

计算机技术与农业现代化

孙月强 著



 电子科技大学出版社

计算机技术与农业现代化

孙月强 著

电子科技大学出版社

图书在版编目 (CIP) 数据

计算机技术与农业现代化 / 孙月强著. —成都:

电子科技大学出版社, 2015.4

ISBN 978-7-5647-2955-4

I. ①计… II. ①孙… III. ①电子计算机—应用—农业—研究—中国②农业现代化—研究—中国 IV.

① F320.1

中国版本图书馆 CIP 数据核字 (2015) 第 081459 号

计算机技术与农业现代化

孙月强 著

出 版: 电子科技大学出版社 (成都市一环路东一段 159 号电子信息产业大厦
邮编: 610051)

策划编辑: 岳 慧

责任编辑: 岳 慧

主 页: www.uestcp.com.cn

电子邮箱: uestcp@uestcp.com.cn

发 行: 新华书店经销

印 刷: 四川永先数码印刷有限公司

成品尺寸: 140mm×203mm 印张 5.5 字数 170 千字

版 次: 2015 年 4 月第一版

印 次: 2015 年 4 月第一次印刷

书 号: ISBN 978-7-5647-2955-4

定 价: 16.00 元

■ 版权所有 侵权必究 ■

◆ 本社发行部电话: 028-83202463; 本社邮购电话: 028-83201495。

◆ 本书如有缺页、破损、装订错误, 请寄回印刷厂调换。

目 录

第一章 微机系统导论	1
第一节 微机系统组成	1
第二节 微机硬件系统结构	5
第三节 微处理器组成	6
第四节 存储器	8
第五节 微机工作过程	11
第二章 微机运算基础	18
第一节 数制	18
第二节 数制之间的转换	20
第三节 二进制编码（代码）	23
第四节 二进制数的运算	26
第三章 输入输出与中断	29
第一节 输入输出接口概述	29
第二节 CPU 与 I/O 接口通讯的方式	32
第三节 中断技术	34
第四节 常见的微机接口介绍	42
第四章 计算机硬件应用技术	53
第一节 微型计算机的概述	54
第二节 微型计算机部件技术	54
第五章 农业现代化与城镇化发展	140
第一节 三农问题	140
第二节 农业现代化	149
第三节 计算机在农业的应用	154

第四节	计算机及信息技术在日本农业上的应用.....	156
第五节	计算机在农业机械中的应用.....	158
第六节	农业可持续发展	164

第一章 微机系统导论

本章首先从总体上说明微型计算机（简称微机）系统组成的基本概念，并对硬件系统和软件系统两部分的具体组成作一简要介绍。

然后，重点讨论典型的单总线微机硬件系统结构，微处理器组织及各部分的作用，存储器组织其读写操作过程。在此基础上，将微处理器和存储器结合起来组成一个最简单的微机模型，通过具体例子说明微机的运行机理与工作过程。

第一节 微机系统组成

一、几个基本定义

微处理器、微型计算机和微型计算机系统，这是三个含义不同但又有着密切依存关系的基本概念。

（一）微处理器

微处理器简称 μp 或 MP (Microprocessor)，是指由一片或几片大规模集成电路组成的具有运算器和控制器功能的中央处理器部件，又称为微处理机。它本身并不等于微型计算机，而只是其中央处理器。有时为区别大、中、小型中央处理器 CPU (Central Processing Unit) 与微处理器，而称后者为 MPU (Microproccssing Unit)。通常，在微型计算机中直接用 CPU 表示微处理器。

（二）微型计算机

微型计算机简称 μC 或 MC (Microcomputer)，是指以微处理器为核心，配上存储器、输入/输出接口电路及系统总线所组成的计算机（又称主机或微电脑）。当把微处理器、存储器和输入/输出接口电路统一组装在一块或多块电路板上或集成在单片芯片上，则分别称之为单板、多板或单片微型计算机。

(三) 微型计算机系统

微型计算机系统简称 μ CS 和 MCS (Microcomputer system), 是指以微型计算机为中心。配以相应的外围设备、电源和辅助电路 (统称硬件) 以及指挥微型计算机工作的系统软件所构成的系统。

