21世纪高等学校计算机规划教材

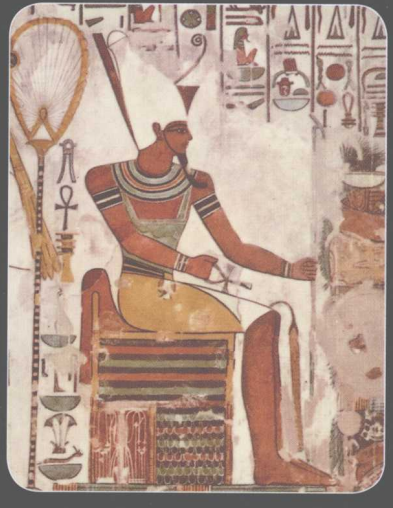
21st Century University Planned Textbooks of Computer Science

单片机原理与技术

Fundamentals of Single-Chip Computers & Technology

周明德 编著

- 强调80C51系列的基础知识
- 突出单片机实际应用
- 提供典型程序对应的C语言代码



名家系列

人民邮电出版社
POSTS & TELECOM PRESS

21世纪高等学校计算机规划教材

21st Century University Planned Textbooks of Computer Science

单片机原理与技术

Fundamentals of Single-Chip Computers & Technology

周明德 编著

名家系列

人民邮电出版社

北京

图书在版编目（CIP）数据

单片机原理与技术 / 周明德编著. —北京：人民邮电出版社，2008.4
21 世纪高等学校计算机规划教材
ISBN 978-7-115-17414-7

Ⅰ. 单… Ⅱ. 周… Ⅲ. 单片微型计算机—高等学校—教材 Ⅳ. TP368.1

中国版本图书馆 CIP 数据核字（2008）第 000461 号

内 容 提 要

本书系统介绍了 80C51 系列单片机的基本工作原理、接口及应用技术。主要包括计算机基础知识、80C51 单片机的体系结构、存储器，指令系统、汇编语言程序设计、并行端口、总线与时序、中断、定时器/计数器、串行通信、抗干扰技术和单片机系统设计等内容。

本书可作为高等院校信息工程、通信工程、电气工程、自动化、计算机应用、机电等专业的教材。

21 世纪高等学校计算机规划教材

单片机原理与技术

◆ 编　　著　周明德
责任编辑　邹文波

◆ 人民邮电出版社出版发行　　北京市崇文区夕照寺街 14 号
邮编　100061　　电子函件　315@ptpress.com.cn
网址　http://www.ptpress.com.cn
北京鸿佳印刷厂印刷
新华书店总店北京发行所经销

◆ 开本：787×1092　1/16
印张：18.5
字数：482 千字　　2008 年 4 月第 1 版
印数：1 – 4 000 册　　2008 年 4 月北京第 1 次印刷

ISBN 978-7-115-17414-7/TP

定价：29.80 元

读者服务热线：(010)67170985　印装质量热线：(010)67129223
反盗版热线：(010)67171154

前 言

单片微型计算机简称为单片机，又称为微控制器，是微型计算机的一个重要分支。随着应用的需要与集成电路技术的发展，单片机由 4 位、8 位、16 位发展到 32 位甚至 64 位，功能与性能有了极大的提高与扩展。目前应用最多的仍是如 Intel 80C51 类的 8 位机，但随着手机、数字电视、PDA 等应用的迅速普及，32 位等高档单片机也取得快速发展。

80C51 是高档单片机的基础，因此本书重点介绍 80C51 单片机。全书以 80C51 系列单片机为主线，全面而系统地介绍了 80C51 单片机应用系统的结构，原理和应用。全书共分 11 章，各章内容安排如下。

第 1 章主要介绍微型机基础以及 80C51 单片机的体系结构。

第 2 章主要介绍存储器技术及 80C51 单片机的存储器组织。

第 3 章主要介绍 MCS-51 指令系统。

第 4 章主要介绍汇编语言程序设计。

第 5 章主要介绍 80C51 的并行端口，包括 80C51 的基本输入/输出、简单的人机接口等。

第 6 章介绍 80C51 的总线、时序与总线扩展技术。

第 7 章介绍中断技术及 80C51 的中断结构。

第 8 章主要介绍定时器/计数器的原理与技术。

第 9 章主要介绍串口通信技术。

第 10 章主要介绍抗干扰技术，包括软件抗干扰技术与硬件抗干扰技术两方面的内容。

第 11 章主要介绍单片机系统设计，并以“单片机控制交流变频调速系统”实例讲述具体的实现方法。

在本书的编写过程中，笔者主要做了以下考虑：

1．以 80C51 系列单片机入手，讲清计算机的 CPU 结构、指令系统、汇编语言、存储器、I/O 端口、并行/串行通信、中断等的原理与技术；

2．以主要篇幅讲透 51 系列，即 8 位单片机。以较少的篇幅讲其扩展与发展；

3．以现代计算机的观点讲 51 系列，不过多涉及历史，不讲过时的观点与内容；

4．全书以约 1/3 的篇幅讲应用，但应用不是简单的罗列，而是着重于一些重要的应用接口，如显示与键盘接口、红外遥控器接口、数字控制器接口、伺服驱动器接口、数字仪表接口等，最后举 1～2 个综合性的应用例子；

