

应用型本科人才培养创新教材出版工程

电子信息类教材系列

可编程逻辑器件与 VHDL语言

主 编 程云长
副主编 王莉莉 陈立力



 科学出版社
www.sciencep.com

●应用型本科人才培养创新教材出版工程

电子信息类教材系列

可编程逻辑器件与 VHDL 语言

主 编 程云长

副主编 王莉莉 陈立力

科 学 出 版 社

北 京

内 容 简 介

本书内容分三部分:可编程逻辑器件 CPLD 和 FPGA 的基本原理;使用上述两种器件要用到的硬件描述语言,即 VHDL 语言;Altera 公司推出的广为流传的数字电路开发工具——MAX+plus II。书中提供了大量实例论述 VHDL 语言的基本知识和使用 VHDL 语言设计逻辑电路的基本方法,并配有 8 个实验供读者学习。

本书可作为高等院校教材,也适合有一定基础的电子工程设计人员阅读参考。

图书在版编目(CIP)数据

可编程逻辑器件与 VHDL 语言/程云长主编. —北京:科学出版社,2005
(应用型本科人才培养创新教材出版工程·电子信息类教材系列)
ISBN 7-03-016103-3

I. 可… II. 程… III. ①可编程逻辑器件-高等学校:技术学校-教材
②硬件描述语言, VHDL—高等学校:技术学校-教材 IV. ① TP332.1
② TP312

中国版本图书馆 CIP 数据核字(2005)第 090745 号

责任编辑:余 丁 王日臣 / 责任校对:李奕莹
责任印制:安春生 / 封面设计:陈 敬

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>

双青印刷厂 印刷

科学出版社发行 各地新华书店经销

*

2005 年 9 月第 一 版 开本:B5(720×1000)

2005 年 9 月第一次印刷 印张:21

印数:1—4 000

字数:398 000

定价:26.00 元

(如有印装质量问题,我社负责调换(环伟))

前 言

在电子设计技术领域,可编程逻辑器件的广泛应用为数字系统的设计带来极大的灵活性。由于该器件可以通过软件编程而对其硬件的结构和工作方式进行构造,使得硬件设计可以如同软件设计那样方便快捷。这一切极大地改变了传统的数字系统设计方法、设计过程,乃至设计观念。

为了适应当代电子技术领域的设计观念,学习当代电子技术领域的设计方法,研究当代电子技术领域的设计过程,本书主要向读者介绍三个方面的内容:

(1) 可编程逻辑器件的基本概念,安排在第2章内。这一章的内容首先介绍了可编程逻辑的发展历史,然后分别介绍可编程逻辑器的两种基本类型:CPLD和FPGA的内部组成。使读者了解逻辑阵列块(LAB)、可配置逻辑块(CLB)、宏单元、I/O控制块等组件的工作原理。为今后使用CPLD或FPGA而设计外围接口电路时有一个正确的指导思想。

(2) 硬件描述语言VHDL。由于VHDL语言是美国国防部确认的标准硬件描述语言,所以各个可编程逻辑器件的生产厂家所提供的开发工具,都支持这一语言环境。学习了VHDL这门通用性强、功能描述能力更强的硬件描述语言,读者就可以面对绝大部分厂家的器件。这部分内容安排在第3章。本章使用了大量实例介绍VHDL语言规则和设计方法,对读者的学习会有很大帮助。

(3) 以Altera公司推出的数字电路开发工具——MAX+plus II为实例,详细介绍它在开发可编程逻辑器件过程中的使用方法,以期达到举一反三的作用,内容安排在第4章。最后还给读者提供了8个实验题目作为第5章的内容。

