

# 模拟 数字

# 可编程器件与电路设计

汤琳宝 陈恒 徐晓夏 顾敏 乐峰 编 著



 科学出版社  
www.sciencep.com

# 模拟 数字 可编程器件与电路设计

汤琳宝 陈恒 徐晓夏  
顾敏 乐峰 编著

科学出版社

北 京

## 内 容 简 介

本书介绍了现代电子系统(模拟、数字)设计中使用的在系统编程技术,全书共分三部分。

第一部分介绍了最新的在系统模拟、数字可编程器件的原理和结构。具体可分为:在系统数字可编程逻辑器件:ispLSI系列、ispMACH4A系列、ispMACH4000系列;现场可编程系统芯片FPSC;无限可重构、可编程门阵列ispXPGA及ispXPLD等器件;在系统可编程模拟器件:ispPAC10、ispPAC20、ispPAC30、ispPAC80、ispPAC81器件。

第二部分讲述了VHDL语言和最新的开发软件,介绍了VHDL硬件描述语言用于电路设计与仿真的方法,并着重介绍模拟在系统可编程器件开发软件PAC Designer和数字在系统可编程器件开发软件ispLEVER的使用方法。

第三部分为模拟、数字电子系统设计的实例介绍。通过实例,读者可深入了解现代电子系统设计的特色。此外,还介绍了在系统模拟器件实验系统。

附录中提供了在系统可编程器件设计中需要的资料。

本书可作为从事电子产品开发和生产的工程技术人员的参考书,也可作为大专院校电子工程、通信工程、工业自动化、仪器仪表、计算机应用技术等专业相关课程的教材和教学参考书,也适合作为企业的培训教材。

### 图书在版编目(CIP)数据

模拟 数字可编程器件与电路设计/汤琳宝,陈恒等编著. —北京:科学出版社,2003

ISBN 7-03-012008-6

I. 模… II. ①汤… ②陈… III. 可编程序控制器 IV. TP332.3

中国版本图书馆CIP数据核字(2003)第066603号

责任编辑:王淑兰/责任校对:陈玉凤 都 岚

责任印制:吕春珉/封面设计:飞天创意

科学出版社 出版

北京东黄城根北街16号

邮政编码:100717

<http://www.sciencep.com>

新蕾印刷厂 印刷

科学出版社发行 各地新华书店经销

\*

2003年11月第 一 版 开本:787×1092 1/16

2003年11月第一次印刷 印张:14 1/2

印数:1—4 000 字数:320 000

定价:22.00元

(如有印装质量问题,我社负责调换〈环伟〉)

## 前 言

在系统可编程技术 ISP (In-System Programmable) 和在系统可编程器件的出现, 冲击了传统数字电子系统的设计方法, 给电子系统设计带来了革命性的变化。近年来出现的在系统可编程模拟器件 (In-System Programmability Programmable Analog Circuits) 又把电子系统设计自动化技术从数字系统设计拓展至模拟电子系统设计。目前, ISP 技术和可编程器件仍在不断迅速发展。如现场可编程系统芯片、可编程器件的高速 I/O 接口、在系统动态可编程模拟器件和 2002 年出现的 ispXP 技术。这些新技术和新器件的发展给电子系统的设计带来了新的生命。掌握在系统可编程技术是当代电子设计工程师及从事电子技术研究和开发人员的必备技能, 也是各类电子技术专业学生学习的重要内容。

本书系统地阐述了最新的在系统可编程模拟、数字器件的结构和工作原理, VHDL 硬件描述语言的结构和语法, 较系统地介绍了最新的用于数字系统开发的软件 is-  
pLEVER 和用于模拟电路开发的软件 PAC Designer 以及基于在系统可编程器件的电子系统设计方法, 并介绍了在系统模拟器件实验系统。

编者综合以往 ISP 技术教学经验, 从传授知识、实际应用出发编写了本书, 其目的是使读者了解最新的在系统可编程技术和可编程器件的发展方向, 理解器件原理和设计方法, 掌握当代先进的电子设计自动化技术。

本书第 3 章、第 7 章由陈恒编写, 第 4 章由陈恒、乐峰编写, 第 5 章由汤琳宝、顾敏编写, 第 6 章由汤琳宝、徐晓夏编写, 其余章节均由汤琳宝编写。在本书编写过程中, 汤琳宝负责统稿, 陈恒负责审稿。

本书在编写过程中上海大学通信学院张懿慧、毛洋林、黄惟冰、陈俊、龚岳松、洪彦琪、沈兆桢等同学做了大量的工作, 在此表示诚挚的感谢!

