

• 高等学校规划教材

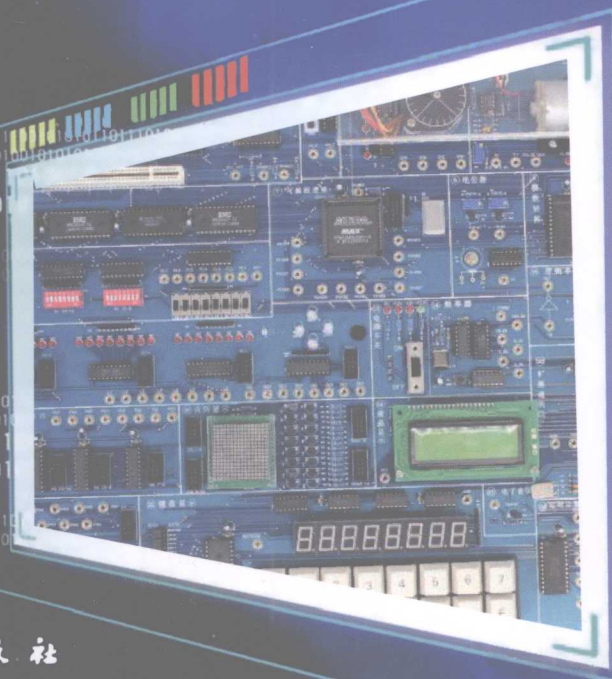


32位微机 原理与接口技术



32WEI WEIJI
YUANLI YU JIEKOU JISHU

马兴录 宋廷强 陈为 等编



化学工业出版社

• 高等学校规划教材



32

位微机

原理与接口技术

32WEI WEIJI

YUANLI YU JIEKOU JISHU



图书在版编目(CIP)数据

32位微机原理与接口技术

张毅刚 等编

北京：化学工业出版社，2002

ISBN 978-7-122-02280-2

I. ① 3… II. 张… III. ①

① 微机原理与接口技术—高等学校—教材



化学工业出版社

· 北京 ·

本书以 Intel 公司的 IA32 系列微处理器为主线, 系统讲述了微机原理、汇编语言程序设计、微型计算机的组成以及接口技术。微机原理部分主要包括: 32 位微处理器的工作原理及其指令系统。汇编语言程序设计详细讲述了汇编语言程序结构、开发过程、系统功能调用、结构化程序设计, 并列举了大量编程实例。微型计算机的组成部分主要包括: 总线技术、存储器系统、中断系统以及 DMA 控制器。接口技术部分包括 I/O 系统、并行接口、定时/计数器、中断控制器、串行通信接口、模拟接口等内容。最后, 还简单介绍了 32 位微处理器的保护模式, 为读者在保护模式下进行开发奠定基础。

本书在每章后面配有习题, 并有配套的《32 位微机原理与接口技术实验指导》及电子课件可供选用。本书可作为高等院校本科教材使用, 也可供工程技术人员参考。

图书在版编目(CIP)数据

32 位微机原理与接口技术 / 马兴录等编. —北京: 化学工业出版社, 2009.8
高等学校规划教材
ISBN 978-7-122-05980-2

I. 3… II. 马… III. ①微型计算机-理论-高等学校-教材②微型计算机-接口-高等学校-教材 IV. TP36

中国版本图书馆 CIP 数据核字 (2009) 第 098469 号

责任编辑: 王清颢
责任校对: 陶燕华

装帧设计: 韩 飞

出版发行: 化学工业出版社 (北京市东城区青年湖南街 13 号 邮政编码 100011)

印 装: 三河市延风印装厂

787mm×1092mm 1/16 印张 19¼ 字数 502 千字 2009 年 9 月北京第 1 版第 1 次印刷

购书咨询: 010-64518888 (传真: 010-64519686) 售后服务: 010-64518899

网 址: <http://www.cip.com.cn>

凡购买本书, 如有缺损质量问题, 本社销售中心负责调换。

定 价: 39.00 元

版权所有 违者必究

前 言

微机原理与接口技术是计算机、信息、通信、自动化等专业教育中一门十分重要的专业基础课。本书编写的目的是让读者从理论和实践上掌握微型计算机的工作原理、汇编语言程序设计、微机的基本组成及常用接口技术，建立微机系统整体概念，了解当今计算机硬件的新技术和新理论。通过对本书内容的学习，读者可以比较系统地了解微机系统的组成原理及其硬件结构，掌握汇编语言的程序设计，掌握基本微机系统的接口电路的设计及编程方法，具备微机应用系统软、硬件开发的初步能力。

当今微处理器的发展一直遵循著名的“摩尔定律”。从 20 世纪 70 年代后期出现第三代 16 位微处理器至今，CPU 经历了飞速发展。在微型计算机中，CPU 已经从 32 位更换为 64 位，从单核 CPU 更换为多核 CPU。在令人眼花缭乱的背后，是技术与制造工艺的不断创新。不过，原理性的东西并没有发生太大的变化，而且当前 32 位 CPU 在市场上，特别是在嵌入式系统中还占主流。因此，本书在编写过程中以 Intel 公司的 IA32 系列微处理器为主线，重点讲述微处理器的工作原理。

全书共分 13 章，从内容组织上可分为四大部分：微机原理、汇编语言程序设计、微机组成及接口技术。微机原理部分主要包括第 2 章、第 3 章和第 13 章。汇编语言程序设计在第 4 章。微机组成主要包括第 5 章、第 6 章、第 8 章和第 12 章，这是构成一台微机的必要组成部分。接口技术则包括第 7 章、第 9 章、第 10 章和第 11 章。

第 1 章是基础知识部分。介绍了计算机的一些基础知识，主要包括计算机发展简史、微型计算机系统的结构及其主要技术指标、计算机中的数制、布尔代数基础、逻辑电路基础、二进制数的运算及其加法电路、计算机中的编码、浮点数基本概念等。

