



PLD

01010101010101010110

MAX+plus II 和 Quartus II 应用与开发技巧

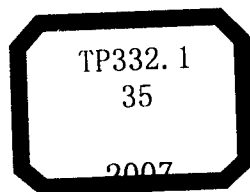
01010101010101010110

王辉 殷颖 陈婷 俞一鸣 等编著



 机械工业出版社
CHINA MACHINE PRESS





可编程逻辑器件实用开发技术丛书

MAX+plus II 和 Quartus II 应用与开发技巧

王 辉 殷 颖 陈 婷 俞一鸣 等编著

机械工业出版社

MAX+plus II 和 Quartus II 是 Altera 公司提供的可编程逻辑器件开发工具, 理解并掌握它们就可以利用 Altera 公司的可编程逻辑器件开发出符合要求的数字系统。

本书从实用的角度出发, 全面系统地介绍了 Altera 公司的可编程逻辑器件及 MAX+plus II 和 Quartus II 开发软件。全书包含 3 部分: 第 1 部分介绍了 CPLD 和 FPGA 的基本结构, 在此基础上介绍了 Altera 公司各个系列的 CPLD 和 FPGA; 第 2 部分通过几个实例全面系统地介绍了 MAX+plus II 软件的功能特点和使用方法; 第 3 部分通过一个较复杂的实例详细介绍了 Quartus II 软件的功能特点和使用方法。

本书可作为广大从事硬件电路工作的技术人员学习和掌握 Altera 公司各系列 CPLD 和 FPGA 以及 MAX+plus II 和 Quartus II 的实用参考书, 也可作为高等院校电子信息与通信等相关专业师生的参考教材。

图书在版编目 (CIP) 数据

MAX+plus II 和 Quartus II 应用与开发技巧/王辉等编著. —北京: 机械工业出版社, 2007.1

(可编程逻辑器件实用开发技术丛书)

ISBN 978-7-111-20726-9

I. M... II. 王... III. 可编程逻辑器件—应用软件, MAX+plus II、Quartus II
IV. TP332.1

中国版本图书馆 CIP 数据核字 (2007) 第 004696 号

机械工业出版社 (北京市百万庄大街 22 号 邮政编码 100037)

责任编辑: 张俊红 版式设计: 霍永明 责任校对: 刘志文

封面设计: 马精明 责任印制: 李 妍

北京铭成印刷有限公司印刷

2007 年 3 月第 1 版第 1 次印刷

184mm×260mm·20.5 印张·509 千字

0 001—4 000 册

标准书号: ISBN 978-7-111-20726-9

定价: 33.00 元

凡购本书, 如有缺页、倒页、脱页, 由本社发行部调换

销售服务热线电话: (010) 68326294

购书热线电话: (010) 88379639 88379641 88379643

编辑热线电话: (010) 88379768

封面防伪标均为盗版

丛 书 序

随着科学技术的迅猛发展，电子工业界经历了巨大的飞跃。集成电路的设计正朝着速度快、性能高、容量大、体积小和微功耗的方向发展。这种发展必将导致集成电路的设计规模日益增大，复杂程度日益增高。基于这种情况，可编程逻辑器件的出现和发展大大改变了传统的系统设计方法，这种方法使得电子系统设计变得更加简单方便、灵活快速，因此掌握可编程逻辑器件和相应的设计技术已经成为从事电子系统设计的设计工程师和科研人员的一项重要设计手段和技能。

可编程逻辑器件和相应的设计技术体现在三个主要方面：一是可编程逻辑器件的芯片技术；二是适用于可编程逻辑器件的硬件编程技术，即 VHDL 技术和 Verilog 技术；三是可编程逻辑器件设计的 EDA 开发工具，它主要用来进行可编程逻辑器件应用的具体实现。

可编程逻辑器件 (Programmable Logic Device, PLD) 的逻辑功能是由设计人员根据系统设计的具体要求，通过相应的器件编程来实现的。另外，由于 PLD 的集成度很高，因此它可以满足大多数数字系统设计的需要。历史上，可编程逻辑器件经历了 PROM、PLA、PAL、GAL、EPLD 到 CPLD 和 FPGA 的发展过程，在结构、制造工艺、集成度、逻辑功能、速度和功耗上都有了很大的提高和改进。其中，CPLD 和 FPGA 由于集成度非常高，因此这两种器件成为目前可编程逻辑器件的主流。

可编程逻辑器件的硬件编程技术主要体现在硬件描述语言的应用中，目前广泛使用的硬件描述语言是 VHDL 和 Verilog。这两种描述语言具有强大的功能和硬件描述能力，易于共享和复用，同时还具有独立于器件和工艺的设计能力，因此它们得到了各种 EDA 工具和集成电路厂商的普遍认同和推广，目前正在全球范围内的电子系统设计领域获得广泛应用。如今，国内外一些用户在购买和使用各种 EDA 工具时，通常都把是否支持 VHDL 和 Verilog 作为 EDA 工具是否先进的标准之一。

