

普通高等教育“十三五”规划教材
卓越工程师培养计划系列教材

数字系统与微处理器

Digital Systems and Microprocessors

潘 曦 闫建华 郑建君 © 编著

普通高等教育“十三五”规划教材
卓越工程师培养计划系列教材

数字系统与微处理器

Digital Systems and Microprocessors

潘曦 闫建华 郑建君 ◎ 编著



内 容 简 介

本书以基于硬件编程语言的数字系统设计和微处理器软核设计为主线，全书内容分为两大部分：数字系统和微处理器。前四章主要是基于 Verilog HDL 硬件描述语言的数字电路设计相关内容；后四章主要涉及微处理器的体系结构和两种基于 FPGA 的微处理器软核的设计和应用。本书突出介绍基于 FPGA 的数字系统与微处理器软核的 SoPC 设计方法与技术，并建立起数字系统与微处理器的桥梁。

本书涉及众多实际工程中设计和实践中的实例，结合了 FPGA 技术的新发展及设计中的具体问题，适用于学习高级数字系统设计课程的高年级本科生和研究生，以及想通过实例学习 Verilog HDL 并对微处理器软核设计感兴趣的专业工程师。

版权专有 侵权必究

图书在版编目（CIP）数据

数字系统与微处理器/潘曦，闫建华，郑建君编著. —北京：北京理工大学出版社，2018.7

ISBN 978 - 7 - 5682 - 5896 - 8

I. ①数… II. ①潘…②闫…③郑… III. ①数字系统 - 系统设计②微处理器 - 系统设计 IV. ①TP271②TP332

中国版本图书馆 CIP 数据核字（2018）第 158705 号

出版发行 / 北京理工大学出版社有限责任公司

社 址 / 北京市海淀区中关村南大街 5 号

邮 编 / 100081

电 话 / (010) 68914775（总编室）

(010) 82562903（教材售后服务热线）

(010) 68948351（其他图书服务热线）

网 址 / <http://www.bitpress.com.cn>

经 销 / 全国各地新华书店

印 刷 / 三河市华骏印务包装有限公司

开 本 / 787 毫米 × 1092 毫米 1/16

印 张 / 22.25

字 数 / 518 千字

版 次 / 2018 年 7 月第 1 版 2018 年 7 月第 1 次印刷

定 价 / 59.00 元

责任编辑 / 陈莉华

文案编辑 / 陈莉华

责任校对 / 周瑞红

责任印制 / 王美丽

图书出现印装质量问题，请拨打售后服务热线，本社负责调换

前言

近年随着 SoPC (System on Programmable Chip) 设计理念的提出, FPGA 不仅能够实现数字逻辑运算, 而且能够通过配置软核或增加硬核支持实现微处理器的功能。该类系统在现实工程中大量应用, 成为一种全新的设计方法。本书结合 FPGA 技术的新发展, 讲述基于 FPGA 的数字系统与微处理器软核的 SoPC 设计方法与技术, 对本科生和研究生后续从事技术研究和工程设计工作具有指导性。

本书分为两个部分: 数字系统和微处理器。数字系统部分包含前四章, 主要讲解基于 Verilog HDL 硬件描述语言的 FPGA 数字电路设计; 微处理器部分为后四章, 重点介绍主流微处理器的体系结构和两种基于 FPGA 的微处理器软核的结构和应用。

本书的编写目的在于: (1) 复习数字电路设计基本原理; (2) 介绍硬件描述语言的工程应用背景; (3) 以 Verilog HDL 为设计语言, 重点讨论适用于现场可编程门阵列 (FPGA) 的数字逻辑实现; (4) 从基本数字电路设计引申至 FPGA 体系及 IP 核的工程应用背景; (5) 在读者已初步掌握 FPGA 设计方法的基础上, 基于常用的 Intel x86 指令集和常见的微处理器类型, 引出 FPGA 中具有微处理器功能的“软核”设计; (6) 重点讲解主流的 Nios II 和 MicroBlaze 两种微处理器软核的结构和应用, 拓展读者的知识面, 具备足够的设计能力, 实现片上可编程系统 (SoPC)。

本书适用于学习高级数字系统设计课程的高年级本科生和低年级研究生, 以及想通过实例学习 Verilog HDL 并对微处理器软核感兴趣的专业工程师。

本书由潘曦、闫建华、郑建君编著, 参加本书编辑及文字校对工作的有李英飞、赵天婵、牛飞、卢士鹏、苏永超、姜志毅、马胜利等。

由于作者的实际经验及水平的限制, 本书难免存在一些错误和不足, 殷切希望广大读者不吝指正。

作者

目 录

CONTENTS

第1章 数字设计基础	001
1.1 数字设计简介	001
1.1.1 模拟系统和数字系统	001
1.1.2 微处理器与微控制器	002
1.1.3 可编程逻辑器件	005
1.1.4 集成电路与片上系统	009
1.1.5 SoPC 系统	011
1.2 数制和编码	012
1.2.1 十进制和非十进制	012
1.2.2 二进制数字系统	013
1.2.3 BCD 码	015
1.2.4 ASCII 码	015
1.2.5 格雷码	016
1.2.6 数据编码	017
1.3 数字电路	024
1.3.1 基本逻辑门电路	024
1.3.2 组合逻辑电路	028
1.3.3 组合逻辑电路中的竞争冒险现象	032
1.3.4 时序逻辑电路	035
1.3.5 CMOS 逻辑	038
1.4 数字系统的设计方法及流程	041
1.4.1 层次化设计流程	041
1.4.2 层次化设计的优缺点	043
1.5 思考题	043
第2章 硬件描述语言	044
2.1 硬件描述语言简介	044

