

WEIJI YUANLI
YU JIEKOU JISHU

微机原理与接口技术

(第2版)

王玉良 吴晓非 张琳 禹可 编著



北京邮电大学出版社
www.buptpress.com

微机原理与接口技术

(第 2 版)

王玉良 吴晓非 张琳 禹可 编著



北京邮电大学出版社

·北京·

内 容 简 介

本书以 IBM-PC 微型计算机及兼容机为背景,全面系统地论述了微型计算机的基本原理与接口技术。全书内容丰富、图文并茂,讲述由浅入深、通俗易懂。

全书共分 8 章,内容安排注重系统性、先进性和实用性。前 4 章先介绍了微机的基础知识和运算基础,接着详细讲述了微处理器与微机的组成原理、系统结构、指令系统和汇编语言程序设计;第 5 章论述了存储器的原理与使用,并对高速缓存作了适当的介绍;第 6 章讨论了 PC 系列微机的各种常用总线;第 7 章阐述了 I/O 接口和中断技术;第 8 章讲述了各种接口技术的原理及应用。书中附有大量例题,各章都配有适当的习题与思考题。

本书可作为高等院校通信工程类、电子信息工程类专业和其他相近专业本科生的教材,也可作为从事微机应用与开发的科研及工程技术人员的参考书。

图书在版编目(CIP)数据

微机原理与接口技术/王玉良等编著. —2 版. —北京:北京邮电大学出版社,2006

ISBN 7-5635-1300-0

I. 微... II. 王... III. ①微型计算机—理论②微型计算机—接口 IV. TP36

中国版本图书馆 CIP 数据核字(2006)第 085985 号

书 名: 微机原理与接口技术

作 者: 王玉良 吴晓非 张琳 禹可

责任编辑: 王晓丹

出版发行: 北京邮电大学出版社

社 址: 北京市海淀区西土城路 10 号(100876)

北方营销中心: 电话 010-62282185 传真 010-62283578

南方营销中心: 电话 010-62282902 传真 010-62282735

E-mail: publish@bupt.edu.cn

经 销: 各地新华书店

印 刷: 北京源海印刷有限责任公司

开 本: 787 mm×1 092 mm 1/16

印 张: 25.5

字 数: 632 千字

印 数: 1—5 000 册

版 次: 2000 年 12 月第 1 版 2006 年 9 月第 2 版 2006 年 9 月第 1 次印刷

ISBN 7-5635-1300-0/TP·255

定价: 36.00 元

• 如有印装质量问题,请与北京邮电大学出版社营销中心联系 •

前 言

由于微型计算机的迅速发展及其应用日益广泛和深入,因此有必要对几年前我们编写的教材《微机原理与接口技术》进行较大的修改,作为本书的第2版。

本教材与作者2000年出版的《微机原理与接口技术》(北京邮电大学出版社)一书相比,在实用性和内容更新等方面,都有较大的改进和明显的提高。原教材中的第9~11章由于课堂教学没有讲授,所以这次修改时把这3章全部删掉了。这样使本教材更适用于本科的教学。

本教材仍以X86微处理器和PC系列微机为背景,全面系统地论述了微型计算机的基本原理与接口技术。

本书在内容的安排上注重系统性、实用性和先进性。讲述由浅入深,在讲清基本原理的基础上,尽可能做到通俗易懂。各章内容尽量做到前后衔接,侧重讲述基本概念和工作原理,做到理论联系实际。根据内容的需要,每章都附有一定数量的例题,书中的程序均调试通过;各章都配有适当的习题和思考题供读者练习。

为了有助于读者加速掌握本书的重点,巩固所学的知识,加深基本概念的理解,检查所做习题的对错,我们配合本书编写了一本《微机原理与接口技术复习指导和习题解答》,作为学习本教材的参考书。

全书共分8章,其中第1、第2章由张琳编写,第3、第5章由禹可编写,第4、第6章由吴晓非编写,第7、第8章由王玉良编写。王玉良负责全书的统稿。由于第2版是在第1版的基础上改编的,在此对第1版的编者杨紫珊教授和戴志涛副教授表示感谢。