第 2 章从微机的简化模型入手，开始讲述微机的内部工作原理；然后以 16 位微处理器 8086 为过渡，讲述 IA-32 架构微处理器的功能结构及编程结构；最后以 32 位微处理器 Pentium 为例，讲述 32 位微处理器的外部引脚及工作时序。

第 3 章介绍了 IA32 系列微处理器的寻址方式及其基本指令集。

第 4 章介绍宏汇编语言程序的结构、伪指令以及程序设计的一般过程和各種基本程序结构，然后介绍系统功能调用，最后给出了大量的编程实例。

第 5 章介绍了目前微机系统中常用的各种总线，包括常用内部总线 PCI 以及工业中常用的 PC104 总线等；外部总线 IEEE-488 总线及 USB 总线。

第 6 章主要讨论作为内存的半导体存储器。在简要介绍存储器分类和基本存储元件电路的基础上，重点介绍了常用的几种典型存储器芯片及其与 CPU 之间的连接与扩展问题，并简要介绍了目前广泛应用的几种新型存储器。

第 7 章介绍了接口技术的基础知识，包括接口的定义、功能及结构，I/O 端口的编址方式以及 CPU 与外设之间的数据传送方式。为学习后续的各类接口奠定基础。

第 8 章介绍了计算机系统不可缺少的重要组成部分——中断系统。介绍了一般中断系统概念、微机的中断系统功能及中断管理专用芯片 8259A。

第 9 章介绍了并行通信及定时/计数技术，重点介绍了可编程并行接口芯片 8255 及可编程定时/计数器 8253 的用法。

第 10 章介绍了串行通信技术以及串行通信接口芯片 8251A。

第 11 章介绍了数模和模数转换的原理以及常用 A/D、D/A 芯片的功能。

第 12 章介绍了为实现大批量数据的快速传输而采用的 DMA 传送方式。重点介绍了 DMA 控制器 8237。

第 13 章介绍了 IA32 系列微处理器保护模式下的运行机制及编程方法。本章内容可作为选学部分。

本书第 1 章由曲英杰编写；第 2 章和第 12 章由马兴录编写；第 3 章和第 4 章由范玮编写；第 5 章和第 6 章由宋廷强编写；第 7 章和第 8 章由陈为编写；第 9 章和第 10 章由朱桂新编写；第 11 章和第 13 章由肖传伟编写。全书由马兴录统稿。

由于编者的实际工作经验及水平的限制，本书必会存在一些不当之处，恳请读者批评指正。

编 者

目 录

第 1 章 计算机基础	1
1.1 绪论	1
1.1.1 电子计算机发展简史	1
1.1.2 计算机应用领域及发展趋势	2
1.1.3 微型计算机系统的组成 及其主要技术指标	3
1.2 计算机中的数制	5
1.2.1 数制的基本概念	5
1.2.2 数制之间的转换	6
1.3 布尔代数基础	7
1.3.1 基本逻辑运算	7
1.3.2 基本运算规律	8
1.3.3 逻辑函数的表示方法	8
1.3.4 真值表与逻辑表达式 之间的相互转换	8
1.3.5 逻辑函数的化简	9
1.4 逻辑电路基础	9
1.5 二进制数的运算及其加法电路	10
1.5.1 二进制数据算术运算规则	10
1.5.2 半加器电路设计	11
1.5.3 全加器电路设计	11
1.5.4 多位二进制数的加法电路设计	12
1.6 计算机中的编码	13
1.6.1 二进制数值数据的编码方法	13
1.6.2 补码加法器/减法器电路	14
1.6.3 其他编码	16
1.7 浮点数基本概念	18
习题	19
第 2 章 微处理器	20
2.1 微型计算机简化模型	20
2.1.1 微型计算机的总体结构	20
2.1.2 简化模型的组成	20
2.1.3 模型机的运行过程	22
2.1.4 指令系统	23
2.2 处理器的功能结构	25
2.2.1 IA-32 架构微处理器的发展历史	25
2.2.2 8086 微处理器的功能结构	26
2.2.3 Pentium 微处理器的功能结构	29
2.3 IA-32 微处理器的编程结构	30
2.3.1 IA-32 微处理器的工作模式	30
2.3.2 IA-32 微处理器的编程结构	31
2.3.3 存储器组织	36
2.4 Pentium 微处理器的外部引脚	37
2.5 Pentium 微处理器的典型工作时序	40
习题	41
第 3 章 指令系统	43
3.1 概述	43
3.2 寻址方式	44
3.3 IA32 微处理器的基本指令集	50
3.3.1 数据传送指令	50
3.3.2 算术运算类指令	58
3.3.3 逻辑指令	67
3.3.4 串处理指令	72
3.3.5 控制转移指令	79
3.3.6 处理机控制指令	89
习题	90
第 4 章 汇编语言程序设计	95
4.1 汇编语言语句	95
4.1.1 汇编语言语句种类及其格式	95
4.1.2 汇编语言语句中各项的表示方法	96
4.2 伪指令	100
4.2.1 处理器选择伪指令	101
4.2.2 段定义伪指令	101
4.2.3 假定伪指令	104
4.2.4 数据定义伪指令	104
4.2.5 符号定义伪指令	105
4.2.6 地址计数器与定位伪指令	105
4.2.7 过程(子程序)定义伪指令	106
4.2.8 源程序开始和结束伪指令	107
4.3 汇编语言程序的结构	107
4.4 汇编语言程序的开发过程	108
4.5 汇编语言程序结构设计	109
4.5.1 顺序程序设计	109
4.5.2 分支程序设计	111