2.2 Verilog HDL 基本程序结构	045
2.2.1 模块	045
2.2.2 模块实例引用	046
2.3 Verilog HDL 语言基础	047
2.3.1 标识符	047
2.3.2 常量	048
2.3.3 变量及数据类型	049
2.3.4 运算符和表达式	051
2.3.5 过程结构	056
2.3.6 控制结构	061
2.3.7 语句块	064
2.4 系统的描述方法	065
2.4.1 结构级描述	066
2.4.2 行为级描述	066
2.5 设计综合与验证	070
2.5.1 设计综合	070
2.5.2 设计仿真验证	071
2.6 思考题	077
第3章 组合逻辑设计	079
3.1 用 Verilog HDL 实现基本逻辑门电路	079
3.2 用 Verilog HDL 实现组合逻辑	082
3.2.1 编码器	082
3.2.2 译码器	090
3.2.3 多路选择器	093
3.2.4 比较器	094
3.2.5 加法器	098
3.2.6 乘法器	103
3.3 组合电路 Verilog HDL 设计技巧	104
3.3.1 合理选择加法电路	105
3.3.2 减少关键路径的逻辑级数	106
3.3.3 合并 if 语句提高设计速度	108
3.3.4 资源共享	108
3.3.5 利用电路的等价性巧妙地“分配”延时	111
3.3.6 复制电路、减少扇出 (fanout)、提高设计速度	112
3.4 思考题	112
第4章 时序逻辑设计	114
4.1 同步时序电路和异步时序电路	114
4.1.1 同步时序电路和异步时序电路	114
4.1.2 同步时序设计的重要性	114

4.2 常用时序逻辑电路设计	115
4.2.1 触发器和锁存器的设计	115
4.2.2 计数器的设计	118
4.2.3 移位寄存器的设计	119
4.2.4 CRC 校验码产生器的设计	120
4.2.5 伪随机序列发生器	123
4.3 状态机设计	127
4.3.1 有限状态机的原理与结构	127
4.3.2 可综合状态机的设计	129
4.3.3 设计实例	135
4.4 流水线设计	137
4.5 再谈阻塞赋值与非阻塞赋值	140
4.6 提高同步设计时序	148
4.6.1 时钟偏移	148
4.6.2 异步输入	148
4.6.3 数据接口的同步方法	155
4.7 思考题	156
第5章 FPGA 体系及 IP 核	158
5.1 典型 FPGA 体系架构	158
5.1.1 FPGA 的基本工作原理	158
5.1.2 FPGA 典型架构	159
5.2 FPGA 内部资源	165
5.2.1 时钟	165
5.2.2 片内存储器	172
5.2.3 DSP 模块	174
5.2.4 高速接口	176
5.3 IP 核	176
5.3.1 IP 核的概念	176
5.3.2 软核	177
5.3.3 硬核	182
5.4 基于 IP 核的快速数字设计	183
5.4.1 基于 IP 核的快速设计流程	183
5.4.2 MATLAB 辅助完成 FPGA 设计	185
5.4.3 设计实例	186
5.5 思考题	194
第6章 微处理器体系结构及关键技术	196
6.1 微处理器体系结构	196
6.1.1 计算机体系结构	196
6.1.2 微处理器体系结构	197

6.1.3	微处理器的工作过程	201
6.1.4	存储系统	203
6.1.5	Pentium 4 处理器	203
6.2	指令系统	204
6.2.1	指令组成	205
6.2.2	寻址方式	207
6.2.3	CISC 和 RISC 指令集	210
6.3	流水线技术	212
6.3.1	流水线技术的优势	212
6.3.2	流水线结构的操作	213
6.3.3	流水线的性能分析	215
6.3.4	流水线结构的冒险	217
6.3.5	流水线结构的局限性	219
6.3.6	超标量处理机	220
6.4	高速缓存技术 (Cache)	223
6.4.1	Cache 的工作原理	224
6.4.2	Cache 的基本结构	226
6.4.3	Cache 的工作过程	231
6.4.4	Cache 的一致性问题的	234
6.4.5	Cache 的性能估算	237
6.5	ARM 微处理器	238
6.6	基于 Xilinx FPGA 平台的微处理器	240
6.7	基于 Altera FPGA 平台的微处理器	241
6.8	思考题	242
第 7 章	MicroBlaze 处理器的结构与应用	245
7.1	MicroBlaze 处理器系统	245
7.1.1	MicroBlaze 体系架构	246
7.1.2	MicroBlaze 指令系统	248
7.1.3	MicroBlaze 事件处理	250
7.1.4	MicroBlaze 存储器系统	252
7.2	MicroBlaze 总线结构	256
7.2.1	PLB 总线系统结构	257
7.2.2	AXI 总线系统结构	261
7.3	MicroBlaze 的 IP 核及设备驱动	264
7.3.1	UART 控制器	264
7.3.2	IIC 控制器	268
7.3.3	MPMC 控制器	270
7.4	基于 MicroBlaze 处理器的开发实例	273
7.4.1	利用 XPS 软件构建硬件平台	274