以上三者的含义及相互关系如图 1.1 所示。

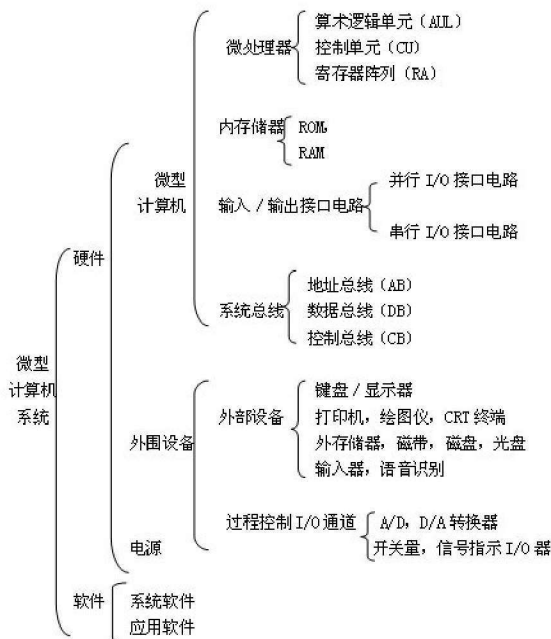


图 1.1

二、微型计算机系统的组成

微型计算机系统与任何其他计算机系统一样, 由硬件和软件两个主要部分组成。

(一) 硬件

微机硬件系统的组成如图 1.2 所示。图中, 微处理器是微机的运算、控制中心, 用来实现算术、逻辑运算, 并对全机进行控制。存储

器（简称主存或内存）用来存储程序或数据。输入/输出（I/O）芯片是微机与输入输出设备之间的接口。

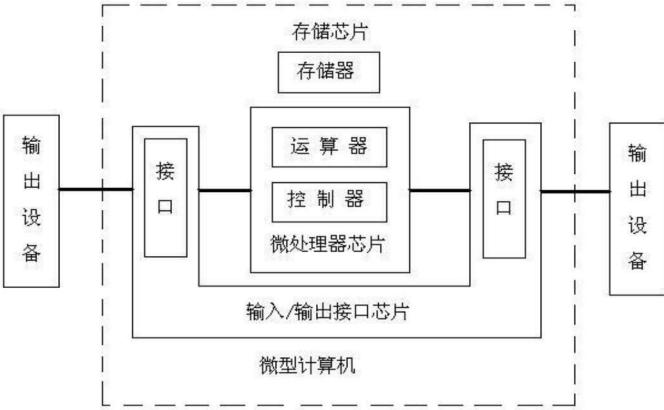


图 1.2 微机硬件系统组成

（二）软件

计算机软件通常分为两大类：系统软件 and 用户软件。系统软件是指不需要用户干预的能生成、准备和执行其他程序所需的一组程序。用户软件是使用者为解题或实现检测与实时控制等不同任务所编制的应用程序。程序的分级结构如图 1.3 所示。

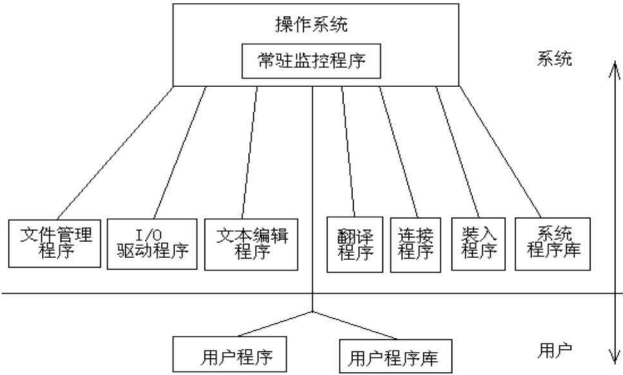


图 1.3 软件的分级结构

操作系统是一套复杂的系统软件，用于提供人机接口和管理、调度计算机的所有硬件与软件资源。其中，最为重要的核心部分是常驻监控程序。计算机开机后，常驻监控程序始终存放在内存中，它通过接收用户命令，并启动操作系统执行相应的操作。

操作系统还包括 I/O 管理程序和文件管理程序。前者用于管理 I/O 操作，后者用于管理存放在外存（或海量存储器）中的大量数据集合。每当用户程序或其他系统程序需要使用 I/O 设备时，通常并不是由这些程序自己直接执行 I/O 操作，而是通过调用操作系统 I/O 管理程序来完成所需的任务。I/O 管理程序是通过相应的 I/O 设备的驱动程序来使用硬件的。文件管理程序与 I/O 管理程序配合使用，就可完成文件的存取、复制等各种文件操作。

