



优秀技术实训教

VHDL

数字系统设计

优技
丛书

李欣 张海燕 编著

简明、系统地讲解VHDL设计方法

跳出单纯的语句、语法介绍

适合电子信息工程、通信工程、自动化等



科学出版社

www.sciencep.com



优秀技术实训教程

VHDL

数字系统设计

李欣 张海燕 编著

简明、系统地讲解VHDL设计方法

跳出单纯的语句、语法介绍

适合电子信息工程、通信工程、自动化等专业

丛书



科学出版社
www.sciencep.com

内 容 简 介

这是一本重点介绍硬件描述语言 VHDL 及其数字系统设计、应用的专业图书。

全书包含 5 部分内容,第 1 章从数字集成电路和可编程逻辑器件的基本知识入手,逐步介绍数字系统的设计工具和设计方法,以及与之相关的知识产权核 (IP Core) 和优化设计等概念;第 2 章至第 4 章将硬件描述语言 VHDL 作为设计手段,介绍基于 VHDL 的数字系统设计方法;第 5 章通过一个具体实例展示了 VHDL 描述的硬件实现过程;第 6 章展示了一些典型数字单元电路的 VHDL 描述实例;第 7 章将一些常用程序包的源代码——特别是包体的源代码介绍给读者,以便了解 VHDL 共享机制的描述技巧。

本书内容浅显,逻辑清晰,知识与实例紧密结合,适合电子信息工程、通信工程、计算机、自动化等专业师生,也可作为授课教材或者主要参考书。

图书在版编目 (CIP) 数据

VHDL 数字系统设计 / 李欣, 张海燕 编著. —北京: 科学出版社, 2009

(优技丛书)

ISBN 978-7-03-025497-9

I. V... II. ①李...②张... III. 硬件描述语言, VHDL—程序设计 IV. TP312

中国版本图书馆 CIP 数据核字 (2009) 第 157941 号

责任编辑: 但明天 / 责任校对: 马 君

责任印刷: 媛 明 / 封面设计: 盛春雨

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>

北京媛明印刷厂印刷

科学出版社发行 各地新华书店经销

*

2009 年 9 月第 1 版

开本: 787mm×1092mm 1/16

2009 年 9 月第 1 次印刷

印张: 15.75

印数: 1—3 000

字数: 362 千字

定价: 26.00 元

前 言

数字系统设计是电子信息类专业本科生的主要专业课程之一。随着专用集成电路(ASIC)技术的发展,以通用集成电路为主的传统设计方法,已经不能适应目前的数字系统设计要求。我们于1998年在电子信息类专业的本科生教学中开设了基于硬件描述语言VHDL的数字系统设计课程,积累了一些教学经验。

本书是在总结数字系统设计课程教学经验的基础上,参考有关文献资料和其他教材编写而成的。主要针对教学课时有限的情况下,使学生能够较快地掌握利用电子设计自动化(EDA)工具设计数字系统的方法,同时学习与数字集成电路和可编程逻辑器件(PLD)有关的基本知识。

第1章从介绍数字集成电路和可编程逻辑器件的基本知识入手,逐步介绍数字系统的设计工具和设计方法,以及与之相关的知识产权核(IP Core)和优化设计等概念,使读者对数字系统设计有一个较全面的初步了解。第2章至第4章,将硬件描述语言VHDL作为设计手段,向学生介绍基于VHDL的数字系统设计方法,并尽力体现描述、划分、综合和验证等工作在数字系统设计中的运用,同时强调了基于寄存器的设计和可综合的寄存器传输级(RTL)编码原则。第5章通过一个三相六拍顺序脉冲发生器的设计实例,从Quartus II集成开发环境的安装和授权文件设置开始,将一个VHDL描述的硬件实现过程展示给读者。第6章展示了一些典型数字单元电路的VHDL描述实例。在实践教学中,可以将这些实例改动后作为实验项目开设实验课,使读者将理论学习与实践动手有机结合,巩固所学知识。第7章将VHDL标准设计库中的标准程序包(STANDARD)、文本输入/输出程序包(TEXTIO),和IEEE VHDL设计库中的Std_Logic_1164、Std_Logic_Arith、Std_Logic_Unsigned、Std_Logic_Signed等常用程序包的源代码,特别是一些包体的源代码介绍给学生,让学生学习和了解VHDL共享机制的描述技巧。

建议本课程的授课时数为50~60学时,其中包括16~20学时的实验课时,即理论学时与实践学时之比约为2:1。

本书主要由李欣、张海燕编写,在本书第7章的编写过程中,得到了姚利华同学的热心帮助,在此表示衷心感谢!另外参与编写的还有管殿柱、宋一兵、李文秋、田东、宋绮、赵景波、赵景伟、张洪信、王献红、付本国、谈世哲、张轩、刘平、张宪海、林晶、林琳、柴永生、宿晓宁、齐薇、马震、李仲等。