7.4.2 利用 SDK 软件开发嵌入式软件	283
7.4.3 GPIO 中断实例软件开发	286
7.5 思考题	291
第 8 章 Nios II 处理器的结构与应用	293
8.1 Nios II 处理器系统	293
8.1.1 Nios II 体系架构	293
8.1.2 Nios II 指令系统	298
8.1.3 Nios II 事件处理	300
8.1.4 Nios II 存储器系统	302
8.2 Nios II 总线结构	306
8.2.1 Avalon 总线	306
8.2.2 Avalon - MM 总线	307
8.2.3 Avalon - Stream 总线	309
8.3 Nios II 的 IP 核及设备驱动	311
8.3.1 定时器	312
8.3.2 PIO 核	315
8.3.3 JTAG UART 核	318
8.4 基于 Nios II 处理器的开发实例	321
8.4.1 利用 Quartus II 和 Qsys 构建硬件平台	323
8.4.2 利用 Nios II EDS 开发嵌入式软件	334
8.4.3 定时器中断实例软件开发	335
8.5 思考题	339
附录 A Verilog HDL (IEEE 1364—2005) 关键字列表	340
参考文献	342

第 1 章

数字设计基础

在过去的 40 多年里,随着集成电路的速度和集成度的快速提高,数字设计经历了非常大的转变。过去,数字设计者用成千甚至上万的门电路和触发器来构建系统,其重点就是最小化和有效地利用芯片和板级资源。

如今,数字系统的设计不再依靠简单的门电路搭建,一个芯片可以包含几千万个晶体管并且可以利用编程的方式创建片上系统。而且,还有一种器件的出现及广泛应用极大地改变了数字系统设计的方式和方法,这就是可编程逻辑器件(PLD),使数字电路由中小规模的集成电路向 PLD 及专用集成电路(ASIC)转变。数字设计手段也发生了变化,由传统的手工方式逐渐转变为以 EDA 作为设计平台的方式,使用硬件语言设计 PLD 成为一种趋势。同时,PLD 技术的发展,使片上系统设计成了可能。片上系统是将处理器、存储器、I/O 口等功能模块都集成到一个可编程器件上,构成一个完整的可编程处理器系统。本书结合数字系统和微处理器,讲述了基于 PLD 的数字系统设计以及片上系统设计。在本章中,介绍了一些基本知识,包括数字电路、微处理器、可编程逻辑器件以及数字系统设计的基本概念。

在后续的章节中,我们将逐步介绍硬件描述语言 HDL(参见第 2 章)以及利用 HDL 设计复杂数字电路(参见第 3 章到第 4 章)的内容。

1.1 数字设计简介

1.1.1 模拟系统和数字系统

模拟信号是连续变化的物理量。自然界产生的物理量基本都是以模拟形式存在的。清晨小鸟鸣叫的声音,形成的声波幅值是连续的模拟信号。水银温度计中水银柱的高度随着房间温度升高或降低,以模拟方式连续地变化。实际电路中,电压、电流呈现出连续变化的值。工程设计上,系统中处理、控制的输入/输出大多是模拟量,如速度、位移、温度、压力等,以模拟形式处理物理量的装置称为模拟系统。

现实世界多以模拟量存在,然而人们在处理这些信息中常常把它们转化为数字形式,一个数值用约定好格式的一串 0 和 1 的组合表示,体现在具体实现即是 ON 和 OFF 的电平组合。也就是说,数字信号是离散的,是模拟信号经过采样、量化和编码后有限精度的数字形式。温度计中的水银柱停在两个相邻刻度之间时,我们选取距离近的那条刻度线作为测量值,比如 37℃,这是量化的过程,之后将 37 按着一定编码规则表示为 0 和 1 的组合(如 00100101),这样我们就把模拟的温度显示转变为数字形式了。雷达探测中,人们把回波信

号数字化后再完成一系列算法的处理。

【例 1-1】 判断如下情况哪些是模拟信号，哪些是数字信号？

①雷达天线发射信号；②液压传感器输出的电压信号；③十位置开关信号；④激光测距仪屏幕显示值；⑤高清电视信号。

解：

①模拟量；②模拟量；③数字量；④数字量；⑤数字量。

很多情况下，我们会把模拟信号变为数字信号来处理。用来处理逻辑信息或以数字形式表示的物理量的装置称为数字系统。数字系统具有以下优势：

(1) 可以实现复杂的数字运算和逻辑运算。实际的模拟电路很难实现复杂的算法，如 FFT、自适应滤波等。

(2) 信息易于存储。数字信息可以大量存储于特定的器件和电路中，而模拟器件存储能力很有限。

(3) 系统的一致性和可重复性好。数字信息处理和存储中不会改变精度，而模拟元器件参数受到温度、湿度和其他噪声影响较大。

(4) 参数配置和算法修改灵活。数字系统的核心多是可编程的信号处理器或可编程逻辑门阵列，通过设计相应程序完成功能，具有相当的灵活性和软件可编程、配置的特点，而模拟系统一旦设计完成，参数和功能都将固定，需要新的参数设置情况下，只能重新设计硬件。