感谢王莉莉老师和陈立力的通力合作,使本书得以完成。由于时间紧迫,书中不妥之处在所难免,恳请读者批评指正。

目 录

前言

第 1 章 绪论	1
1.1 现代数字系统	1
1.1.1 数字系统的概念	1
1.1.2 数字系统的实现	1
1.1.3 数字系统的设计工具	3
1.2 EDA 技术的发展	4
1.3 现代数字系统的实现方法	5
思考题	6
第 2 章 可编程逻辑器件	7
2.1 可编程逻辑器件的发展历程	7
2.2 可编程逻辑器件的基本结构与分类	9
2.3 简单可编程逻辑器件	10
2.3.1 可编程逻辑阵列	11
2.3.2 可编程阵列逻辑和通用阵列逻辑	12
2.4 复杂可编程逻辑器件	14
2.4.1 CPLD 的基本结构	15
2.4.2 MAX7000 系列器件的主要特点及性能	16
2.4.3 MAX7000 器件结构	18
2.4.4 MAX7000 器件的配置	25
2.4.5 MAX7000 器件的编程	26
2.5 现场可编程门阵列 FPGA	28
2.5.1 FPGA 的基本结构	28
2.5.2 XC4000 系列器件的主要特点及性能	30
2.5.3 XC4000 的器件结构	31
2.5.4 可编程互连	51
2.5.5 XC4000 器件的编程	60

2.6 选择 CPLD 还是 FPGA	65
思考题	66
第 3 章 VHDL 硬件描述语言	67
3.1 VHDL 概述	67
3.1.1 VHDL 简介	67
3.1.2 VHDL 的基本结构及其优点	72
3.1.3 VHDL 术语	75
3.2 VHDL 程序结构	76
3.2.1 实体	76
3.2.2 结构体	80
3.2.3 程序包、库和配置	92
3.3 VHDL 的数据类型及运算操作符	101
3.3.1 VHDL 的数据对象	101
3.3.2 VHDL 词法规则与标识符	105
3.3.3 VHDL 的数据类型	107
3.3.4 VHDL 的运算操作符	117
3.4 VHDL 的描述语句	121
3.4.1 顺序语句	122
3.4.2 并行语句	136
3.4.3 属性语句	151
3.5 基本电路的 VHDL 模型	154
3.5.1 组合逻辑电路的设计	155
3.5.2 时序逻辑电路的设计	162
3.5.3 存储器	171
3.6 有限状态机	177
3.6.1 有限状态机的基本模型	178
3.6.2 状态机的建立过程	183
3.6.3 状态机的状态编码	185
3.6.4 状态机剩余状态处理	188
3.7 仿真与逻辑综合	189
3.7.1 仿真	189

3.7.2 延时模型	191
3.7.3 仿真 Δ	192
3.7.4 仿真激励信号的产生	193
3.7.5 综合	199
思考题	201
第4章 MAX+plus II 设计系统	204
4.1 MAX+plus II 简介	204
4.2 用 MAX+plus II 开发数字系统的流程	209
4.3 设计输入	216
4.3.1 文本输入方式	216
4.3.2 图形输入方式	221
4.3.3 创建和编辑图元	227
4.3.4 波形输入方式	230
4.3.5 使用其他 EDA 工具生成的文件进行输入	236
4.4 层次化设计	236
4.5 MAX+plus II 的工程设置	238
4.6 MAX+plus II 的工程编译	245
4.6.1 MAX+plus II 编译器简介	245
4.6.2 编译设置	247
4.6.3 工程编译	256
4.6.4 消息处理器	257
4.6.5 查看报告文件	258
4.7 器件适配的查看与编辑	259
4.7.1 平面编辑器简介	259
4.7.2 ACF 和 FIT 文件	264
4.7.3 查看和编辑适配结果	267
4.8 工程仿真	273
4.8.1 仿真简介	273
4.8.2 文件操作	275
4.8.3 运行仿真	280
4.8.4 仿真结果分析	282

4.9 时序分析	286
4.9.1 时序分析简介	286
4.9.2 时序分析设置	288
4.9.3 运行时序分析	291
4.10 编程下载	296
4.10.1 编程文件	296
4.10.2 程序下载方式	300
4.10.3 下载编程与配置文件	302
4.10.4 下载编程的相关操作	305
4.11 Quartus II 简介	308
思考题	310
第 5 章 数字电路和系统实验	311
实验一 3-8 线译码器	311
实验二 同步计数器	312
实验三 BCD 码全加器	313
实验四 4 位乘法器	314
实验五 FIFO 设计	315
实验六 七段数码显示	316
实验七 电子秒表的实现	317
实验八 数字闹钟的实现	318
参考文献	320
附录 A EPM7096 管脚封装资料	321
附录 B XC4005E/XL 管脚封装资料	324