限于作者的经验和水平,不足之处在所难免,恳请读者批评指正。电子邮箱:eleceng@ouc.edu.cn。

零点工作室交流平台: www.zerobook.net

作 者

目 录

第1章 概论	1
1.1 数字集成电路分类	2
1.1.1 按生产工艺分类	2
1.1.2 按生产目的分类	2
1.1.3 按制造方法分类	3
1.2 可编程逻辑器件简介	5
1.2.1 PLD 的分类	5
1.2.2 PLD 的发展历程	11
1.3 数字系统的设计工具与设计流程	12
1.3.1 数字系统设计自动化技术的发展历程	12
1.3.2 数字系统的设计流程	13
1.4 知识产权核 (Core, IP Core)	20
1.5 数字系统设计中的其他问题	21
1.5.1 优化设计	21
1.5.2 时钟信号与复位信号设计	24
1.5.3 数字系统的可观察性设计	25
1.6 本章小结	26
1.7 习题	27
第2章 硬件描述语言 VHDL 入门	28
2.1 VHDL 的由来	28
2.2 位全加器的描述实例	29
2.3 基本的 VHDL 模型结构	30
2.3.1 设计实体	30
2.3.2 实体声明	30
2.3.3 结构体	33
2.4 VHDL 标识符	36
2.4.1 基本标识符	36
2.4.2 扩展标识符	36
2.5 VHDL 对象	37
2.6 VHDL 数据类型和子类型	39
2.6.1 文字	39
2.6.2 标量类型	41
2.6.3 复合类型	43
2.6.4 子类型	45
2.6.5 类型转换	46

2.7	属性	46
2.8	运算符与聚合赋值	49
2.8.1	算术运算符	50
2.8.2	逻辑运算符	51
2.8.3	关系运算符	51
2.8.4	连接运算符	52
2.8.5	聚合赋值	53
2.9	本章小结	53
2.10	习题	54
第 3 章	VHDL 基本语句	56
3.1	仿真与延迟	56
3.1.1	仿真 Δ 机制	57
3.1.2	延迟	58
3.2	进程语句与 WAIT 语句	59
3.2.1	进程语句	60
3.2.2	WAIT 语句	61
3.3	顺序语句	62
3.3.1	变量赋值语句	63
3.3.2	信号赋值语句	63
3.3.3	多驱动源信号——决断信号	65
3.3.4	IF 语句	66
3.3.5	CASE 语句	70
3.3.6	NULL 语句	74
3.3.7	LOOP 语句	74
3.3.8	NEXT 语句与 EXIT 语句	77
3.3.9	过程调用语句与 RETURN 语句	78
3.3.10	断言语句与 REPORT 语句	78
3.4	并行语句	79
3.4.1	块语句	80
3.4.2	并行信号赋值语句	80
3.4.3	并行过程调用语句	83
3.4.4	并行断言语句	83
3.4.5	元件例化语句	84
3.4.6	生成语句	88
3.5	本章小结	90
3.6	习题	90
第 4 章	VHDL 深入	92

4.1	子程序	92
4.1.1	函数	92
4.1.2	过程	94
4.2	程序包和设计库	95
4.2.1	程序包	96
4.2.2	预定义程序包	96
4.2.3	十字路口交通信号灯控制器	98
4.3	重载	103
4.3.1	子程序重载	103
4.3.2	运算符重载	107
4.4	决断信号与决断函数	108
4.4.1	决断信号的声明	108
4.4.2	决断函数	108
4.5	配置	111
4.5.1	默认连接和默认配置	112
4.5.2	元件配置	114
4.5.3	结构体中声明的元件配置	118
4.5.4	块的配置	119
4.6	本章小结	122
4.7	习题	122
第 5 章 VHDL 描述的实现		125
5.1	EDA 集成软件 Quartus II	125
5.1.1	安装 Quartus II	125
5.1.2	设置授权文件路径	129
5.2	VHDL 描述的硬件实现	131
5.2.1	创建工程项目文件	132
5.2.2	输入设计文件	134
5.2.3	器件设置	136
5.2.4	编译设计项目	138
5.2.5	仿真设计项目	142
5.2.6	下载编程	146
5.3	本章小结	151
第 6 章 典型电路描述实例		152
6.1	组合逻辑电路描述实例	152
6.1.1	BCD 码——7 段 LED 显示译码器	152
6.1.2	4 位数值比较器	154
6.1.3	双 4 位缓冲器	155

