

高级微型计算机系统 及接口技术

苏广川 沈瑛 编著



GAO JI
WEI XING JI SUAN JI
XI TONG JI JIE KOU JI SHU

北京理工大学出版社

高级微型计算机系统 及接口技术

苏广川 沈 璞 编著

北京理工大学出版社

内 容 简 介

本书全面介绍了高性能微型计算机系统(80X86)的结构、组成及工作原理,重点阐述了实地址方式和保护方式汇编语言程序设计方法,这对理解 Windows 98 操作系统内核极为有益。在程序设计方面强调了结构化、模块化程序设计方法和进程管理方法,使程序设计具有先进性、实用性,可缩短程序的开发周期,书中对现代微型机系统总线技术、存储器技术、中断技术、并行接口与串行接口技术、人机界面接口技术以及高级图形控制编程技术等都作了深入的探讨。

本书以 32 位微型机为主线,内容丰富、系统性强、选材适当,将基础理论与高级微处理机技术有机结合,使软件与硬件设计融于一体,是非计算机专业本科生计算机教学的理想用书和计算机工程人员参考书。

图书在版编目(CIP)数据

高级微型计算机系统及接口技术/苏广川,沈瑛等编著.

—北京:北京理工大学出版社,2001.6

ISBN 7-81045-819-1

I. 高… II. ①苏… ②沈… III. ①微型计算机—计算机系统 ②微型计算机—接口 IV. TP36

中国版本图书馆 CIP 数据核字(2001)第 0028411 号

责任印制:母长新 责任校对:陈玉梅

北京理工大学出版社出版发行

(北京市海淀区中关村南大街 5 号)

邮政编码 100081 电话(010)68912824

各地新华书店经售

涿州市星河印刷厂印刷

*

787 毫米×1092 毫米 16 开本 34 印张 835 千字

2001 年 6 月第 1 版 2001 年 6 月第 1 次印刷

印数:1—4000 册 定价:44.00 元

※图书印装有误,可随时与我社退换※

前 言

近代计算机科学技术迅猛发展,促进了信息技术产业的新革命,使社会生产力得到极大的提高,使人们的生活方式发生了深刻的变革,可以说微型计算机已深入到人类生活的各个方面,与人类结下了不解之缘,学习和掌握计算机原理及技术已成为人们的迫切愿望。

多年的微型计算机教学,使我们经常思考这样一个问题,计算机技术发展如此之快,怎样才能使学生在较短的时间内,系统地学习和掌握微型计算机的基础理论,同时又不失时机地跟上微机的发展步伐,及时了解微机的新技术,增强学生的软件、硬件设计能力,本书正是基于这一宗旨写成的,既考虑到学习的顺序性,又考虑到教材的先进性。

本书全面介绍了高性能微型计算机系统(80X86)的结构、组成及工作原理,重点阐述了实地址方式和保护方式汇编语言程序设计方法,这对理解 Windows 98 操作系统内核极为有益、在程序设计方面强调了结构化、模块化程序设计方法和进程管理方法,使程序设计更具先进性和实用性,可缩短程序的开发周期。书中对现代微型机系统总线技术、存储器技术、中断技术、并行接口与串行接口技术、人机界面接口技术以及高级图形控制编程技术等都作了深入的探讨。

本书以 32 位微型机为主线,内容丰富、系统性强、选材适当,将基础理论与高级微处理机技术有机结合,使软件与硬件设计融于一体,是非计算机专业本科生计算机教学的理想用书和计算机工程人员参考书。

全书共分十一章:第一章微型计算机的硬件基础,介绍微型机技术的发展概况,给予人们以创新的启迪,同时还介绍了一般微型机的组成原理和典型结构;第二章高档微处理器的工作模式,重点介绍 80386 微处理器的内部结构;第三章微处理器实模式软件结构及程序设计,系统介绍了 80X86 的指令系统及汇编语言程序设计方法,学习结构化、模块化设计技巧,利用进程管理方法直接对可执行程序进行集成,掌握磁盘文件管理的编程方法;第四章保护模式软件结构,介绍现代微机先进的工作模式——虚拟地址保护模式,学习多任务切换技术以及保护模式程序设计方法;第五章半导体存储器技术,介绍半导体存储器分类和微型机存储器系统的设计方法;第六章微型计算机系统总线,除对 ISA、EISA 总线作一般介绍外,重点介绍了 PCI 总线的工作原理和总线结构;第七章输入输出接口技术,全面介绍了并行接口、定时器/计数器、DMA 控制器以及人机界面接口等;第八章中断技术,着重介绍中断的概念、8259A 中断控制器的工作原理以及中断服务程序的设计方法;第九章串行通信接口及其应用,介绍串行通信的基本概念以及 8250 异步适配器及其编程技术;第十章数据采集系统及其应用,介绍数据采集系统的组成,A/D、D/A 接口电路的设计技术;第十一章图形显示高级实用编程技术,讨论文本显示和图形显示的各种方法,重点介绍图形控制器的编程技术,可在底层直接对图形进行快速读写,同时还介绍了弹出式菜单以及鼠标接口编程方法。本书第二、三章由沈瑛编写,其余章节由苏广川执笔。

