

国家信息产业部电子人才交流中心参与规划
“十一五”高等院校应用型规划教材

微机原理

W e i J i Y u a n L i

与接口技术

Y u J i e K o u J i S h u

立足系统，面向应用
先进性与实用性并重

科海策划

厉荣卫 主编

陈鉴富 黄海军 副主编



科学出版社

微机原理与接口技术

主编 厉荣卫

副主编 陈鉴富 黄海军

科学出版社

内 容 简 介

微机原理与接口技术是高等院校理工科类专业的一门重要的计算机技术基础课程。本书以 16 位微机和 32 位微机为基础,系统地阐述了现代微机的基本结构和接口及总线的基本原理与应用。内容主要包括微机基本结构及其原理、微处理器的基本结构和发展、I/O 端口地址译码技术、总线及接口、DMA 技术、中断技术、定时/计数技术、并行接口、串行接口、人机交互接口、A/D 与 D/A 转换接口及 PC 机高级编程技术等。书中既包含了微机原理部分与典型的接口技术,也介绍了新型接口与总线技术;既讲述了接口的硬件部分,也强调了接口的软件编程。本书涉及到很多新的技术,如 PCI 总线、IEEE1394、USB、下一代新型总线、815EP 芯片组、保护模式下的中断技术、Windows 9x 及 Windows 2000 下的设备驱动程序的编写等,反映了现代微机技术发展的最新水平和趋势,并体现出微机接口技术中硬件设计和软件驱动的统一。

本书内容充实、概念清晰、重点突出、实例丰富,先进性与实用性并重,可作为高等院校理工科相关专业的微机原理与接口技术课程的专用教材。

图书在版编目(CIP)数据

微机原理与接口技术/厉荣卫主编. —北京:

科学出版社, 2006

ISBN 7-03-017717-7

I. 微… II. 厉… III. ①微型计算机—理论

—高等学校: 技术学校—教材②微型计算机—

接口—高等学校: 技术学校—教材 IV.TP36

中国版本图书馆 CIP 数据核字(2006)第 083165 号

责任编辑: 何 武 王金柱 / 责任校对: 张 楠

责任印刷: 科 海 / 封面设计: 林 陶

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>

北京科普瑞印刷有限责任公司印刷

科学出版社发行 各地新华书店经销

*

2006 年 8 月第一版

开本: 16 开

2006 年 8 月第一次印刷

印张: 22.25

印数: 1-4000

字数: 541 千字

定价: 38.00 元

(如有印装质量问题, 我社负责调换)

前 言

微型计算机原理与接口技术是理工科学生学习和掌握微机技术的重要课程之一，课程的任务是使学生从理论和实践上掌握微型计算机的基本组成、工作原理、接口技术及硬件连接，建立微型计算机系统的整机概念，使学生具有微型计算机系统软硬件开发和应用的初步能力。

随着微型计算机价格的不断下降，微型计算机迅速在各个领域得到广泛的应用，因而掌握微型计算机系统知识及接口技术就显得特别重要。同时，微处理器、微机接口，以及微机操作平台方面，新的技术也在不断涌现，因此无论是工程技术人员还是大专院校的学生，都应该对微机新技术有所了解。

我们在微机原理及接口技术的教学当中还感觉到，学生不仅应该掌握原理，更要学会应用，要做到理论与实际相结合。另外，接口技术的学习不仅仅是硬件上的设计与分析，还应该包括接口软件的编写。随着微机操作系统平台由 DOS 向 Windows 的转变，掌握新的操作系统平台的接口软件编写也是微机接口设计者应该具备的能力。

目前，我国微机原理与接口技术方面的教材已经不少，它们对于人才的培养发挥了积极的作用。但近几年来，由于微机系统的迅速发展，大部分教材的内容已不能满足人才培养的需要。也就是说，用原有的教材培养出来的毕业生已不能适应用人单位的需求。鉴于此，不少同仁在教材内容的改革方面进行了不少尝试：有的在原有教材的基础上添加一些新的知识介绍，但这类教材总体上大部分知识仍较陈旧，新知识与老知识之间缺乏连贯性；有的是全新编写，这类教材知识很新，但由于完全抛弃了原来此类教材的写作结构，而且过多地涉及处理器内部而削弱了接口方面的介绍，所以影响了教师对此类教材的选用。我们认为，微机原理与接口技术的课程教材，既要更新内容，也要注意重点还是应放在接口设计与应用上。为此，根据我们多年的教学实践，在教材内容的改革上也进行了大胆的尝试，编写了本书。本书具有以下一些特点。

(1) 立足于系统，面向应用。本书首先从系统的角度，介绍了微机的基本结构和发展的特点，以及现代微机中占重要地位的总线的概念，让读者一开始就对微机系统有个全面的了解，之后的各个章节描述微机各种接口的具体内容。

(2) 内容新。全书以现代微机的主流 CPU 为切入点，以微机接口技术为重点，较全面地介绍了现代微机的原理与接口技术。其中包含了 PCI 总线、USB 总线、IEEE1394、Pentium 的中断、APIC、AC'97 等新的技术。