此外，系统软件还包括各种高级语言翻译程序、汇编程序、文本编辑程序以及辅助编写其他程序的程序。程序设计分为三级：

1. 机器语言程序设计；
2. 汇编语言程序设计；
3. 高级语言程序设计；

机器语言程序是计算机能理解和直接执行的程序。汇编语言程序是用助记符语言表示的程序，计算机不能直接“识别”，需经过称之为汇编程序的程序将其翻译为机器语言后方能执行。机器语言指令与汇编语言指令基本上一一对应，都面向机器。而高级语言是不依赖于具体机器它是一种面向应用的程序设计语言，由它所编写的程序，需经过编译程序或解释程序的翻译方能执行。

文本编辑程序是供输入或修改文本（字母、数字或标点等组成的一组字符或代码序列）用的程序，存于海量存储器中；它有几种用途，主要可用来生成程序。

系统程序库是操作系统所提供的一系列供用户使用的子程序库，而链接程序与装入程序是编译系统提供的用以将源程序文件转换成可运行的程序的工具程序。系统程序库中的子程序可以被任何系统程序或用户程序调用。把已编译的程序与程序库及其他已翻译好的程序连接起来所用的工具称为连接程序或连接编辑程序；另一种工具程序是

用来把待执行的程序送入内存，称为装入程序。有时，连接与装入功能可合成为一个程序。

以上分别谈了软件和硬件，其实一个微型计算机系统是一个有机的整体。即是说，只有软件和硬件很好地配合在一起工作时，才能最大的发挥计算机的作用。我们可以把硬件看成是微型计算机的基础，就像盖房子的地基一样，而软件则是在些基础上才能正确地运行；一个微型计算机系统有很多硬件，而助使用者还会在此基础上再增加自己需要的硬件，为了使这些硬件能够有序地工作，为使用者服务，就必须有软件来对硬件进行管理。因此，可以说，硬件是基础，软件是灵魂。微型计算机系统中缺了硬件，就会使软件无所依，而微型计算机缺了软件，只能是废铁一堆，毫无用处。随着计算机技术的发展，计算机的硬件系统和软件系统之间的分界线越来越不明显了，有一些硬件可完成软件的功能，如嵌入式系统，而一些软件同样可完成硬件的一些功能，如仿真程序。从这可以看出，两者的趋势是要相互融合，相互促进。

人是通过软件系统与硬件系统发生关系的，通常，由人使用程序设计语言编制应用程序，在系统软件的干预下使用硬件系统。

一个具体的微型计算机系统，它所包括的硬件和软件数量各不相同，究竟应包括多少，要根据应用场合对系统功能方面的要求来确定。

第二节 微机硬件系统结构

微机硬件系统结构指的是按照总体布局的设计要求，将各部件构成一个系统时的连接方式。

一种典型的微机硬件系统结构如图 1.4 所示。图中，用系统总线将各个部件连接起来。

总线是用来传送信息的公共导线，它们可以是电缆，也可以是印刷电路板（PCB）上的连线。所有的信息都通过总线传送。通常，根据所传送信息的内容与作用不同，可将总线分为三类：数据总线 DB（Data Bus）、地址总线 AB（Address Bus）、控制总线 CB（Control Bus）。

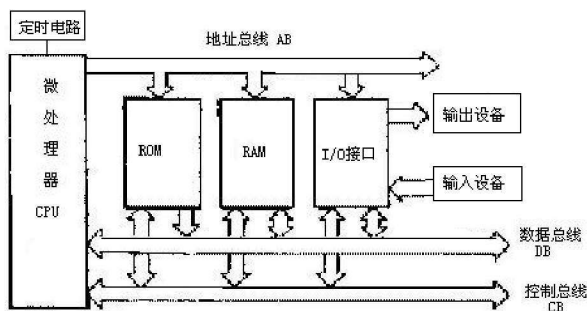


图 1.4 微机硬件系统结构

在微型计算机中有两股信息流（数据信息流和控制信息流）在流动。在单总线系统中，通过单总线实现微处理器、存储器和所有 I/O 设备之间的信息交换。由于各部件均以同一形式挂在单总线上，结构简单、易于扩充，所以目前绝大多数微机硬件系统均采用这种结构。