5．在程序设计方面，考虑到学习的难度，部分程序给出了对应的 C 语言程序代码，将有助于提高学生编写单片机程序的能力。

本书可作为高等院校信息工程、通信工程、电气工程、自动化、计算机应用、机电等专业的教学用书。

由于时间仓促和水平所限，书中难免存在一些错误和不妥之处，敬请读者批评指正。

周明德

2007 年 12 月

目 录

第1章 概　述

1981 年 IBM 公司进入了微型机领域并推出了 IBM-PC 以后，计算机的发展开创了一个新的时代—微型机时代。微型计算机（简称微型机）的迅速普及，使计算机真正广泛应用于工业、农业、科学技术领域以及社会生活的各个方面。随着微型机应用的普及及技术的发展，微型机功能已经远远超过了 20 世纪 80 年代以前的中、小型机甚至超过了大型机。

随着超大规接集成电路技术的发展，将 CPU、存储器以及各种外部设备的接口做在一块集成电路芯片上，构成了一个计算机，这样就将其称为单片机。单片机常常嵌入至仪器、仪表，各种智能设备与应用系统中，称其为嵌入式应用。

随着应用的需要与集成电路技术的发展，单片机也由 4 位、8 位、16 位发展到 32 位甚至 64 位，功能与性能有了极大的提高与扩展。随着手机、数字电视、PDA 等应用的迅速普及，32 位等高档单片机也得到快速发展。但目前，单片机应用的主流仍是 8 位单片机，尤其是 Intel 8051 系列，它们是高档单片机的基础，故本书重点介绍 Intel 8051 系列。

1.1 计算机基础

1.1.1 计算机的基本结构

自计算机诞生以来，经历了电子管、晶体管、小规模集成电路和超大规模集成电路等四代，计算机的规模、运行速度、用途等有极大的不同。但从计算机的原理来看，计算机的基本结构未变，如图 1-1 所示。

计算机最早是作为运算工具出现的，显然，它首先要有能进行运算的部件，称之为运算器；其次要有能记忆原始题目、原始数据和中间结果以及为了使机器能自动进行运算而编制的各种命令，将这种器件称为存储器；再次，要有能代替人起控制作用的控制器，它能根据人们事先给定的命令发出各种控制信息，使整个计算过程能一步步地自动进行。但是，仅有这三部分还不够，原始的数据与命令要输入，所以需要有输入设备；而计算的结果（或中间过程）需要输出，就要有输出设备，这样就构成了一个基本的计算机系统，

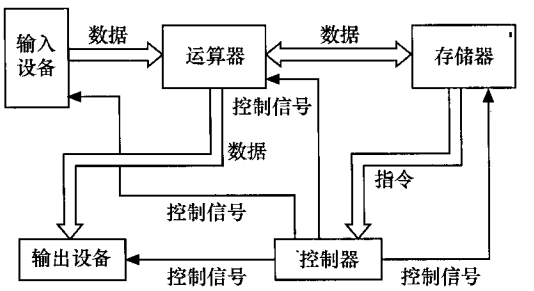


图 1-1　计算机的基本结构图

如图 1-1 所示。

计算机要能对数进行运算，首先要能识别数（或说计算机要能表示数）。由于二进制数的表示简单、方便、可靠，二进制数的运算也是最简单的，故现代计算机中都以二进制表示数。若各种信息（例如字母、各种符号、汉字、声音、图像等）也能以若干位的二进制组合（即二进制编码）表示，计算机就能正确地识别这些信息，另外再加上进行数的运算的运算器，这样除了能作算术运算外还能进行逻辑运算（逻辑操作），于是运算器就能进行信息处理。存储器能存放信息，输入/输出设备能输入/输出信息。于是，原来作为数运算的计算机就成为信息处理机（信息处理的工具），从而大大扩展了计算机的应用领域。

