

高等学校规划教材

IBM PC 微机原理及接口技术

田艾平 魏廷德 李义杰 编

煤炭工业出版社

高等学校规划教材

IBM PC 微机原理及接口技术

田艾平 魏廷德 李义杰 编

煤炭工业出版社

(京)新登字 042 号

内 容 简 介

本书以 8088 为主体,系统地介绍了计算机基础、汇编语言、微机系统结构、接口技术以及 80286、80386/80486 的基础知识。为深入学习和使用高档微机打下基础。本书结构比较合理,使用取舍灵活,适应面较广。

本书为计算机应用专业本科教材,同时适用于通信、自动控制等非计算机专业本科生,亦可供有关工程技术人员参考。

高等学校规划教材 IBM PC 微机原理及接口技术

田艾平 魏廷德 李义杰 编

责任编辑:高 专

煤炭工业出版社 出版

(北京安定门外和平里北街 21 号)

煤炭工业出版社印刷厂 印刷

新华书店北京发行所 发行

开本 $787 \times 1092 \text{mm}^{1/16}$ 印张 $33^{1/4}$

字数 797 千字 印数 1—4, 415

1995 年 11 月第 1 版 1995 年 11 月第 1 次印刷

ISBN 7—5020—1179—X/TP32

书号 3945 A0320 定价 25.30 元

25H 是中断向量 NO: AL = 寄存器类型
DS: (DX) = 入口地址

35H 取中断向量 NO: AL = 寄存器类型
ES: (BX) = 入口地址

前言

随着计算机技术的迅速发展,微型机已广泛地应用于事务管理、数值计算、数据采集及处理、自动控制及通信工程等领域。为适应计算机应用日益发展、水平不断提高的形势,适应高等院校计算机教学的需要,编写了这本集 8088/8086 微机原理、接口技术、80386/80486 高档微机基础知识于一体的教材。

本书取材是从读者(计算机专业、非计算机专业本科生、工程技术人员、培训班学员等)需要出发。以 8088/8086 16 位微机为主体,详细地讲述了 8088 微机原理、汇编语言及接口技术;增加了 32 位高档微机的内容,讲述了 80286、80386/80486 的结构和功能特点;提供了 32 位微机开发、应用的基础知识。

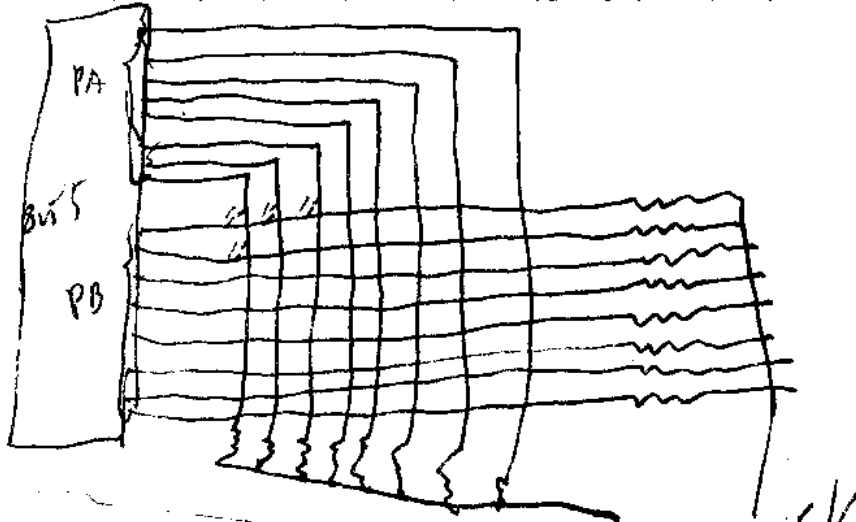
本书结构合理、取舍灵活、适应面广,全书分为上下两篇。上篇包括两部分:计算机基础知识、汇编语言,共六章;下篇包括三部分:微机系统结构、接口技术和 80286、80386/80486,共十三章。其中上篇 1、2、3 章为计算机基础知识,4、5、6 章为汇编语言;下篇 1、2、3、4、5、6 章为微机系统结构,7、8 为一般接口技术,9、10、11、12 章为专用接口技术,13 章为 80286/80486 高档微机基础知识。

计算机专业可选用汇编语言、微机系统结构、接口技术及高档微机部分;非计算机专业可根据需要选用计算机基础知识、汇编语言、接口技术中一般接口部分;根据教学大纲要求可选用不同方案以满足不同层次的教学要求。经适当取舍后,仍能形成连贯、相对完整的教材。

本书由田艾平任主编,并编写上篇 1、2、3、5 章、下篇 1、2、7、12、13 章;魏廷德编写下篇 3、4、5、6、8、9、10、11 章;李义杰编写上篇 4、6 章。

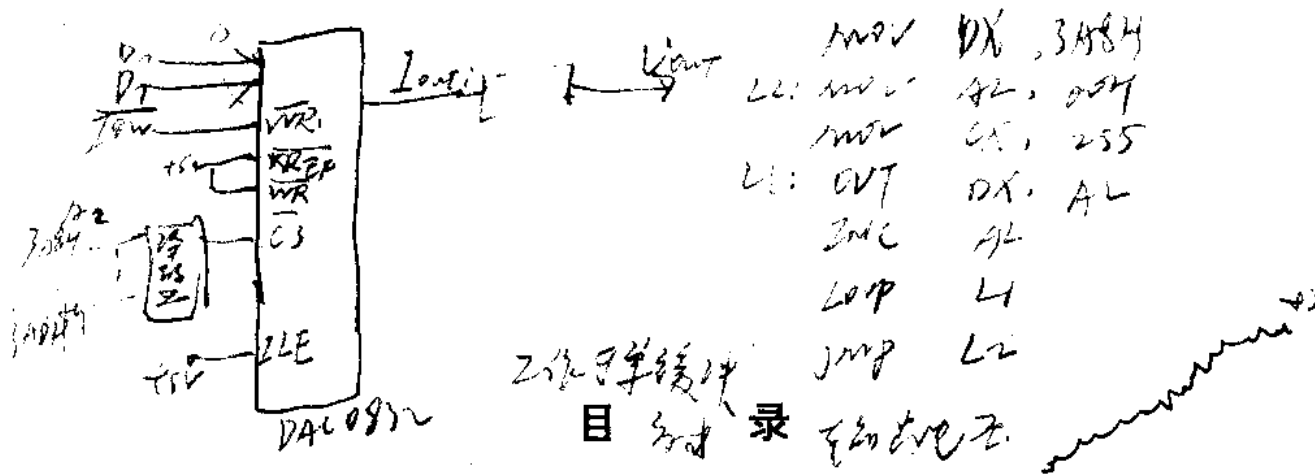
在编写过程中得到煤炭部科教司教材编审室及中国矿业大学、阜新矿业学院、西安矿业学院领导的关心和支持,在此一并致谢。

