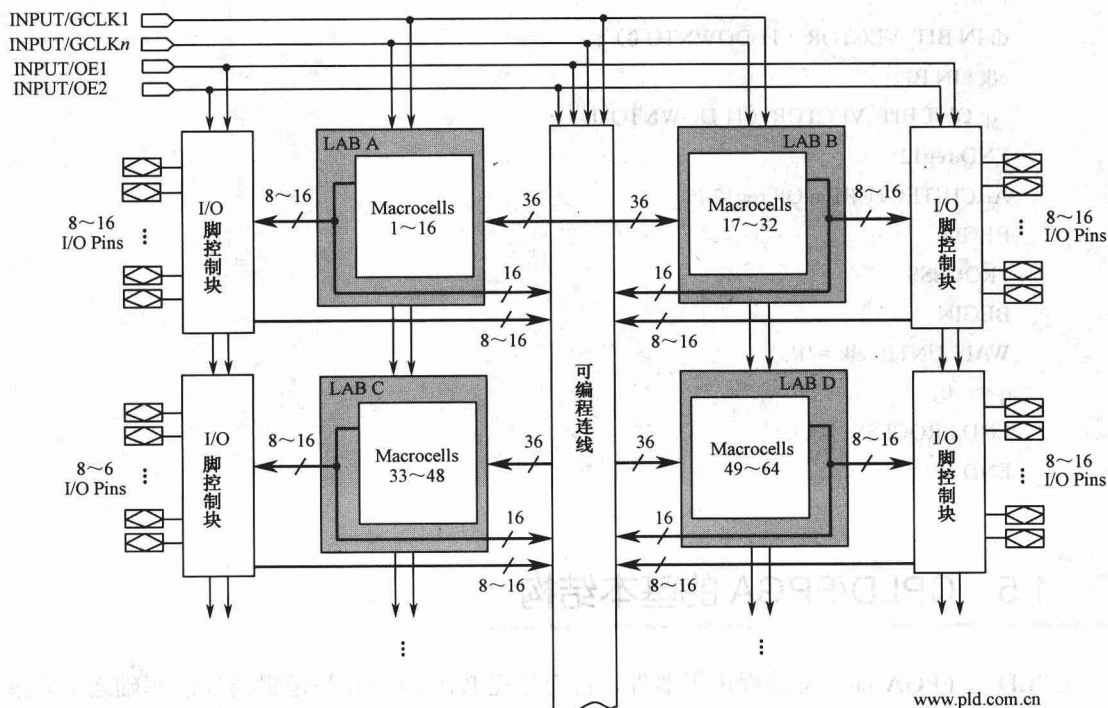


图 1-3 基于乘积项的 PLD 的内部结构

这种 PLD 可分为三块结构：宏单元 (Macrocell)、可编程连线 (PIA) 和 I/O 控制块。宏单元是 PLD 的基本结构，由它来实现基本的逻辑功能。图 1-3 中核心部分是多个宏单元的集合 (因为宏单元较多，没有一一画出)。可编程连线负责信号传递，连接所有的宏单元。I/O 控制块负责输入/输出的电气特性控制，比如可以设定集电极开路输出、摆率控制、三态输出等。图 1-3 中左上角的 INPUT/GCLK1、INPUT/GCLK_n、INPUT/OE1、INPUT/OE2 是全局时钟信号、清零信号和输出使能信号，这几个信号有专用连线与 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。



宏单元的具体结构如图 1-4 所示。

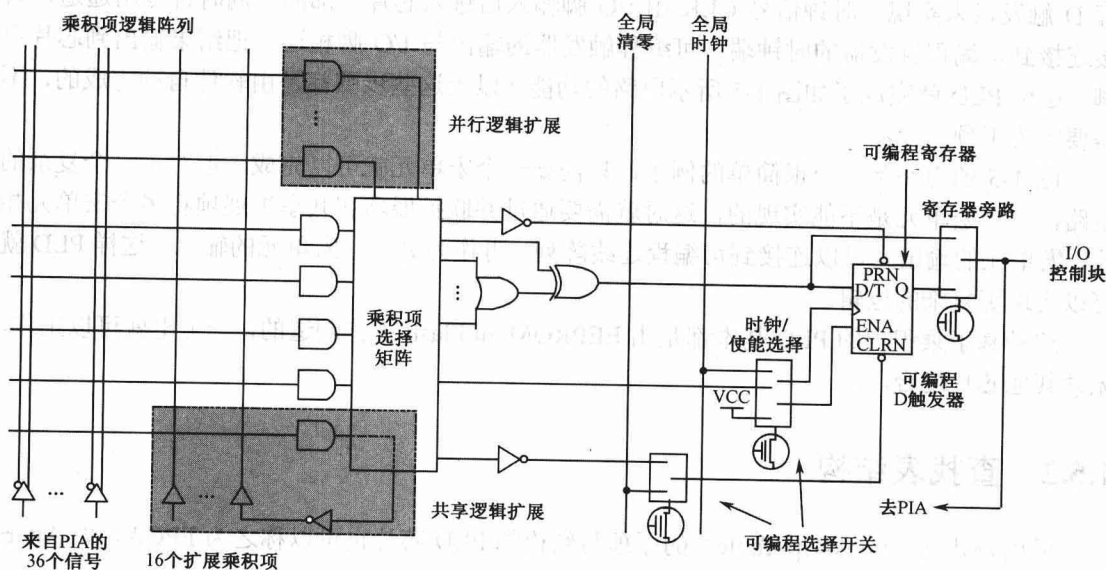


图 1-4 宏单元的结构

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。图右侧是一个可编程 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，信号直接给 PIA 或输出到 I/O 脚。

下面我们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 1-5 所示。

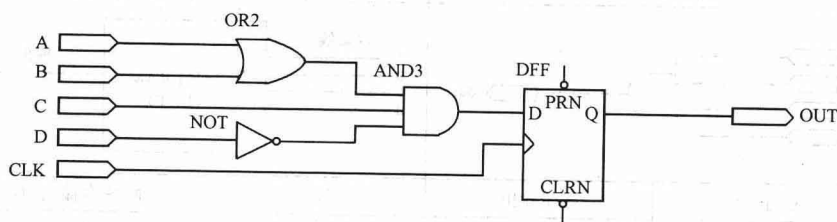


图 1-5 简单电路

假设组合逻辑的输出（AND3 的输出）为 f ，则

$$f = (A+B) \cdot C \cdot \bar{D} = A \cdot C \cdot \bar{D} + B \cdot C \cdot \bar{D}$$

PLD 将以如图 1-6 所示的方式来实现组合逻辑 f 。

A、B、C、D 由 PLD 芯片的引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。图中每一个叉表示相连（可编程熔丝导通），所以得到： $f = f_1 + f_2 = (A \cdot C \cdot \bar{D}) + (B \cdot C \cdot \bar{D})$

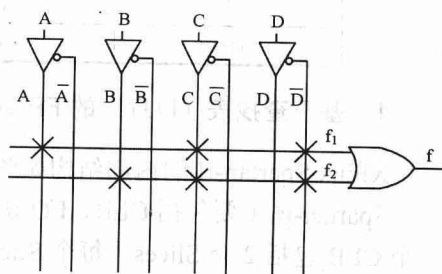


图 1-6 电路的 PLD 实现方式