

微处理机的理论和应用

〔美〕 G. A. 斯特赖特马特 V. 菲奥雷 著

白英彩 译

科学出版社

1984

内 容 简 介

本书较全面地介绍了微处理机的原理、构造、应用和程序设计技术，并以8080和M6800为例着重介绍了这两种具有代表性的机种的的实际应用技术。书中对于研制微型计算机目标系统的开发系统也作了较详尽的论述。

书中主要内容有微处理机导论，二进制的基础知识，微处理机用的数字电路概念，指令系统，软件导论，总线控制，输入/输出，存贮器，高级软件，8080的体系结构及其配套电路，8080的应用，M6800的体系结构及其配套电路，M6800的应用和微处理机领域的展望等。通过阅读本书，读者对微处理机和微型计算机系统可以获得较全面的了解，打下较坚实的基础并提高实际运用微型计算机的能力。

本书可作为大专院校计算机专业及计算机培训班的教材，亦可作为科研、设计和工厂企业等部门的工程技术人员的自学参考书。

Gene A. Streitmatter Vito Fiore
MICROPROCESSORS THEORY AND APPLICATIONS
Reston Publishing Company, 1979

微处理机的理论和应用

【美】 G. A. 斯特赖特马特 V. 菲奥雷 著

白英爱 译

责任编辑 李淑兰 孙月湘

科学出版社出版

北京朝阳门内大街137号

中国科学院印刷厂印刷

新华书店北京发行所发行 各地新华书店经营

*

1984年5月第一版 开本: 787×1092 1/16
1984年5月第一次印刷 印张: 22
印数: 0001-19,300 字数: 505,000

统一书号: 15031·564

本社书号: 3514·15-8

定价: 3.40元

译者的话

如果说电子计算机的出现是科学技术领域中划时代的进步,那么也可以说,微处理机的出现又是计算机科学领域中划时代的进步。其实,作为电子计算机中的一支新秀——微处理机的诞生和迅速发展,将对整个社会产生巨大的影响,也对人类生活有着深远的意义。因此,有人把微处理机比拟为第二次工业革命。引起第一次工业革命的蒸汽机给人类以“动力”;而引起第二次工业革命的微处理机则给人类以“智能”。

自从1971年微处理机问世以来,在这十多年的发展历程中,其产品经历了两代更新,已进入了第三代,并出现了以32位为特征的第四代产品。它正以惊人的速度向前发展并迎接旺盛生长的“青年期”的到来。

由于微型计算机具有体积小、重量轻、价格低、可靠性高、功耗低、使用灵活和方便等优点,因而获得了极为广泛的应用。可以说,微处理机的发展历史就是其应用的历史。

近年来,国内在微型计算机应用方面也获得了迅速发展,这已成为当前我国计算技术的重要发展趋势。要适应这种发展趋势的需要,除了发挥专业计算机技术人员的骨干作用外,还应积极创造条件,使得那些凡是想要应用微型计算机的各行各业的人们都可以而且能够独立地胜任这一工作。因此,就有为数众多的非计算技术专业人员需要学习微型计算机知识的问题,主要是学习微型计算机的基本理论和应用技术。这本《微处理机的理论和应用》一书,就是为了满足此种需要而翻译的,俾供广大读者作为参考。

在微处理机方面,如果你还是个新手,则建议从头阅读本书;如果你已经有了初步基础或者你是个计算机专业人员,则可以快速地浏览第一至第九章,而仔细阅读第十至第十四章。可以帮助你解决应用有关Intel 8080和M6800(与国产的DJS-050和DJS-060系列微型计算机兼容)方面的实际问题。

本书内容丰富,在叙述上具有深入浅出、通俗易读、理论联系实际等特点,是一本较为实用的教学和自学参考书。

该书全部译稿承蒙刘寿和、顾良士和程杰等同志予以悉心校对,华南工学院计算机科学系副主任陈兴叶同志审读了该书译稿。他们曾为本书译稿提出了有益的意见和建议,在此表示感谢。由于译者水平有限,疏漏和欠妥之处在所难免,敬请读者批评指正。

白 英 彩

1982年8月

序 言

大规模集成电路技术的进展推动了称为微处理机的固态器件的发展。由于微处理机标志着集成电路技术领域里的一次跃进，而且有望成为未来电子学的核心，所以本书对从传统的控制和数据处理技术转向灵活的可编程控制器所需的知识作一介绍。本书对硬件和软件并重，不侧重于任一方面，把软件和硬件的应用结合起来。