由于编者水平有限,不足之处在所难免,恳请读者批评指正。



编者

1994 年 3 月于西安



上篇 计算机基础知识及 8088 汇编语言程序设计

第一章 计算机基本组成概述	1
第一节 计算机系统组成	1
第二节 计算机系统的层次结构	25
第三节 计算机应用及系列机的概念	26
第二章 计算机中常用的数制和码制	27
第一节 计算机中的数制	27
第二节 计算机中常用的码制——原码、反码和补码	36
第三节 定点数和浮点数	43
第四节 二进制数的十进制编码——BCD 码	45
第五节 字母与字符的编码——ASCII 码	48
第六节 校验码——奇偶校验码	49
第三章 IBM PC/XT 系统结构	51
第一节 IBM PC 机的推出及微机发展趋势	51
第二节 IBM PC 机的基本硬件装置	52
第三节 IBM PC 机的软件系统	57
第四节 8088 CPU	59
第四章 8088 指令系统	69
第一节 8088 指令的寻址方式	69
第二节 8088 指令系统	73
第五章 汇编语言基本语法规则	107
第一节 概述	107
第二节 8088 汇编语言标记 (Token)	108
第三节 运算符 (Operators) 与表达式 (Expressions)	112
第四节 指令语句和伪指令语句	120
第五节 宏指令语句	131
第六章 汇编语言程序设计方法	145
第一节 汇编语言程序设计概述	145
第二节 顺序程序设计	148
第三节 分支程序设计	149
第四节 循环程序设计	154
第五节 子程序调用程序设计	163
第六节 综合程序设计举例	168

第一章 IBM PC 的 CPU 系统	174
第一节 8088 微处理器的引脚信号及其功能	174
第二节 最小和最大模式	178
第三节 8088 系统中 CPU 的外围电路	181
第四节 实例——IBM PC/XT 微机的控制核心	194
第二章 系统扩充槽信号及总线操作时序	198
第一节 扩充槽信号	198
第二节 8088 系统总线操作时序	200
第三章 存储器	209
第一节 存储器配置	209
第二节 半导体存储器	210
第三节 存储器与 CPU 的连接	214
第四节 磁盘存储器	219
第五节 虚拟存储器	222
第四章 中断系统	226
第一节 概述	226
第二节 8088 中断系统	230
第三节 中断控制电路	236
第五章 总线系统	237
第一节 RS-232C	237
第二节 STD	239
第六章 输入/输出	246
第一节 概述	246
第二节 输入输出接口	248
第三节 I/O 口地址译码	254
第七章 系统板上的中断控制器及 I/O 芯片	259
第一节 中断控制器 Intel 8259 A	260
第二节 8255A-5 可编程并行接口	276
第三节 8253-5 定时器/计数器	297
第四节 8237 DMA 控制器	314
第八章 异步串行通信接口	337
第一节 串行通信概述	337
第二节 8250	340
第三节 异步通信 I/O 功能程序	348
第九章 键盘、显示器接口	363
第一节 键盘	363
第二节 键盘电路与接口	363
第三节 BIOS 的键盘功能调用	366
第四节 显示器	367
第五节 CRT 显示原理和配置	367

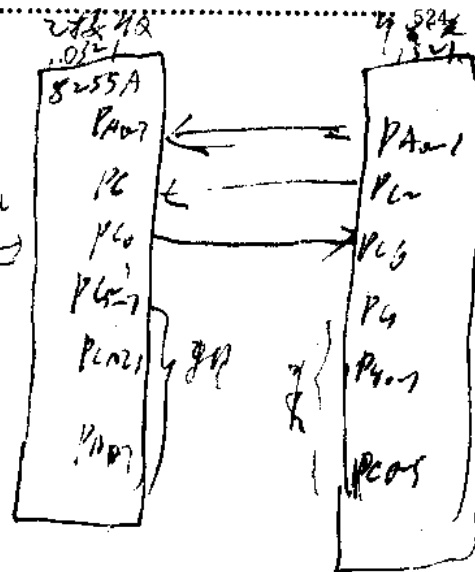
8255A 在系统内与板级 CPU 有并行及串行数据传送。单片发送，双片接收。7 机一板。
 的 8255A 采用 1 号接口。2 机一板与 8255A 采用 0 号方式工作。每板的 CPU 与
 口之间都采用直接查询方式交换数据。每板 1K 字节。

第六节 显示功能程序	376
第十章 打印机接口	383
第一节 TH3070 打印机	383
第二节 打印机适配器和 I/O 功能程序	392
第十一章 磁盘驱动器与接口	402
第一节 软磁盘驱动器	402
第二节 软磁盘适配器	405
第三节 BIOS 磁盘 I/O 功能调用	409
第四节 硬磁盘驱动器	414
第五节 硬磁盘适配器	419
第六节 磁盘 I/O 功能调用	423
第十二章 模拟输入/输出技术	434
第一节 概述	434
第二节 数/模 (D/A) 转换器	444
第三节 模/数 (A/D) 转换器	457
第十三章 微处理器 80286、80386、80486 功能结构简介	472
第一节 从 Intel 8088/8086 到 Intel 80286	473
第二节 Intel 80386/80486 的基本结构和功能	496