在本书的编写过程中,得到了王士宏副教授的大力支持,为本书的文字编辑和图形绘制做了大量的工作,在此表示衷心地感谢。由于时间仓促和水平所限,书中会存在许多不足和问题,敬请读者批评指正。

编 者

2001 年 3 月

目 录

第一章 微型计算机的硬件基础	1
1.1 微型计算机技术的发展概况	1
1.2 微型计算机系统的基本组成	4
1.2.1 硬件	4
1.2.2 软件	7
1.3 计算机中的数制与编码	8
1.3.1 二进制格式	8
1.3.2 浮点数格式	9
1.3.3 BCD 格式	10
1.3.4 ASCII 码	11
1.4 地址	11
1.5 计算机基本操作过程	13
习题	15
第二章 高档微处理器的工作模式	17
2.1 80386CPU 内部结构	17
2.2 80386CPU 寄存器结构	18
2.3 80386 微处理器接口信号	25
2.4 80386 非流水线和流水线总线周期	30
2.5 读和写总线周期时序图	33
2.6 存储器地址空间的硬件结构	36
2.7 80386 微机系统结构	37
习题	41
第三章 微处理器的实模式软件结构及程序设计	42
3.1 引言	42
3.2 微处理器实模式软件模型	43
3.3 微处理器的寻址模式	49
3.4 实模式指令系统	58
3.4.1 数据传送指令	59
3.4.2 算术运算指令	64
3.4.3 逻辑运算和移位指令	77
3.4.4 程序转移指令	85
3.4.5 串操作指令	92
3.4.6 处理机控制指令	97
3.5 汇编语言程序设计	98
3.5.1 汇编语言和汇编程序	98
3.5.2 汇编语言语句	99
3.5.3 汇编语言程序结构	100
3.5.4 宏指令及条件汇编	111
3.5.5 汇编语言程序设计举例	116

3.5.6 结构化模块化程序设计·····	144
3.5.7 进程管理·····	149
3.5.8 程序驻留(TSR)·····	165
3.6 磁盘文件管理 ·····	167
3.6.1 磁盘存储结构·····	167
3.6.2 磁盘文件管理 ·····	173
习 题 ·····	183
第四章 保护模式软件结构 ·····	185
4.1 引言 ·····	185
4.2 保护模式存储器管理和地址变换 ·····	185
4.3 描述符与页表项 ·····	190
4.4 多任务与特权保护 ·····	194
4.5 虚拟 8086 模式(VIRTUAL 8086 MODE) ·····	201
4.6 保护模式下操作系统类指令及其编程方法 ·····	202
4.6.1 保护模式下操作系统类指令·····	202
4.6.2 特殊汇编伪指令及系统初始化·····	205
4.6.3 在保护方式下程序设计举例·····	206
习 题 ·····	211
第五章 半导体存储器接口技术 ·····	212
5.1 半导体存储器分类 ·····	212
5.2 存储器的主要性能指标 ·····	214
5.3 只读存储器 ROM ·····	214
5.4 静态随机存储器(SRAM) ·····	224
5.5 动态随机存储器(DRAM) ·····	234
5.6 视频 RAM(VRAM) ·····	243
习 题 ·····	246
第六章 微型计算机系统总线 ·····	247
6.1 PC XT 总线 ·····	247
6.2 ISA 总线 ·····	249
6.3 EISA 总线 ·····	251
6.4 VESA 局部总线(VL-Bus) ·····	259
6.5 PCI 局部总线 ·····	262
6.5.1 PCI 总线信号定义 ·····	266
6.5.2 总线命令·····	269
6.5.3 PCI 总线的数据传输 ·····	270
6.5.4 总线仲裁·····	272
6.5.5 PCI 配置寄存器 ·····	273
6.5.6 PCI 总线应用 ·····	278
习 题 ·····	279
第七章 输入输出接口技术 ·····	280
7.1 微机接口的基本概念 ·····	280
7.1.1 接口功能·····	280
7.1.2 端口编址方式·····	281

7.1.3 输入输出传送方式·····	282
7.2 8255A-5 可编程并行接口及应用·····	284
7.2.1 8255A-5 芯片的结构及功能·····	285
7.2.2 8255A-5 在系统中的应用·····	289
7.3 8253-5 可编程定时器及其应用·····	294
7.3.1 8253-5 的内部结构·····	294
7.3.2 8253-5 的工作方式·····	296
7.3.3 读写说明·····	298
7.3.4 8253-5 在系统中的应用·····	300
7.4 8237A-5 DMA 控制器及其应用·····	304
7.4.1 8237A-5 芯片的硬件结构和功能·····	304
7.4.2 DMA 读写操作·····	310
7.4.3 工作方式·····	311
7.4.4 DMA 传输类型·····	312
7.4.5 几个特殊操作的说明·····	312
7.4.6 8237A-5 在系统中的应用·····	315
7.5 人——机输入接口·····	322
7.5.1 触摸屏·····	322
7.5.2 光笔·····	323
7.5.3 打印机接口技术·····	324
7.5.4 键盘接口·····	331
习题·····	336
第八章 中断技术 ·····	338
8.1 中断类型和中断向量表·····	338
8.2 中断响应过程·····	339
8.3 8259A 中断控制器及其应用·····	340
习题·····	357
第九章 串行通信接口及其应用 ·····	358
9.1 异步通信接口·····	358
9.1.1 串行通讯的基本概念·····	358
9.1.2 串行总线标准 RS-232C·····	362
9.1.3 20mA 电流环标准·····	365
9.2 异步通讯适配·····	366
9.2.1 8250 可编程异步通讯接口·····	366
9.2.2 8250 内部寄存器功能·····	368
9.2.3 INS 8250 通讯编程·····	373
9.3 RS-422 标准接口总线·····	378
9.3.1 RS-422 标准接口总线·····	378
9.3.2 RS-485 标准接口总线·····	379
9.3.3 RS-485 的应用举例·····	382
习题·····	382
第十章 数据采集系统及其应用 ·····	383
10.1 数据采集系统的组成·····	383