4.5.3 循环程序设计	114	6.4.3 存储器连接举例	183
4.5.4 子程序设计	119	习题	184
4.5.5 宏指令	124	第7章 输入输出接口	186
4.6 系统功能调用	128	7.1 概述	186
4.6.1 键盘功能调用	128	7.1.1 设置接口电路的目的	186
4.6.2 显示功能调用	130	7.1.2 接口的基本功能	186
4.6.3 返回操作系统	130	7.1.3 接口电路中的信息	187
4.7 汇编语言设计实例	131	7.1.4 接口的基本结构	187
4.7.1 键盘及显示器操作	131	7.2 I/O 端口的编址	188
4.7.2 代码转换	135	7.2.1 I/O 端口的编址方式	188
4.7.3 数值计算和数据处理	136	7.2.2 端口地址译码	188
4.7.4 字符串处理	139	7.3 CPU 与外设之间的数据传送方式	191
习题	142	7.3.1 程序控制方式	191
第5章 总线技术	146	7.3.2 中断传送方式	194
5.1 总线的基本概念	146	7.3.3 直接存储器存取方式	194
5.1.1 总线的分类	146	习题	195
5.1.2 总线标准	148	第8章 中断系统	197
5.1.3 总线主要性能指标	149	8.1 中断的基本概念	197
5.1.4 总线控制部件与总线传输	150	8.1.1 中断的定义	197
5.1.5 总线的层次化结构	153	8.1.2 中断请求信号的产生	197
5.2 常用内部总线	155	8.1.3 中断优先级	198
5.2.1 STD 总线	155	8.1.4 中断过程	199
5.2.2 PC 系列总线	155	8.2 微型计算机的中断系统	200
5.3 常用外部总线	163	8.2.1 中断的分类和中断类型码	200
5.3.1 IEEE-488 总线	163	8.2.2 中断向量和中断向量表	201
5.3.2 通用串行总线	166	8.2.3 中断响应过程与时序	203
习题	170	8.3 可编程中断控制器 8259A	205
第6章 存储器系统	171	8.3.1 8259A 的内部结构和工作原理	205
6.1 概述	171	8.3.2 引脚信号	206
6.1.1 存储系统的层次结构	171	8.3.3 8259A 的工作过程	207
6.1.2 半导体存储器的分类	172	8.3.4 8259A 的工作方式	207
6.1.3 存储器的基本组成	172	8.3.5 8259A 的编程	209
6.1.4 存储器的主要技术指标	173	8.3.6 8259A 的级联	213
6.2 常用存储器	174	8.3.7 8259A 的应用举例	214
6.2.1 随机存储器	174	习题	215
6.2.2 只读存储器	176	第9章 并行接口及定时/计数技术	217
6.2.3 闪存	178	9.1 并行接口概述	217
6.2.4 常用存储器参数	178	9.2 并行接口芯片 8255A	217
6.3 存储器扩展技术	178	9.2.1 8255A 的内部结构	217
6.4 存储器与 CPU 的连接	180	9.2.2 8255A 的引脚功能	218
6.4.1 存储器与 CPU 连接时问题	180	9.2.3 8255A 的工作方式	219
6.4.2 常用译码电路	181	9.2.4 8255A 的编程及应用	222

9.3 可编程定时/计数器 8253	227	第 12 章 DMA 控制器	263
9.3.1 8253 芯片结构及引脚	227	12.1 概述	263
9.3.2 8253 的读写以及初始化操作	228	12.2 DMA 控制器 8237A	264
9.3.3 8253 的工作方式及时序	229	12.2.1 8237A 的功能结构和外部引脚	264
9.3.4 8253 应用举例	232	12.2.2 8237A 的通道操作过程	267
习题	233	12.2.3 8237A 的内部寄存器	268
第 10 章 串行通信接口	234	12.3 8237A 的编程	271
10.1 串行接口与通信概述	234	习题	275
10.1.1 串行通信方式	234	第 13 章 保护模式	276
10.1.2 数据传送方式	235	13.1 保护模式内存管理	276
10.1.3 信号传输方式	235	13.1.1 分段与分页	276
10.2 串行接口标准	236	13.1.2 逻辑地址和线性地址	278
10.2.1 RS-232-C 总线	236	13.1.3 段选择子与段描述符	278
10.2.2 RS-422 和 RS-485 总线	240	13.1.4 全局描述符表 (GDT) 和局部描述符表 (LDT)	279
10.3 可编程串行接口芯片 8251A	242	13.1.5 段寄存器	280
10.3.1 8251A 的基本性能	242	13.1.6 分页与分段管理	280
10.3.2 8251A 的内部结构	242	13.1.7 页表和页目录表	281
10.3.3 8251A 的引脚功能	244	13.1.8 段到页的映射	281
10.3.4 8251A 的编程	246	13.2 保护模式下任务管理	282
10.3.5 8251A 应用举例	248	13.2.1 任务结构	283
习题	251	13.2.2 任务状态	283
第 11 章 模/数和数/模转换	252	13.2.3 执行任务	284
11.1 数/模(D/A)转换器	252	13.2.4 任务管理数据结构	284
11.1.1 数/模 (D/A) 转换器的 工作原理	252	13.2.5 任务切换	289
11.1.2 数/模 (D/A) 转换器的 主要性能参数	253	13.3 保护模式下的中断和异常	290
11.1.3 数/模 (D/A) 转换器及 接口电路	254	13.3.1 中断和异常的分类	290
11.2 模/数 (A/D) 转换器	256	13.3.2 异常和中断向量	291
11.2.1 A/D 转换器的工作原理	257	13.3.3 中断和异常的处理过程	292
11.2.2 模/数 (A/D) 转换器的主要 性能参数	258	13.3.4 中断描述符表 (IDT)	293
11.2.3 模/数 (A/D) 转换器芯片 及接口电路	258	13.3.5 特权指令	294
习题	262	13.4 输入/输出保护	295
		13.4.1 输入/输出保护	295
		13.4.2 重要标志保护	297
		习题	298
		参考文献	299

第 1 章 计算机基础

本章主要包括电子计算机发展简史、计算机应用领域及发展趋势、微型计算机系统的结构及其主要技术指标、计算机中的数制、布尔代数基础、逻辑电路基础、二进制数的运算及其加法电路、计算机中的编码、浮点数基本概念等内容。