EDA 开发工具主要包括编辑器、仿真工具、检查/分析工具和优化/综合工具等。其中，编辑器用来对设计输入进行图形或者文本等方面的编辑操作；仿真工具是用来完成设计仿真操作的 EDA 开发工具，主要包括逻辑仿真工具和时序仿真工具；检查/分析工具用来对设计的逻辑产生可能性、电路的电气特性以及时序关系等进行检查和分析；优化/综合工具用来把一种硬件描述转化为底层描述，在转化的过程中伴随着设计的某种优化。现在，高级的 EDA 开发工具都是一种集成的开发环境，即集成了上述的所有开发工具，这样就可以用一种集成开发环境来完成所有的设计工作。

可见，对于可编程逻辑器件的设计来说，上面三个主要方面是相辅相成不可分割的，一个高水平的可编程逻辑设计人员必须掌握这三个方面的技术，这样才能够满足实际设

计工作的需要,从而实现成本低、设计简单和资源优化的完美设计。现在,可编程逻辑器件获得了极其广泛的应用,无论是电子设计工程师还是高等院校的学生,都迫切需要系统地来学习相应的技术,因此也就需要一套理论严谨、内容新颖、实用性较强的可编程逻辑器件丛书来满足广大读者的学习需要。基于这一点,机械工业出版社的领导和编辑组织了这套“可编程逻辑器件实用开发技术丛书”,这套丛书重点介绍了 Xilinx 公司和 Altera 公司的 CPLD/FPGA、相应的 EDA 开发工具 ISE 和 MAX+plus II/Quartus II 以及相应的 VHDL 和 Verilog 设计技术。本套丛书具体包括:

- ◀ 《VHDL 设计实例与仿真》
- ◀ 《通信收发信机的 Verilog 实现与仿真》
- ◀ 《面向 CPLD/FPGA 的 Verilog 设计》
- ◀ 《面向 CPLD/FPGA 的 VHDL 设计》
- ◀ 《MAX+plus II 和 Quartus II 应用与开发技巧》
- ◀ 《ISE 应用与开发技巧》
- ◀ 《Xilinx 可编程逻辑器件的应用与设计》
- ◀ 《Altera 可编程逻辑器件的应用与设计》

为了保证这套丛书的高质量和实用性,特组织了一批具有丰富可编程逻辑器件设计经验的工程师来进行相应丛书的编写。这套丛书从实际应用的角度出发,全面系统、由浅入深地介绍了可编程逻辑器件的各个相关技术,可以使广大读者快速高效地掌握可编程逻辑器件的知识。本套丛书读者范围十分广泛,它既可作为高等学校计算机和电子工程专业的研究生和本科生的教材或教学参考书,也可作为广大电子电路设计工程师、ASIC 设计人员和系统设计人员的参考书。

由于可编程逻辑器件技术发展十分迅速,加上编写时间相对紧张,书中难免存在不足,恳请广大读者和专家批评指正,联系信箱: buptzjh@163.com。

丛书编委会

前 言

21 世纪是一个信息化的社会，电子信息技术将会随着时间的推移而不断向前发展。如果说前 20 年的历史主要是 PC 的需求驱动电子技术发展的话，那么后 20 年除 PC 仍会继续发挥影响外，与因特网相结合的、可移动的、网络化的、智能化的、多媒体的实时信息设备将是主要的驱动力。半导体行业在 20 世纪 90 年代进入数字消费电子、网络技术的时代后，人类对电子产品的需求就从 PC 扩大到数字消费电子。这种趋势的产品体现是数字相机、DVD、电子游戏机、数字电视机和移动通信设备。这种趋势一直延续到今天，使得全球进入数字化时代。在这样的时代背景和技术背景下，半导体行业关注的重点在于对器件的可编程能力、多处理器架构、SoC(片上系统)和 SiP(Silicon in Package)以及相应的下一代半导体工艺。其中，可编程能力将成为主导技术。在新的数字消费时代，产品的生命期只有一年的时间，可编程能力成为必须具备的一个要求。可编程能力已成为业界共识的下一代半导体器件的一个主要特性。从整个半导体行业的发展步伐上来看，1987~1997 年是 ASIC 风行的时期；而 1997~2007 年是现场可编程器件的大好时光，在这个阶段中，制造标准化但应用定制化是一个明显的特征；而 2007 年后，可自动配置的 SoC 和 SiP 将成为下一个发展的主流。

可编程器件的发展趋势及其当前在电子系统中所占的地位，决定了可编程器件的熟练应用是当今电子工程师必须具备的一项基本能力。应用可编程器件就必须使用相应的 EDA (Electronic Design Automation) 工具。目前，任何电子系统的开发都离不开 EDA 工具的使用。EDA 就是以计算机为工作平台，以 EDA 软件工具为开发环境，以电子系统设计为应用方向的电子产品自动化设计过程。EDA 工程广义的定义范围包括半导体工艺设计自动化、可编程器件设计自动化、电子系统设计自动化、印制电路板设计自动化、仿真与测试故障诊断以及型式验证自动化。随着半导体工艺水平的不断提高，芯片中已能够集成几百万门电路，一个完整的数字系统集成于一块芯片上 (System on Chip, SoC) 已成为可能，而用经典的电子设计方法完成这样的设计已十分困难。随着电子技术、计算机硬/软件技术的不断发展，计算机应用水平的不断提高，人们已能利用计算机进行电子系统辅助设计，大大提高了设计效率，减轻了设计人员的劳动，缩短了设计周期，提高了设计成功率，减少了设计缺陷。