10.2 A/D 接口电路设计	388
10.3 D/A 转换器及其应用	397
习题	400
第十一章 图形显示高级实用编程技术	401
11.1 显示模式	401
11.1.1 文本显示	403
11.1.2 图形显示	409
11.2 图形高级编程技术	418
11.2.1 建立宏库	418
11.2.2 画点	420
11.2.3 画直线	422
11.2.4 画圆	426
11.2.5 图形的拷贝	431
11.2.6 动画图形显示	433
11.2.7 填充矩形	433
11.2.8 在屏幕上开窗	435
11.2.9 存储屏幕图形	437
11.2.10 将磁盘中的图形文件写入显示缓冲区	440
11.3 高分辨图形系统及其编程技术	443
11.3.1 彩色图形调色板	443
11.3.2 TMS34010 图形处理器	446
11.4 窗口管理文件及编程技术	455
11.4.1 物理视频缓冲区	455
11.4.2 逻辑视频缓冲区	455
11.5 鼠标输入器及其编程	466
11.5.1 鼠标器程序设计接口	467
11.5.2 鼠标器功能调用简介 INT 33H	469
11.5.3 对鼠标及键盘的编程	474
11.6 菜单设计及其编程	489
习题	505
附录	506
附录一	506
附录二	519
附录三	521
附录四	526
附录五	530
附录六	534
参考文献	535

第一章 微型计算机的硬件基础

1.1 微型计算机技术的发展概况

近年来,微型计算机得到了飞速的发展,其应用领域也越来越广泛,从科学计算到商业、金融以及军事等方面,处处都显示了微型计算机的重要地位。微型计算机与通信、网络紧密结合,又促使微型计算机得到进一步发展,微型计算机的发展不仅体现在数量方面,而更重要的体现在性能的提高,性能的提高主要依赖于两个因素,一是采用新材料新工艺的超大规模集成技术,使微处理器的体积更小,性能更卓越;另一方面是微处理器采用先进的结构和技术,结构的变革促使微处理器性能的提高远远超过了材料工艺进步所取得的效果。微型计算机技术的发展历程给予人们许多启迪,使后人可以从中找到新的发展途径。下面以 Intel 的 80X86 为例,说明微型计算机在各个不同阶段通过采用新技术使处理器的能力不断增强。

1972 年,第一代微处理器(Microprocessor)4004 问世,数据宽度为 4-bit,包含大约10000 个晶体管。由于性能低,仅作一些简单的诸如计算器的应用。

1974 年,第二代微处理器相继问世,其代表产品有 8008、8080 和 8085,它们是 8 bit 微处理器。这些新型微处理器具有较高的操作性能,较强的系统能力和较方便的编程环境,能满足许多应用的需求,例如电子仪器、现金收入记录机以及打印机等。

70 年代中期,半导体制造商提出了开发第三代 16 bit 微处理器的计划,1979 年第一台 16 位机 8086 诞生。次年,微处理器内部总线为 16 位而外部数据总线为 8 位的 8088 也随之问世,从此出现了 80X86 家族。

16 位高性能微处理器在专用和通用微型计算机中得到广泛的应用;它能处理 8 位、16 位以及特殊用途的数据类型,并且具有强大的指令系统。

8086 是 Intel 公司继 8085 之后推出的 16 位微处理器;内部寄存器、内部总线都是 16 位;外部的数据总线也是 16 位;其地址总线从 8 位机的 16 位增加到 20 位,可寻址的内存空间达 1MB,供程序执行的内存空间为 640KB。

8086 微处理器的内存管理采用分段管理模式,利用四种段寄存器:代码段 CS、数据段 DS、堆栈段 SS 和附加数据段 ES,分别指示不同的数据类型在内存中存放的起始位置,以便于集中管理。段寄存器和偏移地址寄存器通过地址加法器形成 20 位物理地址,完成对内存的访问。

8086 微处理器另一重要特点是在执行指令时采用了流水线方式。微处理器将总线接口部件(BIU)和执行部件(EU)分成两个独立的处理单元,可以进行并行和异步工作。在总线接口部件中增加一个 6 字节长的指令队列,存放预取指令。指令译码和指令执行任务均由执行部件完成,克服了 8 位微处理器取指令—译码—执行—回写结果串行操作的缺点,提高了处理器的工作效率。