在计算机中，基本上有两种信息在流动，一种为数据，即各种原始数据、中间结果、程序等，这些数据由输入设备输入至运算器，再存于存储器中；在运算处理过程中，数据从存储器读入运算器进行运算；运算的中间结果要存入存储器中，或最后由运算器经输出设备输出。人向计算机发出的各种命令（即程序），也以数据的形式在计算机运行之前，存放在存储器中。计算机启动后由存储器送入控制器，由控制器经过译码后变为各种控制信号。故有另一种信息流为控制命令，由控制器控制输入装置的启动或停止；控制运算器按规定一步步地进行各种运算和处理；控制存储器的读和写；控制输出设备输出结果等。所以，一台计算机或称为信息处理系统的功能就是要能表示与识别信息、加工与处理信息（通过运算器）、存放信息（通过存储器）与传送信息（通过输入/输出设备与网络）。学习与掌握计算机就是要了解信息是如何表示（识别）、存储、处理与传送。

图 1-1 所示各部分构成了计算机硬件（Hardware）。在计算机硬件中，人们往往把运算器、控制器和存储器合在一起称为计算机的主机；而把各种输入/输出设备统称为计算机的外围设备（或称外部设备——peripheral）。

在主机部分中，又把运算器和控制器合在一起称为中央处理单元——CPU（central processing unit）。随着半导体集成电路技术的发展，可以把整个 CPU 集成在一个集成电路芯片上，称为微处理器（microprocessor）。现在在市场上销售的 Intel 公司的奔腾芯片（Pentium II、Pentium III 和 Pentium 4）以及 AMD 公司的速龙等 x86 系列都是这样的微处理器，它们从功能上说是一个中央处理单元（运算器与控制器的集合）。以微处理器（CPU）为核心加上一定数量的存储器以及若干个外部设备（通过 I/O 接口芯片与 CPU 接口），就构成了微型机。早期（例如 1981 年刚推出的 IBM-PC），由于 CPU 的速度较低（当时 CPU 的工作频率为 5MHz），内存容量较小（例如 128KB），外部设备的数量很少，只能用于处理个人事务，故称之为个人计算机——PC（personal computer）。目前，人们仍把这种微型机称之为 PC，但实际上 CPU 的工作频率已超过 1GHz，内存容量已为几百兆字节，硬盘容量已为几十吉字节，其性能已远远超过 20 世纪 80 年代的大型机。

总之，人们把以微处理器为核心构成的计算机，称为微型机，最典型的就是上述的 PC。若内存的容量较小，输入/输出设备少，整个计算机可以安装在一块印刷电路板上，这样的计算机，称为单板计算机。若能把整个计算机集成在一个芯片上，就称为单片机。

不论计算机的规模大小，CPU 只是计算机的一个部件，如图 1-1 所示，有 CPU，有存储器，有输入/输出设备，才称之为计算机。

单片机，例如 8051 系列的芯片上有 CPU，有一定容量的存储器，还有串行与并行接口、有定时器/计数器，有的单片机中还有 A/D 与 D/A 转换器等，它是一个完整的计算机（当然，它们还可以扩展）。

1.1.2　常用的名词术语和二进制编码

1. 位、字节、字及字长

位、字节、字及字长是计算机常用的名词术语。

- 位（bit）。“位”指一个二进制位。它是计算机中信息存储的最小单位。
- 字节（Byte）。“字节”指相邻的 8 个二进制位。1024 个字节构成 1 个千字节，用 KB 表示。1024KB 构成 1 个兆字节，用 MB 表示。1024MB 构成 1 个千兆字节，用 GB 表示。B、KB、MB、GB 都是计算机存储器容量的单位。
- 字（Word）和字长。“字”是计算机内部进行数据传递处理的基本单位。通常它与计算机内部的寄存器、运算装置、总线宽度相一致。

一个字所包含的二进制位数称为字长。常见的微型机的字长，有 8 位、16 位、32 位和 64 位之分。

但是，目前在 PC 中，把字（word）定义为 2 字节（16 位），双字（double word）为 4 字节（32 位），四字（quad word）为 8 字节（64 位）。

2. 数字编码

由于二进制有很多优点，所以计算机中的数用二进制表示，但人们与计算机打交道时仍习惯用十进制，在输入时计算机自动将十进制转换为二进制，而在输出时又将二进制转换为十进制。为便于机器识别和转换，计算机中的十进制数的每一位用二进制编码表示，这就是所谓的十进制数的二进制编码，简称二—十进制编码（BCD 码）。

二—十进制编码的方法很多，最常用的是 8421 BCD 码。8421 BCD 码有 10 个不同的数字符号，逢 10 进位，每位用四位二进制表示。例如：

83.123 对应的 8421 BCD 码是 1000 0011.0001 0010 0011。

同理，111 1001 0010.0010 0101 BCD 对应的十进制数是 792.25。

3. 字符编码

字母、数字、符号等各种字符也必须按特定的规则用二进制编码才能在计算机中表示。字符编码的方式很多，世界上最普遍采用的一种字符编码是 ASCII。

ASCII 用七位二进制编码，它有 128 种组合，可以表示 128 种字符，包括 10 个阿拉伯数字字符（0～9），大、小写英文字母（52 个）等。在计算机中用一个字节表示一个 ASCII 字符，最高位置为 0。例如，00110000～00111001（即 30H～39H）是数字 0～9 的 ASCII，而 01000001～01011010（即 41H～5AH）是大写英文字母 A～Z 的 ASCII。