6.1.4	8 位双向缓冲器	156
6.2	触发器描述实例	157
6.2.1	主从式 J-K 触发器	157
6.2.2	D 触发器	159
6.3	时序逻辑电路描述实例	161
6.3.1	整数分频器	161
6.3.2	串行输入、并行输出移位寄存器	162
6.3.3	并行输入、串行输出移位寄存器	164
6.3.4	单脉冲发生器	167
6.3.5	波形发生器	168
6.3.6	HDB3 编码器	174
6.4	本章小结	180
第 7 章 常用程序包		181
7.1	STD 库中的程序包	181
7.1.1	标准程序包 STANDARD	181
7.1.2	文本输入/输出程序包 TEXTIO	183
7.2	IEEE VHDL 库中的常用程序包	185
7.2.1	标准逻辑程序包 STD_LOGIC_1164	185
7.2.2	标准逻辑算术程序包 STD_LOGIC_ARITH	219
7.2.3	标准逻辑无符号数组扩展程序包 STD_LOGIC_UNSIGNED	224
7.2.4	标准逻辑带符号数组扩展程序包 STD_LOGIC_SIGNED	231
7.3	本章小结	239
附录 A VHDL 保留字		240
附录 B VHDL 预定义属性		241
B.1	类型和子类型的属性	241
B.2	数组的属性	241
B.3	其值为信号值的属性	242
B.4	其值与信号有关的属性	242
B.5	为块和设计实体声明的属性	243
参考文献		244

第1章 概 论



学习目标

本章主要介绍数字集成电路、可编程逻辑器件、电子设计自动化技术、数字系统的设计流程、知识产权核、优化设计等基本概念，使读者对诸如双极型工艺与 MOS 工艺，通用集成电路与专用集成电路，全定制制造方法与半定制制造方法，CPLD 与 FPGA，CAD、CAE 与 EDA，自顶向下和自底向上的设计方法，描述、划分、综合与验证等设计工作，软件 IP、固件 IP 与硬件 IP，优化资源利用率、优化工作速度与优化布局布线等名词有一个初步的认识。

学 习 重 点

数字集成电路

数字系统设计流程

可编程逻辑器件

知识产权核

电子设计自动化

优化设计

当今世界，电子技术飞速发展，新器件和新产品不断涌现，人类已进入到数字化时代。从家用电器到电子计算机，从通信设备到医疗仪器，数字技术已经渗透到人们生活的诸多领域。例如，数码相机、数码摄像机、DVD 播放机、数字电视机顶盒、3G 手机、程控交换机、B 超、计算机 X 射线断层扫描 (CT)、核磁共振成像 (MRI) 等仪器及产品已经和人们的生活息息相关。

在电子技术领域，通常将使用数字技术传输和处理信息的电子系统称为数字系统，数字系统的主要硬件构成是数字集成电路。图 1-1 为一款 CDMA-1X 手机的印刷电路板 (PCB) 视图，可以看出，PCB 上焊接有大大小小的黑色模块，这就是集成电路 (Integrated Circuit, IC) 芯片。

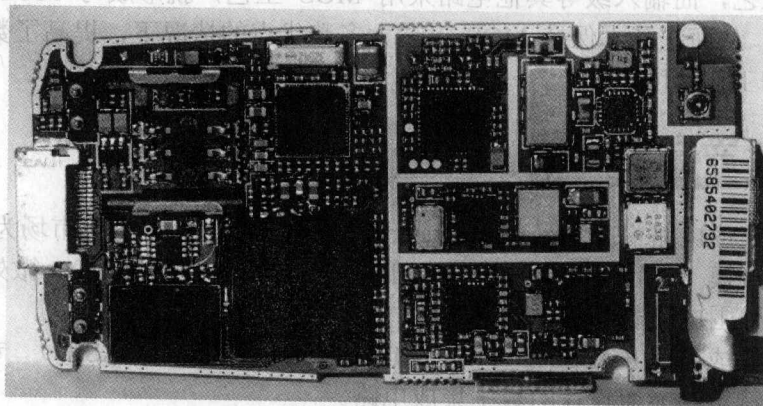
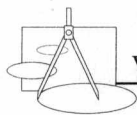


图 1-1 一款 CDMA-1X 手机的 PCB 视图



1.1 数字集成电路分类

数字集成电路是数字系统的主要硬件构成，它有多种分类方式。下面只列举了按照不同的生产工艺、生产目的和制造方法的分类情况。

1.1.1 按生产工艺分类

- 双极型(Bipolar)集成电路 常见的有晶体管——晶体管逻辑(Transistor Transistor Logic, TTL)、发射极耦合逻辑(Emitter Coupled Logic, ECL)和集成注入逻辑(Integrated Injection Logic, I²L)等几种工艺。
- 金属氧化物半导体(Metal Oxide Semiconductor, MOS)集成电路。常见的有 p 沟道 MOS (pMOS)、n 沟道 MOS (nMOS) 和互补型 MOS (CMOS) 3 种工艺。
- 双极与 MOS 混合(BiMOS)集成电路。同时含有双极型晶体管和 MOS 场效应管, 结合双极型晶体管的高跨导、强驱动能力、高频低噪声和 MOS 场效应管的高集成度、低功耗、抗干扰能力强等优点, 制造出高速、高集成度和高性能的器件。

