

微型计算机入门

微型计算机入门

WEIXINGJISUANJI RUWEN

王玉龙 潘正运 编

社

36

1/1

社

社

新 时 代 出 版 社

微型计算机入门

王玉龙 潘正运 编

新时代出版社

内 容 简 介

本书以微计算机的初学者和使用者为对象,深入浅出地介绍了微计算机的各个组成部分。全书共九章。第一章从人们易于接受的直观概念出发介绍微计算机的最基本工作原理。第二章介绍了深入学习微计算机所必需的基础知识,如数制及逻辑电路。第三至七章介绍了微计算机的各部分,如微处理器、存储器、指令系统、汇编语言程序设计及接口等。尽管这些内容以 Intel 8080 为背景讲述,但都强调了基本概念及组成原理。第八章以一单板机为例说明一个简单微计算机系统的组成及工作原理。第九章简要介绍了 Intel 8086 16 位微处理器,它的指令系统及简单程序。

因此,本书既是一本微计算机的“入门”书,又是一本可供实用的参考书。它可用作微计算机短训班、文理科非计算机专业的微计算机原理课的教材,也可供具有高中文化程度的读者自学。

IS04/3/22

微 型 计 算 机 入 门

王玉龙 潘正冠 编

新时代出版社出版 新华书店北京发行所发行

国防工业出版社印刷厂印刷

787×1092毫米 16开本 19.75印张 458千字

1986年4月第1版 1986年4月北京第1次印刷

印数: 0001—7500 册

统一书号: 15241·87

定价: 4.05元

前 言

自1971年美国Intel公司生产出第一片微处理器及其相应的微计算机MCS4以来,仅十余年,微计算机已取得惊人的发展。就微计算机的核心部件——微处理器而言,已经从字长为4位发展到字长为8位和16位,并且已经出现字长为32位的微处理器。4位微处理器一般采用PMOS工艺,其典型产品有Intel公司的4004和4040;8位微处理器一般采用NMOS工艺,其典型产品有Intel公司的8080, MOTROLA公司的6800和Zilog公司的Z80;16位微处理器采用的是高密度短沟道NMOS(即HMOS)工艺,其典型产品有Intel公司的8086, MOTROLA公司的68000和Zilog公司的Z8000;新近出现的32位微处理器,其性能已接近中型计算机,典型产品有Intel公司的IAPX432系列和Zilog公司的Z80000系列。尽管已经出现32位字长的微处理器,但8位及16位微处理器仍占据很重要的地位。这是因为这些微计算机的品种齐全,有相当成熟的配套软件和开发系统,而且应用相当普及。

本书是一本微计算机的“入门”书,其目的是使从未接触过计算机的读者,能直接掌握有关微计算机的基本知识。全书以Intel8080和8086为背景,深入浅出地讲述微计算机的基本工作原理和程序设计的基本方法。

本书共分九章。在第一章,我们从人的计算过程引出一般计算机的最基本工作原理,进而说明微计算机的基本组成及其工作原理。通过这一章的学习将使初学者建立一个微计算机的整体概念。第二章概括了深入了解微计算机内部结构所必须具备的基础知识。这一章是专门为没有学过数字逻辑电路的读者编写的,对于零星学过这些内容的读者也可起到系统复习的作用。在第三章,我们以Intel8080微处理器为例,说明微处理器的基本组成及工作原理。第四章概括了微计算机中所采用的各种半导体存储器,其重点在于说明存储器的基本结构及工作原理,并在此基础上简要介绍了8080系列中的各种存储器芯片。在第五章和第六章中,我们以Intel8080指令系统为例,介绍了微计算机指令的格式及种类,以及利用这些指令编制程序的基本方法,并给出了某些实例。在第七章,我们说明了接口的基本组成及一般工作原理,并在此基础上介绍了8080系列中的某些接口芯片。第八章简要介绍了TK80单板机及其监控程序,以使读者熟悉一个简单而完整的微计算机系统,并了解用程序管理微计算机的基本概念。第九章集中介绍了Intel8086微处理器的组成、指令系统及程序设计,读者在掌握了前面各章的内容后是不难学习本章内容的。

本书是在原“微计算机原理”讲义基础上改写而成的,该讲义曾多次用作微计算机学习班和计算机应用进修班的教材。在编写本书过程中,我们得到了中国人民解放军郑州电子工程学院及北京航空学院计算机科学系有关领导的支持和鼓励,并得到张永福、周保群、杨金根、曲善广和邵洪余等同志的大力帮助,在此对这些同志表示深切的感谢。限于编者水平,书中难免存在缺点和错误,恳切希望读者批评指正。