第 1 章 绪 论

1.1 现代数字系统

1.1.1 数字系统的概念

20 世纪 80 年代以来,随着电子技术,尤其是计算机软硬件技术和大规模集成电路设计制造技术的迅速发展,人类已进入数字化时代。由于数字逻辑技术在处理和传输信息方面的各种优点,数字技术已经进入人类生活的各个领域,从计算机到人们日常生活中常用的消费类电子产品(如数字电话、家用影音设备等),从民用产品到军用仪器设备,处处都可以看到数字技术的身影。

现在的电子设备中,单纯的模拟电路已经很少见了,通常只在完成信号放大、大功率输出、频率变换、电源基准以及数/模转换等特定功能时使用模拟电路,其余部分均采用数字电路。因此,对大多数电子设备而言,其主体部分是由数字系统构成的。

数字电路是由具有各种逻辑功能的数字逻辑器件组成的。数字电路按其结构可分成组合逻辑电路和时序逻辑电路两大类。逻辑器件有很多种,常用的包括基本门电路、编码器、译码器、加法器、乘法器、触发器和计数器等,其中计数器是数字电路中应用最广泛的数字逻辑器件。

本书把由数字电路实现的处理和传输信息的系统都称为数字系统。类似于计算机系统,数字系统通常包括控制器、处理器和存储器三大部分。一般的,只有包含了控制器的数字电路才被称为数字系统。

1.1.2 数字系统的实现

数字系统经历了由分立元件、SSI(小规模集成电路)、MSI(中规模集成电路)到 LSI(大规模集成电路)、VLSI(超大规模集成电路)的过程。传统的数字系统像其他电子系统一样,往往采用搭积木式的方法实现,先由器件搭成电路板,再用电路板搭成电子系统。早期的数字系统中,其最基本的“积木块”是具有固定功能的通用标准集成电路,如 74 系列(TTL)、4000 系列(CMOS)芯片以及一些可以实现比较复杂功能的大规模集成电路。用户只能根据需要从这些集成电路中选择相应的器件组成系统。因此,早期的数字系统体积大、重量大、功耗大、生产周期长、成本高。更重要的是由于单个 SSI、MSI 器件仅能完成简

单的逻辑功能, 实现一个复杂功能的系统需要使用大量的 SSI、MSI 器件, 其结果是整个系统的可靠性很差。可靠性差的原因主要有两个: 一是由于所用器件多, 需要连接的环节就会多, 所以出现故障的机会大大增加; 二是每个器件有一定的失效时间, 一个系统所用器件数量多, 其平均失效时间将大大下降。

随着半导体技术发展, 传统的数字系统发生了质的飞跃, 而微控制器 (MCU) 或称单片机的出现, 使数字系统发生了革命性的变化。

微控制器是在单一芯片上实现的 CPU, 它是一种通用型器件, 只需配以一定的程序 (通常固化于外部的 ROM 中) 和一些外围电路, 即可任意实现复杂的逻辑功能。用微控制器做成的系统避免了早期数字系统的缺点。随着集成电路制造工艺的发展, 出现了嵌入式 EEPROM/FLASH 制造工艺, 利用这些新工艺, 新一代的微控制器可以直接将程序存入微控制器中, 并且程序还可以根据需要进行修改, 这使得系统的器件数量更少, 而使用却更加灵活。现在, 基于通用微控制器进行程序开发已经成为数字系统设计最常用的方法。显而易见, 基于通用微控制器进行数字系统设计的方法具有灵活高效、成本低、系统开发风险小以及可移植性好等诸多优点。微控制器的出现从根本上改变了传统的数字系统设计方法, 是对传统数字系统的一次革命。

随着信息技术的迅速发展, 人们对电子设备尤其是数字系统有了更高的要求。基于微控制器的设计方法的缺点也逐步显现: 其一是系统速度低, 系统完成一个操作, 用硬件实现时通常只需若干级器件延迟时间即可完成, 而微控制器采用软件实现时却需运行一段程序, 耗费若干时钟周期才可完成, 显然基于微控制器的数字系统并不适用于对系统速度要求很高的场合; 其二是基于微控制器的系统存在着程序易受干扰而使系统瘫痪的问题; 其三, 目前尚无可靠的方法对程序进行加密使之免遭破解, 而基于微控制器的数字系统其核心在于程序的设计开发, 因此整个系统很容易为他人复制。