与 MOS 工艺相比, 双极型工艺流程复杂, 其功耗大、集成度低、生产成本低, 但工作速度较快。

在双极型工艺中, ECL 工艺最为复杂, 功耗大、集成度低、生产成本低, 但工作速度最快。I²L 工艺电路结构简单、集成度高、功耗低, 但是输出电压幅度小, 抗干扰能力较差, 而且工作速度较低。目前 I²L 工艺主要用于制作大规模集成电路的内部逻辑电路。TTL 工艺则介于二者之间, 是目前应用较多的双极型工艺。

在 MOS 工艺中, CMOS 工艺结合了 pMOS 和 nMOS 两种工艺, 所以称之为互补型 MOS (Complementary MOS)。CMOS 工艺与 pMOS 和 nMOS 工艺相比, 具有功耗小、速度快、抗干扰能力强和工作电压范围宽等优点, 虽然 CMOS 工艺比 pMOS 和 nMOS 工艺复杂, 但仍比双极型工艺简单, 因此 CMOS 集成电路是目前应用最为广泛的集成电路。

如果利用双极型工艺驱动能力强(当然功耗也大)的特点, 在集成电路中输出级采用双极型工艺, 而输入级等其他电路采用 MOS 工艺, 就形成与 MOS 工艺混合的 BiMOS 工艺。从而在基本保持 MOS 工艺大多数优点的情况下, 提高了数字集成电路的驱动能力和工作速度。

1.1.2 按生产目的分类

- 通用集成电路(Universal Integrated Circuit, UIC) 以供应市场为目的。例如, 中小规模(SSI/MSI)标准逻辑电路(74 系列、4000 系列)、微处理器、存储器、外围电路芯片等。
- 专用集成电路(Application Specific Integrated Circuit, ASIC) 专门为某种或几种特定功能而设计的数字集成电路。

在通用集成电路中, 各种逻辑门、触发器、编码/译码器、多路转换器、寄存器、



计数器和小容量存储器等逻辑器件,被制作成SSI/MSI的标准产品,其生产批量大、成本低、器件工作速度快,是数字系统在传统设计中最为常用的逻辑器件。但由于这类器件的集成度低,由它们构成的数字系统所用的芯片数量多,系统的硬件规模大,印刷电路板面积大、走线复杂、焊点多,从而导致系统的可靠性降低、功耗增大。另外,这类器件的功能确定,用户无法修改,系统的保密性低,而且印刷电路板制成后,修改设计也很困难。

后来出现的大规模/超大规模(LSI/VLSI)通用集成电路,例如微处理器、单片机、存储器和可编程外围电路芯片等,具有集成度高、功耗较小的优点,而且很多器件的逻辑功能可以由软件进行配置,因此在很大程度上减小了数字系统的硬件规模,系统的可靠性和灵活性也大大提高了。但这类器件的工作速度不高,而且仍需要若干SSI/MSI标准集成电路与之配合才能构成完整的系统。

专用集成电路(ASIC)是专门为某种或几种特定功能而设计制造的,其集成度高、功耗小、工作速度快,一片ASIC能够代替一块包含若干片通用集成电路的印刷电路板,甚至一个完整的数字系统。所以,ASIC可以大大降低设备价格,缩短研制周期,简化数字系统的生产过程,降低功耗,减少体积,减轻重量,提高设备的可靠性,同时也使得设备难以被仿制。目前,在数字系统中,已大量采用ASIC来简化系统设计,提高数字系统的可靠性和降低成本。

1.1.3 按制造方法分类

- 全定制方式(Full-Custom Design Approach) 芯片的各层掩膜都是按特定电路功能专门设计制造的,设计者综合考虑了芯片版图的布局布线等技术细节,使芯片的性能、面积、功耗和成本等指标达到最佳,从而使得设计周期变长,设计成本提高,而且风险大。因此全定制方式只适用于设计成熟、生产批量非常大的场合。全定制方式既适用于通用集成电路的制造,也适用于专用集成电路(ASIC)的制造。

- 半定制方式(Semi-Custom Design Approach) 设计者在集成电路制造厂商提供的半成品(例如通用母片、可编程逻辑器件等)的基础之上增加互连线掩膜或者设定逻辑功能,从而缩短设计周期、降低设计成本。半定制方式适用于生产批量不大的场合。