4. 汉字编码

用计算机处理汉字，每个汉字必须用代码表示。键盘输入汉字是输入汉字的外部码。外 部码必须转换为内部码才能在计算机内进行存储和处理。为了将汉字以点阵的形式输出，还要将内部码转换为字形码。不同的汉字处理系统之间交换信息采用交换码。

（1）外部码

汉字主要是从键盘输入，每个汉字对应一个外部码，外部码是计算机输入汉字的代码，是代表某一个汉字的一组键盘符号。外部码也叫输入码。汉字的输入方法不同，同一个汉字的外部码可能就不一样。目前已有数百种汉字外部码的编码方案，大致可以归纳为 4 种类型：数字码、音码、形码和音形码。

数字码是将汉字按某种规律排序，然后赋予它们数字编号，这个数字编号就作为汉字的编码。

常见的数字码，如区位码等。这种编码方法无重码，可以找到其他编码方法难于找到的汉字，但是难于记忆，要有手册备查。

音码是以汉语拼音作为汉字的编码，只要学习过汉语拼音，一般不需要经过专门训练就可以掌握，但是，用拼音方法输入汉字同音字多，需要选字，影响输入速度，不知道读音的汉字也无法输入。

形码是把一个汉字拆成若干偏旁、部首、字根，或者拆成若干种笔画，使偏旁、部首、字根或笔画与键盘对应编码，按字形敲键输入汉字。形码输入汉字重码率低、速度快，只要能看到的字形就可以拆分输入，但是必须经过专门训练，并需大量记忆编码规则和汉字拆分原则。最常见的形码方案有五笔字型码等。

音形码是拼音和字形相结合的一种汉字编码方案，如自然码、钱码等。

（2）内部码

汉字内部码也称汉字内码或汉字机内码。在不同的汉字输入方案中，同一汉字的外部码不同，但同一汉字的内部码是唯一的。内部码通常是用其在汉字字库中的物理位置表示，可以用汉字在汉字字库中的序号或者用汉字在汉字字库中的存储位置表示。汉字在计算机中至少要用两个字节表示（有用三字节、四字节表示的）。在微型机中常用的是两字节汉字内码。两字节汉字内码，就是汉字的国标码（用两个 7 位编码）的两个字节的最高位都改为“1”形成的。例如汉字“啊”，国标码为 0110000、0100001，即 30H，21H。内码为 10110000、10100001，即 B0H，A1H。在计算机中通常处理的是以 ASCII 表示的字符，一个字符在机器内以一个字节的二进制编码表示。实际上 ASCII 只需 7 位，故在计算机内的字符编码的最高位是“0”。由此可见，以字节的最高位是 0 还是 1，很容易区分是 ASCII 字符还是汉字。

（3）交换码

计算机之间或计算机与终端之间交换信息时，要求其间传送的汉字代码信息完全一致。为此，国家根据汉字的常用程度定出了一级和二级汉字字符集，并规定了编码，这就是国标 GB 2312—1980《信息交换用汉字编码字符集基本集》。GB 2312—1980 中汉字的编码即国标码。该标准编码字符集共收录汉字和图形符号 7 445 个，其中包括：

① 一般符号 202 个，包括间隙符、标点、运算符、单位符号、制表符等；

② 序号 60 个，它们是 1～20（20 个）、（1）～（20）（20 个）、①～⑩（10 个）和（一）～（十）（10 个）；

③ 数字 22 个，它们是 0～9 和Ⅰ～Ⅻ；

④ 英文字母 52 个，大、小写各 26 个；

⑤ 日文假名 169 个，其中平假名 83 个，片假名 86 个；

⑥ 希腊字母 48 个，其中大、小写各 24 个；

⑦ 俄文字母 66 个，其中大、小写各 33 个；

⑧ 汉语拼音符号 26 个；

⑨ 汉语注音字母 37 个；

⑩ 汉字 6 763 个，这些汉字分为两级，第一级汉字 3 755 个，第二级汉字 3 008 个。

这个字符集中的任何一个图形符号及汉字都是用两个 7 位的字节表示（在计算机中当然用两个 8 位字节，每个字节的最高位为 1 来表示）。其中汉字占 6 763 个，第一级汉字 3 755 个，按汉语拼音字母顺序排列，同音字以笔画顺序为序；第二级汉字 3 008 个，按部首顺序排列。GB 2312—1980 中，7 445 个字符和汉字分布在 87 个区中，每区最多 94 个字符。分布情况如下：