主要参考文献

8255A 接口程序
 MOV AX, 0304H
 MOV ES, AX
 MOV BX, 00H
 MOV CX, 3FFH
 MOV DX, 0FFFEH
 MOV AL, 10100000B
 OUT DX, AL
 L: MOV DX, 0FFFC0H
 IN AL, DX
 AND AL, 08H
 JZ L
 MOV DX, 0FFFB0H
 MOV AL, ES:[BX], AL
 OUT DX, AL
 INC BX
 DEC CX
 JNZ L
 HLT

2 机接口程序
 MOV AX, 040H
 MOV ES, AX
 MOV BX, 00H
 MOV CX, 3FFH
 MOV DX, 0FFFEH
 MOV AL, 10011000B
 OUT DX, AL
 MOV AL, 00000001B
 OUT DX, AL
 L: MOV DX, 0FFFC0H
 IN AL, DX
 AND AL, 10H
 JNZ L1
 MOV ES:[BX], AL
 MOV DX, 0FFFB0H
 MOV AL, 00000000B
 OUT DX, AL
 NOP
 MOV AL, 00000001B
 OUT DX, AL
 INC BX
 DEC CX
 JNZ L1
 HLT



第一章 计算机基本组成概述

本章首先介绍计算机系统的两大组成部分,即软件系统和硬件系统。简要介绍它们的基本组成、功能特点以及计算机最基本的工作原理。为后续章节提供一个关于计算机的初步概貌和常用的基本名词术语,然后给出几种最常见总线结构的计算机系统形式。最后讨论计算机系统的层次结构,使读者能初步从整体观点上了解计算机的工作原理。对于已学过《计算机组成原理》的读者,可跳过本章和下一章的内容直接从第三章开始学习。

第一节 计算机系统组成

一个完整的计算机系统是由硬件系统(Hard Ware)和软件系统(Soft ware)两大部分组成。硬件是指构成计算机的物理实体,包括了各种逻辑部件、接线、附属设备(如系统板、功能卡、显示器、打印机)等,硬件是计算机系统的物质基础。软件是指管理使用计算机的程序系统,它是发挥机器功能的关键。例如操作系统、语言的编译解释系统、数据库系统、软件开发工具、应用程序等等。所以对计算机系统而言,必须包括软硬件两部分。只有硬件而无软件的机器叫裸机,是无法进行任何操作的;无硬件的支持,软件的功能也无从谈起。因此我们了解计算机的工作原理必须从硬件、软件两方面入手。这一点对于微型机来说更为重要,因为微型计算机系统的软件硬件之间,相互结合、相互渗透更加密切。

一、计算机的基本硬件结构——冯·诺依曼机的基本结构

自从 1946 年第一台计算机问世以来,计算机取得了科学史上最惊人的发展速度,机器运行速度由每秒几千次达到几百亿次。功能相同的计算机的体积由占地上百平方米缩小到仅占几个平方厘米的半导体芯片上。尽管如此,当前计算机的基本组成原理仍遵循着五十年前美籍匈牙利数学家冯·诺依曼所提出的基本计算机的工作原理。也就是说,到目前为止,计算机的主流仍未突破冯·诺依曼机的范畴。冯·诺依曼机应具有以下几点基本功能:

- (1) 程序和数据能以机器可识别的二进制信息形式存入机器中;
- (2) 程序、数据可以在机器中长期记忆保留;
- (3) 可完成程序指定的各种算术运算、逻辑运算以及传送等数据处理功能;
- (4) 机器的工作应当在指令的控制下协调进行;
- (5) 程序运行的结果是可以输出的。

要完成以上的基本功能,计算机要有把程序和数据输入到机器中的输入设备(Input Device);记忆程序和数据的存储器(Memory);能完成数据加工、处理的运算器(Arithmetic Logic Unit);控制程序执行的控制器(Controller)和可将运算结果输出的输出设备(Output Device)。这就是组成冯·诺依曼机的五大基本功能部件如图 1-1 所示。该图表明了五大基本组成部件及它们之间的关系,该图中还包括有两股信息流,其中双线表示数据流,包括程序、原始数据、中间结果和最后结果;虚线表示控制信息流。现代计算机的信息流通常

是在一簇公共的通道中分时控制流通的，该通道叫总线（BUS）。

由上所述，可知冯·诺依曼机是一种快速计算工具，在统一的控制器作用下，它能接收数字化的输入信息，根据存放在存储器中的程序，对这些信息进行处理，产生结果信息并输出。简而言之，“存储程序控制”就是冯·诺依曼机的基本组成思想。

随着计算机技术和大规模集成电路的发展，使计算机的结构和功能发生了巨大的变化，利用大规模集成电路技术把计算机的运算器和控制器做在一块芯片上称做中央处理单元 CPU（Central Processing Unit）。CPU 和主存储器（CPU 直接打交道的存储器）称为主机。同时随着计算机的发展，输入/输出设备（总称外部设备）的数量和种类大大增加，为便于外部设备和主机的连接及用户的使用，外部设备要通过一种叫做接口的电路（Interface）和主机相连。因此计算机的基本硬件结构又可以表示为图 1-2。

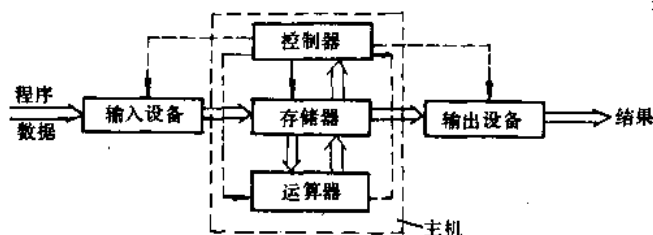


图 1-1 计算机基本组成框图

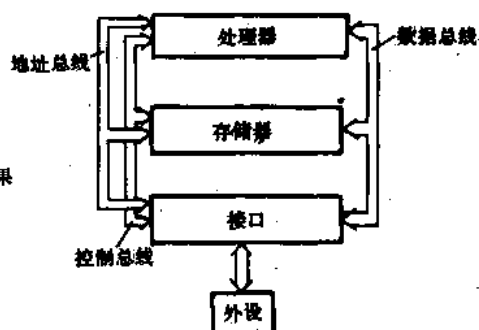
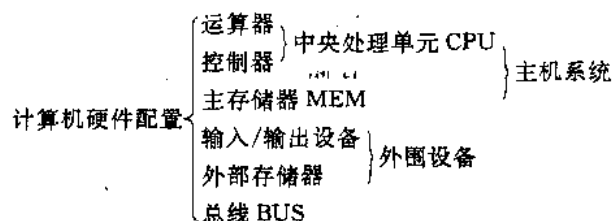


