微计算机软件设计基础

〔美〕MGI 管理学院 修订
凌瑞骥 张一飞 邹立华 译
齐国生 顾书偃

清华大学出版社

微计算机软件设计基础

[美] MGI 管理学院 修订

凌瑞骥 张一飞
齐国生 顾书偃

邹立华 译

清华大学出版社

内 容 简 介

本书是本国外内部发行的微型计算机软件设计函授教材，它共有十三章，其中基础篇三章，正文十章。在基础篇中深入浅出地介绍了微型计算机的基本概念、各种类型软件以及在设计系统时对硬件和软件方案作出权衡的方法。正文部分则先从解决实际课题入手来阐述如何评价一台微型计算机，然后再对软件设计各个步骤分别作了分析和说明。第三章至第五章介绍了一些重要的基本编程技术，它们包括：各种类型的判定与分支结构、循环结构、栈和子程序等，并以脉冲检测、译码、编码等大量实用程序为例介绍了如何应用这些基本编程技术。第六章介绍了在微型计算机应用中常见的整数、多倍精度数、浮点数、十进制数算术运算的各种算法和程序。第七章、第八章详细地介绍了串行/并行输入/输出、定时器、中断、实时编程、数据块传送等各种输入/输出编程方法，并且着重以一个复杂的住宅安全系统程序为例介绍了如何综合应用各种编程技术和方法。第九章介绍了高级语言的编程特点。第十章介绍软件系统的模块化设计方法和程序调试方法。

本书适合广大在职科研人员、工程技术人员自学使用，也可以作为大专院校微型计算机软件课程的一部实用教材。

微计算机软件设计基础

[美] MGI 管理学院 修订

凌瑞骥 张一飞
齐国生 顾书偃 邹立华 译

★

清华大学出版社出版

北京 清华园

北京新华印刷厂印刷

新华书店北京发行所发行 • 各地新华书店经售

★

开本：787×1092 1/16 印张：23 1/4 字数：600 千字

1984 年 10 月第一版 1984 年 10 月第一次印刷

印数：000,001～120,000

统一书号：15235•114 定价：3.30 元

译 者 序

这是一本在国外内部发行的专门讲授微计算机软件设计的函授教材。同国外大量流行的微计算机软件设计教材相比，我们感到它有以下特点：

1. 它是专为大量在职的科研人员、工程技术人员自学微计算机软件基础知识而编写的，它最适合于早已从事工作，而原来未曾学过计算机，现在又迫切需要通过自学掌握微计算机的人们来使用。

2. 它是一本入门教材，但又有一定的深度和广度。它不像很多微计算机用户手册那样只是就事论事地解释某种微计算机的结构、原理和用法，而是从计算机科学的基础知识出发，系统的讲授微计算机的软件方法和应用，在讲述微计算机时有意识地加进一些计算机科学的背景知识。

3. 它很注重应用，有大量的例题。在应用中又偏重于把微计算机用作控制手段，而不是单纯的计算工具，所以最适合于从事自动控制、仪器仪表系统的设计人员来使用。

4. 它备有大量的自我测验和技巧练习题，供读者检验、巩固和加深自学的成果。

总之，我们认为它是一本有价值的自学参考书，所以把它译出，以馈读者。

虽然本教材中的实例是用 8080 指令系统编写的，但是教材中所论述的软件基础知识和软件设计方法以及书中所例举的算法，对于各种类型的微计算机都是适用的。至于本教材所介绍的结构编程方法在微计算机软件设计方面则更有指导意义。书中用 8080 指令所写出的一些应用程序，在 8085、Z80 以至 8086、Z8000等微计算机中也只需稍作改动甚至不作改动即能使用。至于由于出版较早(1977年)书中某些过时的论断和未及反映近年来微机与超级微机的巨大发展则是不足之处。

对于完全没有计算机预备知识的读者来说，自学这本教材时在碰到涉及广泛计算机科学背景知识的议论部分会感到有些困难。但是，我们建议读者不要因此而放弃，最好坚持读下去，在抽象议论的后面有浅显易懂、结合实际的讲解，这些讲解对初学者来说不会是困难的，学完这些，再回过头去看那些议论就比较容易理解了。