(5) 集成度高。多数的数字电路都集成在集成电路（Integrated Circuit，IC）芯片中，具有很高的集成度，具有体积小、功耗低、一致性好的特点。

1.1.2 微处理器与微控制器

1. 中央处理器

中央处理器（Central Processing Unit，CPU）是一块超大规模的集成电路，是计算机的运算和控制核心，它的主要功能是解释执行计算机指令以及处理数据。CPU 主要包括运算器和高速缓冲存储器及实现它们之间联系的数据、控制及状态的总线。它与内部存储器和输入/输出设备合称为电子计算机三大核心部件。由集成电路制造的 CPU 通常也称为微型处理器。从 20 世纪 70 年代中期开始，单芯片微型处理器几乎取代了所有其他类型的 CPU。

2. 微处理器

微处理器（Microprocessor Unit，MPU）是用于进行算术逻辑运算、控制系统和存储数据等操作的小型计算机或 CPU。一般单片集成的 CPU 与微处理器并不做太多区分。微处理器处理输入/输出（I/O）外设的数据，并将结果返回给它们运行。微处理器结构框图如图 1-1 所示。

微处理器最初使用冯·诺依曼体系结构。在该结构中，数据存储器和程序存储器安排在一个存储器中。处理器在处理存储器的指令或 I/O 接口的请求时，先通过总线读取指令，然后在相应寄存器中处理，运算数据及结果同样通过总线读取或写入内存中。这种架构的缺点是速度慢，数据操作和指令操作不能同时进行。后来开发的哈佛结构，数据存储器和程序存储器独立存在，单独的总线连接可以互相通信。两种结构框架如图 1-2 所示。

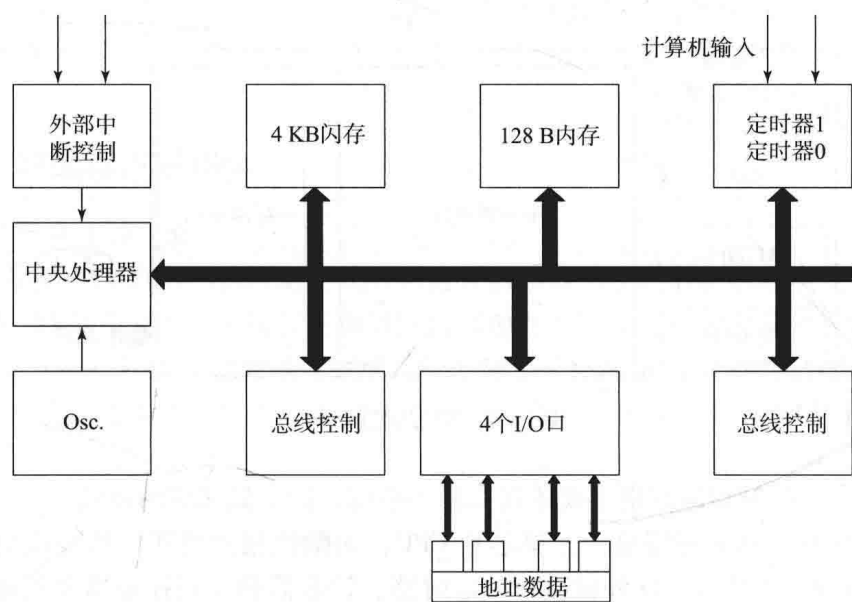


图 1-1 微处理器结构框图

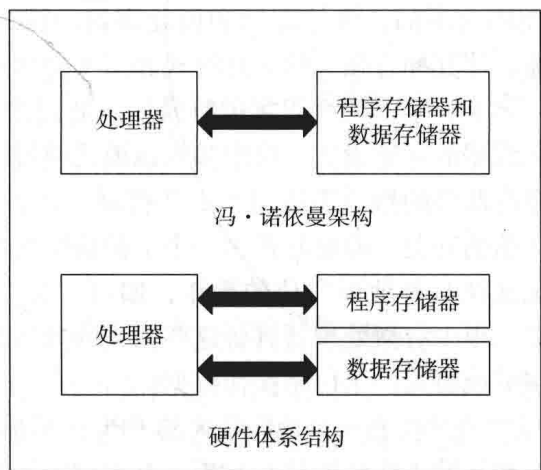


图 1-2 微处理器构架

指令集结构（ISA）在微处理器中也起着重要的作用，指令集可分为复杂指令集计算机（CISC）和精简指令集计算机（RISC）。CISC 指令复杂，执行速度慢，例如 x86 架构；RISC 中指令集小，执行速度快，实现简单并且不需要复杂的架构，被广泛用于嵌入式应用。由于本书的一个重点就是采用 SoPC（可编程片上系统）方式实现微处理器，在本书第 6 章，对于微处理器体系结构和指令系统等还有详细介绍。