(3) 实用性强。本书和其他同类教材的最大不同点在于接口软件的介绍上，我们将接口软件的编写从传统的实模式下的 MS-DOS 平台上升到保护模式下的 Windows 9x/2000 的平台上，对于当前微机的主流应用领域具有很强的针对性和实用性。

(4) 适用面广。全书给出了不少应用的例子和各种接口的电气和物理特性，甚至还给

出了一些主要扩展卡的尺寸，不仅可以作为大专院校学生的教材，对工程技术人员也很有参考价值。

本书内容丰富，实际教学中，可以根据教学目的和学时数选择有关章节讲解。对于学过“计算机组成原理”的读者，可以省去有关微处理器、总线的概念，中断的概念，DMA等概念方面的内容。对于以教授PC机接口软件技术为主的课程，可以加讲汇编语言部分，并可在讲完中断后，先讲述第10章的内容，同时可以去掉8251和8255芯片的讲授内容，并可适当减少总线一章的内容。根据不同的专业，授课时间可以在48学时到64学时之间。对于计算机专业的本科生，需要先修“数字逻辑电路”、“计算机组成原理”、“8088汇编语言”和“C语言程序设计”；电类非计算机专业的本科生至少要先修“计算机硬件基础”，如果先修过“计算机组成原理”、“8088汇编语言”和“C语言程序设计”则更好，否则，至少需要用12~20学时左右的时间讲授汇编语言部分。

本书由厉荣卫担任主编，陈鉴富、黄海军任副主编，参加本书编写的还有姜捷、吕萍老师，徐敏、耿宗琳等同学参加了本书的编排与绘图工作。在本书的编写过程中得到了江苏技术师范学院教学改革项目的资助，在此表示感谢。

由于微机发展迅速，再加上作者水平有限，难免会有不足之处，殷切希望得到广大同仁和读者的批评指正。

作者联系方式：0519-6999635 0519-8598128

E-mail:liwei@jstu.ed

编 者

2006年5月

目 录

第 1 章 微型计算机概述	1
1.1 微型计算机发展概述	1
1.2 微机的组成	2
1.2.1 微机系统的组成	2
1.2.2 微机的硬件结构	3
1.3 微处理器结构	7
1.3.1 8086/8088 微处理器	7
1.3.2 Pentium 微处理器	20
1.3.3 Pentium II 微处理器	25
1.3.4 Pentium III 微处理器	27
本章小结	28
习题与思考	28
第 2 章 输入/输出与接口技术	30
2.1 I/O 接口概述	30
2.1.1 接口	30
2.1.2 外部设备与 I/O 接口	31
2.1.3 I/O 接口的基本功能	33
2.1.4 I/O 接口的组成	35
2.2 输入/输出数据的传输控制方式	36
2.2.1 程序控制传输方式	37
2.2.2 中断传输方式	40
2.2.3 直接存储器存取方式 (DMA 方式)	42
2.2.4 4 种 I/O 方式的比较	43
2.2.5 专用 I/O 处理机方式	43
2.3 I/O 端口的编址方法	44
2.3.1 I/O 统一编址	44
2.3.2 I/O 独立编址	44
2.3.3 I/O 端口地址译码	44
2.4 接口设计与分析基本方法	48
2.4.1 接口硬件设计方法简介	48
2.4.2 接口软件设计方法简介	51
本章小结	51
习题与思考	51
第 3 章 微机总线与接口标准	52
3.1 总线技术概述	52
3.1.1 总线的基本概述和总线的标准	52
3.1.2 总线的分类	53
3.1.3 总线的主要性能参数	53
3.1.4 总线标准的特性	54
3.1.5 总线组成	54
3.1.6 总线操作和总线传送控制	55
3.2 系统总线	57
3.2.1 ISA 总线	57
3.2.2 其他系统总线	59
3.3 PCI 总线	59
3.3.1 PCI 总线的特点	59
3.3.2 PCI 信号定义	60
3.3.3 PCI 插槽和 PCI 扩展卡	62
3.3.4 PCI 总线命令	65
3.3.5 PCI 总线数据传输过程	65
3.3.6 PCI 总线仲裁	66
3.3.7 PCI 总线配置	66
3.3.8 PCI BIOS	68
3.4 PCI-X 总线	68
3.5 PCI Express 总线	69
3.5.1 PCI Express 总线概述	69
3.5.2 PCI Express 总线的技术特点	69
3.5.3 PCI Express 的数据传输过程	70
3.6 通用串行总线 USB	71
3.6.1 USB 系统组成	71
3.6.2 USB 系统的接口信号和 电气特性	72
3.6.3 USB 数据流类型和传输类型	72
3.6.4 USB 交换的包格式	73
3.6.5 USB 设备状态和总线枚举	73
3.6.6 USB 外围芯片及应用	74