Altera 公司是 20 世纪 90 年代以后发展起来的最大的可编程逻辑器件供应商之一，与另一供应商 Xilinx 公司共同占领了全球 PLD/FPGA 市场份额的 60% 以上，可以说 Altera 和 Xilinx 公司共同决定了 PLD 技术的发展方向。熟练应用 Altera 公司的可编程逻辑器件，不仅可以实现设计数字系统的需求，还可以跟上 PLD 技术发展的步伐。MAX+plus II 和 Quartus II 是 Altera 公司提供的 CPLD/FPGA 开发软件。MAX+plus II 是第 3 代设计工具，

适合于中小规模的可编程器件的设计,提供了从设计输入到器件编程的全部功能。Quartus II是 Altera 公司推出的新一代 CPLD/FPGA 开发软件,适合于大规模的复杂逻辑电路设计,支持绝大部分 Altera 公司的 CPLD/FPGA,它是 Altera 公司的第 4 代可编程逻辑器件集成开发环境。同 MAX+PLUS II 相比,Quartus II 设计软件增加了网络编辑功能,提升了调试能力,解决了潜在的设计延迟,为其他 EDA 工具提供了方便的接口,可以在 Quartus II 集成环境中自动运行其他 EDA 工具。

本书从实用的角度出发,全面系统地介绍了 Altera 公司的可编程逻辑器件及 MAX+plus II 和 Quartus II 开发软件。全书包含 3 部分:第 1 部分介绍 CPLD/FPGA 的基本结构及 Altera 公司各个系列的 CPLD 和 FPGA,使读者掌握 CPLD/FPGA 的基本结构,在此基础上熟悉 Altera 公司各个系列的可编程逻辑器件的具体结构,从而加深对可编程逻辑器件的理解,便于在设计中很好地应用它们;第 2 部分通过几个实例全面系统地介绍了可编程逻辑器件的整个设计流程及 MAX+plus II 软件的功能特点,使读者能够理解并掌握可编程逻辑器件的开发流程,在此基础上熟练应用 MAX+plus II 软件;第 3 部分通过一个较复杂的实例详细的介绍了 Quartus II 软件的功能和使用方法,使读者能够熟练应用 Quartus II 软件。

本书内容丰富、图文并茂、讲解细致,可使读者快速、全面地掌握 Altera 公司的可编程逻辑器件及其开发技术。此外,本书的讲解由浅入深,第 1 部分是后面各章节的基础,如果不熟悉 Altera 公司的可编程逻辑器件,建议读者先阅读本部分。对于第 2 部分介绍的 MAX+plus II 和第 3 部分介绍的 Quartus II,笔者侧重于介绍 MAX+plus II,因为 MAX+plus II 是第 3 代软件,作为第 4 代的 Quartus II 继承了它的许多功能特点,读者理解并掌握了 MAX+plus II,便很容易过渡到 Quartus II,同时这也符合学习中由浅入深、循序渐进的特点。

需要说明的是,为了保证部分资料的原样,书中部分图形符号和文字符号并未按国家标准做统一修改,这点请读者注意。

本书由王辉、殷颖、陈婷和俞一鸣共同编写,书中包含着作者对数字系统的理解以及多年来进行可编程逻辑器件设计的经验。其中,陈婷编写了第 1 章,王辉和俞一鸣编写了第 2~7 章,殷颖编写了第 8~10 章,全书由王辉统稿。蒋建新、吴鹏、郭健、曹建军、许灵军、方华刚、葛树涛、赵海波、曹霖、刘磊、李晓凯、渠丰沛、张鹏、田雪、林丽君、秦冀、尤晓丽、张博、王志强、孙玉林、姜海燕等完成了书中资料的整理工作以及部分章节的编写工作,这里一并对他们的辛勤劳动表示衷心的感谢!另外,作者在编写本书的过程中参考了不少专家和学者的著作、学术论文和经验总结等,在此对他们表示最诚挚的谢意!