1.1 绪论

1.1.1 电子计算机发展简史

根据制造计算机所使用的元器件不同,电子计算机的发展依次经历了电子管时代、晶体管时代、中小规模集成电路时代、大规模和超大规模集成电路时代、甚大规模和极大规模集成电路时代等几个不同的发展阶段。

(1) 电子管计算机时代(1946~1959年)

电子管是封装在玻璃外壳内的一种电真空器件,用它可以设计出实现反相功能的反相器线路,在此基础上,再实现出计算机使用的全部组合逻辑线路,诸如加法器、译码器等线路,和触发器、寄存器、计数器等各种时序逻辑线路。用电子管线路实现的计算机称为电子管计算机。

世界上第一台电子数字计算机 ENIAC (Electronic Numerical Integrator And Calculator), 1946 年由美国宾夕法尼亚大学研制,字长 12 位,运算速度 5000 次/s,使用 18800 个电子管、1500 个继电器,功耗 150kW,占地 170m²,重达 30 t,造价 100 万美元(见图 1-1)。

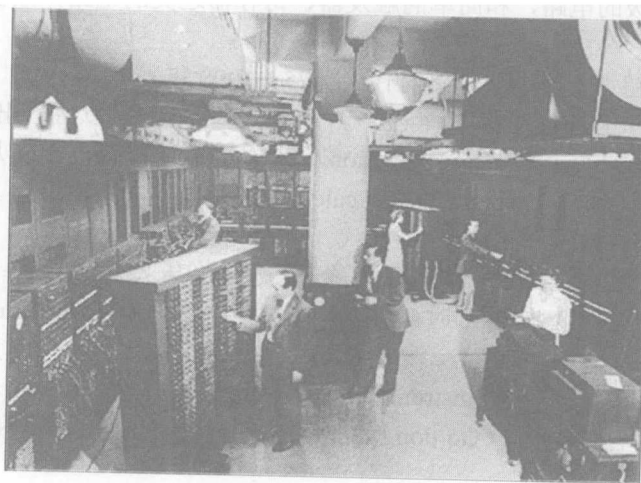


图 1-1 第一台电子数字计算机 ENIAC

ENIAC 计算机存在两个主要缺点,一是存储容量太小,只能存 20 个字长为 10 位的十进制数,二是用线路连接的方法来编排程序,因此每次解题都要依靠人工改接连线,准备时间大大超过实际计算时间。

在 ENIAC 计算机研制的同时,冯·诺依曼(Von Neumann)与莫尔小组合作研制 EDVAC

计算机,采用了存储程序方案,其后开发的计算机都采用这种方式,称为冯·诺依曼计算机。

一般认为冯·诺依曼机具有如下基本特点。

- ① 计算机由运算器、控制器、存储器、输入设备和输出设备五部分组成。
- ② 采用存储程序的方式,程序和数据放在同一个存储器中,指令和数据一样可以送到运算器运算,即由指令组成的程序是可以修改的。
- ③ 数据以二进制码表示。
- ④ 指令由操作码和地址码组成。
- ⑤ 指令在存储器中按执行顺序存放,由指令计数器(即程序计数器 PC)指明要执行的指令所在的单元地址,一般按顺序递增,但可按运算结果或外界条件而改变。
- ⑥ 机器以运算器为中心,输入输出设备与存储器间的数据传送都通过运算器。

60多年来,随着技术的发展和 new 应用领域的开拓,对冯·诺依曼机做了很多改革,使计算机系统结构有了很大发展,如某些机器程序与数据分开存放在不同的存储器中,程序不允许修改,机器不再以运算器为中心,而是以存储器为中心等等,虽然有以上这些突破,但原则变化不大,习惯上仍称之为冯·诺依曼机。

(2) 晶体管计算机时代(1959~1964年)

晶体管,通常指的是晶体三极管,是用半导体材料制作出来、封装在一个金属壳内的带有三个管脚的小器件,1958年进入批量生产阶段。用它设计出实现反相功能的反相器线路,在此基础上,再实现出计算机使用的全部组合逻辑线路,以及触发器、寄存器、计数器等各种时序逻辑线路。用分立的晶体管线路实现的计算机称为晶体管计算机。

(3) 中小规模集成电路时代(1964~1975年)

随着半导体器件生产工艺与技术上的进步,在一片半导体基片上,可以生产出多个晶体管,并用它们形成具有一定处理功能的逻辑器件,这就是集成电路(Integrated Circuit, IC)。此时集成到一个芯片内的晶体管数量还相当有限,实现的还只限于简单的、完成基本处理功能的组合逻辑门一级的电路,和简单的触发器、寄存器之类的电路,故被称为中、小规模集成电路(MSI、SSI)。

(4) 大规模和超大规模集成电路时代(1975~1990年)

半导体器件生产工艺的改进,使得在一片半导体基片上,可以生产出数量更多的晶体管,就形成了大规模集成(Large Scale Integration, LSI)电路,若在一个芯片上的晶体管数量达到更多,就被叫做超大规模(Very Large Scale Integration, VLSI)电路。

(5) 甚大规模和极大规模集成电路时代(1990年至今)

单个芯片内的晶体管数量达到百万个时被叫做甚大规模电路(Ultra Large Scale Integration, ULSI),达到 1×10^8 个时被叫做极大规模电路(Extremely Large Scale Integration, ELSI)。

从1958年德州仪器(Texas Instruments)的工程师基尔比制造出第一块IC到现在为止,集成电路的发展一直遵循摩尔(Gordon Moore)定律,即芯片的集成度和速度每18个月提高一倍。