3.7 高性能串行总线标准 IEEE1394.....75	5.2.2 保护方式下的中断.....117
3.7.1 IEEE1394 的主要性能特点.....75	5.3 8259A 中断控制器.....119
3.7.2 IEEE1394 拓扑结构.....76	5.3.1 8259A 的引脚信号和内部结构.....120
3.7.3 IEEE1394 地址分配.....77	5.3.2 8259A 的初始化编程.....129
3.7.4 IEEE1394 的主要技术规范.....77	5.3.3 8259A 在微机中的应用.....131
3.7.5 IEEE1394 与 USB 的比较.....77	5.4 实模式下中断处理程序的设计.....135
3.8 其他总线和接口.....78	5.5 PCI 中断.....137
3.8.1 SCSI 接口标准.....78	5.5.1 微机中对 PCI 中断的操作.....137
3.8.2 AGP 接口.....78	5.5.2 PCI 中断响应周期.....138
3.9 PCMCIA 总线及扩展卡.....79	5.5.3 PCI 中断的共享.....138
3.9.1 16 位 PC 卡.....79	5.6 高级可编程中断控制子系统 APIC.....139
3.9.2 卡总线 (CardBus).....79	5.6.1 APIC 子系统的组成.....139
本章小结.....80	5.6.2 82801 BA 的 I/O APIC 模块中 的寄存器.....141
习题与思考.....80	5.6.3 APIC 总线功能.....143
第 4 章 DMA 技术.....81	5.6.4 APIC 子系统的工作过程.....147
4.1 DMA 的概念.....81	5.7 串行中断.....148
4.1.1 DMA 的组成和作用.....81	本章小结.....150
4.1.2 DMA 传送的工作过程.....83	习题与思考.....150
4.1.3 DMA 传送的方式.....84	第 6 章 定时与计数技术.....151
4.2 DMA 控制器.....86	6.1 概述.....151
4.2.1 DMAC 的工作状态.....86	6.1.1 定时与计数.....151
4.2.2 82C37A 的内部结构和 引脚功能.....87	6.1.2 定时方法.....152
4.2.3 82C37A 的内部寄存器.....91	6.2 可编程定时器/计数器 8254.....153
4.2.4 82C37A 的端口地址.....96	6.2.1 8254 的基本功能.....154
4.2.5 操作命令.....97	6.2.2 8254 的内部结构和外部引脚.....154
4.2.6 DMAC 的工作时序.....98	6.2.3 8254 计数器的工作方式.....158
4.3 DMAC 编程和应用举例.....100	6.2.4 8254 的控制字.....164
4.3.1 82C37A 的初始化.....100	6.2.5 8254 的编程.....165
4.3.2 82C37A 应用编程举例.....103	6.3 8254 在微机中的应用.....167
本章小结.....106	6.3.1 计数器 0——系统计时器.....168
习题与思考.....107	6.3.2 计数器 1——动态存储器定时 刷新控制.....169
第 5 章 微型计算机的中断系统.....108	6.3.3 计数器 2——扬声器音频 发生器.....169
5.1 基本概念.....108	6.4 实时钟电路及其应用.....170
5.1.1 中断与异常.....108	6.4.1 MC146818 的外部特性和 工作原理.....170
5.1.2 中断过程.....112	6.4.2 CMOS-RAM 中的实时钟信息.....172
5.1.3 外部中断源的管理.....114	6.4.3 实时钟的状态寄存器.....173
5.2 实地址方式和保护方式下的中断.....116	
5.2.1 实地址方式下的中断.....116	

6.4.4 RT/CMOS RAM 操作	174	8.4.3 8251A 的编程	230
本章小结	175	8.4.4 8251A 应用举例	234
习题与思考	175	8.5 异步通信适配器	236
第 7 章 并行接口	176	8.5.1 INS 8250 外部引脚	237
7.1 并行接口概述	176	8.5.2 UART 内部结构	238
7.1.1 并行接口的特点	176	8.5.3 采用 UART 的异步通信适配器 硬件逻辑	238
7.1.2 可编程并行接口	177	8.5.4 INS8250 的编程	239
7.2 可编程并行接口芯片 82C55	178	8.5.5 串行通信程序设计	245
7.2.1 82C55 的基本功能	178	本章小结	253
7.2.2 82C55 的内部结构和外部引脚	179	习题与思考	254
7.2.3 82C55 的工作方式	181	第 9 章 人机交互接口与 A/D,	
7.2.4 82C55 的控制字与初始化编程	186	D/A 转换器接口	255
7.2.5 82C55 应用举例	187	9.1 PC 机键盘接口	255
7.3 并行接口标准	196	9.1.1 PC 机键盘接口原理	255
7.3.1 打印机并行接口	196	9.1.2 键盘与主机之间的通信方式	257
7.3.2 IEEE1284 标准	198	9.2 鼠标器接口	258
7.4 打印机接口编程	201	9.2.1 鼠标器的分类与工作原理	258
7.4.1 打印机适配器端口直接编程	201	9.2.2 鼠标器的接口	259
7.4.2 打印机功能调用	203	9.2.3 鼠标器的驱动程序	260
本章小结	204	9.3 显示器与显示卡	261
习题与思考	204	9.3.1 CRT 显示器	261
第 8 章 串行接口	205	9.3.2 CRT 显示器的主要性能参数	262
8.1 串行通信基本概念	205	9.3.3 显示卡	263
8.1.1 串行通信	205	9.3.4 液晶显示器 (LCD)	266
8.1.2 数据传送方式	206	9.4 网络接口	270
8.1.3 波特率和收/发时钟	207	9.4.1 调制解调器	270
8.1.4 信号调制解调	208	9.4.2 ISDN	272
8.1.5 差错控制	209	9.4.3 宽带接入技术	272
8.1.6 串行通信基本方式	210	9.4.4 82801 BA 中的 LAN 控制器	275
8.2 串行通信协议	211	9.5 AC'97 音频和调制解调器控制器	277
8.2.1 异步串行通信协议	211	9.5.1 AC'97 概述	277
8.2.2 同步串行通信协议	212	9.5.2 AC'97 的数字链接 AC_link	278
8.3 串行接口标准	215	9.5.3 现代微机系统实现 AC'97 标准的方法	282
8.3.1 RS-232C 标准	215	9.6 D/A 和 A/D 转换器	282
8.3.2 RS-449 与 RS-423/422/485 接口	221	9.6.1 D/A 转换器	282
8.4 可编程串行接口芯片 8251A	226	9.6.2 A/D 转换器	289
8.4.1 基本性能	226	本章小结	297
8.4.2 内部结构与外部引脚功能	226		