由于时间仓促,书中难免存在一些错误或不足之处,恳请广大读者和相关专家批评指正。

作者

可编程逻辑器件实用开发技术丛书

编 委 会

主 编：姜宇柏

副主编：张俊红 俞一鸣

编 委：李新新 潘天保 石 英 王 涛

王 冠 王 辉 吴钰淳 吴 鹏

尤晓丽 张海风 张 博 石 新

目 录

丛书序

前言

第 1 章 可编程逻辑器件.....1

1.1 专用集成电路.....1

1.2 PLD 发展概述.....2

1.3 CPLD/FPGA 的基本结构.....3

1.3.1 CPLD 的基本结构.....3

1.3.2 FPGA 的基本结构.....3

1.3.3 CPLD 和 FPGA 的异同.....4

1.4 Altera 公司 CPLD/FPGA 介绍.....5

1.4.1 Classic 系列.....5

1.4.2 MAX 系列.....7

1.4.3 FLEX 系列.....11

1.4.4 ACEX 系列.....16

1.4.5 Cyclone 系列.....18

1.4.6 APEX 系列.....20

1.4.7 Stratix 系列.....25

1.5 小结.....31

第 2 章 MAX+plus II 概述.....32

2.1 MAX+plus II 的特点.....32

2.2 MAX+plus II 的运行环境.....33

2.3 MAX+plus II 的安装和卸载.....34

2.4 MAX+plus II 的设计流程.....45

2.5 入门实例.....46

第 3 章 设计输入.....57

3.1 图形输入.....57

3.1.1 原理图设计.....58

3.1.2 符号编辑.....68

3.2 文本输入设计.....75

3.3 波形输入设计.....81

3.4 混合输入设计.....91

3.4.1 创建顶层图形设计文件.....91

3.4.2 保存并检查项目.....94

3.4.3 查看项目的层次结构.....94

3.5 设计输入规则.....96

第 4 章 设计进阶.....99

4.1 MAX+plus II 的环境设置.....99

4.2 MAX+plus II 的文件系统.....102

4.3 老式宏函数.....103

4.4 使用 LPM 参数化模块库.....112

4.4.1 使用 LPM 库模块.....112

4.4.2 使用 LPM 库模块的优点.....118

4.4.3 LPM 库模块列表.....118

4.5 使用 MegaWizard Plug-In Manager.....119

4.6 Altera 器件支持的 IP 核.....127

第 5 章 项目编译.....128

5.1 编译过程概述.....128

5.2 器件选择.....130

5.2.1 器件系列和芯片命名.....130

5.2.2 器件设置.....133

5.3 器件引脚分配.....140

5.3.1 平面图编辑器.....140

5.3.2 分配引脚.....144

5.4 其他的资源分配及设置选项.....151

5.4.1 Pin/Location/Chip 选项.....151

5.4.2 定时要求选项.....152

5.4.3 Clique 选项.....152

5.4.4 逻辑选项.....154

5.4.5 探针选项.....162

5.4.6 Connected Pins 选项.....163

5.4.7 Local Routing 选项.....163

5.5 全局项目设置.....163

5.5.1 全局项目器件选择项.....164

5.5.2 全局项目参数.....164

5.5.3 全局项目定时要求.....165

5.5.4 全局项目逻辑综合.....166

5.5.5 忽略项目设置.....167

5.5.6 清除项目设置.....168

5.5.7 反向标注	168	8.3.4 资源管理区	242
5.5.8 Convert Obsolete Assignment Format	168	8.3.5 工程工作区	242
5.5.9 编译顶层设计文件及查看编译 结果	169	8.3.6 编译状态显示窗	242
5.6 启用设计医生工具	173	8.3.7 信息显示窗	242
5.6.1 设计规则的设置	173	8.4 Quartus II 的设计流程	242
5.6.2 编译设计文件	175	8.4.1 图形用户界面设计	242
5.7 其他编译处理选项	177	8.4.2 命令行设计	244
5.7.1 功能仿真网表提取	177	第 9 章 Quartus II 设计实例	246
5.7.2 适配规则设置	178	9.1 实例设计说明	246
5.7.3 报告文件的设置及查看	179	9.1.1 FIFO 的功能	246
5.7.4 再编译方式的选择	184	9.1.2 FIFO 的结构	246
5.8 MAX+plus II 与其他 EDA 工具的 接口	184	9.1.3 FIFO 的实现方法	246
第 6 章 项目校验	186	9.1.4 模块设计规划	247
6.1 时序仿真	186	9.2 建立工程	249
6.1.1 底层模块单独仿真	186	9.2.1 相关说明	249
6.1.2 修改设计后再仿真	196	9.2.2 操作实例	250
6.1.3 顶层设计仿真	197	9.3 使用 Altera 宏功能	255
6.2 定时分析	203	9.3.1 宏功能模块概念	255
6.2.1 传播延迟分析	203	9.3.2 MegaWizard 管理器的使用	255
6.2.2 建立和保持时间分析	204	9.3.3 操作实例	255
6.2.3 时序逻辑电路性能分析	205	9.4 建立设计输入文件	260
第 7 章 器件编程与配置	207	9.4.1 设计输入方式	260
7.1 编程硬件	207	9.4.2 操作实例	260
7.2 编程或配置模式	208	9.5 分析综合	267
7.3 驱动程序安装	218	9.5.1 综合控制选项	267
7.4 编程或配置文件	223	9.5.2 操作实例	270
7.5 MAX+plus II 编程操作	225	9.6 布局布线	272
第 8 章 Quartus II 概述	230	9.6.1 布局布线参数选项	272
8.1 Quartus II 的特点	230	9.6.2 操作实例	275
8.2 Quartus II 的系统配置和安装	231	9.7 建立约束重编译	276
8.2.1 Quartus II 的系统配置	231	9.7.1 时序约束参数设置	276
8.2.2 Quartus II 的安装及卸载	231	9.7.2 Assignment Editor 工具的使用	279
8.3 Quartus II 界面概览	239	9.7.3 Pin Planner 工具的使用	282
8.3.1 标题栏	239	9.7.4 操作实例	283
8.3.2 菜单栏	239	9.8 编程及配置	285
8.3.3 工具栏	242	9.8.1 配置方式	285
		9.8.2 编程配置文件	286
		9.8.3 操作实例	290
		第 10 章 Quartus II 辅助工具	293

