



中国电子教育学会高教分会推荐
普通高等教育电子信息类“十三五”课改规划教材



微机原理与接口技术

——基于8086和Proteus仿真

主编 马宏锋



西安电子科技大学出版社
<http://www.xduph.com>

微机原理与接口技术

——基于 8086 和 Proteus 仿真

马宏锋 主编

西安电子科技大学出版社

内 容 简 介

本书以 Intel 8086 为背景, 基于 Proteus 仿真, 介绍了微型计算机结构、典型微处理器、存储器技术、汇编语言程序设计、输入/输出接口技术、微机总线技术等内容, 其中以微型计算机的关键技术(如 Cache、存储管理、中断、DMA、系统总线、异步接口等)为本书的重点内容。此外, 书中简单介绍了微型计算机应用系统的设计, 以扩充读者的知识面。

本书可作为应用型本科电气电子、通信和计算机等学科相关专业的“微机原理与接口技术”课程教材, 也可作为从事微型计算机硬件和软件开发的工程技术人员学习和应用的参考书。

图书在版编目(CIP)数据

微机原理与接口技术: 基于 8086 和 Proteus 仿真/马宏锋主编. —西安: 西安电子科技大学出版社, 2016.12

ISBN 978-7-5606-4071-6

I. ① 微… II. ① 马… III. ① 微型计算机—理论 ② 微型计算机—接口技术 IV. ① TP36

中国版本图书馆 CIP 数据核字(2016)第 299877 号

策 划 刘玉芳

责任编辑 曹 超 毛红兵

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)88242885 88201467 邮 编 710071

网 址 www.xduph.com 电子邮箱 xdupfb001@163.com

经 销 新华书店

印刷单位 陕西华沐印刷科技有限责任公司

版 次 2016 年 12 月第 1 版 2016 年 12 月第 1 次印刷

开 本 787 毫米×1092 毫米 1/16 印张 21.5

字 数 511 千字

印 数 1~3000 册

定 价 38.00 元

ISBN 978-7-5606-4071-6/TP

XDUP 4363001-1

如有印装问题可调换

前 言

“微机原理与接口技术”是工科院校电气电子、通信和计算机等学科相关专业的重要基础课程，具有很强的应用性和实践性。随着微型计算机软、硬件技术的不断升级换代，其教学内容也要求不断更新。鉴于此，编者编写了本书。本书的编写适应高等教育的快速发展，满足教学改革和课程建设的需求，体现了应用型本科的特点，且注重应用型人才专业技能和实用技术的培养。

本书内容以基本概念为基础，以技术发展为主线，以关键技术为重点，特别加强对关键部件的逻辑与时序的分析，强调系统扩展和设计案例分析，紧密结合实验教学，且给出基于 Proteus 平台的仿真实例，以逐步培养微型计算机应用和研发人员所必需的资料阅读能力、时序分析和接口设计能力，以及系统设计、软件编程和硬件调试的能力。

本书的特点具体如下：

(1) 结合应用型本科计算机类和电子信息类的课程体系改革，在兼顾基础知识的同时，强调实用性和可操作性。

(2) 突出计算机硬件系统和 I/O 处理技术的主要概念，通过应用系统开发实例，培养学生应用系统硬件开发和驱动程序设计的能力，体现以能力为本位的教学思想。

(3) 增加微机系统的新技术、实用内容和知识点，采用由浅入深、循序渐进、层次清楚的编写方式，突出实践技能和动手能力。

(4) 提供配套的多媒体课件(教师用)和网络课件(学生用)，并将陆续建设习题指导和实验实训等辅助教学资源。

本书的编写工作得到广州风标电子技术公司的支持，在此向他们表示诚挚的谢意！

本书内容较丰富，建议学时为 64~80 学时(含实验)，各校可按照实际情况进行调整。由于作者水平有限，书中难免存在错误与不妥之处，请广大读者批评指正。

作者

2016 年 10 月

目 录

第 1 章 微型计算机概论 1	
1.1 微型计算机概述..... 1	
1.2 计算机中数据信息的表示..... 3	
1.2.1 数据格式及机器数..... 3	
1.2.2 数字信息编码..... 4	
1.3 逻辑单元与逻辑部件..... 6	
1.3.1 二进制的逻辑运算与逻辑电路..... 6	
1.3.2 常用逻辑部件..... 7	
1.4 微型计算机的基本结构..... 12	
1.4.1 微型计算机的硬件结构..... 12	
1.4.2 微型计算机的软件系统..... 13	
1.4.3 微型计算机的工作过程..... 14	
本章小结..... 15	
习题..... 15	
第 2 章 典型微处理器 17	
2.1 8086 微处理器简介..... 17	
2.1.1 8086 CPU 的内部功能结构..... 17	
2.1.2 8086/8088 CPU 的引脚功能..... 20	
2.2 8086 系统的存储器组织及 I/O 组织..... 23	
2.2.1 8086 系统的存储器组织..... 23	
2.2.2 8086 系统的 I/O 组织..... 25	
2.3 8086 系统的工作模式..... 25	
2.3.1 最小模式和最大模式..... 25	
2.3.2 最小模式系统..... 26	
2.3.3 最大模式系统..... 30	
2.4 8086 总线的操作时序..... 33	
2.4.1 系统的复位和启动操作..... 33	
2.4.2 最小与最大模式下的总线操作..... 34	
2.5 80x86 / Pentium 系列微处理器..... 38	
2.5.1 80286 微处理器..... 38	
2.5.2 80386 微处理器..... 40	
2.5.3 80486 微处理器..... 41	
2.5.4 Pentium 微处理器..... 42	
2.5.5 Itanium(安腾)微处理器..... 43	
2.5.6 嵌入式处理器..... 43	
本章小结..... 44	
习题..... 45	
第 3 章 指令系统与汇编语言 46	
3.1 指令格式与寻址方式..... 46	
3.1.1 指令格式..... 46	
3.1.2 寻址方式..... 47	
3.2 8086/8088 CPU 指令系统..... 49	
3.2.1 数据传送类指令..... 50	
3.2.2 算术运算类指令..... 53	
3.2.3 位操作类指令..... 58	
3.2.4 串操作类指令..... 62	
3.2.5 控制转移类指令..... 68	
3.2.6 处理器控制类指令..... 73	
3.3 汇编语言的程序与语句..... 74	
3.3.1 汇编语言源程序的格式..... 74	
3.3.2 汇编语言的语句..... 75	
3.4 汇编语言的伪指令..... 81	
3.4.1 符号定义伪指令..... 81	
3.4.2 数据定义伪指令..... 82	
3.4.3 段定义伪指令..... 84	
3.4.4 过程定义伪指令..... 86	
3.5 汇编语言程序设计基础..... 87	
3.5.1 程序设计的一般步骤..... 87	
3.5.2 程序设计的基本方法..... 88	

