

ELECTRONIC  
ENGINEER

XIDIAN UNIVERSITY PRESS

Design and Application of FPGA

# FPGA 设计及应用

褚振勇 翁木云 编著

1.2

*Design and Application of FPGA*



西安电子科技大学出版社

<http://www.xduph.com>

**Design and Application of FPGA**

# **FPGA 设计及应用**

褚振勇 翁木云 编著

西安电子科技大学出版社

## 内 容 简 介

本书系统介绍了有关可编程逻辑器件的基本知识以及相关软件的使用方法,讲述了FPGA电路设计的方法和技巧,并给出了设计实例。本书主要内容包括:可编程逻辑器件概述;Altera可编程逻辑器件;Altera可编程逻辑器件开发软件;第三方工具软件;Altera器件编程与配置;FPGA设计中的基本问题;MAX+PLUSⅡ开发软件中的宏模块及其应用;FPGA电路设计实例;FPGA高端开发技术。

本书内容全面,取材新颖,叙述清楚,理论联系实际,使用大量图表说明问题,便于读者对内容的理解和掌握。为方便读者,本书附一张光盘,其中包含所有设计实例的源程序和Altera公司全线产品的技术资料及开发软件。

本书既可用作高等工科院校电子类专业高年级本科生和研究生的教材,又可作为电子类工程技术人员的参考书。

## 图书在版编目(CIP)数据

**FPGA 设计及应用/褚振勇,翁木云编著.**

—西安:西安电子科技大学出版社,2002.7

ISBN 7-5606-1132-X

I. F… II. ①褚… ②翁… III. 可编程逻辑器件-基本知识 IV. TP211

中国版本图书馆CIP数据核字(2002)第025012号

责任编辑 臧延新

出版发行 西安电子科技大学出版社(西安市太白南路2号)

电 话 (029)8227828

邮 编 710071

http://www.xduph.com

E-mail: xdupfxb@pub.xaonline.com

经 销 新华书店

印 刷 西安兰翔印刷厂

版 次 2002年7月第1版 2003年1月第2次印刷

开 本 787毫米×1092毫米 1/16 印张 20.5

字 数 487千字

印 数 4 001~10 000册

定 价 30.00元(含光盘)

ISBN 7-5606-1132-X/TP·0574

**XDUP 1403001-2**

\*\*\* 如有印装问题可调换 \*\*\*

# 前 言

现场可编程门阵列(FPGA, Field Programmable Gate Array)的出现是超大规模集成电路(VLSI)技术和计算机辅助设计(CAD)技术发展的结果。FPGA 器件集成度高、体积小,具有通过用户编程实现专门应用的功能。它允许电路设计者利用基于计算机的开发平台,经过设计输入、仿真、测试和校验,直到达到预期的结果。使用 FPGA 器件可以大大缩短系统的研制周期,减少资金投入。更吸引人的是,采用 FPGA 器件可以将原来的电路板级产品集成为芯片级产品,从而降低了功耗,提高了可靠性,同时还可以很方便地对设计进行在线修改。FPGA 器件成为研制开发的理想器件,特别适合于产品的样机开发和小批量生产,因此有时人们也把 FPGA 称为可编程的 ASIC。

近年来, FPGA 市场发展十分迅速, 各大 FPGA 厂商不断采用新技术来提高 FPGA 器件的容量, 增强软件的性能。如今, FPGA 器件广泛应用于通信、自动控制、信息处理等诸多领域, 越来越多的电子设计人员在使用 FPGA, 熟练掌握 FPGA 设计技术已经是对电子设计工程师的基本要求。

本书的作者较早接触 FPGA 技术, 并将其应用于科研和工程实践, 深切感受到 FPGA 技术是数字电路设计的利器, 从中受益颇深。但是, 目前专门讲述 FPGA 设计及应用方面的书籍还很少, 更多情况下需要设计人员直接查找和阅读英文资料, 以获取所需信息。同时, FPGA 设计技术是一项实践性非常强的专业技术, 需要一定的经验积累, 这就给许多迫切需要了解和掌握 FPGA 设计技术的人员带来一定的困难。为了能使更多的人尽快掌握 FPGA 设计技术, 并能应用于实际, 作者在阅读和参考大量中英文资料的基础上, 结合科研和工程实践经验, 最终写成此书。

全书系统地介绍了有关可编程逻辑器件的基本知识以及相关软件的使用方法。着重讲述了 FPGA 电路设计的方法和技巧, 并给出了设计实例。本书共分为 9 章。第 1 章分析了可编程逻辑器件的结构特点、基本设计方法和设计流程。第 2 章重点介绍了 Altera 公司各个系列器件的特点、结构及功能。第 3 章详细介绍了 Altera 可编程逻辑器件开发软件 MAX+PLUS II 和 Quartus II 的安装和使用方法。第 4 章主要介绍第三方工具软件, 特别是目前较为常用的 FPGA Express 和 LeonardoSpectrum 软件的特点、设计流程及使用方法。第 5 章介绍了 Altera 公司 PLD 器件的命名方法, 以及 PLD 器件的多种编程和配置方式。第 6 章详细介绍了 FPGA 设计时需要注意的一些基本问题, 例如数的表示方法、时钟、逻辑竞争和冒险、信号的时延等问题。第 7 章主要讲述 MAX+PLUS II 开发软件中的宏模块及其应用, 详细介绍了时序电路、运算电路和存储器三大类宏模块的组成和使用方法。第 8 章给出了几个 FPGA 电路设计实例, 这些设计实例均来源于科研实践和工程设计项目, 其中包括序列产生器、数字相关器、汉明距离的电路计算、交织编译码器、直接数字频率合成器等, 本章最后以误码率在线测试电路为例, 给出了一个完整的 FPGA 设计。第 9 章涉及 FPGA 高端开发技术, 主要包括可再配置计算、可编程单芯片系统(SOPC)以及 IP 模块。本书的三个附录分别给出了 MAX+PLUS II 文件的后缀、相关网址检索和光盘文件索引。

本书所附的光盘上主要包含第 8 章设计实例的源程序以及 Altera 公司全线产品的技术

资料和 MAX+PLUS II 10.1 基本版、Quartus II 2.0 Web 版等开发软件。

FPGA 技术发展日新月异,新技术、新方法、新器件层出不穷。本书在撰写时虽加入了目前最新的资料,但是读者在阅读本书时,可能又出现了更新的器件。所以本书主要是向大家提供有关 FPGA 设计与应用方面较为基础的内容,读者可以从 EDA 厂商的网站上获取更新的资料,也可以从销售商或可编程逻辑器件中文网站(<http://www.fpga.com.cn> & <http://www.pld.com.cn>)上获取相关的信息和技术支持。

