高等学校信息工程类专业系列教材

新编单片机原理与应用

A New Course in Principles and Applications of Single-Chip Computers

谭东锦 编著

面向 21 世纪高等学校信息工程类专业系列教材

新编单片机原理与应用

潘永雄　编著

张晓蓟　主审

西安电子科技大学出版社

2003

内 容 简 介

本书以增强型 MCS-51 单片机原理及应用为主线，系统地介绍了 8XC5X（包括 8XC5X2）、8XC51RX、89C6XX2、P87LPC76X 系列 MPU 芯片的内部结构、指令系统、资源及扩展方法、接口技术，以及单片机应用系统硬件结构、开发手段与设备等。在编写过程中，尽量避免过多地介绍程序设计的方法和技巧，着重介绍硬件资源及使用方法、系统构成及连接，注重典型性和代表性，以期达到举一反三的效果。在内容安排上，力求兼顾基础性、实用性、先进性。

本书可作为高等学校电子类专业“单片机原理与应用”课程的教材或教学参考书，亦可供从事单片机技术开发、应用的工程技术人员阅读。

本书配有电子教案，需要者可与西安电子科技大学出版社发行部联系，免费索取。

图书在版编目（CIP）数据

新编单片机原理与应用 / 潘永雄编著. —西安：西安电子科技大学出版社，2003.2

ISBN 7 – 5606 – 0843 – 4

Ⅰ. 新… Ⅱ. 潘… Ⅲ. 计算机网络—高等教育—自学考试—自学参考资料 Ⅳ. TP3

中国版本图书馆 CIP 数据核字（2003）第 091196 号

策　　划　马乐惠

责任编辑　李　佳

出版发行　西安电子科技大学出版社（西安市太白南路 2 号）

电　　话　(029)8227828　　邮　编　710071

http://www.xduph.com　　E-mail: xdupfxb@pub.xaonline.com

经　　销　新华书店

印　　刷　西安兰翔印刷厂

版　　次　2003 年 2 月第 1 版　　2003 年 2 月第 1 次印刷

开　　本　787 毫米×1092 毫米　1/16　印张 20.75

字　　数　490 千字

印　　数　1～4 000 册

定　　价　22.00 元

ISBN 7 – 5606 – 0843 – 4 / TP・0438

XDUP 1114A01-1

＊＊＊ 如有印装问题可调换 ＊＊＊

前　言

单片机技术作为计算机技术的一个重要分支，广泛应用于工业控制、智能化仪器仪表、家用电器，甚至电子玩具等各个领域，它具有体积小、功能多、价格低廉、使用方便、系统设计灵活等优点。因此，越来越受到工程技术人员的重视，目前国内中高等学校电子技术、电力技术、自动控制、计算机硬件等专业均开设了“单片机原理与应用”课程。

本书以单片机在电子技术中的应用为主线，以需要掌握和使用单片机技术的中高等学校有关专业学生、工程技术人员作为主要的服务对象。从实用角度出发，力争用通俗易懂的语言，由浅入深，系统、详细地介绍增强型 MCS-51 系列单片机的硬件结构、指令系统、程序设计方法、接口技术等方面的基本知识，然后结合典型应用实例介绍单片机应用系统的开发过程、手段和设备。在本书编写过程中，尽量避免过多地介绍程序设计方法和技巧，而着重介绍系统的硬件组成及连接、系统调试方法，注重典型性和代表性，以期达到举一反三的效果。内容安排上力求兼顾基础性、实用性、先进性。

本书共分 8 章，考虑到部分读者可能没有学过“计算机原理”方面的基础知识，在第 1 章中先介绍计算机系统的基本结构、工作原理等基础知识，为学习后续章节内容奠定基础；第 2 章和第 4 章全面系统地介绍增强型 MCS-51 内核单片机芯片的内部结构、数据存储器扩展方法、中断控制器、定时/计数器、串行接口电路的功能和使用方法，是本书的重点；第 3 章简要介绍 MCS-51 指令系统、单片机应用程序结构、设计规则，是本书的必学内容；为了进一步提高读者的单片机开发应用水平，拓宽芯片选择范围，第 5、6 章简要介绍了 P89C51RX、P89C6XX2、P89C66X、P87LPC76X 等系列单片机芯片新增硬件资源及使用方法，是本书的选学内容；第 7 章介绍了常用输入/输出接口电路；第 8 章扼要介绍了单片机应用系统的设计规则、开发设备及过程。