本书的出版得到了北京邮电大学白中英教授的大力支持, 在此表示衷心的感谢。

由于编者水平有限, 错误之处, 恳请读者指正。

编 者

2003 年 1 月

# 目 录

|                                             |        |
|---------------------------------------------|--------|
| 第 1 章 现代电子系统的设计 .....                       | ( 1 )  |
| 1.1 ISP 技术 .....                            | ( 1 )  |
| 1.1.1 ISP 技术 .....                          | ( 1 )  |
| 1.1.2 设计 .....                              | ( 2 )  |
| 1.1.3 制造 .....                              | ( 3 )  |
| 1.1.4 未来 .....                              | ( 5 )  |
| 1.2 ISP 器件 .....                            | ( 6 )  |
| 1.3 开发软件 .....                              | ( 6 )  |
| 第 2 章 在系统数字可编程器件 .....                      | ( 7 )  |
| 2.1 在系统数字可编程逻辑器件 ispLSI 系列 .....            | ( 7 )  |
| 2.1.1 结构 .....                              | ( 7 )  |
| 2.1.2 通用逻辑块 GLB (Gereric Logic Block) ..... | ( 8 )  |
| 2.1.3 布线区 .....                             | ( 10 ) |
| 2.1.4 I/O 单元 .....                          | ( 11 ) |
| 2.1.5 时钟分配网络 .....                          | ( 12 ) |
| 2.1.6 宏块结构 .....                            | ( 12 ) |
| 2.1.7 ispLSI1032 的技术参数 .....                | ( 13 ) |
| 2.2 ispMACH4A 系列器件 .....                    | ( 13 ) |
| 2.2.1 ispMACH4A 器件结构 .....                  | ( 14 ) |
| 2.2.2 ispMACH4A 器件的特点 .....                 | ( 17 ) |
| 2.2.3 技术指标 .....                            | ( 21 ) |
| 2.3 ispMACH4000 系列 .....                    | ( 21 ) |
| 2.3.1 结构 .....                              | ( 21 ) |
| 2.3.2 主要性能和应用范围 .....                       | ( 25 ) |
| 2.4 现场可编程系统芯片 FPSC .....                    | ( 26 ) |
| 2.4.1 ORCR4 结构 .....                        | ( 26 ) |
| 2.4.2 存储器模式 .....                           | ( 28 ) |
| 2.4.3 特殊功能块 .....                           | ( 30 ) |
| 2.4.4 ORT8850 FPSC .....                    | ( 31 ) |
| 2.4.5 FPSC 器件与应用 .....                      | ( 31 ) |
| 2.5 无限可重构可编程门阵列 ispXPGA .....               | ( 32 ) |
| 2.6 ispXPLD 器件 .....                        | ( 34 ) |
| 2.6.1 器件结构 .....                            | ( 34 ) |
| 2.6.2 多功能块 .....                            | ( 35 ) |

|              |                                          |               |
|--------------|------------------------------------------|---------------|
| 2.6.3        | 用于板级运作的系统输入/输出 .....                     | ( 36 )        |
| 2.7          | 在系统可编程通用数字开关 ispGDS 和互连器件 ispGDX/V ..... | ( 37 )        |
| 2.7.1        | 通用数字开关器件 GDS .....                       | ( 37 )        |
| 2.7.2        | 在系统数字互连器件 ispGDX、ispGDXV .....           | ( 40 )        |
| 2.8          | 编程接口和编程——ISP 方式和 JTAG 方式 .....           | ( 42 )        |
| 2.8.1        | ISP 方式 .....                             | ( 42 )        |
| 2.8.2        | JTAG 方式 .....                            | ( 43 )        |
| 2.8.3        | 下载 .....                                 | ( 44 )        |
| <b>第 3 章</b> | <b>在系统可编程模拟器件 ispPAC .....</b>           | <b>( 45 )</b> |
| 3.1          | 在系统可编程模拟器件的原理 .....                      | ( 45 )        |
| 3.1.1        | PAC 块的传递函数 .....                         | ( 45 )        |
| 3.1.2        | 用二个 PAC 块设计滤波器 .....                     | ( 47 )        |
| 3.1.3        | PAC 块的特点 .....                           | ( 50 )        |
| 3.1.4        | 差分输出差分输入电器的优越性 .....                     | ( 50 )        |
| 3.2          | 各种在系统可编程模拟器件的结构 .....                    | ( 51 )        |
| 3.2.1        | ispPAC10 .....                           | ( 51 )        |
| 3.2.2        | ispPAC20 .....                           | ( 52 )        |
| 3.2.3        | ispPAC80 .....                           | ( 54 )        |
| 3.2.4        | ispPAC30 .....                           | ( 55 )        |
| 3.2.5        | ispPAC81 .....                           | ( 59 )        |
| 3.3          | PAC 的接口电路 .....                          | ( 60 )        |
| 3.4          | ispPAC 的增益调整方法 .....                     | ( 64 )        |
| 3.5          | 动态可编程重构技术 .....                          | ( 68 )        |
| <b>第 4 章</b> | <b>VHDL 的基本结构与语法 .....</b>               | <b>( 73 )</b> |
| 4.1          | VHDL 的基本结构 .....                         | ( 73 )        |
| 4.1.1        | 实体 .....                                 | ( 74 )        |
| 4.1.2        | 结构体 .....                                | ( 75 )        |
| 4.1.3        | 配置 (Configuration) .....                 | ( 76 )        |
| 4.2          | 标识符、数据对象、数据类型及属性 .....                   | ( 77 )        |
| 4.2.1        | 标识符 .....                                | ( 77 )        |
| 4.2.2        | 数据对象 .....                               | ( 78 )        |
| 4.2.3        | 数据类型 .....                               | ( 80 )        |
| 4.2.4        | 属性 .....                                 | ( 81 )        |
| 4.2.5        | VHDL 运算符 .....                           | ( 82 )        |
| 4.3          | 程序包和设计库 .....                            | ( 83 )        |
| 4.3.1        | 程序包 .....                                | ( 83 )        |
| 4.3.2        | 库 .....                                  | ( 85 )        |
| 4.4          | VHDL 基本语句 .....                          | ( 85 )        |
| 4.4.1        | 并行语句 .....                               | ( 85 )        |