在本书的撰写过程中,得到了空军工程大学电讯工程学院和西安电子科技大学综合业务网理论及关键技术国家重点实验室的领导、老师和同事们的支持,特别是西安电子科技大学易克初教授,空军工程大学电讯工程学院谢德芳教授、黄国策副教授、梁俊副教授以及西北工业大学何明一教授给予本书大力的支持与帮助。西安电子科技大学田斌副教授、田红心讲师和王凯东博士生为本书提出了很好的改进意见。骏龙科技公司上海办事处的胡晟工程师,西安办事处的董祥、杨晓云工程师和可编程逻辑器件中文网站为本书提供了许多技术资料和技术支持。西安电子科技大学出版社的臧延新编辑和有关工作人员也为本书的出版付出了艰辛的劳动。书中还参考和引用了许多专家和学者的著作及研究成果。在此向上面提到的所有人员表示衷心的感谢。

本书的第 1~4 章、第 5 章的 5.3~5.7 节以及第 8 章的 8.7 节主要由翁木云撰写,第 6~9 章、第 5 章的 5.1~5.2 节以及附录主要由褚振勇撰写,第 6 章的 6.3 节由胡晟工程师整理提供。本书所附光盘由褚振勇整理。空军工程大学电讯工程学院外语室的高楷娟老师完成了本书大量的英文翻译和校对工作。全书最后由褚振勇统稿。

由于作者水平有限,书中难免有不妥甚至错误之处,欢迎各位读者和同行批评指正。

作 者

2002 年 5 月于西安

# 目 录

|                                   |    |                                |     |
|-----------------------------------|----|--------------------------------|-----|
| <b>第 1 章 可编程逻辑器件概述</b> .....      | 1  | 3.1.1 设计软件流程图 .....            | 75  |
| 1.1 EDA 和 PLD 发展概况 .....          | 1  | 3.1.2 MAX+PLUS II 开发软件简介 ..... | 76  |
| 1.1.1 EDA 技术发展概况 .....            | 1  | 3.1.3 Quartus II 开发软件简介 .....  | 77  |
| 1.1.2 PLD 的发展概况 .....             | 3  | 3.1.4 多平台及其它 EDA 工具 .....      | 77  |
| 1.2 可编程逻辑器件的基本结构 .....            | 6  | 3.2 MAX+PLUS II 开发软件 .....     | 78  |
| 1.2.1 简单 PLD 的基本结构 .....          | 6  | 3.2.1 MAX+PLUS II 的安装 .....    | 78  |
| 1.2.2 EPLD 和 CPLD 的基本结构 .....     | 10 | 3.2.2 操作环境 .....               | 80  |
| 1.2.3 FPGA 的基本结构 .....            | 15 | 3.2.3 设计输入 .....               | 81  |
| 1.3 可编程逻辑器件的设计 .....              | 19 | 3.2.3.1 图形设计输入方法 .....         | 81  |
| 1.3.1 基本设计方法 .....                | 19 | 3.2.3.2 文本设计输入方法 .....         | 90  |
| 1.3.2 设计流程 .....                  | 22 | 3.2.3.3 创建顶层设计文件 .....         | 91  |
| <b>第 2 章 Altera 可编程逻辑器件</b> ..... | 25 | 3.2.3.4 层次显示 .....             | 92  |
| 2.1 Altera 产品概述 .....             | 25 | 3.2.3.5 编辑用户库 .....            | 93  |
| 2.1.1 Altera PLD 的特点 .....        | 25 | 3.2.4 设计项目的编译 .....            | 93  |
| 2.1.2 Altera 器件系列 .....           | 26 | 3.2.4.1 编译前准备 .....            | 93  |
| 2.2 FLEX 系列 .....                 | 28 | 3.2.4.2 运行编译器 .....            | 94  |
| 2.2.1 FLEX10K 系列 .....            | 28 | 3.2.4.3 在底层平面图编辑器中             |     |
| 2.2.2 ACEX1K 系列 .....             | 43 | 观察适配结果 .....                   | 100 |
| 2.2.3 FLEX8000 系列 .....           | 43 | 3.2.4.4 引脚锁定 .....             | 101 |
| 2.2.4 FLEX6000 系列 .....           | 45 | 3.2.5 模拟仿真和定时分析 .....          | 102 |
| 2.3 MAX 系列 .....                  | 46 | 3.2.6 器件编程 .....               | 109 |
| 2.3.1 MAX9000 系列 .....            | 46 | 3.3 Quartus II 开发软件 .....      | 111 |
| 2.3.2 MAX7000 系列 .....            | 52 | 3.3.1 Quartus II 的安装 .....     | 111 |
| 2.3.3 MAX5000 系列 .....            | 52 | 3.3.2 设计输入 .....               | 113 |
| 2.3.4 MAX3000A 系列 .....           | 52 | 3.3.3 设计项目的编译 .....            | 124 |
| 2.3.5 Classic 系列 .....            | 53 | 3.3.3.1 编译设置 .....             | 124 |
| 2.4 APEX 系列 .....                 | 53 | 3.3.3.2 资源分配 .....             | 127 |
| 2.4.1 APEX20K 系列 .....            | 53 | 3.3.3.3 编译设计 .....             | 127 |
| 2.4.2 APEX II 系列 .....            | 61 | 3.3.3.4 在最后的编译平面图中             |     |
| 2.5 Mercury 系列 .....              | 63 | 查看适配结果 .....                   | 129 |
| 2.6 Excalibur 系列 .....            | 66 | 3.3.3.5 分配逻辑到 ESB .....        | 129 |
| 2.7 Stratix 系列 .....              | 68 | 3.3.4 延时分析和仿真 .....            | 131 |
| 2.8 其它 PLD 公司及其产品简介 .....         | 69 | 3.3.4.1 Quartus II 的延时分析 ..... | 131 |
| 2.8.1 其它 PLD 公司简介 .....           | 69 | 3.3.4.2 仿真 .....               | 132 |
| 2.8.2 Xilinx 公司产品 .....           | 70 | 3.3.5 器件编程 .....               | 139 |
| 2.8.3 Lattice-Vantis 公司产品 .....   | 73 | 3.3.5.1 打开下载窗口 .....           | 139 |
| <b>第 3 章 Altera 可编程逻辑器件</b>       |    | 3.3.5.2 设置下载电缆 .....           | 139 |
| 开发软件 .....                        | 75 | 3.3.5.3 开始编程 .....             | 139 |
| 3.1 概述 .....                      | 75 |                                |     |

