

# 微型计算机 原理与 接口技术

● 张新义 江世成 主编

机械工业出版社

TP364  
21271

379629

# 微型计算机原理与接口技术

主 编 张新义 江世成

副主编 杜守旭 刘世美 韩进宏



机械工业出版社

(京) 新登字 054 号

本书以 Z-80 和 MCS-51 单片机为主线, 介绍其中央处理器的功能组成和应用。本书内容突出功能与接口、原理与应用并重的特点, 放弃原来以一种系列芯片 (Z-80 或单片机) 为主的教材体系, 把单板机、单片机原理与应用合为一体, 取材丰富、完整。

主要内容有: 微型计算机的基本结构和数制、中央处理器、指令系统、汇编语言程序设计、存储器及其接口、中断处理技术、可编程芯片及接口、外设接口技术、微机应用系统抗干扰设计和应用实例。

本书可作为高等学校非计算机类各专业本、专科教材, 也可作为职大、电大、函大及其职业技术学校的教材。同时也非常适合对于需要微型机知识的广大科技人员和管理干部自学。

JS159/14

#### 图书在版编目 (CIP) 数据

微型计算机原理与接口技术 / 张新义, 江世成主编. — 北京: 机械工业出版社, 1994.12  
ISBN 7-111-04304-9

I. 微...

II. ①张...②江...

III. 微型计算机-接口

IV. TP36



中国版本图书馆 CIP 数据核字 (94) 第 000000 号

出版人: 马九荣 (北京百万庄南街 15 号 邮政编码 100037)  
责任编辑: 徐 彤 版式设计: 张世琴 责任校对: 肖新民  
封面设计: 郭景云 责任印制: 路 琳  
机械工业出版社印刷厂印刷 · 新华书店北京发行所发行  
1994 年 12 月第 1 版 · 1994 年 12 月第 1 次印刷  
787mm × 1092mm<sup>1</sup> / 16 · 24.75 印张 · 604 千字  
0 001- 6 000 册  
定价: 20.00 元

# 前 言

随着微型计算机技术的不断发展与成熟,微型计算机的应用已深入到各个领域。在我国的高等院校,尤其是理、工、农、医学院,已将微型计算机原理与应用课程列为必修课,并出版了不少这类优秀教材,为我国的计算机应用和普及起到了积极作用。近几年来,单片机技术异军突起,已成为微型计算机在工业控制、智能化仪器仪表等方面应用的主流。由于目前的微机原理与应用教材多以 Z-80 微处理器为主的单板机体系,以使使学生所学与所用尚不一致。为适应单片机技术的发展与应用,很多高校在原微机原理与应用课程之外,补充或再开设一门单片机原理与应用课程。实践中使我们感到,两门课程其研究方法、内容体系都有很多共性与交叉;独立开设,势必部分内容重叠,造成学时浪费。为此,在研究了两门课程的结构和特点、内容和体系、理论和应用的联系后,我们把它们合二为一,编写了这本微型计算机原理与接口技术教材。

本书以 Z-80 和 MCS-51 微处理器为主线,内容安排上强调以基本理论、基本概念为主,并注重实际应用的原则;选材尽可能完整、详尽、实用和系统。全书共分十章,第一章至第五章是基本原理部分,不仅较详细地介绍了微型计算机的发展、基本结构、中央处理器和半导体存储器的类型结构及应用,而且重点讨论了 Z-80 和 MCS-51 系列的指令系统及其汇编语言程序设计方法。第六章至第八章是接口技术部分,着重介绍实际应用中的中断处理技术,可编程并、串行芯片及其接口以及外部设备的接口技术;并以 Z-80 和 MCS-51 单片机两种不同规模的系统组成和应用接口技术为重点,力求达到一点(主线)突破、触类旁通的效果。第九章、第十章是应用部分,针对微机应用系统设计中的实际问题,介绍硬、软件设计的抗干扰技术,应用系统的设计方法步骤及实例。通过本教材的学习,可以对 Z-80 为核心的单板机及其 MCS-51 单片机系列,都有一个较为全面的了解和认识,并具备一定的设计微型计算机应用系统的能力。

本书组织多年讲授微机原理与应用课程的教师编写,由张新义、江世成任主编,杜守旭、刘世美、韩进宏任副主编,张新义负责统稿。各章的编者如下:第一章薛广兰;第二章张新义;第三章刘世美,第四章马龙军;第五章李庆中;第六章江世成;第七章韩进宏;第八章辛世界;第九章高松;第十章杜守旭。