习题与思考	297	10.2.3 WDM 驱动程序的层次模型	314
第 10 章 微型计算机的高级		10.3 WDM 驱动程序的基本结构	315
编程技术	298	10.4 驱动程序设计的硬件基础	327
10.1 保护模式下的内存管理	298	10.5 WDM 开发驱动程序的过程	332
10.1.1 系统地址寄存器	298	本章小结	337
10.1.2 段式存储管理	299	习题与思考	337
10.1.3 Pentium II/III 的虚拟页式		附录一 调试程序 DEBUG	338
存储管理	304	附录二 8086/8088 指令系统	
10.2 WDM 设备驱动程序简介	308	一览表	342
10.2.1 WDM 驱动程序概述	308	参考文献	348
10.2.2 WDM 驱动程序的种类	312		

第 1 章 微型计算机概述

学习目标

1. 了解微型计算机的发展方向
2. 掌握基本的微型计算机组成与结构
3. 掌握基本的 8086/8088 微处理器结构
4. 了解 Pentium 系列微处理器
5. 了解最新的微型计算机与微处理器的发展情况

1.1 微型计算机发展概述

微型计算机简称微计算机或微机。人们通常将微型计算机的发展划分为以下几个阶段。

第一阶段（1971~1973 年）是 4 位或 8 位低档微处理器和微型计算机时代，通常称之为第一代，其典型产品是 Intel 4004 和 Intel 8008 微处理器以及分别由它们组成的 MCS-4 和 MCS-8 微型计算机。它们的基本特点是采用 PMOS 工艺，集成度低（1200~2000 个晶体管/片），系统结构和指令系统均比较简单，主要采用机器语言或简单的汇编语言，基本指令的执行时间约为 10~20 μ s，适用于家用电器和简单的控制场合。

第二阶段（1974~1978 年）是 8 位中高档微处理器和微型计算机时代，通常称之为第二代，其典型产品是 Intel 公司的 8080/8085 等微处理器。它们的特点是采用 NMOS 工艺，较之第一代产品，集成度提高 4 倍左右（5000~9000 个晶体管/片），运算速度提高 10~15 倍（基本指令的执行时间约 1~2 μ s），指令系统比较完善，已具有典型的计算机体系结构以及中断、DMA 等控制功能，软件除采用汇编语言外，还配有 BASIC、FORTRAN、PL/1 等高级语言及其相应的解释程序和编译程序，并在后期配上了操作系统，如 CP/M 就是当时世界上流行的一种微机操作系统。

第三阶段（1978~1984 年）是 16 位微处理器和微型计算机时代，通常称之为第三代，其典型产品是 Intel 公司的 8086/8088 及 80286 等微处理器。它们的特点是采用 HMOS 工艺，集成度（20000~70000 个晶体管/片）和运算速度（基本指令的执行时间约为 0.5 μ s）比第二代产品提高了一个数量级，指令系统更加丰富、完善，采用多级中断系统、多种寻址方式、段式存储器结构，以及硬件乘除部件，并配置了强有力的软件系统。这一时期的微机产品有 IBM PC/XT（8088）、IBM PC/AT（80286）及其兼容机等。

第四阶段（1985~1991 年）是 32 位微处理器和微型计算机时代，通常称之为第四代，其典型产品是 Intel 公司的 80386/80486 等微处理器，以及相应的 IBM PC 兼容机，如 386、

486 机等。它们的特点是采用 HMOS 或 CMOS 工艺，集成度高达 100 万个晶体管/片以上，具有 32 位数据总线和 32 条地址总线，工作主频达 25MHz 以上，基本指令的工作速度达到或超过 25MIPS (Million of Instructions Per Second, 百万指令/秒)。微型计算机的功能已经达到甚至超过当时的超级小型机，完全可以胜任多用户、多任务的作业。