按照不同的逻辑实现方法,半定制方式主要有门阵列法、门海法、标准单元法和可编程逻辑器件法四类。

- 门阵列(Gate Array)法 用大量规则排列的预制门阵列形成电路中的基本门电路,例如与非门、或非门、反相器、传输门或其他电路单元等。在门阵列之间留有布线通道,从而构成门阵列母片。设计者在门阵列母片上按不同的电路功能追加金属连线的掩膜,最终完成芯片的电路设计。由于事先留下的每一布线通道的布线容量有限,如果连线过多则布通率就会下降。虽然可以通过增加金属连线层来提高布通率,但这会降低芯片面积的利用率。
- 门海(Sea of Gate)法 又称为无通道门阵列法,与有通道门阵列法的区别是,在母片中没有设置布线通道,如果位于门电路之间的金属连线通过某个门单元



的话,那么该门单元就作废了。因此门海法是以牺牲门阵列中的门电路单元来换取布通率的。如果连线太多,也可以采用增加布线层(单独设计金属连线层,大数目的掩模)的办法来提高布通率。

门阵列法和门海法通常用于 ASIC 的制造,与全定制方式相比,其设计和生产周期短、成本低、风险小,比较适用于生产批量不太大的场合。但是其芯片面积利用率低、布通率低、灵活性差,目前已经较少使用。

● 标准单元(Standard Cell)法也称为多元胞(Polycell)法,它是将各种电路元件或电路模块,在物理版图级按照最佳设计原则构造成等高不等宽的“标准单元”。布局时将标准单元按行排列,在拼接过程中,同一行的标准单元的电源和地线自动连接在一起。行与行之间留有布线通道,其版图布局示意图如图 1-2 所示。同行或者相邻行的单元相连,可以通过上下两个布线通道完成。隔行单元之间的垂直方向互连则借用标准单元中预留的“走线道”来完成,也可以在两个标准单元之间设置“走线道单元”或者“空单元”来完成连线。标准单元法的优点是灵活性好、设计效率高、布通率高(可以达到 100%),可以使设计者更多地关注电路的性能和优化等问题。但是其设计周期和制造成本要高于门阵列法和门海法。

● 可编程逻辑器件(Programmable Logic Device, PLD)法 芯片的各层均已由厂家制造完毕,但其逻辑功能却并未确定,设计者可以使用 EDA 工具按照自己的设计来对芯片进行设定(俗称编程),以实现特定的逻辑功能。这种方式对生产厂家而言是制造通用集成电路,可以批量生产以降低成本;而对数字系统设计者而言,可以按照不同的设计来对芯片进行不同的编程而使其成为专用集成电路。PLD 具有集成度高、工作速度快、设计周期短、成本低和保密性强等优点,而且大多数 PLD 还可以重复编程。因此 PLD 的出现改变了数字系统的传统设计方法,成为实现新型数字系统的理想器件。但是 PLD 的制造成本依然远高于全定制方式,所以在电路设计成熟、生产批量很大的情况下,通常还是采用全定制方式。PLD 更适用于产量小的试制产品,特别适用于设计开发阶段和研制样机的场合。

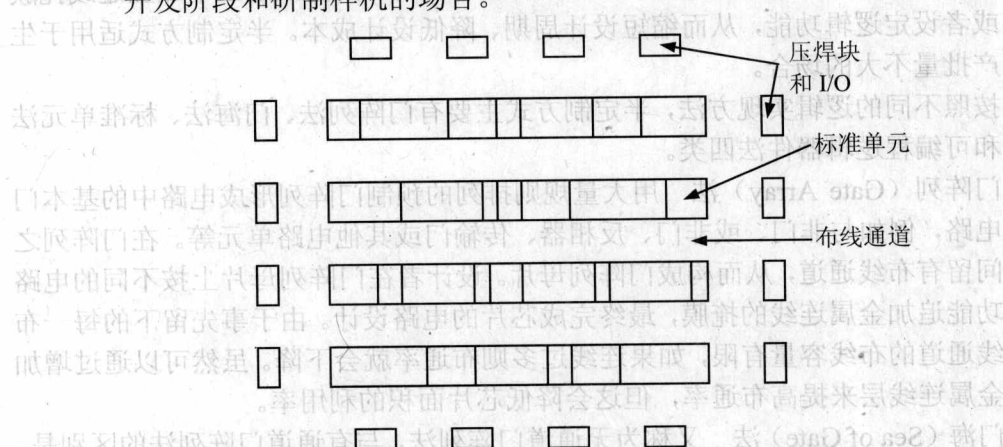


图 1-2 标准单元法的版图布局示意图——遵循等高不等宽原则

1.2 可编程逻辑器件简介

数字系统的设计方式通常采用“积木式”方法进行设计。

传统方法是选择各种固定功能的通用集成电路作为“积木”，然后按照所设计的系统功能搭建成数字电路。这种方法所需要的芯片种类多而且数量大，设计者选择器件的局限性也很大，无设计的灵活性可言。