目 录

第一章 微计算机概述 1	3.1 概述49
1.1 电子计算机的基本组成	3.2 微处理器 (8080 A) 的组成50
原理 1	3.2.1 运算部分50
1.2 微计算机的基本结构 3	3.2.2 寄存器阵列及地址逻辑54
1.3 微计算机的简单工作原理 5	3.2.3 控制部分56
1.4 微计算机的软件10	3.2.4 总线及总线缓冲器60
1.5 微计算机的应用12	3.3 微处理器 (8080 A) 的工作原理61
第二章 基础知识14	3.3.1 8080 A 的指令格式61
2.1 微计算机中数的表示方法14	3.3.2 8080 A 的定时62
2.1.1 进位制数及其相互转换14	3.3.3 8080 A 中的指令执行过程63
2.1.2 二进制数的定点及浮点表示17	3.3.4 8080 A 的外部信号及其操作65
2.1.3 二进制数的原码、反码及补码表示19	3.3.5 8080 A 的状态转换69
2.1.4 编码22	3.4 时钟发生器 (8224) 及系统控制器 (8228)72
2.2 微计算机中的运算方法23	3.4.1 时钟发生器 (8224).....72
2.2.1 二进制数的四则运算24	3.4.2 系统控制器 (8228).....74
2.2.2 补码加减运算26	3.5 8080 微处理器的控制信号小结75
2.2.3 十进制运算29	第四章 存储器78
2.2.4 逻辑运算30	4.1 半导体存储器的组成及分类78
2.3 逻辑代数及逻辑电路31	4.1.1 存储器的组成框图78
2.3.1 逻辑代数的初步知识32	4.1.2 存储矩阵的基本结构79
2.3.2 门电路34	4.1.3 用现成芯片构成存储器81
2.3.3 正、负逻辑的基本概念39	4.1.4 半导体存储器的分类84
2.3.4 触发器40	4.2 只读存储器85
2.4 基本逻辑部件44	4.2.1 掩模式只读存储器 (ROM)85
2.4.1 全加器44	
2.4.2 译码器45	
2.4.3 多路转换器46	
2.4.4 寄存器47	
2.4.5 计数器47	
第三章 微处理器49	

4.2.2 可编程只读存储器 (PROM)	87	6.4 程序设计的举例	150
4.2.3 可擦可编程只读存储器 (EPROM)	88	6.4.1 定时程序	150
4.3 随机存取存储器(RAM)	91	6.4.2 数制转换程序	151
4.3.1 静态RAM	91	6.4.3 二进制运算程序	154
4.3.2 动态RAM	97	6.4.4 十进制运算程序	161
第五章 指令系统	103	第七章 外部设备的接口	164
5.1 微计算机指令的基本格式及 分类	103	7.1 外部设备及其接口概述	164
5.1.1 指令的基本格式	103	7.1.1 外部设备简介	164
5.1.2 指令的分类	104	7.1.2 接口的组成及其功能	170
5.2 微计算机指令的寻址方式	105	7.1.3 外部设备的管理方式	172
5.2.1 直接寻址	106	7.2 简单通用接口 (Intel 8212 芯片)	174
5.2.2 立即寻址	107	7.2.1 8212的组成	174
5.2.3 间接寻址	107	7.2.2 8212的工作原理	177
5.2.4 变址寻址及相对寻址	108	7.3 可编程通用并行接口 (Intel 8255芯片)	180
5.3 Intel 8080的指令系统	110	7.3.1 8255的组成	180
5.3.1 数据处理指令	110	7.3.2 8255的控制字格式	182
5.3.2 数据传送指令	119	7.3.3 8255的三种工作方式	185
5.3.3 程序控制指令	122	7.3.4 8255的应用举例	190
5.3.4 机器控制指令	127	7.4 可编程通信接口 (Intel 8251 芯片)	194
5.3.5 8080指令系统小结	128	7.4.1 串行通信中的几个 术语	194
第六章 程序设计基础	130	7.4.2 8251的组成	195
6.1 8080汇编语言	130	7.4.3 8251的工作原理	199
6.1.1 字符集	130	7.4.4 8251的应用举例	203
6.1.2 基本指令语句	131	7.5 中断原理及中断控制器 (Intel 8214芯片)	208
6.1.3 伪指令语句	133	7.5.1 中断的一般过程	208
6.1.4 宏指令语句	136	7.5.2 中断源的识别	210
6.2 程序设计的基本步骤	138	7.5.3 中断优先级的判别	213
6.2.1 熟悉所用的微计算机	138	7.5.4 8214中断控制器	216
6.2.2 编制源程序的基本 步骤	140	7.6 DMA原理及DMA控制器 (Intel 8257芯片)	222
6.2.3 源程序的汇编与调试	142	7.6.1 DMA控制器的一般 组成	223
6.3 程序设计的基本方法	143	7.6.2 DMA的一般工作过程	224
6.3.1 简单程序	143	7.6.3 8257DMA控制器	226
6.3.2 分支程序	144		
6.3.3 循环程序	145		
6.3.4 子程序	147		

第八章 TK80 单板微计算机简介.....235**8.1 TK80单板微计算机的系统****结构.....235****8.1.1 键盘输入及其工作****原理.....235****8.1.2 TK80 的存储器.....239****8.1.3 TK80 的显示电路.....242****8.1.4 TK80 的单步操作****电路.....245****8.2 TK80 的监控程序简介.....245****第九章 Intel8086微处理器259****9.1 8086微处理器及其系统****结构.....259****9.1.1 8086微处理器的内部****结构.....259****9.1.2 8086微计算机的简单系统****结构.....265****9.2 8086的指令系统.....271****9.2.1 8086的指令格式.....271****9.2.2 8086的寻址方式.....273****9.2.3 8086的指令系统.....274****9.3 8086的简单程序.....284****附录一 ASCII 码表289****附录二 8080指令的功能表290****附录三 TK80 监控程序清单 ...298**

第一章 微计算机概述

这一章将简要介绍微计算机（简称微型机）的基本组成及基本工作原理，其目的在于使读者掌握微计算机的某些基本概念及术语，并对微计算机的整体有一个初步了解，以便于学习以后各章的内容。在介绍微计算机之前，我们先从人的计算过程自然地引入电子计算机的一般组成原理。