针对微控制器的缺点, 专用集成电路 (ASIC), 即为某一特定数字系统设计、生产的集成电路越来越受到人们的欢迎。一个复杂的数字系统只要一片或数片 ASIC 即可实现, 因而制成的设备体积小、重量轻、功耗低、速度高、成本低。相对于程序而言, ASIC 不易被破解, 可保护设计者的成果不被盗用。从市场的角度看, 一个数字系统就是一个产品, 对于市场需求量大且生命周期长的成熟的产品可以采用 ASIC 方法实现, 这样可以将产品的成本降至最低。目前, ASIC 已经成为现代数字系统设计的主要手段。但 ASIC 的设计开发和生产需要比较长的时间, 设计费用高昂, 并且生产完成后不易修改。

对于生产批量少或处于试制阶段的产品, 利用 ASIC 实现显然并不是最好的选择, 而可编程逻辑器件 (PLD) 的出现很好地解决了这一问题。PLD 的优点在于产品研制时间短, 设计费用低, 风险接近于零。PLD 的出现及其迅速发展,

使本来是微电子工程师才能完成的 ASIC 设计, 完全转移给了电路设计工程师。这就大大缩短了数字系统设计的时间。

基于可编程逻辑器件的设计可以对器件直接进行设计, 通过设计芯片来实现系统功能。这种基于芯片的设计方法允许设计者定义、编辑和修改器件的内部逻辑和管脚, 使得硬件设计变得和软件设计一样方便。同时, 这种方法也将原来由电路板设计完成的大部分工作放在芯片设计中进行, 这样不仅可以通过芯片设计实现多种数字逻辑系统功能, 而且由于管脚定义的灵活性, 大大减轻了电路板设计的工作量和难度, 从而有效地增强了设计的灵活性, 提高了工作效率。另外可编程逻辑器件的价格越来越低, 使它的应用范围也不仅局限于产品试制阶段, 而是被直接用于产品生产中。

当今的数字系统需要完成更多复杂的功能, 同时需要更快的速度和更高的性能, 因此单纯的使用微控制器或是 ASIC/PLD 已经无法满足设计要求。为了高效地实现数字系统, 人们越来越多地采用软硬件协同设计的方法进行系统设计。同时, 集成电路制造工艺的迅速发展使微控制器嵌入 ASIC 或 PLD 成为现实, 因此, 一颗芯片即可完成整个系统功能, 这就是近几年得到迅速发展的片上系统 (SoC) 设计技术。

1.1.3 数字系统的设计工具

在现代电子系统领域, EDA 工具的使用大大提高了设计的效率和准确度, 并已逐渐成为电子系统设计的最重要手段。随着大规模集成电路制造工艺的不断发展以及市场的需求, 逻辑芯片和数字系统的规模和设计的复杂程度都在快速增长, 用手工方式已经无法完成逻辑芯片和数字系统的设计。现在, 数字系统的设计工作不仅需要完成功能设计, 而且还需要对时序、功耗等性能指标进行优化, 仅仅依靠手工是无法完成设计工作的, 所有的设计工作都需要在计算机上通过专用的 EDA 工具完成。

EDA (electronic design automation) 即电子设计自动化, 这里所说的自动化是指利用计算机完成电子系统的设计。EDA 技术以计算机为工具, 代替人完成电子系统的设计仿真、逻辑综合和布局布线等工作。设计者只需要完成对系统功能的描述, 同时设定设计的约束条件, 就可以由计算机软件进行自动处理, 进而得到设计结果。借助 EDA 工具, 数字系统和芯片的设计调试就可以像软件的调试修改一样方便。所以, 利用 EDA 工具进行设计, 可以极大地提高设计效率。

EDA 工具是一系列设计工具的组合, 主要包括设计输入工具、仿真工具、综合工具和布局布线工具。设计输入工具用于输入设计源文件, 主要包括代码输入和原理图输入工具。综合和布局布线工具是设计实现工具, 用于完成设计的物