第一至九章介绍适用于各种型号的微处理机的一般概念。先介绍基本原理，然后循序渐进地介绍到完整的系统。第十至十四章介绍有关莫托洛拉 MC6800 和英特尔 8080 微处理机系统的内容。由于英特尔 8080 和莫托洛拉 MC6800 的软件和硬件在目前的微处理机中比较典型，所以书中都以这两个机种为例进行说明。

第一章介绍微处理机的一般概念，使初学者熟悉微处理机方面的术语，获得一些基本概念。前后各章的内容都是紧密衔接的。例如，第五章（软件导论）介绍了程序设计的基本技巧和过程。由于第六至八章论述了有关总线控制、输入/输出和存储器方面的内容，为掌握更具体的内容提供了必要的背景知识，所以在第九章（先进的软件）中就能较详细地讨论第五章所提出的那些基本概念。

最后几章（第十至十四章）详细介绍了英特尔 8080 和莫托洛拉 MC6800 两种微处理机及其应用，综合运用了前面所学到的全部基本概念。

书末还附有术语汇编，汇总了各章和其他一些术语。

附录中提供了丰富的参考资料，包括莫托洛拉 MC6800 和英特尔 8080 的指令系统以及一些辅助的技术数据，例如数制、转换表、集成电路清单、标准逻辑符号和编码等。

在学完本书内容之后，读者应该获得理解和学习微处理机所必需的基础知识。可以以这些知识为纲，继续研究这一新兴的现代化技术。

目 录

译者的话

序言

第一章 微处理机导论	1
1.1 微处理机	1
1.2 微处理机的主要概念	4
1.3 软件概念	7
习题	8
第二章 二进制的基础知识	9
2.1 数制	9
2.2 算术运算	12
2.3 求补码	15
2.4 小数	21
摘要	22
习题	23
第三章 微处理机用的数字电路概念	24
3.1 有源门电路	24
3.2 触发器的应用	30
3.3 译码器	33
3.4 代码转换器	36
摘要	38
习题	38
第四章 指令系统	39
4.1 CPU 程序设计模型	39
4.2 存储器里的程序	40
4.3 有关指令系统的术语	42
4.4 指令	46
摘要	48
习题	50
第五章 软件导论	51
5.1 程序设计的基本概念	51
5.2 程序设计过程	52
5.3 程序设计的示例	55
5.4 几个常用的概念	58
习题	60
第六章 总线控制	62
6.1 系统概论	62

• ▼ •

6.2 时序和控制	62
6.3 地址总线	66
6.4 数据总线	69
6.5 控制总线中各条线的定义	69
习题	70
第七章 输入/输出	71
7.1 端口以及端口的工作过程	71
7.2 非可编程端口	72
7.3 可编程端口	78
7.4 成品器件	80
摘要	86
习题	86
第八章 存储器	88
8.1 CPU与存储器的关系	88
8.2 RAM	88
8.3 ROM	90
8.4 EPROM	92
8.5 PROM	94
摘要	95
习题	96
第九章 高级软件	97
9.1 引言	97
9.2 汇编语言程序设计	97
9.3 一些常用的技术	103
9.4 调试	108
摘要	110
习题	110
第十章 英特尔8080的体系结构及其配套电路	112
10.1 CPU的体系结构	112
10.2 8080CPU的体系结构	114
10.3 配套电路	116
10.4 中断型端口	121
10.5 中断用的支援电路	123
10.6 直接存储器存取	129
习题	133
第十一章 英特尔8080的应用	134
11.1 电路的介绍	134
11.2 对程序的讨论	141
11.3 输出端口	146
摘要	148
第十二章 莫托洛拉6800的体系结构及其配套电路	149
12.1 系统部件的说明	149
12.2 微处理机的引线	153

12.3 外围器件与存贮器的引线	154
12.4 静态特性和动态特性	156
12.5 最小系统	162
12.6 微处理机的寻址方式	164
12.7 微处理机的控制	168
摘要	173
第十三章 莫托洛拉6800的应用	174
13.1 IEEE488-1975 通用接口总线	174
13.2 直观显示发生器(VDG)	183
摘要	189
习题	189
第十四章 微处理机领域的展望	190
14.1 关于进一步学习的建议	190
14.2 英特尔公司的进展	191
14.3 莫托洛拉公司的进展	193
附录一 数制	215
附录二 ASCII码转换表	218
附录三 集成电路	219
附录四 标准逻辑符号	226
附录五 指令系统	227
术语汇编	319
参考文献	331
汉英对照索引	333