在翻译中我们痛感国内计算机词汇和译法不一之苦。书中还有一些词，从现已出版的各种词典和专业词典上都找不出译法，只好自己杜撰，并在注释中略加说明。

由于译者在计算机科学领域中都还是小学生，外语水平又有限，所以译文中必定会有不当之处，衷心地希望读者指正。

原　　序

欢迎您学习 MGI 的新教程：《通过软件设计，掌握微处理机》。您将发现本教程同您学过的其它教程相比，有许多不同之处。

首先，这是一部自学参考教材。学习本教程，读者可以根据本人的具体情况选择自学起点而不是由教材的编者武断地规定该从何处开始。

本教材有下列特点：

- 基础篇中每一章都备有一套自我测验题用以帮助读者判断该从何处开始学习。
- 每章均有一套复习题编排在教材的特定页上。
- 每章还有一套训练技巧的作业，读者可将完成的作业寄送 MGI 评阅。（译者按：这是针对国外而言的，对我国读者 MGI 是不受理的。）
- 每一单元都附有完整的索引。读者随时可以用它找到教材中自己感兴趣的课题。

《通过软件设计，掌握微处理机》是专为在职工程技术人员编写的，它使工程技术人员得以把自己已经具备的技术素养和有条不紊、一步步地解决问题的技巧应用到一个崭新的微计算机领域之中去。

读者从本教程中学到的知识和技巧是不会过时的，因为微计算机的硬件及其概念几乎是日新月异的，软件则不然，而读者从这本重要和新颖的教程中所学到的正是软件。

本教程的程序编制例题均以 8080 型计算机为例，其原因有二：

一、8080 是当前这代微处理机中使用最广、软硬体配套最完备的机型。

二、8080 的总体结构具有很大的通用性，本教程中以 8080 为例编写的程序可以毫不费力地转换到其它类型微处理机上去使用。

学完本教程后，读者将具备一系列新的技能，足以胜任下列工作：

- 在自己工作的企业中，发现能够应用微处理机的机遇；
- 在企业中，倡议通过应用微计算机开辟新的产品和全新的经营领域；
- 开展技术咨询，协助小企业采用微计算机；
- 研究与发展微计算机的新用途、新产品和所有其它重要的方面；
- 由于读者掌握了这一重要的新知识，从而得到所在企业对自己的器重。

请您有空的时候开始学习这门重要而新颖的教程。先流览全书，阅读基础篇的自我检验习题和各章后面的重点复习题，然后决定您该从何处入手。

我们将用一切方法帮助您在这一重要领域增长技能，如果您感到有什么地方不清楚或难于理解，请写信给我们。如对本教程教材及其附录、说明、有什么疑问，或对改进本教程有什么建议，均请给我们来信。凡属来信，定将及时作复。

祝学习顺利、称心如意。

目　录

基础篇 A　计算机的基本概念

在基础篇 A、B、C 三部份的开头，都有一个相类似的自我测验练习，这些测验练习的目的是帮助你测定这部分所涉及的问题你是否早已掌握。每个基础篇的末尾都有自我测验的答案。

怎样使用这些自我测验练习呢？请按照下面三个简单步骤执行。

1. 在每个问题下面的空白处写上简短的答案。

2. 比较你的答案和本基础篇末尾的答案。

3. 每个问题解答的末尾都有参考页号，即本题所讨论的内容在书中的页号。对于那些你还没有掌握的课题，可以阅读参考页号所指出的那部分内容，使你在这部分基础知识上得到有益的帮助。