第五阶段 (1992 年以后) 是奔腾系列微处理器和奔腾系列微型计算机时代，通常称之为第五代，其典型产品是 Intel 公司的 Pentium，内部采用超标量指令流水线结构，并具有相互独立的指令和数据高速缓存，工作频率越来越高，基本指令的工作速度也越来越高。

随着 Pentium MMX、Pentium Pro、Pentium II、Pentium III 以及 Pentium IV 微处理器的出现，使微机的发展在网络化、多媒体化和智能化等方面跨上了更高的台阶，而且这一阶段以 Pentium 为前缀命名的处理器不断改进，性能不断完善。可以说这一阶段是微型计算机发展最快，新技术最多、最先进，影响最大的阶段。

应该指出，尽管 Pentium~Pentium IV 处理器的内部与工作寄存器直接相连的是 32 位数据总线宽度，但外部数据总线却是 64 位的。与第五阶段同步并行发展的还有 64 位微处理器和微型计算机，如 Intel 公司的 Itanium、Itanium II 等处理器。

1.2 微机的组成

1.2.1 微机系统的组成

一台完整的微型计算机必须由硬件和软件两大部分组成，其中硬件是基础，软件是灵魂，两者缺一不可。通常，我们把这种包含硬件和软件的“完整微型计算机”称之为微型计算机系统，如图 1-1 所示。

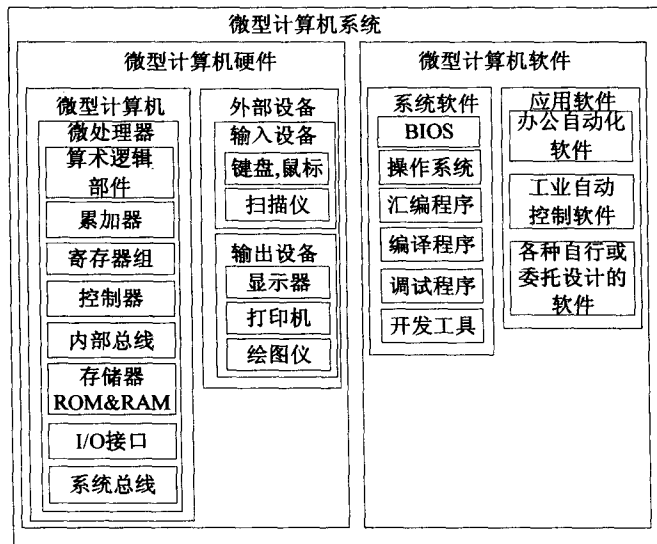


图 1-1 微型计算机系统的基本组成

1.2.2 微机的硬件结构

1. 微型计算机的基本结构

微型计算机与通用计算机没有本质区别，它同样应用冯·诺依曼计算机的基本原理，也需要存储器、运算器、控制器以及输入输出接口等部件，不同的只是微型计算机采用了大规模、超大规模集成电路技术，各组成部件均集成在半导体集成电路芯片上，并且都有相对独立和相对完整的功能，从而决定了微型计算机在组成上有它自己的特点。

如图 1-2 所示，微型计算机通常由微处理器、存储器、输入输出接口电路、总线以及其他支持逻辑电路所组成。

存储器是用来存放数据和程序的部件。为了满足存储容量和存取速度的需要，存储器一般采用分级存储方式，即用速度较高的半导体存储器作为内存储器，而用容量较大、存取速度相对较低的磁表面存储器或光盘存储器作为外存储器。

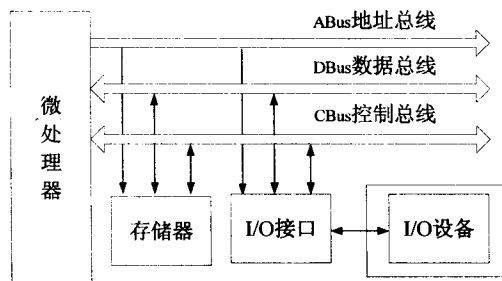


图 1-2 微型计算机的基本组成

输入输出接口即 I/O 接口（Input/Output Interface），它是微型计算机与外部设备之间交换信息的通路，不同的外部设备与微型计算机相连都需要配备不同的接口。

总线是连接上述各部件的公共线路。按照传送信号的性质，总线可分为数据总线、地址总线和控制总线，它们分别用于传送数据、地址和控制信号；而按照总线连接对象的不同，总线又可分为系统总线、局部总线和外部总线，其中，系统总线用于微机内各部件之间的连接，局部总线用于微机内 CPU 与各外围芯片之间的连接，而外部总线则用于微机与外部设备之间的连接。

总之，微处理器、存储器、输入输出接口以及总线是构成微型计算机的主要部件。下面介绍这几个主要组成部分的基本结构和工作原理。

2. 微处理器

微处理器主要包括运算器和控制器两大部件，有些系统还会包括数学协处理器。图 1-3 给出典型微处理器的基本结构。下面分别介绍这 3 个部件的基本组成和工作原理。

(1) 运算器

如图 1-3 所示，运算器由算术逻辑部件（Arithmetical and Logical Unit, ALU）、寄存器组以及一些控制数据传送的电路组成，其中，算术逻辑部件 ALU 是运算器的主要部件，加、减、乘、除等基本算术运算都在这里进行。此外，该部件还具有移位功能，并可以执行与、或、非等逻辑运算和求补操作。