3.5.3 子程序设计及调用技术	92	5.3 存储器管理	146
3.6 DOS 功能子程序的调用	96	5.3.1 IBM PC/XT 中的存储	
3.6.1 概述	96	空间分配	146
3.6.2 基本 DOS 功能子程序	97	5.3.2 扩展存储器及其管理	147
本章小结	102	5.4 内部存储器技术的发展	150
习题	102	5.5 外部存储器	151
第 4 章 Proteus 仿真平台的使用	105	5.5.1 硬盘及硬盘驱动器	152
4.1 Proteus 简介	105	5.5.2 光盘存储器	154
4.2 Proteus ISIS 的基本使用	106	5.6 新型存储器	155
4.2.1 Proteus ISIS 操作界面及工具	106	5.6.1 Flash 存储器	155
4.2.2 基本操作	109	5.6.2 蓝光光盘	157
4.2.3 元件的使用	110	5.6.3 固态硬盘	158
4.2.4 连线	112	本章小结	159
4.2.5 器件标注	114	习题	159
4.2.6 编辑窗口的操作	115	第 6 章 输入/输出接口技术	161
4.3 Proteus ISIS 下 8086 的仿真	117	6.1 输入/输出接口概述	161
4.3.1 Proteus ISIS 电路原理图设计	117	6.1.1 输入/输出接口电路	161
4.3.2 Proteus 中配置 8086 编译工具	118	6.1.2 CPU 与外设数据传送的方式	164
4.3.3 Proteus 中编译 8086 汇编文件	120	6.1.3 I/O 端口的编址方式	167
4.3.4 仿真调试	122	6.2 中断系统	169
4.4 Proteus ISIS 下 8086 汇编语言程序		6.2.1 中断系统基本概念	169
设计示例	124	6.2.2 可编程中断控制芯片 8259A	176
4.4.1 顺序结构程序设计	124	6.2.3 8259A 的应用举例	183
4.4.2 循环结构程序设计	126	6.3 并行接口	186
4.4.3 分支结构程序设计	127	6.3.1 并行通信与并行接口	186
本章小结	128	6.3.2 可编程并行通信接口	
习题	128	芯片 8255A	188
第 5 章 存储器技术	129	6.3.3 8255A 的编程及应用	193
5.1 存储器简介	129	6.4 串行接口	196
5.1.1 存储器分类	129	6.4.1 串行通信及串行接口	196
5.1.2 存储器的主要性能参数	130	6.4.2 可编程串行通信接口	
5.1.3 存储系统的层次结构	131	芯片 8251A	199
5.2 读写存储器	131	6.4.3 8251A 的编程及应用	205
5.2.1 静态读写存储器 SRAM	131	6.5 DMA 控制技术	208
5.2.2 动态读写存储器 DRAM	133	6.5.1 可编程 DMA 控制器 8237A	208
5.2.3 EPROM	137	6.5.2 8237A 的编程及应用	213
5.2.4 EEPROM(E ² PROM)	138	6.6 定时器/计数器	215
5.2.5 闪速 EEPROM	140	6.6.1 可编程定时器/计数器 8253	215
5.2.6 存储器的连接	142	6.6.2 8253A 的编程及应用	218