第一章 微处理机导论

由于大规模集成电路 (LSI) 技术的进展, 使得整个中央处理部件 (CPU) 可以集成在一块集成电路 (IC) 芯片上。因此, 在计算机领域中产生了一系列新的术语。计算机和微处理机之间有何区别? RAM, ROM, PROM, ALU, CPU, 总线、状态锁存器是什么? 希望熟悉这一新技术的人们通常都会提出这样一些类似的问题。

所以, 本章旨在帮助初学者建立一种业务语言, 并对微处理机的一些基本概念有一个初步的了解。这将对初学者在微处理机及其应用方面的学习有所裨益。

1.1 微处理机

1.1.1 定义

微处理机是一种可程序的逻辑器件, 即这种器件完成的功能或逻辑操作随其输入端所加的“指令字”不同而不同。

上述定义虽然是正确的, 但有点笼统。在技术方面, 微处理机这个词现在是指小型计算机系统的中央处理部件 (CPU)。单独的微处理机起不了什么作用, 但当它与数量不多的一组支持电路结合起来时, 就具有计算机经典定义中所包含的大多数特性。由于微处理机的速度稍慢, 字长有限, 所以大计算机制造厂往往有某种优越感, 不屑于理会这种“微不足道的计算机”。

微型计算机就是以微处理机集成电路片为基础构成的一个完善的工作系统, 微处理机本身则包含了计算机的大部分功能。微型计算机系统应满足计算机的所有最基本的要求, 包括如下内容:

- 能够输入和输出数据, 这些数据通常取数字形式。可以在微型计算机和一些普通的输入/输出设备 (诸如电传机、CRT 显示器、纸带输入机、软盘存贮器、数字磁带机、盒式磁带机和实验仪器) 之间交换数据。
- 包含一个算术逻辑单元, 用来完成算术或逻辑运算, 例如加、减、比较、左环移或右环移, 逻辑乘 (与)、逻辑加 (或)、取反、异或等。
- 包含有可直接寻址的存贮器, 其中既可存放数据又可存放指令字。
- 是可程序的, 即数据和指令可以按各种需要的顺序进行安排。在这方面, 微处理机和袖珍式计算器不同。计算器的功能通常是固定的, 而且要求按规定的顺序进行键盘操作, 并不能随意改变。

在本书中, 就整个系统而言时, 术语“微型计算机”和“微处理机”是可以互通的。这与目前的实际情况也是一致的。

1.1.2 应用

目前,微型计算机的应用介于分立元件逻辑电路和小型计算机之间,如图 1-1 所示。微处理机填补了分立元件电路和相当复杂的小型计算机之间的一大段空白。微处理机的价格也介于分立元件电路和小型计算机之间。