图 1-2 计算机基本结构框图

由此可见，计算机的基本硬件配置可以表示为：



实际上计算机还必须配有电源等辅助设备，而且现代计算机各部件密切结合在一起，很难机械地加以划分。但从原理上，计算机的硬件结构总可以看成由以上几部分构成。

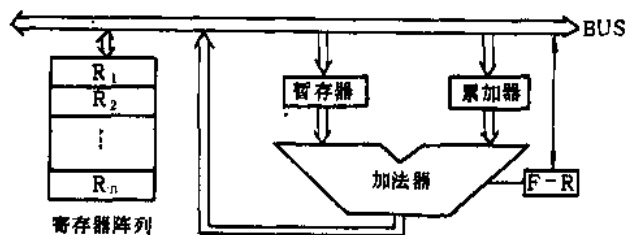


图 1-3 运算器结构框图

下面就按框图 1-2 的划分简要介绍计算机各组成部件的基本功能、结构和工作原理。

（一）运算器——ALU（Arithmetic Logic Unit）

运算器的基本功能是对信息进行加工处理，在控制器发出的各种控制命令作用下，可完成算

术运算、逻辑运算及数码移位等操作。运算器的简单结构如图 1-3 所示，包括加法器、内部寄存器阵列 (Registers) 及连接加法器和寄存器芯片的总线等。其中寄存器阵列一般应包括累加器、数据寄存器、暂存器和标志寄存器等。寄存器 (简称 R) 可提供一个操作数；累加器 (Accumulator Register 简称 AC) 是不仅能提供一个操作数而且可以保存中间结果的寄存器；标志寄存器是 (Flag Register 简称 FR) 记录 CPU 运行结果特征的特殊寄存器，其内容常称为程序状态字 PSW (Program Status Word)。

(二) 控制器 (Controller)

1. 控制器功能及特点

控制器是整个机器的指挥系统，它通过向机器的各部件发出控制信号来指挥整个机器自动协调的工作。控制器是通过程序来工作的，而程序是人们按着自己的解题意图、解题步骤事先编好，存放在内部存储器中，解题的每一个步骤就对应一条指令 (即操作的命

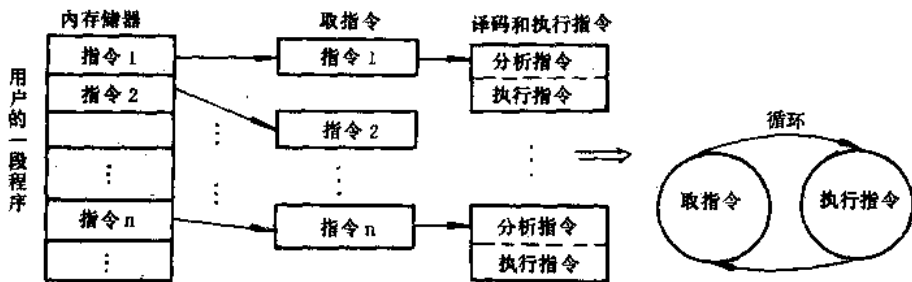


图 1-4 执行程序过程的示意图

令)，而一条条指令所构成的有序的集合就是程序。可以说控制器是按照能体现人的意图的程序来指挥机器工作的，而程序必须事先存放在内部存储器中，所以说控制器的主要特点是“存储程序控制”，这就是冯·诺依曼计算机的基本控制思想。由此可以得出结论：计算机自动工作的过程就是自动执行程序的过程，也就是自动完成取指令、分析指令和执行指令的循环过程，如图 1-4 所示。简而言之，控制器的功能是获取指令和解释指令的功能，即：从内存中取一条指令，并指出下一条指令的位置；对指令进行译码和测试；发出控制信号 (微操作命令)、控制数据流在各部件中的流动方向。

早期计算机取指令和执行指令是一串行的循环过程，现代计算机包括高档微型机采用流水线技术使取指令与执行指令并列进行 (Pipeline Operation)，如图 1-5 所示。

2. 指令及指令系统

由以上可知程序是为完成某一种具体任务而编写的指令序列，而指令是要计算机执行的某种操作的命令形式。例如加操作的指令：ADD AC, R1，其意义是将累加器 AC 与

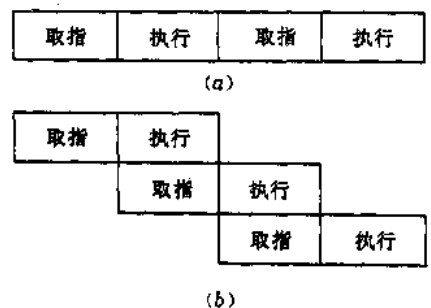


图 1-5 取指、执行指令的

并行操作示意图

a—指令的串行操作过程；

b—指令的并行操作过程

寄存器 R1 的内容相加, 其和送累加器 AC 保存。由此可以看出每一条指令必须包含两个基本内容: 操作码与操作数, 即指令的一般格式为:

操作码 OP	地址码 AD
--------	--------

操作码——指明操作的性质;

地址码——指明操作数的位置, 即操作数的地址或直接可操作的数。

例如加指令: ADD AX, BX, 其中 ADD 为操作码, AX、BX 为寄存器名, 指令完成的功能为 $(AX) + (BX) \rightarrow AX$, 即寄存器 AX、BX 的内容相加, 结果送 AX。操作数和操作码都用二进制数表示, 然后排列出来就构成一条指令的机器码形式称做机器码 (Machine Code), 每一条指令都有唯一的一个机器码与其一一对应, 例如:

机器指令	对应的机器码	功 能
INC AX	01000000B	累加器 AL 内容加 1
MOV AL, BL	10001010B 11000011B	寄存器 BL 内容送 AL

指令的数码是非常重要的, 使得程序和数据同样地存放在存储器中, 此外由于指令数码化使计算机对指令可按一定规律修改然后执行。这种由程序自己修改自己的能力, 增加了计算机的解题能力, 这是存储程序控制的重要特点。某一类机器的全部指令的集合称做该机器的指令系统, 通常不同类的机器其指令系统是不同的, 指令越丰富其机器的功能越强。

3. 控制器的基本结构

控制器的基本组成框图如图 1-6 所示, 可分为三部分: 指令部件、时序部件和微操作控制部件。

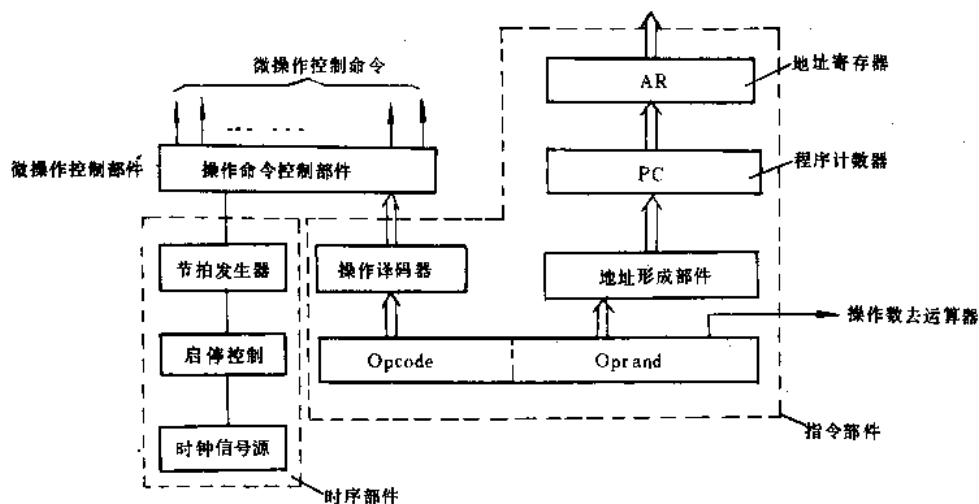


图 1-6 控制器原理图

1) 指令部件

(1) 指令寄存器 IR (Instruction Register) 存放从内存中取出来的指令, 也就是保留当前正在执行的指令, 一条指令执行完装入下一条指令。

(2) 地址形成部件及地址寄存器 AR (Address Register) 所谓指令地址或操作数地址是指指令或操作数存放在内存单元的位置编号。具体地址是根据指令的要求按不同的方法产生的, 对某一内存单元进行存数或取数操作的地址码将送到地址寄存器 AR 中。

(3) 指令计数器或称程序计数器 PC (Program Counter) 用来存放将要取出的下一条指令的地址。由于程序中的指令一般是顺序存放的, 执行时通常是顺序执行的, 因此每执行完一条指令后, 下一条指令的地址为 $(PC) + 1$, 其中 (PC) 为当前所执行的指令的地址。所以指令计数器 PC 实际为自动加 1 计数器, 每取出一条指令后, 自动完成 $(PC) + 1 \rightarrow PC$ 的操作, 形成后继指令地址。若程序执行过程发生跳转, 则转移的目标地址成为后继地址并送入 PC。

(4) 指令译码器 ID (Instruction Decoder) 将操作码转换为各种基本控制电位信号以控制指令的执行。例如从内存中取出一条指令的操作又应分解为以下步骤进行。

微操作	意义
$(PC) \rightarrow AR$	; 将指令地址送地址寄存器 AR
$(PC) + 1 \rightarrow PC$	; 形成下一条要取出的指令地址
$((AR)) \rightarrow IR$	; 按指令地址取指令送入指令寄存器 IR
$(IR) \rightarrow ID$	; 指令送译码器
$\vdots$	
$\vdots$	

以上这些微操作都是最基本的不能再分解的操作, 而且这些微操作必须严格地按照时间的顺序一拍一拍地进行。这就如同乐队演奏必须按指挥的拍子进行一样, 计算机中这种打拍子的部件就叫做时序部件。

2) 时序部件 时序部件是计算机内部的定时系统, 该部件用来形成指令执行过程中所必须的时间基准, 时钟脉冲、节拍电位、时标序列脉冲等定时信号。时序电路主要由时钟信号源 (通常为高稳定石英振荡源)、节拍发生器和启停控制电路组成。图 1-7 表明机器中定时的时序信号, 有指令周期、机器周期和状态周期。指令周期为取出并执行一条指令的时间, 不同指令, 指令周期的长短不同; 一个指令周期又包含若干机器周期, 又称 CPU 周期或总线周期, 一个机器周期通常取决于对存储器进行一次存或取操作的最短时间; 每

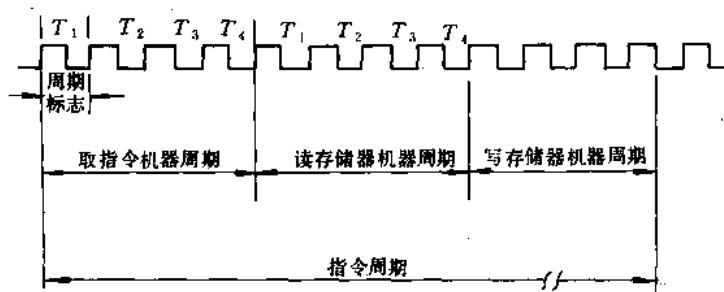


图 1-7 Z80 典型的时序图期

一机器周期由若干系统时钟周期组成，每一时钟周期称之为状态周期，它是机器状态延续的最小单位。

3) 微操作控制部件 微操作控制部件是控制器中最主要的部分，该部件根据指令译码器产生的电信号再与时序信号综合产生一系列微操作控制电平或脉冲信号以完成指令规定的全部操作如图 1-8 所示。