1.1 电子计算机的基本组成原理

电子计算机就其基本工作原理而言，只是模拟了人的计算过程。因此，只要考察一下人是怎样进行计算的，就不难理解电子计算机的必要组成。

例如，要一个人用纸、笔和算盘计算一个简单题目： $(7 - 4) + (8 - 2) = ?$ 求解这个题目的步骤大致如图 1.1 所示。

第 1 步，将 7 与 4 在算盘上相减，并把减得的中间结果 3 记录在纸上。

第 2 步，将 8 与 2 在算盘上相减，并把减得的结果 6 保留在算盘上。

第 3 步，将 3 与 6 在算盘上相加，并把算得的最后结果 9 记录在纸上。

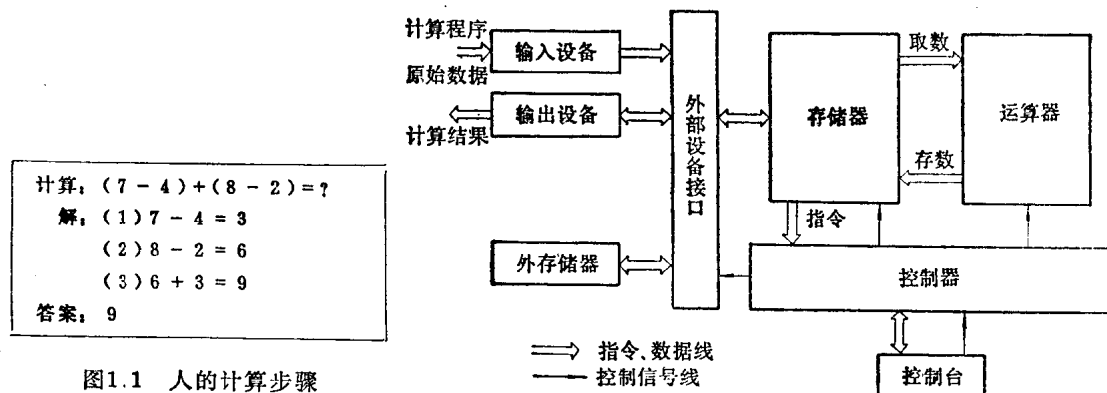


图1.1 人的计算步骤

图1.2 电子计算机的组成

分析上述计算过程就会发现，算盘是计算工具，纸是记录工具，而人的大脑和手则是控制工具。

电子计算机就是由具备这些功能的装置组成的：即运算器、存储器、控制器、控制台以及输入、输出设备及其接口（见图1.2）。

运算器是计算机中的运算工具，它能实现各种算术运算与逻辑运算。不过，计算机中所有运算的数不是常用的十进制数，而是二进制数。二进制数仅由 0、1 两个数字按“逢二进一”的规律组成。

存储器是计算机中的记录工具，它能储存算题中的原始数据、中间结果和最后结果，并能储存算题的计算步骤。存储器的结构可以用一幢大楼来比喻。一幢大楼是由许多房间组成的，每个房间都有一个房间号。存储器则是由许多“存储单元”组成的，每个存

存储单元也有一个编号，称为“地址”。图1.3给出了一个存储器的示意图，它有 256 个存储单元，单元的地址依次为 0, 1, 2, ..., 255。

控制器是计算机中的控制工具，它实现了对全机的自动控制，以使全机各部分协同动作。输入设备是向计算机输入原始数据和计算步骤的工具，输出设备则是从计算机输出计算结果的工具。外存储器是为了扩充计算机的记忆能力而设置的，它比内存存储器（即图1.2中的存储器，亦称主存储器）有更大的存储容量，但工作速度要慢得多。控制台可以实现人对计算机的干预，也可简单地检查机器的工作情况。

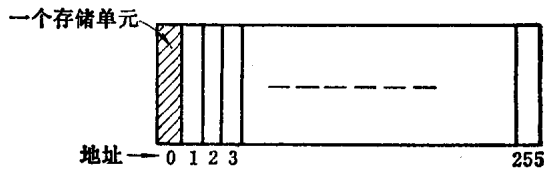


图1.3 存储器的示意图

通常，把输入设备、输出设备和外存储器统称为计算机的外部设备，把运算器和控制器统称为中央处理机（英文缩写为CPU），把运算器、控制器和内存存储器统称为主机。外部设备和主机之间的信息交换是通过图1.2中的“外部设备接口”这一部件实现的。

现在，我们进一步说明用计算机计算 $(7 - 4) + (8 - 2)$ 的大致过程。要在计算机上算题，必须先编制好计算该题的计算步骤，并通过输入设备将该计算步骤及原始数据送入计算机的存储器中。根据图1.2的计算机结构，我们将上述的人工计算步骤改为表1.1所示。表中的每一计算步骤给计算机下达了计算给定题目的一条命令，计算机按照表中的顺序逐条执行这些命令便可求得所需的结果。用计算机的术语讲，这一条条的解题命令称为指令，完成某一具体工作的一组指令的有序集合称为计算程序。表1.1就是求解 $(7 - 4) + (8 - 2)$ 的计算程序，它由9条指令组成。不过它是以文字形式编写的，故不能为计算机所记忆或识别。真正的计算程序是用二进制码表示的，它可储存在存储器中，并能为控制器所识别，进而可按指令的要求在运算器中完成相应的运算，这将在下一节中介绍。

表1.1 计算 $(7 - 4) + (8 - 2)$ 的程序(文字形式)

计算步骤	解 题 命 令
1	从存储器中取出 7 到运算器
2	从存储器中取出 4，并在运算器中与 7 相减，得差 3
3	将差 3 由运算器存入存储器
4	从存储器取出 8 到运算器
5	从存储器取出 2，并在运算器中与 8 相减，得差 6
6	从存储器取出中间结果 3，并在运算器中与 6 相加，得最后结果 9
7	将最后结果 9 存入存储器
8	从输出设备将最后结果 9 输出
9	停机