附录 A 安排了较为详细的、可操作的实验内容。

本书既可作为高等学校有关专业“单片机原理与应用”课程的教材或教学参考书，也可作为需要掌握和使用单片机技术的工程技术人员的参考资料。

为方便教学，避免重复劳动，本书配有电子教案，在 Word 2000 环境下打开其中的文档，即可将其中的程序或程序段复制到仿真开发编辑器内，直接汇编（因仿真开发环境不尽相同，部分程序段可能需要略做修改后才能汇编）、运行。

本书在编写过程中，得到了广州周立功单片机发展有限公司、南京伟福实业公司的大力支持；西北工业大学的张晓蓟老师审阅了全书，在此一并表示感谢。

由于编者水平有限，书中不当之处在所难免，恳请读者批评、指正。

编著者

2002 年 10 月

目　录

注：加*的部分为扩展内容，教学时可根据学时作适当删节。

第1章 基础知识

1.1 码 制

这一节主要介绍单片机课程常用到的 ASCII 码，以及原码、反码、补码、十进制数的二进制表示法——BCD 码等方面的基本知识。

计算机内部所有的数据均采用二进制代码表示，但通过输入设备(如键盘)输入的信息和通过输出设备(如显示器、打印机)输出的信息却是多种多样的，既有字母、数字，又有各种控制字符，甚至汉字。为了方便，人们对计算机中常用的符号进行编码，当通过输入设备向计算机输入某个字符时，计算机会自动将该字符转化为对应的二进制数再进行处理，同时也将处理结果还原为对应的字符。于是，字符所对应的二进制数就称为该字符的编码。

1.1.1 英文字符的表示方法——ASCII 码

由于计算机只能处理二进制数，因此除了数值本身需要用二进制数形式表示外，字符，包括数码(如 0，1，2，3，4，5，6，7，8，9)、字母(如 A，B，C，D，…，X，Y，Z 及 a，b，c，d，…，x，y，z)、特殊符号(如%，!，+，−，=等)也必须用二进制数表示，即在计算机中需将数码、字母、特殊符号等代码化，以便于计算机识别、存储和处理。

英文属于典型的拼音文字，由字母、数字、特殊符号等组合而成，但这些字母、数字、特殊符号的数目毕竟有限，不过百余个。我们知道 7 位二进制数可以表示 128 种状态，如果每一种状态代表特定的字母或数字，则 7 位二进制数可表示 128 个字符。

例如：可用 0110000B 表示数字 0，0110001B 表示数字 1，1000001B 表示大写字母 A 等。但这种编码方式并不惟一，如用 0110000B 表示数字 A，0110001B 表示数字 B，1000001B 表示数字 0 也未尝不可。为了便于不同计算机系统和不同操作者之间的信息交换，就有必要规范字母与 7 位二进制数之间的对应关系。目前计算机系统中普遍采用美国标准信息交换代码(American Standard Code for Information Interchange II，简称 ASCII 码)。

该标准用 7 位二进制数表示一个字符，最多可以表示 128 个字符，编码与字符之间的对应关系如附录 C 所示。

在计算机系统中，存储单元的长度通常为 8 位二进制数(即一个字节)，为了存取方便，规定一个存储单元存放一个 ASCII 码，其中低 7 位表示字母本身的编码，第 8 位(bit7)用作奇偶校验位或规定为零(通常如此)。因此，也可以认为 ASCII 码的长度为 8 位(但 bit7 为 0)。128 个字符对于某些特殊应用来说，可能不够，因此就采用 8 位的 ASCII，即扩展 ASCII 码(共有 256 个代码)。其中前 128 个(高位为 0)编码用于表示基本的 ASCII 码，基本 ASCII

码主要用于表示数字、英文字母(大、小写)、标点符号、控制字符等；后 128(高位为 1)个编码用于表示扩展的 ASCII 码，扩展 ASCII 用于表示一些特殊的符号，如希腊字母等。

1.1.2 BCD 码(二进制编码的十进制数)

十进制毕竟是人们最习惯的记数方式，在向计算机输入数据时，常用十进制数输入，但计算机只认识二进制数，因此每 1 位十进制数必须用二进制数表示。1 位十进制数包含 0～9 十个数码，必须用 4 位二进制数表示，这样就需要确定 0～9 与 4 位二进制数 0000B～1111B 之间的对应关系，其中较常用的 8421 BCD 码规定了十进制数 0～9 与 4 位二进制数编码之间的对应关系如下：

十进制数	8421 BCD 码	十进制数	8421 BCD 码
0	0000	5	0101
1	0001	6	0110
2	0010	7	0111
3	0011	8	1000
4	0100	9	1001