|                                         |     |                                                 |     |
|-----------------------------------------|-----|-------------------------------------------------|-----|
| 3.3.6 Quartus I 2.0版设计软件<br>的新特点 .....  | 140 | 6.1.4 带符号小数的二进制补码 .....                         | 199 |
| <b>第4章 第三方工具软件</b> .....                | 142 | 6.1.5 格雷码 .....                                 | 200 |
| 4.1 概述 .....                            | 142 | 6.1.6 带符号整数 .....                               | 201 |
| 4.1.1 HDL 语言 .....                      | 142 | 6.1.7 偏移二进制补码 .....                             | 201 |
| 4.1.2 常用的第三方工具软件 .....                  | 143 | 6.1.8 浮点数和块浮点数 .....                            | 201 |
| 4.2 FPGA Express 开发工具软件 .....           | 145 | 6.2 有限字长的影响 .....                               | 202 |
| 4.2.1 特点 .....                          | 145 | 6.3 时钟问题 .....                                  | 202 |
| 4.2.2 设计流程 .....                        | 145 | 6.3.1 全局时钟 .....                                | 202 |
| 4.2.3 设计分析 .....                        | 154 | 6.3.2 门控时钟 .....                                | 203 |
| 4.2.4 FPGA 脚本工具 FST .....               | 156 | 6.3.3 多级逻辑时钟 .....                              | 206 |
| 4.2.5 提示与技巧 .....                       | 158 | 6.3.4 行波时钟 .....                                | 207 |
| 4.3 Leonardo Spectrum 开发工具软件 .....      | 160 | 6.3.5 多时钟系统 .....                               | 208 |
| <b>第5章 Altera 器件编程与配置</b> .....         | 165 | 6.4 建立和保持时间 .....                               | 210 |
| 5.1 Altera 器件的命名 .....                  | 165 | 6.5 冒险现象 .....                                  | 210 |
| 5.2 PLD 器件测试电路板 .....                   | 166 | 6.6 清零和置位信号 .....                               | 214 |
| 5.3 ByteBlaster 并口下载电缆 .....            | 167 | 6.7 信号的延时 .....                                 | 214 |
| 5.3.1 原理与连接 .....                       | 167 | 6.8 器件结构与实际系统的匹配 .....                          | 217 |
| 5.3.2 被动串行(PS)模式 .....                  | 171 | 6.9 电路结构与器件速度和<br>成本之间的关系 .....                 | 217 |
| 5.3.3 JTAG 模式 .....                     | 171 | 6.10 器件加密 .....                                 | 217 |
| 5.3.4 软件编程和配置步骤 .....                   | 176 | 6.11 设计文档 .....                                 | 218 |
| 5.4 ByteBlasterMV 并口下载电缆 .....          | 176 | <b>第7章 MAX+PLUS II 开发软件中的<br/>宏模块及其应用</b> ..... | 219 |
| 5.4.1 特点 .....                          | 176 | 7.1 时序电路宏模块 .....                               | 219 |
| 5.4.2 功能描述 .....                        | 177 | 7.1.1 触发器 .....                                 | 219 |
| 5.4.3 软件编程和配置步骤 .....                   | 179 | 7.1.2 锁存器 .....                                 | 222 |
| 5.5 MasterBlaster 串行/<br>USB 通信电缆 ..... | 180 | 7.1.3 计数器 .....                                 | 223 |
| 5.5.1 特点 .....                          | 180 | 7.1.4 分频器 .....                                 | 228 |
| 5.5.2 功能描述 .....                        | 180 | 7.1.5 多路复用器 .....                               | 231 |
| 5.5.3 被动串行(PS)模式 .....                  | 183 | 7.1.6 移位寄存器 .....                               | 234 |
| 5.5.4 JTAG 模式 .....                     | 184 | 7.2 运算电路宏模块 .....                               | 238 |
| 5.6 BitBlaster 串行下载电缆 .....             | 187 | 7.2.1 加法和减法器 .....                              | 238 |
| 5.6.1 特点 .....                          | 187 | 7.2.2 乘法器 .....                                 | 242 |
| 5.6.2 功能描述 .....                        | 188 | 7.2.3 除法器 .....                                 | 244 |
| 5.7 FPGA 的芯片配置 .....                    | 189 | 7.2.4 绝对值运算 .....                               | 246 |
| 5.7.1 对单个器件的配置 .....                    | 191 | 7.2.5 数值比较器 .....                               | 247 |
| 5.7.2 对多个器件的配置 .....                    | 192 | 7.2.6 编码器和译码器 .....                             | 249 |
| <b>第6章 FPGA 设计中的基本问题</b> .....          | 195 | 7.2.7 奇偶校验器 .....                               | 251 |
| 6.1 数的表示方法 .....                        | 195 | 7.3 存储器宏模块 .....                                | 253 |
| 6.1.1 无符号整数 .....                       | 196 | 7.3.1 RAM 宏模块 .....                             | 253 |
| 6.1.2 二进制补码 .....                       | 197 | 7.3.2 FIFO 宏模块 .....                            | 256 |
| 6.1.3 无符号小数 .....                       | 198 | 7.3.3 ROM 的设计 .....                             | 259 |

|                                |     |                                         |     |
|--------------------------------|-----|-----------------------------------------|-----|
| 7.3.4 存储器设计中应注意的<br>一个问题 ..... | 263 | 8.7.2 误码检测电路的波形仿真 .....                 | 294 |
| <b>第8章 FPGA 电路设计实例</b> .....   | 264 | 8.7.3 误码检测电路的测试及实现 .....                | 297 |
| 8.1 m 序列产生器 .....              | 264 | 8.7.4 误码率在线测试电路的设计 .....                | 299 |
| 8.2 任意序列产生器 .....              | 267 | 8.7.5 硬件电路的调试与实现 .....                  | 301 |
| 8.3 数字相关器 .....                | 270 | <b>第9章 FPGA 高端开发技术</b> .....            | 303 |
| 8.4 汉明距离的电路计算 .....            | 273 | 9.1 可再配置计算 .....                        | 303 |
| 8.4.1 计数法 .....                | 274 | 9.2 可编程单芯片系统 .....                      | 304 |
| 8.4.2 逻辑函数法 .....              | 274 | 9.3 IP 模块 .....                         | 306 |
| 8.4.3 查找表法 .....               | 275 | 9.3.1 IP 模块的定义 .....                    | 306 |
| 8.4.4 求和网络法 .....              | 275 | 9.3.2 IP 模块的分类 .....                    | 307 |
| 8.4.5 组合应用 .....               | 276 | 9.3.3 Altera 公司的 IP 模块及<br>其使用流程 .....  | 307 |
| 8.5 交织编码器 .....                | 280 | 9.3.4 OpenCore 的安装 .....                | 309 |
| 8.5.1 交织编码的原理 .....            | 280 | 9.4 开发具有自主知识产权的 IP 模块 .....             | 309 |
| 8.5.2 利用移位寄存器实现<br>交织编码 .....  | 281 | <b>附录 A MAX+PLUS I 文件的<br/>后缀</b> ..... | 311 |
| 8.5.3 利用存储器实现交织编码 .....        | 283 | <b>附录 B 相关网址检索</b> .....                | 313 |
| 8.6 直接数字频率合成 .....             | 288 | <b>附录 C 光盘文件索引</b> .....                | 315 |
| 8.7 误码率在线测试 .....              | 293 | <b>参考文献</b> .....                       | 320 |
| 8.7.1 误码检测电路的设计 .....          | 293 |                                         |     |