(2) 控制器

正如前面所述，微型计算机是根据冯·诺依曼关于程序存储和程序控制的基本原理设计出来的。因此，微型计算机的整个工作过程是周而复始地取指令、分析指令、执行指令的过程。

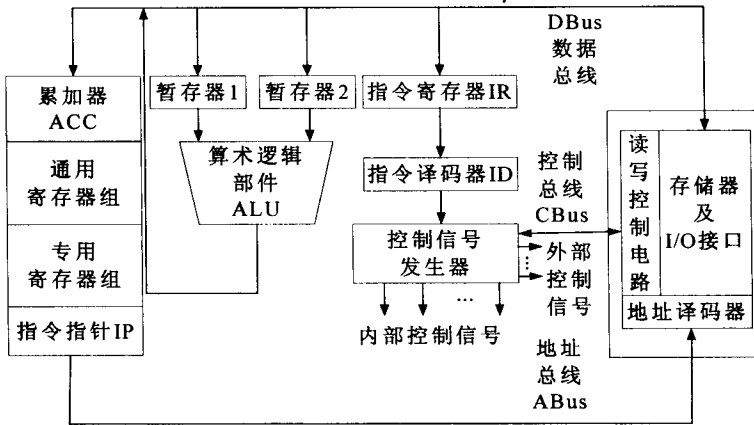


图 1-3 简化的微处理器基本结构

为了实现这一过程，控制器是一个非常关键的部件，它根据预先存放在存储器中的程序对计算机进行控制。每当取出一条指令，就对该指令进行分析，然后根据指令的要求向各部件发出控制信号（例如进行加法或减法运算），并接收执行部件向控制器发回有关指令执行情况的反馈信息。控制器的这种工作过程实质上就是取指令、分析指令、执行指令再取下一条指令，周而复始地使计算机工作的过程。

控制器的组成与指令格式、控制方式、总线结构等因素有关，并因机型不同而稍有差异。但一般来说，控制器必须包含以下几个部件。

① 指令指针

指令指针（Instruction Pointer, IP）又称为指令计数器或指令地址寄存器，或称为程序计数器。它的功能是指示程序执行的顺序。在取指令阶段，它用于指示本指令的地址；而当指令执行完毕后，它又用来存放下一条将要执行的指令的地址。

② 指令寄存器

指令寄存器（Instruction Register, IR）的功能是保存计算机正在执行的指令代码，该代码是从存储器读出后送来的。一般情况下，指令执行期间指令寄存器的内容是不会改变的。当一条指令执行完毕后，新的指令将会从存储器读入该寄存器中。

③ 指令译码器

指令译码器（Instruction Decoder, ID）就是指令分析器，它根据指令的内容及各种标志进行分析后，产生本条指令所需要的各种操作信号，并送往各个执行部件。

④ 控制信号发生器

微型计算机是一种极为复杂的电子装置，它的每一个操作步骤都是严格按照时序要求进行的。不同的指令，执行的时间也不相同。控制信号发生器（或称时序部件）用来产生执行各种基本操作所需要的一系列控制信号，以保证计算机能够正确完成规定的运算任务。

(3) 数学协处理器

数学协处理器是为了增强主处理器的数学处理功能，与 CPU 配套使用的专用微处理器。它具备浮点、双字长整数等多种数据类型的三角函数、对数、指数等指令，在执行这

类指令时，能使处理速度成倍地提高。所以，对注重数值计算的系统，一般都需要配上数学协处理器。数学协处理器也称为数值数学协处理器或浮点运算器。

随着计算机技术的发展，其他类型的协处理器也应运而生，如加快图形处理的图形协处理器，总线主适配卡中的处理器等等。但大多数人所谈到的协处理器就是指 Intel 87 系列的协处理器（486 以后，数学协处理器被集成到 CPU 内部）。

3. 存储器

存储器是计算机中存储信息的部件。按照存储器在计算机中的作用，可分为主存储器（即内存储器，简称内存）、辅助存储器（即外存储器，简称外存）、高速缓冲存储器等几种类型，它们都能完成数据的存取工作，但性能及其在计算机中的作用却有很大差别。

(1) 主存储器

主存储器用于存放计算机当前执行的程序和需要使用的数据，它的存取速度快，CPU 可以直接对它进行访问。主存储器主要由半导体存储器件组成，下面说明它的组成和基本工作原理。

主存储器包括存储体、地址寄存器、选址部件、数据缓冲寄存器以及读写控制电路等基本部件，其中存储体是存放信息的实体，它被分为若干个存储单元，每个存储单元存放一串二进制数（例如一个字节）。为了能够区分存储体中的不同单元，需要按照一定顺序（如按字节）对它们进行编号，这些编号就称为存储地址，简称地址。如图 1-4 所示，存储体共有 N 个存储单元，地址编号为 $0 \sim (N-1)$ ，每个地址中存放的数据称为地址的内容（简称内容），CPU 可以对每个地址中的内容进行读写。