本书是编者在多年从事微机原理与接口技术教学与研究的基础上并参考了国内外大量文献资料编写而成,在此,特向有关作者表示感谢。

由于编者水平有限,错误与不妥之处,敬请读者与专家赐正。

编 者
2006年6月于北邮

目 录

第 1 章 微型计算机的基础知识

1.1 微型计算机的基本组成	1
1.1.1 微型计算机的基本硬件构成	1
1.1.2 中央处理器、微处理器和微控制器	2
1.1.3 微型计算机系统的组成	3
1.2 微型计算机的工作原理	4
1.2.1 指令和控制器的指令部件	4
1.2.2 CPU 内的寄存器	5
1.2.3 计算机的工作过程	5
1.3 微型计算机的特点、应用和发展方向	7
1.3.1 计算机的特点	7
1.3.2 计算机的应用	7
1.3.3 计算机的发展方向	8
1.4 计算机运算基础	9
1.4.1 进位计数制	9
1.4.2 进位计数制间的转换	10
1.4.3 二进制数中无符号数的运算	12
1.4.4 符号数的表示和运算	15
1.4.5 十进制数的二进制表示和运算	20
1.4.6 数的浮点表示及运算	21
1.4.7 逻辑变量的表示和运算	24
1.4.8 文字在计算机内的表示	24
习题与思考题	25

第 2 章 微处理器与系统结构

2.1 微处理器的基本结构	28
2.1.1 算术逻辑单元 ALU	28
2.1.2 控制与定时部件——控制器	29
2.1.3 总线与总线缓冲器	32
2.1.4 寄存器阵列	33

2.2 Intel 8086 微处理器	34
2.2.1 8086 的寄存器结构	35
2.2.2 8086 CPU 的功能结构	37
2.2.3 8086 CPU 的引脚及其功能	39
2.3 8086 中的标志寄存器和堆栈	44
2.3.1 标志寄存器	44
2.3.2 堆栈	46
2.4 8086 系统的组成	47
2.4.1 存储器组织与存储器分段	47
2.4.2 输入/输出结构	51
2.4.3 总线接口部件	51
2.4.4 8086 的两种组态	54
2.5 8086 系统时钟和总线周期	57
2.5.1 系统时钟	57
2.5.2 总线周期	60
2.6 80386 微处理器	62
2.6.1 80386 微处理器的主要特性	62
2.6.2 80386 内部基本结构	62
2.6.3 80386 内部寄存器	64
2.6.4 80386 处理器引脚信号	66
2.6.5 80386 工作模式	67
2.7 80486 微处理器	69
2.7.1 80486 内部结构	70
2.7.2 80486 CPU 的特点	70
2.7.3 80486 CPU 主要引脚信号	72
2.8 Pentium 处理器	73
2.8.1 Pentium 处理器的特点	73
2.8.2 Pentium 处理器内部框图与信号功能	74
2.8.3 80486 与 Pentium 总线之间的主要区别	77
习题与思考题	78

第3章 指令系统

3.1 指令格式与寻址方式	80
3.1.1 指令格式	80
3.1.2 寻址方式	82
3.1.3 指令的分类	86
3.2 数据传送类指令	87

3.2.1 通用数据传送指令·····	88
3.2.2 交换指令·····	89
3.2.3 堆栈操作指令·····	89
3.2.4 地址传送指令·····	90
3.2.5 累加器专用传送指令·····	91
3.2.6 标志寄存器传送指令·····	92
3.3 算术运算类指令·····	93
3.3.1 加法指令·····	93
3.3.2 减法指令·····	95
3.3.3 乘法指令·····	98
3.3.4 除法指令·····	99
3.3.5 比较指令·····	100
3.4 逻辑运算指令·····	102
3.4.1 单操作数逻辑指令·····	102
3.4.2 双操作数逻辑指令·····	104
3.5 字符串操作指令·····	106
3.6 程序控制类指令·····	108
3.6.1 调用、转移与返回指令·····	108
3.6.2 条件转移指令·····	109
3.6.3 循环指令·····	112
3.6.4 中断控制指令·····	112
3.7 CPU 控制指令·····	113
3.8 80X86/Pentium 指令系统·····	115
3.8.1 80286 新增指令·····	115
3.8.2 80386/80486 新增指令·····	118
3.8.3 Pentium 系列处理器新增指令·····	120
习题与思考题·····	122