现在假定计算机能够储存并识别表1.1所示的程序，那么一旦将该程序送入计算机后，便可启动机器自动工作。计算机一开始计算，就从存储器取出第1条指令送入控制器，控制器对该指令进行解释，并发出“取数”操作所需要的控制信号，在取数信号控制下将原始数据 7 由存储器取至运算器，至此第1条指令执行完毕。接着，控制器又从存储器取出第2条指令，并对它进行解释，以发出完成“减法”操作所需要的控制信号，在这些信号控制下便可在运算器中完成 7 减 4 的操作，求得差 3，至此第2条指令执行

完毕。然后，控制器又从存储器取出第3条指令，等等。这样周而复始地取指令、解释指令、执行指令、再取下一条指令，直到执行完第9条“停机”指令，计算机才停止工作。注意，该题的计算结果已在执行第8条指令时从输出设备输出。

由上述可知，计算机的自动计算过程就是执行一段程序的过程，而执行程序的过程就是依次执行一条条指令的过程，执行一条指令的过程又可分解为取指令、解释指令、完成指令所规定的操作，并为取出下一条指令作好准备等步骤。计算机的这一工作方式是建立在两个基本技术基础上的，这就是“程序存储”和“二进制”。具体地说，要在计算机上算题，必须先编制程序，并把用二进制表示的指令与数据送入存储器存放起来，才能启动计算机工作。计算机只能按照人们预先确定的程序不折不扣地一步一步工作，从这一点讲，计算机只是以最原始的方式模拟了人的计算步骤。

图1.2中所介绍的仅仅是计算机的“硬件”部分。所谓“硬件”，是指那些看得见、摸得着的构成计算机本身的器件及电源等辅助设备。作为计算机另一方面的重要部分，就是计算机的“软件”。所谓“软件”，是指专门为计算机配置的一套“程序系统”，借助于软件可充分发挥硬件的效能。因此，讲到计算机的组成时，一定要包括“硬件”与“软件”两个方面。一般说，计算机的硬件是基础，没有硬件谈不上有什么计算机。然而只有硬件而没有丰富的软件的计算机，决不是一台性能良好的计算机，甚至可以说，没有软件的计算机是一台“死机器”。

1.2 微计算机的基本结构

大规模集成电路的出现与发展，使得可以把计算机的一些硬件部分制作在数片或一片大规模集成电路芯片上，从而出现了微计算机。微计算机的组成原理与一般计算机没有本质的差别。只是由于受大规模集成电路工艺条件的限制（如芯片的集成度、封装管的引脚数目等），使计算机微缩为微计算机时，在逻辑功能的划分、各部分之间的连接方式等基本结构上具有某些特点而已。

微计算机的典型结构如图1.4所示，它由微处理器（MPU）、随机存取存储器（RAM）、只读存储器（ROM）、输入/输出接口（I/O接口）及系统总线等部分组成。

1. 微处理器

微处理器就是上一节提到的中央处理机，它包括计算机的运算器和控制器两部分，并制作在一片（或几片）大规模集成电路芯片上。微处理器常用英文缩写MPU或CPU表示，其基本组成如图1.5所示。由图可知，微处理器至少包括算术逻辑部件（ALU）、累加器（ACC）[●]、数据寄存器（DR）、指令寄存器（IR）、程序计数器（PC）、地址寄存器（AR）、指令译码器及控制部件等。

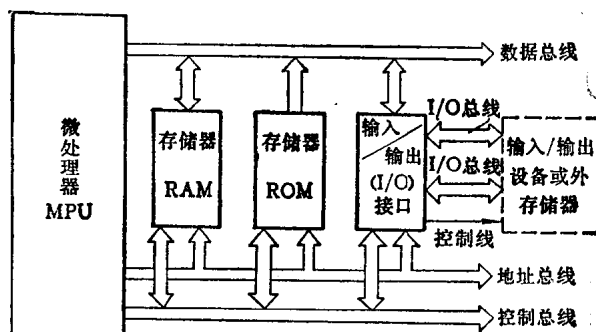


图1.4 典型的微计算机结构

[●]为简单起见，以下章节中ACC有时简写成A。

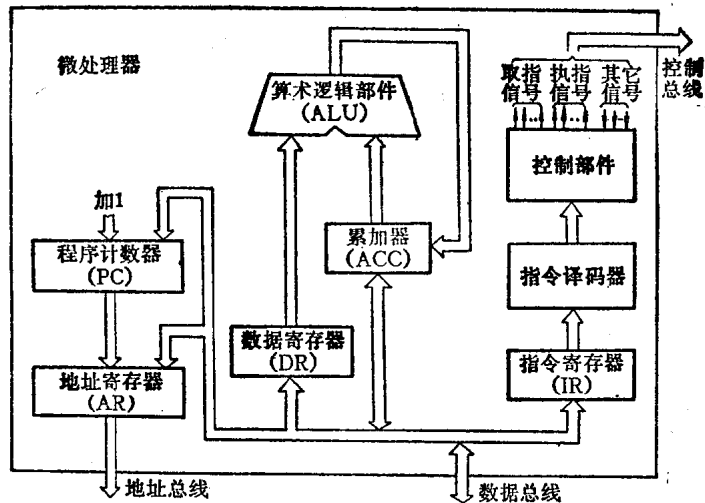


图1.5 微处理器的基本组成

累加器是一种具有双重功能的寄存器。运算前，它保存一个操作数，运算后，它保存运算结果。

数据寄存器用来保存一个数据，并作为算术逻辑部件的一个输入。

指令寄存器用来保存一条指令，以送入指令译码器进行解释。

程序计数器用来控制指令在程序中的执行顺序，它通常按递增的次序（0，1，2，3，…）指出要从存储器取出的指令。从下文可知，这个次序就是指令在存储器中的存放地址，也就是说程序计数器中保存了要取出的那条指令的地址。