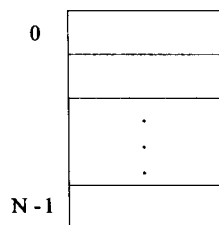


图 1-4 存储单元示意图

其他几个部件与存储体之间的关系如图 1-5 所示，它们的作用分别如下。

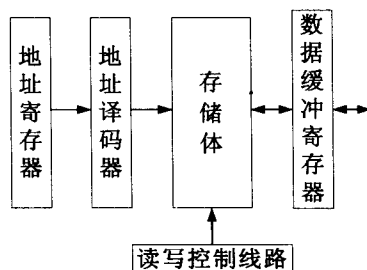


图 1-5 主存储器结构示意图

地址寄存器和选址部件（通常称为地址译码器）用于接受 CPU 送来的地址代码，保存待访问的存储单元地址，并将这种地址编码转换为使对应单元被选中的信号。

数据缓冲寄存器用于暂存准备写入存储器的数据或暂存从存储器读出的数据，它位于存储器和 CPU 之间，肩负着缓冲和传递信息的作用。

读写控制电路接受 CPU 送来的读或写命令，并把这些命令变为控制整个存储器协调工作的时序信号，保证存储器完成读或写操作。

(2) 辅助存储器

辅助存储器是作为主存储器的后备和补充而被人们广泛使用的存储设备，它的特点是存储容量大、成本低，并可脱机保存信息，主要用于存放不是当前正在运行的程序和用到的数据。由于辅助存储器的存取周期比主存储器长，不直接和 CPU 交换数据，而是先和主存成批交换数据，然后再由主存与 CPU 通信，因此它被称为外部存储器。在微型计算机中，常见的辅助存储器有软盘存储器、硬盘存储器、光盘存储器、移动硬盘以及 U 盘等。

(3) 高速缓冲存储器

主存储器的读写速度对整个系统的性能有很大影响。近年来，CPU 时钟频率大大提高，因而存储器的存取速度已经成为整个系统的瓶颈。为了解决这一问题，出现了所谓 Cache 技术。Cache 即高速缓冲存储器，它是位于 CPU 和主存之间的、规模较小但速度很高的存储器，用于保存主存储器中一部分内容的拷贝。当主机读写数据时，首先访问 Cache，只有在 Cache 中不含有所需要的数据时，CPU 才会去访问主存，从而很好地解决了 CPU 和主存之间的速度匹配问题。

4. 输入输出接口及外部设备

微型计算机使用的外部设备不仅种类繁多，工作速度比微处理器慢，而且和主机交换的信息也有多种形式，有数字量、模拟量或开关量等各种数据信息，也有标志外部设备工作的状态信息和主机向外部设备发出的控制信息。因此，当外部设备和主机交换信息时，不能简单地直接相连，除外部设备本身的控制驱动电路之外，它们之间还需要一个中间桥梁，这就是所谓的接口电路（通常也称为适配器），如图 1-6 所示。通过该接口电路，不仅可以在主机和外部设备之间传送数据、状态和控制信息，转换编码格式，而且可以使它们之间的工作速度得到协调。

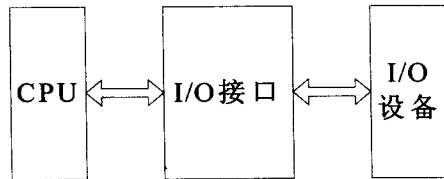


图 1-6 主机和外部设备之间的接口

外部设备是计算机系统的重要组成部分，各种人机交互操作、程序和数据的输入、计算结果或中间结果的输出、被控对象的监测和控制等，都必须通过外部设备才能实现。

外部设备主要有键盘、鼠标、显示器、打印机、绘图仪、扫描仪等。

5. 总线

总线是连接各主要部件的公共信息通道。按照信号类型可将总线分为数据总线、地址总线和控制总线。其中，数据总线主要传输数据，为双向的，既可以输入，又可以输出；地址总线传送地址信息，为单向的，决定数据或命令传送给“谁”；控制总线传送各种控制信号。按照连接对象的不同，总线又分为局部总线（Local Bus）、系统总线（System Bus）和外部总线（External Bus），其中，局部总线用于微机内 CPU 与各外围芯片之间的连接，系统总线用于微机内各部件之间的连接，而外部总线则是用于微机与外部设备之间的连接。

1.3 微处理器结构

1.3.1 8086/8088 微处理器

1. 8086/8088 微处理器内部结构

Intel 公司 1978 年推出的 8086 采用 HMOS 工艺技术制造, 单一+5V 供电, 芯片的内部数据总线和外部数据总线都是 16 位, 地址总线为 20 位, 可最大寻址 1MB 的存储空间。随后 Intel 公司于 1979 年推出了成本更低的 8088 芯片, 8088 外部的数据信号线降到 8 条, 以使 8088 能够获得已开发的 8 位硬件的支持。8088 芯片最早于 1981 年用于 IBM PC/XT(IBM Personal Computer/eXtended Type) 中。