# 第 1 章 可编程逻辑器件概述

电子设计自动化 EDA 和可编程逻辑器件 PLD 的发展都非常迅速。本章首先对 EDA 技术发展和 PLD 的发展历程进行介绍,然后详细分析从简单 PLD 到 CPLD/FPGA 的结构特点,并对 PLD 的基本设计方法和设计流程进行介绍,使读者对可编程逻辑器件的原理、特点、结构、功能及开发应用有一个总体概念。

## 1.1 EDA 和 PLD 发展概况

### 1.1.1 EDA 技术发展概况

电子设计自动化(EDA, Electronic Design Automation)是指利用计算机完成电子系统的设计。EDA 技术是以计算机和微电子技术为先导,汇集了计算机图形学、拓扑、逻辑学、微电子工艺与结构学和计算数学等多种计算机应用学科最新成果的先进技术。

EDA 技术以计算机为工具,代替人完成数字系统的逻辑综合、布局布线和设计仿真等工作。设计人员只需要完成对系统功能的描述,就可以由计算机软件进行处理,得到设计结果,而且修改设计如同修改软件一样方便,可以极大地提高设计效率。

从 20 世纪 60 年代中期开始,人们就不断开发出各种计算机辅助设计工具来帮助设计人员进行电子系统的设计。电路理论和半导体工艺水平的提高,对 EDA 技术的发展起了巨大的推进作用,使 EDA 作用范围从 PCB 板设计延伸到电子线路和集成电路设计,直至整个系统的设计,也使 IC 芯片系统应用、电路制作和整个电子系统生产过程都集成在一个环境之中。根据电子设计技术的发展特征,EDA 技术发展大致分为三个阶段。

#### 1. CAD 阶段(20 世纪 60 年代中期~20 世纪 80 年代初期)

第一阶段的特点是一些单独的工具软件,主要有 PCB(Printed Circuit Board)布线设计、电路模拟、逻辑模拟及版图的绘制等,通过计算机的使用,从而将设计人员从大量繁琐重复的计算和绘图工作中解脱出来。例如,目前常用的 Protel 早期版本 Tango,以及用于电路模拟的 SPICE 软件和后来产品化的 IC 版图编辑与设计规则检查系统等软件,都是这个阶段的产品。这个时期的 EDA 一般称为 CAD(Computer Aided Design)。

20 世纪 80 年代初,随着集成电路规模的增大,EDA 技术有了较快的发展。许多软件公司如 Mentor, Daisy System 及 Logic System 等进入市场,开始供应带电路图编辑工具和逻辑模拟工具的 EDA 软件。这个时期的软件主要针对产品开发,按照设计、分析、生产和测试等多个阶段,不同阶段分别使用不同的软件包,每个软件只能完成其中的一项工作,通过顺序循环使用这些软件,可完成设计的全过程。但这样的设计过程存在两个方面的问题:第一,由于各个工具软件是由不同的公司和专家开发的,只解决一个领域的问题,若将一个工具软件的输出作为另一个工具软件的输入,就需要人工处理,过程很繁琐,影响

了设计速度；第二，对于复杂电子系统的设计，当时的 EDA 工具由于缺乏系统级的设计考虑，不能提供系统级的仿真与综合，设计错误如果在开发后期才被发现，将给修改工作带来极大不便。

## 2. CAE 阶段(20 世纪 80 年代初期~20 世纪 90 年代初期)

这个阶段在集成电路与电子设计方法学以及设计工具集成化方面取得了许多成果。各种设计工具，如原理图输入、编译与连接、逻辑模拟、测试码生成、版图自动布局以及各种单元库已齐全。由于采用了统一数据管理技术，因而能够将各个工具集成为一个 CAE (Computer Aided Engineering) 系统。按照设计方法学制定的设计流程，可以实现从设计输入到版图输出的全程设计自动化。这个阶段主要采用基于单元库的半定制设计方法，采用门阵列和标准单元设计的各种 ASIC 得到了极大的发展，将集成电路工业推入了 ASIC 时代。多数系统中集成了 PCB 自动布局布线软件以及热特性、噪声、可靠性等分析软件，进而可以实现电子系统设计自动化。

## 3. EDA 阶段(20 世纪 90 年代以来)

20 世纪 90 年代以来，微电子技术以惊人的速度发展，其工艺水平达到深亚微米级，在一个芯片上可集成数百万乃至上千万只晶体管，工作速度可达到 GHz，这为制造出规模更大，速度更快和信息容量很大的芯片系统提供了条件，但同时也对 EDA 系统提出了更高的要求，并促进了 EDA 技术的发展。此阶段主要出现了以高级语言描述、系统仿真和综合技术为特征的第三代 EDA 技术，不仅极大地提高了系统的设计效率，而且使设计人员摆脱了大量的辅助性及基础性工作，将精力集中于创造性的方案与概念的构思上。下面简单介绍这个阶段 EDA 技术的主要特征。

(1) 高层综合(HLS, High Level Synthesis)的理论与方法取得较大进展，将 EDA 设计层次由 RT 级提高到了系统级(又称行为级)，并划分为逻辑综合和测试综合。逻辑综合就是对不同层次和不同形式的设计描述进行转换，通过综合算法，以具体的工艺背景实现高层目标所规定的优化设计，通过设计综合工具，可将电子系统的高层行为描述转换到低层硬件描述和确定的物理实现，使设计人员无须直接面对低层电路，不必了解具体的逻辑器件，从而把精力集中到系统行为建模和算法设计上。测试综合是以设计结果的性能为目标的综合方法，以电路的时序、功耗、电磁辐射和负载能力等性能指标为综合对象。测试综合是保证电子系统设计结果稳定可靠工作的必要条件，也是对设计进行验证的有效方法，其典型工具有 Synopsys 公司的 Behavioral Compiler 以及 Mentor Graphics 公司的 Monet 和 Renoir。