注：在 BCD 码中，不使用 1010(0AH)～1111(0FH)。

例如：4567.89 的 BCD 码为：0100 0101 0110 0111.1000 1001(每 1 位十进制数用相应的 4 位二进制数表示即可)。

尽管 BCD 码比较直观，但 BCD 码与二进制数之间的转换并不方便，需要转成十进制数后，才能转换为二进制数，反之亦然。

1.1.3 计算机中带符号数的表示方法

在计算机中，对于带符号数来说，一般用最高位表示数的正、负。对于正数，最高位规定为“0”，对于负数，最高位为“1”，例如：

56H 可以表示 0 1010110(对于 8 位二进制数来说，b7 位表示数的正负，b6～b0 表示数的绝对值)；－56H 可以表示 1 1010110。

0256H 可以表示 0 000 0010 0101 0110(对于 16 位二进制数来说，b15 位表示数的正负，b14～b0 表示数的绝对值)；－0256H 可以表示 1 000 0010 0101 0110。

1. 原码

对于带符号数来说，用最高位表示数的正负，其余各位表示该数的绝对值，这种表示方法称为原码表示法，如上所示。

2. 反码

带符号数也可以用反码表示，反码与原码的关系是：

正数的反码与原码相同，如 $[56H]_{反}=[56H]_{原}=0\ 1010110B$。

负数的反码等于对应正数的原码按位求反。因此，求－56H 反码的过程如下：

对应正数 56H 的原码为 0 101 0110；按位求反后为 1 0101001，即－56H 的反码为

10101001。或者说，正数的反码与原码相同，而负数的反码是对应负数原码除符号位外，绝对值部分按位取反。

3. 补码

在计算机内，带符号数并不是用原码或反码表示，而是用补码表示，引入“原码”、“反码”的目的只是为了方便理解补码概念而已。不用原码表示的原因是：用原码表示时，“0”的原码并不惟一，0 的原码可以表示为 0 0000000(即+0)，也可以表示为 1 0000000(即− 0)，这会造成混乱；再就是用原码表示时，减法并不能转化为加法运算，反码也存在类似问题。在计算机中，带符号数用补码表示后，减法可以转化为加法运算，例如：

56H−23H=56H−23H+100H　(100H 是 8 位二进制能表示的最大数，加上 100H 后，对计算结果没有影响，原因是 8 位二进制无法存放 100H 中的“1”，即 b8 位)

=56H+100H−23H

=56H+0DDH

=1 33H

=33H　(由于 8 位二进制不能存放 b8 位，结果 133H 中最高位“1”自然丢失)

显然，56H−23H 的结果与 56H+0DDH 相同，即引入补码后，减法可以用加法来完成。可见，在 8 位二进制中，23H 与 0DDH 互为补码。

补码的定义如下：

正数的补码与反码、原码相同；

负数的补码等于它的反码加 1。

因此，求−23H 补码的过程如下：

对应正数 23H 的原码为 0 0100011；按位求反后为 1 1011100，即−23H 的反码；反码加 1 后为 1 1011101，即−23H 的补码为 1 1011101(相当于无符号数的 0DDH)。

可见，用补码表示时，最高位为 0 时，表示该数为正数，数值部分就是真值；而最高位为 1 时，为负数，数值部分并不是它的真值，必须再求补后，才得到该数的绝对值，如上例中的−23H 的补码为 1 1011101, 按位取反后为 00100010，加 1 后为 00100011，即 23H。

对于 8 位二进制数来说，补码表示的范围是−128～+127；对于 16 位二进制数来说，补码表示的范围是−32768～+32767。

例如，求 16 位二进制数中−23H 的补码过程如下：

对应正数 23H 的原码为 0 000 0000 0010 0011，按位取反后为 1 111 1111 1101 1100；加 1 后为 1 111 1111 1101 1101(即 16 位二进制数−23H 的补码，相当于无符号数的 0FFDDH)。可见，对于同一个数，作为 8 位二进制数的补码和作为 16 位二进制数的补码不同，这一点要特别留意。