地址寄存器用来保存正在执行的操作要用到的存储单元的地址，也就是说根据地址寄存器所提供的地址可以从存储器取出一个数据或向存储器写入一个数据。

指令译码器是对指令进行解释（或称译码）的一个部件。当指令从存储器取出并送入指令寄存器后，便由指令寄存器送入指令译码器，以确定该指令要实现什么操作，并“告诉”控制部件。

控制部件用来产生取指令、执行指令所需要的各种控制信号。对于不同的指令，取指令的操作顺序都是相同的，因而控制部件在取指令阶段将发出相同的一组信号序列。然而执行不同指令所需要的操作顺序是不同的，因此控制部件在执行指令阶段将发出不同的控制信号，以完成不同指令所规定的操作。

2. 存储器

微计算机的存储器是由大规模集成电路制成的半导体存储器，通常分为只读存储器（ROM）和随机存取存储器（RAM）两类。只读存储器只能读出信息而不能写入信息，它的信息是在制造芯片时写入的，即使掉电信息也不会丢失。在微计算机中只读存储器用来存放固定不变的程序。随机存取存储器既可读出信息又可写入信息，故又称为随机读/写存储器。在掉电后，随机存取存储器中的信息将丢失，故在微计算机中用来存放数据或可变的程序。

3. 输入/输出接口

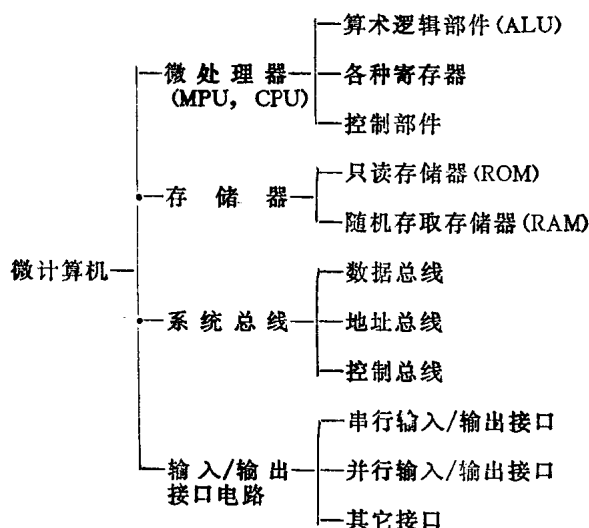
输入/输出接口用来实现主机（微处理器和存储器）与外部设备的连接，解决主机

与外部设备之间的信号同步和信号格式变换等问题。这里,外部设备同样包括输入设备、输出设备及外存储器。

4. 总线

所谓总线就是各部件之间公用的一组信息传输线,它可以把信息从一个或多个信号源传送到一个或多个目的地。微计算机中的总线通常分为三组,即数据总线、地址总线与控制总线。数据总线用来在各部件之间传送数据,地址总线用作微处理器向存储器或输入/输出接口发送地址,控制总线用来在各部件之间传送控制信号。

为使读者清楚地区分微处理器与微计算机两个不同的概念,我们把它们的关系归纳如下:



微计算机有单片机、单板机、多板机及位片机之分。所谓单片机,就是把构成微计算机的微处理器、存储器和输入/输出接口都做在一片大规模集成电路芯片上。如果把构成微计算机的上述三部分做成若干片芯片,并放在一块或多块印制电路板上,便是单板机或多板机,在这种微计算机中微处理器只占一片。位片机则是微处理器本身由多个芯片构成的微计算机,其中微处理器按位划分,一般做成2位或4位一片。

一台(或一片)微计算机,再加上外部设备、电源及系统软件,便构成一个微计算机系统。

1.3 微计算机的简单工作原理

前已指出,微计算机是微缩在大规模集成电路芯片上的计算机,因而它的基本工作原理与一般计算机毫无差别。这一节将仍以1.1节所列举的算题为例,结合微计算机(图1.4)及微处理器(图1.5)框图,更具体地说明微计算机是怎样工作的。

表1.1所示的计算 $(7-4)+(8-2)$ 的文字形式的程序,是不能为计算机所存储或识别的,那么怎样把它改造为计算机能存储和识别的形式呢?分析表1.1不难看出,每一条指令无非是向计算机提供两个信息:一是执行什么操作,另一是参与这一操作的数(即操作数)是什么。例如,表1.1中的第一条指令表明,指令执行的是“取数”操作,而取出的数是“7”。因此,我们可以把表1.1改写为表1.2形式。

表1.2 表1.1的改写形式

指令顺序	指令内容	
	执行的操作	操作数
1	取数	7
2	减法	4
3	存数	中间结果(3)
4	取数	8
5	减法	2
6	加法	3
7	存数	最后结果(9)
8	输出	9
9	停机	

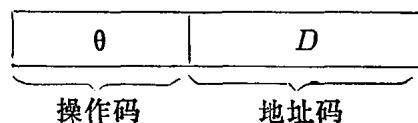
表1.3 操作码表

操作名称	操作码
取数	01001110
存数	01110001
加法	10000001
减法	11000001
输出	11010011
停机	01110110

表1.4 二进制数

十进制数	二进制数
0	00000000
1	00000001
2	00000010
3	00000011
4	00000100
5	00000101
6	00000110
7	00000111
8	00001000
9	00001001