3. 微控制器

微控制器是将 CPU、存储器单元（RAM、ROM）、I/O 外设、定时器、计数器等嵌入在一个集成电路中，可以容易地连接到外部设备，如串口、ADC、DAC、蓝牙、Wi-Fi 等，与微处理器接口相比，微控制器外部接口更灵活。大多数微控制器使用 RISC 架构，也有一些微控制器使用 CISC 架构，如 8051、Motorola 等。微控制器结构图如图 1-3 所示。

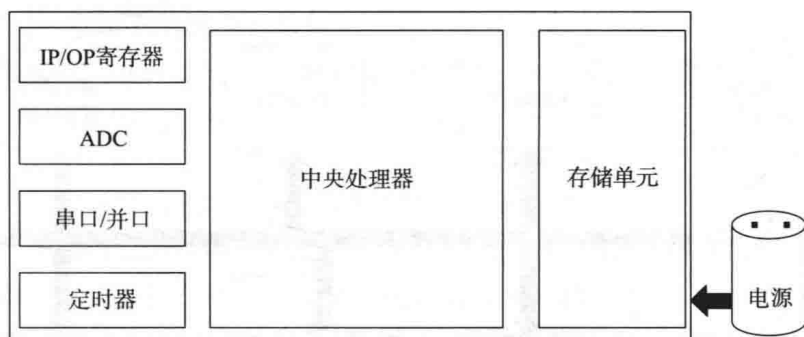


图 1-3 微控制器结构

微处理器和微控制器的区别主要体现在硬件结构、指令集和应用领域。

(1) 硬件结构。微处理器是一个单芯片 CPU，而微控制器则在一块集成电路芯片中集成了 CPU 和 ROM、RAM、I/O 外设接口、定时器、计数器和中断控制器等其他电路，构成了一个比较完整的微型计算机系统。虽然片上 RAM 的容量比普通微型计算机系统小，但由于微控制器的灵活性，其应用范围非常广泛。

(2) 指令集。由于应用场合不同，微控制器和微处理器的指令集也有所不同。微处理器的指令集增强了处理功能，使其拥有强大的寻址模式和适于操作大规模数据的指令。微处理器的指令可以对半字节、字节、字，甚至双字进行操作。通过使用地址指针和地址偏移，微处理器提供了可访问大批数据的寻址模式。自增和自减模式使得以字节、字或双字为单位访问数据变得非常容易。微控制器的指令集适用于 I/O 控制。许多输入/输出的接口是单位的。例如，电磁铁控制着马达的开关，而电磁铁由一个 1 位的输出端口控制。微控制器具有设置和清除单位的指令，也能执行其他面向位的操作，如对“位”进行逻辑与、或和异或的运算，根据标志位跳转等。很少有微处理器具备这些强大的位操作能力，因为设计者在设计微处理器时，仅考虑以字节或更大的单位来操作数据。

(3) 应用领域。微处理器通常代表一个功能强大的 CPU，不是为任何专用目的而设计，这种芯片往往是个人计算机和高端工作站的核心 CPU。微控制器是将一个计算机集成到一个芯片中，系统设计追求小型化和应用灵活性，其功能也许不如 CPU 强大，但一个微控制器、少量的外部元器件和存储在 ROM 中的控制程序就能够实现众多实际工程中控制和算法的需求。

早期的微控制器是在一片芯片中“嵌入”了微型计算机实现嵌入式应用，故又称单片机 (Single Chip Microcomputer)。随后，为了更好地满足嵌入式应用，单片机中不断扩展一些满足控制要求的电路单元。值得说明的是，由于集成芯片设计及工艺水平的飞速发展，微处理器和微控制器技术越来越相互融合，界限也不再那么清晰。比如出现了众多从微处理器发展而来的微控制器，Advanced RISC Machines 公司的 ARM、Silicon Graphics 公司的 MIPS、IBM 和 Motorola 的 Power PC、Intel 的 x86 和 i960 芯片、AMD 的 Am386EM 等，它们与单片机一样，也可以称为嵌入式处理器。另外一种处理器为数字信号处理器 (Digital Signal Processor, DSP)，DSP 里专门设计的 CPU 是特别适合于数字信号处理运算的微处理器，主要应用是实时快速地实现各种数字信号处理算法，比如音频和视频通信、图像处理等场合。

DSP 芯片专门设计了乘加器、快速 RAM、专用中断处理和 I/O 支持硬件以及改进的哈佛总线结构和流水线操作等,运算速度更快,比较常用的有 TI 的 TMS320Cxx 系列和 ADI 的 SHARC 系列。

1.1.3 可编程逻辑器件

1. 数字逻辑器件的分类

大多数数字系统不再由简单门电路或者特定逻辑功能的逻辑器件芯片实现,而是采用可编程逻辑器件(PLD)完成。可编程逻辑器件内部包含生成逻辑功能所需要的电路,这些器件不像 MCU 或 DSP 一样通过指令编程完成设计,而是采用电的方法对电路内部硬件电路各点之间的连接或者不连接进行配置。下面从数字逻辑器件分类的角度引出 PLD,如图 1-4 所示。