1 ~ 9 图形字符；

10 ~ 15 空间未用；

16 ~ 55 一级汉字；

56 ~ 87 二级汉字。

在 GB 2312—1980 标准中，对每个图形字符或汉字给出了两种汉字代码：一种是用两个字节二进制数给出的国标码（即内部码中所用到的）；另一种是四位十进制的区位码，其中高二位是某字符或汉字所在的区号，低二位是在区中的位置号。例如“啊”字的国标码是 3021H，区位码是 1601D。

随着计算机应用的扩展，GB 2312 中的 6 000 多个汉字已远远不能满足需要，从而制定了 GB 18030、GB 13000 等标准。

（4）输出码

汉字输出码又称汉字字形码或汉字发生器的编码。众所周知，汉字无论字形有多少变化，也无论笔画有多有少，都可以写在一个方块中；一个方块可以看做 m 行 n 列的矩阵，称为点阵。一个 m 行 n 列的点阵共有 $m \times n$ 个点。例如 16 × 16 点阵的汉字，共有 256 个点。每个点可以是黑点或非黑点，凡是笔画经过的点用黑点，于是利用点阵描绘出了汉字字形。汉字的点阵字形在计算机中称为字模，图 1-2 所示为汉字“中”的 16 × 16 点阵字模。

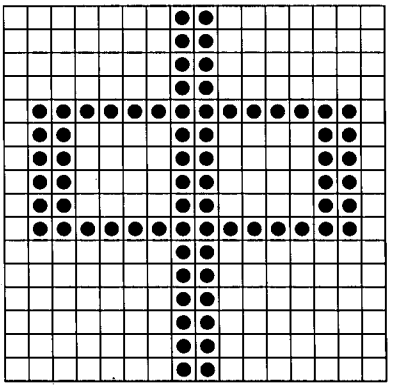


图 1-2　汉字“中”的 16 × 16 点阵字模

在计算机中用一组二进制数字表示点阵，用二进制数 1 表示点阵中的黑点，用二进制数 0 表示点阵中的非黑点。一个 16 × 16 点阵的汉字可以用 16 × 16=256 位的二进制数来表示，这种用二进制数表示汉字点阵的方法称为点阵的数字化。汉字字形经过点阵的数字化后转换成一串数字，称为汉字的输出码。

同一汉字的输出码，即字形码，因选择点阵的不同而不同。一个字节含 8 个二进制位，所以 16 × 16 点阵汉字需要 2 × 16=32 个字节表示；24 × 24 点阵汉字需要 3 × 24=72 个字节表示；32 × 32 点阵汉字需要 4 × 32=128 个字节表示。点阵的行列数越多，所描绘的汉字越精细，但占用的存储空间也越多。16 × 16 点阵基本上能表示 GB 2312—1980 中的所有简体汉字。24 × 24 点阵则能表示宋体、楷体、黑体等多字体的汉字。这两种点阵是比较常用的点阵，前一种一般用于显示，而后一种一般用于打印，除此之外还有 32 × 32、40 × 40、48 × 48、64 × 64、72 × 72、96 × 96、108 × 108 等点阵，主要用于印刷。

1.1.3　指令程序和指令系统

在 1.1.1 小节中我们提到了计算机的几个主要部分，这些构成了计算机的硬件基础，也是计算机进行计算的物质条件。但是，光有这样的硬件，还只是具有了计算的可能。要使计算机真正能够进行计算，还必须使这些硬件按照人的要求运行起来，这就必须要有软件的配合，首先就是各种程序（program）。

我们知道，计算机所以能脱离人的直接干预，自动地进行计算，这是由于人把实现这个计算的一步步操作用命令的形式——即一条条指令（instruction）预先输入到存储器中，在运行时，机器把这些指令一条条地取出来，加以翻译和执行。

就拿两个数相加这一最简单的运算来说，就需要以下几步（假定要运算的数已在存储器中）。

第一步：把第一个数从它所在的存储单元（location）中取出来，送至运算器；

第二步：把第二个数从它所在的存储单元中取出来，送至运算器；

第三步：相加；

第四步：把相加的结果，送至存储器中指定的单元。

所有这些取数、送数、相加、存数等都是一种操作，我们把要求计算机执行的各种操作用命令的形式写下来，这就是指令。通常一条指令对应着一种基本操作，但是，计算机怎么能辨别和执行这些操作呢？这是由设计时设计人员赋予它的指令系统决定的；一个计算机能执行什么样的操作，能做多少种操作，是由设计计算机时所规定的指令系统决定的。计算机所能执行的全部指令，就是计算机的指令系统（instruction set），这是计算机所固有的。