(2) 采用硬件描述语言 HDL(Hardware Description Language)来描述 10 万门以上的设计，并形成了 VHDL(Very High Speed Integrated Circuit HDL)和 Verilog HDL 两种标准硬件描述语言。它们均支持不同层次的描述，使得复杂 IC 的描述规范化，便于传递、交流、保存与修改，也便于重复使用。它们多应用于 FPGA/CPLD/EPLD 的设计中。大多数 EDA 软件都兼容这两种标准。

(3) 采用平面规划(Floorplaning)技术对逻辑综合和物理版图设计进行联合管理，做到在逻辑综合早期设计阶段就考虑到物理设计信息的影响。通过这些信息，设计者能更进一步进行综合与优化，并保证所作的修改只会提高性能而不会对版图设计带来负面影响。这在深亚微米级布线延时已成为主要延时的情况下，加速设计过程的收敛与成功是有所帮助

的。在 Synopsys 和 Cadence 等公司的 EDA 系统中均采用了这项技术。

(4) 可测性综合设计。随着 ASIC 的规模与复杂性的增加, 测试难度与费用急剧上升, 由此产生了将可测性电路结构制作在 ASIC 芯片上的想法, 于是开发了扫描插入、BLST (内建自测试)、边界扫描等可测性设计(DFT)工具, 并已集成到 EDA 系统中。其典型产品有 Compass 公司的 Test Assistant 和 Mentor Graphics 公司的 LBLST Architect、BSD Architect、DFT Advisor 等。

(5) 为带有嵌入 IP 模块(IP 模)的 ASIC 设计提供软硬件协同系统设计工具。协同验证弥补了硬件设计和软件设计流程之间的空隙, 保证了软硬件之间的同步协调工作。协同验证是当今系统集成的核心, 它以高层系统设计为主导, 以性能优化为目标, 融合逻辑综合、性能仿真、形式验证和可测性设计, 产品如 Mentor Graphics 公司的 Seamless CAV。

(6) 建立并行设计工程 CE(Concurrent Engineering)框架结构的集成化设计环境, 以适应当今 ASIC 的如下一些特点: 数字与模拟电路并存, 硬件与软件设计并存, 产品上市速度要快。在这种集成化设计环境中, 使用统一的数据管理系统与完善的通讯管理系统, 由若干相关的设计小组共享数据库和知识库, 并行地进行设计, 而且在各种平台之间可以平滑过渡。

全球 EDA 厂商有近百家之多, 大体可分两类: 一类是 EDA 专业软件公司, 较著名的有 Mentor Graphics、Cadence Design Systems、Synopsys、Viewlogic Systems 和 Protel 等; 另一类是半导体器件厂商, 为了销售他们的产品而开发 EDA 工具, 较著名的公司有 Altera、Xilinx、AMD、TI 和 Lattice 等。EDA 专业软件公司独立于半导体器件厂商, 推出的 EDA 系统具有较好的标准化和兼容性, 也比较注意追求技术上的先进性, 适合于搞学术性基础研究的单位使用。而半导体厂商开发的 EDA 工具, 能针对自己器件的工艺特点作出优化设计, 提高资源利用率, 降低功耗, 改善性能, 比较适合于产品开发单位使用。在 EDA 技术发展策略上, EDA 专业软件公司面向应用, 提供 IP 模块和相应的设计服务; 而半导体厂商则采取三位一体的战略, 在器件生产、设计服务和 IP 模块的提供上下工夫。

### 1.1.2 PLD 的发展概况

当今社会是数字化社会, 数字集成电路应用非常广泛, 其发展从电子管、晶体管、小规模集成 SSI、MSI、LSI、VLSI(几万门以上)到超大规模集成电路 ULSI 和超位集成电路 GSI, 其规模几乎平均每 1~2 年翻一番。集成电路的发展大大促进了 EDA 的发展, 先进的 EDA 已从传统的“自下而上”的设计方法改变为“自上而下”的设计方法。ASIC(Application Specific Integrated Circuit)的设计与制造, 已不再完全由半导体厂商独立承担, 系统设计师在实验室里就可以设计出合适的 ASIC 芯片, 并且立即投入实际应用之中, 这都得益于可编程逻辑器件 PLD(Programmable Logic Device)的出现。现在应用最广泛的 PLD 主要是现场可编程门阵列 FPGA(Field Programmable Gate Array)、复杂可编程逻辑器件 CPLD(Complex Programmable Logic Device)和可擦除可编程逻辑器件 EPLD(Erasable Programmable Logic Device)。

ASIC 是专门为某一应用领域或某一专门用户需要而设计制造的 LSI 或 VLSI 电路, 具有体积小、重量轻、功耗低、高性能、高可靠性、高保密性等优点。ASIC 的分类如图 1.1 所示。

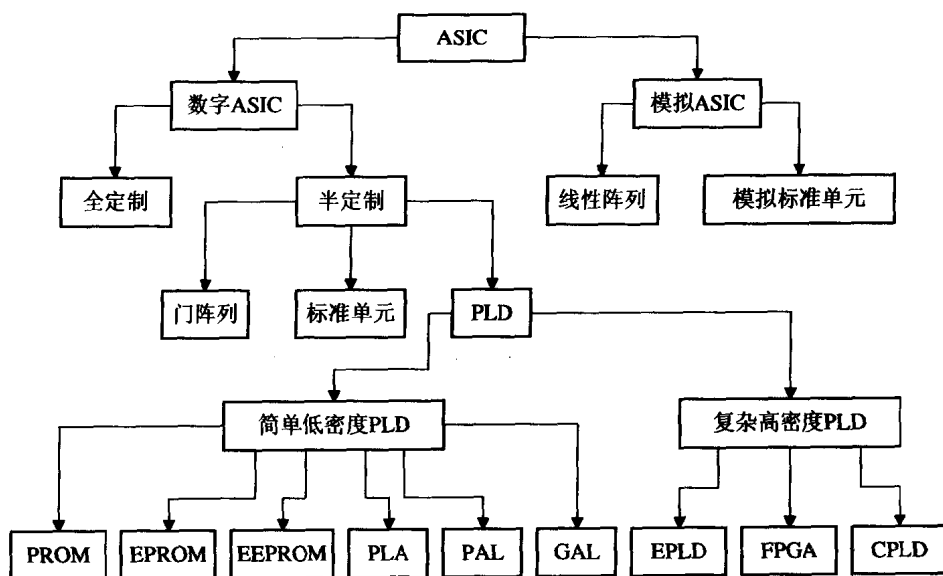


图 1.1 ASIC 的分类

### 1. 模拟 ASIC

除目前传统的运算放大器、功率放大器等电路外，模拟 ASIC 由线性阵列和模拟标准单元组成，与数字 ASIC 相比，它的发展还相当缓慢，其原因是模拟电路的频带宽度、精度、增益和动态范围等暂时还没有一个最佳的办法加以描述和控制。但模拟 ASIC 可减少芯片面积，提高性能，降低费用，扩大功能，降低功耗，提高可靠性，缩短开发周期，因此其发展也势在必行。科学的发展要求系统具有高精度、宽频带、大动态范围的增益和频带实时可变性等性能，因此在技术上要求采用数字和模拟混合的 ASIC，以提高整个电子系统的可靠性。