这些自我测验练习不需要评定分数，也不需要把答案送回 MGI。它的唯一目的是帮助你尽快地把注意力集中到对你来说是比较重要的部份中去。

自 我 测 验 A

A—1　什么是算法？

A—2　画出计算机基本结构的框图

A—3　定义 RAM，磁心和读/写存储器。

A—4　计算机的四类基本结构指令是什么？每一类指令的功能是什么？

A—5　定义计算机的“字”、“字节”和“位”。

A—6　定义“寄存器”並且举出两个寄存器的例子。

A—7　什么是标志位？它们都用于什么目的？

A—8　定义间接寻址。

A—9　程序启动的数据传输和中断启动的数据传输之间有什么区别？

A—10　什么是总线？

A—11　在执行任何机器指令时，计算机要经过两个什么阶段？

A—12　执行一条指令，计算机需要什么信息？

导言：为了充分地理解微计算机所能提供的各种便利，我们首先必须回过头来了解一下通用的计算机，从这一观点不难看出微型计算机只不过是通用数字计算机的后裔而已。在最初设计微处理机和其相应的微计算机时，由于半导体工艺的限制而遇到严重的障碍。最新的微处理机则具有与最现代小型机完全相同的通用计算机结构。并且只有当你懂得了一个通用计算机所具有的特性及其结构之后，才能够理解微计算机事实上就是一个通用计算机。

A.1　计算机的基本组成

从单价为 9.95 美金的袖珍计算器到极复杂的计算机网，所有的计算机都有一个共同的目的：按照明确规定的步骤和过程，把一种形式的数据转变成另一种形式。如果用最通用的方

式来看待计算机，计算机是一个黑方框，输入方框的是已知数据，而从方框输出的是所求得的数据。被转换的数据称作原始数据，而转换后的数据称为处理后数据。数据转换的方法称为算法。（见图 A.1）

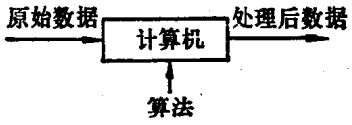

图 A.1 通用计算机操作

计算机之所以具有令人难以置信的多种功能，是由于它有着为各种目的进行数据转换的特性。在实际应用中，可以设想同样的计算机既能用于器件测试，过程控制、防盗警报、工资单计算，也能用于其它可由算法定义的过程。事实上，计算机面临的应用是多种多样的，而有些计算机在某些种类的应用中比在其它种类的应用中能够执行得更好。另一个事实是：只要具备了适当的硬件功能，任何一台计算机都能执行任何一个可以定义的算法。

在通用黑框图中，我们需要若干种器件来处理数据。数据可以采用现实世界中的许多形式，它可以是信号、数码、在卡片上用穿孔表示的码等等。要设计一个能够处理任何形式的原始数据的计算机是不可能的。首先必须把我们感兴趣的数据收集起来，随即需要找到某种方法将这些数据变成计算机所能接受的形式，然后我们需要用某些方法来告诉计算机对所存储的数据应进行些什么样的操作，最后，在处理完成以后我们必须把处理完的数据转换成所能使用的形式。因此，中央处理机显然是计算机的一个重要部分，但它只不过是数据转换所需要的若干部件中的一个。一个中央处理机如果没有必要的配套器件，又没有采集、储存和把数据从一种形式转换成另一种形式的程序，那么它将是完全无用的，计算机只有作为一个系统才能是有用的。如果我们进一步来研究这个总框图，就会发现它由六个通用的单元组成，即：输入、存储单元、算术逻辑单元、控制器、输出和程序，这一通用的结构如图 A.2 所示。

现在让我们来研究每个单元的功能。

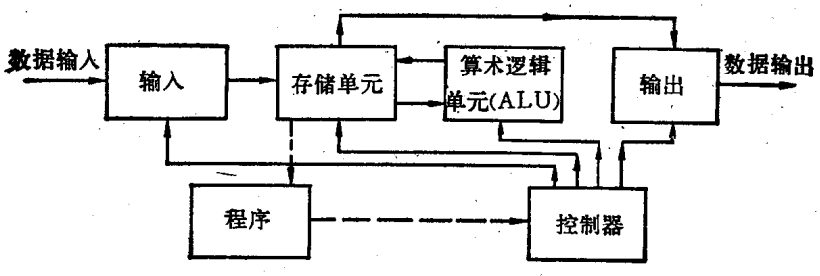

图 A.2 通用计算机结构

A.1.1 输入装置

输入装置是一个把数据从外界送入计算机的工具。各种输入装置的复杂程度相差悬殊，从最简单的搬键开关到复杂的光学页式阅读机均是。由于输入装置日益复杂，它们常常执行那些在通常情况下要由计算机来完成的操作，这叫做预处理，它使处理机更为有效。例如：许多键盘都能够把掀合的键转成预先定义好的 ASCII 码（美国标准信息交换标准码）*。因为不需要费时间去计算掀合键对应的是什么码，于是计算机就能有更多的时间去完成别的任务。