在微计算机中,数和指令都是按一定的地址存放在存储器内的,假定表1.2中的操作数的存放地址如表1.5所示,则可得表1.6的计算程序。显然,该表中的每一条指令都是用16位二进制数表示的,前面8位用来表示操作,称为操作码;后面8位用来表示操作数的存放地址,称为地址码。可见,一条指令通常由操作码和地址码两部分组成,如下所示:



由于我们已经假定存储器的每个存储单元只能存放8位二进制数,因此要存放表1.6中的一条指令需用两个存储单元,前一个存放操作码,后一个存放地址码。设表1.6中的9条指令顺序地存放在地址为00011000~00101000(即十进制数24~40)的存储单元中,则存储器中数据和指令的存放情况便如图1.6所示。图中第17~22号单元中存放的是数据,称数据区;第24~40号单元中存放的是指令,称程序区。需强调指出的是,存储器中各存储单元的地址是由空间位置确定的,图中假定地址也用8位二进制数表示,则它们能表示的最小地址为00000000(即十进制数“0”),最大地址为11111111(即十进制数“255”)。存储单元的内容可以是数据,也可以是指令的操作码或地址码。

如前所述,计算机只能存储和识别由“0”或“1”组成的二进制数,因而必须把表1.2中的操作数表示为二进制数,并用不同的二进制码对各种操作进行编码。假定图1.4所示微计算机的字长为8位二进制,即它的存储器中的每个存储单元只能存放8位二进制数,且在微处理器中一次只能对8位二进制数进行运算。这样,我们就可用8位二进制码对表1.2中的各种操作进行编码(见表1.3),并用8位二进制数表示表1.2中的十进制数(见表1.4)。

表1.5 操作数的存放单元

数的存放地址	存放的数	说 明
00010001	00000111	7 4 8 2 } 原始数据
00010010	00000100	
00010011	00001000	
00010100	00000010	
00010101		中间结果
00010000		最后结果

表1.6 用二进制表示的计算程序

指令顺序	指 令 内 容	
	操作码	地址码
1	01001110	00010001
2	11000001	00010010
3	01110001	00010101
4	01001110	00010011
5	11000001	00010100
6	10000001	00010101
7	01110001	00010110
8	11010011	00010110
9	01110110	

现在我们将图 1.5 和图 1.6 连接在一起, 便构成图 1.7 所示的微计算机框图。下面以该图为基础说明微计算机的基本工作原理, 即微计算机是怎样完成取指令、分析指令和执行指令的。

要使微计算机进行计算, 必须先通过输入设备将计算程序和原始数据送入存储器, 并通过控制台(或控制面板)将存放第一条指令的地址(00011000)打入程序计数器, 也就是使 $(PC)=00011000$ 。作好上述准备工作后, 便可启动微计算机自动工作。

如前所述, 微计算机的自动计算过程就是周而复始地取指令、分析指令和执行指令的过程。对于第一条指令而言, 该过程的具体步骤如下:

第 1 步, 取出第一条指令的操作码, 并对它进行译码。在控制部件所发信号的控制下, 完成下列动作:

(1) 程序计数器的内容经①线送入地址寄存器, 记为 $(PC) \xrightarrow{\text{①}} AR$ 。此时, $(AR)=00011000$ 。

(2) 地址寄存器的内容经地址总线②送入存储器, 记为 $(AR) \xrightarrow{\text{②}} \text{存储器}$ 。因为 $(AR)=00011000$, 故选中存储器中地址为“00011000”的存储单元, 并将其内容(01001110)读出, 经数据总线③送入指令寄存器, 使 $(IR)=01001110$ 。在这一步中, 送入存储器的是第一条指令的地址, 从存储器读出的是第一条指令的操作码。所经过的途径是