X

10.1 原理图观察工具·····	293	10.4 优化建议工具·····	310
10.1.1 RTL Viewer·····	293	10.4.1 Resource Optimization Advisor···	310
10.1.2 Technology Map Viewer·····	297	10.4.2 Timing Optimization Advisor···	312
10.2 Timing Closure Floorplan·····	298	10.5 调试工具·····	312
10.2.1 功能简介·····	298	10.5.1 SignalTap II Logic Analyzer···	312
10.2.2 界面说明·····	298	10.5.2 SignalProbe·····	314
10.3 底层编辑器 Chip Editor·····	305	10.5.3 In-System Memory Content Editor·····	316
10.3.1 功能简介·····	305	参考文献·····	318
10.3.2 界面视图说明·····	306		
10.3.3 资源特性编辑器·····	310		

第 1 章 可编程逻辑器件

在过去的 20 年, 专用集成电路 (Application Specific Integrated Circuit, ASIC) 和可编程逻辑器件 (PLD) 是电子设计的主流技术。其中, 可编程逻辑器件以 SRAM、Flash 存储器或 EEPROM 为存储基础, 由用户根据自己的需要对其进行编程, 确定芯片的功能, 然后应用在用户的电路系统中。随着半导体技术的进步, 可编程逻辑器件的规模、密度、性能有了飞速的发展, 使其广泛应用于集成电路设计、计算机、数字系统设计、通信系统、信号处理、工业自动化控制、仪器仪表等领域。

1.1 专用集成电路

通常采用芯片厂商提供的中小规模集成电路可以组成用户所需要的电路系统, 但是为了减小系统电路的体积、重量、功耗和提高可靠性, 芯片设计人员会把设计的系统直接做成一片大规模或超大规模集成电路, 这种为专门用途设计的集成电路就称为专用集成电路。

ASIC 按制造方法可分为全定制 (Full Custom) 产品, 半定制 (Semi Custom) 产品和可编程逻辑器件 (PLD)。

全定制芯片的各层掩膜都是按特定电路功能专门制造的。设计人员从晶体管的版图尺寸、位置和互连线开始设计, 以求达到芯片面积利用率高、速度快、功耗低的最优性能。要经过电路设计、逻辑模拟、版图设计和集成电路的各道生产工序, 才能制造出符合要求的专用集成电路芯片。它的设计制作成本高、周期长, 还带有较大的风险性, 因此全定制的专用集成电路只在特大批量生产的情况下才适用。

半定制芯片的功能也是专门针对某一应用的, 但是其构成和设计则具有某种程度的通用性, 即把某些已经设计好的单元电路按照设计的需要连接而成。可见, 半定制芯片的特点能够简化设计, 缩短设计周期, 提高芯片的成品率和降低成本, 因此它得到了广泛的应用。

以上两种 ASIC 的设计和制造都离不开芯片生产厂商, 用户拿到芯片后只能根据芯片的功能加以应用, 而不能用于自己特定的应用。随着集成电路的发展, 电子工程师们更愿意自己设计出合适的 ASIC 芯片, 而且在使用中还可以修改。可编程逻辑器件 (Programmable Logic Device, PLD) 就是为了满足这一需求而产生的。

PLD 芯片上的电路和金属引线都是事先由芯片生产厂商做好的, 但是其逻辑功能在出厂时并没有确定, 用户可根据需要通过使用 PLD 开发工具对其进行“编程”的办法来确定。可见, 电子工程师们通过对 PLD 编程就可以把设计的系统集成在一块芯片上, 而不再需要由厂商来设计制造专用集成电路, 这样便解决了专用集成电路的专用性和成本较高以及开发周期长的缺点。

由此可见, ASIC 的发展降低了产品的生产成本, 提高了系统的可靠性和灵活性, 缩短

了设计的周期，推动了技术的数字化进程。

1.2 PLD 发展概述

由于 PLD 的特点，使其得到了迅速发展。它从最初的可编程只读存储器 (Programmable ROM, PROM)，发展到可编程逻辑阵列 (Programmable Logic Array, PLA)、可编程阵列逻辑 (Programmable Array Logic, PAL)、通用阵列逻辑 (Generic Array Logic, GAL)、复杂可编程逻辑器件 (Complex Programmable Logic Device, CPLD) 以及现场可编程门阵列 (Field Programmable Gate Array, FPGA)。

可编程只读存储器 (PROM) 是 20 世纪 70 年代初期出现的第 1 代 PLD，它由全译码的与阵列和可编程的或阵列组成，用来实现以最小项形式表示的各种组合逻辑。PROM 采用熔丝工艺编程，只能写一次，不可擦除或重写。随着技术的发展和应用要求，又出现了 EPROM 和 E^2 PROM。由于阵列规模大，因此它们的主要用途还是用作存储器。

可编程逻辑阵列 (PLA) 是 20 世纪 70 年代中期推出的可编程逻辑器件，它是由可编程的与阵列和可编程的或阵列组成的一次性编程器件。虽然其阵列规模大为减小，提高了芯片的利用率，但由于编程复杂，支持 PLA 的开发软件有一定难度，因而也没有得到广泛应用。

可编程阵列逻辑 (PAL) 是 20 世纪 70 年代末期由 AMD 公司推出的 PLD，它具有多种形式的输出结构，在设计中有一定的灵活性，因而成为第 1 个得到普遍应用的 PLD。