1.2　计算机的基本知识

为了理解计算机的硬件组成、工作原理及过程，我们先来看用算盘计算如下代数式的过程：

12×34+56÷7−8=

首先要有算盘作为计算工具，在计算机里用“运算器”(简称 ALU，即算术逻辑运算单元)作为计算工具，由它承担算术运算和逻辑运算。因为在计算机里，除了需要对数据进行加、减、乘、除四则运算外，还需要进行逻辑与、或、非、异或等逻辑运算。其次需要纸和笔记录算式、计算步骤、中间结果及最终结果。在计算机中，起到纸和笔作用的器件是存储器和寄存器(寄存器在 CPU 即中央处理器内，特点是存取速度快，但数量少，用于存放中间结果；而存储器由成千上万个存储单元组成，容量大，与寄存器相比，存取速度慢一些，常用于存放数据、运算步骤，即指令)。在计算上述算式时，先计算 12×34，并把中间结果记录下来；然后再计算 56÷7，再记录中间结果；把上述两步中间结果相加，并记录下来；再减 8。以上计算步骤由人脑控制，如果改用计算机进行，可用计算机指令写出如下的计算步骤：

```
MOV B,#34       ; 将乘数 34 传送到 CPU 内寄存器 B
MOV A,#12       ; 将被乘数 12 传送到 CPU 内寄存器 A
MUL AB          ; 计算 12 × 34，结果的高 8 位存放在寄存器 B 中，低 8 位存放在寄存器 A 中
MOV R2,A
MOV R3,B        ; 将中间结果暂时保存到寄存器 R2、R3 中
MOV B,#7        ; 将除数 7 传送到 CPU 内寄存器 B
MOV A,#56       ; 将被除数 56 传送到 CPU 内寄存器 A
DIV AB          ; 计算 56 ÷ 8，商存放在寄存器 A 中，余数存放在寄存器 B 中
ADD A,R2        ; 低 8 位相加
MOV R2,A        ; 把结果暂存到 R2 中
MOV A,R3        ; 把高 8 位传送到 A 中
ADDC A,#00      ; 低 8 位相加时可能产生进位，用 ADDC 指令将 A 与 00 相加，即可将低 8 位
                ; 相加产生的进位加到高 8 位中
MOV R3,A        ; 把结果暂存到 R3 中
MOV A,R2        ; 把低 8 位传送到 A 中
CLR C           ; 清进位标志
SUBB A,#08      ; 低 8 位减 8
MOV R2,A        ; 把结果传送到 R2 中
MOV A,R3        ; 把高 8 位传送到 A 中
SUBB A,#00      ; 12 × 34 + 56 ÷ 7 获得的中间结果减 8 时，低 8 位可能产生借位，因此需要用
                ; SUBB 指令将高 8 位与 00 相减
MOV R3,A        ; 把结果传送到 R3 中。这样 12×34+56÷7−8 的最后结果就保存在 R3、R2 中
```

将寄存器中的结果存放在存储器中，然后由计算机内的控制器完成，控制器在时钟信号的控制下，从存储器中取出计算步骤(即指令)和数据，并根据指令操作码内容发出相应的控制信号。此外，为了向计算机输入数据、指令，还需要输入设备，如键盘；同时，为了输出处理结果或显示机器的状态，还需要输出设备，如显示器。因此，计算机系统的基本结构如图 1-1 所示。

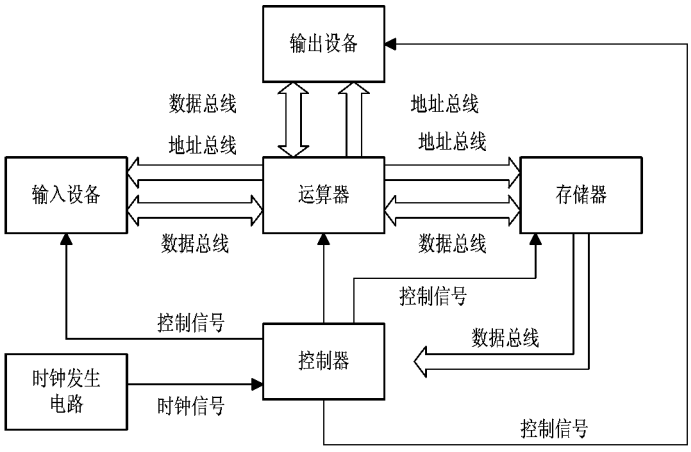

图 1-1　计算机基本结构

在计算机中，往往把运算器、控制器做在一个芯片上，称为中央处理器(Central Processor Unit，简称 CPU)，有时也称为微处理器(Micro Processor Unit，即 MPU)。为了进一步减小电路板面积，提高系统可靠性，将输入、输出接口电路、时钟电路以及存储器、运算器、控制器等部件集成在一个 CPU 芯片内，就成为单片机(也称为微控制器，即 Micro Controller Unit，简称 MCU)，其含义是一个芯片就具备了一个完整计算机系统所必须的基本部件。为了适应不同的需求，将不同功能的外围电路如定时器、中断控制器、A/D 及 D/A 转换器、串行通信接口电路等集成在 CPU 管芯内，形成系列化产品，就构成了所谓“嵌入式”单片机控制芯片。