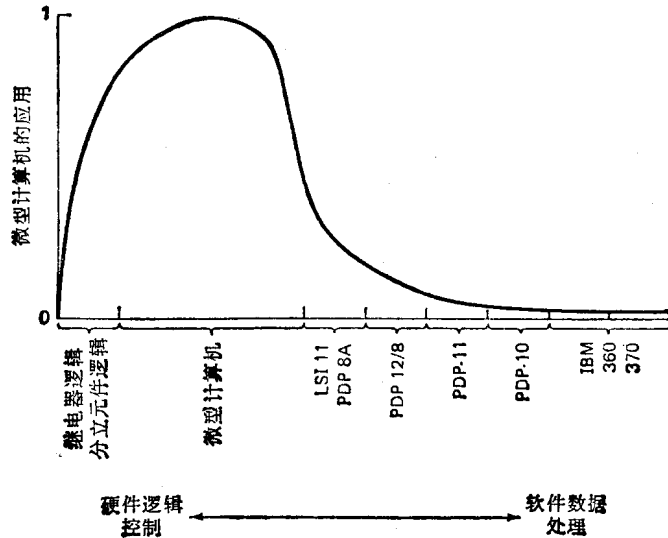


图 1-1 微型计算机的应用领域

表 1-1 微型计算机的价格范围

字长	1	2	4	8	16	32	64
复杂性	硬接线逻辑	程序逻辑阵列	计算器	微处理机	小型计算机	大型计算机	
应用	控制			专用计算		廉价通用 数据处理	高性能通用 数据处理
价格	\$100 以下			\$1,000		\$10,000	\$100,000 以上
程序	只读						可浮动
存贮容量	很小 0—4 字	小 2—10 字	中等 10—1000 字		大 1000—1 M		很大 > 1 M
速度	实时	慢	中等			适应于 数据吞吐量	
输入—输出	与其他部分 是一个整体		几种 简单设备		一些复杂设备		满间屋子的设备
设计	逻辑	逻辑+微程序			微程序 宏程序	宏程序、高级 语言、软件系统	
产量	大						小

因为微机系统相当便宜,而且灵活,所以在家庭和小型的商业系统中得到广泛的应用。它满足了一些小厂的要求,这些小厂对于各公司提供的大型计算机系统无力购置,或

者暂且不需要。

目前已投放市场的一些应用是

- 电视游戏机;
- 计算机智能终端;
- 过程控制器;
- 电话交换控制器;
- 商品盘存系统;
- 可编程序的家用电器;
- 计算机化的汽车电子系统。

微处理机也可以更广泛地用作工业设备和机械的专用控制部件。因为微处理机是可编程序的逻辑系统,所以可用于实现各种各样的作业功能(以前要实现这些功能的话,必须分别为每种功能设计专用的电路)。由于生产成本低,使得微处理机富有极大的吸引力。

也许正是由于这种把硬件和软件相结合的特点,才使微处理机产生这么大的影响。现在可用同样的硬件,凭借不同的程序实现全然不同的电路功能。因而,微处理机被认为是最终能使两个独立的领域(硬件设计员和程序员的工作领域)结合起来的一种器件。

1.1.3 以微处理机为部件进行设计

优点 微处理机的诞生使系统设计方法产生了显著的变化。随着大规模集成电路元件的推广使用,设计员就可以选用大而现成的系统功能模块,从而使整个系统的设计简单易行。

以微处理机为部件进行系统设计,与目前用工厂生产的预制构件建造房屋的作法相类似。根据传统技术,每座房屋都是由一块一块砖头砌成的,窗框和门框都是在工地上定做的,每堵墙壁也都是根据建筑设计图进行施工的。然而,用预制件建筑的房屋则是根据现有的标准预制构件进行设计的。设计者只须规定特别的门窗尺寸。随后,再把门窗与房屋的其余部分一起送至工地。最后的工作就是完成房屋装配,这样做既保证了质量,又降低了设计和产品的成本。

用微处理机进行设计同用分立元件进行逻辑电路设计之间的差别,很类似于上述两种建造房屋的方法之间的差别。用分立元件组成的系统是按设计工程师的技术说明书逐件装配而成。当电路功能有较大变化时,就要对电路本身重新进行设计。与此相反,当用微处理机进行设计时,只需要确定系统应达到的指标并选购适宜的功能部件即可,其余的工作就是系统的组装。只要改变一下程序,就实现了系统功能的转换。

以微处理机为基础的系统主要优点是

- 成本低;
- 系统设计容易;
- 灵活;
- 功能易于变更。

缺点 微处理机系统所遇到的主要问题与微处理机本身几乎没有关系,而仅与其配套使用的支援电路有关。这些系统附带的文件和文献常常不够完整,许多东西要靠用户去想

象。未入门的人可能会误认为买一个“单片计算机”就能完成其所有的设计任务，并准备开始工作。其实并非如此，操作员必须能够与其设备互相通信。

当用户使用微处理机时，必须准备好该微处理机与输入/输出设备的接口。由于缺乏完善的文献资料和操作须知，在试图研制一个完整的工作系统时，势必要由用户自己解决所遇到的问题。如果想在微处理机电子学领域中获得成功，必须先弄清楚一些基本概念和原理。