通用阵列逻辑 (GAL) 是 Lattice 公司于 20 世纪 80 年代初开发的电可擦写、可重复编程、可设置加密位的 PLD。与 PAL 相比，GAL 增加了一个可编程的输出逻辑宏单元 (Output Logic Micro Cell, OLMC)，通过对 OLMC 配置可以得到多种形式的输出和反馈。在实际应用中，由于 GAL 对 PAL 具有完全的兼容性，所以 GAL 几乎完全代替了 PAL，并得到广泛应用。PAL 和 GAL 结构简单，易于编程，但规模小，难以实现复杂的逻辑功能。

可擦除的可编程逻辑器件 (EPLD) 是 Altera 公司于 20 世纪 80 年代中期推出的一种大规模可编程逻辑器件。EPLD 的基本结构与 GAL 并无本质区别，但其集成度比 GAL 高得多，使其在一块芯片内能够实现更多的逻辑功能。

复杂可编程逻辑器件 (CPLD) 是 20 世纪 90 年代初期出现的 EPLD 改进型器件。同 EPLD 相比，CPLD 增加了内部连线，改进了内部结构体系，因而比 EPLD 性能更好，设计更加灵活，其发展非常迅速。

现场可编程门阵列 (FPGA) 是 1985 年由美国 Xilinx 公司首先推出的一种新型的 PLD。FPGA 在结构上由逻辑功能模块排列成阵列，并由可编程的内部连线连接这些功能模块来实现一定的逻辑功能。FPGA 的功能由逻辑结构的配置数据决定，这些配置数据存放在片外的 EPROM 或其他存储器上。工作时，需要从芯片外部的存储器中加载配置数据存放到片内的 SRAM 上，在现场修改器件的逻辑功能。

由于 CPLD 和 FPGA 的规模大、集成度高，灵活性更强，实现的逻辑功能更广，因而，逐步成为复杂数字硬件电路设计的首选。下面就这两种 PLD 的结构原理进行详细介绍。

1.3 CPLD/FPGA 的基本结构

1.3.1 CPLD 的基本结构

CPLD 是由 GAL 发展起来的，其主体仍是基于乘积项的与或阵列。它的基本结构主要由可编程逻辑单元、可编程连线阵列和可编程输入/输出 (I/O) 单元组成，如图 1-1 所示。

1. 可编程逻辑单元

CPLD 的可编程逻辑单元被称为宏单元 (Macrocell)，主要包括与阵列、或阵列、触发器和多路选择器。其中，与或阵列和多路选择器用于实现组合逻辑功能，触发器用于实现时序逻辑功能。

2. 可编程连线阵列

CPLD 的可编程连线阵列采用集中式布线结构，用一个可编程的开关矩阵实现宏单元之间的互连。由于 CPLD 的连线阵列结构固定，所以 CPLD 的输入引脚到输出引脚的标准延时固定。

3. 可编程 I/O 单元

CPLD 的集成规模很大，每片中的宏单元多达数千，而其 I/O 引脚最多不超过数百。因此，在芯片中设置了可编程 I/O 单元，通过编程实现宏单元与 I/O 引脚的连接。同时，通过对 I/O 单元编程完成不同电气特性下对输入/输出信号的驱动和匹配。

除了上述基本结构外，CPLD 还配有以下辅助功能模块：JTAG (联合测试组) 编程模块，用于在系统编程；全局时钟、全局复位/置位、全局使能模块；有些型号还配置了片内 RAM、锁相环电路、高速计数器和高速乘法器模块。

1.3.2 FPGA 的基本结构

FPGA 的电路结构与 CPLD 不同，它基于查找表 (Look Up Table, LUT) 加寄存器的结构，由若干独立的可编程逻辑模块组成。由于这些模块的排列形式和门阵列 (Gate Array, GA) 形式相似，所以被称为现场可编程门阵列。

FPGA 的基本结构由可编程逻辑单元、可编程 I/O 单元和可编程连线资源组成，如图 1-2 所示。

1. 可编程逻辑单元

FPGA 的可编程逻辑单元一般由查找表和寄存器构成。查找表结构的核心是 $2^N \times 1$ 位

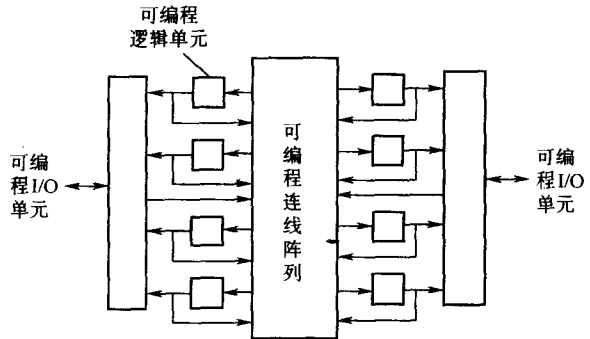


图 1-1 CPLD 的基本结构框图

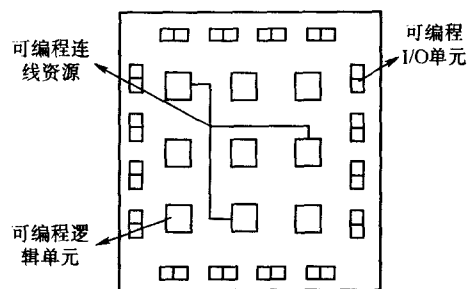


图 1-2 FPGA 的基本结构框图

SRAM, 用 $2^N \times 1$ 位 SRAM 存储一个 N 输入组合逻辑函数的真值表, 其输出即为所期望的组合逻辑; 可编程逻辑单元的寄存器可以配置为触发器或锁存器。因而, 用查找表和寄存器的组合可以实现任意组合逻辑和时序逻辑的设计。