|                                                |         |
|------------------------------------------------|---------|
| 4.4.2 顺序 (Sequential) 语句 .....                 | ( 90 )  |
| 4.5 常用电路的 VHDL 描述 .....                        | ( 93 )  |
| <b>第 5 章 ispLEVER 软件的使用</b> .....              | ( 103 ) |
| 5.1 ispLEVER 编程软件 .....                        | ( 103 ) |
| 5.1.1 ispLEVER 简介 .....                        | ( 103 ) |
| 5.1.2 ispLEVER 软件环境 .....                      | ( 103 ) |
| 5.2 原理图输入方式 .....                              | ( 106 ) |
| 5.2.1 原理图输入方式 .....                            | ( 106 ) |
| 5.2.2 设计的编辑与仿真 .....                           | ( 111 ) |
| 5.3 VHDL 语言和原理图混合输入 .....                      | ( 119 ) |
| 5.3.1 启动 ispLEVER System .....                 | ( 119 ) |
| 5.3.2 建立顶层的原理图 .....                           | ( 120 ) |
| 5.3.3 建立用 VHDL 语言编写的宏器件的逻辑符号 .....             | ( 120 ) |
| 5.3.4 完成原理图 .....                              | ( 121 ) |
| 5.3.5 编写宏模块的 VHDL 源文件 (底层设计) .....             | ( 121 ) |
| 5.3.6 编译 VHDL .....                            | ( 124 ) |
| 5.3.7 仿真 .....                                 | ( 124 ) |
| 5.3.8 层次化操作方法 .....                            | ( 125 ) |
| 5.4 VHDL 和 Verilog 语言的输入方式 .....               | ( 126 ) |
| 5.4.1 VHDL 设计输入 .....                          | ( 126 ) |
| 5.4.2 Verilog 设计输入的操作步骤 .....                  | ( 129 ) |
| 5.5 可编程器件引脚、属性及其他参数的设置 .....                   | ( 130 ) |
| <b>第 6 章 PAC Designer 软件的使用</b> .....          | ( 132 ) |
| 6.1 PAC Designer 编程软件概述 .....                  | ( 132 ) |
| 6.2 设计输入方法 .....                               | ( 132 ) |
| 6.2.1 建立设计文件 .....                             | ( 133 ) |
| 6.2.2 设计输入方法 .....                             | ( 135 ) |
| 6.3 电路性能仿真 .....                               | ( 137 ) |
| 6.4 设计方案下载 .....                               | ( 139 ) |
| 6.4.1 硬件 .....                                 | ( 139 ) |
| 6.4.2 下载 .....                                 | ( 139 ) |
| 6.5 其他功能 .....                                 | ( 140 ) |
| 6.5.1 “Tools” → “Run Macro” 菜单——“宏”设计 .....    | ( 140 ) |
| 6.5.2 “File” → “Browse Library” —— “库”使用 ..... | ( 141 ) |
| 6.6 ispPAC80 器件的软件设计方法 .....                   | ( 142 ) |
| <b>第 7 章 数字电子系统设计实例</b> .....                  | ( 145 ) |
| 7.1 序列检测器设计 .....                              | ( 145 ) |
| 7.2 ALU 设计 .....                               | ( 149 ) |
| 7.3 总线仲裁电路的设计 .....                            | ( 151 ) |

|              |                             |                |
|--------------|-----------------------------|----------------|
| 7.4          | 电子数字闹钟 .....                | ( 155 )        |
| 7.5          | 异步收发器 .....                 | ( 163 )        |
| 7.6          | 用 CPLD 和单片机设计电子系统 .....     | ( 168 )        |
| 7.7          | 数据传输与 I/O 接口标准 .....        | ( 170 )        |
| <b>第 8 章</b> | <b>模拟电子系统设计实例 .....</b>     | <b>( 174 )</b> |
| 8.1          | 单片电路设计 .....                | ( 174 )        |
| 8.1.1        | 过电压监控器 .....                | ( 174 )        |
| 8.1.2        | 动态性能可变的有源滤波器 .....          | ( 175 )        |
| 8.1.3        | 用 ispPAC30 组成温度监控电路 .....   | ( 177 )        |
| 8.2          | 小信号测量系统 .....               | ( 179 )        |
| 8.2.1        | 硬件框图 .....                  | ( 180 )        |
| 8.2.2        | 软件 .....                    | ( 182 )        |
| 8.2.3        | 调试 .....                    | ( 185 )        |
| <b>第 9 章</b> | <b>在系统可编程模拟器件实验系统 .....</b> | <b>( 188 )</b> |
| 9.1          | 智能化通用实验系统的基本结构 .....        | ( 188 )        |
| 9.1.1        | 模拟可编辑模块 .....               | ( 188 )        |
| 9.1.2        | 可编程下载电路 .....               | ( 189 )        |
| 9.1.3        | 直流信号输入区 .....               | ( 189 )        |
| 9.1.4        | 交流信号输入区 .....               | ( 190 )        |
| 9.1.5        | 电压信号监视区 .....               | ( 191 )        |
| 9.1.6        | 单片机控制电路 .....               | ( 191 )        |
| 9.2          | 实验系统操作与步骤 .....             | ( 195 )        |
| 9.2.1        | 系统操作步骤 .....                | ( 195 )        |
| 9.2.2        | 实验举例 .....                  | ( 196 )        |
| <b>附录</b>    | <b>.....</b>                | <b>( 200 )</b> |
| 附录一          | 宏库 .....                    | ( 200 )        |
| 附录二          | 各种器件的下载电路 .....             | ( 208 )        |
| 附录三          | 主要器件引脚图 .....               | ( 212 )        |
| 附录四          | 缩略语词汇表 .....                | ( 219 )        |
| <b>参考文献</b>  | <b>.....</b>                | <b>( 223 )</b> |