目前，生产厂家可提供由线性阵列和标准单元构成的运算放大器、比较器、振荡器、无源器件和开关电容滤波器等产品，对标准单元的简单修改仅需几小时，新单元设计只需几天，同电路相匹配的最佳电阻、电容值在几小时内即可获得，并且阵列的使用率高达 100%。

### 2. 数字 ASIC

#### 1) 全定制(Full custom design approach)ASIC

全定制 ASIC 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，以达到芯片面积利用率高、速度快、功耗低的最优化性能。设计全定制 ASIC，不仅要求设计人员具有丰富的半导体材料和工艺技术知识，还要具有完整的系统和电路设计的工程经验。全定制 ASIC 的设计费用高，周期长，比较适用于大批量的 ASIC 产品，如彩电中的专用芯片。

#### 2) 半定制(Semi-custom design approach)ASIC

半定制 ASIC 是一种约束型设计方法，它是在芯片上制作好一些具有通用性的单元元件和元件组的半成品硬件，用户仅需考虑电路逻辑功能和各功能模块之间的合理连接即

可。这种设计方法灵活方便,性价比高,缩短了设计周期,提高了成品率。半定制 ASIC 包括门阵列、标准单元和可编程逻辑器件三种。

门阵列(Gate Array)是按传统阵列和组合阵列在硅片上制成具有标准逻辑门的形式,它是不封装的半成品,生产厂家可根据用户要求,在掩膜中制作出互连的图案(码点),最后封装为成品,再提供给用户。

标准单元(Standard Cell)是由 IC 厂家将预先设置好、经过测试且具有一定功能的逻辑块作为标准单元存储在数据库中,包括标准的 TTL、CMOS、存储器、微处理器及 I/O 电路的专用单元阵列。设计人员在电路设计完成之后,利用 CAD 工具在版图一级完成与电路一一对应的最终设计。标准单元设计灵活,功能强,但设计和制造周期较长,开发费用也较高。

可编程逻辑器件(Programmable Logic Device)是 ASIC 的一个重要分支,是厂家作为一种通用性器件生产的半定制电路,用户可通过对器件编程实现所需要的逻辑功能。PLD 是用户可配置的逻辑器件,它的成本比较低,使用灵活,设计周期短,而且可靠性高,风险小,因而很快得到普遍应用,发展非常迅速。

PLD 从 20 世纪 70 年代发展到现在,已形成了许多类型的产品,其结构、工艺、集成度、速度和性能都在不断改进和提高。PLD 又可分为简单低密度 PLD 和复杂高密度 PLD。最早的 PLD 是 1970 年制成的 PROM(Programmable Read Only Memory),即可编程只读存储器,它是由固定的与阵列和可编程的或阵列组成。PROM 采用熔丝工艺编程,只能写一次,不能擦除和重写。随着技术的发展和应用要求,此后又出现了 UVEPROM(紫外线可擦除只读存储器)、EEPROM(电可擦除只读存储器),由于它们价格低,易于编程,速度低,适合于存储函数和数据表格,因此主要用作存储器。典型的 EPROM 有 2716、2732 等。

可编程逻辑阵列 PLA(Programmable Logic Array)于 20 世纪 70 年代中期出现,它是由可编程的与阵列和可编程的或阵列组成,但由于器件的资源利用率低,价格较贵,编程复杂,支持 PLA 的开发软件有一定难度,因而没有得到广泛应用。

可编程阵列逻辑 PAL(Programmable Array Logic)器件是 1977 年美国 MMI 公司(单片存储器公司)率先推出的,它由可编程的与阵列和固定的或阵列组成,采用熔丝编程方式,双极性工艺制造,器件的工作速度很高。由于它的输出结构种类很多,设计很灵活,因而成为第一个得到普遍应用的可编程逻辑器件,如 PAL16L8。

通用阵列逻辑器件 GAL(Generic Array Logic)器件是 1985 年 Lattice 公司最先发明的可电擦写、可重复编程、可设置加密位的 PLD。GAL 在 PAL 基础上,采用了输出逻辑宏单元形式 E<sup>2</sup>CMOS 工艺结构。具有代表性的 GAL 芯片有 GAL16V8、GAL20V8,这两种 GAL 几乎能够仿真所有类型的 PAL 器件。在实际应用中,GAL 器件对 PAL 器件仿真具有百分之百的兼容性,所以 GAL 几乎完全代替了 PAL 器件,并可以取代大部分 SSI、MSI 数字集成电路,如标准的 54/74 系列器件,因而获得广泛应用。

PAL 和 GAL 都属于简单 PLD,结构简单,设计灵活,对开发软件的要求低,但规模小,难以实现复杂的逻辑功能。随着技术的发展,简单 PLD 在集成密度和性能方面的局限性也暴露出来,其寄存器、I/O 引脚、时钟资源的数目有限,没有内部互连,因此包括 EPLD、CPLD 和 FPGA 在内的复杂 PLD 迅速发展起来,并向着高密度、高速度、低功耗以

及结构体系更灵活、适用范围更宽广的方向发展。

可擦除可编程逻辑器件 EPLD(Erasable PLD)是 20 世纪 80 年代中期 Altera 公司推出的基于 UVEPROM 和 CMOS 技术的 PLD, 后来发展到采用 E<sup>2</sup>CMOS 工艺制作的 PLD。EPLD 基本逻辑单元是宏单元。宏单元由可编程的与或阵列、可编程寄存器和可编程 I/O 三部分组成。从某种意义上讲 EPLD 是改进的 GAL, 它在 GAL 基础上大量增加输出宏单元的数目, 提供更大的与阵列, 灵活性较 GAL 有较大改善, 集成密度大幅度提高, 内部连线相对固定, 延时小, 有利于器件在高频率下工作, 但内部互连能力十分弱。世界著名的半导体器件公司如 Altera、Xilinx、AMD、Lattice 均有 EPLD 产品, 但结构差异较大。