理实现。仿真工具用于验证设计是否正确,是进行设计验证的最重要的手段,在设计的不同阶段需要进行不同的仿真:逻辑仿真用于验证设计的功能正确与否;时序仿真可以估算系统的延迟;高级的仿真工具还可以对功耗、电磁兼容性、机械特性和热特性进行仿真。利用 EDA 工具进行数字系统的设计,设计者可以预知所设计系统或芯片的功能和性能,减少设计的盲目性。

数字系统的设计和实现是密不可分的,EDA 工具只能完成设计工作,而实现则需要可编程逻辑器件的支持,可编程逻辑器件和 EDA 工具的结合给今天的硬件系统设计者提供了强有力的工具。世界上著名的可编程逻辑器件厂商(如 Altera, Xilinx 和 Lattice)都提供针对自己可编程逻辑器件产品的 EDA 工具。现在,只要拥有一台计算机、空白的可编程逻辑器件芯片和一套相应的 EDA 软件,在实验室就可以完成数字系统的设计与实现。可以说,当今的数字系统设计离不开可编程逻辑器件和 EDA 设计工具。

1.2 EDA 技术的发展

历史上,EDA 技术经历了几次重大的变革:

(1) 20 世纪 70 年代,EDA 工具开始被应用到系统设计中,供应商只有几家,产品几乎全部面向 LSI 或 PCB 布线设计,这个时期的 EDA 工具仅能完成系统的辅助设计工作,无法完成系统的功能和性能的验证工作。这个阶段的 EDA 一般被称为电路 CAD (computer aided design, 计算机辅助设计)。

(2) 20 世纪 80 年代,情况发生了一些变化,出现了一些设计和验证工具。当时的 Mentor、Daisy System 及 Logic System 等公司推出了包含电路图编辑工具和逻辑模拟工具的 EDA 软件。这个时期的 EDA 工具是以数字电路设计验证工具为代表,主要是解决电路,尤其是数字电路设计完成之前的功能验证问题。

(3) 从 20 世纪 90 年代到本世纪初,EDA 工具包括了硬件行为仿真、逻辑综合、参数分析、物理实现和测试等一系列的工具。这一时期的设计工具门类齐全,能够提供系统设计需要的全部工具,如描述设计功能的设计输入工具、验证设计和评估性能的仿真工具、具有逻辑综合和设计优化能力的设计工具以及完成芯片和系统实现的物理实现工具。这些工具覆盖了现代电子设备和系统设计流程中的每一个环节,能够对系统中从系统级到门级的各个层次进行设计描述、物理实现和仿真验证。从某种意义上说,电子设计的自动化开始真正实现。

从本世纪初短短几年时间 EDA 技术的发展来看,将来的 EDA 技术将向广度和深度两个方向发展:

(1) 在广度上,EDA 技术会日益普及,成为每一个电子工程师主要的设计工具。过去,由于 EDA 软件价格昂贵,对硬件环境要求很高,一般只能运行在

使用 UNIX 操作系统的工作站或是小型机上, 因此只有在少数学校中有条件的实验室和具有一定经济实力的设计公司中才能使用 EDA 工具。从 2000 年开始, EDA 软件开始向 PC 扩展, 出现了可以基于 Windows NT 或 Windows 2000/XP 操作系统运行的 EDA 软件。随着 Linux 操作系统的迅速普及, 越来越多的 EDA 软件可以工作在 PC 平台上。这些 PC 平台上的 EDA 软件具有整套的设计、仿真和实现工具, 可以像工作站上的 EDA 软件一样完成整个系统的设计工作。随着 PC 机性能的提高, PC 平台上的 EDA 软件功能将会更加完善。EDA 软件的 PC 平台化有利于 EDA 技术的普及, 降低前期设计成本。

(2) 在深度上, 从目前 EDA 技术发展的趋势看, EDA 工具将朝着一体化方向发展。目前的各种 EDA 工具, 如系统仿真、PCB 布线、逻辑综合、布局布线设计工具等彼此是独立的, 每种设计工具有多家供应商, 而不同供应商的设计工具使用的数据文件格式各不相同, 因此在设计流程中的每一个环节都有可能遇到文件数据格式转换的问题, 给设计工作的顺利进行增添了不必要的麻烦。