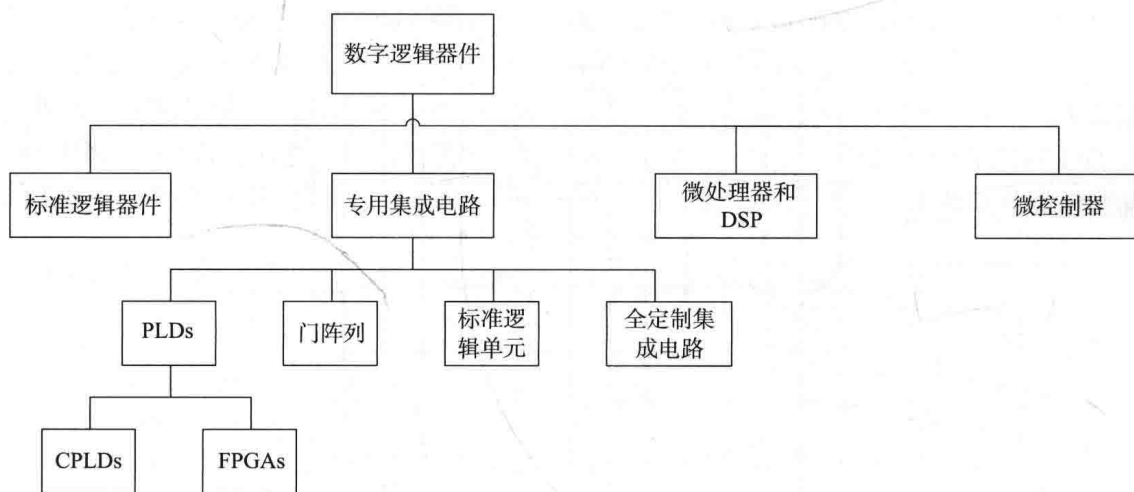


图 1-4 数字逻辑器件的分类

标准逻辑器件是指具备基本功能的数字逻辑器件,例如门电路、触发器、译码器、寄存器、计数器等,以中小规模的集成电路形式出现,实现功能比较单一。基本功能数字逻辑在本章 1.3、1.4 节有详细分析。

微处理器和 DSP 以及微控制器在 1.1.2 节有相应介绍,这两种器件都是通过编写并执行应用程序指令来完成相应数字设计功能,相比较硬件设计方案,软件设计方案设计灵活,但是速度相对慢。

PLD 有时也叫作现场可编程逻辑器件(FPLD),从简单的门电路及数字逻辑到复杂的数字系统,都可以由用户配置 PLD 实现,PLD 配置方式可以通过图形文件,更多地采用硬件编程语言(HDL)编程实现。本书主要内容是讨论通过 PLD 实现数字系统以及片上可编程系统(SoPC)的功能,在第 3~4 章对基本数字逻辑的 HDL 实现有详细讲解。而从第 5 章起开始讨论微处理器及 SoPC 的相关内容。

图 1-4 中,专用集成电路还包括门阵列、标准逻辑单元以及全定制集成电路。门阵列是拥有几十万个门电路的超大规模集成电路。与 PLD 类似,若要达到其逻辑功能,用户需要设计门电路的相互连接,就单个器件而言,这类器件比具有同等数量门电路的 PLD 价格要低。标准逻辑单元专用集成电路采用预先确定的逻辑功能模块生成要求的数字系统,与 PLD 不同,

集成电路中每个逻辑单元的布局事先已设计好，可用的逻辑单元存在计算机的数据库中。与门阵列相比，实现同样逻辑功能电路，所设计的基本逻辑单元的电路要小，工作速度较高。全定制集成电路中所有元件（晶体管、电阻、电容）和它们之间的连线均由集成电路设计者为用户定制，设计时间较长，成本较高，但设计的专用芯片由于其专用性，面积最小。

2. PLD 的结构和特点

PLD 最常用的是复杂可编程逻辑器件（CPLD）和现场可编程门阵列（FPGA）。

典型的 CPLD 器件是在一个芯片上把多个可编程逻辑阵列（Programmable Array Logic, PAL）集成为一个阵列，PAL 是一种可编程逻辑器件，由固定的“或”阵列和可编程的“与”阵列构成，通过对与逻辑阵列编程可以获得不同形式的组合逻辑函数。PAL 结构如图 1-5 所示。