1. 总线概念

在计算机系统中，常包含以下几种总线：

(1) 地址总线(Address Bus，简称 AB)为单向，用于传送地址信息，如图 1-1 中运算器与存储器之间的地址线，地址线的数目决定了可以寻址的存储空间。一根地址线有两种状态，即可以区分两个不同的存储单元，或者说可以寻址两个存储单元；两根地址线有四种状态，可以寻址四个存储单元；……；8 位微处理器通常有 16 根地址线，可以寻址 2^{16} 种状态，即 64 K 个存储单元。一般存储单元的大小为一个字节，因此 8 位微处理器的寻址范围为 64 KB。

(2) 数据总线(Data Bus，简称 DB)一般为双向，用于 CPU 与存储器、CPU 与外设，或外设与外设之间的传送数据(包括实际意义的数据和指令码)信息。在计算机中，为了提高处理速度，总是一次处理由多位二进制数组成的信息，即在运算器中，数据线的数目应与待处理的数据位数相同。因此，运算器数据线的数目往往不止一条，一般为 4 条、8 条或 16 条。运算器内数据线的多少称为微处理器的“字长”。字长是衡量微处理器功能、运算速度以及精度的重要指标之一，也是划分微处理器档次的重要依据。根据字长，可以将微处理器分为 1 位机、4 位机、8 位机、16 位机、32 位机、64 位机等。1 位机的运算器只有一根数据线，每次只能处理一位二进制数，工业上常用它取代继电器，用于控制线路的通和断、设备的开和关；4 位机有四根数据线，常用于家用电器，如电视机、空调机、洗衣机等的控制电路中。8 位机功能强大，不仅可用于工业控制、家用电器，也可以作为通用微机系统的

中央处理器。

(3) 控制总线(Control Bus，简称 CB)，是计算机系统中所有控制信号线的总称，在控制总线中传送的信息是控制信息。

2. 时钟周期、机器周期及指令周期

(1) 时钟周期：计算机在时钟信号的作用下以节拍方式工作，因此必须有一个时钟生发器电路。输入微处理器的时钟信号的周期称为时钟周期。

(2) 机器周期：机器完成一个动作所需的时间称为机器周期，一般由一个或一个以上的时钟周期组成，例如在 MCS-51 系列单片机中，一个机器周期由 12 个时钟周期组成。

(3) 指令周期：执行一条指令(如“MOV A,#34H”，该指令的含义是将立即数 34H 传送到微处理器内的累加器 A 中)所需时间称为指令周期，它由一个到数个机器周期组成。指令周期的长短取决于指令的类型，即指令将要进行的操作步骤及复杂程度：简单指令，如 INC A(累加器 A 的内容加 1)可能只需要一个机器周期，而复杂指令，如 MUL AB(累加器 A 乘以寄存器 B，并将结果放在寄存器 B 和累加器 A 中)将需要多个机器周期。

1.2.1 计算机的工作过程及内部结构

1. CPU 的内部结构

图 1-1 中的运算器和控制器等部件往往做在同一芯片内，称为中央处理器(CPU)。8 位通用微处理器内部基本结构可用图 1-2 描述，它由算术逻辑运算单元(Arithmetic Logic Unit, 简称 ALU)、累加器 A(8 位)、寄存器 B(8 位)、程序状态字寄存器 PSW(8 位)、程序计数器 PC(有时也称为指令指针，即 IP，16 位)、地址寄存器 AR(16 位)、数据寄存器 DR(8 位)、指令寄存器 IR(8 位)、指令译码器 ID、控制器等部分组成。