本书编写过程中,参考了近几年出版的同类优秀教材及应用成果,编者在此向收录于本书参考文献中的各位编著者表示真诚的谢意。

由于作者水平有限,可能存在许多错误和疏漏之处,恳请读者指正。

编 者  
1994 年 2 月

# 目 录

|                          |    |
|--------------------------|----|
| 前言                       |    |
| 第一章 微型计算机基础知识与基本结构       | 1  |
| 第一节 微型计算机的发展及应用          | 1  |
| 一、微型计算机的发展概况             | 1  |
| 二、微型计算机的应用               | 2  |
| 第二节 微型计算机的基本组成和结构        | 3  |
| 一、微型计算机的基本组成             | 3  |
| 二、微型计算机的基本结构             | 4  |
| 第三节 微型计算机的数制             | 6  |
| 一、进位计数制                  | 6  |
| 二、各种进制数的转换               | 6  |
| 三、带符号数的表示                | 8  |
| 四、数的定点与浮点表示              | 10 |
| 思考与练习题                   | 10 |
| 第二章 中央处理器                | 12 |
| 第一节 Z-80CPU 的组成和结构       | 12 |
| 一、Z-80CPU 的基本组成          | 12 |
| 二、Z-80CPU 寄存器组的内部结构及功能   | 12 |
| 三、Z-80CPU 的引脚及功能         | 15 |
| 第二节 Z-80CPU 的基本时序信号      | 17 |
| 一、取指令周期                  | 18 |
| 二、存储器读/写周期               | 18 |
| 三、输入/输出周期                | 20 |
| 四、总线请求/响应周期              | 21 |
| 五、中断请求/响应周期              | 21 |
| 六、非屏蔽中断请求/响应周期           | 21 |
| 七、暂停状态的退出周期              | 23 |
| 第三节 单片机的发展和结构            | 23 |
| 一、单片机的发展概况               | 23 |
| 二、MCS-51 单片机的结构与功能       | 23 |
| 第四节 MCS-51 单片机 CPU 的基本时序 | 32 |
| 第五节 各种 CPU 应用特点          | 33 |
| 一、通用 CPU 应用系统            | 34 |
| 二、单片机应用系统                | 34 |
| 思考与练习题                   | 35 |
| 第三章 指令系统                 | 36 |
| 第一节 概述                   | 36 |
| 一、指令的基本格式                | 36 |
| 二、寻址方式                   | 36 |
| 三、指令系统的分类                | 38 |
| 第二节 Z-80 指令系统            | 38 |
| 一、数据传送类指令                | 38 |
| 二、数据操作类指令                | 48 |
| 三、程序控制类指令                | 56 |
| 四、CPU 控制类指令              | 61 |
| 第三节 MCS-51 单片机指令系统       | 64 |
| 一、数据传送类指令                | 64 |
| 二、算术操作类指令                | 68 |
| 三、逻辑操作类指令                | 70 |
| 四、控制程序转移类指令              | 73 |
| 五、布尔变量操作类指令              | 78 |
| 思考与练习题                   | 81 |
| 第四章 汇编语言程序设计             | 83 |
| 第一节 汇编语言指令格式与伪指令         | 83 |
| 一、低级语言与高级语言              | 83 |
| 二、汇编语言指令格式               | 85 |
| 三、伪指令                    | 86 |
| 第二节 汇编语言程序设计方法           | 87 |
| 一、汇编语言程序的开发过程            | 87 |
| 二、程序流程图                  | 87 |

|                                        |     |                               |     |
|----------------------------------------|-----|-------------------------------|-----|
| 三、汇编程序简介 .....                         | 88  | 单片机的接口 .....                  | 148 |
| 四、程序的基本结构 .....                        | 89  | 一、程序存储器 ROM 与 MCS-51 .....    |     |
| 五、子程序 .....                            | 90  | 单片机的接口 .....                  | 148 |
| 六、程序基本结构和子程序编程 .....                   |     | 二、数据存储器 RAM 与 MCS-51 .....    |     |
| 举例 .....                               | 91  | 单片机的接口 .....                  | 154 |
| 第三节 Z-80 汇编语言程序 .....                  |     | 思考与练习题 .....                  | 158 |
| 设计举例 .....                             | 102 | <b>第六章 中断处理技术</b> .....       | 159 |
| 一、转移表程序 .....                          | 102 | 第一节 输入输出的传送方式 .....           | 159 |
| 二、BCD 码—二进制数转换程序 .....                 | 103 | 一、无条件传送方式 .....               | 159 |
| 三、多位十进制数的加减法运算 .....                   | 104 | 二、条件传送方式 .....                | 159 |
| 四、查表方法 .....                           | 106 | 三、中断传送方式 .....                | 159 |
| 第四节 MCS-51 汇编语言程 .....                 |     | 四、直接数据通道传送方式 (DMA) ...        | 159 |
| 序设计举例 .....                            | 111 | 第二节 中断的功能及优先级 .....           | 160 |
| 一、原码加、减法运算程序 .....                     | 112 | 一、中断的功能 .....                 | 160 |
| 二、无符号二进制数乘法 .....                      | 113 | 二、中断源 .....                   | 160 |
| 三、无符号双字节快速乘法 .....                     | 115 | 三、中断优先级 .....                 | 161 |
| 四、无符号二进制除法 .....                       | 116 | 第三节 Z-80CPU 的中断方式 .....       | 161 |
| 五、查表程序 .....                           | 118 | 一、中断方式 .....                  | 161 |
| 六、命令序号查找程序 .....                       | 118 | 二、中断优先级排队 .....               | 164 |
| 七、软件模拟逻辑电路程序 .....                     | 120 | 第四节 MCS-51CPU 的中断 .....       |     |
| 八、浮点数运算 .....                          | 121 | 系统 .....                      | 166 |
| 思考与练习题 .....                           | 127 | 一、中断源 .....                   | 167 |
| <b>第五章 存储器及其与 CPU</b> .....            |     | 二、中断控制 .....                  | 167 |
| 的接口技术 .....                            | 129 | 三、中断优先级控制 .....               | 168 |
| 第一节 存储器概述 .....                        | 129 | 第五节 中断响应条件及过程 .....           | 169 |
| 第二节 读写存储器 RAM .....                    | 130 | 一、Z-80CPU .....               | 169 |
| 一、静态 RAM (SRAM) .....                  | 130 | 二、MCS-51 单片机 .....            | 170 |
| 二、动态 RAM (DRAM) .....                  | 132 | 思考与练习题 .....                  | 171 |
| 第三节 只读存储器 ROM .....                    | 133 | <b>第七章 可编程芯片及其与 CPU</b> ..... |     |
| 一、掩模式 ROM .....                        | 133 | 接口技术 .....                    | 172 |
| 二、一次编程的 ROM (PROM) .....               | 134 | 第一节 PIO 及其与 Z-80CPU .....     |     |
| 三、紫外线可擦除 ROM (EPROM) ...               | 134 | 接口 .....                      | 172 |
| 四、电擦除可编程 ROM (E <sup>2</sup> PROM) ... | 135 | 一、PIO 的基本结构 .....             | 172 |
| 第四节 存储器与 Z-80CPU .....                 |     | 二、PIO 与 Z-80CPU 的接口 .....     | 177 |
| 的接口 .....                              | 135 | 三、PIO 的编程方法 .....             | 178 |
| 一、ROM 与 Z-80CPU 的接口 .....              | 136 | 第二节 CTC 及其与 Z-80CPU .....     |     |
| 二、ROM 与 Z-80CPU 的接口 .....              | 142 | 接口 .....                      | 187 |
| 三、负载驱动 .....                           | 147 | 一、CTC 的主要特性 .....             | 187 |
| 第五节 存储器与 MCS-51 .....                  |     | 二、CTC 的结构 .....               | 188 |