* ASCII 码在附录 B 的第 B—1 页详细叙述。

用什么方法把数据送入计算机？通常要由输入装置的速度来决定。慢速装置通常是在收到数据时就进行传送。快速装置则常常先将数据累积起来然后再送给处理机，这就叫做缓冲。因为只在积累起一组数据之后才进行传送，缓冲作用提高了处理机的效率。在不规则情况下收集数据时也常常需要缓冲，因为输入装置承担等待数据的任务的同时，计算机可以做些其它的工作。最复杂的输入装置经常完全独立地把数据送入计算机。当它们有了完备的缓冲器时，就可以利用称作 DMA（直接存储存取）的计算机特殊功能把数据直接送入处理机的存储器，不过这种传送方法只限于在极快速度输出输入装置上使用，因为当 DMA 进行时，处理机通常必须停止所有的其他操作。无论采用什么样的方法，输入装置都具有收集数据并把它送进计算机的特点。

A.1.2 存储器

存储器是计算机存储数据和程序的地方。其存储内容有：输入数据、部分处理完毕的数据、等待输出的数据和将要执行的程序。凡是计算机所要用的指令码、数据以及处理结果都必须存储在某种存储器里。

A.1.2.1 存储器分类

计算机记忆装置（或常常被称为存储器）可以大致分成随机存取存储器(RAM)和顺序存取存储器。RAM 的组成使得对任何一个储存单元读或写的时间都是相同的，即对存储器随机地存取数据要用同样的时间。实际应用的 RAM 器件有磁心存储器、半导体读写存储器和半导体只读存储器。从顺序存取装置中的一个存储单元读和写所需要的时间取决于该存储单元距数据传输部位的远近。例如，对于磁带来说，存取时间取决于磁带上的数据与磁头之间的距离。顺序存取存储器有磁盘、磁带和纸带几种类型。

存储器还可以根据存储方式进行分类。它既可以是读写存储器，也可以是只读存储器（只写存储器在应用上是有局限性的）。读写存储器存储的数据能够读出，而且在程序控制下能够修改。只读存储器则正像它的名字那样，只能读出来而不能修改。读写存储器在一般应用中为最常见的类型，大多数大型和小型计算机几乎全部采用读写存储器。只读存储器（ROM）正在日益广泛地流行，尤其在微处理机的应用中，它能十分有效地用来储存那些完全调试好的、而且使用频繁的程序。值得注意的是只读存储器一般只用来保存引导子程序，利用这个引导程序把更为复杂的程序送入计算机。微处理机和可编程的计算器则常常在ROM中存放整套程序系统，从而扩大了只读存储器的用途。而读写存储器则只用来储存运行当中的程序。

由于在存储器结构、存储器件类型和存储方式这三者之间的混淆，从而产生了名词、术语方面的混乱现象。RAM（指结构类型）或磁心存储（指器件类型）这个技术名词通常是指读写存储器（一种存储类型）。早在随机存取结构不言而喻几乎都是磁心读写存储的时候，上述名词术语方面的混乱现象就已存在。在那时候，RAM、磁心存储器和读写存储器指的全是一样的存储器。固态读写存储器一出现就发生了问题，因为这一来，读写存储器既可以是半导体存储器，也可以是磁心存储器。随着在系统中引入并日益增多地采用只读存储器（ROM）就更加剧了这种混乱，这是由于 ROM 的组成一般是和随机存取存储器一样的。为了避免这种混乱，下面将把读写存储器叫做 RAM，而把只读存储器叫做 ROM。言外之