第三节 微处理器组成

图 1.5 给出了一个简化的微处理器结构。由图中可知，微处理器由运算器、控制器和内部寄存器阵列三部分组成。现将各部件的功能简述如下。

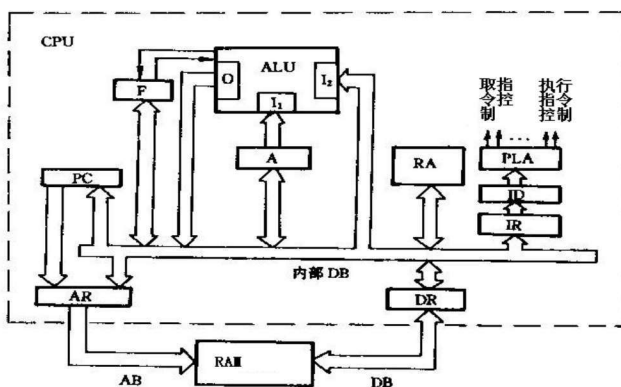


图 1.5 微处理器结构

一、运算器

运算器又称为算数逻辑单元 ALU (Arithmetic Logic Unit)，用来进行算术或逻辑运算以及位移循环等操作。参加运算的两个操作数，通常，一个来自累加器 A (Accumulator)，另一个来自内部数据总线，可以是数据寄存器 DR (Data Register) 中的内容，也可以是寄存器阵列 RA 中某个寄存器的内容。运算结果往往也送回累加器 A 保存。

二、控制器

(一) 指令寄存器 IR (Instruction Register)

指令寄存器用来存放从存储器取出的将要执行的指令（实为其操作码）。

(二) 指令译码器 ID (Instruction Decoder)

指令译码器 ID 用来对指令寄存器 IR 中的指令进行译码，以确定该指令应执行什么操作。

(三) 可编程逻辑阵列 PLA (Programmable Logic Array) (也称为定时与控制电路)

可编程逻辑阵列用来产生取指令和执行指令所需的各种微操作控制信号。由于每条指令所执行的具体操作不同，所以，每条指令将对应控制信号的某一种组合，以确定相应的操作序列。

三、内部寄存器阵列

通常，内部寄存器阵列包括若干组寄存器。

(一) 累加器 A

累加器是用得最频繁的一个寄存器。在进行算术逻辑运算时，它具有双重功能：运算前，用来保存一个操作数；运算后，用来保存结果。

(二) 数据寄存器 DR

数据寄存器 DR 用来暂存数据或指令。从存储器读出时，若读出

的是指令，经 DR 暂存的指令通过内部数据总线送到指令寄存器 IR；若读出的是数据，则通过内部数据总线送到有关的寄存器或运算器。

向存储器写入数据时，数据是经数据寄存器 DR，再经数据总线 DB 写入存储器的。

（三）程序计数器 PC（Program Counter）

程序计数器 PC 中存放着正待执行的指令的地址。根据 PC 中的地址，准备从存储器中取出将要执行的指令。通常，程序按顺序逐条执行其中的指令，因此，在任何时刻，PC 总是指向下一条将要执行的指令的起始地址。

（四）地址寄存器 AR（Address Register）

地址寄存器 AR 用来存放正要取出的指令的地址或操作数的地址。

在取指令时，将 PC 中存放的指令地址送到 AR，根据此地址从存储器中取出指令。

在取操作数时，将操作数地址通过内部数据总线送到 AR，再根据此地址从存储器中取出操作数；在向存储器存入数据时，也要先将待写入数据的地址送到 AR，再根据此地址向存储器写入数据。

（五）标志寄存器 F（Flag Register）

标志寄存器 F 用来寄存执行指令时所产生的结果或状态的标志信号。关于标志位的具体设置与功能将视微处理器的型号而异。根据检测有关的标志位是 0 或 1，可以按不同条件决定程序的流向。