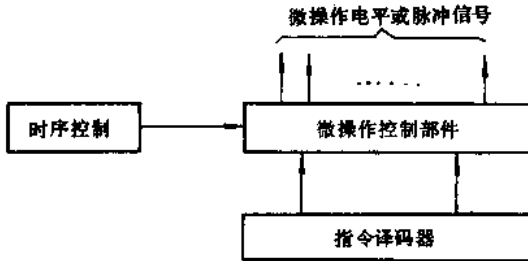


图 1-8 微操作控制部件功能

微操作控制部件由两种方式构成，一种是组合逻辑控制器，主要由硬件连接构成，通过组合逻辑实现，这是一种传统的控制器的设计方法，控制速度快，但结构不规整，设计、修改都比较复杂。另一种设计方式、即微程序控制器，微程序控制器是将指令系统中的每一条指令分解成若干条微操作，每一个微操作对应一条微命令，将这些微命令存入一个只读存储器 ROM (Read Only Memory) 中称控制存储器 CM。执行一条指令过程就相当于执行一段由微命令组成的微程序的过程，这一过程称取指令的解释过程，如图 1-9 所示。

控制存储器 CM 中，对应于每一条机器指令的微程序的入口由该机器的操作码和一定的判断条件所决定。这种每一条机器指令由一段微程序来实现的过程称解释过程，用微程序解释机器指令，意味着用软件技术实现硬件的功能。这种硬件软化，大大简化了控制器的设计，具有提高控制器的规整化、便于修改和扩充控制器功能等优点。所以微程序控制器已成为现代流行的控制器的设计方法。图 1-10 为微程序控制器的原理结构图。其中 μIR 是微指令寄存器；CM 为控制存储器和转移逻辑构成微程序控制部件的核心部件。除了传统的组合逻辑和微程序控制逻辑，随着门阵列器件 PLD 的发展，新型的阵列控制器，将是控制器的一个新的发展方向。

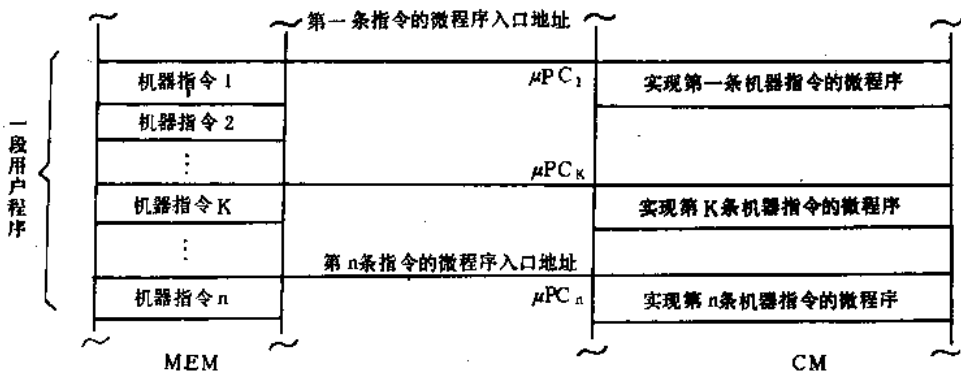


图 1-9 用微指令程序解释机器指令

以上简要介绍了运算器和控制器的功能和结构，运算器和控制器组成了中央控制单元 CPU。CPU 的主要功能可以概括为以下几点：即完成指令的控制以保证程序的顺序执行；完成操作的控制以保证机器各部件协调工作；进行时间控制以保证各部件动作的严格定时关