随着 EDA 技术的发展及缩短电子系统设计周期的要求, 设计者希望设计流程中使用的所有工具可以在统一的数据库及管理框架环境下工作。基于这一要求, EDA 工具供应商致力于建立从系统到电路的统一语言 (如 System C 和 System Verilog), 帮助设计者在设计的过程中同时考虑仿真、实现与测试, 在系统定义时统一考虑相关的约束条件并把它们添加到设计综合中, 统一进行设计描述和优化, 提高设计的效率和成功率。

Magma 公司率先推出了一体化的 ASIC 设计工具, 随后一些知名的 EDA 工具供应商如 Synopsys、Cadence 以及 Mentor 纷纷推出了基于本公司原有设计工具的一体化设计工具和流程。而可编程逻辑器件的供应商也针对自己器件的特性开发出了相应的一体化设计工具, 如 Altera 的 Max+plusII 和最新的 QuartusII 以及 Xilinx 的 ISE。

一体化的设计工具和流程要求设计者对整个设计从系统级到 RTL (register transfer level, 寄存器传输级) 甚至门级进行综合考虑, 尽量避免传统设计中不可避免的在不同设计环节之间的迭代。因此, 一体化的设计方法对设计者自身的水平提出了更高的要求, 也对 EDA 技术的发展具有跨时代的意义。

1.3 现代数字系统的实现方法

现代数字系统的实现一般采用自顶向下的模块化设计流程。这里“顶”指的是整个数字系统的功能和定义, “下”指的是组成系统的功能部件 (一般称为子模块)。自顶向下的设计, 就是根据整个系统的功能, 按照一定的原则把系统划分成若干个子模块, 然后分别设计实现每个子模块, 最后将这些子模块组装成完

整的数字系统。值得注意的是，每个子模块的功能应该是相对独立的。图 1-1 给出了一个基于可编程逻辑器件数字系统的设计流程。

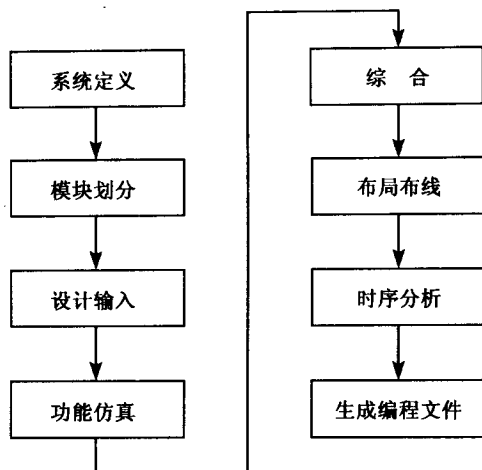


图 1-1 基于可编程逻辑器件数字系统的设计流程

除了系统定义和模块划分外，其余的步骤均由 EDA 软件完成。需要注意的是，这里给出的只是一个设计流程的示意图，实际设计不可能一次成功，因此设计过程中会在几个步骤之间来回循环（迭代），直到设计成功。在生成编程文件后，需要将文件下载到相应的可编程逻辑器件中进行整个系统的验证，验证通过后整个系统的设计才算真正完成。

本书将分别介绍可编程逻辑器件（重点介绍近年来比较流行的大规模可编程逻辑器件）、硬件描述语言 VHDL 和可编程逻辑器件的开发工具软件。随着大规模集成电路制造技术的迅猛发展，可编程逻辑器件的 EDA 技术的发展也非常迅速，器件和软件几乎每年都在更新换代。同时，这门课程也是实践性非常强的一门课程，因此，学习时应力求掌握基本概念和方法，这样在今后的实践中才能够做到举一反三。对器件的学习，我们选取具有代表性的 Altera 公司的 MAX7000 系列 CPLD 器件和 Xilinx 公司的 XC4000 系列 FPGA 器件加以介绍。同时详细介绍配合 MAX7000 系列器件的开发软件系统 MAX+plus II 的使用，以及硬件描述语言 VHDL。希望通过对这些内容的学习，读者可以初步掌握基于可编程逻辑器件数字系统的设计流程和方法，为今后学习和使用更先进的设计方法和工具打下基础。