第四节 存储器

一、基本概念

存储器用来存放数据和程序。在计算机内部，数据和程序都用二进制代码的形式表示。

在微机中，一般用 8 位二进制代码作为一个字节（Byte）。用一个或几字节组成一个字（Word）。一般情况下，一个字的字长是两个字节，

即 16 位二进制数。如果用字表示一个数，称为数据字；表示一条指令，称为指令字。数据字和指令字的长度可以是字的整数倍，如双字长数据，或多倍字长指令等。

微机的字长多为 16 位，即一个字，高档微机的字长可达 64 位，即 4 字长。一个存储器可划分为很多存储单元。存储单元中的内容为数据或指令。为了能识别不同的单元，我们分别赋予每个单元一个编号。这个编号称之为地址。显然，各存储单元的地址与存储单元的内容是完全不同的概念，不可混淆。

二、存储器组成

现假定存储器由 256 个单元组成，每个单元存储 8 位二进制信息，即字长为 8 位，图 1.6 示出一种随机存取存储器的结构简图（随机存取存储器 RAM（Random Access Memory）。所谓“随机存取”即所有存储单元均可随时被访问，既可以读出也可以写入信息）。这种规格的随机存取存储器，通常称为 256×8 位的存储器。

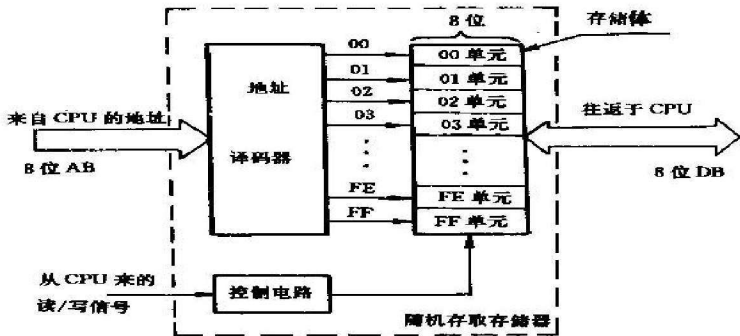


图 1.6 随机存取存储器结构简图

从图中可见，随机存取存储器由存储体、地址译码器和控制电路组成。存储体共有 256 个存储单元，其编号从 00H（十六进制表示）到 FFH，即从 00000000 到 11111111。地址译码器接收从地址总线 AB 送来的地址码，经译码器译码选中相应的某个存储单元，然后要读出或写入的数据就会出现在数据总线上，最后即可被 CPU 取走（读出），

或保存（写入）到该存储单元中。控制电路用来控制存储器的读/写操作过程。

三、读/写操作过程

从存储器读出信息的操作过程如图 1.7a 所示。假定 CPU 要读出存储器 04H 单元的内容 10010111 即 97H，则：

- （1）CPU 的地址寄存器 AR 先给出地址 04H 并放到地址总线上，经地址译码器译码选中 04H 单元；
- （2）CPU 发出“读”控制信号给存储器，指示它准备把被寻找的 04H 单元中的内容 97H 放到数据总线上；
- （3）在读控制信号的作用下，存储器将 04H 单元中的内容 97H 放到数据总线上，经它送至数据寄存器 DR，然后由 CPU 取走该内容作为所需要的信息使用。

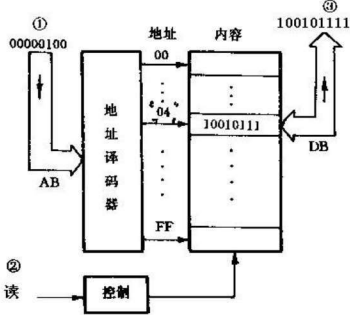


图 1.7 a

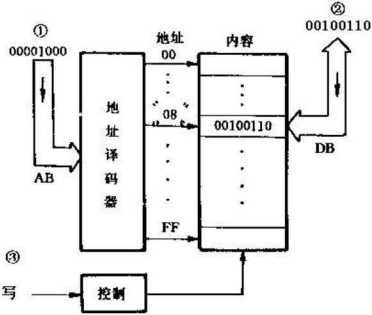


图 1.7 b

应当指出，读操作完成后，04H 单元中的内容 97H 仍保持不变，这种特点称为非破坏性读出（NDRO—Non Destructive Read Out）。

向存储器写入信息的操作过程如图 1.7b 所示。假定 CPU 要把数据寄存器 DR 中的内容 00100110 即 26H 写入存储器 08H 单元，则：

- （1）CPU 的地址寄存器 AR 先把地址 08H 放到地址总线上，经地址译码器选中 08H 单元；
- （2）CPU 把数据寄存器中的内容 26H 放到数据总线上；
- （3）CPU 向存储器发送“写”控制信号，在该信号的控制下，将