系和完成对信息的处理，以正确完成用户所提交的任务。

程序执行的一般过程如图 1-11 所示。

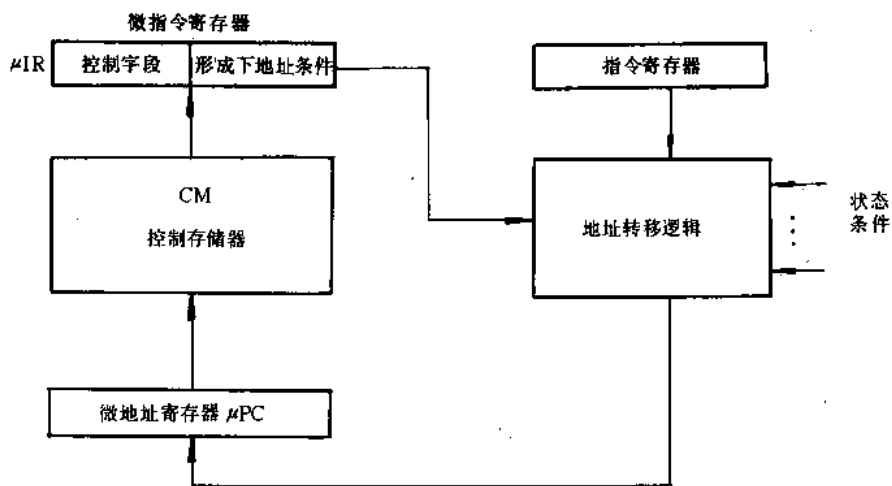


图 1-10 微程序控制器原理图

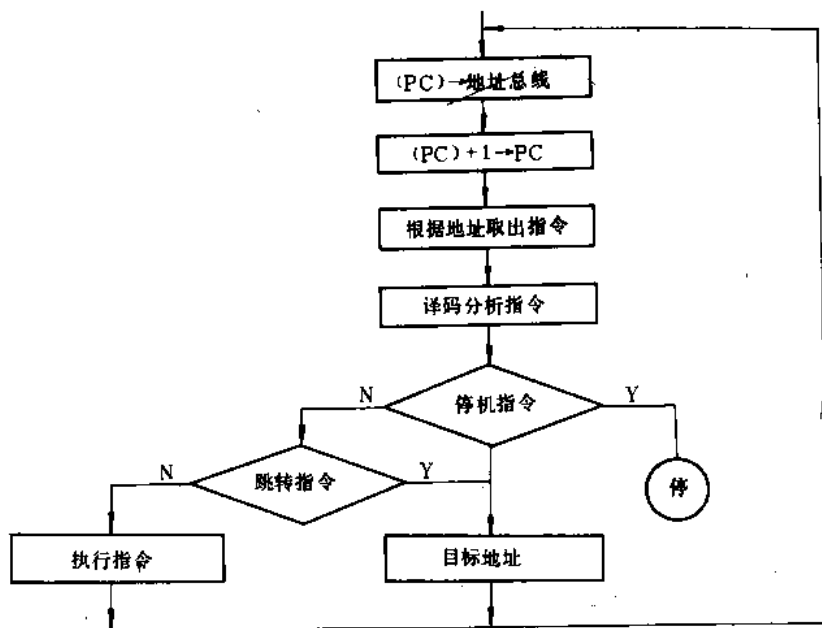


图 1-11 执行指令的一般流程图

（三）存储器（Memory）

存储器是计算机的记忆装置，用来存放二进制表示的程序和数据，因而存储器必须具备两个基本功能：一是数据和指令可以记录到存储器的指定位置上，称做存储器的写入；二是在需要时可以从指定位置上取出指令或数据，称做存储器的读出或取出。

储器的读/写操作又称为对存储器的访问。计算机所以能自动地连续工作正是依赖于存储器的这种保存和存取信息的功能。

1. 存储器的一般结构

存储器的基本结构如图 1-12 所示。一个主存储器系统应包括存储体、地址寄存器及地址译码器、数据缓冲器和读/写控制逻辑等部分组成。其中存储体是存储器的核心部件，其他属存储器的外围电路。

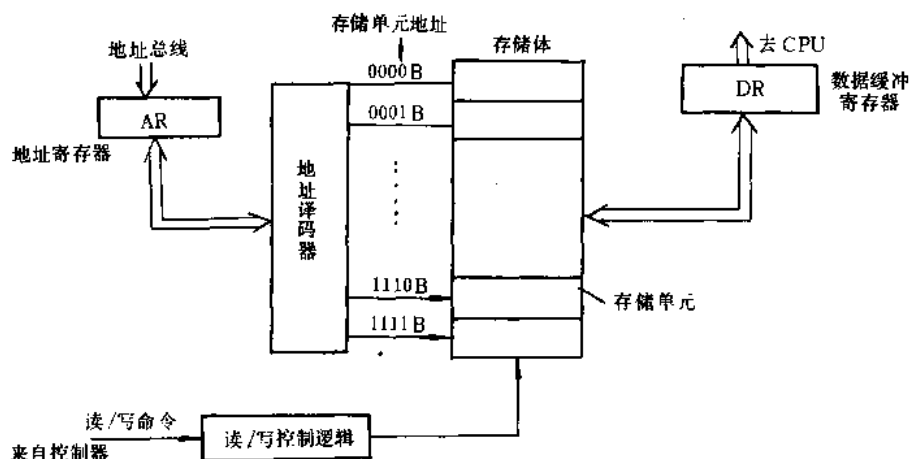


图 1-12 存储器的结构原理图

1) 存储体、存储元件、存储单元 存储器要存入大量的二进制信息，因此存储器是由许许多多基本存储元件（简称存储元）构成。每一个存储元可存储一位二进制信息即“0”或“1”。构成存储元的器件很多，只要具有两个稳定的物理状态且在一定条件下其状态可相互转化，这种器件就有可能构成一个存储元。例如，半导体存储器中每一位存储元可由一位双稳触发器构成。触发器的“导通”和“截止”状态分别记录了“0”或“1”信息，或 LSI-MOS 电路中栅漏电容“有”或“无”充电电荷的状态表示“0”或“1”。在磁盘、磁带等磁表面存储器中利用单元磁介质的正向磁化或反向磁化的剩磁状态分别表示“0”或“1”信息等等。“位”（bit）是二进制信息的最基本单位，存储元可存储一位二进制信息，是存储器的最小单位。但在计算机内对信息的处理不是一位一位的进行，而是一组一组信息的进行。这种具有固定位数的“信息组”称为一个机器字（简称字 Word），8 位二进制称 1 个字节（Byte）。不同类型机器规定机器字的长度是不同的。若机器字的长度为一个字节则称为 8 位机，字的长度为 2 个字节则称为 16 位机，4 个字节则称为 32 位机等。一位存储元件与一位二进制信息相对应。所有存储元的集合就构成了存储矩阵也就是存储体。存储体的结构根据存储元件的排列方式分为字存储体结构和位存储体结构两种结构方式。

字存储体是一根地址选择线可同时选中一个字中的所有位，也就是地址选择方法是采用一维地址译码方法。图 1-13a 表示的只是一个只有 16 个 8 位二进制数的半导体存储体结构，一共有 $16 \times 8 = 128$ 个存储元件，每一行由 8 个存储元件组成一个机器字，用一根 W 线把一个字节的 8 个元件连在一起，每一根地址线与一根 W 线相连，W 线也称字选择线，

一根字线也就选中该字中的所有位, n 位地址码经地址译码器产生 2^n 根地址选择线, 当存储器容量较大时, 地址译码器因译码线增多变得非常复杂, 因而这种方法只适用于容量小, 地址数较少的情况。