# 第1章 现代电子系统的设计

随着当今科技的日新月异, 半导体集成电路技术和计算机技术的飞速发展, 电子科技的发展也呈现出一副欣欣向荣的景象; 而作为电子技术一部分的电子器件也依旧遵循着摩尔定律, 以每 18 个月为一个进阶在电子器件的种类和功能上不断突破创新。现代电子系统正朝着小型化、微型化、多功能化的方向迈进, 而作为当代电子系统基础的电子器件也经历了由分立元件、小规模集成电路、中规模集成电路到大规模集成电路、超大规模集成电路的发展过程。在超大规模集成电路中, 可编程器件, 尤其是在系统可编程 (ISP) 器件是目前最先进的可编程器件。

1992 年美国 Lattice 公司发明了在系统可编程技术 (In-System Programmability), 彻底改变了传统数字电子系统的设计和实现方法, 开创了数字系统设计的里程碑。

在 21 世纪来临的前夕, 1999 年 11 月, Lattice 公司又推出了在系统可编程模拟电路 (In-System Programmability Programmable Analog Circuits)。Lattice 的在系统模拟可编程技术提供了一种简化和加快模拟电路设计开发的方案, 翻开了模拟电路设计方法的新篇章。为电子设计自动化 (EDA) 技术的应用开拓了更广阔的前景。

2002 年 7 月, Lattice 推出的 ispXPGA (in-system programmable eXpanded Programmable Gate Array) 是业界第一个在系统可编程且动态可重构的、瞬时上电的 FPGA 产品系列。ispXPGA 系列在一个非易失性结构中结合了在芯片 EE 存储器和 SRAM 单元, 从而允许无限可重构。这一独特的合并技术被称为 ispXP (eXpanded Programmability)。ispXPGA 系列中的 ispXP 器件能在上电后的几微秒时间内自举, 在电子系统电源开启的瞬间就能够正常工作。由于采用了在芯片 EE 存储器, 在编程过程中逻辑信号不会外露, 而且其加密位可防止 FPGA 内容的回读, 因此该产品具有很高的安全性。

与传统的电子器件相比, 可编程器件具有非常巨大的优势。可编程器件具有开发成本低, 开发周期短, 应用广泛, 在投入最少的情况下根据客户要求设计制造产品, 灵活多变, 保密性好不易被仿制, 而且可在线仿真和在线调试, 开发的产品周期较之于传统的电子产品周期大大缩短, 为企业能够带来更低的研发成本和更大的经济效益。

本章主要介绍的是 ISP 技术和 ISP 器件, 从而使读者能够掌握近代电子系统的设计方法。

## 1.1 ISP 技术

### 1.1.1 ISP 技术

ISP 技术使用户能够在无需从电路板上拆下芯片或者从系统中拔下电路板的情况下, 改变芯片的内容或整个电子系统的功能。

对 ISP 器件编程是无需编程器的, 只要通过下载电缆和 PC 机相连 (如图 1-1 所示), 直接在用户自己设计的目标系统对 ISP 器件编程。它打破了传统可编程电路设

计必须先编程后装配的流程，在系统技术提供了一种自上而下，先装配后编程的设计方法，而且在产品成型后，还可以反复编程升级，使生产维护和系统更新都发生了革命性的变化，将系统设计提升到一个新的高度。

由于 ISP 技术的优点，以及它给用户所带来的时间和经济的效益，使得它的应用越来越广。可以预测，未来的硬件系统将越来越多的使用 ISP 器件，系统设计者将需要接受新的思想去开发硬件。而硬件将不再是固定不变的，ISP 器件的采用将使硬件设计更具有通用性，个性化。ISP 器件广泛的应用前景和快速的发展代表了可编程系统的新时代已经来临了。

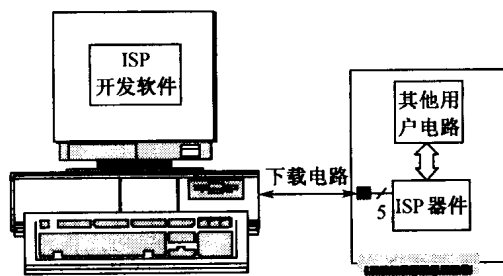


图 1-1 用 PC 机并行口进行 ISP 编程

ISP 技术改变了电子系统的设计程式。用传统的方法搭电路，调试电路功能是很费时间的，常常众多的接线及接触不良给调试带来了很大的麻烦。而这些硬件的调试可被计算机仿真所取代。在仿真过程中，可对未达到设计要求的电路进行修改，直到满足设计者要求为止，最后，烧录到 PLD 芯片之中（称为下载）。这样即可节省时间又能避免不必要的损失，加快了电子系统的开发、研制过程。