ASIC 的出现使构成数字系统的“积木”大为简化。在多数情况下，构成数字系统仅需要三类“积木”，即 CPU+MEMORY+ASIC 的模式。甚至在个别情况下，一个数字系统的电路可以仅由一片 ASIC 构成。使用全定制 ASIC 和除 PLD 之外的半定制 ASIC 制造方式，一旦制造完毕则无法修改，因此这些制造方式在数字系统的设计研制和现场升级等方面不如 PLD 灵活。随着 VLSI 的发展，PLD 的集成度和工作频率不断提高，成本却在逐渐下降，所以 PLD 将被广泛应用于各种数字系统中。

1.2.1 PLD 的分类

PLD 有多种分类方法，下面介绍几种常见的分类方法。

1. 按照不同的结构划分

- PLD 乘积项结构器件，其基本结构为与或阵列。PLD 的基本组成如图 1-3 所示。
- FPGA 有查找表结构、多路开关结构和多级与非门结构 3 种类型的器件。



图 1-3 PLD 基本组成原理结构图

(1) 查找表结构器件是由简单查找表组成的可编程门阵列，其查找表单元如图 1-4 所示。Xilinx 公司的 XC 系列 FPGA 和 Altera 公司的 FLEX8000 系列 FPGA 等都采用了查找表结构。

(2) 多路开关结构为可编程的多路开关，其多路开关逻辑块如图 1-5 所示。

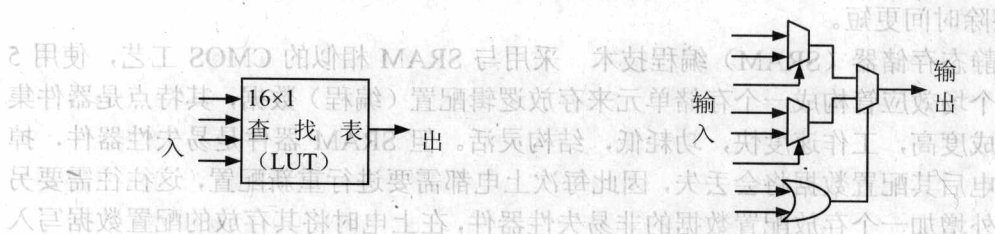
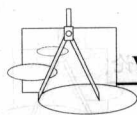


图 1-4 查找表单元原理结构图

图 1-5 多路开关逻辑块原理结构图



(3) 多级与非门结构是基于一个与-或-异或逻辑块,如图 1-6 所示。在逻辑块的基础上增加了触发器和多路开关来扩展输出功能。这种结构与 EPLD 非常类似,因此也有人将其归类于 PLD,而不是 FPGA。Altera 公司的许多 FPGA 产品就采用了多级与非门结构。

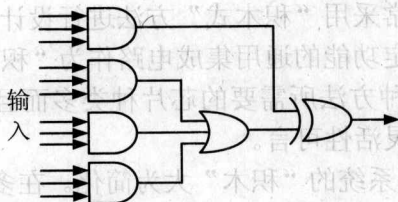


图 1-6 与-或-异或逻辑块原理结构图

由于可编程逻辑器件是从乘积项结构器件起步,逐步发展丰富而来的,所以狭义的 PLD 特指乘积项结构器件,而广义的 PLD 则包含所有 PLD 和 FPGA。

2. 按照不同的制造工艺划分

- 熔丝 (Fuse) / 反熔丝 (Antifuse) 编程技术 双极型熔丝工艺的特点是器件工作速度快、功耗大、集成度低。而在反熔丝工艺中,可编程低阻电路元件 (PLICE) 工艺是采用高电压击穿夹在两层导体之间的 PLICE 介质来实现连通机制,某集成度比熔丝工艺要高; Vialink 和 Micro Via 工艺大都采用高压击穿的原理来实现连通机制。采用熔丝/反熔丝工艺的器件,其编程都是一次性的,编程后则无法修改,因此这类器件适用于已定型的设计。

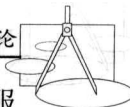
- 浮栅编程技术 包括 UVC MOS、E²CMOS 和 Flash CMOS 工艺,其特点是可以重复擦除和编程,并且是非易失性器件。