意 RAM 和 ROM 两者是按随机存取方式组成，顺序存取设备则将根据设备种类来称呼。（亦即：从纸带上读，写入磁盘等等。）

A.1.2.2 *存储分级（或存储层次）*

除了各种存储器的组成及设备或器件类型以外，经常需要按照存储分级来考虑计算机存储器。存储分级以在程序执行过程中，储存在存储器内的数据能以多快的速度存取而定。

存储系统有两个明确的特征——存取速度的快慢及存储容量的大小是划分存储等级的标准。所有的数据和程序执行的指令都必须存入存储器。为了使计算机能以适当的速度处理数据，就需要在存储器中高速地存取数据。由于同样的原因，必须有足够的存储器来存储系统所用的全部数据和程序。在理想的情况下，只用一种类型的存储器就能存储所有的数据并以适当的速度检索之。遗憾的是，这不是一个现实可行的方案，因为高速存取和大存储容量常常是互相矛盾的。为了解决这个问题，我们经常把少量的高速存储器和容量不断增大而速度越来越低的存储器协同工作。例如把正在执行的程序存放在高速存储器中，并按照需要把其它数据和程序送入低速存储器或从低速存储器中取出。（见图 A.3）

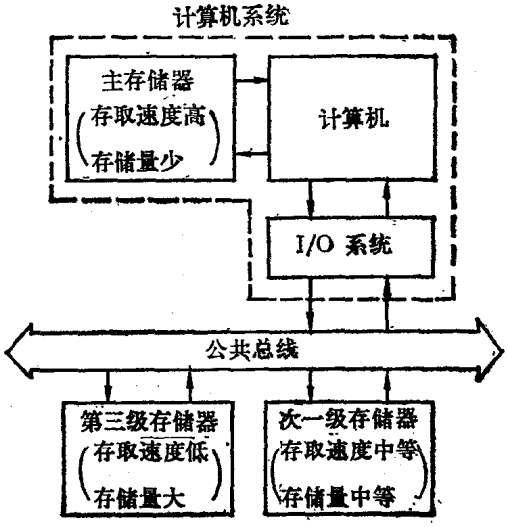

图 A.3 存储器层次结构

主存储器由高速随机存取器件组成，在这部分存储器中我们可以找到读写存储器和只读存储器。现代计算机主存储器的存取时间范围在 100ns—1μs 之间。各种应用中的主存储器的容量差别是很大的，微型计算机通常有 4K—32K 的主存储单元，大型计算机可以有几兆字节（MB）。

主存储器结构通常被设计成整个计算机中的一部分。以后，在谈到存储器寻址方式时，我们将再讨论如何存取主存储器中的数据。

存储分级中的次一级是由容量中等和存取时间为中到高速的器件组成。在这些器件中存储的数据处于“联机”状态，就是说计算机随时可以调用，只不过它的存取时间比主存储器长。

现代计算机系统中次一级存储中最常用的是磁盘。单个磁盘的存储容量，其范围可由软磁盘的 256K 数据单元直到某些高速大型计算机系统磁盘的 20M（兆）数据单元，数据存取时间随数据在磁盘上的位置而变化，典型的平均存取时间是 5 至 100ns。然而这样说是不恰当的，因为我们很少以单个数据单元的方式从磁盘传送数据，而常常以数据块的方式传送。一旦找到数据块的第一个数据单元，就能以每秒 250K 至 2M 数据单元的速率来传送。

第三级即最低级的存储器为大存储量和低的存取速度的器件。这些器件经常用来脱机存储数据，即在读取数据之前，必须首先把它们装到一个机构上，用这个机构来读取数据。常见的第三级存储媒介有磁带、纸带和穿孔卡片。如果把装载存储媒介的时间计入，这类存储器的时间将以“分”计算。正如我们前面讲过的，第三级存储器件是大容量存储器。一个

单个的磁带能保存三千万以上的数据单元，当需要调用数据时，该数据将被送进主存储器或次一级存储器中。

设计中把主存储器当作计算机的一个组成部分，而次一级和第三级存储系统则常与主计算机不在一起，这就意味着，它们要通过输入输出(I/O)结构将数据送进计算机或从计算机中输出。从这方面看，这些存储器件在运行操作的角度上可以被当成输入输出设备，但是，它们在系统中的主要功能仍是存储器的一部分。