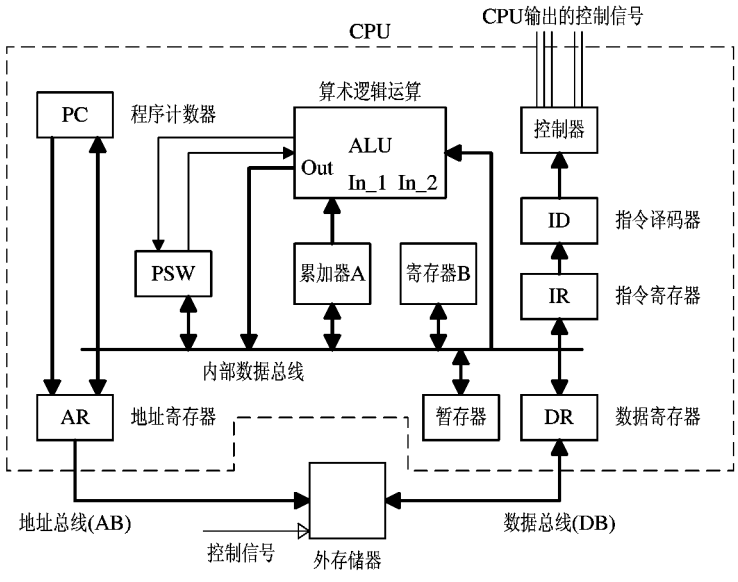

图 1-2　CPU 的内部结构简图

(1) 程序计数器 PC 是 CPU 内部的寄存器，用于记录将要执行的指令代码所在存储单元的地址编码。一般来说，PC 长度与 CPU 地址线引脚数目一致，例如 8 位微机的 CPU 一般具有 16 根地址线(A15～A0)，PC 的长度也是 16 位。复位后，PC 具有确定值，例如在 MCS-51 系列单片机中，复位后，PC=0000H，即复位后将从程序存储器的 0000H 单元读取第一条指令码。由于复位后，PC 的值就是第一条指令代码存放的单元，因此设计程序时，必须了解复位后 PC 的值是什么，以便确定第一条指令码从存储器哪一存储单元开始存放。PC 具有自动加 1 的功能，即从存储器中读出一个字节的指令码后，PC 会自动加 1(即指向下一存储单元)。

(2) 地址寄存器 AR(Address Register，16 位)用于存放将要寻址的外部存储器单元的地址信息，指令码所在存储单元的地址编码，由程序计数器 PC 产生；而指令中操作数所在存储单元的地址码，由指令的操作数给定。地址寄存器 AR 通过地址总线 AB 与外部存储器相连。

(3) 指令寄存器 IR(Instruction Register)用于存放取指阶段读出的指令代码的第一字节，即操作码，使指令译码器 ID 的输入保持不变，存放在 IR 中的指令码经指令译码器 ID 译码后，输入控制器，产生相应的控制信号，使 CPU 完成指令规定的动作。

(4) 数据寄存器 DR 用于存放写入外部存储器或 I/O 端口的数据信息。由此可见，数据寄存器 DR 对输出数据具有锁存功能，数据寄存器与外部数据总线 DB 直接相连。

(5) 算术逻辑运算单元 ALU 主要用于算术(加、减、乘、除)、逻辑(与、或、非以及异或)运算。由于 ALU 内部没有寄存器，参加运算的操作数必须放在累加器 A 中，同时也用于存放运算结果。例如，执行：

```
ADD A,B   ;A←A+B
```

指令时，累加器 A 中的内容通过输入口 In_1 输入 ALU，寄存器 B 通过内部数据总线经输入口 In_2 输入 ALU，A+B 的结果通过 ALU 的输出口 Out 及内部数据总线送回累加器 A。

(6) 程序状态字寄存器 PSW 用于记录运算过程中的状态，如是否溢出、进位等。

例如，累加器 A 的内容 83H，执行如下指令后，将产生进位，因为和的结果为 1 0DH，而累加器 A 只有 8 位，只能存放低 8 位，即 0DH，无法存放结果中的最高位 b8。为此，在 CPU 内设置一个进位标志 C，当执行加法运算出现进位时，进位标志 C 为 1。

```
ADD A,#8AH              ；累加器 A 与立即数 8AH 相加，并把结果存在 A 中
```

程序状态字寄存器中各标志位含义与 CPU 类型有关，在 2.3 节中将详细介绍 MCS-51 CPU 内各标志位的含义。

2. 存储器

存储器是计算机系统中必不可少的存储设备，主要用于存放程序(指令)和数据。尽管寄存器和存储器均用于存储信息，但 CPU 内的寄存器数量少，存取速度快，它主要用于临时存放参加运算的操作数和中间结果；而存储器一般在 CPU 外(但单片机 CPU 除外，其内部一般均含有一定容量的存储器)单独封装。在存储器芯片内，存储单元数目多，从几千字节到数百兆字节，能存放大量的信息，但存取速度比 CPU 内部的寄存器要慢得多。目前，存储器的存取速度已成为制约计算机运行速度的关键因素之一。

存储器的种类很多，根据存储器能否随机读写，将存储器分为两大类：只读存储器(Read Only Memory，简称 ROM)和随机读写存储器(Random Access Memory，简称 RAM)。根据存储器存储单元结构和信息保存方式的不同，又可以将随机读写存储器分为静态 RAM(采用双极型晶体管结构，存取速度快，无需刷新，但构成一个存储单元所需的晶体管数目较多，集成度低，价格略高)和动态 RAM(采用 CMOS 工艺，依靠 MOS 管栅极与衬底之间的寄生电容保存信息，一般均为单管结构，集成度高，但寄生电容容量小，漏电大，信息保存时间短，仅为毫秒级，需要刷新电路，致使动态 RAM 存储器系统电路复杂化，不适用于仅需要少量存储容量的单片机系统)。