此外 8086 微处理器可以在两种模式下工作,即最小工作模式和最大工作模式。微机系统通常选用最大工作模式。在这种工作模式下,8086 主微处理器可以利用局部总线与协处理器和从处理器一起构成处理器系统,将浮点运算和函数运算的任务交给协处理器完成;将 I/O 处理如 DMA 接口等任务交给从处理器完成,这一特点一直为以后 80X86 的新一代产品所沿用。

80286 是 1982 年推出的一种 16 位超级微处理器,其内部结构由总线部件(BU)、指令部件(IU)、执行部件(EU)和地址部件(AU)四部分组成,它们能够独立并行工作,可以实现流水线化作业,使总线利用率达到最佳状态。其中总线部件用来产生访问片外存储器和 I/O 口时所需要的地址、数据和命令信号。另外总线部件还控制与 80287 扩充数值运算处理器(NPX)的接口,80287 对存储器的访问全部通过这个扩充处理器接口进行,不像在 8086 和 8087 系统中,8087 本身可以对存储器直接访问。同 8086 一样,在这个部件内也有一个 6 字节的指令队列寄存器,利用总线的空闲时间执行指令的预取操作。

80286 的指令部件从预取队列寄存器每取 1 个字节的指令代码,就对其进行译码,并把被译码的指令展开成 69 位的内部形式,存放到 3 条已译码指令队列寄存器(69 位 $\times$ 3)中。

80286 的执行部件包括 ALU,1536 字 $\times$ 35 位的微代码 ROM 和寄存器文件,另外还有中断控制电路以及向段寄存器加载时用的测试电路等,执行部件能用很高的效率执行由指令部件译码的指令。

在 80286 的地址部件中,通过偏移量加法器计算出有效地址的偏移量之后,对该偏移量的段边界和访问权进行检查,最后执行虚拟地址变换。另外,在这个部件中还包括段寄存器和系统表寄存器的高速缓冲器。

80286 在技术上的显著特点是实现虚拟存储系统。它可以支持两种工作方式,一种是 8086 实地址模式,只能在 1MB 内存范围内执行程序,可以在代码一级上与 8086 指令系统兼容。另一种是保护方式下的虚拟地址模式。保护方式可以提供存储器管理和保护的手段,支持操作系统和任务的分离,并对程序和数据实现保护。80286 的实际存储空间为 16MB,但在虚拟地址方式下,每个用户可用的虚拟存储空间为 1024MB(1GB),在运行时,使用 80286 的存储器管理能力,自动地将每项任务的虚拟地址空间转换成 16MB 物理存储空间。

在 1985 年,Intel 公司推出了第一个 32 bit 微处理器 80386DX,其性能达到了小型计算机的水平,随后于 1988 年推出了外部总线为 16 位的 32 位机 80386SX。80386 微处理器的出现象征着微处理器技术进入了一个新的里程碑。它的寄存器、数据总线和地址总线都扩展到 32 位,使得 80386 可以寻址 4GB 的内存,其虚拟存储器空间达 16 TB(1TB=1024 GB)。

80386 微处理器由六个主要功能块组成,它们是:总线接口单元、代码预取单元、解码单元、执行单元、内存页管理单元和内存段管理单元。比 80286 增加了代码预取单元和内存页管理单元,使处理器功能进一步增强。采用分页技术,可将需要载入内存的信息,以 4KB 为单位从硬盘读至内存,这使微处理机中的内存得到有效的使用,并且可以提高加载的速度。同样地以页为单位对硬盘写入,也将提高写入速度。

80386 的微处理器内部增加了一块 16 位预取高速缓冲器,用以存放将要执行的后几条指令,这样加速了指令执行速度。

80386 采用了新的内存管理技术,除像 80286 一样支持实地址方式和保护方式外,还支持虚拟 8086 方式(简称 V86)。所谓 V86 模式是指一个多任务环境,即模拟多个 8086 微处理器

同时执行多个任务。V86 模式允许 80386 将内存划分成多个部分,每一个部分由操作系统分配给不同的应用程序,操作系统可按时间平均或优先权方式,给每一个应用程序分配执行时间。有了 V86 模式,可使 Windows、OS2 等操作系统在多任务环境下顺利运行。

Intel 公司于 1989 年推出了 80486 微处理器,这是 80386 的改进型,在微处理内部增加了浮点运算单元和高速缓冲单元,与 80386 相比,具有如下一些新的特点:

① 80486CPU 采用了简化指令系统(RISC)技术,使芯片上不规则的控制部件减少,用布线硬逻辑代替以前的微代码程序控制,进一步缩短可变长指令的译码时间,达到基本指令可以用一个时钟周期完成。在相同的工作频率下,80486CPU 的处理速度,比 80386 提高 2~4 倍。

② 80486 内部采用单倍时钟或在内部对外时钟实现 2 倍频、3 倍频,使电路工作稳定性进一步提高。由于外部振荡频率较低,也相应减少了系统的噪声干扰。

③ 80486 内部增设 8KB 的高速缓冲器,用以存放指令和数据。高速缓冲器采用 4 路成组相关方式,内部总线宽度为 128 位,为频繁访问的指令和数据提供了快速通路,同时也可降低对系统总线的占有率。