|                                           |     |                                  |     |
|-------------------------------------------|-----|----------------------------------|-----|
| 三、CTC 的引脚说明 .....                         | 189 | 第三节 键盘接口 .....                   | 261 |
| 四、CTC 与 Z-80CPU 的接口 .....                 | 191 | 一、键盘与微机接口要点 .....                | 261 |
| 五、CTC 的编程方法 .....                         | 191 | 二、编码键盘和非编码键盘 .....               | 262 |
| 六、CTC 的编程举例 .....                         | 194 | 三、键盘的结构 .....                    | 262 |
| 七、MCS-51 内部 CTC 接口及<br>应用 .....           | 200 | 四、非编码键盘的接口 .....                 | 264 |
| 第三节 SIO 及其与 Z-80CPU<br>接口 .....           | 206 | 五、键盘/显示器接口芯片 Intel<br>8279 ..... | 277 |
| 一、串行通信的基本概念 .....                         | 206 | 第四节 打印机接口 .....                  | 283 |
| 二、SIO 的结构与引脚功能 .....                      | 209 | 一、点阵式打印机结构及基本工作<br>原理 .....      | 283 |
| 三、SIO 与 Z-80CPU 的接口 .....                 | 212 | 二、打印机的主要特性指标 .....               | 284 |
| 四、SIO 的控制字规约及其说明 .....                    | 212 | 三、打印机与微机的接口方式 .....              | 285 |
| 五、SIO 的中断方式 .....                         | 224 | 四、打印机接口举例 .....                  | 286 |
| 六、SIO 初始化程序设计举例 .....                     | 225 | 第五节 模拟量接口 .....                  | 289 |
| 第四节 8155 及其与 MCS-51<br>单片机接口 .....        | 229 | 一、D/A 转换器与微机的接口 .....            | 289 |
| 一、8155 的结构和引脚 .....                       | 230 | 二、A/D 转换器与微机的接口 .....            | 301 |
| 二、8155 的 RAM 和 I/O 口地址<br>编程 .....        | 231 | 思考与练习题 .....                     | 311 |
| 三、8155 的编程 .....                          | 231 | 第九章 微型计算机应用系统的<br>抗干扰设计 .....    | 313 |
| 四、8155 与 MCS-51 单片机的<br>接口与基本操作编程举例 ..... | 236 | 第一节 干扰的来源及种类 .....               | 313 |
| 第五节 8255 及其与 MCS-51<br>单片机的接口 .....       | 238 | 一、供电系统干扰 .....                   | 313 |
| 一、8255A 的结构 .....                         | 238 | 二、过程通道干扰 .....                   | 313 |
| 二、8255A 的引脚和控制 .....                      | 239 | 三、场干扰 .....                      | 313 |
| 三、8255A 与 MCS-51 单片机接口<br>及其编程方法 .....    | 244 | 四、印制板及电路间产生的相<br>互干扰 .....       | 313 |
| 思考与练习题 .....                              | 247 | 第二节 硬件抗干扰设计 .....                | 313 |
| 第八章 微型计算机与外部设备<br>接口技术 .....              | 249 | 一、供电系统的干扰及其抑制措施 .....            | 313 |
| 第一节 外设接口的基本问题 .....                       | 249 | 二、过程通道的干扰及其抑制<br>措施 .....        | 316 |
| 一、I/O 接口的基本任务 .....                       | 250 | 三、场干扰及其抑制措施 .....                | 320 |
| 二、I/O 接口的编址方式 .....                       | 250 | 四、印制电路板及电路的抗干扰<br>设计 .....       | 321 |
| 三、CPU 与外设之间的接口信号 .....                    | 251 | 第三节 软件抗干扰设计 .....                | 324 |
| 第二节 LED 显示器接口 .....                       | 252 | 一、干扰对系统形成的危害 .....               | 324 |
| 一、LED 显示器工作原理 .....                       | 252 | 二、CPU 中 PC 指针出错的处理<br>措施 .....   | 325 |
| 二、LED 显示器与微机接口要点 .....                    | 253 | 三、RAM 区数据的恢复措施 .....             | 327 |
| 三、LED 显示器接口举例 .....                       | 254 | 四、消除数据采集误差的措施 .....              | 327 |
|                                           |     | 五、防止控制状态混乱的措施 .....              | 329 |