内容 26H 写入被寻址的 08 单元。

第五节 微机工作过程

微机的工作过程就是执行程序的过程，而程序由指令序列组成，因此，执行程序的过程，就是执行指令序列的过程，即逐条地执行指令，由于执行每一条指令，都包括取指令与执行指令两个基本阶段，所以，微机的工作过程，也就是不断地取指令和执行指令的过程。微机执行程序过程示意图如图 1.8 所示。

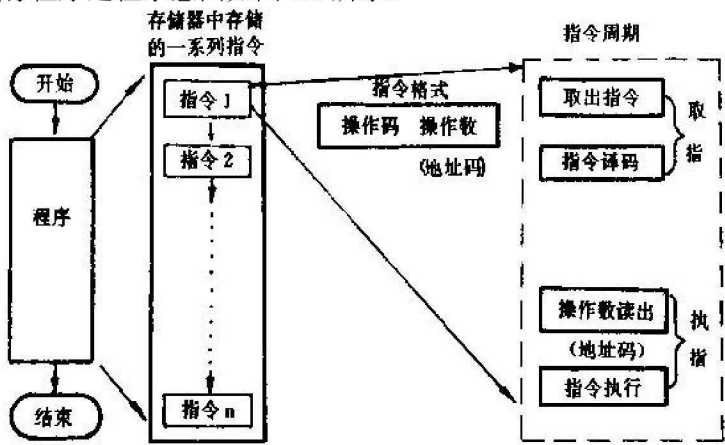


图 1.8 程序执行过程示意图

假定程序已由输入设备存放到内存中。当计算机要从停机状态进入运行状态时，首先应把第一条指令所在的地址赋给程序计数器 PC，然后机器就进入取指阶段。在取指阶段，CPU 从内存中读出的内容必为指令，于是，数据寄存器 DR 便把它送至指令寄存器 IR；然后由指令译码器译码，控制器就发出相应的控制信号，CPU 便知道该条指令要执行什么操作，在取指阶段结束后，机器就进入执行指令的阶段，这时，CPU 执行指令所规定的具体操作。当一条指令执行完毕以后，就转入了下一条指令的取指阶段。这样周而复始地循环一直进行到程序中遇到暂停指令时方才结束。

取指阶段都是由一系列相同的操作组成的，所以，取指阶段的时间总是相同的，它称为公操作。而执行指令阶段将由不同的事件顺序组成，它取决于被执行指令的类型，因此，执行指令阶段的时间从一条指令到下一条指令变化相当大。

应当指出的是，指令通常包括操作码（Operation Code）和操作数（Operand）两大部分。操作码表示计算机执行什么样的具体操作，而操作数表示参加操作的数的本身或操作数所在的地址，也称之为地址码。在 8 位机中，由于一个存储单元只能存放一个字节，而指令根据其所含内容不同而有单字节、双字节、三字节用至最多四字节之分，因此，在执行一条指令时，就可能要处理 1—4 个不等字节数目的代码信息，包括操作码、操作数或操作数的地址。

为了进一步说明微机的工作过程，我们来具体讨论一个模型机怎样执行一段简单的程序。例如，计算机如何具体计算 $3+2=?$ 虽然这是一个相当简单的加法运算，但是，计算机却无法理解。人们必须要先编写一段程序，以计算机能够理解的语言告诉它如何一步一步地去做，直到每一个细节都详尽无误，计算机才能正确地理解与执行。

在编写程序之前，必须首先查阅所使用的微处理器的指令表（或指令系统），它是某种微处理器所能执行的全部操作命令汇总。不同系统的微处理器各自具有不同的指令表。假定查到模型机的指令表中可以用三条指令求解这个问题。表 1.1 示出了这三条指令及其说明。

表中第一列为指令的名称。编写程序时，写指令的全名是不方便的，因此，人们给每条指令规定了一个缩写词，称之为助记符，第二列即助记符。第三列为机器码，机器码用二进制或十六进制两种形式表示（注：计算机只识别二进制数，用十六进制表示，是为了缩短长度），计算机和程序员用它来表示指令。最后一列，确切地说明了执行一条指令时所完成的具体操作。