

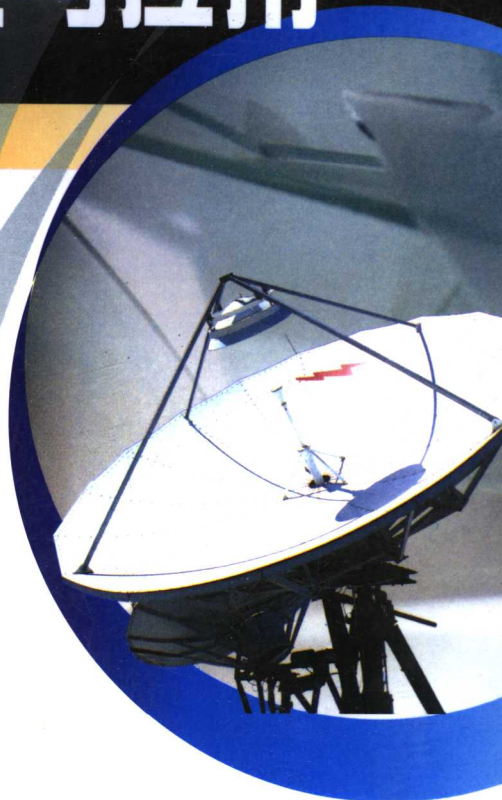
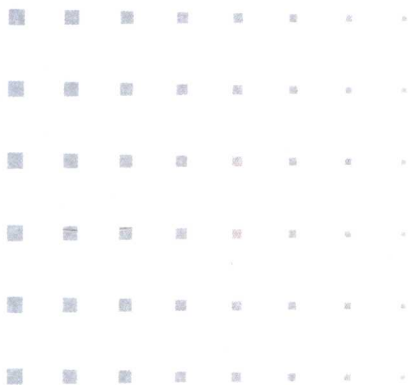


高等学校信息工程类专业规划教材

新编单片机原理与应用

(第二版)

潘永雄 编著



西安电子科技大学出版社
<http://www.xduph.com>

面向 21 世纪高等学校信息工程类专业规划教材

新编单片机原理与应用

(第 二 版)

潘永雄 编著

西安电子科技大学出版社

2007

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 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的硬件结构、开发手段与设备等。在编写过程中,本书尽量避免过多地介绍程序的设计方法和技巧,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,本书也力求兼顾基础性、实用性和先进性。

本书既是“单片机原理与应用”的入门教材,也可作为中、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参考。

★ 本书配有电子教案,需要者可与出版社联系,免费提供。

图书在版编目(CIP)数据

新编单片机原理与应用 / 潘永雄编著. —2 版. —西安: 西安电子科技大学出版社, 2007. 2

面向 21 世纪高等学校信息工程类专业规划教材

ISBN 978 - 7 - 5606 - 0843 - 3

I. 新… II. 潘… 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(2007)第 013271 号

策 划 马乐惠

责任编辑 王 婷 马乐惠

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)88242885 88201467 邮 编 710071

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

经 销 新华书店

印刷单位 西安文化彩印厂

版 次 2003 年 2 月第 1 版 2007 年 2 月第 2 版 2007 年 2 月第 5 次印刷

开 本 787 毫米×1092 毫米 1/16 印张 20

字 数 475 千字

印 数 18 001~22 000 册

定 价 24.00 元

ISBN 978 - 7 - 5606 - 0843 - 3/TP · 0438

XDUP 1114A12 - 5

*** 如有印装问题可调换 ***

本社图书封面为激光防伪覆膜,谨防盗版。

前 言

《新编单片机原理与应用》第一版出版后,在三年多的时间里,我们依据单片机技术的新成果、开发工具与开发环境的新发展,以及我们近几年来在单片机教学、开发应用中获得的经验与体会,在保留第一版教材架构的基础上,对其中的内容作了全面修改与调整,并逐字逐句纠正了其中模糊或不当的表述。第二版书中绝大部分实例、例题及习题均取材于作者近三年来的十多个单片机开发应用项目,是作者多年来单片机开发应用实践的经验总结,具有一定的实用性。

第二版修改与调整的内容如下:

(1) 重写了第一版中第1~5、第7、8章的内容。根据单片机技术现状,删除了第一版中一些过时的内容,适当介绍了一些新的单片机芯片的功能及特征;全面纠正了第一版中的错漏;并根据使用第一版教材的教师及学生的意见和建议,按教材体例调整了部分教学内容的顺序,重写并充实了全部的例题和习题。

(2) 删除了第一版第6章的内容。

(3) 鉴于本教材配套实验指导书——《新编单片机原理与应用实验》已于2005年出版,第二版删除了原附录中的实验、I²C接口器件原理与MCS-51连接等内容。

(4) 基于内置一定容量的扩展RAM单片机芯片已成为主流,删除了第一版第7章中与“8155/8156 并行接口芯片”有关的内容,增加了LCD显示器件原理与接口方式等内容。

第二版教材中加*的部分为扩展内容,教学时可根据学时作适当删减。

感谢使用第一版教材的教师提出了许多宝贵的意见和建议。广东工业大学物理与光电工程学院周展怀等教师参与了部分内容的编写工作,我院研究生胡敏强利用假期校对了全书内容,在此一并表示感谢。

尽管我们力求做到尽善尽美,但由于水平有限,书中不当之处在所难免,恳请读者批评指正。

编 者

2006年11月于广州

第一版前言

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相关专业学生、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,也可以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。

本书在写作过程中,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蓓老师审阅了全书,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

第 1 章 单片机基础知识.....	1
1.1 计算机的基本知识.....	1
1.1.1 计算机系统的工作过程及其内部结构.....	4
1.1.2 指令及其指令系统.....	8
1.2 计算机的寻址方式.....	15
1.3 单片机及其发展概况.....	18
1.3.1 单片机的特点.....	19
1.3.2 单片机技术现状及发展趋势.....	20
1.3.3 增强型 MCS-51 单片机芯片简介.....	23
习题 1.....	26
第 2 章 增强型 MCS-51 单片机.....	27
2.1 增强型 MCS-51 单片机性能综述.....	27
2.2 内部结构和引脚功能.....	29
2.2.1 内部结构.....	29
2.2.2 引脚功能.....	31
2.3 输入/输出(I/O)口.....	36
2.3.1 P1 口内部结构及使用情况.....	36
2.3.2 P0 口内部结构及使用情况.....	37
2.3.3 P2 口内部结构及使用情况.....	38
2.3.4 P3 口内部结构及使用情况.....	39
2.3.5 I/O 口负载能力.....	39
2.3.6 读锁存器和读引脚指令.....	40
2.4 存储器系统及访问方式.....	41
2.4.1 片内数据存储器.....	42
2.4.2 程序存储器.....	50
2.4.3 外部数据存储器.....	51
2.5 MCS-51 外部存储器的连接.....	52
2.5.1 CPU 地址总线与存储器地址总线的连接.....	53
2.5.2 MCS-51 控制系统中程序存储器的连接.....	56
2.5.3 数据存储器的连接.....	57
2.6 操作时序.....	61
*2.6.1 对外部程序存储器的读操作时序.....	61
2.6.2 外部数据存储器读/写时序.....	62
2.6.3 “6 时钟/机器周期”模式下的时序.....	65

2.7 复位及复位电路	65
2.7.1 CPU 内部复位电路	65
2.7.2 外部复位电路	66
2.8 节电运行状态和掉电运行状态	70
习题 2	72
第 3 章 MCS-51 指令系统与汇编语言程序设计基础	74
3.1 MCS-51 指令系统	74
3.1.1 数据传送指令	75
3.1.2 算术运算指令	82
3.1.3 逻辑运算指令	91
3.1.4 位操作指令	93
3.1.5 控制及转移指令	95
3.2 汇编语言程序设计基础	101
3.2.1 汇编语言程序结构	101
3.2.2 多任务程序结构及实现	112
3.2.3 汇编语言程序的编辑与执行方式	117
*3.2.4 对汇编语言程序的基本要求	117
习题 3	120
第 4 章 中断控制、定时/计数器与串行口	123
4.1 CPU 与外设通信方式概述	123
4.1.1 查询方式	123
4.1.2 中断传输方式	123
4.2 增强型 MCS-51 中断控制系统	124
4.2.1 中断源及中断标志	125
4.2.2 中断控制	126
4.2.3 中断响应过程及中断服务程序入口地址	129
4.2.4 中断初始化及中断服务程序结构	131
4.3 增强型 MCS-51 定时/计数器	133
4.3.1 定时/计数功能概述	133
4.3.2 定时/计数器 T0、T1 的结构及控制	133
4.3.3 定时/计数器 T2 的结构及控制	139
4.3.4 定时/计数器的初始化及应用	145
4.4 串行通信系统	151
4.4.1 串行通信的概念	151
4.4.2 增强型 MCS-51 串行通信接口控制及其初始化	154
4.4.3 串行口的工作方式及其应用	158
4.4.4 帧错误检测及其应用	166
4.4.5 多机通信及地址自动识别技术	169
4.4.6 RS-232C 串行接口标准及应用	172