## 第十章 微型计算机的典型应用..... 330

### 第一节 微型计算机应用系统的

设计方法及步骤..... 330

一、应用系统的总体方案设计 ..... 330

二、微型计算机应用系统的研制

步骤 ..... 331

### 第二节 微型计算机在工业顺

序控制中的应用..... 331

一、问题的提出和方案设计 ..... 331

二、接口电路 ..... 332

三、程序设计 ..... 332

四、进一步的简化 ..... 335

### 第三节 微型计算机在双坐标

控制系统中的应用..... 336

一、基本原理 ..... 336

二、系统组成 ..... 336

三、系统软件 ..... 338

### 第四节 单片机在电阻炉温度

控制中的应用..... 355

一、系统的组成 ..... 355

二、温度控制器的控制算式 ..... 356

三、过零检测电路及双向晶闸管

的触发 ..... 357

四、软件设计 ..... 357

## 附录 ..... 360

### 附录 A ASCII (美国标准信息

交换码) 表 ..... 360

### 附录 B Z-80 单板机指令功能表 ... 360

### 附录 C MCS-51 系列单片机

指令表 ..... 381

参考文献..... 387

# 第一章 微型计算机基础知识与基本结构

微型计算机具有体积小、功能强、价格低和使用方便的特点。近二十年来，发展十分迅速，显示出了强大的生命力，深入到各个技术领域。

本章首先介绍微型计算机的发展及应用，接着介绍微型计算机的基本组成与结构，最后介绍微型计算机的常用数制，使读者对微型计算机有一个基本的了解。

## 第一节 微型计算机的发展及应用

### 一、微型计算机的发展概况

微型计算机出现于 70 年代初期，是一种现代化电子设备，属于实现信息处理的自动化机器。

微型计算机自出现以后，发展极为迅速。就机种来说，据 1982 年不完全统计，已达 1100 多种，174 个系列。微型计算机所以得到这样迅速的发展是与大规模集成电路（LSI）或超大规模集成电路（VLSI）技术的迅速发展分不开的。

其实微型计算机在系统结构和基本工作原理方面，和其它计算机没有本质上的区别，都是按照匈牙利学者冯·诺依曼的原理设计制造的；所不同的是它广泛采用了集成度相当高的元器件、部件，尤其是它的核心部件——中央处理器 CPU（Central Processing Unit），采用了大规模或超大规模集成电路芯片，其体积只有以前其它电子计算机核心部件的千分之一或万分之一。微型计算机每次在技术上的新突破，都直接反映了大规模集成电路技术所取得的新进展。因此，在划分微型计算机技术的发展阶段时，往往就是以 LSI 技术的集中体现者——微处理机 MP 的新进展为主要标志的。微处理机芯片的集成度差不多每两年就有一次重大发展，也称换代。一般可以分为下列几个发展阶段：

**第一代：**1973 年以前为第一代。1971 年美国 Intel 公司为高级计算器设计了 I4004 芯片。经过改进，生产了第一代典型产品 Intel4004 和 I8008 微处理机。字长是 4~8 位（比特），平均指令周期约为 20 $\mu$ s，芯片的集成度约为 2000 个晶体管元件/片。

**第二代：**1973 年后进入第二代。典型产品有 I8080 和美国 Motorola 公司的 M6800；Signetics 2650 等。字长一般为 8 位，平均指令周期为 2 $\mu$ s，集成度已达 5000 个晶体管元件/片。开始用于控制和仪表中。

**第三代：**1976 年进入第三代。典型产品有 I8085 和美国 Zilog 公司的 Z80；MC6809 等。它们的字长均为 8 位，最短的指令周期已达 1 $\mu$ s，集成度已达到 10000 个晶体管元件/片。16KB 和 64KB 的存储器也已生产出来。在同一块集成电路（IC）芯片中，除中央处理器外，还包含有上述容量的存储器和部分接口电路，被人们称为单片微型计算机，可以在不附加其它电路的情况下，构成体积很小的微型计算机，使仪器仪表智能化。其典型产品有 Intel8748 和 MCS-48 系列，Texas 仪器公司的 TMS-1000 系列，还有 PPS-4/1 系列和 Z-8 等。

**第四代：**1978 年进入第四代。这个阶段为超大规模集成电路的微型计算机的发展阶段。微处理机的字长已达 16 位，指令周期小于  $0.5\mu\text{s}$ ，集成度已超过了 30000 个晶体管元件 / 片。它们在功能上已达到或接近中档小型机的水平，是微型计算机中的高档机。典型产品有 Intel8086、Intel8088、Z8000 和 M68000。IBM 公司利用 8088，80286CPU 研制成 IBM-PC / XT 及 IBM-PC / AT 个人计算机，使微型计算机应用深入到各个领域。

**第五代：**1981 年以后进入第五代。典型产品为 IAPX43201 和 80386；MC68010 和 MC68020 等 32 位高档机。这种高档微机已完全达到了传统的超级小型机水平，接近大、中型计算机的前期水平。在速度和集成度方面有更高的突破。

在超大规模集成电路 (VLSI) 支持下，目前微型计算机进入了极盛时期。为满足广泛应用的需要，微型计算机的发展形成了两个分支：一是不断扩大微处理器的功能，向以赶超小型机、超级小型机乃至大型计算机为目标的方向发展；另一个分支即是单片机的诞生和发展，以至形成了世界范围内的“单片机热”。进入 90 年代，微型计算机的发展方兴未艾，不仅出现了多功能、高速、高性能的新型微型计算机，而且在技术上逐步实现了硬件和软件产品的标准化、系列化、外部设备的多样化以及以微型计算机为主体的网络系统和多机系统。

## 二、微型计算机的应用

微型计算机的出现和发展开拓了计算机广泛应用和普及的新纪元。大到空间探索，小到揭示微观世界，从尖端科学技术到日常生活，几乎无所不包。下面仅例举在几个方面的某些典型应用事例。

1. 数值计算 这是微型计算机应用的一个基本方面，被广泛应用于科研和工程技术的计算。

2. 工业控制 目前用于逻辑顺序控制的可编程序控制器，有一半以上的主 CPU 是采用单片机（如东芝、三菱、立石、西门子等公司的多数产品和 A-B、歌德、西屋等公司的部分产品）；连续工业过程监控中用微型计算机实现的智能式变速、调节、执行、记录等单元仪表及 DDZ-S 系列仪表和 STD 总线工控机等；工业机器人控制；电机拖动控制；机床数控、数显；FMS、FMC、CIMS 中的控制单元等。

3. 智能仪器仪表 利用微型计算机的存储、数据处理、数值计算、逻辑判断、通信、人一机接口电路等功能，开发的具有自动校零、自动补偿、自动诊断乃至人工智能、专家系统等特点的仪器仪表。