④ 80486 的内部数据总线宽度为 64 位,采用一种称之为猝发式(Burst BUS)的总线技术,在取得一地址后,与该地址相关的一组数据可以连续进行输入/输出,有效地解决了 CPU 和存储器之间数据交换问题。此外,80486 与协处理器 FPU 的数据通道为 64 位,CPU 与 Cache(高速缓冲器)以及 Cache 与 Cache 之间的数据通道是 128 位,这些都大大加快了指令运行速度。

在 1993 年,Intel 公司推出了新型的 Pentium 微处理器(简称 80586),是为迎接 Windows 操作系统与多媒体时代的来临而全新设计的,其主要特点是:

① Pentium 采用全新设计的增强型浮点运算器,使得浮点运算速度比 80486DX 快 3~5 倍。

② Pentium 微处理器采用了超标量指令流水线结构(Super Scalar Pipeline),它有两条指令流水线,分别命名为 U 和 V 指令流水线。其中 U 指令流水线可执行所有的整型与浮点指令,而 V 指令流水线则执行简单的整数指令与 FXCH 浮点指令。在最佳状况下,Pentium 可在一个时钟周期内执行完两条指令。

③ Pentium 采用双重分离式高速缓存,将指令 Cache 和数据 Cache 分离,各自拥有 8KB 的高速缓存,体现了哈佛结构的优点,可使取指令和取数据同时进行。

④ Pentium 采用 64 位数据总线,数据传输能力高达 528MB/s,比 80486 的传输率高出 5 倍。同时,还采用了 PCI 局部总线,解决了与存储器、外设通信的数据阻塞问题。

⑤ Pentium 微处理器设置了分支目标缓冲器(Branch Target Buffer,BTB),可以预测分支指令,在分支指令进入指令流水线之前,预先安排指令顺序,从而使指令流水线不致产生停滞或混乱。

在 1995 年,Intel 公司又推出 Pentium pro(6×86),它是高速、高性能的新型微处理器,其指令运行速度是原 Pentium 处理器的两倍。Pentium pro 的主要技术特点如下:

① 14 级超级指令流水线;

② 三路超级标量微结构;

③ 无序执行、动态分支预测和推理执行;

④ 在 CPU 内部有 8KB 两路相关联的指令高速缓存,8KB 四路相关联数据高速缓存;

⑤ 在片外有与 CPU 速率相匹配的 256KB 二级高速缓存;

Pentium Pro 是针对数据密集型运算的工作站而设计的,它使用 Windows NT 和 Windows 95 操作系统时,可以发挥最佳的性能。

微处理器从应用角度上可以将其分成两类:可再编程的微处理器(Reprogrammable Microprocessors)和嵌入式的微控制器(Embedded Microcontrollers)。

嵌入式控制还可以进一步分成两类,即主事件控制和数据控制。工业过程控制计算机是主事件控制的典型例子,它按事件的时序进行编程。另一方面,硬盘控制接口可以视为数据控制的应用,它处理的数据块或数据库必须在硬盘与内存之间迅速传输。

嵌入式控制的应用需要各种各样的系统特点和多种性能指标。根据市场的需求而开发的器件通常是价格低廉且集成度高,从图 1.1 可以看出,早先由多片 8080 完成的任务,现在只需由高度集成的 8 位单片机如 8048、8051 来完成。这些单片机主要用于事件控制,8051 的性能比 8080 约高一个数量级,它有更强的指令系统,在片内具有 ROM、RAM、定时/计数器、通用异步收发器(UART)和可编程的并行接口,现在这些类型的器件都称之为微控制器。

此后出现的器件如 80186 和 80188 专门设计用于数据控制,它不仅集成度高、性能好,还增加字符串指令和 DMA 通道,可以更好地用于数据传输控制。

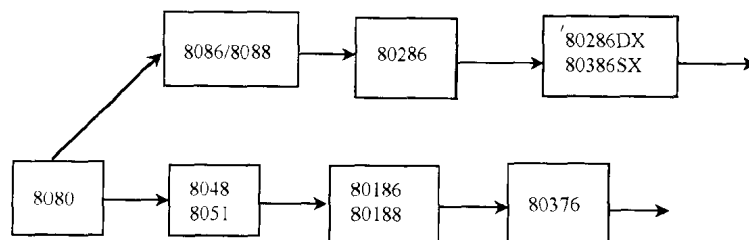


图 1-1 再编程和嵌入式控制的应用

再编程微处理器通常用于实现通用计算机 8086、80286、80386DX、80486、Pentium 等微处理器,主要用于再编程微型计算机。结构兼容性是开发再编程微处理器的一个关键问题,一般都采用向上兼容(Upward Compatible),即 8086/8088 的机器码可以在 80286 和 80386DX 机器上运行,而相反的情况是不许可的。

1.2 微型计算机系统的基本组成

微型计算机系统由两大基本部分组成:硬件(Hardware)和软件(Software)。其中硬件主要是指电路及设备,而软件是程序的集合。

1.2.1 硬件

微型计算机的典型系统结构如图 1-2 所示。

它由中央处理器 CPU(Central Processing Unit)、时钟电路、存储器、输入/输出子系统、总线控制逻辑和系统总线组成。在微型计算机中,CPU 通常也称作为 MPU(Microprocessor Unit)。它的主要用途是对指令译码,产生时序控制信号,执行算术运算和逻辑运算。时钟电路是微型计算机的心脏,它可使微处理器和总线控制器之间实现同步。微处理器通常需要 1~4 个相时钟。