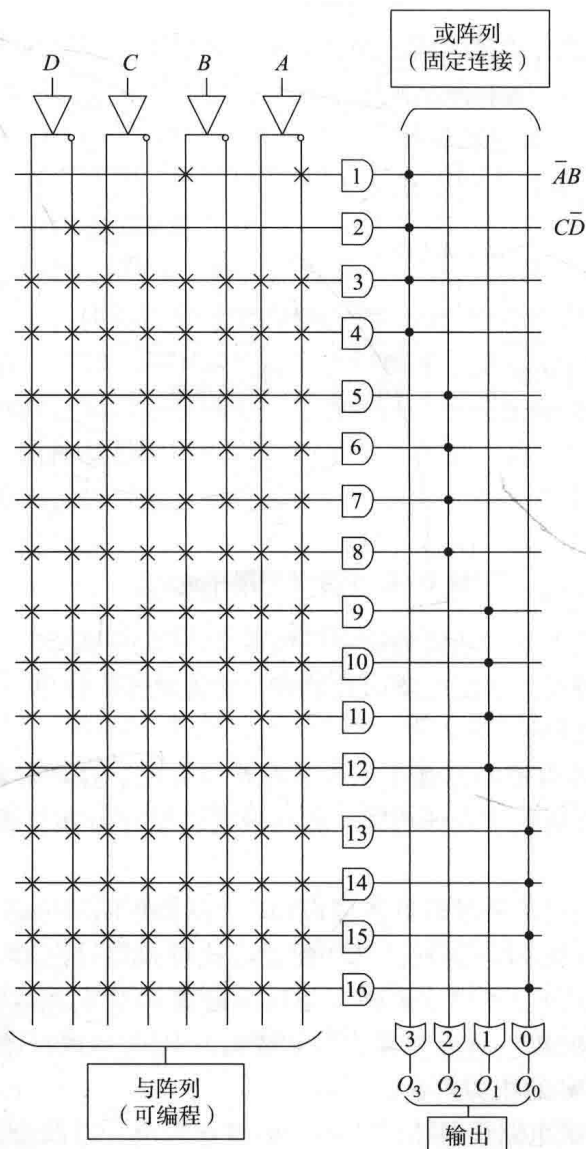


图 1-5 PAL 结构示意图

图 1-5 中，与门的输入是可编程的，交叉点有“x”表示相应与门线的熔丝连通，而

交叉点没有“×”表示熔丝烧断。对于与门，烧断熔丝时输入处于高电平。PAL 的或门的输入与与门的输出是固定连接的，该图中所示的或门输入限制了仅由 4 个与门连接。以图中 O_3 为例，与门编程后输出：

$$O_3 = \bar{A}B + C\bar{D} + 0 + 0$$

之所以后面为 0，是因为 $A\bar{A}B\bar{B}C\bar{C}D\bar{D} = 0$ 。

CPLD 器件是由多个 PAL 型器件组成的与/或阵列以及宏单元库构成。CPLD 最基本的单元是宏单元，一个宏单元包含一个寄存器（使用多达 16 个乘积项作为其输入），每个宏单元可实现组合或时序逻辑功能或者寄存器功能。当需要处理更多数据时，多个宏单元可以一起实现逻辑功能。

以 Xilinx XC9500 系列 CPLD 为例（见图 1-6），CPLD 由多个功能块（Function Block, FB）和 I/O 块（I/O Block, IOB）通过开关矩阵连接起来。FB 由 18 个独立的宏单元构成，产生 18 个输出用于驱动快速连接开关矩阵，同时和它们相对应的输出使能信号也驱动 IOB。每个宏单元中，有 5 个来自“与”阵列乘积项可以作为基本数据输入（到或门或者异或门）来实现组合逻辑功能，或者作为时钟、复位和输出使能的控制输入。快速连接开关阵列连接 FB 和 IOB，可以通过编程实现信号输入/输出的相同延迟。IOB 是内部逻辑和用户 I/O 引脚之间的接口，每个 IOB 包括输入缓冲区、输出驱动器、输出使能选择复用器以及编程控制。