A.1.3 控制器

控制器可以产生一系列定时的和控制的信号，从而驱使计算机完成所需要的操作。为了作到这一点，控制器从主存储器取出数据并进行译码，被控制器译码的数据称为机器指令。当检查计算机的存储内容时，无法辨别检验的究竟是程序数据还是机器指令。完全可能像数据那样对一个计算机程序进行运算以修改它的操作。但很少会这样作，因为这会使程序故障分析变得十分困难。

主存储器中将要执行的程序是按顺序存放的，控制器则按顺序取出每一条指令进行译码并用来控制寄存器、总线、运算器和其它计算机资源。针对计算机指令系统中的每条指令，这些控制信号都是唯一的，计算机所能完成的全部唯一的控制时序总称为指令系统。根据所完成操作的基本性质，可以把计算机指令系统分成四个基本类别。数据传送指令、算术逻辑指令、控制传送指令和处理机控制指令。数据传送指令用来把数据输进、输出处理机或在处理机内运转。算术逻辑指令是利用确定的算术运算或逻辑函数来变换数据。控制传送指令是用来控制程序执行的顺序。处理机控制指令是用来控制某些计算机硬件功能的。（上述所有的各类指令均将在第一章中加以讨论和阐明）一般说来，计算机越大，指令系统就越大，功能也越多，但这不是选择计算机的唯一标准，必须把指令系统考虑在整个计算机总体结构之中，计算机结构和它的指令系统是密切相关的。在整个课程中，我们将讨论指令系统、计算机和为解决问题所用的程序这三者之间的内部关系。

A.1.4 算术逻辑单元

算术逻辑单元是计算机中进行数据变换的那一部分，这些变换是按照一定顺序完成的，而这一顺序则是由正在执行的程序确定的。算术逻辑单元由寄存器、移位器、标志位和逻辑电路组成，它们是使计算机完成一系列算术逻辑运算所必需的，每条独一无二的指令代表这一系列运算操作中的一个。

不同的计算机所执行的算术逻辑功能在数量上有很大的差别。最简单的算术逻辑单元只能完成一些基本功能，如：加法、求反、移位和简单的比较，这就要求用户在各类比较简单的指令之外再建立起比较复杂的功能（如乘法和除法）。大型计算机的 ALU 经常能以单条指令完成这些复杂的功能，因此编程就容易得多。

当考虑任意一个计算机的算术逻辑单元所完成的功能时，必须时刻牢记：对于一个指令系统的关键是要求逻辑上的完备程度，一个逻辑上完备的指令系统可以用来构成任何一种复杂的算法。能做到这一点，还不过是个很简单的指令系统，但实际上每个现存的计算机都具备比满足逻辑还要多得多的功能。一个大的指令系统虽然可以使计算机编程变得容易，工作效率得到提高，但是从解题角度来考虑，并非一定要大的指令系统不可，任一具有逻辑上完备

的指令系统的计算机都能表示任何一个定义明确的算法。

算术逻辑单元在进行算术和逻辑运算时，会根据运算结果把某些触发器或标志位“置1”，于是控制器就能通过检测这些特征位来确定运算结果。根据这个信息就能决定下一步该进行什么操作，这就是条件指令，这些指令使计算机能够作出判定。犹如功能一样，标志位的数目也随计算机的不同而异，较大的和较复杂的计算机一般有较多的标志位，相应地也就有较多的条件指令。

A.1.5 输出

输出设备把处理完的数据从计算机送回外部世界。它可以为人们提供可以使用的输出信息（打印机、显示终端）或者产生用来启动和控制别的设备的输出信号。此外，那些在技术上作为存储装置的寄存器件、纸带、磁带等等有时甚至也经常被当作输出设备。

在计算机和各种输出设备间传送数据同计算机和输入设备间传送数据有着许多相同之处。例如单个数据传送、中断、缓冲及直接存储存取（DMA），它们既能够用于输入，也能够用于输出。