在使用计算机时，必须把要解决的问题编成一条条指令，但是这些指令必须是所用的计算机能识别和执行的指令，也即每一条指令必须是一台特定的计算机的指令系统中具有的指令，而不能随心所欲。这些指令的集合就称为程序。用户为解决自己的问题所编写的程序，称为源程序（source program）。

指令通常分成操作码（opcode，即 operation code）和操作数（operand）两大部分。操作码表示计算机执行什么操作；操作数是此指令要操作的对象。指令中的操作数部分常规定参加操作的数的本身或操作数所在的地址。

因为计算机只能识别二进制数码，所以计算机的指令系统中的所有指令，都必须以二进制编码的形式来表示。例如在 Intel 8086 中，从存储区取数（以 SI 变址寻址）至累加器 AL 中的指令的编码为 8A04H（两字节指令），一种加法指令的编码为 02C3H，向存储器存数（一种串操作指令）的编码为 AAH（一字节指令）等。这就是指令的机器码（machine code）。一个字节的编码能表达的范围（256 种）较小，不能充分表示各种操作码和操作数。所以，有一字节指令，有两字节指令，也有多字节指令如四字节指令。

计算机发展的初期，就是用指令的机器码直接来编制用户的源程序，这就是机器语言阶段。但是机器码是由一连串的 0 和 1 组成的，没有明显的特征，不好记忆，不易理解，易出错。所以，编程序成为一种十分困难、十分烦琐的工作。因而，人们就用一些助记符（mnemonic）—通常是指令功能的英文词的缩写来代替操作码。如在 x86 系列中，数的传送指令用助记符 MOV（MOVE 的缩略），加法用 ADD，转移用 JMP 等。这样，每条指令有明显的特征，易于理解和记忆，也不易出错，比机器语言前进了一大步，此阶段称为汇编语言阶段。该阶段用户使用汇编语言（操作码用助记符代替，操作数也用一些符号——symbol 来表示）来编写源程序。

要求机器能自动执行这些程序，就必须把这些程序预先存放到存储器的某个区域。程序通常是顺序执行的，所以程序中的指令也是一条条顺序存放的。计算机在执行时要能把这些指令一条条取出来加以执行，必须要有一个电路能追踪指令所在的地址，这就是程序计数器（program counter，PC）。在开始执行时，给 PC 赋予程序中第一条指令所在的地址，然后每取出一条指令（确切地说是每取出一个指令字节）PC 中的内容自动加 1，指向下一条指令的地址（address），以保证指令的顺序执行。只有当程序中遇到转移指令、调用子程序指令或遇到中断时，PC 才把控制转到所需要的地方去。

1.1.4 初级计算机

当开始接触计算机内部结构时，一个实际的微型机结构就显得太复杂了，会使人抓不住基本

部件、基本概念和基本工作原理。因此，先从一个以实际结构为基础、经过简化的模型机着手来分析基本原理，然后加以扩展，再回到实际结构中去。

图 1-3 所示为微型机的结构图。它是由微处理器（CPU）、存储器、接口电路等组成，通过三条总线（BUS）——地址总线（address bus）、控制总线（control bus）和双向数据总线（data bus）来连接。为了简化问题，我们先不考虑外部设备以及接口电路，认为要执行的程序以及数据，已存入存储器内。