1.2 微处理机的主要概念

1.2.1 计算机的基本概念

大多数计算机基本上由四个部分组成：存贮器、算术逻辑单元、控制器和输入/输出设备。

每一部分的功能随机器不同而略有变化。但在所有的机器中，均能找到这几个部分。下面就来叙述各个部分的基本操作。

控制器向所有其他部分提供定时指挥信号和同步信号。这些信号使其他部件能传输数据、处理数值、输出和输入信息。所有这些操作都由驻留在存贮器中的用户编写的程序决定。

存贮器既存放指令又存放数据，而且可以相间地进行存放。每一种计算机都有一系列基本的内部功能。程序员使用指令系统中的指令编出程序，而后将其存入存贮器。如果程序编制得当，就可使计算机完成很复杂的任务。控制器依次逐条地取出存放在存贮器中的指令，并把每条指令“解释”为控制脉冲，然后把控制脉冲发送出去以执行这条指令。当这一工作完成时，控制器就顺序地再取出下一个字或指令。

如前所述，当数据量超过了运算器的保存能力时，数据也可以存放在存贮器中。计算机无法区分数据和指令。因此，程序员必须把存贮器中的数据和指令加以分离，以免计算机把数据解释为指令或把指令当作运算操作数使用。

算术逻辑单元（ALU）是含有完成加、减之类的基本算术运算以及移位、取反和布尔代数运算等逻辑功能的逻辑电路。算运单元是大多数计算机的活动中心，即大多数数据流从一处流向另一处时都要通过算运单元，所有的算术和逻辑运算都是在算运单元内完成的，并在算运单元中进行条件转移测试。

除非能与计算机进行通信，否则上述操作就毫无用处。要实现这种通信过程，就必须把输入/输出端口和接口电路与计算机发送信号使用的主总线相连接。

1.2.2 微型计算机的组织

在大多数以微处理机为基础的计算机系统中，控制器和算运单元制作在同一块集成电路芯片内，这块芯片就是微处理机。

微型计算机各部件间相互通信的一种典型方法如图 1-2 所示。

在图 1-2 中，I/O 端口能直接对存贮器进行存取，从而使算运单元及控制器可以对数据和指令进行操作。控制器还可选择输入和输出端口并进行控制。

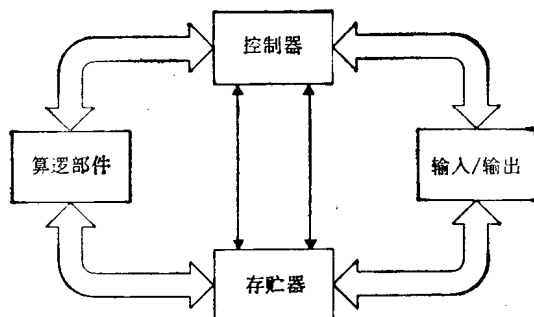


图1-2 各部件之间的通信方法

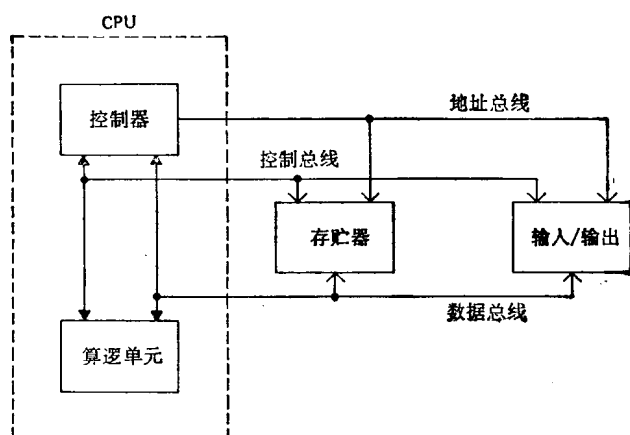


图1-3 通常采用的通信配置方案

另一种更普遍采用的通信配置方法如图 1-3 所示。

应当注意的是，图 1-3 中虚线框内所指出的就是一个微处理机芯片的典型内容。微型计算机的很大一部分功能包含在中央处理部件内。微处理机系统中颇常见的总线缓冲、地址译码和双向驱动电路的细节，在图 1-3 中均未画出。但是，信息流的一些主要概念可借助于这张图进行讨论。

1.2.3 总线

图 1-3 表明，微型计算机系统中有三组线或三条总线。这里介绍一下与这些总线有关的术语，即“单向总线”和“双向总线”。单向总线意味着数据只能沿一个方向流动，如同“单行道”一样；双向总线允许数据和信息在两个方向上流动。但是，数据每次只能沿一个方向流动，犹如正在修筑的公路上的单车道临时道路一样。