复杂可编程逻辑器件 CPLD(Complex PLD)是 20 世纪 80 年代末 Lattice 公司提出了在线可编程(ISP, In System Programmability)技术以后于 20 世纪 90 年代初出现的。CPLD 是在 EPLD 的基础上发展起来的, 采用 E<sup>2</sup>CMOS 工艺制作, 与 EPLD 相比, 增加了内部连线, 对逻辑宏单元和 I/O 单元也有重大的改进。CPLD 至少包含三种结构: 可编程逻辑宏单元、可编程 I/O 单元和可编程内部连线。部分 CPLD 器件内部还集成了 RAM、FIFO 或双口 RAM 等存储器, 以适应 DSP 应用设计的要求。其典型器件有 Altera 的 MAX7000 系列, Xilinx 的 7000 和 9500 系列, Lattice 的 PLSI/ispLSI 系列和 AMD 的 MACH 系列。

现场可编程门阵列 FPGA(Field Programmable Gate Array)器件是 Xilinx 公司 1985 年首家推出的, 它是一种新型的高密度 PLD, 采用 CMOS-SRAM 工艺制作。FPGA 的结构与门阵列 PLD 不同, 其内部由许多独立的可编程逻辑模块(CLB)组成, 逻辑块之间可以灵活地相互连接。FPGA 的结构一般分为三部分: 可编程逻辑块、可编程 I/O 模块和可编程内部连线。CLB 的功能很强, 不仅能够实现逻辑函数, 还可以配置成 RAM 等复杂的形式。配置数据存放在片内的 SRAM 或者熔丝图上, 基于 SRAM 的 FPGA 器件工作前需要从芯片外部加载配置数据。配置数据可以存储在片外的 EPROM 或者计算机上, 设计人员可以控制加载过程, 在现场修改器件的逻辑功能, 即所谓现场可编程。FPGA 出现后受到电子设计工程师的普遍欢迎, 发展十分迅速。Xilinx、Altera 和 Actel 等公司都提供高性能的 FPGA 芯片。

高密度 PLD 在生产工艺、器件的编程和测试技术等方面都有了飞速发展。目前 PLD 的集成度非常高, 如 Altera 公司的 APEX II 系列器件, 采用 0.15  $\mu\text{m}$  工艺, 各层都是铜金属布线, 其中 EP2A90 的密度可达 400 万典型门, 可为用户提供 1140 个 I/O 引脚, 1 Gb/s 数据速率。世界各著名半导体器件公司, 如 Altera、Xilinx、Lattice、Actel 和 AMD 公司均可提供不同类型的 CPLD、FPGA 产品。众多公司的竞争促进了可编程集成电路技术的提高, 使其性能不断改善, 产品日益丰富, 价格逐步下降。可以预计, 可编程逻辑器件将在结构、密度、功能、速度和性能等方面得到进一步发展, 结合 EDA 技术, PLD 将在现代电子系统设计中得到非常广泛的应用。

## 1.2 可编程逻辑器件的基本结构

### 1.2.1 简单 PLD 的基本结构

简单 PLD 包括 PROM、PLA、PAL 和 GAL。其结构特点是由与阵列和或阵列组成,

能有效地实现“积之和”形式的布尔逻辑函数，与或阵列在硅片上也非常容易实现。与或表达式是布尔代数的常用表达式形式，根据布尔代数的知识，所有的逻辑函数均可以用与或表达式描述。在数字电路课程中，大家已经学过如何利用卡诺图、摩根定理和 Q-M 表，将真值表或其它形式的逻辑函数转换成与或表达式的方法。与或阵列的结构可以通过改变与或阵列的连接来实现不同的逻辑功能。不论改变与阵列的连接，还是改变或阵列的连接，都可以使所实现的逻辑函数发生变化。

简单 PLD 的基本结构框图如图 1.2 所示，图中与阵列和或阵列是电路的主体，主要用来实现组合逻辑函数。输入由缓冲器组成，它使输入信号具有足够的驱动能力并产生互补输入信号。输出电路可以提供不同的输出方式，如直接输出(组合方式)或通过寄存器输出(时序方式)。此外，输出端口上往往带有三态门，通过三态门控制数据直接输出或反馈到输入端。

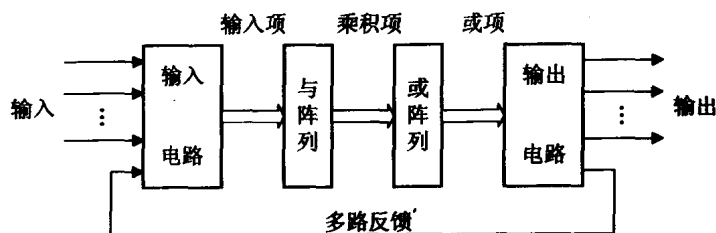


图 1.2 简单 PLD 的基本结构

根据与或阵列电路中只有部分电路可以编程以及组态的方式不同，PROM、PLA、PAL 和 GAL 四种 PLD 电路的结构特点如表 1.1 所示。

表 1.1 四种简单 PLD 电路的结构特点

| 类 型  | 阵 列 |     | 输 出 方 式          |
|------|-----|-----|------------------|
|      | 与   | 或   |                  |
| PROM | 固定  | 可编程 | TS(三态), OC(可熔极性) |
| PLA  | 可编程 | 可编程 | TS, OC           |
| PAL  | 可编程 | 固定  | TS, I/O, 寄存器反馈   |
| GAL  | 可编程 | 固定  | 用户定义             |

另外，由于 EPLD 和 CPLD 是在 GAL 的基础上发展起来的，其结构也是与阵列可编程或阵列固定。

图 1.3、图 1.4、图 1.5 分别画出了 PROM、PLA 和 PAL(GAL)的阵列结构图。在这些图中，左边部分为与阵列，右边部分为或阵列，与门采用“线与”的形式；在交叉点上的符号，实点表示固定连接，“\*”号表示可编程连接。输入信号通过互补缓冲器输入，通过交叉点上的连接加到函数的与或表达式的乘积项中。与阵列产生的多个乘积项，通过或阵列的交叉点连接，完成函数的或运算。其中 PAL 和 GAL 基本门阵列结构相同，均为与阵列可编程，或阵列固定连接，编程容易实现且费用低。一般在 PAL 和 GAL 产品中，最多的乘积项数可达 8 个。