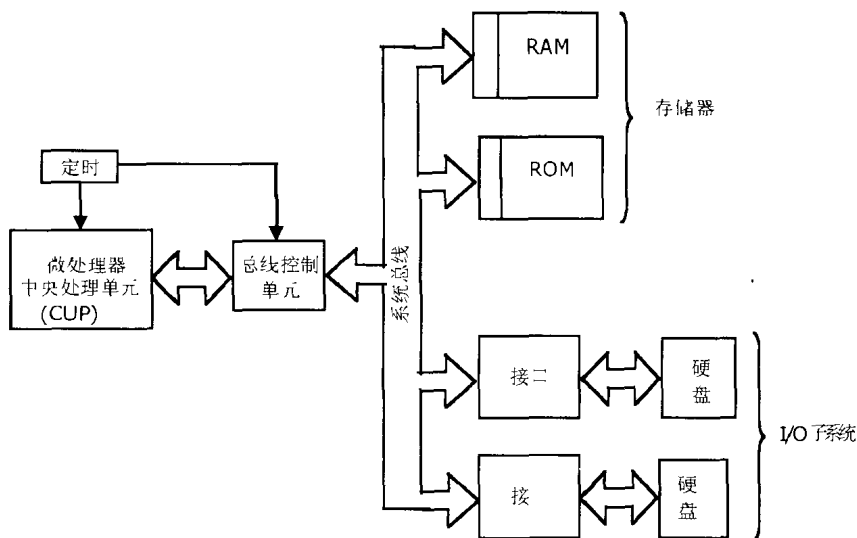


图 1-2 典型的微型计算机结构

存储器(Memory)既用于存储指令也用于存储数据。存储器常常分成多个模块,每个模块包含成千上万的存储单元,每个存储单元都用存储器地址(Memory Address)或简称地址来标识。存储单元用于存储信息,‘存储’即意味着存储器有能力保持它的信息,以便用于处理或输出。

在微型计算机系统中,存储器可以分为内存和外存。内存通常是指 CPU 可以通过地址直接读写的那些存储单元,一般存储量较小,暂时存放诸如操作系统等有用信息、当前要执行的程序和要处理的数据。内存可以进一步细分为程序存储器和数据存储器。程序存储器部分用于存储操作系统和应用程序的指令。数据存储器通常存放程序要处理的数据或程序执行的结果。当然程序也可以存放在数据存储器中。早期的微型计算机大都采用所谓冯·诺依曼(Von·Neumann)结构,即程序和数据共存于同一存储区中,取指令和取数据利用同一组数据总线,其优点是结构简单,但影响程序运行速度,因为取指令时就不能取数据。而现代计算机又主要采用哈佛(Harvard)结构或采用改进的哈佛结构,它将程序和数据分成两个存储器空间,各自均有自己的总线来分别传送指令和数据,这样取指令和取数据可以在同一时钟周期内进行,大大提高了程序运行速度。所谓改进的哈佛结构,是将存储器分成三个存储空间,即程序存储器、数据存储器空间和程序与数据共用的存储器空间,这样可以更具有灵活性。

典型的内存可以使用只读存储器(Read - Only Memory, ROM) 和随机存储器(Random - Access Read/Write Memory, RAM)集成电路来实现。ROM 中的信息属于非易失性(Non-volatile),即在电源关闭后,信息不会丢失。这类存储器只能读不能写。

另一方面,经常变化的信息或被处理的数据必须要存放在内存中,它们由微处理器读出,通过处理进行修改,然后再回写到存储器中去。因此,内存必须是 RAM 而不是 ROM。微型计算机的操作系统启动后必须从硬盘或软盘加载到 RAM,以便使微机在零等待状态下执行。RAM 是易失性(Volatile)存储器,即当电源关闭后,它保存的信息将全部丢失。

输入和输出单元是 MPU 用来与外界实现通信的重要手段。例如输入设备—键盘(Keyboard),允许用户向 MPU 输入信息或命令,又如鼠标(Mouse)可以实现更友好的用户输入界

面。

使用最为广泛的输出设备有显示器(Display)和打印机(Printer)。微机的输出单元用于向用户反馈信息。例如从键盘输入的字符同时在屏幕上显示,这可使用户确认输入是否正确。由 MPU 处理的结果也可以送至显示器或打印机。

系统总线(System Bus)是在微机中连接各部件的一组公共通信线,它们可以是电缆或印刷线路板上的一组引线。总线的两端一般设有发送门和接收门。根据总线的不同用途,可以将总线分成双向传输总线、单向传输总线。每一根总线上的电平可以有三种状态:逻辑 1、逻辑 0 和浮空(或高阻)状态。在总线上可以允许有多个发送门和多个接收门,但在同一时刻,只允许其中的一个发送门打开,向总线输出信息。假若有两个或两个以上的发送门同时打开,将引起总线竞争,并使系统产生难以预料的结果,系统将处于瘫痪状态。