4. 医疗仪器设备 如人工心脏定时控制系统、电脑中频电疗仪、神经电脉冲刺激仪，以及 B 超、CT、核磁共振等诊断仪。

5. 数据处理及计算机外围设备 高档便携式计算器、彩色及黑色复印机、智能图形终端、键盘管理、打印机、绘图仪、数字化仪、温式硬盘驱动器等均用单片机。

6. 航空、航天 如波音 747 客机的监控系统中就采用了 168 块不同类型的单片机。

7. 兵器工业 导弹制导、导航、火箭控制中的激光瞄准与跟踪、火力控制中的解算装置、反坦克导弹控制、鱼雷控制、反弹道航空炸弹、无线遥控可编程定时引信、通用数字航路仪等。

8. 通信设备 微型计算机可用于调制解调、无中心控制的多频道共用系统及手提式智能通信机、分布网络的节点控制器可编程标准通信接口等。

9. 汽车工业 如每辆进口高级轿车均装有 20~30 块单片机，以实现点火、变速、排

气、防滑等控制。

10. 民用产品 主要用于高级家用电器（冰箱、空调、彩电、录相机等）和高级电子玩具（游戏机、激光盘驱动器等）。

综上所述，微型计算机应用是非常广泛的。它已成为科学计算、工程设计、信息处理、生产控制、企业管理、遥控遥测、军事指挥及武器控制等各部门的必不可少的强有力的工具。它不仅能代人做某些体力劳动，节省时间、人力和物力，而且能够代替脑力劳动和某些职能，改善劳动条件，节约原料消耗，降低成本，提高产量、质量和劳动生产率。总之，凡是能归结为算术运算的计算或能严格规则化的工作都可由微型计算机来做。它是人类的一个得力工具。

## 第二节 微型计算机的基本组成和结构

### 一、微型计算机的基本组成

一个可供使用的微型计算机系统通常包含硬件系统和软件系统两大范畴。微型计算机的硬件系统指由电子线路、元器件和机械部分等构成的设备实体。主要包括中央处理器、存储器、输入/输出（I/O）接口电路及外部设备（键盘、显示器、打印机等）等。

一个微型计算机系统应具备哪些基本功能，包含哪些部件，这些部件按什么结构方式相互连接成有机的整体，各部件应具备何种功能，采用什么样的器件和电路构成，以及在工艺上如何进行组装等，都属于硬件设计的范畴。

图 1-1 是微型计算机的基本组成方框图。

微型计算机的软件系统相对硬件而言，泛指微型计算机系统所使用的各种程序和文件的集合。它们实际上是由一些算法和它们在微型计算机中的表示所构成，体现为一些触摸不到的二进制状态，所以称为软件。软件系统主要包括系统中配置的各种系统软件和为满足用户需要编制的各种应用软件以及编制程序所依赖的文件。其中系统软件包括用于管理微型计算机的操作系统、监控程序；程序设计语言及用来把高级语言翻译成机器语言的编译程序；用于诊断机器故障的诊断程序等。

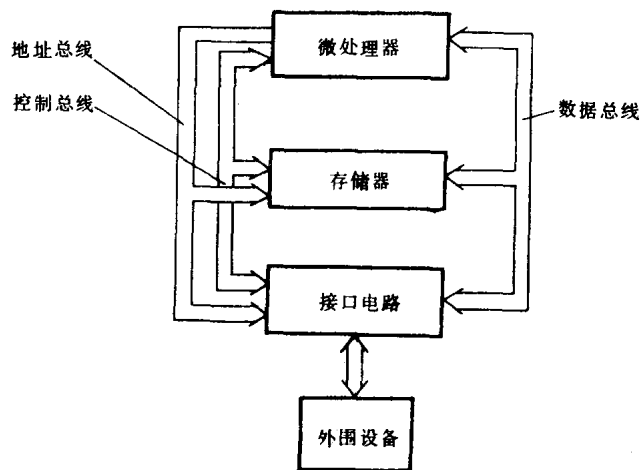


图 1-1 微型计算机的组成方框图

显然，硬件与软件二者是相互依存的，硬件是物质基础，没有良好的硬件支持就谈不上软件的执行。反之，没有完善良好的软件，微型计算机就不能高效率地工作。即软件是在硬件支持下工作的，而应用软件又是在系统软件支持下工作的。

## 二、微型计算机的基本结构

微型计算机由大规模集成电路组成，可以做成单板机、单片机，也可做成多板机。主要特点是体积小，重量轻，可靠性高，价格便宜，用途广。

微型计算机主要包括中央处理器、存储器和输入输出装置等几部分。其中核心部分是中央处理器。中央处理器从控制存储器取指令，完成算术运算逻辑操作和数据处理。寄存器用来暂时存放运算过程中产生的中间结果。电子时钟不断发出时钟脉冲，保证整个系统有节奏地进行工作。输入输出装置是微型计算机系统与外部世界交换信息的通路。