第4章 汇编语言及其程序设计

4.1 8086 系列汇编语言格式·····	126
4.1.1 字符集·····	126
4.1.2 汇编语言语句·····	127
4.2 伪指令语句·····	133
4.2.1 符号定义语句·····	133
4.2.2 数据定义语句·····	134
4.2.3 段定义语句·····	135
4.2.4 过程定义语句·····	139

4.2.5	分组语句	140
4.2.6	结束语句	140
4.3	部分宏指令	141
4.3.1	宏定义和使用	141
4.3.2	简化 DOS 段定义	145
4.4	汇编语言的编程环境	146
4.4.1	源文件编辑	146
4.4.2	汇编和连接	147
4.4.3	运行和调试	149
4.5	DOS 与 BIOS 功能调用	150
4.5.1	什么是 DOS 与 BIOS 功能调用	150
4.5.2	DOS 功能调用方法	151
4.5.3	DOS 功能调用举例	152
4.5.4	BIOS 中断和功能调用	158
4.6	程序设计基本方法	160
4.6.1	顺序结构	160
4.6.2	分支结构	161
4.6.3	循环结构	163
4.6.4	子程序结构	167
4.7	宏汇编语言程序设计举例	175
4.7.1	应用程序类型	175
4.7.2	应用举例	176
	习题与思考题	185

第 5 章 存储器及存储器子系统

5.1	存储器概述	189
5.1.1	存储器的技术指标	189
5.1.2	存储器的分类	190
5.1.3	内存的基本组成	192
5.1.4	存储系统的层次结构	192
5.2	半导体静态存储器	194
5.2.1	SRAM 存储器	194
5.2.2	UV-EPROM 存储器	196
5.2.3	EEPROM 存储器	197
5.2.4	闪速存储器	198
5.3	动态 RAM 存储器	204
5.3.1	动态 RAM 的基本存储单元	204

5.3.2	DRAM 的管脚信号与读/写操作	204
5.3.3	DRAM 的刷新	205
5.3.4	DRAM 控制器	207
5.3.5	PC 机的 DRAM 存储器	207
5.4	存储器的接口设计	214
5.4.1	存储器的接口信号	214
5.4.2	存储器设计需要考虑的问题	215
5.4.3	存储器接口举例	215
5.5	高速缓冲存储器	218
5.5.1	cache 的工作原理	218
5.5.2	cache 地址映射和地址变换	219
5.5.3	替换算法	221
5.5.4	cache 的一致性问题——更新算法	222
5.6	虚拟存储器	222
5.6.1	虚存的概念	222
5.6.2	虚存的地址变换	224
5.6.3	虚存的替换算法	226
	习题与思考题	227

第 6 章 总线技术

6.1	概述	229
6.1.1	总线上的信息传送方式	229
6.1.2	总线的分类	230
6.1.3	总线的标准化与总线规范	232
6.1.4	总线的性能指标	233
6.2	总线判决和握手技术	234
6.2.1	总线的操作过程	234
6.2.2	总线使用权的分配	235
6.2.3	总线仲裁技术	235
6.2.4	总线传输握手技术	239
6.3	PC 系列微机的系统总线	242
6.3.1	ISA 总线	242
6.3.2	PCI 总线	251
6.4	微机常用系统总线	266
6.4.1	PC/104 总线	266
6.4.2	STD 总线	267
6.5	微机常用通信总线	267

6.5.1 ATA 系列总线接口	267
6.5.2 SCSI 总线	268
6.5.3 GP-IB/HP-IB/IEC-IB/IEEE 488 总线	270
6.5.4 USB 总线	271
习题与思考题	276

第7章 I/O 接口与中断技术

7.1 I/O 接口	278
7.1.1 I/O 接口的重要作用	278
7.1.2 I/O 接口的主要功能	279
7.1.3 I/O 端口的编址方式	280
7.1.4 CPU 与 I/O 接口之间传送信息的方式	281
7.1.5 I/O 端口的地址分配	285
7.1.6 I/O 端口地址译码	286
7.2 中断的基本原理	288
7.2.1 中断请求	288
7.2.2 中断判优	288
7.2.3 中断响应	291
7.2.4 中断处理	292
7.2.5 中断返回	292
7.3 8086/8088 的中断系统	292
7.3.1 8086/8088 的中断源	292
7.3.2 8086/8088 响应中断的过程	294
7.3.3 中断向量表与中断向量号	296
7.3.4 对中断请求 INTR 的响应时序	297
7.3.5 中断服务程序	297
7.4 可编程中断控制器 8259A	298
7.4.1 8259A 基本构成与引脚信号	298
7.4.2 8259A 的工作原理	300
7.4.3 8259A 编程方法	308
7.4.4 8259A 的初始化命令序列和各命令寄存器初始状态	312
7.5 IBM PC-XT/AT 中的外部中断逻辑	313
习题与思考题	315