大多数微处理机系统有以下三条主要总线：

1. 地址总线：通常为 16 条单向线。
2. 数据总线：通常为 8 条双向线。
3. 控制总线：线的数目视系统而定，既有单向的，也有双向的。

通过研究图 1-4 所示的框图可以更好地了解中央处理部件如何处理总线结构。图 1-4

与图 1-3 实质上是一样的, 但图 1-4 更详细地说明了如何完成数据流的协调任务。

假如中央处理部件在执行一条指令时需要从存储器或某一个输入/输出设备获取一个信息, 或向它们发送出一个数据; 就把该设备或存储单元的地址放到地址总线上。地址的

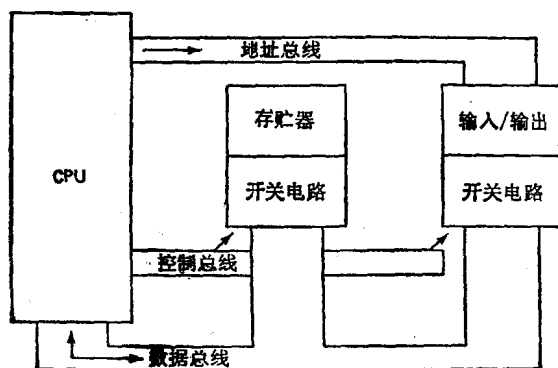


图1-4 常见的总线系统框图

建立就使特定的设备或存储单元把可用数据置于开关电路上, 或用开放来自开关电路的信号线的方法接收中央处理部件的输出信息。地址总线所做的唯一一件事, 就是在适当的时刻启动设备和存储单元。以供控制总线和数据总线使用。

然后, 由控制总线发出同步脉冲, 用来打开或关闭开关系统, 使数据能通过开关系统出入于数据总线。

在各部件之间实际信息的传送是通过数据总线实现的。数据总线通常是 8 条并行线, 同时传送构成一个数据字的各个脉冲。数据总线上的数据可以在两个方向上流动, 其方向由中央处理部件通过控制器电路送到控制总线上的定时和开关控制信号决定。地址总线通常有 16 条线, 它起始于 CPU, 并通过缓冲器、译码器电路。根据该总线上的信息决定数据的目的地。

上述简短说明只是用来表述各条总线之间的基本操作。通过对 CPU 所执行的基本操作的简单讨论, 将使我们对于微型机系统有更完整的理解。

1.2.4 指令周期

CPU 最基本的操作是取指令操作和执行操作, 这两种操作构成了指令周期。

CPU 内的程序计数器能自动地跟踪并指出内存中下一条指令的位置。当系统开机或复位时, 就自动地把程序计数器的内容设置为存放第一条指令的地址。把来自程序计数器的地址放到地址总线上, 就能从存储器中取出该指令。然后把指令放入控制器, 由控制器对指令进行译码, 决定 CPU 如何完成这条指令。在一条指令执行完毕后, 程序计数器能自动加 1, 并以另一次取指令操作为开端, 开始一个新的周期循环。

只要时钟加在 CPU 上, 这个取指、译码、程序计数器增量、执行、取指、译码、程序计数器增量的过程就周而复始地继续下去。

时钟频率决定了完成这种指令周期所需的时间。在指令周期的执行阶段所发生的动作与取出的指令有关。必须理解的是: 每个指令周期可以包含几个机器周期, 这些机器周期是 CPU 的内部周期, 完全由制造厂进行的逻辑设计和芯片总体结构规定。每个机器周期又可由几个跃变状态组成, 每个时钟周期发生一次跃变。

在指令周期的执行阶段, CPU 可能需要使用存储器和 I/O 设备, 以便完成指令功能。如果必须这样做, 就再次使用上面介绍过的三条总线。

微处理机执行特定指令系统的能力, 以及执行这些指令的复杂程度和速度随微处理机

的不同而不同。但是，下列基本操作是共同的，

1. 取指令；
2. 存储器读；
3. 存储器写；
4. 堆栈读；
5. 堆栈写；
6. 输入；
7. 输出；
8. 停机；
9. 中断。

必须记住，任何指令周期的第一个机器周期都是取指周期。执行各条指令所发生的具体跃变状态和机器周期将在后面有关软件部分中加以讨论。

1.2.5 指令系统

每个微处理机制造厂都花费了大量时间和精力为其产品提供一套功能强而完善的指令系统。用户购得的微处理机能完成独特的一组操作，这些功能是由制造厂制备在芯片内的。这些操作可归纳为以下一些基本类型，