6.7 显示接口	221	7.2.2 AGP 总线	290
6.7.1 CRT 显示系统	221	7.2.3 新型总线 PCI Express	293
6.7.2 LCD 显示及其接口	226	7.3 外总线	295
6.7.3 LED 显示器及其接口	228	7.3.1 RS232C 总线	295
6.8 键盘、鼠标接口	230	7.3.2 IEEE-488 总线	297
6.8.1 键盘接口	230	7.3.3 SCSI 总线	297
6.8.2 鼠标接口	234	7.3.4 USB 总线	297
6.9 并行打印机接口	236	7.3.5 IEEE 1394 总线	303
6.9.1 常用打印机及工作原理	236	7.4 现场总线	306
6.9.2 主机与打印机接口	236	7.4.1 现场总线的产生	306
6.9.3 打印机编程应用	239	7.4.2 现场总线控制系统的 技术特点	306
6.10 A/D 及 D/A 接口	244	7.4.3 现场总线技术的现状及 发展前景	307
6.10.1 D/A 转换器及其与 CPU 的接口	245	7.4.4 现场总线	308
6.10.2 A/D 转换器及其与 CPU 的接口	248	本章小结	314
6.11 Proteus ISIS 下输入/输出接口技术 应用示例	251	习题	314
6.11.1 8255 并行接口应用举例	251	第 8 章 微型计算机应用系统设计	
6.11.2 8253 定时/计数器应用举例	253	案例	315
6.11.3 8259 应用编程举例	254	8.1 微型计算机应用系统设计	315
6.11.4 8251 串行接口应用举例	258	8.1.1 概述	315
6.11.5 8237 应用举例	261	8.1.2 微型计算机应用系统 设计举例	317
6.11.6 A/D 转换举例	265	8.2 PCI 总线、USB 总线接口设计	322
6.11.7 D/A 转换举例	267	8.2.1 PCI 总线与 DSP 通信接口电路 设计	322
6.11.8 七段数码管显示应用举例	268	8.2.2 USB 总线与 DSP 通信接口电路 设计	325
6.11.9 4×4 矩阵键盘应用举例	271	8.3 Windows 驱动程序设计	329
6.11.10 16×16 点阵显示举例	274	8.3.1 驱动程序概述	330
本章小结	279	8.3.2 USB 设备 WDM 驱动 程序设计	333
习题	280	本章小结	334
第 7 章 微型计算机总线技术	282	习题	335
7.1 总线基本知识	282	参考文献	336
7.1.1 微型计算机总线概述	282		
7.1.2 微型计算机总线技术的现状	283		
7.1.3 计算机总线技术的未来 发展趋势	285		
7.1.4 总线分类和总线标准	286		
7.2 系统总线	288		
7.2.1 PCI 总线	288		

第 1 章 微型计算机概论

学习目标

本章重点介绍微型计算机的发展历程和系统组成,计算机中数据信息的表示及常用的逻辑单元和逻辑部件等内容。要求读者熟悉和掌握微型计算机的发展历程、工作特点、组成分类、应用领域等相关知识,为后续章节的学习打下良好的基础。

学习重点

- (1) 微型计算机的发展历程及各代微处理器的特点。
- (2) 微型计算机系统的组成及性能评价指标。
- (3) 计算机中的数据格式及机器数、带符号数和无符号数的表示方法。
- (4) BCD 码、ASCII 码的概念和应用。
- (5) 基本的逻辑运算的规律和常用的逻辑部件的特点。

1.1 微型计算机概述

计算机的发明标志着人类文明进入了一个新的历史阶段。20 世纪 70 年代初期,微电子技术和超大规模集成电路技术的发展,促成了以微处理器为核心的微型计算机的诞生。微型计算机现已渗透到国民经济的各个领域,极大地改善了人类的工作、学习以及生活方式,成为信息时代的主要标志。

1. 微型计算机的发展

自 1946 年第一台电子计算机问世以来,计算机的发展已经历了电子管、晶体管、中小规模集成电路、大规模和超大规模集成电路等 4 个阶段。进入 21 世纪后,随着生物科学、神经网络技术、纳米技术的飞速发展,生物芯片、神经网络技术也开始进入计算机领域——计算机的发展进入第 5 个发展阶段。

按体积、性能和价格的不同,计算机可分为巨型计算机、大型计算机、中型计算机、小型计算机和微型计算机。微型计算机是指以微处理器为核心,并配以存储器、输入/输出接口电路及其设备的计算机。微型计算机采用超大规模集成电路技术,将运算器和控制器——微处理器(Microprocessor)集成在一片硅片上。

随着微电子与超大规模集成电路技术的发展,微型计算机技术的发展基本遵循摩尔定律,即微处理器集成度每隔 18 个月翻一番,芯片性能随之提高一倍左右。通常,微型计算机的发展是以微处理器的发展为表征的。以其字长和功能来分,微处理器的发展经历了如下几个阶段:

- (1) 1971—1973 年为 4/8 位低档微处理器时代,代表芯片是 Intel 4004 和 Intel 8008,采

用 PMOS 工艺,集成度为 2300 元件/片,基本指令执行时间为 $20\ \mu\text{s}\sim 50\ \mu\text{s}$,主频在 500 kHz 以下,基本指令有 48 条。第一代微处理器主要用于家电和简单控制场合。

(2) 1973—1977 年为 8 位中档微处理器时代,代表芯片是 MC6800、Z80、Intel 8080/8085 等,采用 NMOS 工艺,集成度较第一代提高 4 倍,基本指令执行时间为 $2\ \mu\text{s}\sim 10\ \mu\text{s}$,主频高于 1 MHz,基本指令包括 70 多条。第二代微处理器主要用于电子仪器等。

(3) 1978—1984 年为 16 位微处理器时代,代表芯片是 Intel 8086/8088、MC6800、Z8000,采用 HMOS 工艺,集成度为 2~7 万元件/片,基本指令执行时间为 $0.5\ \mu\text{s}$,主频为 4 MHz~8 MHz。第三代微处理器的计算机指令系统完善,采用流水线技术、多级中断、多种寻址方式、段寄存器等结构,能够与协处理器相配合进行浮点运算。

(4) 1985—1992 年为 32 位微处理器时代,它标志着微处理器跨入了第四代,代表芯片是 Intel 80386、Intel 80486、MC68040 等,采用 HOMS/CMOS 工艺,集成度为 100 万元件/片,基本指令执行速度为 25 MIPS,主频为 16 MHz~25 MHz。第四代微处理器引入了高速缓存和采用精简指令集,其体系结构较 16 位机发生了概念性变化。