习题 4	177
第 5 章 MCS-51 内核衍生型单片机芯片及应用	178
5.1 89C51RX 系列单片机概述	178
5.2 P89C51RX 系列芯片引脚功能	181
5.3 P89C51RX 系列芯片片内存储器结构	182
5.3.1 片内程序存储器	186
5.3.2 片内数据存储器	186
5.4 可编程计数器阵列 PCA 及其应用	187
5.4.1 PCA 内部结构及其控制电路	187
5.4.2 PCA 模块初始化步骤	191
5.4.3 PCA 模块工作模式	191
5.5 89C51RX 系列中断控制系统	197
5.6 硬件看门狗	198
5.7 P89C6XX2 系列简介	200
*5.8 P89C66X 系列简介	200
5.8.1 封装形式及引脚功能	201
5.8.2 PCA 模块	201
5.8.3 中断系统	201
5.9 SST 公司 SST89E(V)RD 及 SST89C5XRD2 系列芯片	203
5.9.1 SST89E(V)系列芯片概述	203
5.9.2 SST89E(V)系列芯片的程序存储器结构及映像	206
5.9.3 Flash ROM 状态寄存器与芯片加密设置位的关系	210
5.9.4 Flash ROM IAP 编程	211
5.9.5 SPI 串行总线	214
5.9.6 硬件看门狗	217
5.9.7 SST 中断控制系统	219
5.10 MCS-51 兼容芯片的差异及其仿真	221
习题 5	223
第 6 章 数字信号输入/输出接口电路	224
6.1 开关信号的输入/输出方式	224
6.2 I/O 资源及扩展	225
6.2.1 利用锁存器、触发器扩展 I/O 口	226
6.2.2 利用“串入并出”及“并入串出”芯片扩展 I/O 口	229
6.2.3 利用 8255 可编程 I/O 芯片扩展 MCS-51 并行 I/O 口	230
6.2.4 利用 CPU 扩展 I/O 口	238
6.3 简单显示驱动电路	239
6.3.1 发光二极管	239
6.3.2 LED 发光二极管驱动电路	240
6.3.3 LED 发光二极管显示状态及同步	241

6.4 LED 数码管及其显示驱动电路	243
6.4.1 LED 数码管	244
6.4.2 LED 数码显示器驱动电路	244
6.4.3 点阵式 LED 显示器及其驱动电路	262
6.5 LCD 显示器及其驱动电路	263
6.5.1 LCD 显示器的结构	263
6.5.2 LCD 显示器驱动电路	264
6.6 键盘电路	275
6.6.1 按键结构及其电压波形	275
6.6.2 键盘电路形式	276
6.6.3 键盘按键编码	278
6.6.4 键盘监控方式	279
6.7 并行接口及应用实例	286
6.7.1 MCS-51 与并行输入/输出设备之间的连接	286
6.7.2 MCS-51 与并行打印机之间的连接	287
6.8 光电耦合器件接口电路	289
6.9 单片机与继电器接口电路	291
习题 6	292
第 7 章 单片机应用系统开发	294
7.1 单片机应用系统开发过程概述	294
7.2 总体设计	295
7.3 硬件设计	296
7.3.1 硬件电路设计及元器件的选择	296
7.3.2 印制电路板设计	300
7.4 软件设计	301
7.4.1 资源分配	301
7.4.2 程序语言及程序结构选择	303
7.4.3 软件可靠性设计	303
7.5 单片机开发工具	306
7.5.1 仿真器	306
7.5.2 其他工具	309
习题 7	310
附录 ASCII 码表	311
参考文献	312

第1章 单片机基础知识

1.1 计算机的基本知识

为了理解计算机系统构成、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：

$$12 \times 34 + 56 \div 7 - 8 =$$

首先要有算盘作为计算工具，在计算机里用“运算器”(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。在计算机中，除了加、减、乘、除四则运算外，还需要“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(寄存器在中央处理器内，存取速度快，但数量少，用于存放中间结果；而存储器一般位于中央处理器外，由成千上万个存储单元组成，与寄存器相比其存取速度慢一些，但容量大，常用于存放数据、计算步骤，即指令)。在计算上述代数式时，先计算 12×34 ，并把中间结果记录下来；然后再计算 $56 \div 7$ ，再记录中间结果；接着将上述两步计算的中间结果相加，并记录下来；最后再减 8。

以上计算步骤由人脑控制，如果改用计算机进行控制，可用计算机汇编语言指令写出如下程序：

MOV A, #12	； 将被乘数 12 传送到 CPU 内寄存器 A
MOV B, #34	； 将乘数 34 传送到 CPU 内寄存器 B
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 A, #56	； 将被除数 56 传送到 CPU 内寄存器 A
MOV B, #7	； 将除数 7 传送到 CPU 内寄存器 B
DIV AB	； 计算 $56 \div 7$ ，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 ; $12 \times 34 + 56 \div 7$ 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 $12 \times 34 + 56 \div 7 - 8$ 的最终结果就保存在 R3、
 ; R2 中

将上述程序存放在存储器中, 然后由计算机内的控制器完成计算。控制器在时钟信号的控制下, 从存储器中取出计算步骤(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, 需要输入设备, 如键盘等; 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, 如各类显示器、指示灯等。计算机系统的基本结构如图 1-1 所示。

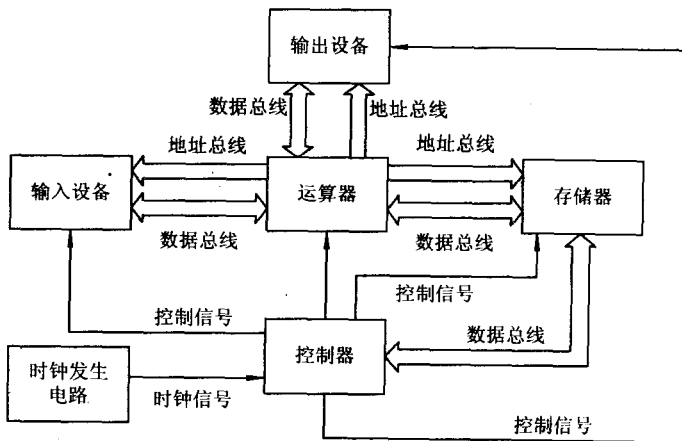


图 1-1 计算机系统的基本结构

在计算机中, 往往把运算器、控制器做在同一芯片上, 称为中央处理器(Central Processor Unit, CPU), 亦称为微处理器(Micro Processor Unit, MPU)。为进一步减小电路板面积, 提高系统可靠性、降低成本, 可将输入/输出接口电路、时钟电路以及一定容量的存储器、运算器、控制器等部件集成到同一芯片内, 从而形成单片机(也称为微控制单元, 即 MicroController Unit, 简称 MCU), 其含义是一个芯片具备了一部完整计算机系统所必需的基本部件。为了满足不同的应用需求, 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(如 UART、SPI、I²C 或 CAN 等)通信接口电路, 甚至 LCD 显示驱动电路等集成在一个管芯内, 形成系列化产品, 就构成了所谓的“嵌入式”单片机控制器(Embedded Microcontroller)。

1. 总线概念

我们知道, 电路系统总是由元器件通过电线连接而成的。在模拟电路中, 器件、部件之间的连线不多, 关系也不复杂, 一般按“串联”方式连接。但在以微处理器为核心的计算机系统电路中, 器件、部件均要与微处理器相连, 如果仍采用模拟电路的“串联”方式

连接(即在微处理器与各器件间单独连线),则所需的连线数量将很多。为此,在计算机电路中普遍采用总线连接方式:每一器件的数据线并接在一起,构成数据总线;地址线并接在一起,构成地址总线;最后与CPU的数据、地址总线相连,形成“并联”关系。为避免混乱,任何时候只允许一个设备与CPU通信,因此需要用控制线进行控制和选择。系统(包括器件)中所有的控制线被称为控制总线。

在计算机系统中,常包含以下几种总线:

(1) 地址总线(Address Bus, AB)。地址总线一般为单向,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单元。一根地址线有两种状态,即可以寻址两个不同的存储单元;两根地址线有四种状态,即可以寻址四个存储单元;依此类推。8位微处理器通常有16根地址线,可以寻址 2^{16} 种状态,即64K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通常为64KB。

(2) 数据总线(Data Bus, DB)。数据总线一般为双向,用于CPU与存储器、CPU与外设或外设与外设之间数据(包括实际意义的数据及指令码)信息的传送。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数据的位数相同。因此,运算器数据线的数目往往不止1根,一般为4根、8根、16根或32根。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1根数据线,每次只能处理1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4根数据线,常用于家用电器,如电视机、空调机、洗衣机、电话机等控制电路中;8位机功能强大,不仅可用于工业控制、家用电器,也可作为通用微机系统的中央处理器。

(3) 控制总线(Control Bus, CB)。控制总线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发生电路。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。机器完成一个基本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,在标准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)所需的时间称为指令周期,它由一个到数个机器周期组成。在采用复杂指令系统的微处理器中,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及复杂程度。简单指令,如“INC A”(累加器A内容加1)一般只需一个机器周期,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)将需要数个机器周期。

1.1.1 计算机系统的工作过程及其内部结构

1. CPU 的内部结构

中央处理器(CPU)的核心部件一般由运算器和控制器构成。8 位通用微处理器内部基本结构如图 1-2 所示, 它由算术逻辑运算单元(Arithmetic Logic Unit, 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、控制器等部件组成。

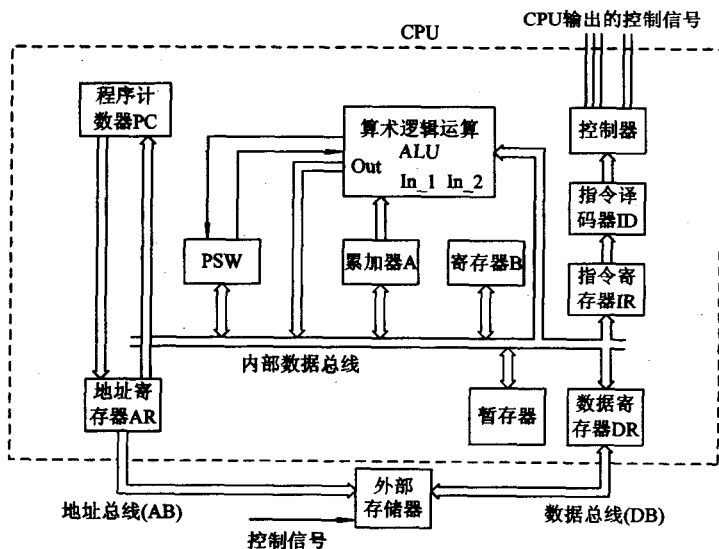


图 1-2 8 位通用微处理器的内部结构简图

(1) 程序计数器 PC(Program Counter)是 CPU 内部的寄存器, 用于记录将要执行的指令码所在存储单元的地址编码。一般来说, PC 的长度与 CPU 地址线的数目一致, 例如 8 位微机的 CPU 一般有 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码存放的单元地址, 因此在程序设计时, 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的哪一个存储单元开始存放。PC 具有自动加 1 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(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) 数据寄存器 DR(Data Register)用于存放写入外部存储器或 I/O 端口的数据信息。数据寄存器 DR 对输出数据具有锁存功能, 它通过数据总线 DB 与外部存储器相连。

(4) 指令寄存器 IR(Instruction Register)用于存放取指阶段读出的指令码的第一字节(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。

(5) 算术逻辑运算单元 ALU 主要用于算术(加、减、乘、除)运算和逻辑(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(运算结果也存放在累加器 A 中),例如执行指令:

ADD A, B ; $A \leftarrow A + B$

该指令执行时,累加器 A 中的内容通过输入口 In_1 输入 ALU,寄存器 B 中的内容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 经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。假设累加器 A 的内容为 83H,则执行如下指令后将产生进位。

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

其进位产生的原因是: $83H+8AH$ 的结果为 $10DH$,而累加器 A 的长度只有 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。

程序状态字寄存器中各标志位的含义与 CPU 类型有关,在 2.3 节将详细介绍 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,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 例外,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,存储单元数量多,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,可将存储器分为只读存储器(Read Only Memory, ROM)和随机读写存储器(Random Access Memory, RAM)。根据存储器存储单元的结构和信息保存方式的不同,又可将随机读写存储器分为静态 RAM 和动态 RAM。

静态 RAM 由多个双极型晶体管或 CMOS 管构成,其存取速度快,无需刷新。但组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。

动态 RAM 依靠 MOS 管栅极与衬底之间的寄生电容保存信息,其多为单管结构,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)且需要刷新电路,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。只读存储器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,目前已被 OTPROM、Flash ROM 取代)、OTPRAM(一次性编程的只读存储器,其内部结构和工作原理与 EPROM 类似,是一种没有擦除窗口的 EPROM)、EEPROM(也称为 E^2 PROM,是一种电可擦写的只读存储器,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,达到擦除目的。也就是说,可通过高电压进行擦除)、Flash ROM(电可擦写只读存储器,其写入速度比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(静态存储器)作为随机读写RAM,使用EEPROM或FRAM(铁电存储器,其读写速度快,操作方式与SRAM相似)作为非易失的数据存储器。尽管这些存储器工作原理不同,但其内部结构基本相同。

1) 存储器的内部结构

EPROM、EEPROM、Flash ROM、SRAM、FRAM等存储器芯片由地址译码器、存储单元和读/写控制电路等部件组成,存储器芯片及内部结构示意图如图1-3所示。

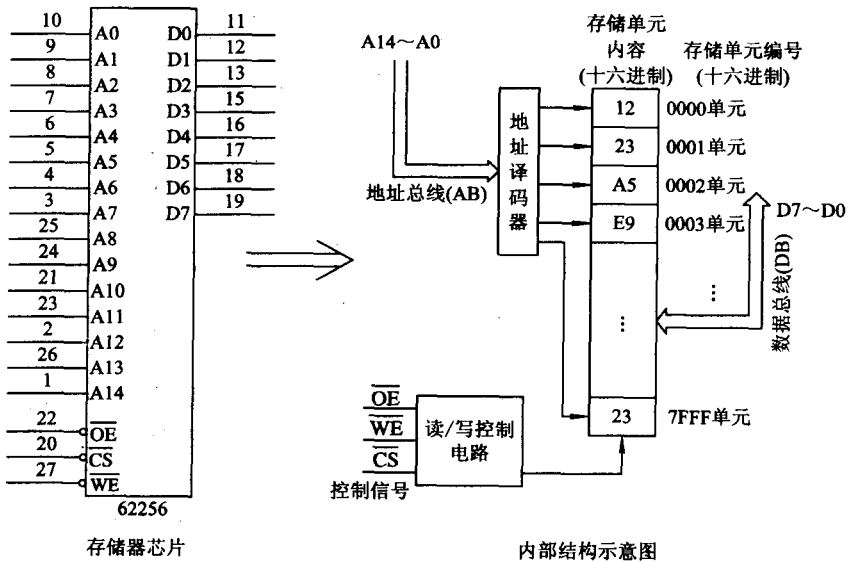


图 1-3 存储器芯片及内部结构示意图

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)往往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的数目与存储器芯片地址线的条数有关。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 32 K 个存储单元。为了便于存取,需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),则图 1-3 中的 32 K 个存储单元的地址编码范围为 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,则该芯片存储容量为 32KB。

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 $1\text{K} \times 12\text{ bit}$,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 8 K 个(13 根地址线)存储单元, 地址编码范围为 0 0000 0000 0000B~1 1111 1111 1111B(用十六进制表示时, 地址编码范围为 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 32 K 个(15 根地址线)存储单元, 地址编码范围为 000 0000 0000 0000B~111 1111 1111 1111B(用十六进制表示时, 地址编码范围为 0000H~7FFFH), 每个存储单元的长度也为 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 1 K×12 bit, 因此存储单元地址编码范围为 00 0000 0000B~11 1111 1111B(用十六进制表示时, 地址编码范围为 000H~3FFFH), 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。又如, 在内置 128 字节 RAM 的 MCS-51 单片机芯片中, 内部 RAM 存储单元的地址编码范围为 00H~7FH, 每个单元内容可以是 00H~FFH 之间的二进制数。

2) 存储器的工作状态

存储器芯片的工作状态由存储器控制信号的电平状态来决定, 如表 1-1 所示。

表 1-1 存储器的工作状态

工作模式	控制信号			输出
	片选信号 $\overline{\text{CS}}$	输出允许信号 $\overline{\text{OE}}$	写允许信号 $\overline{\text{WE}}$	
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 读取存储器中某一存储单元信息的操作过程, 如图 1-4 所示。

① CPU 地址寄存器 AR 给出将要读取的存储单元的地址信息, 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 存储单元。

④ CPU 给出读控制信号 $\overline{\text{RD}}$ (接存储器的 $\overline{\text{OE}}$ 端), 将选中的 0000H 存储单元的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 存储单元的内容 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, 被读出的存储单元原有信息将保持不变。

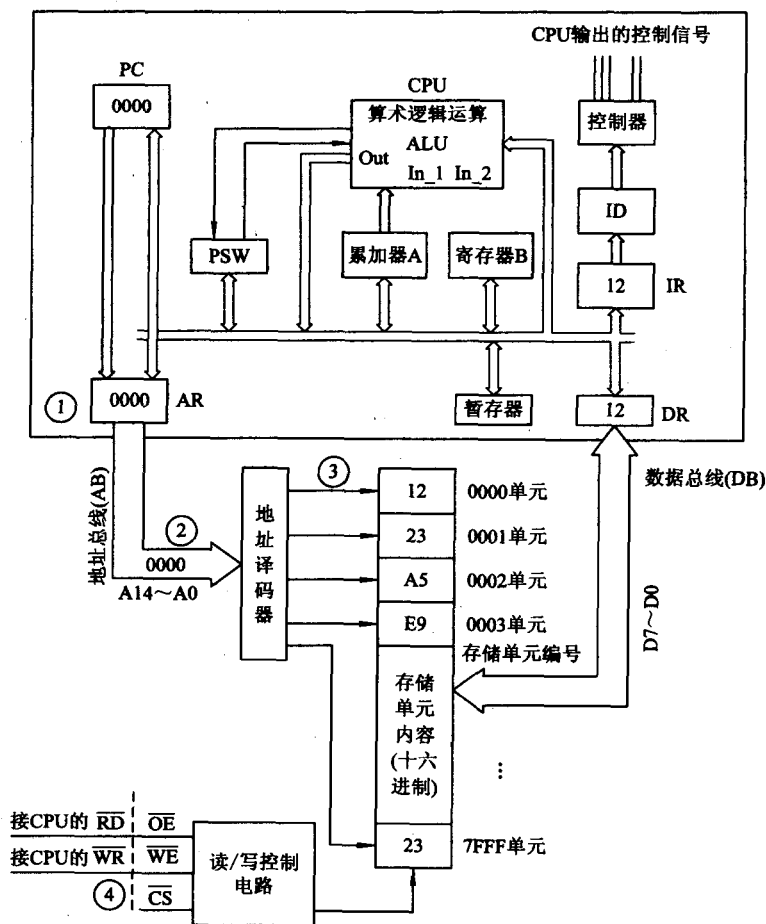


图 1-4 CPU 读取存储器操作过程示意图

4) 存储器写操作

存储器写操作是将某一数据写入存储器内某一存储单元的操作。例如，将 55H 写入 0003H 存储单元的操作过程如下：

① CPU 地址寄存器 AR 给出待写入的存储单元的地址编码，如 0003H。该地址编码信息通过地址总线 A15~A0 输入到存储器芯片地址线 A14~A0 上。

② 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 0003H 存储单元。

③ 在写操作过程中，写入的数据 55H 存放在 CPU 内的数据寄存器 DR 中，当 CPU 写控制信号 $\overline{WR}$ 有效时(与存储器写允许信号 $\overline{WE}$ 相连)，DR 寄存器中的内容 55H 就通过数据总线 D7~D0 传输到存储器中被选中的 0003H 存储单元，结果 0003H 存储单元的内容即刻变为 55H，这样就完成了存储器的写操作过程。

可见，执行写操作后，被写入的存储单元原有信息将不复存在。

1.1.2 指令及其指令系统

计算机通过执行一系列的指令来完成复杂的计算、判断、控制等操作过程。从 CPU 内