1.1.2 计算机应用领域及发展趋势

目前计算机已经广泛应用于各行各业,极大地促进了社会的发展。下面简要介绍计算机的几个主要应用领域。

- ① 科学计算 计算机作为计算工具,完成各种复杂的科学计算是它的一个重要应用领域。

② 数据处理 计算机作为数据处理工具，在政府办公，企、事业单位的管理等领域发挥着重要作用。

③ 计算机控制 计算机作为具有高速和灵活的逻辑处理能力的工具，广泛用于工业生产、航天发射等过程的实时控制。

④ 人工智能 计算机作为具有高速和灵活逻辑处理和推理能力的工具，被广泛应用在人工智能领域，完成诸如数学定理证明、自然语言理解、知识表示和挖掘、计算机翻译等工作。

⑤ 计算机网络 随着计算机网络的出现和发展，计算机已经成为在宽广的范围内传播信息和实现人员沟通的重要工具，极大地改变了人类的生活环境和交流方式。

⑥ 计算机辅助设计/制造(CAD/CAM) 对于日益复杂的工程设计项目和生产制造领域，计算机辅助设计/制造已经成为必不可少的重要手段，是提高劳动生产率、设计可靠性和产品质量的重要保障和有效方法。比如说，在集成电路设计的建模、仿真、综合优化、布局布线、时序分析等各个阶段普遍使用了计算机辅助设计技术，否则，要想高效、可靠地设计超大规模、甚大规模和极大规模集成电路，几乎是不可想象的。

⑦ 嵌入式应用 嵌入式计算机是指嵌入到应用系统中的计算机，比如在手机等通信设备，MP4等各种消费类电子产品，以及电视机、冰箱、洗衣机等家电产品中，都有嵌入式计算机。单片机是应用非常广泛的一类嵌入式计算机。所谓单片机即单片微型计算机，是将计算机主机(CPU、内存和I/O接口)集成在一小块硅片上的微型机，如图1-2所示。通常为工业控制和测试而设计，又称微控制器，具有三高优势(集成度高、可靠性高、性价比高)。单片机主要应用于工业检测与控制、计算机外设、智能仪器仪表、通信设备、家用电器等领域。

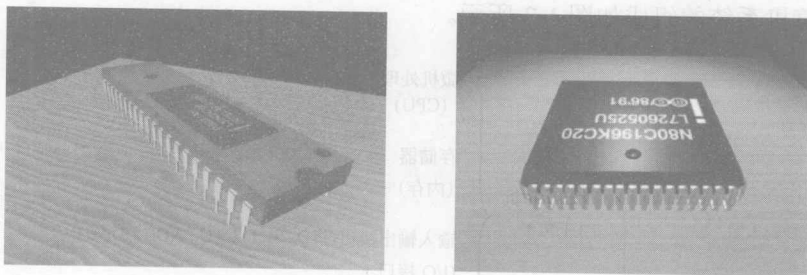


图1-2 单片机芯片

未来计算机的发展呈现出以下几个发展趋势。

① 微型化 便携式、低功耗的计算机系统成为人们追求的目标。

② 高性能 尖端科技领域的信息处理，需要超大容量、高速度的计算机系统。人们对于高性能计算机的追求是无止境的。

③ 智能化 模拟人类大脑思维和交流方式，具有多种处理能力。比如智能机器人等。

④ 系列化、标准化 便于各种计算机硬、软件兼容和升级。

⑤ 网络化 计算机网络普及的必然结果。

⑥ 多机系统 利用多个计算机构成一个更加庞大的系统，使得多个计算机之间可以并行地、协调地工作，从而提高计算机系统的整体性能。如分布式系统、网络计算等。

1.1.3 微型计算机系统的组成及其主要技术指标

微型计算机系统包括硬件和软件两大部分，仅有硬件或者仅有软件微机系统都不能正常

工作，只有两者互相配合才能实现特定功能。

微型计算机系统的硬件部分包括微型计算机和外围设备，其中微型计算机包括微处理器、存储器、总线及输入输出接口，外围设备包括外部设备和辅助设备。微处理器由运算器和控制器构成，运算器是计算机中进行数据加工的部件，其主要功能是执行算术逻辑运算并暂时存放中间结果；控制器是计算机中控制指令执行的部件，向计算机各功能部件提供每一时刻协同运行所需要的控制信号。存储器是计算机中用于存储程序和数据的部件。总线是计算机与外部环境进行交互的通道，包括数据总线、控制总线和地址总线三种类型，输入输出接口是外部设备和计算机通过总线进行通信的接口电路。微型计算机的外部设备包括输入设备、输出设备和辅助存储器等，输入设备是向计算机中送入程序和数据的具有一定独立功能的设备，通过接口和总线与计算机主机连通，用于人机交互联系，如计算机键盘和鼠标等。输出设备是计算机中用于送出计算机内部信息的设备，例如打印机、显示器等。辅助存储器是大容量、永久性的存储器，以弥补内存的不足，如磁带、磁盘、光盘等。

微型计算机系统的软件部分包括系统软件、应用软件和程序设计语言等。程序设计语言是计算机使用者用于进行程序设计的语言，包括机器语言、汇编语言、高级语言等。机器语言又称为二进制代码语言，也就是计算机的指令系统；汇编语言是采用助记符表示的机器语言；高级语言是为了减少自然语言与机器语言之间的语义差距、降低编程难度而开发的程序设计语言，包括 FORTRAN、BASIC、PASCAL、COBOL、C、C++、JAVA 等。