只读存储器中“只读”的含义是信息写入后，只能读出，不能随机修改，适合存放系统监控程序。

在单片机应用系统中，所需的存储器容量不大，外围电路应尽可能简单，因此几乎不使用动态 RAM，常使用 PROM(可编程的只读存储器)、EPROM(紫外光可擦写的只读存储器)、OTP ROM(一次性编程的只读存储器，内部结构、工作原理与 EPROM 相似，是一种没有擦除窗口的 EPROM)、EEPROM(也称为 E^2PROM，是一种电可擦写的只读存储器，其结构与 EPROM 类似，但绝缘栅很薄，高速电子可穿越绝缘层，中和浮栅上的正电荷，起到擦除目的，也就是说可通过高电压擦除)、Flash ROM(电可擦写只读存储器，写入速度比 EEPROM 快，因此也称为闪烁存储器)等只读存储器作为程序存储器，使用 SRAM(随机读写的静态存储器)、E^2PROM 作为数据存储器。尽管这些存储器工作原理不同，但内部结构基本相同。

1) 内部结构

EPROM、EEPROM、Flash ROM、SRAM 等存储器内部结构可以用图 1-3 描述，由地址译码器、存储单元、读写控制电路等部分组成。

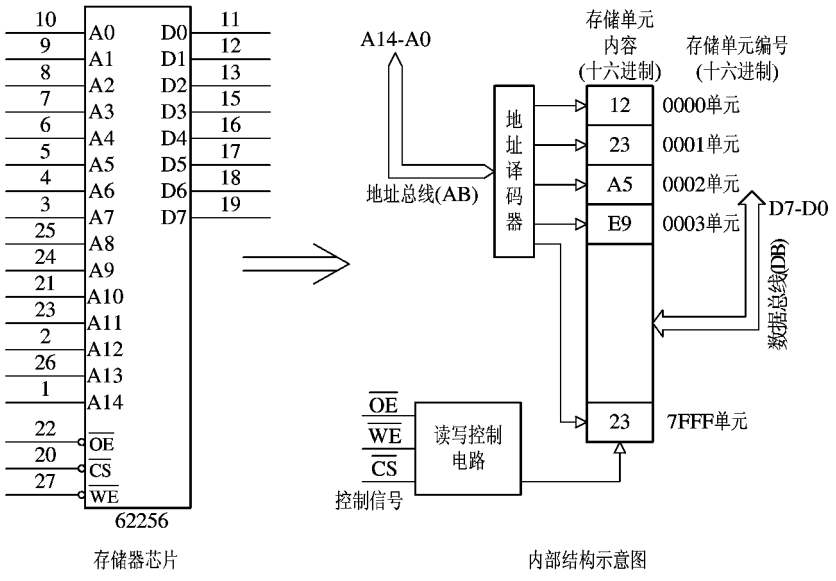

图 1-3　存储器芯片及内部结构

寄存器或存储器中的一个存储单元等效于一组触发器，每个触发器有两个稳定状态，可以记录一位二进制数。每一存储单元包含的触发器的个数称为存储单元的字长，对于并行存取的存储器芯片，存储单元内包含的触发器的个数与存储器芯片数据线的条数相同。

例如由 8 个触发器并排在一起构成的存储单元的字长为 8 位，它可以存放一个 8 位二进制数(一个字节)。在计算机中，为了提高处理速度，一次操作(如数据传送或运算)要同时处理多位二进制数，因此在并行存取的存储器芯片中，一个存储单元的容量通常为 8 位。

存储器芯片内存储单元数目与存储器芯片地址线条数有关。例如，图 1-3 中的 62256 随机读写静态存储器芯片含有 15 根地址线(A14～A0)，可以寻址 2^{15} 共 32K 个存储单元。为了便于存取，给每个存储单元编号(通常用存储器地址线状态编码作为存储单元的编号)，图 1-3 中的 32K 个存储单元的地址编码为 0000H～7FFFH。由于该芯片每个存储单元可以容纳一个 8 位二进制数，即该芯片存储容量为 32 KB。

但存储单元长度也可以大于或小于 8 位，例如 PIC16C56 单片机内的程序存储器容量为 1K×12 位，即共有 1024 个存储单元，每个存储单元可以存放 12 位二进制数，即存储单元的字长为 12 位。