1. CPU 的结构

一个模型机的 CPU 结构，如图 1-4 所示。

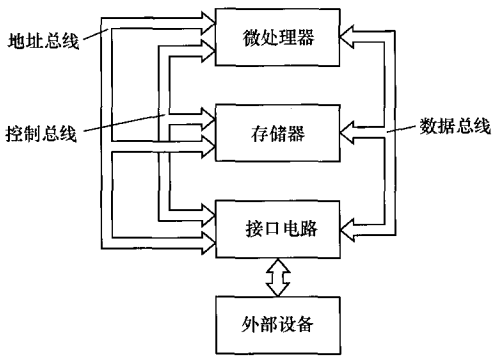


图 1-3　微型机结构图

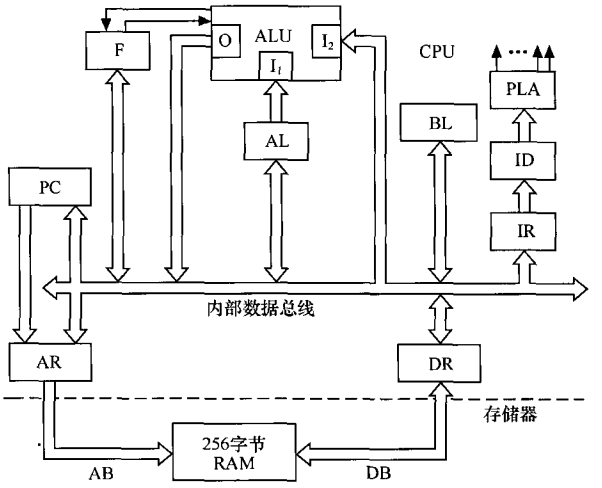


图 1-4　一个模型机的 CPU 结构

算术逻辑单元（arithmetic logic unit，ALU）是执行算术和逻辑运算的装置，它以累加器（accumulator，AL）的内容作为一个操作数；另一个操作数由内部数据总线供给，可以是寄存器（Register）BL 中的内容，也可以是由数据寄存器（data register，DR）供给的由内存读出的内容等；操作的结果通常放在 AL 中。

F（Flag）是标志寄存器，由一些标志位组成。它反映运算结果的一些特征，例如，结果是否为零、结果的正负、结果是否产生进位或借位等，详细的功用我们在后面分析。

要执行的指令的地址由程序计数器（PC）提供，AR（address register）是地址寄存器，由它把要寻址的单元的地址（可以是指令——地址由 PC 提供；也可以是数据——地址要由指令中的操作数部分给定）通过地址总线送至存储器。

从存储器中取出的指令，由数据寄存器送至指令寄存器（instruction register，IR），经过指令译码器（instruction decoder，ID）译码，通过控制电路，发出执行一条指令所需要的各种控制信息。

在模型机中，字（word）长（通常是以存储器一个单元所包含的二进制信息的位数表示的）为 8 位，即为一个字节（在字长较长的机器中为了表示方便，把 8 位二进制位定义为一个字节），故 AL、BL、DR 都是 8 位的，因而双向数据总线也是 8 位的。在模型机中又假定内存为 256 个单元，为了能寻址这些单元，则地址也需 8 位（2^8=256），可寻址 256 个单元。因此，这里的 PC 及 AR 也都是 8 位的。

在 CPU 内部各个寄存器之间及 ALU 之间数据的传送也采用内部总线结构，这样扩大了数据传送的灵活性，减少了内部连线，因而减少了这些连线所占的芯片面积，但是采用总线结构，则在任一瞬时，总线上只能有一个信息在流动，因而使速度降低。

2. 存储器

模型机的存储器结构如图 1-5 所示。

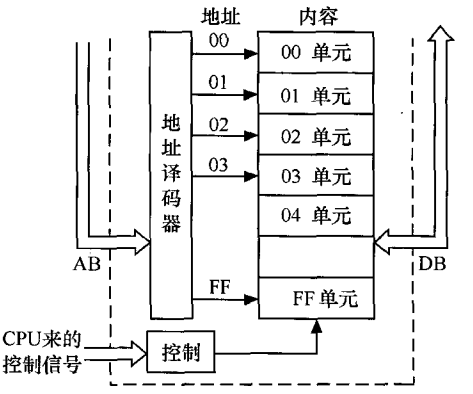


图 1-5 模型机的存储器结构图

存储器的主体是存放信息的单元（location），在图 1-5 所示的模型机中，它由 256 个单元组成。每个存储单元由若干个二进制位组成。目前，在微机中，一个基本存储单元为 8 位（一个字节），若要存放 16 位二进制数就要占用两个存储单元。为了能区分不同的单元，对这些单元分别编了号，称为存储单元的地址。地址也要用二进制编码表示，由存储单元的数量确定表示地址的位数。在图 1-5 所示的模型机中，假定的存储单元是 256，故要用 8 位二进制数也即 2 位十六进制数表示地址，如 00，01，02，…，FF 等；而每一个单元可存放 8 位二进制信息（通常也用 2 位十六进制数表示），这就是它们的内容。

每一个存储单元的地址和这个地址中存放的内容这两者是完全不同的，千万不要混淆了。

存储器中的不同存储单元，是由地址总线上送来的地址（8 位二进制数），经过存储器中的地址译码器来寻找（区分）的（每给定一个地址号，可从 256 个单元中找到相应于这个地址号的某一单元），然后就可以对这个单元的内容进行读或写的操作。

（1）读操作

若已知在 04 号存储单元中，存的内容为 10000100（即 84H），要把它读出至数据总线上，则要求 CPU 的地址寄存器先给出地址号 04，然后通过地址总线送至存储器，存储器中的地址译码器对它进行译码，找到 04 单元，再要求 CPU 发出读的控制命令，于是 04 号单元的内容 84H 就出现在数据总线上，由它送至数据寄存器（DR），如图 1-6 所示。现代存储器当信息从存储单元读出后，存储单元的内容并不改变，只有当把新的信息写入该单元时，才由新的代替旧的。