计算机系统软件是为了便于人们使用计算机而设计的一些软件，常用的系统软件包括操作系统、编译器、解释器、汇编器等。应用软件是针对某种特定应用而开发的软件。

通常用高级语言编写的程序首先由编译器或解释器翻译成汇编程序，然后由汇编器翻译成机器语言程序，再由连接器连接构成目标代码，最后才能在计算机硬件上执行。

微型计算机系统的组成如图 1-3 所示。

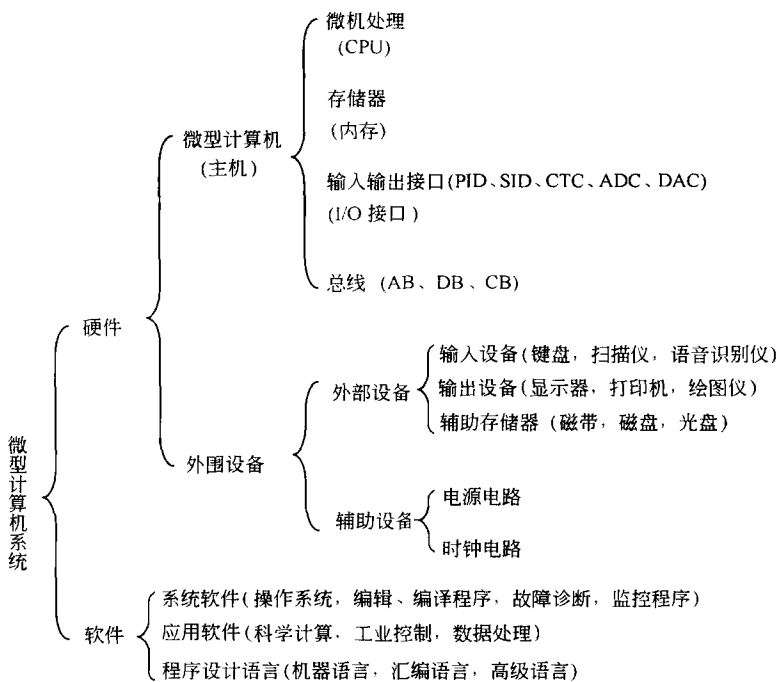


图 1-3 微型计算机系统的组成

微处理器、微型计算机、微型计算机系统这三个概念之间既有联系又有区别，图 1-4 反映了三个概念之间的联系与区别。

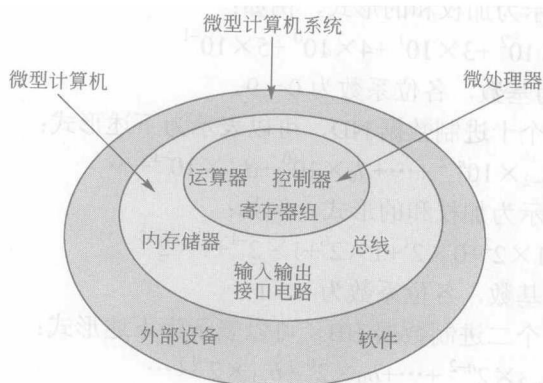


图 1-4 微处理器、微型计算机、微型计算机系统之间的联系与区别

不同的计算机系统的性能是不同的，经常采用一些技术指标来说明计算机系统的性能，常用的计算机技术指标如下。

① 字长 指 CPU 能够并行处理的二进制数据的位数，如：8 位机、16 位机、32 位机和 64 位机。

② 内存容量 指内存中能够存储的二进制信息的数量，单位可以是位、字节或字，通常采用字节作为存储容量的单位。

容量单位之间的换算： $1K=2^{10}=1024$ ， $1M=2^{20}=1KK$ ， $1G=2^{30}=1KM$ ， $1T=2^{40}=1KG$ 。

③ 运算速度 指 CPU 处理速度。

相关参数：时钟频率、主频、每秒运算次数等，如：100MHz、3.2GHz。

④ 内存存取周期 指内存读写速度，如：50ns、70ns、200ns。

1.2 计算机中的数制

1.2.1 数制的基本概念

数制是利用符号来记数的科学方法，计算机科学中经常使用的数制有十进制、二进制、八进制和十六进制。

① 十进制 (decimal system) 用十个数码 0~9 来记数，记数规则是逢十进一。十进制是人们最熟悉的计数体制。

② 二进制 (binary system) 用两个数码 0、1 来记数，记数规则是逢二进一。二进制为计算机中的数据表示形式。

③ 八进制 (octave system) 用八个数码 0~7 来记数，记数规则是逢八进一。

④ 十六进制 (hexadecimal system) 用十六个数码 0~9 和 A~F 来记数，记数规则是逢十六进一。

八进制和十六进制能够简化二进制数的表示。

不同进制数以下标或后缀区别，其中，b 或 B 表示二进制，o 或 O (有时用 q 或 Q) 表示八进制，d 或 D 表示十进制，h 或 H 表示十六进制。十进制数可不带下标或后缀。如：

101、101D 表示十进制的 101，101B 表示二进制的 101，101O 表示八进制的 101，101H 表示十六进制的 101。

十进制数据可以表示为加权的形式，例如：

$$1234.5=1 \times 10^3 + 2 \times 10^2 + 3 \times 10^1 + 4 \times 10^0 + 5 \times 10^{-1}$$

加权展开式以 10 为基数，各位系数为 0~9。

推而广之，任何一个十进制数据 ND，可以表示为下述形式：

$$ND = d_{n-1} \times 10^{n-1} + d_{n-2} \times 10^{n-2} + \dots + d_0 \times 10^0 + d_{-1} \times 10^{-1} + \dots$$

二进制数据可以表示为加权的形式，例如：

$$1101.101B = 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 + 1 \times 2^{-1} + 1 \times 2^{-2}$$

加权展开式以 2 为基数，各位系数为 0、1。

推而广之，任何一个二进制数据 NB，可以表示为下述形式：

$$NB = b_{n-1} \times 2^{n-1} + b_{n-2} \times 2^{n-2} + \dots + b_0 \times 2^0 + b_{-1} \times 2^{-1} + \dots$$

十六进制数据可以表示为加权的形式，例如：

$$DFC.8 = 13 \times 16^2 + 15 \times 16^1 + 12 \times 16^0 + 8 \times 16^{-1}$$

展开式以 16 为基数，各位系数为 0~9，A~F。

推而广之，任何一个十六进制数据 NH，可以表示为下述形式：

$$NH = h_{n-1} \times 16^{n-1} + h_{n-2} \times 16^{n-2} + \dots + h_0 \times 16^0 + h_{-1} \times 16^{-1} + \dots$$