第8章 接口技术

8.1 计时器	317
8.1.1 计时器的一般工作原理	318

8.1.2	计时器的启动(触发)方式	318
8.1.3	计时器在计算机中的应用	318
8.1.4	计时器功能的改进:输入捕获和输出比较	319
8.1.5	PWM 输出功能与直流马达转速控制	320
8.1.6	定时器芯片举例	320
8.2	并行传输及其接口	326
8.2.1	简单并行口	326
8.2.2	选通并行口及其联络	328
8.2.3	并行接口设计举例	329
8.2.4	通用并行接口芯片举例	331
8.2.5	三线联络举例——CENTRONICS 打印接口	337
8.2.6	IEEE-488 总线及其三线联络	337
8.3	DMA 传输和 DMA 控制器	339
8.3.1	DMA 传输及 DMA 控制器	339
8.3.2	DMA 控制器的分类	339
8.3.3	源口及目的口的构成	342
8.3.4	DMAC 的工作类型和 DMA 传输方式	342
8.3.5	DMA 请求的生成方式	343
8.3.6	DMAC 的时序控制能力	343
8.3.7	DMAC 的块链接能力	344
8.3.8	DMAC 的中断源	346
8.3.9	DMA 控制器芯片举例	346
8.4	串行传输及串行接口	352
8.4.1	串行传输的应用	352
8.4.2	串行传输需要解决的问题	352
8.4.3	串行传输的信息格式	353
8.4.4	串行接口标准	355
8.4.5	串行接口芯片举例	356
8.4.6	其他串行传输协议及其接口	364
8.5	模拟接口	364
8.5.1	DAC 原理	365
8.5.2	DAC 芯片举例	367
8.5.3	A/D 转换原理	368
8.5.4	ADC 芯片举例	370
8.5.5	模拟通道的技术指标	372
8.5.6	模拟接口的工艺问题	372
	习题与思考题	373

附录

附录 A 8086 指令对标志位的影响 376

附录 B ASCII 码表 377

附录 C 伪操作表 378

附录 D 中断向量地址表 381

附录 E DOS 软中断及系统功能调用 382

附录 F BIOS 功能调用 388

附录 G 调试程序(DEBUG) 393

参考文献..... 394

第 1 章 微型计算机的基础知识

本章主要介绍关于微型计算机的基本概念、组成、工作原理、特点以及计算机内的信息表示和运算。

1.1 微型计算机的基本组成

1.1.1 微型计算机的基本硬件构成

通常所说的计算机指的是“电子数字计算机”，目前使用的计算机几乎都是这种类型的计算机。计算机系统是由硬件(Hardware)和软件(Software)组成的。硬件指的是各种设备，而软件则是指使用设备的手段。

就硬件的结构而言，计算机由运算器、控制器、存储器、输入设备、输出设备和总线组成，如图 1.1 所示。

1. 运算器

运算器又称为算术/逻辑运算单元(ALU; Arithmetical/Logical Unit)，完成数据的算术和逻辑运算以及移位等操作，所以它是一个信息加工部件。由于计算机中以二进制表示数据，所以运算器采用二进制运算，如二进制的加、减、乘、除、与、或、非等运算。

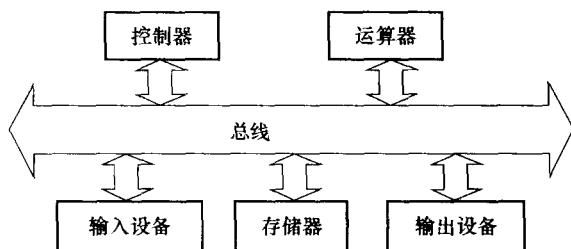


图 1.1 计算机的硬件组成

运算器包括一定数目的二进制位〔即比特(bit)，一般用小写字母 b 表示〕，该数目称为运算器的字长，一般为 8 位、16 位、32 位或 64 位。一个 8 位运算器每次可完成 8 位二进制数的运算，更多位数的运算可以分解成多次 8 位运算。字长是计算机的一个基本性能指标，字长越长，硬件电路也就越复杂，其运算速度也越快。