输出设备正如输入设备一样，可以包括从最简单的指示灯到最复杂的实时接口，这些接口可以控制整个工厂及练油厂。同样，完全可以设想一台计算机能够当作另一台计算机的输入或输出设备，由此便引出计算机网和多处理机的概念。这是现代数据处理技术中人们最感兴趣的两个领域。

A.1.6 程序

程序是一种手段，用它来定义计算机所要执行的任何一个过程。程序可以按不同的级别来写成（见基础篇B），但是要执行它，必须把所有的程序最终变成一连串选自计算机指令系统的机器指令，然后把这些机器指令存进计算机的主存储器并且顺序地进行译码及由控制单元去执行。

如果不考虑程序是针对何种计算机而设计的，则一切程序统称为软件。本书的主要内容包括软件设计和专门介绍某些计算机上具体的程序设计。与通常的逻辑设计相比，软件的优点则在于通过改变程序就能改变系统的功能，其效果同代价昂贵的重新设计硬件是一样的。因此，掌握性能良好而又通用的软件设计原则对于未来的数字系统设计者而言，将是必不可少的。

A.2 计算机体系结构的组成

在谈到计算机时，常常会听到关于各种计算机体系结构优点的评论，由于每个人都有自己喜爱的体系结构，因而这种讨论经常是很活跃的。“计算机体系结构”是什么呢？它是一种手段，通过它设计者能将前面所讲过的通用计算机的各种功能付诸实现。它包括：数据的表示方式、寄存器、算术逻辑功能、存储结构、输入/输出结构、数据通路、控制时序以及其它能使计算机传送数据的性能。由于计算机体系结构影响着计算机运行操作的一切方面，因此设计师一般都侧重于计算机的运算逻辑单元（ALU）及控制器的设计。在这些设计中还包括了为控制在主存储器、输入输出设备、算术逻辑控制单元之间数据传送所需要的信号。

运算逻辑单元（ALU）和控制器合起来称为“中央处理机”（CPU）。

通过前面的阐述我们已经了解到所有的通用计算机包括五个基本硬件功能（输入、存储、控制、运算逻辑和输出），我们还将看到计算机都是通过把若干个基本的结构元件组合起来用以实现这些基本功能。下面先讨论这些结构元件，然后再讨论当它们被组成计算机时是怎样工作的。

A.2.1 数据单元的表示方法（字、字节和位）

数据单元的表示方式看起来似乎同硬件功能无关，但实际上它是计算机设计的基础。通过数据单元的表示方式，我们来表明计算机所用的基本数据单元的大小。数据单元的表示方式之所以重要，原因有二：根据它来决定其它结构元件的大小；它对计算机表示数据、指令和存储地址的方法有影响。

数字计算机能识别二进制数据。所有二进制数都由二个数字“0”和“1”组成，因为用硬件开关逻辑电路来表示二进制数十分容易，开关闭合可被当作“1”；开关断开可被当作“0”。在实际应用的计算机里，这些开关是由高速集成电路逻辑器件来实现的。

二进制数据的最小单位称为二进制的数字或“位”。构成一台基本数据单元只有一位的计算机是不实际的。现实的作法是把几个数组成位组，这个组或“字”成了基本的数据单元。随着计算机不同的应用，字长也不一样。微处理机所用的最小字长是4位，而大型计算机的字长可达64位。一般说来，计算机字长越长，计算机功能就越强，其价格也就越昂贵。常见的微型机字长是八位，这八位字长是国际商用机器公司（IBM）为他们的360计算机系统所选定的32位字长的四分之一。为了方便起见，他们给这八位数据单元起了一个绰号，叫作“字节”（每32位字有4个八位字节）。这个绰号终于叫响了，于是“字节”就成为用来表示任何八位数据单元的通用术语。因此，当你听到一位微型计算机用户谈到存储器中的“字节”时，他所指的就是存贮器中的八位字。