### (一) 中央处理器 (CPU)

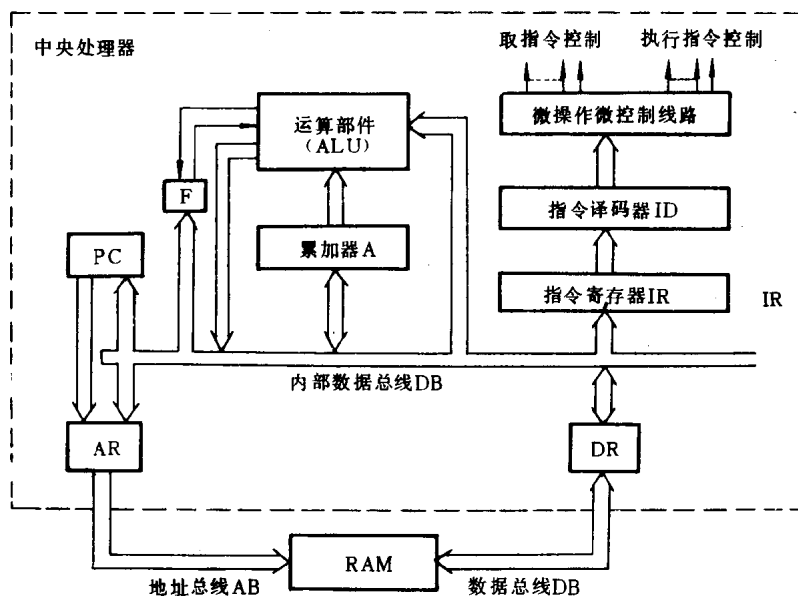


图 1-2 中央处理器的结构框图

图 1-2 为中央处理器的结构框图。中央处理器是微型计算机的核心部件，除主要进行各种算术和逻辑运算外，还负责控制和指挥其它各部件协调工作。

中央处理器中主要是算术逻辑运算部件（ALU）。参加运算的两个操作数，一个来自累加器 A，另一个来自数据寄存器 DR。运算结果通常又送到累加器 A 保存。

程序计数器 PC 用于控制程序执行的顺序，它存放的是指令地址。

地址寄存器 AR 用来存放指令地址或操作数地址。

在取指令时，将 PC 中存放的指令地址送 AR，从对应的存储器中取出指令。在取操作数时，可将操作数地址通过内部数据总线送到 AR，以便从存储器中取出操作数；如向存储器存入数据，也要给出地址送 AR 保存，以便存入数据。

指令寄存器 IR 用来存放从存储器取出的将要执行的指令。指令译码器 ID 用来翻译来自 IR 的指令，以便控制微操作控制线路发出各种控制信号，完成规定的操作。

状态标志寄存器 F，用来寄存算术和逻辑运算产生的状态标志，根据这些标志可决定程序的流向。

尽管目前各种 CPU 在性能指标方面有所差别，但一般都具备了以下几种基本功能：

- 1) 可进行算术或逻辑运算。
- 2) 具有接受或发送数据给存储器或外部设备的能力。
- 3) 可暂存少量数据。
- 4) 能对指令进行译码并执行指令所规定的操作。
- 5) 提供系统所需的定时和控制信号。
- 6) 可响应其它部件发出的中断请求。

## (二) 存储器

图 1-3 为存储器结构示意图。存储器是用来存放程序和各种数据（原始数据、中间结果和最终结果）信息的记忆装置。它由地址译码器、存储体和控制部分组成。目前可作为存储体的有磁心、半导体集成电路、磁表面记录介质等。地址译码器根据来自 CPU 的地址总线上的地址进行地址译码，以选择相应的存储单元，在读写控制电路作用下，可从存储器中读出信息，送到 8 位数据总线上；或经 8 位数据总线向存储器写入信息。

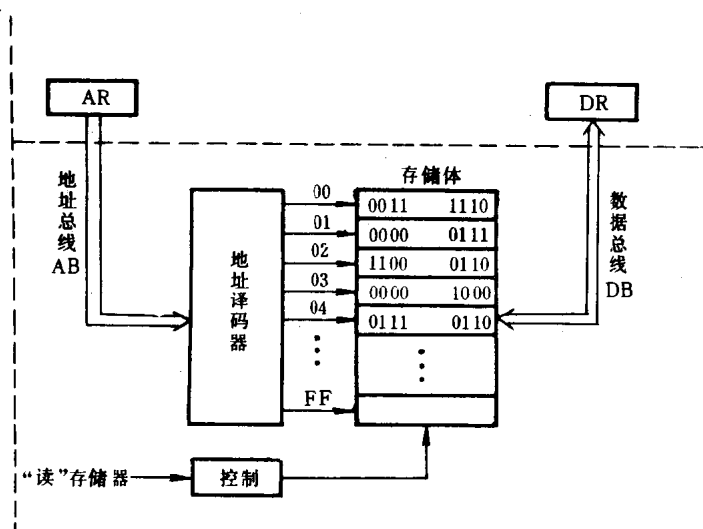


图 1-3 存储器结构示意图

## (三) I/O 接口

程序及原始数据通过输入设备（如穿孔信息输入设备；键盘信息输入设备；外存储器；模/数转换装置；图形、字符、语言信息的识别与输入装置等）转换成计算机能够识别的代码送到存储器中保存起来；计算结果或人们所需要的其它信息通过输出设备（如打印设备、绘图设备、CRT 显示器、外存储器、数/模转换装置等）传送出来。

在 CPU 与外部设备进行数据交换时，接口芯片起数据交换、缓冲和锁存作用。

## (四) 总线

微处理器的各个部件和外界的信息交换是通过总线进行的。总线是一组公共的信息传输线，它把微型计算机的 CPU、存储器、I/O 接口电路连接起来。微型计算机系统采用三总线结构，即数据总线 DB、地址总线 AB 和控制总线 CB。

数据总线 DB 用于各芯片之间传递数据，是双向的。Z80 使用 8 根数据线。

地址总线 AB 用于传送内存地址或 I/O 接口的地址。Z80 使用 16 根地址线。

控制总线 CB 用于传递 CPU 发出的各种控制信号，或者接收 I/O 接口等部件送来的状态信号、请求信号等。

微型计算机通过总线 CB 发出时序、读写信号，通过地址总线 AB，总线 DB 发出的数据来完成某一操作。

### 第三节 微型计算机的数制

#### 一、进位计数制

按进位的方法进行计数，称为进位计数制，十进计数制早已为大家所熟悉。1 个十进制数，它的数值是由 10 个不同的数码 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 来表示。并且是“逢十进一”。这样的若干个数码并在一起可以表示一个十进制的数。