2. 存储器

存储器(Memory)是计算机的记忆部件。它存储控制计算机操作的命令信息(指令)和被处理(加工)的信息(数据)，也存储加工的中间结果和最终结果。这样，存储器内的信息分成两类：一类是命令信息，这类信息被计算机理解为命令，并能被计算机所执行，用它指挥计算机系统工作，以完成所要求的任务，这类信息被存放在存储器的代码区或程序区；另一类

是数据,是被处理的对象或者结果,这类信息被放在数据区。所有的数据和指令均以二进制数的形式存放在存储器中。

一般将存储器分为两级:内存和外存。平时将程序保存在外存,执行时将其调到内存中执行。目前使用的内存均是半导体存储器。狭义的存储器仅指内存存储器。

存储器的一个基本指标是存储器的容量,即存储器能保存的二进制信息的数量。通常将若干个二进制位组成一个存储单元(例如 8 比特构成一个存储单元)。存储器由多个存储单元构成,每个单元有一个编号,称为存储器的地址。向存储器送一个地址,可以将该地址对应的存储单元存储的所有二进制位数据读出,也可以向该存储单元写入数据。

存储器的容量一般表示为存储单元数 $\times$ 位数/单元,如 $1\text{K}\times 8$ 比特表示 1 024 个存储单元,每个单元 8 比特($1\text{K}=2^{10}=1\,024$, $1\text{M}=1\,024\text{K}$, $1\text{G}=1\,024\text{M}$)。

在计算机中,8 个比特的二进制序列称为一个字节(Byte,通常用大写字母 B 表示)。

3. 控制器

控制器(Control unit)是整个系统的指挥部件,它的任务是从内存中取出指令加以分析,然后发出控制信号执行某种操作。

每条指令可以完成一次算术或逻辑运算,或是存取数据的操作。能完成某种功能的一串指令的序列就称为程序。控制器根据程序指挥系统工作,以完成程序所规定的功能。

4. 输入设备和输出设备

输入/输出设备(I/O 设备;Input/Output equipment),也称为外围设备,其作用是进行信息形式的转换,也就是把外界的语言文字、声音、机械动作等信息形式转换成计算机能识别的电信号表示的二进制数的形式,或是进行相反方向的转换。

输入设备和输出设备是计算机与计算机外部交换信息的手段。通过输入设备,可以把给计算机的命令信息(程序)和被处理信息(数据)输入到计算机的存储器;通过输出设备,则可得到处理的结果,或者检查程序的毛病。

为使计算机对人们有用,它至少应有一台输入/输出设备。比如,计算器的键盘和显示屏,钟表的调节按钮和表盘,键盘和调节按钮是输入设备,其他是输出设备。

5. 总线

现代的计算机系统广泛采用总线(Bus)结构。总线是计算机各部件间传送信息的公共通路。各部件分时复用总线,以保证数据、地址、指令和控制信息在各部件之间的传送。

基本总线分为数据总线(DB;Data Bus)、地址总线(AB;Address Bus)和控制(命令)总线(CB;Command/Control Bus)。地址线和控制线的信息由总线控制器发出,表明程序(指令)所要访问的部件(存储器或外设)的地址和对该设备的操作性质(读出或写入),而数据总线则传送写入到该设备的信息,或从该设备读出的信息。

1.1.2 中央处理器、微处理器和微控制器

在计算机中,通常把运算器和控制器以及数量不等的寄存器作成一个独立部件,用一片 VLSI 实现,称为中央处理器,缩写为 CPU(Central Processing Unit),如图 1.2 所示。

微型计算机的中央处理器也称为微处理器 MPU(Micro-Processing Unit)或 μP 。

当把 μP 和存储器,以及 I/O 接口和总线在一个芯片中实现时,就成为微控制器(MCU: Micro-Controller Unit),也称单片机,其构成如图 1.3 所示。MCU 加上适当外部设备和相应软件即可构成一个微控制器系统。由于它价格便宜、软件控制灵活,在自动控制、仪器仪