### 1.1.2 设计

电子系统的电路设计一般分为两部分：模拟电路部分和数字电路部分。

采用 ISP 技术后，工程师能够对已焊接在电路板上的数字、模拟可编程器件进行重构。这种性能给设计开发，电路板级调试和系统升级换代带来了极大的方便。

用 ISP 技术设计打破了传统样机的设计步骤，这是一种全新的电子系统设计方式，提高了设计的工作效率。

#### 1. 样机的制作——预先确定印刷电路板

采用 ISP 技术后，在设计工程中能尽早的确定电路的布局，甚至可把器件先安装在印刷线路板上。用开发软件对电路进行设计，然后通过一根编辑电缆将设计下载到 ISP 器件中去。无论增加或修改逻辑设计，都可以在几秒钟内把新的逻辑下载到器件中。由此可见，硬件的设计（即电路板的设计），像软件一样方便。而且重构整个系统功能不需改变电路布局。这就能大大的缩短了设计试制的周期，降低了试制成本，是一种全新的设计方法。

## 2. 系统调试

硬件系统与软件配合常为系统调试一个较大的难题。采用 ISP 技术后, 这种硬件与软件之间的沟通就很容易建立, 并且能够更快的实现软硬之间的最初连接。可以用开发软件对电路进行仿真, 对 ISP 器件中的内容(数字、模拟)进行修改, 从而加速了系统调试, 实现协同设计方案的最优化。最为实际的好处是减少电路板的返工, 从而迅速的完成系统级的调试工作。

## 3. 方案保密

设计方案的保密是设计工程师所希望拥有的功能, ISP 技术支持设计方案的保密, 保护设计成果。

### 1.1.3 制造

ISP 技术的采用使得生产制造过程也发生了巨大的变化。

#### 1. 降低生产成本

与基于标准元件所组建起来的传统电路设计相比, ISP 器件由减少使用特定单元的数量来控制制造成本。通过高度整合及将电路板空间减到最小需求, 以此来降低制造费用。而且由于可编程器件的应用, 减少了外围元件的需求, 这样就大大增加了电路板制造业的产量及减少了电路板重做的费用。

由于 ISP 技术可以在器件被焊接在电路板上的情况下对系统编程及重构。因此在设计系统的时候, 可以先制作电路板, 然后再对 ISP 器件编程。由于对于硬件设计方案的改动不需要更改外部元件和电路, 只要通过相应的开发工具在现场完成。所以虽然是硬件电路设计的改动, 但是却像修改软件一样灵活方便。

与数字的在系统可编程大规模集成电路一样, 在系统可编程模拟器件允许设计者使用开发软件在计算机中设计、修改模拟电路, 进行电路特性模拟, 最后通过编程电缆将设计方案下载至芯片中。

ISP 技术能利用自动测试设备(ATE)进行 ISP 器件的编程与测试。用 ATE 进行 ISP 器件的生产性编程。可省去与器件单独编程有关时间与开销。也可开发 ISP 器件的使用程序来提高电路板的质量与可靠性。

采用 ATE 测试设备进行 ISP 器件的编程不需额外的硬件设备和操作人员培训, 可把 ISP 器件编程与原来的测试环境结合起来以便提高产品质量, 降低成本。

#### 2. 简化了生产的流程

由于 ISP 器件集成度高, 而且可以装配在电路板上编程, 省去了插拔器件所带来的引脚损伤, 因此系统的可靠性更高, 体积可以更小。图 1-2 是 ISP 制造流程与非 ISP 流程的比较, 从图中可知 ISP 器件可以在产品上调试, 消除了单个编程和标记等工序, 简化了生产流程, 缩短了生产时间。

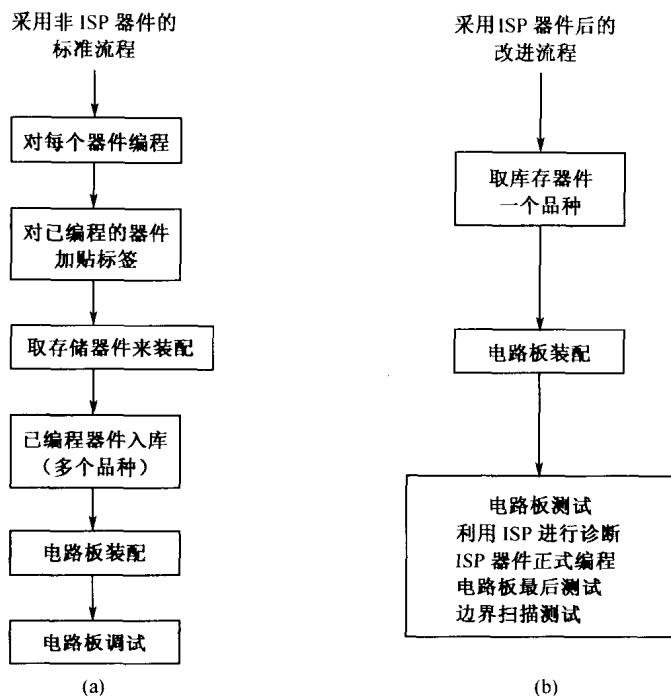


图 1-2 ISP 制造流程与非 ISP 流程的比较图

### 3. 产品的系统的升级和维护

由于 ISP 技术的特点，允许对系统的维护 and 升级可以在现场通过一台笔记本电脑和下载电缆而完成。也使得通过互联网或其他通信手段来对系统进行升级和维护成为可能。