图 1-7 是 8086/8088 的内部结构框图, 从中可以看出 8086/8088 微处理器由两个既相互独立又相互配合的重要部件组成, 一个是总线接口部件 (Bus Interface Unit, BIU), 另一个是执行部件 (Execution Unit, EU)。

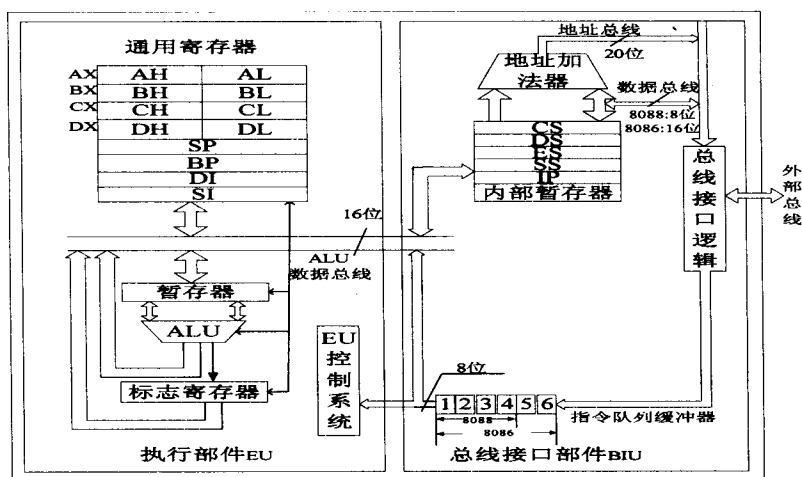


图 1-7 8086/8088 内部结构

(一) 总线接口部件 BIU

(1) BIU 的功能

BIU 是联系微处理器内部与外部的重要通道, 主要功能是负责微处理器内部与外部 (存储器和 I/O 接口) 的信息传递。具体地讲, BIU 完成以下几个主要任务。

① 取指令

BIU 从内存取出指令送到指令队列中 (这时 EU 可以取其中的指令来执行)。只要指令队列中不满 (6 个字节指令队列的 8086 空 2 个字节以上, 4 个字节指令队列的 8088 空 1 个字节以上都称为不满), BIU 即通过总线控制逻辑从内存单元中取指令代码送往指令队列中。当 EU 执行转移类指令时, 指令队列立即清除, BIU 又重新开始从内存中取转移目标

处的指令代码送往指令队列。

② 传送数据

EU 在执行指令过程中需要内存或 I/O 端口的数据时, BIU 从外部 (内存或 I/O 接口) 取数据 (读或输入) 或把 EU 执行的结果送到外部 (写或输出)。当 EU 需要 BIU 访问外部时, EU 就向 BIU 发总线请求, 如果此时 BIU 空闲 (即无取址操作), 则 BIU 会立即响应 EU 的总线请求, 进行数据传送; 如果此时 BIU 正在忙于取指令, 则 BIU 在完成当前的取指操作后才去响应 EU 的总线要求。

③ 形成物理地址

BIU 无论是取指令, 还是传送数据, 都必须指示内存单元 (取指或传送数据) 或 I/O 端口 (传送数据), 这就需要指明具体的实际地址 (物理地址), 这个任务由 BIU 的地址加法器完成。它将 16 位段地址左移 4 位形成 20 位 (相当于乘以 16 后, 再与 EU 送来的 16 位偏移地址通过地址加法器相加得到 20 位物理地址), 如图 1-8 所示。

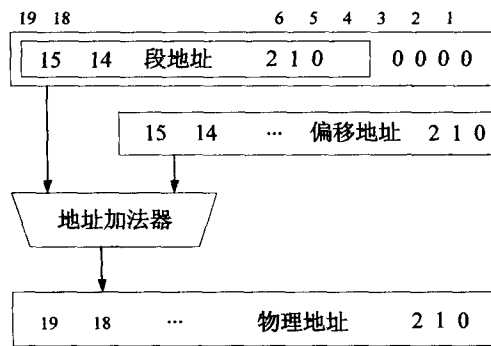


图 1-8 物理地址形成

(2) BIU 的组成

BIU 由段寄存器、指令指针、指令队列和地址加法器组成。这些组成部分的含义及用途说明如下。

① 4 个 16 位的段地址寄存器

代码段寄存器 (Code Segment, CS): 存放程序代码段起始地址的高 16 位。

数据段寄存器 (Data Segment, DS): 存放数据段起始地址的高 16 位。

堆栈段寄存器 (Stack Segment, SS): 存放堆栈段起始地址的高 16 位。

扩展段寄存器 (Extended Segment, ES): 存放扩展数据段起始地址的高 16 位。

② 1 个 16 位的指令指针 IP

它用于存放下一条要执行指令的偏移地址, 不能作为一般寄存器使用。

③ 20 位的地址加法器

它负责由段地址与偏移地址向 20 位物理地址的合成。

④ 指令队列

指令队列 (Queue) 用于存放预取的指令, 采用预取指令的方法可以减少微处理器的等待时间, 提高运行效率。由图 1-9 可知, 由于 BIU 取指令与 EU 执行指令是相对独立的并