在计算机中常用的是二进制，即以 2 为基数，逢 2 进位。用二进制的理由是：

1. 数的状态简单，容易表示 二进制只有 0 和 1 两个状态。从原则上讲，可以用任何两个对立的物理状态表示 0 和 1。例如用电位的“高”、“低”或脉冲的“有”、“无”表示 1 和 0；磁芯的某方向磁化表示 0，反方向磁化表示 1；纸带上无孔表示 0，有孔表示 1；指示灯不亮表示 0，亮表示 1 等等。这种简单的两种稳定状态会使计算机工作可靠，抗干扰能力强。

2. 算术运算简单 二进制四则运算特别简单。例如 1 位二进制数 A 和 B 相加和相乘各只有 4 种可能的组合，而 1 位十进制数的相加和相乘，却各有 100 种组合，计算机实现十进制这种逻辑要困难得多。

二进制运算规则简单，会使完成运算的电路简单、控制简化。

3. 采用二进制可以节省设备 例如用十进制数表示 00~99 之间的数。每位十进制数要具有 0~9 十个数码，需 10 个设备状态。两位数共需 20 个设备状态。

而用二进制表示 000~127（即 0000000~1111111）因每位只有两种数码，七位共需 14 个设备状态。由此看出，用二进制时可以节省设备。

4. 采用二进制，可以利用逻辑代数来综合 这就为计算机的逻辑设计提供了方便。

电子计算机内部使用二进制进行工作。但使用二进制需用较多的位数，书写很繁。为读写方便，常使用十六进制作为二进制的缩写方式。

十六进制使用 16 个符号，它们是 0~9, A, B, C, D, E, F；对于十六进制的加法计算是逢 16 进位。

而人们在日常生活中，习惯用十进制。为了避免使用多种进位制的混乱，可使用后缀表明数的进制：

后缀 B——表示二进制；

后缀 H——表示十六进制；

后缀 D——表示十进制（也可不加后缀）；

例如：十进制的 155 用不同的进制表示如下：

10011011B；

9BH；

155D；

## 二、各种进制数的转换

### (一) 二进制和十六进制数的相互转换

由于  $2^4 = 16^1$ , 所以十六进制数可用 4 位二进制数表示。

1. 十六进制转换为二进制 十六进制转换为二进制时, 只要把每 1 位十六进制的数用相应的 4 位二进制数代替就可以了。例如:

$$\begin{array}{cccc} \text{B} & 7 & \cdot & 5 & 4 \\ \downarrow & \downarrow & & \downarrow & \downarrow \\ 1011 & 0111 & & 0101 & 0100 \end{array}$$

$$\text{即 } B7 \cdot 54H = 10110111 \cdot 01010100B$$

2. 二进制转换为十六进制 这种转换可分两步进行。对整数部分, 从小数点向左数, 每 4 位二进制数一组, 最后不足四位的前面补 0; 对小数部分, 从小数点向右数, 每 4 位一组, 最后不足 4 位的后面补 0。然后把每 4 位二进制组用相应的 16 进制数代替, 即可转换为 16 进制数。例如:

$$\begin{array}{cccc} 1001 & 1011 & \cdot & 1010 & 0110 \\ \downarrow & \downarrow & & \downarrow & \downarrow \\ 9 & B & & A & 6 \end{array}$$

$$\text{即 } 1001 \ 1011 \cdot 1010 \ 0110 \ B = 9B \cdot A6H$$

### (二) 二进制、十六进制与十进制数的相互转换

1. 十六进制数转换为十进制数 因为任意一个十六进制数  $x$  都可用下式表示:

$$x = x_m \times 16^m + \cdots + x_0 \times 16^0 + x_{-1} \times 16^{-1} + \cdots + x_{-n} \times 16^{-n} = \sum_{i=-n}^m x_i \times 16^i$$

因此对给定的十六进制数, 只要按上述公式展开即可得到对应的十进制数。例如:

$$\begin{aligned} ABCDH &= 10 \times 16^3 + 11 \times 16^2 + 12 \times 16^1 + 13 \times 16^0 \\ &= 10 \times 4096 + 11 \times 256 + 12 \times 16 + 13 \\ &= 43981D \end{aligned}$$

微机中一个字节为八位二进制数, 若用十六进制表示, 其数的范围为 00H~FFH。而 FFH = 255

2. 二进制数转换为十进制数 同样, 对任意二进制数  $x$  都可用权展开为:

$$x = x_m \times 2^m + \cdots + x_0 \times 2^0 + x_{-1} \times 2^{-1} + \cdots + x_{-n} \times 2^{-n} = \sum_{i=-n}^m x_i \times 2^i$$

计算结果即为相应十进制数。这种转换是比较容易的。例如:

$$\begin{aligned} 1101 \times 1011B &= 1 \times 2^3 + 1 \times 2^2 + 1 \times 2^0 + 1 \times 2^{-1} + 1 \times 2^{-3} + 1 \times 2^{-4} \\ &= 13 \cdot 6875D \end{aligned}$$

3. 十进制整数转换为二进制数 把十进制整数转换为二进制数, 一般采用除 2 取余法, 即将已知的十进制数除以基数 2 并记下每次的余数, 一直用 2 去除到出现商为 0 为止。第一次的余数为低位, 最后一次余数为高位。例如:

$$215D = x_m \times 2^m + \cdots + x_0 \times 2^0$$

只要决定  $x_m, x_{m-1}, \cdots, x_1, x_0$  的值, 就可写出二进制数。因  $2^0 = 1$ , 所以  $(215 - x_0)$