表、通信设备、前端处理和家用电器、儿童玩具等嵌入式应用领域获得了广泛应用。

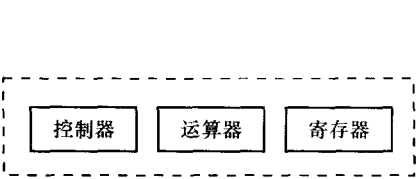


图 1.2 CPU/MPU 的构成

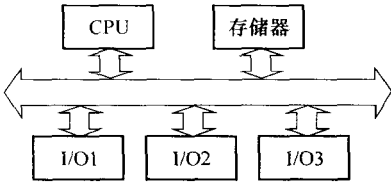


图 1.3 MCU 的构成

1.1.3 微型计算机系统的组成

微型计算机系统由硬件和软件两部分组成。

1. 硬件部分

硬件部分包括主机和外部设备。

(1) 主机包括 CPU、内存储器、I/O 接口、总线和电源。主机是能够完成数据处理功能的硬件，就是一般所指的微型计算机(Microcomputer, 缩写为 MC 或 μC)。

(2) 外部设备包括输入设备和输出设备。输入设备有键盘、鼠标器、光笔、扫描仪等，输出设备包括 CRT 显示器、打印机、绘图仪等。此外，硬盘驱动器、软盘驱动器、光盘驱动器等外存储器也属于外部设备。

2. 软件部分

软件部分包括系统软件和应用软件。

(1) 系统软件

系统软件包括操作系统、程序设计语言的编译程序和其他程序。

① 操作系统(OS: Operating System)是常驻内存的软件系统,包括系统资源管理(CPU 管理,存储器管理,I/O 管理和驱动程序)、任务管理、文件管理和程序库。它为使用者提供强有力的灵活的系统操作功能,使系统资源得到最充分而有效的利用。简单的计算机系统中,操作系统可能只是一个简单的监控程序(monitor)。

② 各种程序设计语言的编译系统为用户开发应用软件提供有力的支持。如汇编语言的汇编程序、各种高级语言的编译程序、连接程序以及各种程序调试工具。

③ 其他程序,如系统诊断程序、故障定位程序、系统配置程序等。

(2) 应用软件(或称用户软件)

应用软件是用户为实现给定的任务而编写或选购/订购的程序。它只适用于给定环境的给定用途,且一般驻留在外部存储器内,只在运行时才调入内存储器。

计算机的硬件和软件是相辅相成的,它们缺一不可。硬件是计算机工作的物质基础,而软件是计算机的灵魂。没有硬件,软件就失去了运行的基础和指挥对象;而没有软件,计算机就不能工作,其效能就不能充分发挥出来。

对一个具体的任务而言,一般既可以用硬件完成,也可以用软件完成。从理论上说,任何软件算法都能由硬件实现,反之亦然,这就是软件与硬件的逻辑等价性。设计计算机系统或是在现有的计算机系统上增加功能时,具体采用硬件还是软件实现,取决于价格、速度、可靠性等因素。

目前的情况是,随着超大规模集成电路的应用越来越广泛,以前由软件实现的功能现在更多地直接由硬件实现。而且在软件和硬件之间出现了所谓的固件(firmware):形式上类似硬件,但从功能上又像软件,可以编程和修改。这种趋势称为软件的硬化和固化。

1.2 微型计算机的工作原理

计算机的工作是在控制器的指挥下完成的,控制器是由存储在代码区的指令(程序)来指挥的,而程序是人编写的。最终,计算机按照人们的意志进行工作,完成人们希望它完成的任务。下面将通过一个例子来说明计算机是如何工作的。

1.2.1 指令和控制器的指令部件

1. 指令格式

指令是计算机能识别并执行的指示和命令,它采用二进编码来表示。每条指令应能表示该指令完成哪种操作以及该操作需要的数据保存在内存的哪个位置。故一条指令可分为两部分:操作码部分和操作数部分。操作码部分表示该指令所要完成的操作类型,如数据传送、加、减、乘、除等;操作数部分则给出操作数本身或者操作数在内存中的存放位置(地址)。