(1) UVC MOS 工艺的特点是器件集成度高、功耗低,具有可擦除、可重复编程的能力,但工作速度慢,擦除时间长,需采用价格较昂贵的石英窗口封装和专门的紫外线擦除设备,目前已很少使用。

(2) E²CMOS 工艺使用电擦除的方法,擦除时间短,可以很方便地反复擦除和改写,克服了 UVC MOS 工艺器件擦除时间长和成本高的缺点,而且其功耗比 UVC MOS 工艺器件更低,工作速度也更快。因此,除了集成度稍逊于 UVC MOS 工艺器件以外,E²CMOS 工艺是一种先进和成熟的 PLD 制造工艺。

(3) Flash CMOS 工艺采用读写块设备的方式,一次擦除一个数据块,比 E²CMOS 工艺的擦除时间更短。

- 静态存储器 (SRAM) 编程技术 采用与 SRAM 相似的 CMOS 工艺,使用 5 个场效应管构成一个存储单元来存放逻辑配置 (编程) 数据,其特点是器件集成度高,工作速度快,功耗低,结构灵活。但 SRAM 器件是易失性器件,掉电后其配置数据将会丢失,因此每次上电都需要进行重新配置,这往往需要另外增加一个存放配置数据的非易失性器件,在上电时将其存放的配置数据写入 SRAM 中。



在上述3种编程技术中,熔丝/反熔丝技术属于一次性编程技术,反熔丝技术克服了熔丝技术的一些缺陷,目前常用于小批量的定型设计中;浮栅技术属于反复擦写器件,而且具有非易失性,其中 E^2 CMOS器件优于UVC MOS器件,成为常用的PLD器件制造技术,Flash CMOS工艺也有很好的发展前景;SRAM技术则具有比前两种技术集成度更高、工作速度更快、功耗和制造成本更低等优势,成为当前FPGA常用的制造工艺。但它具有易失性,需要在每次上电时重新配置数据,正因为如此才将FPGA称其为现场可编程门阵列,即每次上电时在现场“编程”。

3. 按照不同的集成度划分

- 低密度器件 包括 PROM、PLA、PAL 和 GAL 器件。
- 高密度器件 包括 EPLD、CPLD 和 FPGA 器件。

PROM 是可编程只读存储器 (Programmable Read-Only Memory),最初作为计算机存储器设计而使用,后来发现它具有组合 PLD 的功能。它的与-或阵列结构为“与”阵列固定、“或”阵列可编程,其特点是价格低、编程容易,适合于存储函数和数据表格,但工作速度慢,芯片面积的利用率低,目前只用于一些特定场合。

图 1-7 是一个 4×2 PROM 的 PLD 与-或阵列结构图。图 1-8 是用 4×2 PROM 构成的一个半加器的内部结构:

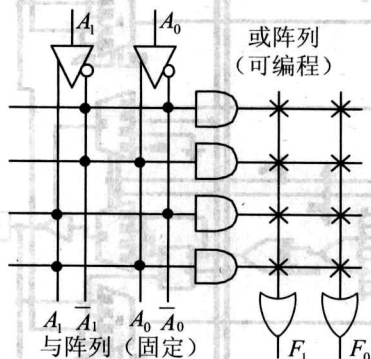
$$F_0 = A_0 \bar{A}_1 + \bar{A}_0 A_1, \quad F_1 = A_0 A_1$$


图 1-7 PROM 的与-或阵列结构图

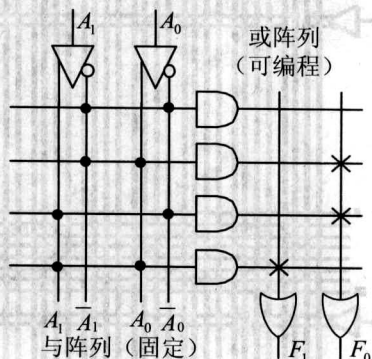


图 1-8 由 PROM 构成的半加器内部结构图

PLA 是可编程逻辑阵列 (Programmable Logic Array),其与-或阵列结构为与、或阵列均可编程,虽然它的工作速度比 PROM 要快,但由于器件的资源利用率低,编程复杂,目前已很少使用。图 1-9 是一个 4×2 PLA 的阵列结构图。

PAL 是可编程阵列逻辑 (Programmable Array Logic),其与-或阵列结构为“与”阵列可编程、“或”阵列固定,特点是器件工作速度快、价格低、编程容易,并且在输出级加上了输出寄存器单元,从而实现可编程时序逻辑电路。为了适应不同的输出要求,衍生出多种不同输出结构的 PAL 产品,种类十分丰富,同时也提高了对应用工程师的要求。图 1-10 是一个 4×2 PAL 内部的与-或阵列结构图。图 1-11 是一种 PAL 器件 PAL16V8 的部分结构图,其输出端是一种带有反馈和异或逻辑的寄存器结构。