将上述概念进行推广，可以将进位计数的概念推广到任何进制。即，任何一个 r 进制数据 $a_{n-1}a_{n-2}\dots a_1a_0.a_{-1}\dots a_{-m}$ 定义为下列加权：

$$a_{n-1}a_{n-2}\dots a_1a_0.a_{-1}\dots a_{-m} =$$

$$a_{n-1} \times r^{n-1} + a_{n-2} \times r^{n-2} + \dots + a_1 \times r^1 + a_0 \times r^0 + a_{-1} \times r^{-1} + \dots + a_{-m} \times r^{-m}$$

其中 r 称为数制的基， r^{n-1} 、 r^{n-2} 、 \dots 、 r^1 、 r^0 、 r^{-1} 、 \dots 、 r^{-m} 称为各位的权， a_{n-1} 、 a_{n-2} 、 \dots 、 a_1 、 a_0 、 a_{-1} 、 \dots 、 a_{-m} 称为各位的系数。

1.2.2 数制之间的转换

(1) 二进制、十六进制数转换成十进制数

方法：各位的系数乘以各位的权，然后全部加起来。

$$\text{举例：} 1011.1010B = 1 \times 2^3 + 1 \times 2^1 + 1 \times 2^0 + 1 \times 2^{-1} + 1 \times 2^{-3} = 11.625$$

$$DFC.8H = 13 \times 16^2 + 15 \times 16^1 + 12 \times 16^0 + 8 \times 16^{-1} = 3580.5$$

(2) 二进制与十六进制数之间的转换

方法：因为 $2^4=16$ ，所以四位二进制数对应一位十六进制数。

$$\text{举例：} 3AF.2H = 0011\ 1010\ 1111.0010 = 1110101111.001B$$

$$1111101.11B = 0111\ 1101.1100 = 7D.CH$$

(3) 十进制数转换成二进制、十六进制数

方法：将整数部分和小数部分分别转换，然后再合并起来。

① 整数转换法 “除基取余”，即十进制整数不断除以转换进制基数，直至商为 0。每除一次取一个余数，从低位排向高位，就形成了转换后的数据。

【例 1-1】 将十进制数 39 转换成二进制数。

解：39 除以 2，商为 19，余数为 1 (b_0)；

19 除以 2，商为 9，余数为 1 (b_1)；

9 除以 2，商为 4，余数为 1 (b_2)；

4 除以 2, 商为 2, 余数为 0 (b_3);

2 除以 2, 商为 1, 余数为 0 (b_4);

1 除以 2, 商为 0, 余数为 1 (b_5);

所以十进制数 39 转换成二进制数为 100111B。

【例 1-2】 将十进制数 208 转换成十六进制数。

解: 208 除以 16, 商为 13, 余数为 0;

13 除以 16, 商为 0, 余数为 13=D;

所以十进制数 208 转换成十六进制数为 D0H。

② 小数转换法 “乘基取整”, 即用转换进制的基数乘以小数部分, 直至小数为 0 或达到转换精度要求的位数。每乘一次取一次整数, 从最高位排到最低位, 就得到了转换后的数据。

【例 1-3】 将十进制数 0.625 转换成二进制数。

解: $0.625 \times 2=1.25$, 乘积中小数为 0.25, 整数为 1(b_{-1});

$0.25 \times 2=0.50$, 乘积中小数为 0.50, 整数为 0(b_{-2});

$0.50 \times 2=1.00$, 乘积中小数为 0.00, 整数为 1(b_{-3})

所以十进制数 0.625 转换成二进制数为 0.101B。

【例 1-4】 将十进制数 0.625 转换成十六进制数。

解: $0.625 \times 16=10.0$, 乘积中小数为 0.0, 整数为 10 (十六进制符号为 A);

所以十进制数 0.625 转换成十六进制数为 0.AH。

【例 1-5】 将十进制数 208.625 转换成十六进制数。

解: 在例 1.2 和例 1.4 中已经分别得到了 208.625 的整数部分和小数部分转换后的十六进制数, 将两者合并起来, 就可以得到十进制数 208.625 转换成十六进制数为 D0.AH。

1.3 布尔代数基础

布尔代数又称为开关代数或逻辑代数, 是在 1847 年由英国数学家乔治·布尔(George Boole)首先创立的, 布尔代数研究逻辑变量之间的相互关系和变化规律, 它是分析和设计数字逻辑电路的理论基础和基本工具。

布尔代数具有下述特点: 变量只有两种可能的取值, 0 或 1; 只有 3 种基本的逻辑运算, “与”、“或”、“非”。

1.3.1 基本逻辑运算

最基本的逻辑操作: “与”(逻辑乘, 符号 $A \cdot B$ 或 AB 或 $A \times B$)、“或”(逻辑加, 符号 $A+B$)、“非”(逻辑非或逻辑反, 符号 \bar{A} , 或 A')。

“与”操作的定义: 当 $A=1$ 且 $B=1$ 时, $AB=1$, 否则, $AB=0$ 。

“或”操作的定义: 当 $A=1$ 或 $B=1$ 时, $A+B=1$, 否则, $A+B=0$ 。

“非”操作的定义: 若 $A=1$ 则 $\bar{A}=0$, 若 $A=0$ 则 $\bar{A}=1$ 。

类似地可以定义多个变量的“与”操作和“或”操作。

多位二进制数的逻辑运算定义为各对应位分别进行相应的逻辑运算。

其他常用的逻辑操作有: 与非、或非、与或非、异或、同或等。

1.3.2 基本运算规律

- ① 0-1 律 $A0=0 \quad A+1=1$
- ② 自等律 $A1=A \quad A+0=A$
- ③ 重叠律 $AA=A \quad A+A=A$
- ④ 互补律 $A\bar{A}=0 \quad A+\bar{A}=1$
- ⑤ 交换律 $AB=BA \quad A+B=B+A$
- ⑥ 结合律 $(AB)C=A(BC) \quad (A+B)+C=A+(B+C)$
- ⑦ 分配律 $A(B+C)=AB+AC \quad A+BC=(A+B)(A+C)$
- ⑧ 吸收律 $A(A+B)=A \quad A+AB=A \quad A+\bar{A}B=A+B \quad A(\bar{A}+B)=AB$
- ⑨ 反演律 (De. Morgan 定理) $\overline{AB}=\bar{A}+\bar{B} \quad \overline{A+B}=\bar{A}\bar{B}$
- ⑩ 双重否定律 (还原律) $\overline{\bar{A}}=A$