(5) 1993 年推出的 32 位 Pentium 微处理器 P5,采用 $0.6\ \mu\text{m}$ 的静态 CMOS 工艺,集成度为 350 万元件/片,基本指令执行时间为 $0.5\ \mu\text{s}$,主频在 60 MHz 以上,采用扩展总线,设置高速程序缓存、数据缓存、超流水线结构。两年后推出的 Pentium Pro 系列微处理器 P6,主频为 133 MHz,设有两级缓存,采用动态执行技术,性能大大提高。而后又推出了具有 MMX 技术,附加多媒体声像处理指令的 Pentium II,可用于多媒体应用领域。

截至目前,Intel 系列的微处理器中,最高主频已达 3.8 GHz。表 1-1 给出了 Intel 80x86/Pentium 系列部分 CPU 的主要性能参数。

表 1-1 Intel 80x86/Pentium 系列部分 CPU 的主要性能参数

微处理器	推出时间	生产工艺(μm)	首批时钟频率(MHz)	集成度(百万个)	寄存器位数	数据总线位数	最大寻址空间	高速缓存大小
8086	1978 年	10	8	0.040	16	16	1 MB	无
80286	1982 年	2.7	12.5	0.125	16	16	16 MB	无
80386DX	1985 年	2	20	0.275	32	32	4 GB	无
80486DX	1989 年	1, 0.8	25	1.200	32	32	4 GB	8KB L1
Pentium	1993 年	0.8, 0.6	60	3.100	32	64	4 GB	16KB L1
Pentium Pro	1995 年	0.6	200	5.500	32	64	64 GB	16KB L1/256KB L2
Pentium II	1997 年	0.35	300	7.500	32	64	64 GB	32KB L1/ 256KB L2
Pentium III	1999 年	0.18	500	9.500	32	64	64 GB	32KB L1/512KB L2
Pentium IV	2000 年	0.13	1300	42.00	32	64	64 GB	128KB L1/512KB L2

2. 微型计算机的特点

微型计算机运算速度快,计算精度高,高集成度使得微处理器非常稳定。由于微型计算机硬件平台开放,易于扩展,适应性强,因此微处理器的配套应用芯片和软件丰富,更新也很快。此外,微型计算机还具有体积小、重量轻、耗电省、维护方便及造价低廉等特点。

3. 微型计算机的应用

科学计算是微型计算机应用的主要领域,其应用包括卫星发射控制、航天飞机制造、高层建筑设计、机械产品设计、生物信息学研究、基因测序、医学病理分析与处理等。

过程控制是微型计算机在工业应用中的重要领域,其应用包括大型工业锅炉控制、铁路调度控制、数控机床控制,以及由上、下位微型计算机构成的分布式工业生产自动控制系统等。嵌入式系统的发展和应用使工业控制的应用领域更加广泛,市场应用前景更加广阔。

低档的微型计算机在仪器仪表和家电的智能控制方面的应用,取代了过去的硬件逻辑电路对仪器仪表和家电的控制,用程序的重复执行以及循环控制,可以做到电路最省、控制更佳,并可通过修改程序来修改控制方案,因而灵活多变,可靠性高。

计算机辅助设计与制造(CAD/CAM)借助微型计算机调整、修改产品设计,CAM 围绕中心数控机床及其自动化设备,用以完成部件的加工、运输、组装、测量、检查等功能,CAD 与 CAM 的集成——CAD/CAM 一体化,是今后工业自动化发展的重要方向。

人工智能的主要目标是利用计算机模拟人的大脑,实现计算机对于知识学习、理解与推理、信息处理的思维过程的研究学科。人工智能理论的新突破,特别是人工神经网络和 DNA 芯片技术的研究,急需大型并行计算机的模拟计算和新型计算机的研究。

利用微型计算机可以构成计算机网络,实现微机系统的软硬件资源和数据资源的共享。

1.2 计算机中数据信息的表示

计算机最主要的功能之一是信息处理,这些信息包括数值、文字、声音、图形和图像等,各种信息以数字化形式传输、存储和处理。因此,各种数制与信息编码是至关重要的。

1.2.1 数据格式及机器数

1. 数据格式

在微处理器中要进行整数和小数运算,如何处理小数点的位置是十分重要的,通常在计算机中经常采用定点格式或浮点格式来表明小数点的位置。

(1) 定点格式。在定点格式中,小数点在数据中的位置固定不变。通常,小数点的位置确定后,在运算中不再考虑小数点的问题,因而,小数点不占用存储空间。定点数表示简单,但数的取值范围小,精度低。

(2) 浮点格式。采用浮点格式的机器中的数据的小数点位置可变。浮点数的一般格式为

$$N = R^e \cdot m$$

其中: N 为浮点数或实数; m 为浮点数的尾数,是纯小数; e 为浮点数的指数,是整数; R 为基数,是常数。

机器中的浮点数用尾数和阶码及其符号位表示。尾数用定点小数表示,用于给定有效数字的位数并决定浮点数的表示精度;阶码用定点整数表示,用于指明小数点在数据中的位置并决定浮点数的表示范围。