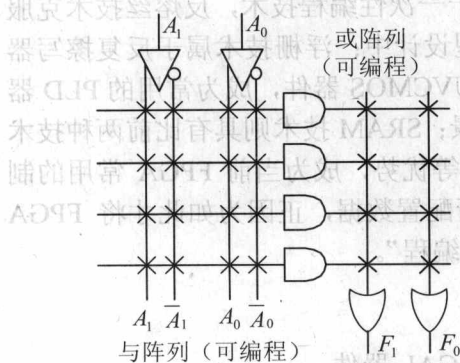
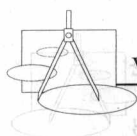


图 1-9 PLA 的与-或阵列结构图

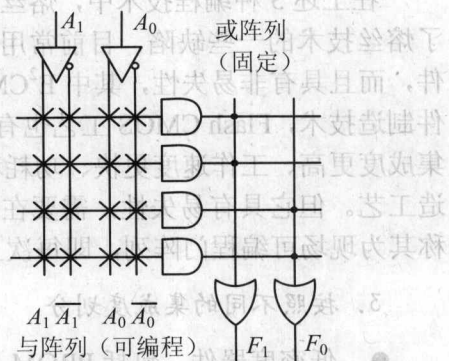


图 1-10 PAL 的与-或阵列结构图

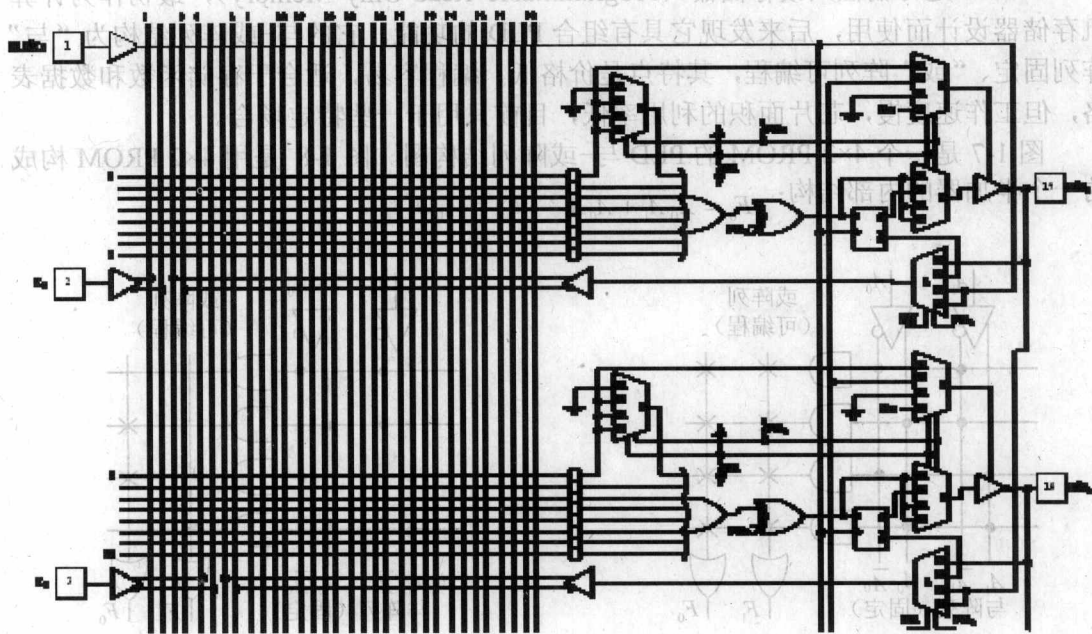


图 1-11 PAL16V8 的部分内部结构图

特别要提出的是在 PAL 基础上改进的通用阵列逻辑 (Generic Array Logic, GAL) 器件, 由于增加了可编程输出逻辑宏单元 (Output Logic Macro Cell, OLMC) 结构和采用了 E²CMOS 工艺, 因此 GAL 器件具有结构灵活、功耗低、工作速度快、易于加密、可电擦除和可重复编程等优点, 已成为目前低密度 PLD 的主要产品而替代了 PAL 器件。图 1-12 是 OLMC 的内部结构示意图。图 1-13 是一种 GAL 器件 GAL16V8 的内部结构图。

通过改变 OLMC 中 4 个多路选择器的状态, 可以将 I/O 端口设置为 5 种模式之一: 输入端口、组合逻辑输出端口、组合逻辑输入/输出双向端口、时序逻辑输出端口和时序逻辑输入/输出双向端口, 并且可以通过改变参与异或操作的 XOR(n) 来决定送往输出端口的逻辑值是否反相。