### 4. ISP 技术的新发展

传统的可编程器件用 E<sup>2</sup>CMOS 或者 SRAM 来存储定义器件功能的配置数据。这两种工艺各有其优缺点。E<sup>2</sup>CMOS 是一种非易失工艺，编程数据不会丢失。但编程的时间比 SRAM 长，且编程次数有所限制。SRAM 是一种易失工艺，可以无限制的重构。但是，当可编程器件失去供电电压时，编程数据马上丢失。通常 FPGA 必须伴有器件外部的引导 PROM，用以配置数据。Lattice 公司采用了独特的方法来配置存储器，即把 E<sup>2</sup>CMOS 和 SRAM 结合在一个器件中。E<sup>2</sup>CMOS 除去了过去 SRAM 器件所依赖的外部配置器件。而 SRAM 给予过去 E<sup>2</sup>CMOS 器件不具有的无限制重构特性。这一独特的合并技术称为 ispXP (eXpanded Programmability)。

ispXP 技术是非易失性与无限可重构性的完美结合。由于采用了新的 ispXP 技术，ispXPLD 系列和 ispXPGA 器件结合了基于电可擦除技术的非易失性 PLD 和基于 SRAM 技术的可重构的 FPGA 的优点，因而具有以下特点：

① 系统启动时“瞬时上电”工作，无需外部初始化即可支持关键的系统“心跳”功能；

- ② 易失性的在系统编程消除了外部的引导 PROM，从而具有较高的集成度；
- ③ 由于系统初始化时编程逻辑信号是不可见的，因此提高了设计的安全性。

通过一个 8 位微处理机端口或 JTAG 边界扫描端口实现了无限可重构性，成为各类电子系统最理想的选择。

图 1-3 为 ispXP 器件的结构，用以说明存储器和用来配置或编程口之间的关系。SRAM 用来控制器件的功能。E<sup>2</sup>CMOS 存储器作为配置存储器。器件一旦上电，SRAM 很快的被配置。这里涉及到两个术语：“配置”和“编程”。“配置”是指下载信息至 SRAM 的行为。“编程”是指下载和存储数据至 E<sup>2</sup>CMOS。

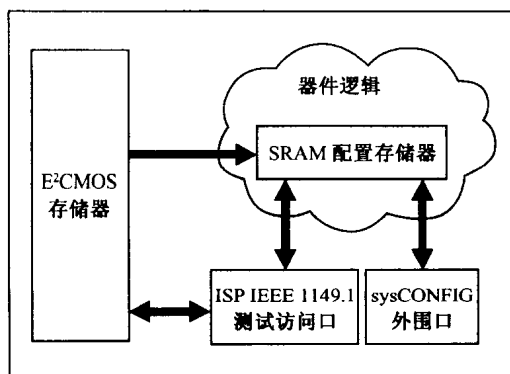


图 1-3 ispXP 器件的结构

从图 1-3 中看到，有两个口可以用来配置 SRAM。ISP 口或者 SysCONFIG 外围数据口。只有 ISP 口可用来对 E<sup>2</sup>CMOS 存储器编程。ISP 口完全与 IEEE 1149.1 测试访问口标准相兼容，也称为 JTAG。ispXP 器件支持 4 线的串行接口信号，这些信号分别是：TCK、TMS、TDI 和 TDO。许多嵌入式系统要求能很快地配置可编程器件。SysCONFIG 可用作微处理机的接口，用以快速配置器件。通过 SysCONFIG 口进行配置只限于 ispXP 器件的 SRAM。

对 SRAM 进行配置的另一种方法是直接通过 E<sup>2</sup>CMOS。在任何时刻 E<sup>2</sup>CMOS 中的内容可以下载至 SRAM。上电时刻，这个过程小于 200μs。

#### 1.1.4 未来

ISP 技术是一门正在突起的新技术，它的应用越来越广。

ISP 未来又将会怎样呢？首先“硬件”软件化，硬件的设计不再是“固定不变”的。由于 ISP 器件的参与，使硬件的设计更具有适用的组态。通过在系统可编程逻辑和互连关系，每个电路板就能根据实际应用需要来重构其逻辑功能。就像一台 PC 机，可根据需要运行 CAE 软件、或运行财务报表软件、或运行多媒体软件……。同样，ISP 技术也可通过对 isp 器件的下载，使同一个电路板做到各种不同的逻辑系统。可以预见，进入 21 世纪，一个真正的“可编程系统”时代将要到来。

## 1.2 ISP 器件

Lattice 的 ISP 器件具有多种类型。有复杂可编程逻辑器件 (CPLD)、在系统模拟可编程器件 (ispPAC)、现场可编程门阵列 (FPGA) 和现场可编程系统器件 (FPSC)。这些在系统可编程器件为当今电子系统的设计提供了全套解决方案。

本书第 2 章中将专门介绍这些 ISP 器件。

## 1.3 开发软件

### 1. ispLEVER 软件

Lattice 的 ispLEVER 设计软件的用户界面沿用了以前版本的界面, 保留了用户熟悉的设计环境。该软件支持除 ispPAC 之外的其他各种 isp 器件。在 ispLEVER 中增加了时序驱动的布局布线技术, 并集成了 Exemplar 和 Synlicity 公司提供的优化综合技术, 能使设计者快速地利用可编程器件的功能和特点, 并取得最大的资源利用率。ispLEVER 通过输入和输出工业标准 EDIF 网表来实现对第三方 EDA 工具的支持。

关于这个软件的使用将在第 5 章中作较为详细的介绍。

### 2. PAC-Designer 软件

PAC-Designer 软件是专门为 ispPAC 器件设计的开发软件。PAC-Designer 软件为不同类型的 ispPAC 提供了器件的图形设计界面和各种元件参数的选择。此外在各种器件中又提供各种适合该器件组成的库和宏, 使器件的设计更加方便, 并为设计者提供了相应的模拟电路幅频、相频特性。能进行电路仿真及器件编程。关于该软件的操作将在第 6 章中作较为详细的介绍。