(3) 带符号数和无符号数。对于整数而言,如果其最高有效位为符号位,则该数为带

符号数；反之，如果其最高有效位为数值位，则该数为无符号数。无符号数不一定是正数。当进行数据处理时，若不需要考虑数的正负，则可以使用无符号数。带符号数和无符号数的取值范围不同，对于字长为 8 位的定点整数，无符号数的取值范围是 $0 \leq X \leq 255$ ，带符号数的取值范围是 $-128 \leq X \leq 127$ 。

2. 机器数的表示方法

在计算机中，带符号数常用的表示方法有原码、反码和补码 3 种。这些表示方法都将数的符号数码化。通常“+”用“0”表示，“-”用“1”表示。为了区分书写时表示的数和机器中编码表示的数，我们称前者为真值，后者为机器数，即数值连同符号数码“0”或“1”在机器中的一组二进制数表示形式称为机器数，而它所表示的数值连同符号“+”或“-”称为机器数的真值。把机器数的符号位也当作数值的数，就是无符号数。

为了表示方便，常把 8 位二进制数称为字节，16 位二进制数称为字，32 位二进制数称为双字。对于机器数，应将其用字节、字或双字表示，所以只有 8 位、16 位或 32 位机器数的最高位才是符号位。

正数的原码、反码、补码相同，即 $[x]_{\text{原}} = [x]_{\text{反}} = [x]_{\text{补}}$ 。

负数的机器数求解方法如下：

(1) 反码：将其原码符号位保持不变，数值位按位取反。

(2) 补码：将反码末位加 1。

当计算机采用不同的码制时，运算器和控制器的结构将不同。由于补码具有唯一性，因此小型计算机和微型计算机大都为补码机。

(1) 计算机中引入补码可以使符号位和数值位成为一体，共同参与运算，运算结果的符号位由运算得出。

(2) 减法可以转换成加法运算来完成，乘法和除法可以通过加法和移位运算来完成。这样，二进制数的四则运算只须加减法和移位运算即可完成。

由此可见，计算机中引入补码的目的是简化运算方法，从而简化运算器的结构和设计。

【例 1-1】 用 8 位字长表示 -109、54、0.625 和 -0.25 的原码、反码、补码。

解	十进制数	二进制数	原码	反码	补码
	-109	-1101101	11101101	10010010	10010011
	54	110110	00110110	00110110	00110110
	0.625	0.101	0.1010000	0.1010000	0.1010000
	-0.25	-0.01	1.0100000	1.1011111	1.1100000

1.2.2 数字信息编码

所谓编码，就是用少量的基本符号，按照一定的排列组合原则表示大量复杂多样信息的一种操作。基本符号的种类和排列组合规则是信息编码的两大要素，下面分别简单介绍计算机中信息编码和常用数据表示的几种方法。

1. 二进制编码的十进制数

由于计算机内部采用二进制数，而外部数据的输入/输出使用十进制数，因此采用编码方式来完成二—十进制数的转换。8421BCD 码就是用 4 位二进制数的编码来表示十进制数，

见表 1-2。采用 8421BCD 码可以直接使用二进制数部件完成十进制数的存储和运算。

表 1-2 常用编码形式十进制数的对应关系

十进制数	十六进制数	8421BCD	十进制数	十六进制数	8421BCD
0	0	0000 0000	8	8	0000 1000
1	1	0000 0001	9	9	0000 1001
2	2	0000 0010	10	A	0001 0000
3	3	0000 0011	11	B	0001 0001
4	4	0000 0100	12	C	0001 0010
5	5	0000 0101	13	D	0001 0011
6	6	0000 0110	14	E	0001 0100
7	7	0000 0111	15	F	0001 0101

2. 字符编码

ASCII 码(American Standard Code for Information Interchange)是国际通用的字符编码标准。ASCII 码采用 7 位二进制数编码表示 128 个字符, 见表 1-3, 其中 34 个起控制作用的编码称为功能码, 其余的 94 个符号称为信息码, 供书写程序和描述命令之用。在确定某个字符的 ASCII 码时, 先确定该字符在表中所对应的行与列, 列对应高位码 $d_6d_5d_4$, 行对应低位码 $d_3d_2d_1d_0$, 高位码与低位码的组合就是该字符的 ASCII 码。

表 1-3 ASCII 码字符表

$d_3d_2d_1d_0$	$d_6d_5d_4$							
	000	001	010	011	100	101	110	111
0000	NUL	DLE	SPACE	0	@	P	`	p
0001	SOH	DC1	!	1	A	Q	a	q
0010	STX	DC2	"	2	B	R	b	r
0011	ETX	DC3	#	3	C	S	c	s
0100	EOT	DC4	\$	4	D	T	d	t
0101	ENQ	NAK	%	5	E	U	e	u
0110	ACK	SYN	&	6	F	V	f	v
0111	BEL	ETB	,	7	G	W	g	w
1000	BSB	CAN	(8	H	X	h	x
1001	TAB	EM)	9	I	Y	i	y
1010	LF	SUB	*	:	J	Z	j	z
1011	VT	ESC	+	;	K	【	k	{
1100	FF	FS	,	<	L	\	l	
1101	CR	GS	-	=	M	】	m	}
1110	SO	RS	.	>	N	^	n	~
1111	SI	US	/	?	O		o	DEL

注: “SPACE”表示空格; “LF”表示换行; “FF”表示换页; “CR”表示回车; “DEL”表示删除; “BEL”表示振铃。

3. 汉字编码

当计算机用于汉字处理时，可用若干位二进制编码来表示一个汉字。通常，一个汉字的编码可用内码、字模码和外码来描述。内码是用于汉字的存储、交换等操作的计算机内部代码。一个汉字内码通常用两个字节表示，且这两个字节的最高位均为 1，以区别英文字符的 7 位 ASCII 码。字模码是汉字的输出编码，字库中存放的就是字模码。外码是汉字的输入码，用来输入汉字的编码。