1. 数据传送；
2. 算术运算；
3. 控制功能；
4. 逻辑操作。

用哪一种方法来完成这些操作以及指令的数量和复杂性都随微处理机型号的不同而不同。因此，在决定购买一种微处理机时，就必须考虑到这一情况。

在购买微处理机时，制造厂将提供完整而详细的指令系统表，还常常提供用户手册，这对于了解该器件的应用限制和能力是很重要的。

在购买了一种特定的微处理机器件之后，用户就得依靠自己的努力变成能有效地使用该器件的行家。用户所设计的微处理机系统的复杂程度和设计效率在很大程度上取决于学习和理解该微处理机指令系统的程度。

现有的各种微处理机完整的指令系统及其在程序设计中的应用将在以后章节中予以讨论。现在只需认识到，对于所购置的微处理机指令系统的理解是很重要的。

1.3 软件概念

1.3.1 不同级别的语言

软件这一术语现在是指计算机中的程序。早期的计算机使用机器语言，即计算机存储器中存贮的是用0和1的形式表示的一系列二进制数。显然，当程序较复杂时，用机器语言为每部机器完成规定的任务编写程序太费时间，因而软件领域开始发展起来。

机器语言（一系列的0和1）是最低一级的语言。发展下去就是符号语言。由于符号

语言是由可以被解释为指令的数字和字母构成的，所以更适于人们使用。为了使程序员不必顾及机器语言，已研制出汇编程序和解释程序，用来把这种助记符指令翻译成机器语言。用助记符和解释性语言进行程序设计的技术称为汇编语言程序设计。

因为某些例行程序在程序设计中要反复使用，并可以出现在许多程序中，因而研制了执行程序 and 监控程序（有时称为实用程序包）。由于实用程序的用法越来越复杂，因而出现了高级语言。高级语言（诸如 Basic, Fortran, Cobol 等）实际上是很复杂的执行性实用程序包，它包括操作员与计算机以某种对话性的术语进行通信所需的全部解释程序和汇编程序。

高级语言程序包将把这些术语解释和汇编为机器语言的指令。用高级语言词汇书写的一条命令可能产生几百条机器语言指令，用来完成所需的操作。

1.3.2 微处理机的发展水平

目前的微处理机只能接受机器语言。微处理机工业正在不断地研制软件，以提高微处理机的语言能力。借助于相当容量的存贮器和制造厂提供的用户程序库，现已研制成了许多允许程序员用高级语言进行工作的系统。这种系统用较大的机器来模拟微处理机，以便研制软件包。然后可将这些程序包装入微处理机系统的存贮器，因而能使程序员摆脱许多耗费时间的工作。

然而，这些被人们称为开发系统的设备价格很贵。单独研制自己的系统的用户常常担负不起这笔投资。因而，用户经常还是要采用汇编语言来编制程序并进行手工汇编。为此，本书中的软件将用汇编语言来编写。

习 题

- 1.1 计算机最低限度的要求是什么？
- 1.2 请列出你见到过的一些微处理机应用的一览表。
- 1.3 微处理机与常规计算机系统相比，其主要优点是什么？
- 1.4 微处理机除了缺乏完善的说明数据以外，还有什么主要缺点？
- 1.5 什么是总线？
- 1.6 请说明单向和双向总线之间的区别。
- 1.7 地址的用途是什么？
- 1.8 请说明指令周期、机器周期和跃变状态之间有何区别？
- 1.9 请列出指令操作的一般类型。
- 1.10 请列出机器周期操作的一般类型。
- 1.11 请用自己的话来说明各种级别的计算机语言的关系。

第二章 二进制的基础知识

研究微处理机，或者研究任何涉及数据处理或计算的数字电子学领域，都要求对二进制数制有一个基本的了解。本章旨在对二进制作一初步介绍。

2.1 数 制

2.1.1 计数

所有的数制都是在用符号来表示事物个数的过程中形成的。数制是以其表示计数值的符号个数来命名的。因为人们最熟悉十进制，所以我们将十进制作为讨论数制的一个良好的起点。

十进制 十进制由十个符号（0，1，2，3，4，5，6，7，8，9）组成，任何计数值都可由这些符号的一定组合来表示。从0到9的任何计数值均可用一个符号来表示。但是，超过这个范围的数，就必须用两个以上的符号的组合来表示。这是通过对每一列数字规定一个位置值（权值）形成的系统来实现的。各列的权值从右到左递增。