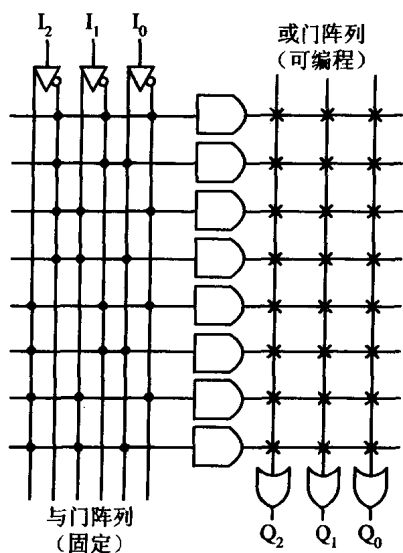


图 1.3 PROM 阵列结构图

PAL 和 GAL 的输出结构并不相同。PAL 的输出结构是固定的，不能编程。芯片型号选定后，输出结构也就选定了，根据输出和反馈的结构不同，PAL 器件主要有：可编程输入/输出结构，带反馈的寄存器型结构，异或结构，专用组合输出和算术选通反馈结构等。PAL 产品有 20 多种不同型号可供设计人员选择。

例如，图 1.6 所示的可编程输入/输出结构，其输出电路是一个三态缓冲器，反馈部分是一个具有互补输出的缓冲器。与阵列的第一个与门的输出控制三态门的输出，当与门输出为“0”时，三态门禁止，输出呈高阻状态，I/O 引脚可作为输入使用；当与门输出为“1”时，三态门被选通，I/O 引脚作为输出使用或阵列的输出信号经缓冲器反

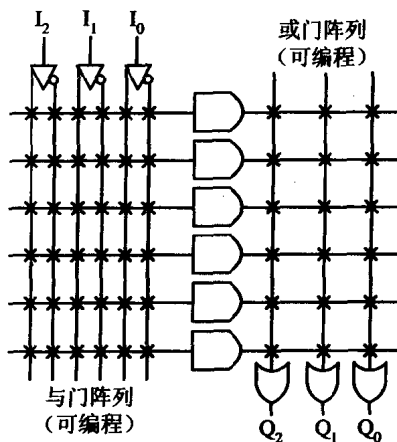


图 1.4 PLA 阵列结构图

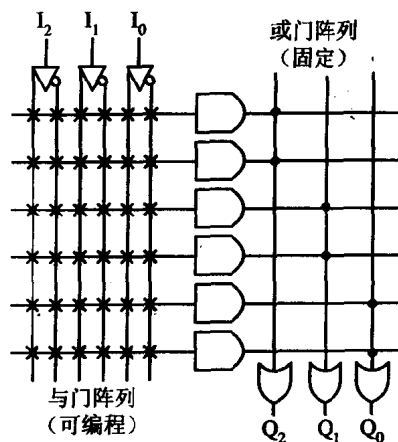


图 1.5 PAL(GAL)的阵列结构图

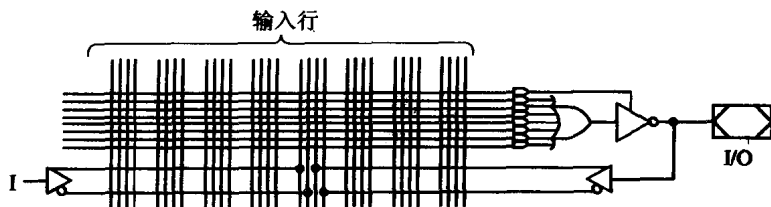


图 1.6 可编程输入/输出结构

相后,一路从 I/O 引脚送出,另一路经互补缓冲器反馈至与阵列的输入端。图 1.6 中只画出了一个输出,如产品 PAL16L8 则有八个输出。

带反馈的寄存器输出结构如图 1.7 所示,产品 PAL16R8(R 代表 Register)就属于寄存器输出结构。当系统时钟 CLK 的上升沿到来后,或门的输出被存入 D 触发器,然后通过选通三态缓冲器送到输出端, D 触发器的 Q 输出经反馈缓冲器送到与阵列的输入端,这样的 PAL 具有记忆功能,能实现时序逻辑功能,而 PROM 和 PLA 没有寄存器结构,不能实现时序逻辑。

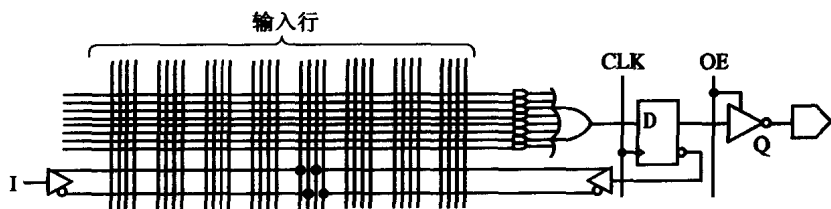


图 1.7 带反馈的寄存器输出结构

GAL 和 PAL 最大的差别在于 GAL 的输出结构可由用户定义,是一种灵活可编程的输出结构。GAL 的两种基本型号 GAL16V8(20 引脚)GAL20V8(24 引脚)可代替数十种 PAL 器件,因而称为通用可编程逻辑器件。GAL 的每一个输出端都集成了一个输出逻辑宏单元 OLMC(Output Logic Macro Cell),图 1.8 是 GAL22V10 的 OLMC 内部逻辑图。

OLMC 中除了包含或门阵列和 D 触发器之外,还多了两个数选器(MUX),其中 4 选 1 MUX 用来选择输出方式和输出极性,2 选 1 MUX 用来选择反馈信号。数选器的状态取决于两位可编程特征码  $S_1S_0$  的控制。编程信息使得  $S_1S_0$  编为 00、01、10、11 中的一个,OLMC 便可以分别被组态为四种输出方式中的一种,如图 1.9 所示。这四种输出方式分别

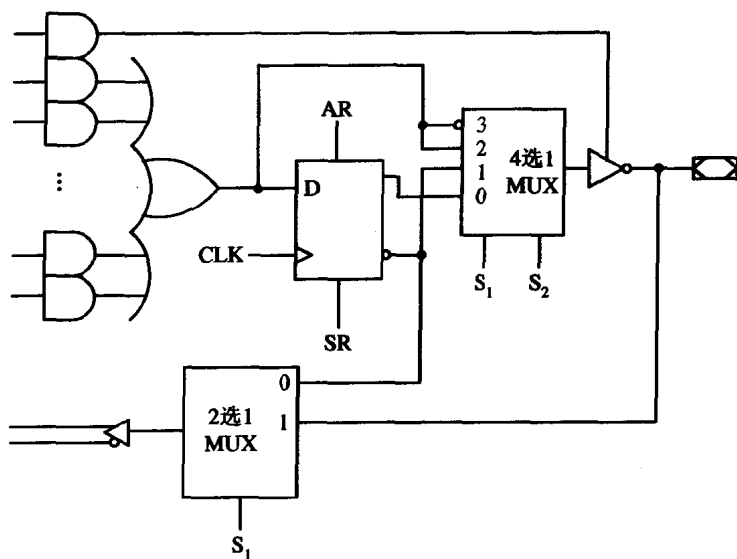


图 1.8 GAL22V10 的 OLMC