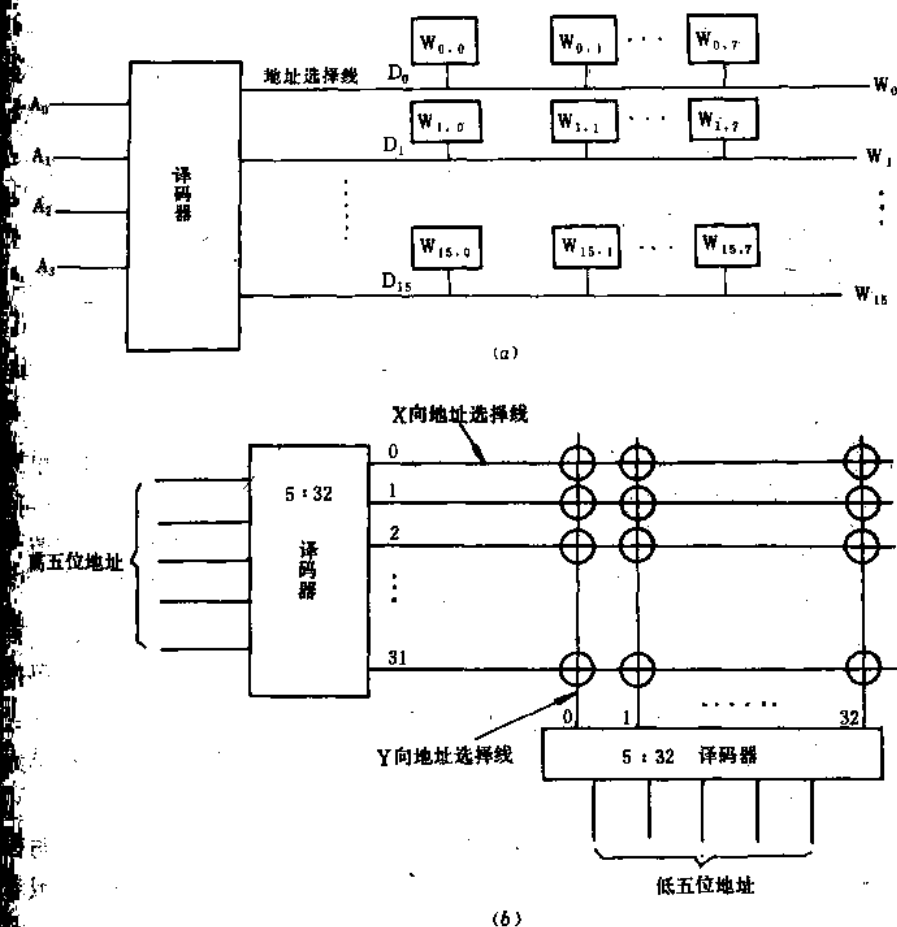


图 1-13 存储器结构示意图

a—单译码器的字存储体结构示意图; b—二维译码器的存储体结构示意图

这种存储体的地址选择采用二维地址译码方法。地址码分别送入 X 向、Y 向两个地址译码器, 只有 X 向、Y 向的地址选择线同时选中的交叉点上的存储元件才被选中。图 1-13b 所示的是容量为 1024 的位结构的寻址示意图。地址选择线的数目为 $2 \times 2^{n/2} = 64$ 根, 大大减少了地址线。这种存储体多采用位存储体结构。为便于信息的读/写操作, 存储体又分成若干存储单元, 每一存储单元由若干存储元构成。微型机系统通常规定一个存储单元的长度为一个字。

(2) 存储单元的地址和存储器容量 对于存储器的每一个存储单元给出一个唯一的编号, 这个编号通常是顺序排列的, 称做该存储单元的地址。存储器的读/写操作是按地址进行的, 因此“地址”是非常重要的概念。如图 1-14 所示。

存储器地址的编码范围取决于给定的二进制地址码的位数。例如，若地址码为 4 位，地址编码最低位为 0000B，最高位为 1111B，可编址的存储单元的数目为 $2^4=16$ 个，即存储器容量为 16 个 byte（即 16×8 位）。若地址码为 16 位，地址编码的范围为 0000H~FFFFH（16 进制的缩写形式）。存储器容量为 $2^{16}=64K$ （字节），（ $2^{10}=1024=1K$ ）所以若 n 为地址码的位数， P 为存储器容量，则 $P=2^n\text{byte}$ 。

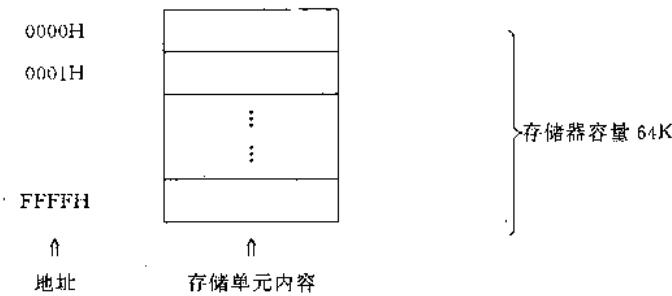


图 1-14 存储单元的地址和内容

应当注意，对于每一个存储单元，有二组二进制信息与其对应：一组为内存单元的地址码；另一组为存储单元的内容。这两组二进制信息的意义是完全不同的，绝对不可混淆。而且应注意到微型机系统基本是按字节编址的，即一个存储单元的长度为一个字节，每一个存储单元有一个地址码相对应。如图 1-14。

3) 地址寄存器与译码线路 地址寄存器与译码线路构成了存储器的寻址系统。首先将地址总线传送来的地址码加载到地址寄存器 AR 中，再经过译码电路选中由地址码指定的单元进行读/写操作。

4) 数据寄存器 (DR) 和读写控制电路 数据寄存器 DR 用来暂存当前从存储单元读出的内容或者 CPU 准备写入存储器的数据。

读/写控制电路和 CPU 的读/写命令再与存储器的时序配合，产生相应的读/写控制信号以正确地完成将数据缓冲寄存器的内容写入地址译码器所选中的存储器单元，或者把指定地址单元的内容读出送入数据缓冲器，再进一步送到目标地址。

由以上存储器的结构可以看出，存储器的工作特点是“按地址存取、以新代旧”，即根据 CPU 或指令提供的地址，对存储器的具体单元进行读出或写入操作，而且注意到读出一概不破坏原来存储单元的内容，当写入时，新写入的内容将取代原来的内容。

2. 主存储器的主要技术指标

存储器的主要技术指标是用来衡量存储器性能好坏，以满足用户对于存储器在容量、存储速度、信息的可靠性以及成本等方面的要求。主要包括以下几项：

1) 存储容量 存储器可容纳的二进制信息总量称为存储器的容量，通常存储器的容量是用所能存储的字数乘以字长表示为

$$\text{存储容量} = \text{字数} \times \text{字长}$$

例如一个存储器能存 4096 个字，字长为 16 位则存储容量可用 4096×16 表示。微机中存储单元的长度多选一个字节，即存储器按字节编址，所以存储容量常用存储器所占有的总的字节数来表示，而且规定 $2^{10}=1024=1K$ 。1024K=1M (1 兆)。若某存储器有 65536 个