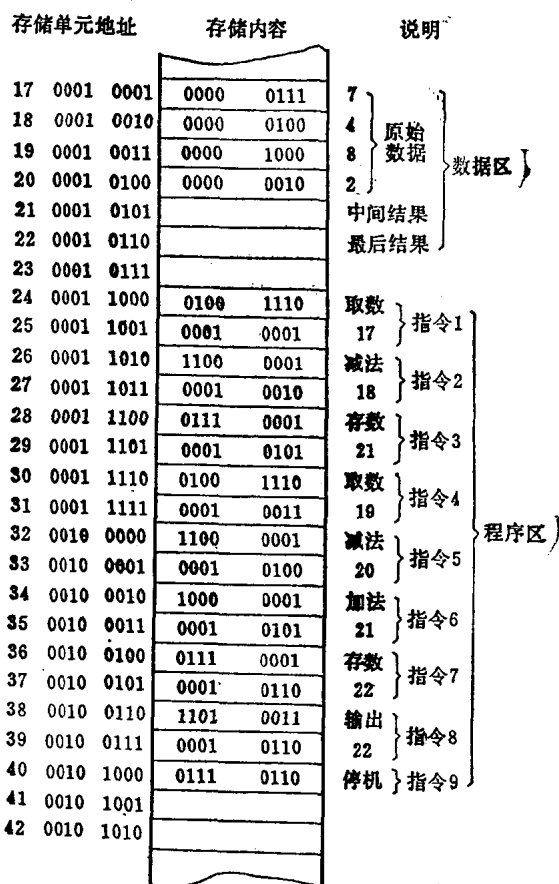


图1.6 存储器的布局

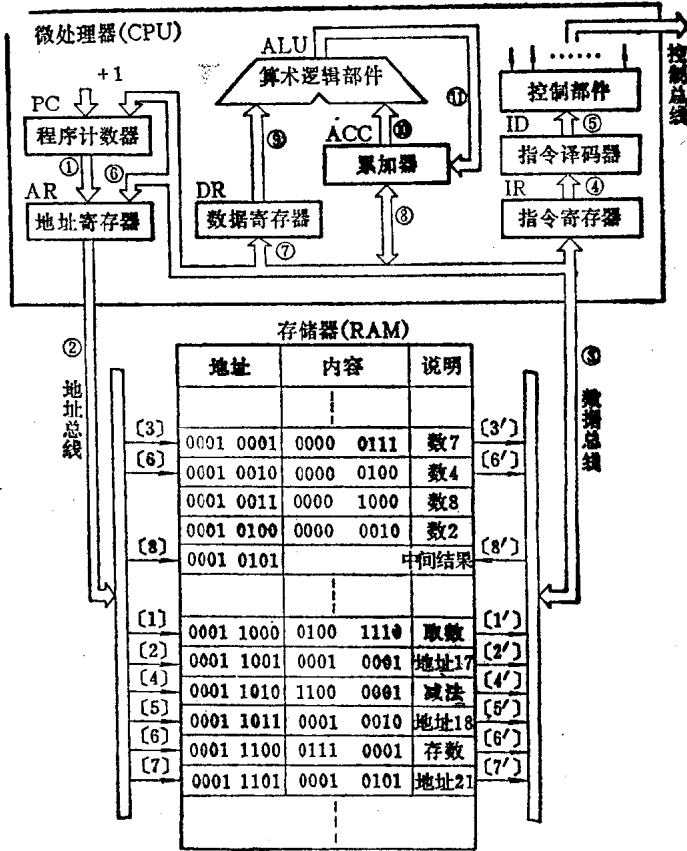


图1.7 微计算机的工作原理框图

(AR) $\xrightarrow{(2)}$ [1] $\xrightarrow{(3)}$ [1'] $\rightarrow$ IR

(3) 指令寄存器的内容经④线送入指令译码器进行译码，记为 (IR) $\xrightarrow{(4)}$ ID。因为 (IR)=01001110，译码器便译出“取数”命令，经⑤线通知控制部件，使控制部件进一步发出完成取数操作所需要的控制信号。至此，第1步的工作全部结束。为清楚起见，我们把这一步的工作过程归纳为图 1.8 所示。

第2步，取出第一条指令的操作数地址。在完成第1步工作后，控制部件使程序计数器加1，即使 (PC)=00011000+1=00011001。第2步的工作过程与第1步相似。不同的是，这一步是从地址为“00011001”的存储单元中取出操作数的地址(00010001)，并把这个地址送入地址寄存器，使 (AR)=00010001 (见图 1.9)。

第3步，执行“取数”操作。在完成第2步工作后，地址寄存器的内容为00010001 (即 17)。本步将完成下列动作 (参见图 1.7)：

(1) 将地址寄存器的内容 (00010001) 送入地址总线②。

(2) 从地址为 00010001 的存储单元中取出操作数 00000111 (7)，送入数据总线③。

(3) 将数据总线上的数据送入数据寄存器，使 (DR)=00000111。

(4) 控制部件发出信号使程序计数器的内容加1，即 (PC)=00011010，为取下一条指令准备好地址。上述过程可用图 1.10 表示。

至此，第一条指令执行完毕。微计算机自动进入取第二条指令，并执行第二条指令，

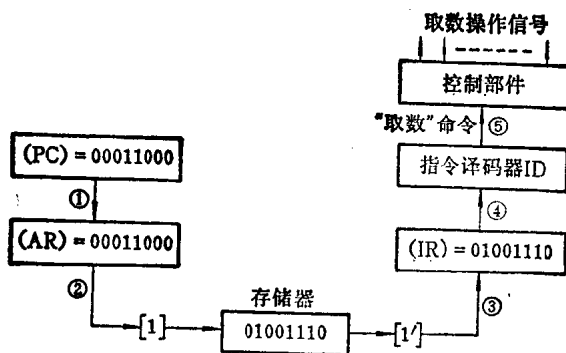


图1.8 取第一条指令的操作码

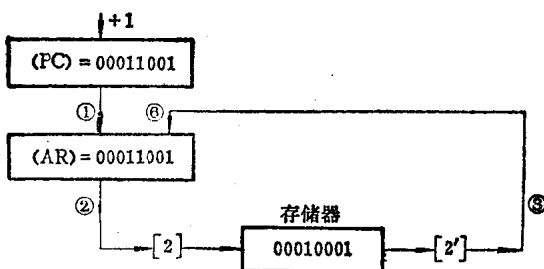


图1.9 取第一条指令的操作数地址

其具体步骤如下:

第1步, 取出第二条指令的操作码, 并对它进行译码。这一步的工作过程与第一条指令的第1步相似, 可用图 1.11 表示。只是取出的操作码为 11000001, 故指令译码器发出“减法”命令, 并由控制部件发出完成减法操作所需要的控制信号。