存储单元地址编码与存储单元中的内容是两个不同的概念，存储单元地址编码的长度由存储器芯片所包含的存储单元的个数决定。例如，6264 存储器芯片含有 8K 个(13 根地址线)存储单元，地址编码为 0 0000 0000 0000B～1 1111 1111 1111B(用十六进制表示时，地址编码为 0000H～1FFFH)，每个存储单元长度为 8 位。因此，每个存储单元的内容可以是 00H～FFH 之间的二进制数。又如，图 1-3 中的 62256 存储器含有 32K 个(15 根地址线)存储单元，地址编码为 000 0000 0000 0000B～111 1111 1111 1111B(用十六进制表示时，地址编码为 0000H～7FFFH),每个存储单元的长度也是为 8 位，图中 0000H 单元内容为 12H，0001H 单元内容为 23H，而 0002H 单元内容为 0A5H。PIC16C56 单片机程序存储器容量为 1K×12bit，因此存储单元地址编码为 00 0000 0000B～11 1111 1111B(用十六进制表示时，地址编码为 000H～3FFH),每个存储单元长度为 12 位。因此，每个存储单元的内容可以是 000H～FFFH 之间的二进制数。

2) 存储器工作状态

存储器芯片工作状态由存储器控制信号电平状态决定，如表 1-1 所示。

表 1-1　存储器工作状态

工作模式	控 制 信 号			输　出
	片选信号 $\overline{CS}$	输出允许 $\overline{OE}$	写允许信号 $\overline{WE}$	
读	L	L	H	数据输出
输出禁止	L	H	H	高阻态
待用(功率下降)	H	×	×	高阻态
写入	L	H	L	数据输入

3) 存储器读操作

下面以 CPU 读取存储器中地址编号为 0000H 的存储单元的内容为例，说明 CPU 读存储器中某一存储单元信息的操作过程，如图 1-4 所示。

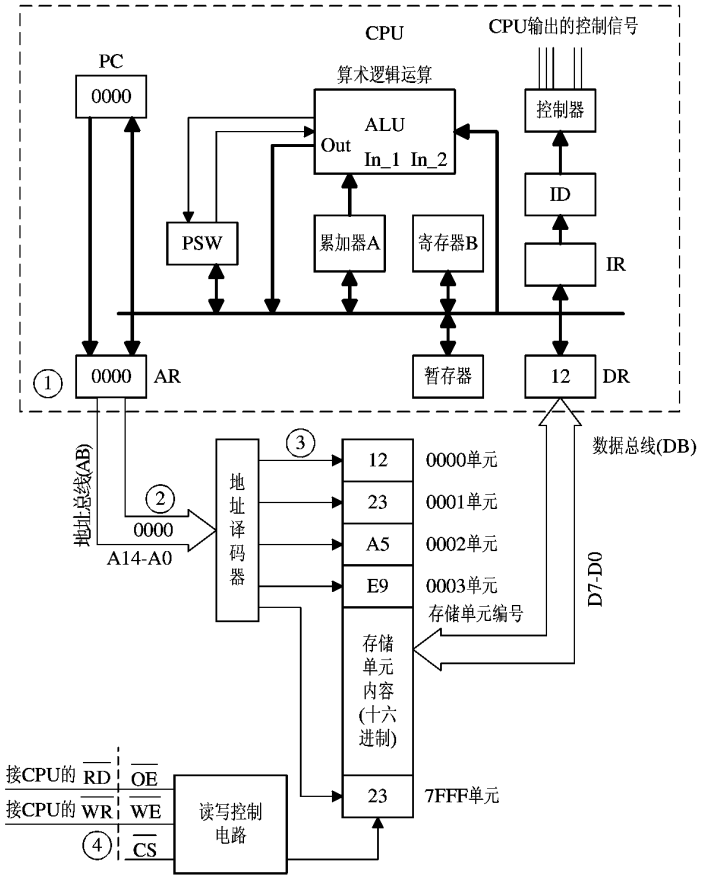

图 1-4　CPU 读取存储器操作过程示意图

(1) CPU 地址寄存器 AR 给出将要读取的存储单元地址信息，即 0000H。

(2) 存储单元地址信息通过地址总线 A15～A0 输入到存储器芯片地址线上(CPU 地址总线与存储器地址总线相连)。

(3) 存储器芯片内的地址译码器对存储器地址信号 A14～A0 进行译码，并选中 0000H 单元。

(4) CPU 给出读控制信号 $\overline{RD}$ (接存储器的 $\overline{OE}$ 端)，将选中的 0000H 存储单元内容输出到数据总线 D7～D0(存储器数据总线与 CPU 数据总线相连)，结果 0000H 单元的内容 12H 就通过存储器数据总线输入到 CPU 内部的数据寄存器 DR 中，然后送到 CPU 内部某一特定寄存器或暂存器内，这样便完成了存储器的读操作过程。

对于存储器来说，读操作后，被读出的存储单元信息将保持不变。

4) 存储器写操作

把某一数据，如 55H 写入存储器内某一存储单元，如 0003H 单元的操作过程如下：