最右边一列中每个符号表示个位数，其数值顺序为0，1，2，3，4，5，6，7，8，9。这个符号系统中每一列的全部计数值被看作为在这一列中形成一个循环。这一列中每个计数值比前一个计数值大1，或 10^0 。指数（在目前情况下是0）表示该列符号代表哪一列完成计数循环的次数。现在不存在更低的列，所以指数0表示这一列是个位数。

第二列中的符号表示第1列的计数循环完成的次数。第二列权值的指数形式是 (10^1) ，说明这一列的符号表示第1列的循环计数过程完成的次数。

数33表示已完成了三次计数循环，并进入了第1列下一个循环的第三个计数状态。

$$\begin{array}{cc} 10^1 & 10^0 \\ 3 & 3 \end{array}$$

同样，数4529₁₀表示

4- 10^3 第3列进行了4次完整的循环。

5- 10^2 第2列进行了5次完整的循环。

2- 10^1 第1列进行了2次完整的循环。

9- 10^0 第1列上完成了9次计数。

每当在某一列上完成一次完整的计数循环，左边相邻的一列就增加一个符号计数值。而该列则恢复为“0”，并重新开始计数。

这样的计数系统并不是唯一的，其独特之处是所用的符号个数，即构成该数制的基数与其他数制不同。十进制使用十个符号，八进制使用八个符号，三进制使用三个符号，十六进制使用十六个符号。所有的系统都采用按位加权计值的办法。

八进制 八进制系统仅有八个符号：0，1，2，3，4，5，6，7。个位数（即

8⁰) 每计八次数完成一次循环, 而第 2 列 (8¹列) 表示第 1 列完成计数循环的遍数。

数 5232₈ 表示:

5-8³ 第 3 列进行了 5 次完整的循环;

2-8² 第 2 列进行了 2 次完整的循环;

3-8¹ 第 1 列进行了 3 次完整的循环;

2-8⁰ 第 1 列完成了 2 次计数。

二进制 二进制与其他数制没有什么区别。但是二进制只用两个符号: 0, 1。

数 1011₂ 表示如下:

1-2³ 第 3 列进行了 1 次完整的循环;

0-2² 第 2 列进行了 0 次完整的循环;

1-2¹ 第 1 列进行了 1 次完整的循环;

1-2⁰ 第 1 列完成了 1 次计数。

由于人们熟悉和习惯于使用十进制, 总是想方设法找出其他数制与十进制的相互关系, 并用十进制符号来表示其他数制的数值。实际上, 为了适应人们表示数值的习惯, 就要增加硬件和软件 (程序)。

数 1011₂ 意味着总共进行了 11 次计数, 就象十进制数 11 表示进行了 11 次计数一样明确。二进制数 1011₂ 看起来有点奇怪, 唯一的原因是大家对它不熟悉。

从事数字系统工作要求熟悉下列几种数制:

十进制;

二进制;

八进制;

十六进制;

二进制编码的十进制 (一种特殊的数制)。

在数字系统中也使用其他一些数制, 但上述这些数制是最常用的。使用所有这些数制都能完成算术运算和数据处理。

为了使工作不断深入下去, 必须了解基本的算术运算过程。由于读者最熟悉十进制, 所以在阐述基本算术运算过程时将以十进制作为示例数制。

机器中使用的数制 机器中主要采用的是二进制数制。机器里所有的基本运算都是用二进制完成的。此后, 只是为了便利用户才把数据转换成其他某种数制, 以供显示。

在书写和使用不同的数制时, 标准的办法是用一个下标注出该数制的基数。例如:

1101₂, 375₈, 375₁₀, 375₁₆。

若不这样注明, 则该数的值将是不确定的。

附录中列出了一系列表格和数值对照表, 以供查阅。

2.1.2 不同数制的转换

假如用数学观点来看的话, 从一种数制转换为另一种数制可能会产生混淆。但是如果统一用机器语言 (即二进制) 格式进行处理, 那么数制转换就会变得容易得多。

数位的分组 在简单二进制与任何其他数制的二进制表示法之间进行转换时, 这些数