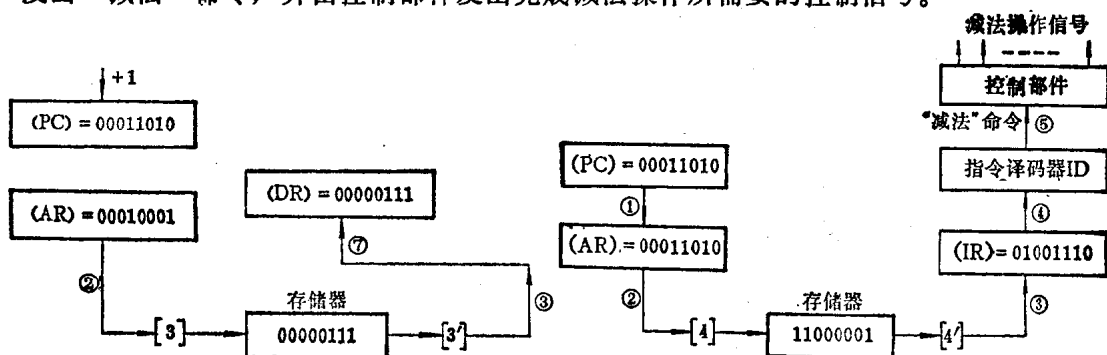


图1.10 取操作数 7

图1.11 取第二条指令的操作码

第2步, 取出第二条指令的操作数地址。本步的工作过程与第一条指令的第2步相似, 可用图 1.12 表示。完成这一步动作后, 地址寄存器的内容为操作数的地址 00010010 (见图 1.13)。

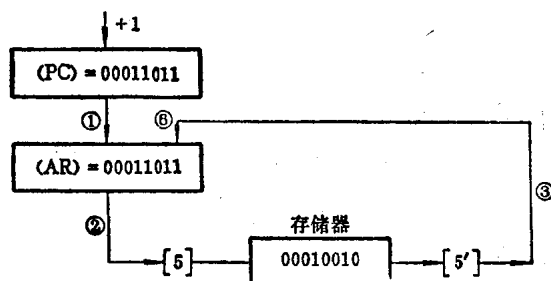


图1.12 取第二条指令的操作数地址

第3步, 执行“减法”操作。本步将完成下列动作 (参见图 1.7),

(1) 将地址寄存器的内容 (00010010) 经地址总线②送入存储器, 并从相应地址的存储单元中取出操作数 00000100 (4)。该数经数据总线③、⑥送入累加器, 使 (ACC) = 00000100。

(2) 将数据寄存器的内容 (00000111) 及累加器的内容 (00000100) 同时送入算

术逻辑部件,并在该部件内完成减法运算,结果回送到累加器。注意,此时原累加器中的内容被冲掉。

(3) 控制部件发出信号使程序计数器的内容加1,即 $(PC)=00011100$,为取下一条指令准备好地址。上述过程可用图 1.13 表示。

至此,第二条指令执行完毕,微计算机自动进入取第三条指令,并执行第三条指令。这样周而复始地进行下去,直到执行“停机”指令,微计算机便自动停止工作。

从上述例子可以看出,在微计算机中每执行一条指令,一般都要经历取指令操作码,取操作数地址及取操作数三个阶段,也就是需要“访问”存储器三次。在取出操作数后,再执行操作码所规定的操作,本条指令就执行

完毕。自动进入下一条指令的执行是通过程序计数器提供下一条指令的地址(例如程序计数器的内容自动加1)实现的,这就是微计算机自动计算的基本原理。

最后,读者应当注意区别执行一条指令过程中所用到的四个信息:指令的存放地址,指令的操作码,操作数的存放地址及操作数。

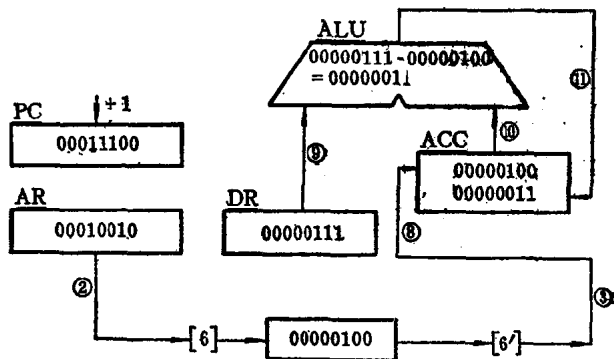


图1.13 执行减法操作

完毕。自动进入下一条指令的执行是通过程序计数器提供下一条指令的地址(例如程序计数器的内容自动加1)实现的,这就是微计算机自动计算的基本原理。

1.4 微计算机的软件

与一般计算机一样,微计算机也是由硬件和软件两部分组成的。只是由于微计算机的规模较小,所配置的软件没有一般计算机的庞大而已。微计算机的软件按用途可分为两类:系统软件和应用软件。系统软件通常包括语言处理程序、操作系统及服务程序,借助于这些软件可简化程序设计、简化微计算机的使用方法、提高工作效率。应用软件是微计算机应用于其它系统时,为解决该系统提出的任务而设计的程序。显然,应用领域不同,应用软件也就不相同。但还是可以研制一些通用程序供不同用户选用,从而避免不必要的重复工作,节省时间和费用。下面仅简要介绍微计算机的系统软件。

1. 语言处理程序

众所周知,人们之间是用“自然语言”交谈和交流信息的。那么,人和计算机之间怎样交换信息呢?这是通过计算机特有的“语言”实现的。计算机所用的语言主要有三种:机器语言、汇编语言和程序设计语言。

所谓机器语言,就是上一节提到的计算机的指令代码,它用二进制码表示,能为机器识别和执行。人们要计算机算一道题,必须用这种语言编制一段程序,计算机执行该程序后,把算得之结果告诉人,于是就完成了人与计算机之间的信息交流。显然,用机器语言编制程序要求人们熟记机器指令的二进制代码,这是很麻烦的。一般微计算机的指令数目可多达一百条以上,即使强制记住了,使用时还很容易出错,错了还难以修改。因此,使用机器语言编制程序不仅工作量大、效率低,而且所编程序只能用于某一特定的计算机,故目前已很少采用。