1.3 逻辑单元与逻辑部件

1.3.1 二进制数的逻辑运算与逻辑电路

计算机除了可进行基本的算术运算外，还可对两个或一个无符号二进制数进行逻辑运算。计算机中的逻辑运算主要包括“逻辑非”、“逻辑与”、“逻辑或”和“逻辑异或”4 种基本运算。下面介绍这 4 种基本逻辑运算及实现这些运算的逻辑电路。

1. 逻辑非

逻辑非运算也称“求反”。对二进制数进行逻辑非运算，就是按位求它的“反”，常在逻辑变量上方加一横线来表示。

例如， $A = 01100001B$ ， $B = 11001011B$ ，对 A 和 B 求逻辑非，则有

$$\bar{A} = 10011110B$$

$$\bar{B} = 00110100B$$

实现逻辑非运算的电路称为非门，又称反相器。它只有一个输入和一个输出。它的国标符号如图 1-1 所示。

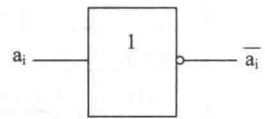


图 1-1 非门的国标符号

2. 逻辑与

对两个二进制数进行逻辑与运算，就是按位求它们的“与”，又称“逻辑乘”，常用符号“ \wedge ”或“ \cdot ”来表示。二进制数逻辑与的规则为 $0 \wedge 0 = 0$ ， $0 \wedge 1 = 0$ ， $1 \wedge 0 = 0$ ， $1 \wedge 1 = 1$ 。

例如， $01100001B$ 和 $11001001B$ 逻辑与的算式如下：

$$\begin{array}{r} 01100001 \\ \wedge 11001001 \\ \hline 01000001 \end{array}$$

即 $01100001B \wedge 11001001B = 01000001B$ 。

实现逻辑与运算的电路称为与门。2 输入与门的国标符号如图 1-2 所示。

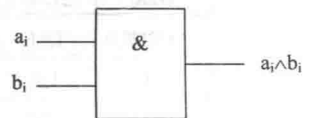


图 1-2 与门的国标符号

3. 逻辑或

对两个二进制数进行逻辑或运算，就是按位求它们的“或”，又称“逻辑加”，常用符号“ \vee ”或“+”来表示。二进制数逻辑加的规则为 $0 \vee 0 = 0$ ， $0 \vee 1 = 1$ ， $1 \vee 0 = 1$ ， $1 \vee 1 = 1$ 。

例如， $01100001B$ 和 $11001001B$ 逻辑加的算式如下：

$$\begin{array}{r}
 01100001 \\
 \vee 11001001 \\
 \hline
 11101001
 \end{array}$$

即 $01100001B \vee 11001001B = 11101001B$ 。

实现逻辑加运算的电路称为或门。2 输入或门的国标符号如图 1-3 所示。

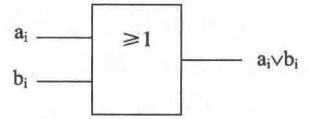


图 1-3 或门的国标符号

4. 逻辑异或

对两个二进制数进行逻辑异或运算，就是按位求它们的模 2 和，所以逻辑异或又称“按位加”，常用符号“ \oplus ”来表示。二进制数的逻辑异或运算规则为 $0 \oplus 0 = 0$ ， $0 \oplus 1 = 1$ ， $1 \oplus 0 = 1$ ， $1 \oplus 1 = 0$ 。

例如， $01100001B$ 和 $11001001B$ 逻辑异或的算式如下：

$$\begin{array}{r}
 01100001 \\
 \oplus 11001001 \\
 \hline
 10101000
 \end{array}$$

即 $01100001B \oplus 11001001B = 10101000B$ 。

注意：按位加与普通整数加法的区别是它仅按位相加，不产生进位。

实现逻辑异或运算的电路称为异或门。2 输入异或门的国标符号如图 1-4 所示。

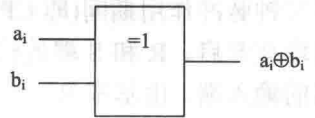


图 1-4 异或门的国标符号

异或门的特点是，只有当输入的两个变量相异时，输出为高(1)，否则输出为低(0)。

1.3.2 常用逻辑部件

逻辑部件是用来对二进制数进行寄存、传送和变换的数字部件，其种类繁多，本书简单地介绍微型计算机中常用的几种逻辑部件。构成逻辑部件的基本单元电路是触发器。

1. 触发器

触发器是具有记忆功能的基本逻辑单元电路。它能接收、保存和输出逻辑信号 0 和 1。各类触发器都可以由逻辑门电路组成。

1) 基本 RS 触发器

基本 RS 触发器是最简单的触发器，它是将两个与非门的输入与输出交叉连接构成的，如图 1-5 所示。触发器的两个输入端分别是 \bar{R} 和 \bar{S} ，其中 \bar{S} 端称为置 1 或置位(Set)端， \bar{R} 端称为置 0 或复位(Reset)端。触发器有两个输出端 Q 和 \bar{Q} ，在正常工作时，它们总是处于互补的状态。用 Q 端的状态来表示触发器的状态。由与非门的逻辑功能决定，要使触发器为 1 状态，可使 $\bar{S} = 0$ ， $\bar{R} = 1$ 。同样，要使触发器为 0 状态，需令 $\bar{R} = 0$ ， $\bar{S} = 1$ 。触发器一旦为 1 状态(或 0 状态)， \bar{S} (或 \bar{R}) 端从 0 变成 1，触发器将保持 1 状态(或