在计算机的寻址方案、算术运算和指令系统中都可以看到计算机字长对运行操作的影响。当一个给定的字长为N位时，它可以表示出2^N个互不重复的数值。所以一个八位的计算机只能表示$2^8=256$个互不重复的数值。如果单个数据单元是用于算术运算，则它只能表示256个的数值，如果它用作存储地址，则只能表示出256个存储单元的地址；如果用它表示机器指令，则这个计算机就只能有256条指令，如果256条指令已足够的话。在绝大多数的应用中，需要比256更多的数值和存储地址，为此，要使用由几个字组成的指令。当M表示指令中的字数，而N表示字中的位数时，这种方法可以把互不重复的单值扩大到2^{NM}个。

在计算机设计中，字长的选择要受到几方面因素的影响。价格当然是重要的，因为字长越长，就需要用更多的硬件来实现寄存器、存储器、算术逻辑单元和控制功能。另一方面，若字长较短则每个字只能表示出少量的数据和指令值，这样就需要使用多个字长指令和地址，这就必然会降低系统速度和编程的灵活性。与此同时，设计者还必须考虑半导体器件制造工艺可能性。设计者对字长的选择不可能是一个随意决定的过程，恰恰相反，他必须衡量他的实际需要，然后在价格和性能、工艺局限之间作出权衡和取舍，以达到一个最佳的折衷方案。

计算机类型同数据单元大小的关系是有些重叠的，但是能够有某种普遍的规律。微处理机

几乎总是采用四位或八位的数据单元，多片微处理机和小型计算机一般采用 12、16、18 和 24 位的字长，中型的和大型的计算机则采用各种不同的字长，最常见的是 24、32 和 60 位。

必须记住：在微型计算机与小型计算机之间、小型机与中型机之间，以及中型机和大型机之间实际存在的差别远不止于字长不同，而是复杂得多。上述字长只不过是个粗略的指标而已，更重要的差别则是包括 CPU、存储器及外围设备在内的整个系统结构。正如客观世界中绝大多数的现象一样，事物并非截然分开的，相反，各种范畴常常有所交融，形成连续的系统分类谱。而每个系统又是独特的，必须个别地加以评价。

A.2.2 寄存器

寄存器是一种存储元件，用以保存计算机所使用的数据。从 CPU 总体结构来说可以把寄存器当作 CPU 内部的存储设备，而不是主存储器的一部分。CPU 利用寄存器来保存为进行运算操作所需要的地址和数据，也用寄存器来接收运算的结果。当执行程序时，CPU 在存储器、输入输出设备、逻辑单元和寄存器之间传送数据，某些指令可以修改所传送的数据，另一些指令则仅仅把数据从一个寄存器传送到另一寄存器。因此大多数寄存器的字长同计算机基本字长相同，而某些专用寄存器字长则可能比基本字节或长或短。

寄存器是根据它们在 CPU 里所发挥的作用来分类的，地址寄存器、数据寄存器、指令寄存器、变址寄存器、程序计算器、状态字寄存器和累加器都属于常用的寄存器。[累加器和状态寄存器从功能上说是寄存器，但从运行上是算术逻辑单元（ALU）的一部分，所以在这一节和下一节都将讨论它们]。

地址寄存器

计算机利用地址寄存器保存一些数据单元的存储地址。通过把地址寄存器的内容放到地址总线上，计算机就能存取或改变该地址储存单元的内容。

数据寄存器

数据寄存器是用来保存 CPU 与存储器、输入输出设备相互间所传送的数据的。数据寄存器也常用来保存 ALU 运算所需的一个运算数。

指令寄存器

指令寄存器保存将由控制单元译码的机器指令，当存储器的内容将被当作指令进行译码时，就把它放到指令寄存器中。

变址寄存器

变址寄存器用来对一些数值常量进行计算或寄存，用这些数值常量可以改变地址寄存器、数据寄存器或累加器的内容（即加一常量）。

程序计数器

程序计数器是用以保存计算机正在执行的指令地址的寄存器。通常情况下，在每条指令之后这个寄存器的内容加 1 并按顺序从存储器里取下一条指令。然而某些指令能改变程序计数器的内容，因此能将程序转移到由新内容所规定的存储单元。