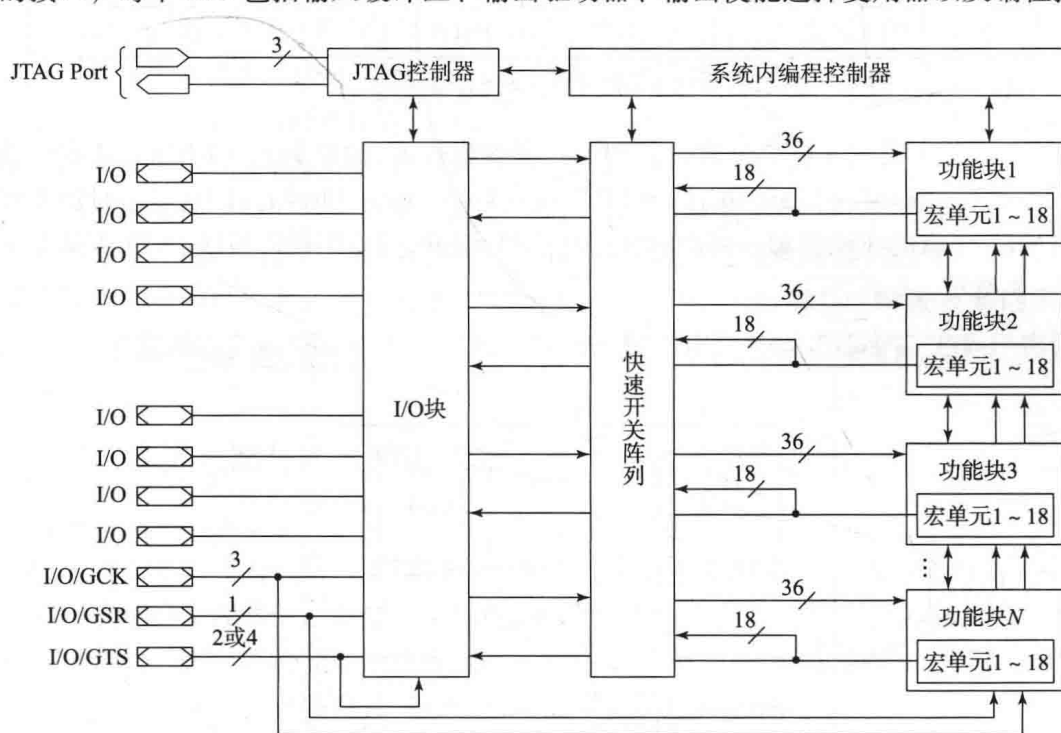


图 1-6 XC9500 CPLD 内部结构

在编程方式上，CPLD 主要是基于 EPROM 或 Flash 存储器编程，在系统断电时编程信息也不丢失。

FPGA 是在 PAL、CPLD 等可编程器件的基础上进一步发展起来的可编程逻辑器件，由于 FPGA 需要被反复烧写，因此主流的 FPGA 都采用了基于 SRAM 工艺的查找表结构，也存在采用 Flash 或反熔丝结构的查找表结构。查找表功能相当于真值表，FPGA 通过烧写文件去

配置查找表的内容，从而在相同的电路情况下实现不同的逻辑功能。可以把查找表（Look - Up - Table, LUT）看作是一个有地址线的 RAM，用户通过原理图或者 HDL 语言设计逻辑电路后，利用开发软件可以将逻辑电路所有可能的结果存入到 RAM 中。当有信号输入时，LUT 的功能是把该输入信号作为一个地址进行查表，找到该地址对应的内容并输出。图 1 - 7 是查找表与实际逻辑电路功能，可见该四输入查找表即实现逻辑电路功能，根据不同输入可以输出所有真值表中对应的值。

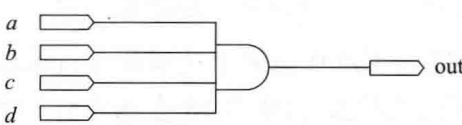
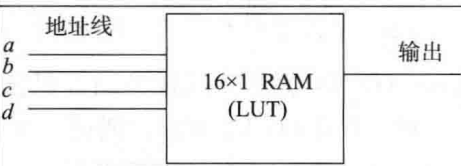
实际逻辑电路		LUT的实现方式	
			
<i>abcd</i> 输入	逻辑输出	地址	RAM中存储的内容
0000	0	0000	0
0001	0	0001	0
.....	0		0
1111	1	1111	1

图 1 - 7 LUT 的实现原理图

FPGA 芯片主要由 6 部分组成，分别为：可编程输入/输出单元（IOB）、基本可编程逻辑单元 [（Configurable Logic Block, CLB）或（Logic Array Blocks, LAB）]、时钟管理、嵌入式 RAM、丰富的布线资源、内嵌的底层功能单元和内嵌专用硬件模块。FPGA 的基本结构如图 1 - 8 所示。

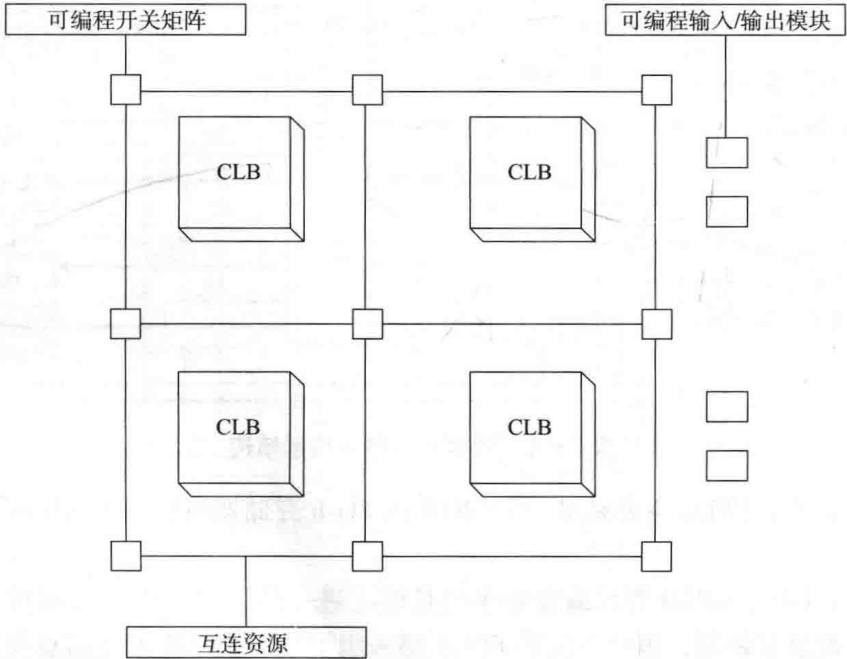


图 1 - 8 FPGA 基本结构