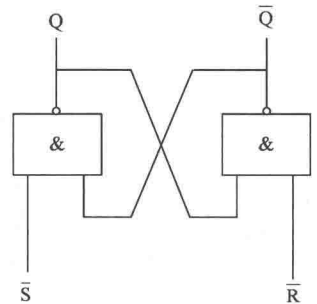


图 1-5 基本 RS 触发器的结构图

0 状态)不变。即 $\bar{R} = 1, \bar{S} = 1$ 时, 触发器的状态不变。 \bar{R} 和 \bar{S} 不能同时为 0, 因为同时为 0 时, Q 和 \bar{Q} 都为 1。当这种输入状态消失时, 触发器的 Q 端可能为 0, 也可能为 1, 但到底是 0 还是 1, 是不确定的。

2) 同步 RS 触发器

基本 RS 触发器中, 输入端的触发信号直接控制触发器的状态。但在实际应用中, 还希望触发器能够受一个时钟信号控制, 做到按时钟信号的节拍翻转。这个控制信号称为时钟脉冲 CP(Clock Pulse)。引入 CP 后, 触发器的状态不是在输入信号(R、S 端)变化时立即转换, 而是等待时钟信号到达时才转换。在多个这种触发器组成的电路中, 各触发器受同一个时钟控制, 触发器都在同一个时刻翻转, 因此称为同步 RS 触发器, 而基本 RS 触发器称为异步 RS 触发器。

同步 RS 触发器的电路结构如图 1-6 所示。该电路由基本 RS 触发器和控制电路两部分组成。在时钟脉冲未到来时(即 $CP = 0$ 时), 由于控制电路的两个与非门均被封锁, 它们的输出都为 1, 因此基本 RS 触发器维持原状态不变。在时钟脉冲作用期间(即 $CP = 1$ 时), 控制电路的两个与非门均被开启, R 和 S 端的输入被反相后送到基本 RS 触发器的输入端。由基本 RS 触发器的逻辑功能可知, $RS = 01$ 时, 触发器被置位; $RS = 10$ 时, 触发器被复位; $RS = 00$ 时, 触发器的状态不变; $RS = 11$ 的输入状态, 同步 RS 触发器是不允许出现的。

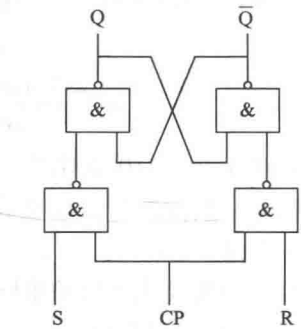


图 1-6 同步 RS 触发器的结构

3) D 触发器

同步 RS 触发器工作时, 不允许 R 和 S 端的输入信号同时为 1。如果将 R 端改接到控制电路另一个与非门的输出端, 只在 S 端加入输入信号, S 端改称为 D 端, 同步 RS 触发器就转换成了 D 触发器。D 触发器的电路结构与逻辑符号如图 1-7 所示。由于总是将 D 端的输入反相后作为另一个与非门的输入信号, 故无论 D 端的状态如何, 都满足 RS 触发器的约束条件, 不会出现不允许的输入状态。由 RS 触发器的特性可直接求出 D 触发器的特性。不管 D 触发器 Q 端的原状态 Q^n 如何, 次态 Q^{n+1} 总是与时钟脉冲来到时 D 端的输入状态相同。

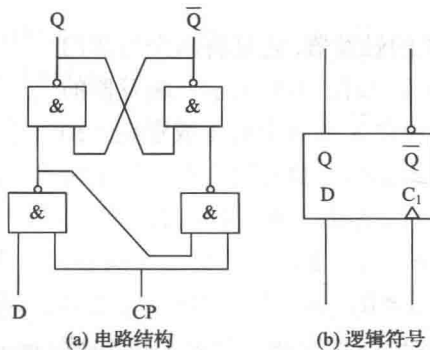


图 1-7 D 触发器的电路结构与逻辑符号

有些 D 触发器还有异步复位端 \overline{R}_D 和异步置位端 \overline{S}_D ，利用它们也能实现置数的功能。

4) JK 触发器

在同步 RS 触发器的基础上，增加 J 和 K 输入端及两条反馈线，即可组成 JK 触发器。JK 触发器的电路结构与逻辑符号如图 1-8 所示。由于 Q 和 \overline{Q} 的互补关系，控制电路的两个与非门不会同时开启，因而 JK 的任一种输入状态都是允许的，不再需要满足 RS 触发器的约束条件。