（2）写操作

若要把数据寄存器中的内容 26H 写入到 10 号存储单元，则要求 CPU 的 AR 地址寄存器先给出地址 10，通过地址总线（AB）送至存储器，经译码后找到 10 号单元；然后把数据寄存器（DR）中的内容 26H 经数据总线（DB）送给存储器；且 CPU 发出写的控制命令，于是数据总线上的信息 26H 就可以写入到 10 号单元中，如图 1-7 所示。

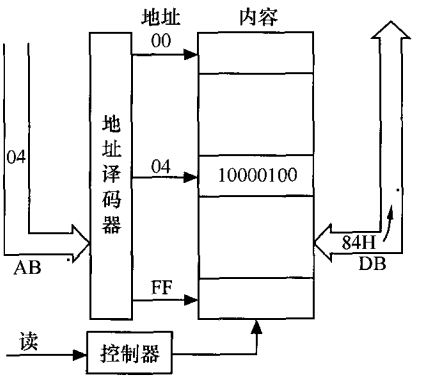


图 1-6 存储器读操作示意图

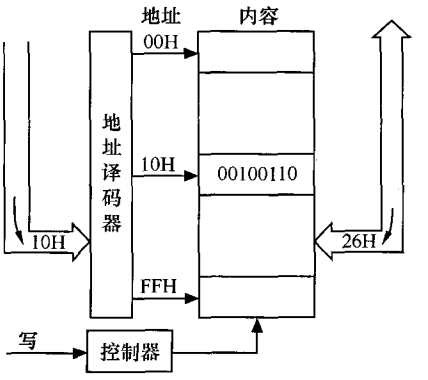


图 1-7 存储器写操作示意图

信息写入后，在没有新的信息写入以前一直是保留的。

3. 执行过程

若程序已存放在内存中，大部分 8 位机执行过程就是取出指令和执行指令这两个阶段的循环。

机器从停机状态进入运行状态，要把第一条指令所在的地址赋予 PC，然后就进入取指（取出指令）阶段。在取指阶段从内存中读出的内容必为指令，所以 DR 把它送至 IR，然后由指令译码器译码，就知道此指令要执行什么操作，在取指阶段结束后就进入执行阶段。当一条指令执行完以后，就进入了下一条指令的取指阶段。这样的循环一直进行到程序结束（遇到停机指令）。

1.1.5　简单程序举例

下面以一个简单的例子来说明程序执行的过程。

若要求机器把两个数 7 和 10 相加。在编程序时首先要查一下机器的指令系统，看机器能用什么指令完成这样的操作。查到可用表 1-1 所示的 3 条指令完成两数相加的操作。

表 1-1　　完成两数相加的指令

名　称	助 记 符	操 作 码	说　明
立即数取入累加器	MOV AL，n	10110000　B0 n　　n	这是一条两字节指令，把指令第二字节的立即数 n 送累加器 AL
加立即数	ADD AL，n	00000100　04 n　　n	这是一条两字节指令，累加器 AL 中的内容与指令第二字节的立即数相加，结果在 AL 中
停　机	HLT	11110100　F4	停止操作

用助记符形式表示的程序为：

```
MOV  AL, 7
ADD      AL, 10
HLT
```

但是，模型机不识别助记符，指令必须用机器码表示，同样，数也只能用二进制（或十六进制）表示。

第一条指令　1011 0000　（MOV AL，n）

　　0000 0111　（n=7）

第二条指令　0000 0100　（ADD AL，n）

　　0000 1010　（n=10）

第三条指令　1111 0100　（HLT）

总共是 3 条指令 5 个字节。

如前所述，程序应放在存储器中，若它们放在地址以 00H（两位 16 进制数）开始的存储单元内，则需要如图 1-8 所示的连续的 5 个存储单元。

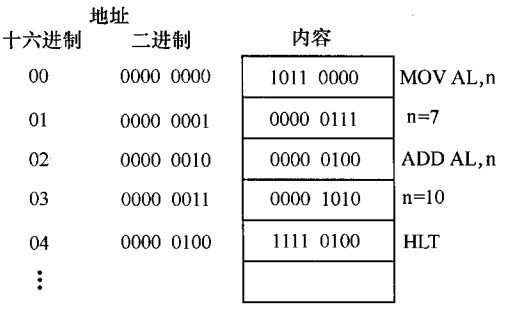


图 1-8　指令的存放

在执行时，给 PC 赋予第一条指令的地址 00H，然后就进入第一条指令的取指阶段，具体操作如下：

① PC 的内容（00H）送至地址寄存器；

② 当 PC 的内容可靠地送入地址寄存器后，PC 自动加 1，变为 01H；

③ 地址寄存器通过地址总线把地址号 00H 送至存储器。经地址译码器译码，选中 00 号单元；

④ CPU 给出读命令；