2. 可编程连线资源

FPGA 的连线方式与 CPLD 存在很大的差异。如前所述, CPLD 的连线采用的是集中式布线方式, 连线资源非常有限。而 FPGA 配置了丰富的连线资源, 连线分布于 FPGA 内部所有单元。这些连线资源根据工艺、长度、宽度和分布位置的不同而被划分为不同的级别:

- 1) 全局性的专用连线资源: 实现器件内部的全局时钟和全局复位/置位的布线。
- 2) 长线连线资源: 实现器件分区之间的高速信号和第二全局时钟信号的布线。
- 3) 短线连线资源: 实现基本可编程逻辑单元之间的逻辑互连。

除上述连线资源以外, 在基本可编程逻辑单元内部还有各种各样的连线资源和控制信号线。

3. 可编程 I/O 单元

FPGA 的可编程 I/O 单元与 CPLD 的可编程 I/O 单元的功能相同, 用以实现可编程逻辑单元与 I/O 引脚的互连, 以及不同电气特性下对输入/输出信号的驱动和匹配。与 CPLD 相比, FPGA 支持更多的 I/O 标准。FPGA 的可编程 I/O 单元支持的电气连接有 PCI、LVTTTL、LVCMOS、LVDS、SSTL、CTT、HSTL 和 LVPECL 等。

除了上述构成 FPGA 基本结构的三种资源以外, 随着工艺的进步和应用系统需求的发展, 一般在 FPGA 中还可能包含以下可选资源: 存储器资源 (块 RAM、分布式 RAM); 数字时钟管理单元 (分频/倍频、数字延迟、时钟锁定); 算术运算单元 (高速硬件乘法器、乘加器、DSP); 多电平标准兼容的 I/O 接口; 高速串行 I/O 接口; 特殊功能模块 (以太网 MAC 等硬 IP 核); 微处理器 (PowerPC405 等硬处理器 IP 核)。

1.3.3 CPLD 和 FPGA 的异同

前面分别对 CPLD 和 FPGA 的基本结构进行了介绍。为了对它们各自的特点有更深地了解, 现就它们的异同进行归纳:

1) CPLD 大都基于乘积项结构, 实现工艺多为 E^2 CMOS, 也有 E^2 PROM、Flash 存储器、反熔丝工艺; FPGA 大都基于查找表加寄存器结构, 实现工艺多为 SRAM, 也有 Flash、反熔丝工艺。

2) FPGA 中触发器数量比 CPLD 中的丰富, 因而 CPLD 更适合完成各种组合逻辑电路, FPGA 更适合于完成时序逻辑电路。

3) CPLD 的集中式布线结构决定了它的时序延迟是均匀和可预测的, 而 FPGA 的分段式布线结构决定了其延迟的不可预测性。

4) CPLD 的速度比 FPGA 快。因为 FPGA 是门级编程, 并且逻辑单元模块之间采用分布式互连; 而 CPLD 是逻辑块级编程, 其逻辑块之间的互连是集总式的。

5) FPGA 的集成度比 CPLD 的高, 具有更复杂的连线结构和逻辑实现。

6) FPGA 比 CPLD 在编程上具有更大的灵活性。CPLD 通过修改具有固定内连电路的逻辑功能来编程, FPGA 主要通过改变内部连线的布线来编程; FPGA 可在逻辑门下编程, 而 CPLD 是在逻辑块下编程。在编程方式上, CPLD 主要是基于 E^2 PROM 或 Flash 存储器编程,

编程次数可达 1 万次，系统断电时编程信息也不丢失。CPLD 又可分为在编程器上编程和在系统编程两类。FPGA 大部分是基于 SRAM 编程，编程信息在系统断电时丢失，每次上电时，需从器件外部将编程数据重新写入 SRAM 中。FPGA 可以编程任意次，可在工作中快速编程，从而实现板级和系统级的动态配置。

7) CPLD 比 FPGA 使用起来更方便，保密性更强。CPLD 的编程采用 E²PROM 或快速 Flash 技术，无需外部存储器芯片，使用简单，而且还能进行加密处理；而 FPGA 的编程信息需存放在外部存储器上，使用方法复杂，而且不易实现加密处理，保密性差。

综合以上特点可以知道，CPLD 可以实现的功能比较单一，适合纯组合逻辑系统的设计。而在设计中包含了复杂的协议处理，或者设计中使用大量的时序元件时，一般选用 FPGA。

1.4 Altera 公司 CPLD/FPGA 介绍

Altera 公司是 20 世纪 90 年代以后发展起来的最大可编程逻辑器件供应商之一，其产品分为高密度 FPGA、低成本 FPGA 和 CPLD 三大类。高密度 FPGA 的主流产品为 APEX 系列和 Stratix 系列；低成本 FPGA 的主要代表为 FLEX 系列、ACEX 系列和 Cyclone 系列；CPLD 的主要代表为 Classic 系列和 MAX 系列。下面分别就其系列结构特点进行介绍。