1.3.3 逻辑函数的表示方法

逻辑函数可以选用布尔代数式表示、真值表表示或卡诺图表示。

所谓布尔代数式表示方法，是将函数表达为由自变量和基本逻辑运算符所构成的一个表达式。

所谓真值表表示方法，是指将自变量的所有取值组合一一列出，同时列出每个自变量取值组合所对应的函数值。

用卡诺图表示逻辑函数：由全部变量或其反变量形成的逻辑乘积项称为最小项。对 n 个变量，共有 2^n 个最小项。卡诺图是一种直观的平面方块图，它将平面划分为 2^n 个小格，用来表示 n 个变量的全部 2^n 个最小项。卡诺图的左边和上边书写的规则必须是这样的：两相邻小格之间只能有一个变量是相反的，而其余的变量都是相同的。为了简单起见，往往把周边变量的原码用 1 表示，反码用 0 表示。在小格中填上对应最小项的取值。由于任何一个函数都可以展开为若干个最小项之和，因此，可用卡诺图表示任意一个逻辑函数。对于有偶数个输入变量的函数，把卡诺图作成正方形，即水平方向和垂直方向的变量数相等；若输入变量为奇数，则水平方向和垂直方向的变量数之差不应大于 1。

【例 1-6】 $X=AB$ 是布尔代数式表示的与函数，其对应的真值表见表 1-1。

【例 1-7】 $X=\overline{AB}$ 是布尔代数式表示的与非函数，其对应的真值表见表 1-2。

表 1-1 $X=AB$ 的真值表

A	B	X
0	0	0
0	1	0
1	0	0
1	1	1

表 1-2 $X=\overline{AB}$ 的真值表

A	B	X
0	0	1
0	1	1
1	0	1
1	1	0

1.3.4 真值表与逻辑表达式之间的相互转换

(1) 已知逻辑表达式求真值表

将自变量的所有取值组合一一列出，同时根据逻辑表达式计算出每个自变量取值组合所对应的函数值，并将它与自变量取值组合列在同一行中即可。

(2) 已知真值表求逻辑表达式

① 找出真值表中函数值为 1 的那些行。

② 写出函数值为 1 的那些行所对应的最小项。在真值表的某一行中,若某自变量取值为 1,则取其原变量,若某自变量取值为 0,则取其反变量,将得到的所有原变量或反变量相与即是该行所对应的最小项。

③ 将上面得到所有最小项相或,就得到了该真值表所表示函数的逻辑表达式。

【例 1-8】 设某函数的真值表如表 1-1 所示,求该函数的布尔代数表达式。

解:根据上述方法可得,该函数的逻辑表达式为 $X=AB$ 。

【例 1-9】 设某函数的真值表如表 1-2 所示,求该函数的布尔代数表达式。

解:根据上述方法可得,该函数的逻辑表达式为 $X=\overline{AB}+\overline{A}B+A\overline{B}=\overline{AB}$ 。

1.3.5 逻辑函数的化简

将一个逻辑函数变成一个形式更简单,与之等效的逻辑函数,称为化简。由于每个逻辑表达式是和—个电路相对应的,因此表达式的化简就能减少实现它的电路所用的元器件,并减少电路的延时。

常用的两种化简方法为:代数化简法和卡诺图化简法。

① 代数化简法 利用布尔代数的基本公式和规则进行化简的方法。

【例 1-10】 $\overline{A}B+\overline{A}B+AB=\overline{A}(\overline{B}+B)+AB=\overline{A}+AB=\overline{A}+B$

② 卡诺图化简法 由于相邻两小格所表示的最小项只有一个变量是互为相反的(位于卡诺图同一行或同一列两端的小格也应理解为相邻),因此,相邻两小格合并后的布尔表达式显然会减少一个变量,这样就化简了逻辑表达式。任意两个相邻小格构成的块叫 1 维块,两个相邻 1 维块合并后又可以减少一个变量,两个相邻 1 维块合并后构成的块称为 2 维块。两相邻 2 维块合并(称为 3 维块)又可以减少一个变量,以此类推。同一个小块可以重复使用多次。若在某些实际问题中,某些最小项不会出现或不被利用,其值可以不加指定,在进行卡诺图化简时,可以根据情况指定为 0 或者 1,以简化逻辑表达式。如图 1-5 所示。

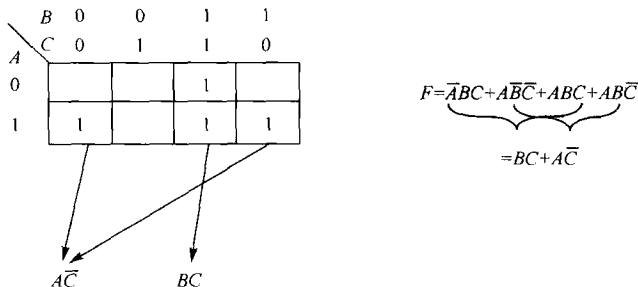


图 1-5 卡诺图化简法举例

1.4 逻辑电路基础

逻辑电路是实现输入信号与输出信号之间逻辑关系的电路,计算机对于信息数据的处理都是由逻辑电路实现的,因此逻辑电路是计算机的硬件基础。

常用的基本逻辑门电路有:与门、或门、非门、与非门、或非门、异或门、同或门、缓冲器等,这些基本门电路是构成逻辑电路的基本成分,利用它们可以搭建多种多样的复杂的逻辑电路。基本逻辑门电路符号及表达式如图 1-6 所示。