## 第2章 在系统数字可编程器件

本章主要介绍 Lattice 公司的 CPLD、FPSC、ispXPGA、ispXPLD 器件。其中 ispLSI、ispMACH 均属于 CPLD 器件。

### 2.1 在系统数字可编程逻辑器件 ispLSI 系列

ispLSI 系列器件是基于与、或阵列结构的复杂可编程器件 CPLD (Complex Programmable Logic Device)。芯片由若干个宏块组成。宏块之间通过全局布线区 GRP 连接起来。

#### 2.1.1 结构

图 2-1 是 ispLSI1032 的内部结构图。

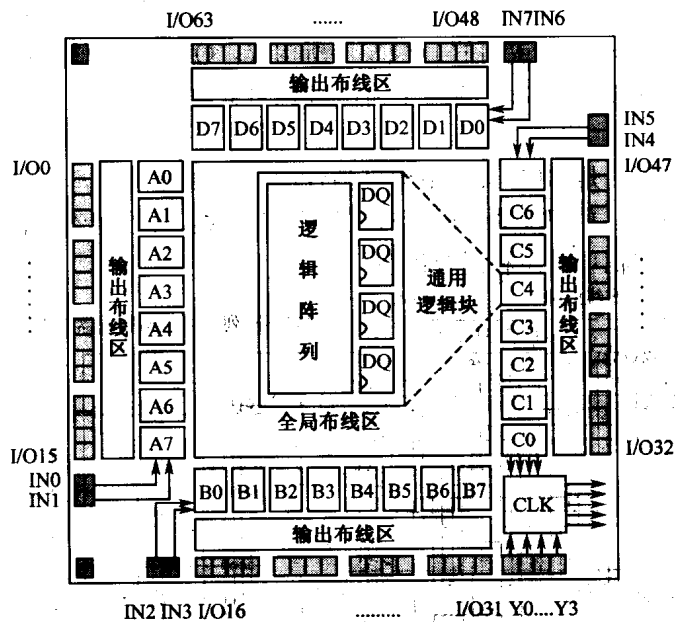


图 2-1 ispLSI1032 功能框图

从图 2-1 上可看出，该器件结构为：4 个宏块 (megablock)，一个全局布线区 (Global Routing Pool) 和一个时钟分配网络 (CLK)。每个宏块中，包含了 8 个通用逻辑块 (GLB)，一个输出布线区 (ORP)，一条输入总线和 18 个引脚。其中，16 个为 I/O 引脚，2 个为直接输入引脚。

图 2-1 表明了信号的大致流向。由 64 个 I/O 引脚 (I/O 0~I/O 63) 输入的信号，

通过输入总线进入全局布线区 (GRP)，在 GRP 中经过编程流向所需要的一个 (或几个) GLB。8 个直接输入端 (IN0—IN7) 则经过全局布线区将信号直接送 GLB。系统主要逻辑功能在 32 个 (A0, A1, … D7) GLB 内完成。GLB 输出可分三路，一路反馈到 GRP。在 GRP 中，可经编程能和任何一个 GLB I/O 的脚输入连接。一路送到输出布线区，一路直接输出，这样经过编程 GLB 的输出信号可分配到各 I/O 脚进行输出。

时钟信号由四个时钟输入端 (Y0-Y3) 输入。经过时钟分配网络后，有五路时钟输出 (CLK0, CLK1, CLK2, IOCLK0, IOCLK1)，送到各个 GLB 作全局时钟，也可送到各个 I/O 单元作为 I/O 时钟。另外，GLB “C0” 的输出信号也可以送到时钟分配网络作为时钟信号。

### 2.1.2 通用逻辑块 GLB (Gereric Logic Block)

通用逻辑块 (GLB) 是整个器件的核心，它提供了强有力而又灵活的逻辑构造模块，有效地支持宽输入或窄输入逻辑功能。

由图 2-2 可知，每个 GLB 包括一个与逻辑阵列，一个乘积项共享和四个输出逻辑宏单元，与阵列有 18 个输入变量，其中 16 个输入变量来自 GRP，其余 2 个输入变量来自直接外部输入，与阵列生成 20 个乘积项，输出到乘积项共享阵列 (PTSA) 由该阵列对乘积项进行再分配，输出到 GLB 的四个功能灵活的输出逻辑宏单元 (OLMC)，每个 OLMC 由一个具有输入异或门的 D 触发器所构成。OLMC 使每个 GLB 输出能够被编程为组合输出或时序输出。GLB 的四个输出端也可共享同一个乘积项，以提高芯片资源利用率。

乘积项共享阵列，是 GLB 具有结构灵活性的一个独特单元。对设计尤其对大型复杂状态机是很有用的。乘积项共享阵列有五种阻态形式：标准组态，高速旁路阻态，异或功能组态，单一乘积项组态和混合组态。