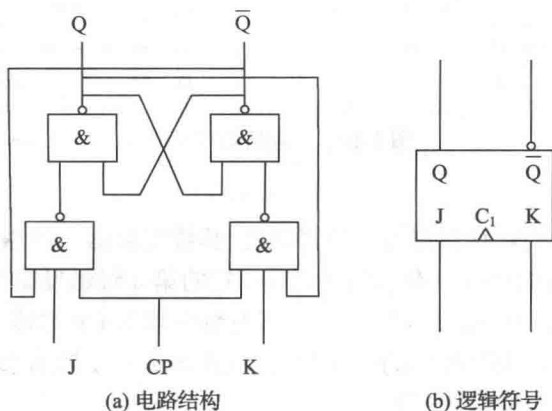


图 1-8 JK 触发器的电路结构与逻辑符号

5) T 触发器

将 JK 触发器的 J、K 两端连在一起作为 T 输入端，即可得到 T 触发器。T 触发器的电路结构与逻辑符号如图 1-9 所示。

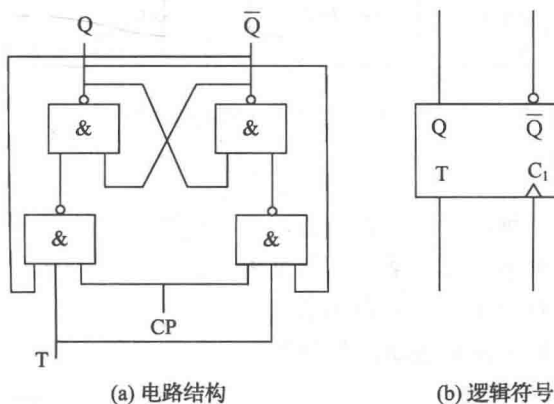


图 1-9 T 触发器的电路结构与逻辑符号

2. 寄存器

寄存器是计算机中用得最多的逻辑部件之一，用来存放二进制信息，具有接收二进制数码和寄存二进制数码的功能。寄存器一般由触发器组成。触发器具有两个稳定状态，每一个触发器可以存放 1 位二进制数，N 个触发器可以构成存放 N 位二进制数的寄存器。图 1-10 为由 4 个具有异步复位端的 D 触发器构成的寄存器的逻辑图。当 $\overline{CR} = 1$ 时 (\overline{CR} 为置零端。当 $\overline{CR} = 0$ 时，寄存器的四个 Q 端都为 0)，时钟脉冲将待送的数码 $D_4D_3D_2D_1$ 送到寄

寄存器的 $Q_4Q_3Q_2Q_1$ 保存起来。

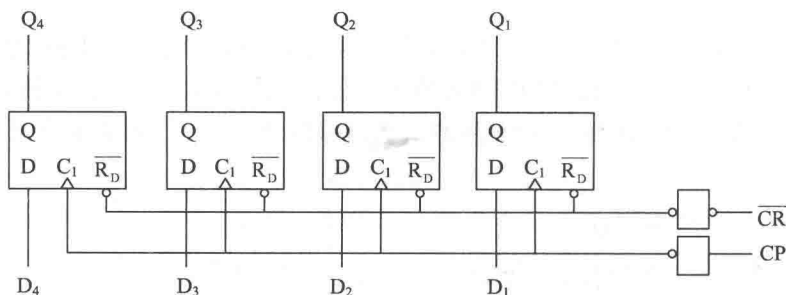


图 1-10 寄存器的逻辑图

3. 移位寄存器

具有移位逻辑功能的寄存器称为移位寄存器。移位寄存器一般由 D 触发器构成。图 1-11 为由 4 个 D 触发器构成的移位寄存器的逻辑图。它的第 4 级触发器的 D 端接输入信号，其余各触发器的 D 端接前一级触发器的 Q 端，所有触发器的 CP 端连在一起接收时钟脉冲信号。每来一个时钟脉冲，来自外部的输入数码(即第 4 级触发器的 D 端的输入信号)便输入一位，已被寄存的数码依次右移一位。

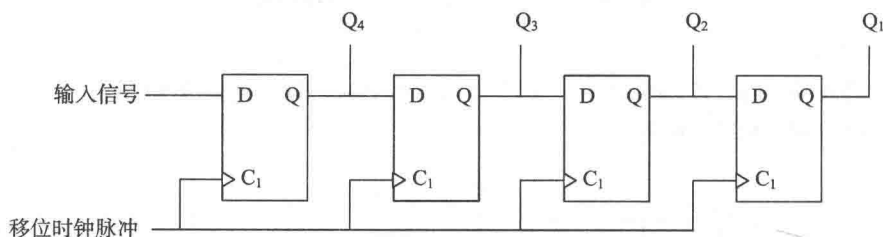


图 1-11 移位寄存器的逻辑图

4. 计数器

计数器是计算机中一种常用的逻辑部件，它不仅能存储数据，而且还能记录输入脉冲的个数。计数器的种类繁多，可以从不同角度来分类：按工作方式，可分为同步计数器和异步计数器；按加减计数顺序，可分为加法计数器和减法计数器；按进位制，可分为二进制计数器、十进制计数器和任意进制计数器等。

1) 异步二进制加法计数器

由 JK 触发器构成的 3 位异步二进制加法计数器的逻辑图如图 1-12 所示。其工作过程如下：初始时，将计数器置为全 0 状态(即 $Q_3Q_2Q_1$ 为 000)。第 1 个计数脉冲来到后，第 1 级触发器翻转， Q_1 由 0 变为 1，第 2、3 级触发器因时钟端无触发脉冲，它们维持原状态不变，故计数器的状态 $Q_3Q_2Q_1$ 为 001。第 2 级计数脉冲来到后，第 1 级触发器又翻转， Q_1 由 1 变为 0，第 2 级触发器因其时钟输入端有脉冲下降沿的作用，也进行翻转， Q_2 由 0 变为 1， Q_3 仍保持原状态，计数器的状态 $Q_3Q_2Q_1$ 为 010。按照这样的顺序工作下去，直至第 7 个计数脉冲来到后，计数器的状态 $Q_3Q_2Q_1$ 为 111。此时再来一个计数脉冲，计数器又回