一定是 2 的整数倍,  $215 \div 2$  所得的余数即为  $x_i$ , 其转换过程如下:

$$\begin{array}{rcl}
 2 & \overline{) 215} & \\
 2 & \overline{) 107} & \cdots \cdots \text{余数} = 1 = x_0 \\
 2 & \overline{) 53} & \cdots \cdots \text{余数} = 1 = x_1 \\
 2 & \overline{) 26} & \cdots \cdots \text{余数} = 1 = x_2 \\
 2 & \overline{) 13} & \cdots \cdots \text{余数} = 0 = x_3 \\
 2 & \overline{) 6} & \cdots \cdots \text{余数} = 1 = x_4 \\
 2 & \overline{) 3} & \cdots \cdots \text{余数} = 0 = x_5 \\
 2 & \overline{) 1} & \cdots \cdots \text{余数} = 1 = x_6 \\
 & 0 & \cdots \cdots \text{余数} = 1 = x_7; \text{商为 } 0, \text{ 转换结束。}
 \end{array}$$

用除 2 取余得  $215D = 11010111B$

4. 十进制整数转换为十六进制数 把十进制整数转换为十六进制数, 同转换为二进制数的道理一样, 可采用除 16 取余法。

例如:  $12345D = 3039H$

5. 十进制纯小数转化成二进制小数 转化时, 采用乘 2 取整法。即将十进制纯小数反复乘基数 2, 每次所得的整数部分即为所求的  $x_i$ , 第一次为高位, 最后一次为低位。二进制小数的位数根据精度要求或设备的情况而定。例如: 把 0.625 转化成二进制数:

$$\frac{2 \times 0.625}{1.25} \text{ 整数部分} = 1 = x_{-1},$$

$$\frac{2 \times 0.25}{0.50} \text{ 取小数部分为新被乘数; 整数部分} = 0 = x_{-2},$$

$$\frac{2 \times 0.50}{1.00} \text{ 取小数部分为新被乘数; 整数部分} = 1 = x_{-3},$$

0.00 小数部分为零, 转换结束。

故  $0.625 = 0.101B$

含有整数和小数部分的十进制数转换为二进制数, 可分别把整数部分和小数部分转换成二进制, 然后合并就可将 1 个任意十进制数转换成二进制数。例如:

$$215.625 = 11010111.101B$$

### 三、带符号数的表示

二进制数可能为正, 也可能为负。在计算机中, 符号也数码化了, 通常一个有符号数的最高位为符号位。这就是数在计算机中的“代码表示”。所谓带符号数的“代码表示”是指带符号数的数值部分和符号部分统一用代码形式 (0 或 1) 表示。一般用 0 表示“+”号, 1 表示“-”号。

既然一个数的数值与符号全部数码化了, 符号位与数值位都是数码, 那么在对数据进行运算操作时, 符号位应如何处理呢? 是否也同数值位一道参加运算操作? 为了妥善解决这些问题, 就产生了将符号位与数值位一起编码的各种方法。这就是原码、补码及反码表示法。

#### (一) 原码

带符号二进制数的原码, 表示方法很简单。如上所述, 正数的符号位用 0 表示, 负数的

符号位用 1 表示, 符号位之后表示数值的大小。

如  $x = +114$ ,  $[x]_{\text{原}} = 01110010\text{B}$

$x = -114$ ,  $[x]_{\text{原}} = 11110010\text{B}$

原码表示法的缺点是原码的“0”有两种:

$[+0]_{\text{原}} = 00000000\text{B}$

$[-0]_{\text{原}} = 10000000\text{B}$

另外, 当两个异号数相加, 就要做减法, 为了把减法运算改为加法运算, 引出了反码和补码。

## (二) 反码

正数的反码与原码相同, 最高位一定为 0, 代表符号。其余位为数值位。

负数的反码其符号位为 1, 与原码相同; 数值位则将其负数的原码的数值位换位求反。

如  $x = -4$ ,  $[x]_{\text{反}} = 11111011\text{B}$

$x = -0$ ,  $[x]_{\text{反}} = 11111111\text{B}$

$x = -127$ ,  $[x]_{\text{反}} = 10000000\text{B}$

所以, 反码中“0”仍有两种表示方法。

## (三) 补码

正数的补码表示与原码相同, 即最高位为符号位, 用“0”表示正, 其余位为数值位。而负数的补码为其反码, 且在最低位加 1 形成。如

$x = -4$ ,  $[x]_{\text{补}} = [x]_{\text{反}} + 1 = 11111100\text{B}$

$x = -0$ ,  $[x]_{\text{补}} = [x]_{\text{反}} + 1 = 00000000\text{B}$

八位二进制补码有以下特点:

1)  $[+0]_{\text{补}} = [-0]_{\text{补}} = 00000000\text{B}$ 。

2) 8 位二进制补码所能表示的数值为  $+127 \sim -128$ 。

3) 当求补码的真值时, 如最高位为 0 时, 其余 7 位即为此数的二进制值。但当最高位为“1”时 (即负数), 需把其余 7 位求反以后最低位加 1, 才是它的二进制值。例如:

$[x]_{\text{补}} = 10010100$

$x = -(1101011+1) = -1101100 = -108\text{D}$

4) 当负数采用补码时, 就可把减法转换为加法。例如:  $x = 24 - 10 = 14$

也可用  $[x]_{\text{补}} = [24]_{\text{补}} + [-10]_{\text{补}}$

即  $00011000$

$+11110110$

$\boxed{1} 00001110$

自然丢失

故  $[x]_{\text{补}} = 00001110$ ,  $x = 14$ , 同减法结果一致。

在日常生活中, 有不少补数的例子。如手表的校对时间, 可顺时针拨或逆时针拨, 这两种校准钟表的方法是等效的, 这里的顺拨与倒拨对模 12 (Mod12) 互为补数。也就是可以将减法化为加法。

采用 8 位二进制时, 其模为 100H, 因此 -1 的补码是  $(100\text{H} - 01\text{H}) = \text{FFH}$ , -10 的补码是  $(100\text{H} - 0\text{AH}) = \text{F6H}$ , 依次类推 -128 的补码是  $(100\text{H} - 80\text{H}) = 80\text{H}$ 。