思考题

1. 简述现代数字系统的概念。
2. 简述现代数字系统的实现方法，并说明自顶向下设计方法的基本步骤。

第 2 章 可编程逻辑器件

可编程逻辑器件是近十年才发展起来的一种新型集成电路，而 1970 年出现的 PROM 被认为是最早出现的可编程逻辑器件。现代的可编程逻辑器件以 EEPROM、SRAM 或 Flash 为基础，由用户根据自己的需要对其进行编程，确定芯片的功能。随着可编程逻辑器件在规模、密度、性能等方面的飞速发展以及成本的不断降低，越来越多的电子工程师采用了这种新型器件进行数字系统的设计。

2.1 可编程逻辑器件的发展历程

从 20 世纪 60 年代至今，数字集成电路的集成度不断提高，目前已经发展到超大规模集成电路（very large scale integration circuits, VLSI）阶段，目前集成度最高的集成电路的规模已经达到了上千万门。在数字集成电路发展的历程中先后出现了不同类型的数字集成电路，大致被分成三类：即标准逻辑器件（如用 TTL 工艺制造的 74/54 系列和用 CMOS 工艺制造的 CD4000 系列）、可由软件配置的器件（如 CPU 和单片机）和专用集成电路 ASIC（application specific integrated circuits）。可编程逻辑器件是 ASIC 的一种，它的优点在于设计者无需将设计交给集成电路生产厂家进行产品生产，只需利用 PLD 开发工具对其进行“编程”即可完成产品的生产，且可以多次重复编程，设计灵活，易于实现。随着集成电路制造工艺的快速发展，PLD 的集成度和可靠性得以迅速提高，工作速度明显加快。因此，PLD 是实现新型数字系统的理想器件。

历史上，可编程逻辑器件经历了从 PLA、PAL、GAL、EPLD 到 CPLD 和 FPGA 的发展过程，在结构、工艺、集成度、功能、速度和灵活性方面都有了很大的改进和提高。

最早的可编程逻辑器件是 1970 年出现的 PROM，它由固定的全译码与阵列和可编程的或阵列组成，采用熔丝工艺制造，只能写一次，不可擦除。其阵列规模大、速度低，主要被作为存储器使用。随着应用的要求和制造技术的发展，后来又出现了 EPROM（紫外线擦除的可编程只读存储器）和 EEPROM（电擦写可编程只读存储器）。现在，使用 PROM 和 EPROM 的场合已经很少了，而 EEPROM 则成为了应用最广泛的非易失性（掉电数据不丢失）存储器之一（另外一种 Flash）。

20 世纪 70 年代中期出现了可编程逻辑阵列（programmable logic array,

PLA) 器件, 它也是基于与或阵列的器件, PLA 的与阵列和或阵列均可编程。PLA 编程复杂, 开发起来相对困难, 并且器件的资源利用率低, 因此现在已不常使用。

20 世纪 70 年代末, 美国 MMI 公司 (Monolithic Memories Inc, 单片存储器公司) 推出了可编程阵列逻辑 (programmable array logic, PAL) 器件, PAL 由可编程的与阵列和固定的或阵列组成, 仍然是基于与或阵列的器件。PAL 可以达到很高的工作速度, 并且它的输出结构种类很多, 设计也很灵活。

20 世纪 80 年代初, Lattice 公司发明了通用阵列逻辑 (generic array logic, GAL) 器件。与 PAL 相比, GAL 增加了输出逻辑宏单元 (OLMC), 并且具有可擦除、可重复编程、数据可长期保存和可重新组合结构等优点。GAL 比 PAL 使用更加灵活, 典型的 GAL 器件 GAL16V8 和 GAL20V8 可以取代几乎所有的 PAL 器件, 因而在 20 世纪 80 年代得到广泛应用, 并逐渐取代了 PAL。

如果按集成度对 PLD 进行分类, PLA、PAL 和 GAL 同属低密度的 PLD, 其规模小, 难以实现复杂的逻辑功能。从 20 世纪 80 年代末开始, 随着集成电路工艺水平的不断提高, PLD 突破了传统的单一结构, 向着高密度、高速度、低功耗以及结构体系更灵活的方向发展, 相继出现了各种不同结构的高密度 PLD。