2. 控制器的指令部件

控制器的最基本任务就是取出一条指令(取指令),分析指令(指令译码)和执行指令;再取下条指令重复上述过程。为完成这种功能,它包含一个指令(执行)部件。该部件包括:

(1) 程序计数器(PC:Program Counter),它总是指向下条指令的首地址;

(2) 指令寄存器(IR:Instruction Register),保存当前正在执行的指令;

(3) 指令译码器(ID:Instruction Decoder),它对指令的操作码部分进行译码,分析该指令的功能;

(4) 操作控制器,它生成该指令操作所需的内部和外部微操作控制信号,指挥系统完成该指令的功能;

(5) 时序发生器,对各种操作实施时间上的控制。

指令部件的构成如图 1.4 所示。

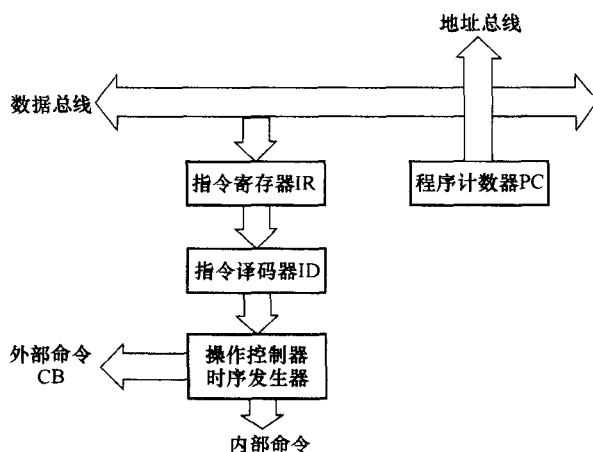


图 1.4 指令部件构成

1.2.2 CPU 内的寄存器

各种 CPU 内都包含数量不等的寄存器,它们可用于暂存数据,做存储器/IO 地址指针、计数器以及存放程序运行的各种状态,如程序状态字寄存器(PSW:Program Status Word)或简称状态寄存器(SR)或标志寄存器(FR)等。由于 CPU 内部寄存器的存取速度远比存储器快,所以,寄存器用于暂时存储程序重复使用的数据、变量和中间结果,可以大大提高程序的运行速度。而寄存器数量的多少也是衡量 CPU 功能强弱的重要指标之一。

通常,寄存器分为数据寄存器、地址寄存器和状态寄存器。

数据寄存器用于存放反复操作的数据、变量或中间结果。

地址寄存器存放存储器或 I/O 口地址指针,以缩短指令长度,加快指令运行速度。地址寄存器作为地址指针的另一个好处是能灵活修改,以便用循环程序处理大量数据的重复操作。

比如,要做两个数的加法,通常是把一个加数放在寄存器 A 内,另一个操作数放在其他寄存器内。经过运算器相加后,结果存放在寄存器 A 内。这样,A 寄存器就是累加器(Accumulator)。有的机器的存储器单元或 I/O 端口也可以作为累加器,这虽然增加了灵活性,但运行速度往往不如寄存器的累加器快。而状态寄存器存放指令(程序)的运行状态,如 CPU 工作状态(监控态或用户态)、存储器工作状态(或管理模式)、指令执行结果的状态(比如是否溢出等)、中断状态(如是否允许中断,以及当前的程序优先级等)以及其他状态。由于各计算机的应用场合和功能强弱不同,其状态寄存器的内容也不同。

1.2.3 计算机的工作过程

通过介绍计算 $y=a+b$ 的过程,说明计算机是如何工作的。

为完成这个计算任务,必须先做如下工作:

- 编写完成该任务的程序;
- 输入计算机;
- 编译或汇编,检错,纠错,编译/汇编通过后,变成“机器码”,放在内存的程序区;
- 运行程序;
- 输出结果。

下面是一个用汇编语言编写的简单例子:

a	db	56h	;被加数单元
b	db	24h	;加数单元
s	dw	1	;和单元
START:	mov	A,[a]	;取被加数
	add	A,[b]	;加加数
	mov	[s],A	;存和
	hlt		;停机
	end	START	;汇编结束

程序经输入设备输入计算机,再经汇编程序汇编后,若没有错误,就把它翻译成计算机能识别并执行的相应二进制编码指令(机器码),这些二进制机器码和符号化的汇编语言指