1.4.1 Classic 系列

Classic 系列器件是基于 EPROM 工艺的 PLD，集成度规模达 300~900 可用门，工作频率可达 100MHz。该系列具有“零功率”模式，在该模式下维持工作状态的电流只需微安级，非常适合低功耗设计的应用。Classic 系列产品的主要特性见表 1-1。

表 1-1 Classic 系列产品的主要特性

特 性	EP610	EP910	EP1810
	EP6101	EP9101	
可用门数	300	450	900
宏单元数	16	24	48
最大用户 I/O 引脚数	22	38	64
引脚延时/ns	10	12	20
工作频率/MHz	100	76.9	50

Classic 系列体系结构包括宏单元、可编程寄存器、输出使能/时钟选择和反馈选择。

1. 宏单元

Classic 系列宏单元如图 1-3 所示，它能够被单独作为时序电路或组合逻辑电路来使用。

每一个宏单元的可编程与阵列有 8 个乘积项，它们通过或门实现“与或式”表达的组合适逻辑函数。

宏单元还提供了两个单独的乘积项：一个乘积项用来作为内部寄存器的异步清零控制，另外一个乘积项用来实现输出使能或作为逻辑阵列产生的时钟。

可编程与阵列的输入信号可以由专用输入引脚提供，也可以由配置为输入的 I/O 引脚的反馈信号提供，还可以由宏单元的输出反馈信号提供。由专用输入引脚提供的信号分布到了

整个芯片内部，因此可以直接作为片上所有宏单元的输入；而由宏单元和 I/O 引脚反馈提供的信号则需通过反馈多路复用器控制其路由。

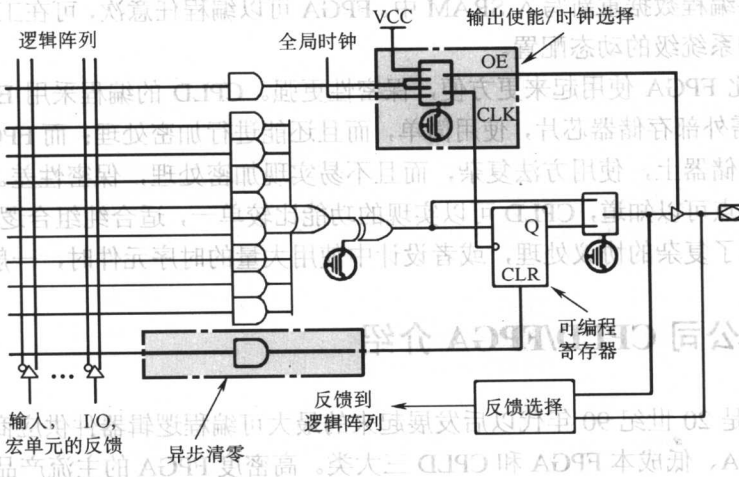


图 1-3 Classic 系列宏单元

2. 可编程寄存器

对于寄存器功能，每一个宏单元的可编程寄存器都能够被单独编程为 D、T、JK 或 RS 触发器。在实现组合逻辑的情况下，该可编程寄存器还可以被旁路掉。可编程寄存器有指定的乘积项来实现异步清零功能。在上电时，它们还会自动清零。另外，宏单元的寄存器可以独立地由全局时钟或任意输入引脚或与阵列反馈通路的信号进行触发。

3. 输出使能/时钟选择

输出使能/时钟选择有两种操作模式：模式 0 和模式 1，如图 1-4 所示。

在模式 0 的情况下，由一个单独的乘积项控制三态输出缓冲器。如果输出使能为高电平，则使能输出缓冲器；如果输出使能为低电平，输出呈现高阻状态。在该模式下，宏单元的触发器的时钟信号由它自身的全局时钟输入信号提供；

在模式 1 的情况下，输出使能缓冲器一直使能，宏单元的寄存器的时钟信号由阵列时钟信号通过乘积项提

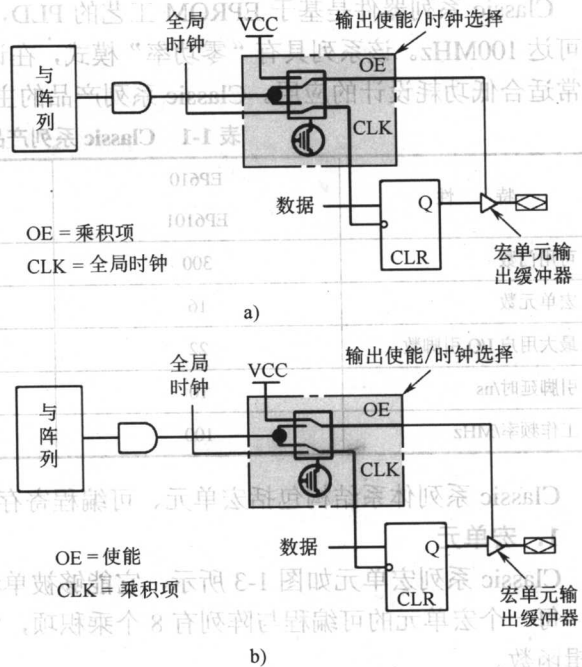


图 1-4 Classic 系列输出使能/时钟选择示意图

a) 模式 0 (寄存器时钟信号为全局时钟信号，输出由乘积项使能)

b) 模式 1 (输出一直使能，寄存器的时钟信号由乘积项提供，

因此可以产生门控时钟)