总线结构使微机在系统结构上具有简单、规整和易于扩展的特点,使整个系统中各功能部件之间的相互关系成为面向总线的单一关系。这样,系统只要将符合总线规范的功能部件接到总线上,系统的功能就可以得到扩展。

准确地说,信息如何在总线上传输将由总线的规范来确定。按照传送的信息类型可将总线分为:地址总线 AB(Address BUS)、数据总线 DB(Data BUS)和控制总线 CB(Control BUS)。

1. 地址总线(AB)

地址总线是专门用来传送地址信息的总线,一般为单向传送,即由 MPU 发出。地址总线的宽度将决定 CPU 可直接寻址的内存空间大小,例如:8 位微型机的地址总线为 16 根(A15 ~ A0),所以此种微型机最大的内存空间仅有 64KB。对于 PC/XT,它有 20 根地址线(A19 ~ A0),可访问的内存空间是 1MB,再如 PC/AT 机,其 MPU 为 80286,地址总线为 24 根,它具有的内存容量为 16MB。

2. 数据总线(DB)

数据总线用于传送数据,具有双向传送功能。数据总线宽度,决定了系统的数据传输率,数据总线根数越多,所能传输的信息范围也就越大,因而微机处理信息的能力就越强,这就是为什么微机常常以数据总线多少来分类,例如 8 位机、16 位机、32 位机、64 位机,它们都可作为微机的重要发展阶段。但有些微型机微处理器内部的数据总线宽度与微处理器外部数据总线宽度不相一致,例如 8088 微处理器内部数据总线为 16 位,而与存储器、I/O 口通信的数据总线宽度仅为 8 位,故称 8088MPU 为准 16 位机。

3. 控制总线(CB)

控制总线用于传送各种控制信号,控制信号包括 MPU 对存储器和 I/O 的读写控制信号,还有外部对 MPU 本身的控制信号如 MPU 的复位信号 RESET,MPU 的总线请求信号 HOLD 以及中断请求信号等。控制信号的流动方向要视控制信号的功能而定,例如 MPU 对存储器和 I/O 的读写信号是 MPU 的输出信号,而外部对 MPU 的复位信号及总线请求信号是 MPU 的输入信号。而在最大模式下的 8086 总线请求和总线允许信号 $\overline{RQ}/\overline{GT}$ 是双向输入/输出信号。

总线上的信号必须同与总线相连的各个元件产生的信号相一致。总线与外设相连所需的电路称之为接口(interface),而总线控制逻辑(BUS Control Logic)是与 MPU 的接口。许多制造商提供多种多样的 IC 器件,这使接口和总线控制逻辑的设计任务变得简单。随着系统的复杂程度不同,总线控制逻辑可以部分或全部集成在 CPU 内部。

1.2.2 软件

计算机软件通常可以分成两大类:系统软件和应用软件。系统软件通常是指系统应该配置的那一部分软件,用于对计算机的管理和使用。例如操作系统、编译程序、汇编程序、调试程序、诊断程序、标准程序库等。操作系统是紧挨着裸机的第一层软件,它提供机器与用户之间的接口,使机器发挥更大的效率。操作系统最重要的部分是驻留监控程序。计算机在接通电源后,一部分操作系统始终驻留在内存中。驻留的监控程序必须能从用户接收命令,并能初始化由操作系统控制的相应装置。

操作系统还包括 I/O 驱动程序和文件管理程序,用于执行 I/O 操作和处理收集的大量数据并将其存储在海量存储设备中。一旦用户程序或其他系统程序需要使用 I/O 外部设备,外设本身并不能执行指定的操作,而需要依靠操作系统的驱动程序去执行任务,这样可免除在用户程序中包括 I/O 子程序。

文件管理是对文件的存储空间进行组织分配、负责文件的存取,并对存入的文件进行保护和检索。

此外,系统软件还包括各种各样的高级语言的编译器、汇编程序、文本编辑等。编程可分为三级,它们是:

- 机器语言
- 汇编语言
- 高级语言

机器语言编写的程序可由计算机理解并直接执行。汇编语言指令与机器语言指令逐条对应,但它是由助记符写成的,因而更容易理解。高级语言指令更接近英语,其结构与人们的编程思想相一致,但最终不管是汇编语言还是高级语言程序都必须由编译器翻译成机器语言。假如编译的是汇编语言,那么翻译器就称之为汇编器(Assembler);假如是高级语言,翻译器就称之为编译器(Compiler)或解释程序(Interpreter)。

因为不同的微处理器具有不同的指令系统,因此用机器语言编写的程序太复杂,阻碍了程序的开发。汇编语言程序由于是用助记符编写的,便于理解和记忆,但由于程序与处理器类型密切相关,因而程序的可移植性差。

用高级语言如 C 和 BASIC 书写的程序与处理器无关,不同的处理器可以采用相应的编译器,因而使程序移植性好,程序开发也很容易。尽管如此,利用汇编语言编程还是具有许多优点:

① 利用汇编语言编写的程序和高级语言编写的程序相比需要较少的存储器和较少的执行时间。

② 汇编语言可使程序员具有解决高技术问题的能力,而这些问题对于高级语言来说是困难的,甚至是无法解决。

③ 汇编语言的知识可以帮助人们更好地理解计算机的结构和工作原理,这是高级语言很难做到的。

④ 尽管许多新的应用程序的开发都使用高级语言,具有较容易编程和维护的特点,但这些程序运行时容易引起瓶颈效应。

⑤ 驻留程序及中断服务程序几乎总是使用汇编语言开发。

1.3 计算机中的数制与编码

从本质上讲,所有的计算机只能识别两种状态,即“0”和“1”。多个 0、1 的组合可以表示许多不同的状态。其中的 0 或 1 称之为位(bit)。8 位二进制的组合称为字节(Byte),计算机中的存储器,不管是内存还是外存,在存储信息时都是以字节为单位的;但存储单元由 9 位组成:8 位数据 + 奇偶校验位(Parity),且通常采用奇校验。例如字母 A 用数值表示为 01000001,该信息在存入磁盘和随机存储器时,自动插入奇偶的校验位,按照奇校验的规则,实际存放存储器的数是 01000001 - 1。计算机利用二进制形式表示数、字母、标点符号或其它信息。计算机中采用三种基本格式的数,它们是:

- 二进制整数
- 浮点数或实数
- BCD 码(Binary Coded Decimal)或十进制数。

1.3.1 二进制格式

数是从现实中抽象出来的,可以用各种规则和形式表示,设非负整数 X 作为基数,该基数具有 X 个不同的符号称之为数字,用一串数字如 $A_{n-1} \cdots A_1 A_0$ 来表示数 $A_{n-1}X^{n-1} + \cdots + AX + A_0$ 。

在数字系统中,基数可以是 2、8、10、16,分别称之为 2 进制,8 进制,10 进制,16 进制。例如:

$(1110)_2$ 该数的真值为 14

$(1110)_{10}$ 的真值为 1110

$(1110)_8$ 的真值为 584

$(1110)_{16}$ 的真值为 4368

将十进制数换成二进制数,采用除 2 取余数的规则,通过多次连除,即可完成转换。注意首次除 2 的余数为 2 进制的最低位数字。

将十进制数化为十六进制格式,则采用除 16 取余的同样规则来实现。

对于十进制小数,用定点 2 进制表示时,采用乘 2 取整的规则,首次取整获得的数字为定点 2 进制小数的最高位,通过多次连乘,直至取整后的余数全为 0 为止,例如:同样对于十六进制的定点小数,可采用乘 16 取整的规则来将十进制小数转换成十六进制定点小数。

$$\begin{array}{r} 0.375 \\ \times 2 \\ \hline \boxed{0}.750 \quad \text{高位} \\ \times 2 \\ \hline \boxed{1}.500 \\ \times 2 \\ \hline \boxed{1}.000 \quad \text{低位} \end{array}$$

结果为 $(0.011)_2$

将二进制转换成十六进制数格式表示的方法非常简单,只需将二进制数由低位到高位按4位逐一分组,每组用1位16进制数表示,例如:

二进制	0010	1010	0110	1010	0011	1100
十六进制	2	A	6	A	3	C

因而可以将16进制数格式视为2进制数的一种简化表示形式,以使书写简明。在计算机中大都采用16进制数形式,尤其在调试程序中,全部采用16进制形式来表示。

在计算机中如何表示有符号数是实现算术运算的一个极为重要的问题,最常见的有符号数编码为补码(2's Complement Format),即将数的符号与数一起统一编码,最高位为0表示正数,最高位为1表示负数。设b为一个数的绝对值,n为位数,则一个负整数-b的补码为:

$$-b = 2^n - b$$

例如 $b = (1234)_{10}$, $n = 16$, 则 $-b$ 的补码为:

$$2^{16} - 1234 = (1111101100101110)_2 = (\text{FB2E})_{16}$$

但对于正整数,直接化成二进数即可,如上例

$$b = (1234)_{10} = (0000010011010010)_2 = (04D2)_{16}$$

计算机中有时也使用反码的概念,正数的反码与原码相同,即符号幅度码;负数的反码可以通过先求该数绝对值的原码,然后逐位求反即得。例如 $n = 8$, $b = 46$, -46 的反码为

$$b = 00101110$$

$$(2^8 - 1) - b = 11010001 \quad \text{反码}$$

对于 -46 的补码可以对该负数的绝对值求反,然后加1(简称求反+1)。即

$$b = 00101110$$

$$(2^8 - 1) - b = 11010001$$

$$\begin{array}{r} + \quad \quad \quad 1 \\ 2^8 - b = 11010010 \quad \text{补码} \end{array}$$

利用补码表示有符号的数,可使算术运算简化,减法也可以通过加法来实现,即加减数的补码即可。例如: $83 - 36 = (37)_{10} = (25)_{16}$

$$\begin{array}{r} 01010011 \\ + 11010010 \\ \hline \end{array}$$

$$\boxed{1}00100101$$

↑

舍去

对于正常进位,舍去即得正确结果。

1.3.2 浮点数格式

浮点数亦称实数,可以表示变化范围极大的操作数,并在计算中采用定长有效尾数。计算机系统数值处理器采用的短实数格式(亦称IEEE浮点格式)为:

