



高职高专“十一五”规划教材

# 80C51

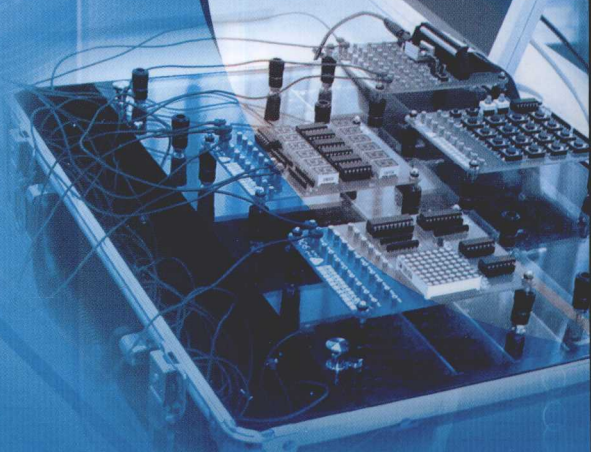
## 单片机及 接口技术

杜伟略 主编

周志德 主审



化学工业出版社



高职高专“十一五”规划教材

# 80C51 单片机及接口技术

杜伟略 主编

周志德 主审



化学工业出版社

· 北京 ·

全书共分 11 章。第 1、2 章介绍微机的基础知识与 80C51 单片机的组成,第 3、4 章介绍 80C51 单片机的指令系统及汇编语言程序设计方法。第 5 章介绍了当今较为流行 Keil C51 编译器及其应用。第 6 章介绍了单片机的中断系统及 80C51 单片机定时计数器。第 7 章讲述串行通信。第 8、9 章介绍并行、串行扩展技术。第 10 章重点介绍了单片机输入输出技术。第 11 章介绍应用系统设计。全书共安排了 9 个实训项目,穿插于不同章节。为保证教材的通用性,本书中设计的实训、实验项目一般高职高专院校都有条件完成,这也是为适应高职高专的教学改革要求,以便于采用“任务驱动”、“问题驱动”式教学方法。

本书可作为高职高专应用电子、电子信息工程、自动化、机电一体化、计算机应用等专业的教材,也可作为相关工程技术人员学习单片机技术的参考书。

编主 郭利林  
审主 谢志国

### 图书在版编目(CIP)数据

80C51 单片机及接口技术 / 杜伟略主编. —北京: 化学工业出版社, 2008.2

高职高专“十一五”规划教材

ISBN 978-7-122-02150-2

I. 8… II. 杜… III. ①单片微型计算机-基础理论-高等学校: 技术学院-教材 ②单片微型计算机-接口-高等学校: 技术学院-教材 IV. TP368.1

中国版本图书馆 CIP 数据核字 (2008) 第 020305 号

责任编辑: 廉 静 张建茹

装帧设计: 史利平

责任校对: 徐贞珍

出版发行: 化学工业出版社 (北京市东城区青年湖南街 13 号 邮政编码 100011)

印 刷: 大厂聚鑫印刷有限责任公司

装 订: 三河市延风装订厂

787mm×1092mm 1/16 印张 17¼ 字数 441 千字 2008 年 3 月北京第 1 版第 1 次印刷

购书咨询: 010-64518888 (传真: 010-64519686)

售后服务: 010-64518899

网 址: <http://www.cip.com.cn>

凡购买本书, 如有缺损质量问题, 本社销售中心负责调换。

定 价: 27.00 元

版权所有 违者必究

# 前 言

随着电子技术的迅猛发展,单片机技术已渗透到航天、国防、工业、农业、日常生活等各个领域,成为当今世界科技现代化不可缺少的重要工具和强有力武器。用单片机研制的各种智能化测量控制仪表周期短、成本低,在仪器、仪表与机电一体化产品的设计中具有明显的优势。80C51系列的单片机以其特有的简单、易学、易用、应用技术成熟、结构典型,成为初学单片机的首选机型。

本书是无锡职业技术学院单片机课程教学一线教师教学经验的结晶,这些教师工程实践背景强,均参加过智能化电子产品的开发和研制,多次带领学生参加全国大学生电子设计大赛,不仅是工程实践经验所得,同时也是教学中针对初学单片机的学生在学习中所遇到困难总结。本书具有如下特点。

① 通俗易懂。全书共安排了9个实训项目,分插于不同章节。为保证教材的通用性,本书中设计的实训、实验项目一般院校都有条件完成。本书以一个单片机最小的系统及简单的信号灯控制为载体,可以进行“任务驱动式”教学,讲解由浅入深,便于学生在感性的基础上学习抽象的知识。

② 注重应用。本书增加了现在热门的各种总线技术,如I<sup>2</sup>C、SPI 串行总线以及CAN总线等,选择80C51系列单片机为实例,适应当代主流单片机和外围接口器件的发展趋势。全书在适当的地方以“实用技术”的写法介绍了单片机设计过程中涉及的其他方面的知识。

③ 利于提高。现在企业大都用C51来编写程序,为适应社会的需求,本书中配有C51编程,有利于教学或学生自学。

④ 便于教学。每章都给出教学目标、学习重点,最后附有习题。

本书由无锡职业技术学院杜伟略主编,参加编写工作的还有潘健、王波、刘瑞涛。其中潘健编写第1、2、5章,王波编写第3、4、10章,杜伟略编写第6~9章,刘瑞涛编写第11章。全书由杜伟略统稿,周志德副教授审稿。

本书在编写过程中,得到相关学院及单位的大力支持,并提出了许多宝贵意见,编者再次表示感谢。

由于水平有限,书中不足之处在所难免,敬请读者批评指正。

编 者

2008年1月

# 目 录

|                                    |    |                               |    |
|------------------------------------|----|-------------------------------|----|
| <b>1 绪论</b> .....                  | 1  | 2.4.2 80C51 单片机指令的取指、<br>执行时序 | 34 |
| 1.1 数制与编码                          | 1  | 习题                            | 35 |
| 1.1.1 数制                           | 1  | 实训 仿真器的功能及使用                  | 36 |
| 1.1.2 编码                           | 4  | <b>3 80C51 单片机指令系统</b>        | 37 |
| 1.2 微型计算机的基本知识                     | 7  | 3.1 80C51 指令系统的概述             | 37 |
| 1.2.1 电子计算机的问世及其<br>经典结构           | 7  | 3.1.1 指令概述                    | 37 |
| 1.2.2 微型计算机系统的概念                   | 7  | 3.1.2 汇编指令的格式                 | 37 |
| 1.3 微型计算机的组成及工作原理                  | 8  | 3.2 寻址方式及指令分类                 | 38 |
| 1.3.1 微型计算机的组成                     | 8  | 3.2.1 寻址方式                    | 38 |
| 1.3.2 微型计算机的工作原理                   | 11 | 3.2.2 指令分类                    | 41 |
| 1.3.3 微型计算机系统组成                    | 13 | 3.3 80C51 单片机指令系统             | 41 |
| 1.4 单片机概述                          | 15 | 3.3.1 数据传送指令                  | 41 |
| 1.4.1 单片机的历史及发展概况                  | 15 | 3.3.2 算术运算指令                  | 47 |
| 1.4.2 单片机的发展趋势                     | 16 | 3.3.3 逻辑运算指令                  | 53 |
| 1.4.3 单片机的应用                       | 17 | 3.3.4 位操作指令                   | 57 |
| 1.4.4 80C51 系列单片机                  | 18 | 习题                            | 58 |
| 1.5 常用单片机功能简介                      | 18 | 实训 指令的应用                      | 60 |
| 1.5.1 Atmel 公司的 AT89 系列单片机         | 18 | <b>4 80C51 汇编语言程序设计</b>       | 64 |
| 1.5.2 Motorola 公司 M68HC08<br>系列单片机 | 19 | 4.1 汇编语言伪指令简介                 | 64 |
| 1.5.3 其他系列单片机                      | 19 | 4.2 汇编语言程序设计步骤与程序结构           | 66 |
| 习题                                 | 19 | 4.2.1 程序编制的步骤                 | 66 |
| 实训 单片机应用系统的开发过程演示                  | 20 | 4.2.2 编制程序的方法和技巧              | 67 |
| <b>2 80C51 单片机的硬件结构</b>            | 21 | 4.2.3 程序结构分类                  | 67 |
| 2.1 80C51 单片机的内部资源                 | 21 | 4.3 分支程序设计                    | 68 |
| 2.1.1 80C51 单片机的内部结构               | 21 | 4.3.1 无条件转移指令                 | 68 |
| 2.1.2 80C51 单片机的基本组成               | 29 | 4.3.2 条件转移指令                  | 70 |
| 2.2 80C51 单片机的端子功能                 | 30 | 4.3.3 分支程序设计实例                | 71 |
| 2.3 80C51 复位及复位电路                  | 32 | 4.4 循环程序设计                    | 73 |
| 2.3.1 复位结构                         | 32 | 4.4.1 循环程序的概念                 | 73 |
| 2.3.2 复位电路及复位操作                    | 32 | 4.4.2 循环程序结构                  | 74 |
| 2.4 80C51 单片机时序                    | 33 | 4.4.3 循环转移指令                  | 75 |
| 2.4.1 时钟周期、机器周期与指<br>令周期           | 33 | 4.4.4 循环程序设计实例                | 75 |
|                                    |    | 4.5 子程序设计与堆栈技术                | 78 |
|                                    |    | 4.5.1 堆栈技术                    | 78 |

|                      |                              |     |                      |                           |     |
|----------------------|------------------------------|-----|----------------------|---------------------------|-----|
| 4.5.2                | 子程序调用与返回指令                   | 80  | 6.3.4                | 定时/计数器的工作方式               | 130 |
| 4.5.3                | 子程序嵌套                        | 81  | 6.3.5                | 定时/计数器的应用举例               | 135 |
| 4.5.4                | 子程序应用举例                      | 82  | 6.4*                 | 中断与定时/计数器的 C 语言编程         | 139 |
| 4.6                  | 实用汇编子程序举例                    | 83  | 6.4.1                | 中断应用的 C 语言编程              | 139 |
| 4.6.1                | 查表程序设计                       | 83  | 6.4.2                | 中断与定时/计数器的 C 语言编程         | 141 |
| 4.6.2                | 显示程序设计                       | 84  | 6.4.3                | 中断的调试                     | 142 |
| 4.6.3                | 查找程序                         | 90  | 习题                   |                           | 143 |
| 4.6.4                | 代码转换程序                       | 91  | 实训                   | 信号灯的 control 2            | 144 |
| 4.6.5                | 数据排序程序                       | 91  |                      |                           |     |
| 习题                   |                              | 93  |                      |                           |     |
| 实训                   | 信号灯的 control 1               | 95  |                      |                           |     |
| <b>5 C51 编译器及其应用</b> |                              | 98  | <b>7 80C51 的串行接口</b> |                           | 146 |
| 5.1                  | C51 概述                       | 98  | 7.1                  | 计算机串行通信基础                 | 146 |
| 5.2                  | C51 的数据类型和变量定义               | 99  | 7.1.1                | 串行通信的基本概念                 | 147 |
| 5.2.1                | C51 的数据类型                    | 99  | 7.1.2                | 串行通信接口标准                  | 150 |
| 5.2.2                | C51 存储类型及存储模式                | 100 | 7.2                  | 80C51 串行接口                | 153 |
| 5.2.3                | C51 中的变量定义                   | 101 | 7.2.1                | 串行口寄存器结构                  | 153 |
| 5.3                  | 单片机集成开发环境                    | 104 | 7.2.2                | 80C51 单片机串行通信工作方式         | 155 |
| 5.3.1                | Keil $\mu$ Vision 集成开发环境     | 104 | 7.3                  | 80C51 串行接口通信技术            | 159 |
| 5.3.2                | Keil $\mu$ Vision 的组成及各部分的功能 | 104 | 7.3.1                | 80C51 双机通信技术              | 159 |
| 5.3.3                | Keil $\mu$ Vision 集成开发环境的使用  | 106 | 7.3.2                | 80C51 多机通信技术              | 165 |
| 习题                   |                              | 115 | 7.4                  | PC 机与 80C51 单片机通信技术       | 168 |
| 实训                   | Keil $\mu$ Vision 集成开发环境的使用  | 115 | 7.5*                 | 串行通信的 C 语言编程              | 170 |
|                      |                              |     | 习题                   |                           | 171 |
| <b>6 中断系统与定时/计数器</b> |                              | 117 | 实训                   | 单片机之间的双机通信                | 172 |
| 6.1                  | 中断系统                         | 117 |                      |                           |     |
| 6.1.1                | 中断的概述                        | 117 | <b>8 并行扩展技术</b>      |                           | 175 |
| 6.1.2                | 中断过程                         | 118 | 8.1                  | 并行扩展概述                    | 175 |
| 6.1.3                | 80C51 中断系统的总体结构              | 120 | 8.1.1                | 总线                        | 175 |
| 6.1.4                | 中断标志和中断控制                    | 121 | 8.1.2                | 扩展总线的实现                   | 176 |
| 6.1.5                | 中断响应                         | 124 | 8.2                  | 程序存储器扩展                   | 177 |
| 6.1.6                | 中断处理                         | 124 | 8.2.1                | 单片机程序存储器的概述               | 177 |
| 6.1.7                | 中断返回                         | 125 | 8.2.2                | EPROM 程序存储器扩展实例           | 179 |
| 6.2                  | 80C51 单片机的外部中断               | 125 | 8.2.3                | EEPROM 扩展实例               | 180 |
| 6.2.1                | 外部中断的应用实例                    | 125 | 8.3                  | 数据存储器扩展                   | 181 |
| 6.2.2                | 外部中断的扩展                      | 127 | 8.3.1                | 单片机 RAM 的概述               | 181 |
| 6.3                  | 80C51 单片机的定时/计数器             | 127 | 8.3.2                | SRAM 扩展实例                 | 182 |
| 6.3.1                | 概述                           | 128 | 8.3.3                | 新型存储器简介                   | 186 |
| 6.3.2                | 定时/计数器的定义                    | 128 | 8.4                  | 简单并行 I/O 接口的扩展            | 187 |
| 6.3.3                | 定时/计数器的控制寄存器                 | 129 | 8.4.1                | 80C51 系列单片机内部并行 I/O 口及其作用 | 187 |
|                      |                              |     | 8.4.2                | 简单的 I/O 口扩展               | 188 |

|        |                            |     |        |                   |     |
|--------|----------------------------|-----|--------|-------------------|-----|
| 8.5    | 并行接口芯片 8255                | 189 | 10.3.2 | A/D 转换器的性能指标      | 230 |
| 8.5.1  | 8255 的主要特性                 | 189 | 10.3.3 | ADC0809 芯片与单片机的接口 | 231 |
| 8.5.2  | 8255 的内部结构                 | 189 | 10.3.4 | AD574A 芯片与单片机的接口  | 235 |
| 8.5.3  | 8255 芯片端子及其功能              | 190 | 10.3.5 | MC14433 芯片与单片机的接口 | 237 |
| 8.5.4  | 8255 的控制字                  | 191 | 10.4   | 开关量接口             | 239 |
| 8.5.5  | 8255 与单片机接口实例              | 193 | 10.4.1 | 开关量输入接口           | 239 |
| 8.6    | RAM/IO 接口芯片 8155           | 196 | 10.4.2 | 开关量输出接口           | 241 |
| 8.6.1  | 8155 内部结构及其端子功能            | 196 | 10.5*  | A/D 转换的 C 语言编程    | 243 |
| 8.6.2  | 8155 内部寄存器及其编程             | 198 | 习题     |                   | 244 |
| 8.6.3  | 8155 内部定时/计数器              | 201 | 实训     | 用 ADC0809 对模拟量的采样 | 245 |
| 8.6.4  | 8155 应用举例                  | 202 |        |                   |     |
| 8.7    | 显示键盘接口技术                   | 203 | 11     | 单片机应用系统设计方法       | 246 |
| 8.7.1  | 显示接口技术                     | 203 | 11.1   | 单片机应用系统的组成        | 246 |
| 8.7.2  | 键盘接口技术                     | 205 | 11.2   | 单片机应用系统基本设计方法     | 246 |
| 8.8*   | 8155 的 C 语言编程              | 208 | 11.2.1 | 单片机应用系统基本设计方法     | 246 |
| 习题     |                            | 209 | 11.2.2 | 单片机应用系统硬件与软件设计    | 247 |
| 实训     | 单片机用 74LS273 进行并行 I/O 接口扩展 | 211 | 11.3   | 单片机应用系统的实用技术      | 249 |
| 9      | 串行扩展技术                     | 213 | 11.3.1 | 低功耗设计             | 249 |
| 9.1    | 串行扩展概述                     | 213 | 11.3.2 | 加密技术              | 251 |
| 9.1.1  | 串行扩展的种类                    | 213 | 11.4   | 单片机应用系统可靠性设计      | 252 |
| 9.1.2  | 串行扩展的特点                    | 213 | 11.4.1 | 电源、地线、传输干扰及其对策    | 252 |
| 9.2    | UART 串行扩展接口应用实例            | 214 | 11.4.2 | 硬件抗干扰措施           | 253 |
| 9.3    | I <sup>2</sup> C 串行扩展总线    | 216 | 11.4.3 | 软件抗干扰措施           | 255 |
| 9.4    | SPI 串行扩展接口                 | 218 | 11.5   | 单片机控制的产品计数装置      | 258 |
| 9.4.1  | SPI 总线的结构原理                | 218 | 11.5.1 | 问题的描述             | 258 |
| 9.4.2  | SPI 总线的软件模拟                | 219 | 11.5.2 | 问题分析              | 258 |
| 9.4.3  | SPI 串行扩展应用实例               | 220 | 11.6   | 电动机控制系统           | 259 |
| 习题     |                            | 221 | 11.6.1 | 步进电动机的功能          | 260 |
| 10     | 单片机应用系统输入输出接口技术            | 222 | 11.6.2 | 步进电动机的工作原理        | 260 |
| 10.1   | D/A 与 A/D 基本概念             | 222 | 11.6.3 | 控制系统硬件、软件设计       | 261 |
| 10.2   | D/A 转换器                    | 223 | 习题     |                   | 265 |
| 10.2.1 | D/A 转换的工作原理                | 223 | 附录     | 80C51 单片机指令速查表    | 266 |
| 10.2.2 | D/A 转换器的性能指标               | 224 | 参考文献   |                   | 269 |
| 10.2.3 | DAC0832 芯片与单片机的接口          | 226 |        |                   |     |
| 10.3   | A/D 转换器                    | 230 |        |                   |     |
| 10.3.1 | A/D 转换器的工作原理               | 230 |        |                   |     |

# 1 绪 论

## 教学目的

熟悉计算机数的表示法,了解编码的概念;掌握数制转换方法与带符号数的原码、反码、补码表示方法,以及带符号数的补码运算方法;掌握数字的 BCD 码、字符的 ASCII 码。了解单片机与微型计算机的基本概念,以及单片机的特点及其发展和应用。

## 学习重点

- ① 二进制数、十进制数和十六进制数之间的换算关系。
- ② 原码、反码和补码的表示方法。
- ③ ASCII 码和 BCD 码的基本概念。
- ④ 单片机概念及特点。

单片机或单片微机是单片微型计算机(Signal Chip Microcomputer)的简称。在刚开始学习单片机时,首先想知道的问题常常是“什么是单片机?它有哪些特点?它有哪些难点?怎样才能学好它?”等。本章就这些常识性问题进行了说明,要求掌握一些基本概念,例如:二进制与十六进制数的转换、补码的定义与作用、ASCII 码的定义与作用、微机结构简图画法、微机与微型计算机系统的区别。

## 1.1 数制与编码

微型计算机最基本的功能就是对数进行计算和处理,计算机是用器件的两种物理状态来表示数的变化,即利用器件的高电平表示“1”和器件的低电平表示“0”这两种状态。不同的 0 和 1 组合就可表示不同的数或值,如:10001000 就表示 88H。计算机在对数进行处理时只认 0 和 1,所以就有以下的不同数制转换。

### 1.1.1 数制

数制是计数的规则。人们使用最多的是进位计数制,数的符号在不同的位置上时所代表的数值不同。

#### (1) 进位计数制

按进位的方式计数的数制,称为进位计数制。

基数是指该进位计数制中允许选用的基本数码个数。

在进位计数中,一个数可以由有限个数码排列在一起构成,数码处在不同的数位上,所代表的数值是不同的,这个数码所表示的数值等于该数码本身乘以一个与它所在数位有关的常数,这个常数称为“位权”,简称“权”。以基数 R 为进位的计数制的进位原则是“逢 R 进一”。任何一种进位制数都可以表示成按位权展开的多项式之和的形式。

$$(X)_R = D_{n-1}R^{n-1} + D_{n-2}R^{n-2} + \cdots + D_1R^1 + D_0R^0 + D_{-1}R^{-1} + \cdots + D_{-m}R^{-m}$$

式中, X 为 R 进制数;  $D_i$  为数码( $i=n-1, n-2, \cdots, 1, 0, -1, \cdots, -m$ ); R 为基数; n 是整数位数;

$m$  是小数位数字；上标表示幂的次数。

为了区分不同计数制的数，常采用括号外面加数字下标的表示方法，或数字后面加相应的英文字母标识来表示。如二进制数的 101101 可表示为  $(101101)_2$  或  $(101101)_B$ 。

计算机内部采用二进制的原因：

- ① 易于物理实现；
- ② 二进制数运算规则简单；
- ③ 机器可靠性高；
- ④ 适合逻辑运算。

## (2) 单片机常用的数制

单片机中常用的数制有二进制、十进制和十六进制，部分十进制、二进制和十六进制数对照见表 1-1。

① 十进制数 (Decimal) 基数是 10，它有 10 个数码，即 0、1、2、3、4、5、6、7、8、9。其中最大数码 9，最小数码是 0。例如 123.45 按权展开如下

$$123.45 = 1 \times 10^2 + 2 \times 10^1 + 3 \times 10^0 + 4 \times 10^{-1} + 5 \times 10^{-2}$$

② 二进制数 (Binary) 基数为 2，它只有两个数码，即 0 和 1。例如 101101.11B 按权展开如下

$$\begin{aligned} 101101.11B &= 1 \times 2^5 + 0 \times 2^4 + 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 + 1 \times 2^{-1} + 1 \times 2^{-2} \\ &= 32 + 0 + 8 + 4 + 0 + 1 + 0.5 + 0.25 = 45.75 \end{aligned}$$

③ 十六进制数 (Hexadecimal) 它的基数为 16，它有 16 个数码，即 0、1、2、3、4、5、6、7、8、9、A、B、C、D、E、F。例如 FAH 按权展开如下

$$FAH = F \times 16^1 + A \times 16^0 = 15 \times 16 + 10 \times 1 = 250$$

十六进制数与二进制数相比，其优点是书写记忆方便，所以在微机编程等场合应用最广。

表 1-1 部分十进制、二进制和十六进制数对照

| 整 数 |      |      | 小 数      |          |      |
|-----|------|------|----------|----------|------|
| 十进制 | 二进制  | 十六进制 | 十进制      | 二进制      | 十六进制 |
| 0   | 0000 | 0    | 0        | 0        | 0    |
| 1   | 0001 | 1    | 0.5      | 0.1      | 0.8  |
| 2   | 0010 | 2    | 0.25     | 0.01     | 0.4  |
| 3   | 0011 | 3    | 0.125    | 0.001    | 0.2  |
| 4   | 0100 | 4    | 0.0625   | 0.0001   | 0.1  |
| 5   | 0101 | 5    | 0.03125  | 0.00001  | 0.08 |
| 6   | 0110 | 6    | 0.015625 | 0.000001 | 0.04 |
| 7   | 0111 | 7    |          |          |      |
| 8   | 1000 | 8    |          |          |      |
| 9   | 1001 | 9    |          |          |      |
| 10  | 1010 | A    |          |          |      |
| 11  | 1011 | B    |          |          |      |
| 12  | 1100 | C    |          |          |      |
| 13  | 1101 | D    |          |          |      |
| 14  | 1110 | E    |          |          |      |
| 15  | 1111 | F    |          |          |      |

## (3) 数制的转换

① 二进制数转换成十六进制数 方法是四位合一位，即以小数点为界，向左、向右每四位二进制数合一位十六进制数，不够四位补‘0’，将 11000011.10010100B 转换为十六进制数的方法如下

$$1100\ 0011.1001\ 0100B = C3.94H$$

↓ ↓ ↓ ↓  
C 3 . 9 4H

② 十六进制数转换为二进制数 方法是每位成四位法，以小数点为界每位十六进制数转换为 4 位二进制数，如

$$\begin{array}{cccc} A & 3 & . & E & 4H \\ \downarrow & \downarrow & & \downarrow & \downarrow \end{array}$$

$$1010\ 0011.1110\ 0100B = A3.E4H$$

## (4) 二进制数、十六进制数与十进制数的转换

① 二进制数、十六进制数转换成十进制数 转换方法是按权展开，然后按照十进制运算法则求和，如

$$1011.101B = 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 + 1 \times 2^{-1} + 0 \times 2^{-2} + 1 \times 2^{-3} = 11.625$$

$$DFC.8H = 13 \times 16^2 + 15 \times 16^1 + 12 \times 16^0 + 8 \times 16^{-1} = 3580.5$$

② 十进制数转换成二进制数、十六进制数。

a. 整数转换法。“除基数取余法”，即十进制整数不断除以转换进制基数，直到商为 0。每除一次取一个余数，从低位排向高位。如

十进制数转换成二进制数

|             |   |   |
|-------------|---|---|
| 53 ÷ 2 = 26 | 余 | 1 |
| 26 ÷ 2 = 13 | 余 | 0 |
| 13 ÷ 2 = 6  | 余 | 1 |
| 6 ÷ 2 = 3   | 余 | 0 |
| 3 ÷ 2 = 1   | 余 | 1 |
| 1 ÷ 2 = 0   | 余 | 1 |

由此可得：53=110101B

十进制数转换成十六进制数

|               |   |       |
|---------------|---|-------|
| 208 ÷ 16 = 13 | 余 | 0=0H  |
| 13 ÷ 16 = 0   | 余 | 13=DH |

由此可得：208=D0H

b. 小数转换法。“乘基数取整法”，即用转换进制的基数乘以小数部分，直至小数为 0 或达到转换精度要求的位数。每乘一次取一次整数，从最高位排到最低位。如

十进制转换成二进制数：

|                         |       |   |
|-------------------------|-------|---|
| $0.375 \times 2 = 0.75$ | 取整数部分 | 0 |
| $0.75 \times 2 = 1.5$   | 取整数部分 | 1 |
| $0.5 \times 2 = 1$      | 取整数部分 | 1 |

由此可得：0.375=0.011B

十进制转换成十六进制数：

|                          |       |   |
|--------------------------|-------|---|
| $0.625 \times 16 = 10.0$ | 取整数部分 | A |
|--------------------------|-------|---|

由此可得：0.625=0.AH

### 1.1.2 编码

在计算机内部是以二进制的编码来描述各种不同的数据（如数字、字母、字符等）。因此，对于输入计算机的数据必须进行编码，再通过输入设备输入到计算机中并进行处理和输出。下面介绍计算机中有符号数、字母、字符和数字的编码等问题。

#### (1) 有符号数的编码

数学中有符号数的正负号分别用“+”和“-”来表示。在计算机内部由于采用二进制，只有“1”和“0”两个数字。一般规定最高位是符号位，最高位为“0”表示正数，为“1”表示负数。在单片机中，以8位二进制数表示有符号数的编码有三种：原码、反码和补码。

① 原码 最高位用“0”表示正数的符号，用“1”表示负数的符号，其余位表示数值，这种表示法称为原码，如

$$X = +1010101B \quad Y = -1010101B$$

$$[X]_{\text{原}} = 01010101B \quad [Y]_{\text{原}} = 11010101B$$

前面一行的数称为真值，即为某数的实际有效值。后面一行是其对应的原码，分别用“0”和“1”代替了“+”号和“-”号。

② 反码 一个数的反码很容易由其原码求得。如果是正数，其反码与原码相同；如果是负数，则其反码的符号位和原码相同，数值位是原码数值位按位取反（即0变1，1变0），如

$$X = +1010101B \quad Y = -1010101B$$

$$[X]_{\text{反}} = 01010101B \quad [Y]_{\text{反}} = 10101010B$$

③ 补码 一个数的补码也很容易由其反码求得。如果是正数，则其补码与反码相同；如果是负数，则其补码为其反码末位加1，如

$$X = +1010101B \quad Y = -1010101B$$

$$[X]_{\text{反}} = 01010101B \quad [Y]_{\text{反}} = 10101010B$$

$$[X]_{\text{补}} = 01010101B \quad [Y]_{\text{补}} = 10101011B$$

补码在计算机中是一种重要的编码形式，它是为了简化有符号数的运算而引入的，有以下两个特点。

a. 与原码、反码不同，数值0的补码只有一个，即 $[0]_{\text{补}} = 00000000B$ 。

b. 若字长为8位，则补码所表示的范围为 $-128 \sim +127$ ；进行补码运算时，应注意所得结果不应超过补码所能表示数的范围。

这两个特点与模的概念和溢出概念有关。

• 模的概念。把一个计量单位称之为模或模数。例如，时钟是以12进制进行计数循环的，即以12为模。在时钟上，时针加上（正拨）12的整数位或减去（反拨）12的整数位，时针的位置不变。14点钟在舍去模12后，成为（下午）2点钟（ $14 - 12 = 2$ ）。从0点出发逆时针拨10格即减去10小时，也可看成从0点出发顺时针拨2格（加上2小时），即2点（ $0 - 10 = -10 = -10 + 12 = 2$ ）。因此，在模12的前提下，-10可映射为+2。由此可见，对于一个模数为12的循环系统来说，加2和减10的效果是一样的；因此，在以12为模的系统中，凡是减10的运算都可以用加2来代替，这就把减法问题转化成加法问题了（注：计算机的硬件结构中只有加法器，所以大部分的运算都必须最终转换为加法）。10和2对模12而言互为补数。

因此，由于计算机的运算部件与寄存器都有一定字长的限制（假设字长为8），当计数器计满8位二进制也就是256个数后会产生溢出，又从头开始计数。产生溢出的量就是计数器的模，显然，8位二进制数的计数器，它的模数为 $2^8 = 256$ 。

- 溢出的概念。如果运算结果超过了计数器的模所能表示的数值范围，称为溢出。

对于+0和-0，它们的原码、反码分别为

$$[+0]_{\text{原}}=00000000 \quad [-0]_{\text{原}}=10000000$$

$$[+0]_{\text{反}}=00000000 \quad [-0]_{\text{反}}=11111111$$

而其补码为

$$[+0]_{\text{补}}=00000000$$

$[-0]_{\text{补}}=[10000000]_{\text{补}}=[10000000]_{\text{反}}+1\text{B}=11111111\text{B}+1\text{B}=(1)00000000\text{B}=00000000\text{B}$ (最高位溢出了，符号位变成了0)。

因此，+0和-0的补码是一样的，都是00000000。对于10000000这个补码表示的是哪个数的补码呢？

其实这是一个规定，这个数表示的是-128，所以8位二进制补码能够表示的范围是-128~+127，比8位二进制原码能表示的数多一个。

## (2) 有符号数的运算

原码虽然比较简单、直观，但采用原码进行加、减运算时，符号位与数值位无法一起参与运算，计算机的电路比较复杂。但是如果采用补码，使符号位能与数值一起参加运算，并且将减法运算转化为加法运算，从而简化计算机电路。

补码加法规则为

$$[X+Y]_{\text{补}}=[X]_{\text{补}}+[Y]_{\text{补}}$$

$$[X]_{\text{原}}=[X]_{\text{补}}$$

$$[X-Y]_{\text{补}}=[X]_{\text{补}}-[-Y]_{\text{补}}=[X]_{\text{补}}+[-Y]_{\text{补}}$$

如计算  $24-11=24+[-11]_{\text{补}}=13$ 。

用二进制运算如下：24的8位二进制原码、反码和补码都是00011000B；-11=-1011B，-11的8位二进制原码为10001011B，反码为11110100B，补码为11110101B。

$$00011000\text{B}$$

$$+ 11110101\text{B}$$

$$(1)00001101\text{B}$$

因为在8位机中，最高位D<sub>7</sub>的进位已超出计算机字长（8位）的范围，产生了溢出。由此可见，在不考虑最高位产生进位的情况下，做减法运算与补码相加的结果完全相同的。

## (3) ASCII (American Standard Coded for Information Interchange) 码

计算机除了经常处理数字信息外，还需要处理大量字母和符号信息，如字母a~z、符号及控制字符等。这些信息都要用特定的二进制码表示。目前在计算机中普遍采用的是ASCII编码，即美国信息变换标准代码。ASCII编码表见表1-2。

ASCII码由7位二进制编码组成，共有128个字符，将其分为两类：一类是控制字符，共34个，在ASCII编码中，前33个和最后一个编码都是控制字符；另一类是各种字符和符号，共94个。在使用时一般用一个8位字节的7位(b<sub>0</sub>~b<sub>6</sub>)表示，最高位(b<sub>7</sub>)用0表示或者用于奇偶校验，称为奇偶校验位。最高位也可以固定为1，用于表示一些图形符号。

## (4) 二-十进制编码

二-十进制数称为二进制编码的十进制数，简称BCD (Binary Coded Decimal) 码。BCD码的种类比较多，常用的有8421码、2421码等，其中8421码是采用10个四位二进制数0000B~1001B给10个十进制数0~9编码。例如，十进制数96用8421BCD码表示，即为10010110B。使用这种数制既考虑了计算机的特点，又顾及到人们使用计算机的习惯。在单

片机中输入和输出数据时采用这种数制。

表 1-2 ASCII 编码表

| ASCII 值 | 控制字符 | ASCII 值 | 控制字符    | ASCII 值 | 控制字符 | ASCII 值 | 控制字符 |
|---------|------|---------|---------|---------|------|---------|------|
| 0       | NUT  | 32      | (space) | 64      | @    | 96      | ,    |
| 1       | SOH  | 33      | !       | 65      | A    | 97      | a    |
| 2       | STX  | 34      | "       | 66      | B    | 98      | b    |
| 3       | ETX  | 35      | #       | 67      | C    | 99      | c    |
| 4       | EOT  | 36      | \$      | 68      | D    | 100     | d    |
| 5       | ENQ  | 37      | %       | 69      | E    | 101     | e    |
| 6       | ACK  | 38      | &       | 70      | F    | 102     | f    |
| 7       | BEL  | 39      | ,       | 71      | G    | 103     | g    |
| 8       | BS   | 40      | (       | 72      | H    | 104     | h    |
| 9       | HT   | 41      | )       | 73      | I    | 105     | i    |
| 10      | LF   | 42      | *       | 74      | J    | 106     | j    |
| 11      | VT   | 43      | +       | 75      | K    | 107     | k    |
| 12      | FF   | 44      | ,       | 76      | L    | 108     | l    |
| 13      | CR   | 45      | -       | 77      | M    | 109     | m    |
| 14      | SO   | 46      | .       | 78      | N    | 110     | n    |
| 15      | SI   | 47      | /       | 79      | O    | 111     | o    |
| 16      | DLE  | 48      | 0       | 80      | P    | 112     | p    |
| 17      | DC1  | 49      | 1       | 81      | Q    | 113     | q    |
| 18      | DC2  | 50      | 2       | 82      | R    | 114     | r    |
| 19      | DC3  | 51      | 3       | 83      | X    | 115     | s    |
| 20      | DC4  | 52      | 4       | 84      | T    | 116     | t    |
| 21      | NAK  | 53      | 5       | 85      | U    | 117     | u    |
| 22      | SYN  | 54      | 6       | 86      | V    | 118     | v    |
| 23      | TB   | 55      | 7       | 87      | W    | 119     | w    |
| 24      | CAN  | 56      | 8       | 88      | X    | 120     | x    |
| 25      | EM   | 57      | 9       | 89      | Y    | 121     | y    |
| 26      | SUB  | 58      | :       | 90      | Z    | 122     | z    |
| 27      | ESC  | 59      | ;       | 91      | [    | 123     | {    |
| 28      | FS   | 60      | <       | 92      | \    | 124     |      |
| 29      | GS   | 61      | =       | 93      | ]    | 125     | }    |
| 30      | RS   | 62      | >       | 94      | ^    | 126     | ~    |
| 31      | US   | 63      | ?       | 95      | _    | 127     | DEL  |

BCD 码与十进制数的相互转换,按照 BCD 的十位编码与十进制的对应关系来实现。BCD 码与二进制之间的转换不是直接的,要先转换为十进制,然后再转换为二进制;反之,过程类似。8421BCD 编码表如表 1-3 所示。

表 1-3 8421BCD 编码表

| 十进制数 | 8421 码 | 十进制数 | 8421 码 | 十进制数 | 8421 码   | 十进制数 | 8421 码   |
|------|--------|------|--------|------|----------|------|----------|
| 0    | 0000   | 4    | 0100   | 8    | 1000     | 12   | 00010010 |
| 1    | 0001   | 5    | 0101   | 9    | 1001     | 13   | 00010011 |
| 2    | 0010   | 6    | 0110   | 10   | 00010000 | 14   | 00010100 |
| 3    | 0011   | 7    | 0111   | 11   | 00010001 | 15   | 00010101 |

BCD 码的加法运算按照“逢十进一”原则相加,当两个 BCD 数相加时,其相加结果在本位上不大于 1001B。由于计算机进行二进制加法时,它在两个相邻 BCD 码之间只能按照“逢十六进一”原则计数,故在计算机进行 BCD 加法时,需要对二进制加法的结果进行修正,使 BCD 码的相加能够实现“逢十进一”原则计数。修正的原则是:如果相加结果的低 4 位大于 9,或者是相加时低 4 位向高 4 位发生进位,则低 4 位加 0110B 修正;如果高 4 位大于 9 时,或者高 4 位的最高位发生进位,则高 4 位加 0110B 修正。修正过程是由专门的十进制修正指令来控制十进制修正电路完成的。

## 1.2 微型计算机的基本知识

计算机之所以能在现代社会中起着极其重要的作用,主要是由它的卓越性能决定的。具有如下特点。

① 高速度 计算机能以人无法比拟的高速度进行信息处理。计算机的运算速度大于每秒几十万次,有些巨型机已达每秒百万亿次。

② 高度自动化 计算机能在程序的控制下,无需人的介入自动处理信息。

③ 具有存储能力 计算机能存储大量的信息,一般计算机能在机内存储几百万、几千万、几亿甚至百亿字节的信息。

④ 具有逻辑判断能力 计算机可进行各种逻辑判断,并根据判断的结果自动决定下一步的工作。

⑤ 高精度和高可靠性 用计算机处理得到的结果,数据的有效位数可达十几位,甚至上百位。计算机的可靠性高,可无故障地连续运行数万小时。

### 1.2.1 电子计算机的问世及其经典结构

电子计算机是一种能对信息进行加工处理的机器,具有记忆、判断和运算能力,能存储和自动运行程序,从而可以代替人的部分脑力劳动,并能对生产过程实施控制。按照冯·诺依曼理论,计算机由控制器、运算器、存储器、输入设备和输出设备五大部分组成。

从 1946 年美国宾夕法尼亚大学研制出世界上第一台电子计算机“ENIAC”以来,电子计算机经历了电子管、晶体管、集成电路和超大规模集成电路四个发展时代。随着第四代超大规模集成电路的发展以及集成化程度的不断提高,微型计算机应运而生,使得第四代电子计算机包含巨型机、小巨型机、主机、小型机和个人计算机五类机型,从而满足了不同需求。由于结构简单、通用性强、价格便宜,微型计算机已成为现代计算机领域中的一个极为重要的分支。

### 1.2.2 微型计算机系统的概念

在 20 世纪 70 年代,计算机的发展进入超大规模集成电路时代,由此产生了微处理器和微型计算机。微型计算机的产生、发展和日益广泛的应用,使人类社会发生了翻天覆地的变化。随着微型计算机的高速发展,单片微型计算机、微型计算机系统和计算机网络工作站等新机种不断涌现。

为了学习并掌握好微型计算机,首先掌握微型计算机系统的概念。通常说的微处理器、微型计算机、微型计算机系统,是从局部到全局构成微型计算机系统的三个层次,弄清这三个层次的含义非常重要。

### 1.1 微处理器 (Microprocessor Unit)

微处理器就是将中央处理单元 (CPU)、控制器和运算器集成在一片半导体芯片上, 称 MPU, 简称为 MP。微处理器中主要包含控制部件、算术逻辑单元和寄存器组三部分, 是构成微型计算机的核心部件。

### (2) 微型计算机 (Microcomputer)

微型计算机就是以微处理器 MPU 为核心, 再配以相应的半导体存储器 (RAM 和 ROM)、I/O (Input 和 Output) 接口和中断系统等, 并由系统总线连接起来组装在一块或数块印刷电路板上所构成的计算机, 通常包括以下几种类型。

① 多板微型计算机 (Multiboard Microcomputer) 多板微型计算机是把构成微型计算机的各功能部件分别组装在多块印刷电路板 (如存储器扩充板、显示器板) 上, 再通过同一机箱内的总线插槽连成一体微型计算机。多板微型计算机功能很强, 通过选用不同的印刷电路插件板达到不同的使用目的。

② 单板微型计算机 (Single Board Microcomputer) 单板微型计算机是把微处理器、半导体存储器、I/O 接口和中断电路等芯片组装在一块印刷电路板上的微型计算机。在这块印刷电路板上, 通常还装有简易键盘和发光二极管, 只读存储器中还固化有容量不大的监控程序。单板微型计算机常做成专用的过程控制机投放市场。

③ 单片微型计算机 (Single Chip Microcomputer) 单片微型计算机是把微处理器、半导体存储器、I/O 接口和中断系统集成在一块硅片上的具有完整功能的微型计算机。单片微型计算机具有体积小、重量轻、价格低和可靠性高等优点, 常在家用电器、智能仪器仪表和工业控制领域中应用。

### (3) 微型计算机系统 (Microcomputer System)

以微型计算机为核心, 再配以相应的外围设备、电源、辅助电路和控制微型计算机工作的软件就构成了完整的微型计算机系统。微型计算机系统由硬件系统和软件系统两大部分组成。

从上述三个概念可知, 微处理器不是计算机, 单纯的微型计算机也不是一个计算机系统, 它们都不能单独工作, 只有微型计算机系统才是一个完整的系统, 可以正常工作。

## 1.3 微型计算机的组成及工作原理

微型计算机是随着大规模集成电路的发展而产生的一种新型计算机, 它在结构上与通用计算机十分相似, 但也有独到之处。

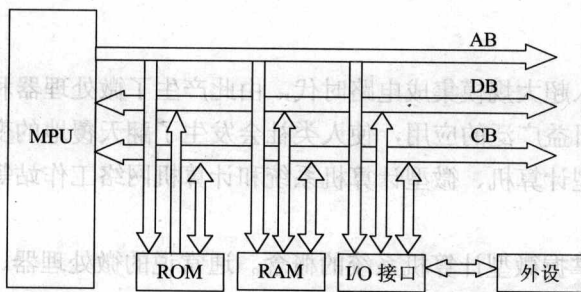


图 1-1 微型计算机三总线结构

### 1.3.1 微型计算机的组成

微型计算机以微处理器为核心, 其一大特点是采用总线结构, 其中三总线结构尤为普遍, 如图 1-1 所示。MPU (微处理器) 是通过 AB (地址总线)、DB (数据总线) 和 CB (控制总线) 与 ROM (只读存储器)、RAM (随机存取存储器) 及 I/O (输入/输出) 接口相连的。在分析微型计算机基本工作原理前先

对其各部件进行介绍。

### (1) 存储器

存储器又称内存或主存,是计算机的存储部件,用来存放程序和数据(原始数据、中间结果及最后结果)。微型计算机内存由半导体存储器 ROM 和 RAM 组成,它们都是采用大规模或超大规模集成电路工艺制成的存储器芯片,具有体积小、重量轻、集成度高等特点。

只读存储器 ROM (Read Only Memory) 是一种在正常工作时只读不能写的存储器,利用特殊手段将固定程序和常数写入 ROM,一旦写入便能长期保存(断电也不丢失),需要时即可读出使用。ROM 通常用来存放不变的程序,如基本 I/O 程序、监控程序、汇编程序、高级语言/解释程序等,还存储各种常用数据和表格。ROM 的种类很多,按照编程方式,分为掩模 ROM、可编程 ROM、可擦除 EPROM、电擦除 E<sup>2</sup>PROM 和闪速存储器 Flash PEROM。Flash PEROM 可以在线写入(自动擦除),并且可以按页连续字节写入,读出速度也很快。

随机存取存储器 RAM (Random Access Memory) 是一种在正常工作时既能读又能写的存储器,用来存放用户要运行的程序及运行程序所需的原始数据、中间结果、最终结果以及实时数据等。RAM 中存储的信息不能长久保存,停电后便立即消失,因此又称为易失性或易挥发性存储器。RAM 存储器的规格品种也很多,如 4KB、8KB、32KB、64KB、256KB、1MB 等。按照存储电路的不同,随机存储器 RAM 又可分为动态 RAM (DRAM)、静态 RAM (SRAM)。DRAM 芯片以 MOS 管栅极电容是否充有电荷来存储逻辑 1 或 0 信息,需要外接刷新电路来周期性地对其电荷进行刷新以维持它所代表的数据。SRAM 芯片以触发器来存储逻辑 1 或 0 信息,无需刷新电路。SRAM 比 DRAM 集成度低、成本高、速度快。SRAM 和 DRAM 均为易失性存储器。