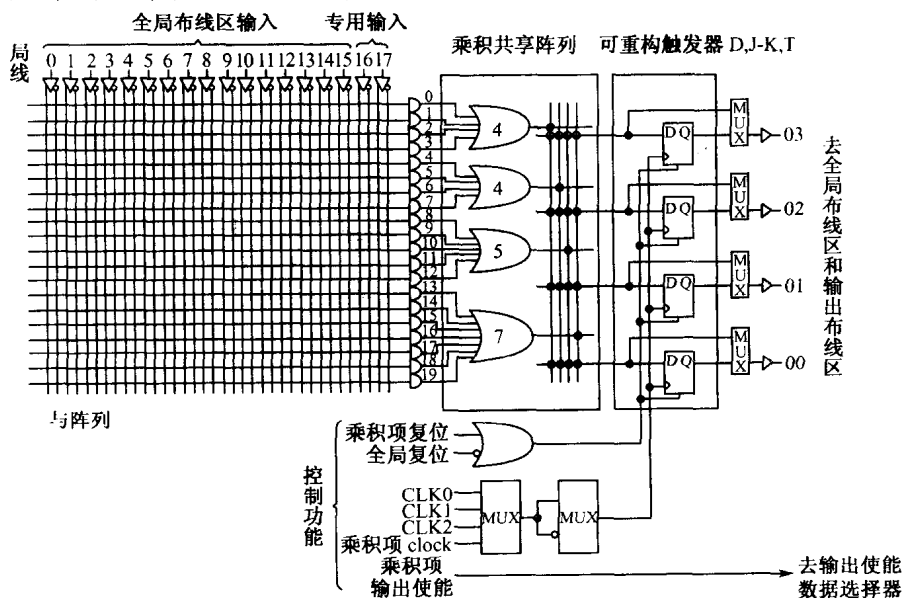


图 2-2 GLB 标准组态模式

标准组态模式如图 2-2 所示。与阵列生成的 20 个乘积项输出经由乘积项数分别为 4、4、5、7 的四个或门输出到可编程或阵列，然后输出到 GLB 的四个输出，每个或门的输出到任何一个 GLB 输出，四个 GLB 输出均可共享同一个乘积项。对于 GLB 输出的最大乘积项为 20。

高速旁路组态模式如图 2-3 所示，每个 GLB 输出对应于一个或门输出，且每个或门只有四个乘积项，GLB 输出之间不能共享同一乘积项，这种组态模式用来支持快速计数器设计。

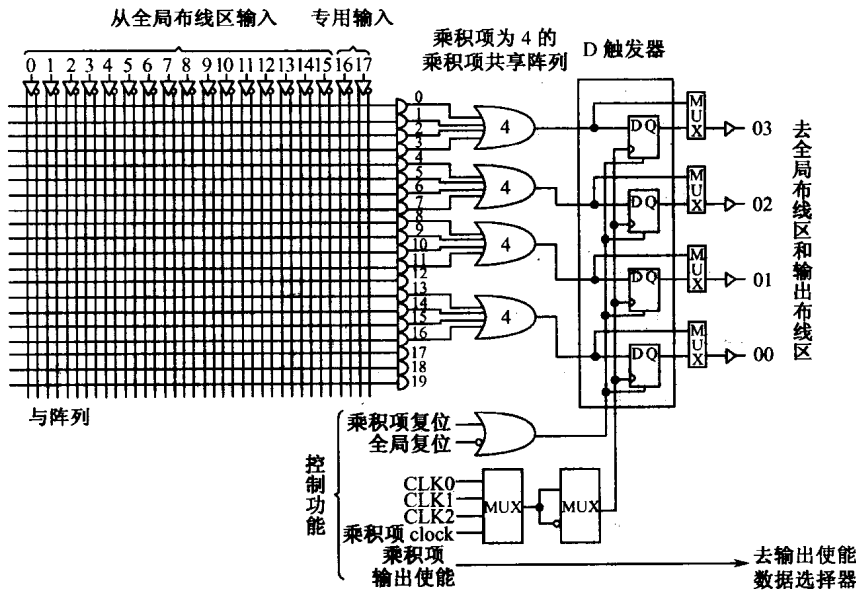


图 2-3 GLB 高速旁路组态模式

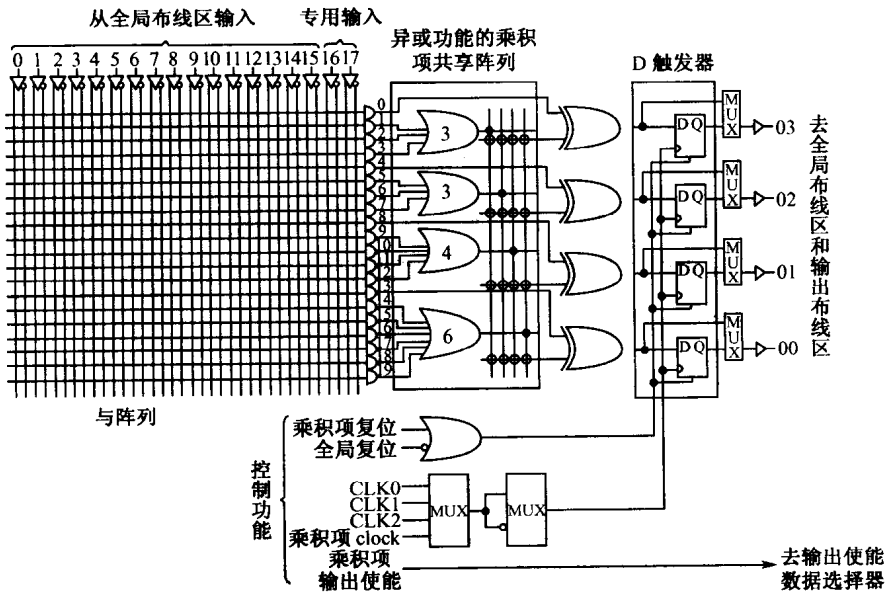


图 2-4 GLB 异或功能组态模式