20 世纪 80 年代中期, Altera 公司推出了一种新型的可编程逻辑器件 (erasable programmable logic device, EPLD), 它基于 CMOS 和 EPROM 工艺制造, 集成度比 PAL 和 GAL 高得多, 设计也更加灵活, EPLD 保留了逻辑块结构, 即使是规模很大的 EPLD 的内部延时也很小。EPLD 最大的缺点在于其内部互连能力比较弱, 现在已不常使用。

1985 年 Xilinx 公司推出了现场可编程门阵列 (field programmable gate array, FPGA) 器件, 它是一种新型的高密度 PLD, 采用标准 CMOS 工艺制造, 其内部由许多独立的可编程逻辑模块组成, 逻辑块之间可以灵活地相互连接, FPGA 的功能由逻辑结构的配置数据决定, 片内的 SRAM 或者熔丝图用于工作时存放配置数据。FPGA 具有密度高、编程速度快、设计灵活和设计可再配置等诸多优点。

20 世纪 80 年代末, Lattice 公司提出了在系统可编程 (in system programmable, ISP) 技术。此后基于这一技术相继出现了一系列具备在系统编程能力的复杂可编程逻辑器件 (complex PLD, CPLD)。CPLD 是在 EPLD 的基础上发展起来的, 增加了内部互连线, 改进了内部结构体系 (包括逻辑宏单元和 I/O 单元), 增加了触发器的数量。CPLD 的性能比 EPLD 更好, 设计也更加灵活。

进入 20 世纪 90 年代以后, 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。FPGA 的集成度已经可以达到 100 万个等效 PLD 门,

最高工作频率可以达到 100MHz 以上；而目前 CPLD 产品的集成度最多可达 25 万个等效门，最高工作速度已达 180MHz。在系统可编程技术、边界扫描技术的出现也使器件在编程技术和测试技术及系统可重构技术方面有了很快的发展。世界各著名的可编程逻辑器件公司，如 Altera、Xilinx、Lattice 等，均可提供不同类型的 FPGA、CPLD 产品，FPGA 和 CPLD 已经成为可编程逻辑器件家族的中坚力量，很多数字系统均可以用这两种器件实现。为了满足特定应用的需要，最新的 PLD 产品中已经集成了 RAM、FIFO 以及嵌入式 CPU 等模块。众多公司的竞争促进了可编程逻辑器件技术的提高，使其性能不断完善，产品日益丰富。现在，可编程逻辑器件已经成为设计和实现现代数字系统的重要手段。

2.2 可编程逻辑器件的基本结构与分类

实际上，所有的组合逻辑均可以用与或表达式来表示，因此现在使用的可编程逻辑器件普遍都是由输入电路、与阵列、或阵列和输出电路几部分构成，其基本结构如图 2-1 所示。

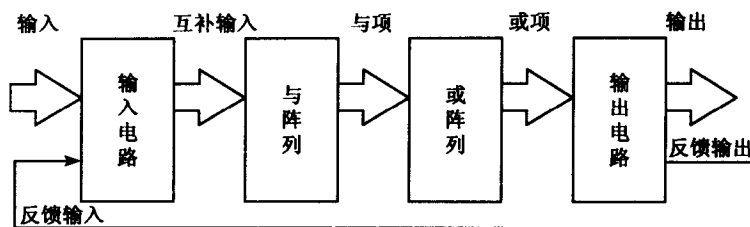


图 2-1 可编程逻辑器件结构示意图

与阵列和或阵列是可编程逻辑器件的核心部分，分别用于产生乘积项和加和项。输入电路一般采用互补输入的形式，而输出电路存在多种形式，包括组合逻辑输出、时序输出以及可编程输出结构，输出还可根据需要通过内部通路反馈到输入端。

值得注意的是，可编程逻辑器件内部的连接非常复杂，一般采用一种简化的表示方式替代传统的表示方式来描述可编程逻辑器件的内部结构。输入端常采用的互补电路的表示方法参见图 2-2，图 2-3 和图 2-4 分别是与门和或门的表示方法。

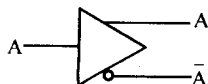


图 2-2 PLD 输入电路



图 2-3 PLD 与门表示法

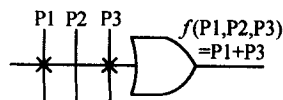


图 2-4 PLD 或门表示法