ROM 和 RAM 的主要区别在于:一是断电后 ROM 内的信息不丢失,而 RAM 中的信息立即消失;二是读/写方式不同,ROM 采用特殊方式写入信息,信息写入后只能读,而 RAM 既能读又能写。

ROM 和 RAM 存储器芯片上的引线通常分为四组:地址线用于向存储器输送地址码;数据线是双向的,传送对芯片中某一单元进行读/写操作的数据;控制线用于传送控制信号,控制操作是进行读还是进行写等;电源线供给芯片+5V 直流电源,其中地址线及数据线的端子数与芯片的存储容量有关。

地址线越多,存储容量就越大;数据线越多,每个单元中存储的二进制数的位数就越长。例如,地址线为  $n$  条的存储器芯片的存储容量为  $2^n$  个单元,数据线的条数  $m$  为每个单元能存储的二进制数的位数,则存储器芯片的存储容量为  $2^n \times m$  bit,即

$$\text{存储容量} = 2^{\text{地址线的条数}} \times \text{数据线的条数 (bit)}$$

如一个存储器有 16 条地址线和 8 条数据线,它的 16 位地址可得到  $2^{16}$  个地址编码,可寻址  $2^{16}$  个单元,即存储容量为  $2^{16}$  个单元,每个单元 8 位,则存储容量为  $2^{16} \times 8$  bit。

### (2) 微处理器 (MPU)

微处理器 MPU 是微型计算机的运算和指挥控制中心,其性能决定了微型计算机的性能。虽然各种微处理器性能和内部结构有差异,指令系统也不同,但它们的基本组成相同,都包含控制器、运算器和内部总线。微处理器与存储器之间的结构可参照图 1-2。

① 运算器 运算器在控制器的控制下对二进制数进行算术运算或逻辑操作。运算器由算术逻辑单元 ALU、累加器 A、暂存器 TMP、程序状态字 PSW 和通用寄存器 PS 五部分组成。

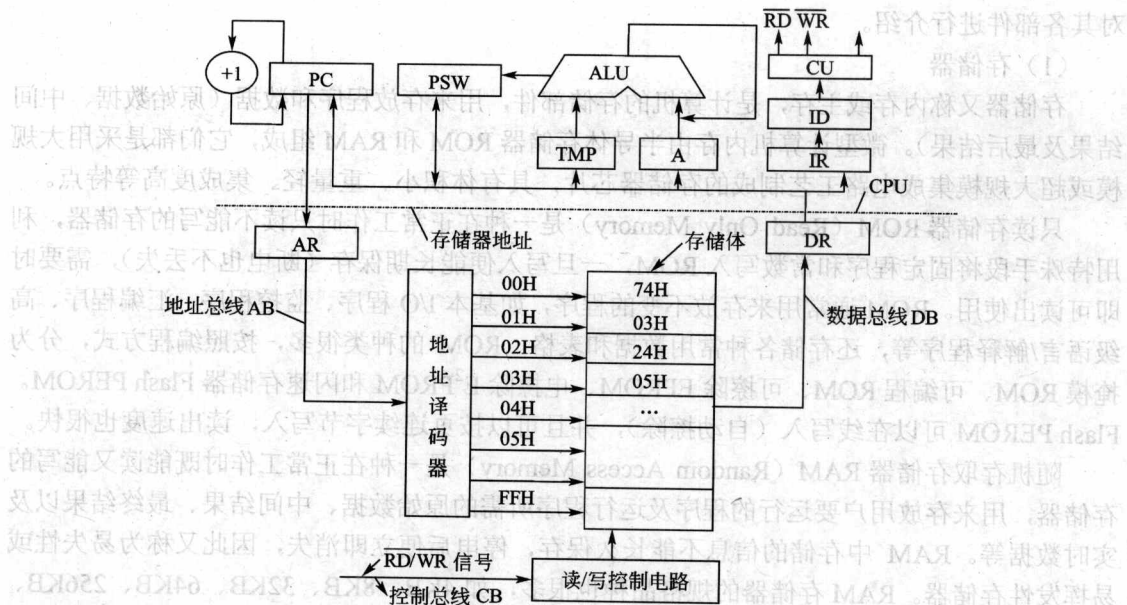


图 1-2 MPU 与存储器结构示意图

a. 算术逻辑单元 ALU ALU 是运算器的核心，它以全加器为基础，并辅以移位和控制逻辑组合而成，在控制信号的控制下，可进行加减乘除等算术运算和各种逻辑运算。

b. 累加器 A A 是由触发器组成的移位寄存器，在运算前存放一个操作数，运算后存放运算结果。

c. 暂存器 TMP TMP 也是一个寄存器，用于暂存另一操作数。

d. 程序状态字 PSW PSW 由 8 位触发器组成，存放 ALU 操作过程中形成的状态。例如：累加器 A 中的运算结果是否为零，最高位是否有进位或借位，低 4 位向高 4 位是否有进位或借位等，都在 PSW 中相应位上进行标志。

e. 通用寄存器 PS PS 用于存放操作数或运算结果。

在运算前，将运算所需操作数送入累加器和暂存器，由控制信号控制 ALU 进行运算，把运算结果存放到累加器 A，并根据运算结果设置程序状态字 PSW 状态位的值。

② 控制器 控制器是发布操作命令的机构，是计算机的指挥中心，相当于人脑的神经中枢，控制计算机的各部分协调工作，用于自动执行程序。控制器由指令部件、时序部件和微操作控制部件三部分组成。

a. 指令部件 指令部件是用来读取指令、分析指令和为完成指令产生控制信号的逻辑部件，也是控制器的核心。通常指令部件由程序计数器 PC、指令寄存器 IR 和指令译码器 ID 等三部分组成。

b. 时序部件 时序部件由时钟系统和脉冲分配器组成，用于产生微操作控制部件所需的定时脉冲信号。时钟系统产生机器的时钟脉冲序列，脉冲分配器又称节拍发生器，用于产生节拍电位和节拍脉冲。

c. 微操作控制部件 微操作控制部件为指令译码器 m 的输出信号配上节拍电位和节拍脉冲，也可和外部进来的控制信号组合，共同形成相应的微操作控制序列，以完成规定的操作。

计算机的工作就是执行程序。要执行一个程序必须将该程序放入内存，而程序是若干指令的有序排列，要执行程序只要从第一条指令开始，逐条读取指令、分析指令、执行指令直