

普通高等院校“十二五”规划教材

单片机原理、接口 与C51应用程序设计

DANPIANJI YUANLI JIEKOU

YU C51 YINGYONG CHENGXU SHEJI

主 编 张先庭
副主编 向 瑛 王 忠 周传璘



国防工业出版社

National Defense Industry Press

普通高等院校“十二五”规划教材

单片机原理、接口 与 C51 应用程序设计

主编 张先庭
副主编 向 瑛 王 忠 周传璘

国防工业出版社

· 北京 ·

内 容 简 介

本书以 MCS-51 单片机为对象,分原理、扩展、接口和应用 4 个层次,讲述了单片机原理和应用技术。全书 12 章,内容包括单片机的基础知识,MCS-51 系列单片机结构,指令系统,汇编语言程序设计,中断系统,内部定时/计数器及串行接口,单片机系统扩展和 SPI、I²C、串行单总线等串行扩展技术,显示、键盘、A/D、D/A 等应用接口技术,函数及 C51 程序设计、系统抗干扰和应用系统设计等。本书内容注重新颖性和工程实用性,力求反映单片机应用领域的最新发展和培养读者的实际应用能力。

本书可作为高等院校电子信息各专业单片机课程教材,也可供单片机爱好者自学和工程技术人员参考。

图书在版编目(CIP)数据

单片机原理、接口与 C51 应用程序设计 / 张先庭主编.

—北京:国防工业出版社,2011 1

普通高等院校“十二五”规划教材

ISBN 978-7-118-07275-4

I ①单... II. ①张... III. ①单片微型计算机-理论-高等学校-教材②单片微型计算机-接口-高等学校-教材③C 语言-程序设计-高等学校-教材 IV.

①TP368 1②TP312

中国版本图书馆 CIP 数据核字(2011)第 006372 号

※

国防工业出版社出版发行

(北京市海淀区紫竹院南路 23 号 邮政编码 100048)

北京奥鑫印刷厂印刷

新华书店经售

*

开本 787×1092 1/16 印张 18½ 字数 421 千字

2011 年 1 月第 1 版第 1 次印刷 印数 1—4000 册 定价 34.00 元

(本书如有印装错误,我社负责调换)

国防书店: (010)68428422

发行邮购: (010)68414474

发行传真: (010)68411535

发行业务: (010)68472764

前 言

单片机是在一块集成电路芯片上集成了 CPU、存储器、I/O 接口等各种功能部件的单片微型计算机,具有集成度高、功能强、可靠性好、性价比高等优点。单片机广泛应用于工业控制、数据采集、智能化仪表、办公自动化以及家用电器等各个领域。在众多的单片机中,MCS-51 系列单片机以其优越的性能、成熟的技术和较高的可靠性,占领了工业控制领域的主要市场。经过了 30 多年的发展,MCS-51 系列单片机已形成了品种多、功能全、用户群庞大的系列产品,成为我国单片机应用领域的主流和高校最为流行的单片机教学机型之一。

本书以 MCS-51 系列为核心,系统介绍了 MCS-51 系列单片机的基本原理、接口技术、汇编和 C51 软件编程知识,其主要特点是:

(1) 注重基础性、层次性和系统性。全书分为原理、扩展、接口、应用 4 个层次,分别讲述了单片机的组成原理、中断系统、定时器和串行通信等功能部件、汇编语言编程;MCS-51 系统存储器和 I/O 并行扩展的原理和方法,可编程并行 I/O 接口扩展原理;包括显示、键盘在内的人机接口硬件设计和软件编程,A/D、D/A 接口的原理和接口电路设计。在此基础上,介绍了 C51 的语法规则和 C51 编程方法,单片机系统设计的原则与方法,以及系统设计中可靠性和抗干扰处理措施,最后用一个工程实例介绍了单片机应用系统的设计过程。

(2) 注重工程实用性,力求培养读者的实际应用能力。本书在汇编语言程序设计中,通过大量例子介绍了单片机应用系统中常用的软件滤波方法和程序设计,数据的排序和查找程序设计,线性补偿查表程序设计以及人机接口中常用的数制转换程序设计;在系统设计中系统讲述了工业测控系统干扰的来源以及相应的软硬件抗干扰处理措施。

(3) 注重新颖性,力求反映单片机应用领域的新技术和新方法。本书介绍了单片机领域广泛应用的 SPI 总线、I²C 总线和串行单总线技术;LED 点阵屏的工作原理和软硬件设计方法;常见的 LCD 字符点阵片原理和接口编程方法;此外还介绍了 C51 硬件接口编程的方式方法。

(4) 由浅入深,重点突出,难点分散。在介绍单片机组成、原理和汇编程序设计的基础上,分别讲述单片机外围扩展和接口方法,然后介绍单片机应用系统设计。在例题、接口电路等的选择上,尽量采用先易后难的原则,考虑与实际工程相结合并插

入大量电路连接图、结构图、时序图和分析说明。

本书由张先庭主编,向瑛、王忠周、周传磷副主编。第1章、第3章由向瑛编写,第6章、第7章、第8章由王忠编写,第2章、第11章由周传磷编写,其余章节由张先庭编写。在本书编写的过程中,得到了吴开志、陈黎娟、熊德鹏、吴国辉、方庭等同事和南昌航空大学的支持和帮助,编写过程中参考了大量的教程和文献,在此一并致以衷心的感谢。

由于时间仓促和编者水平有限,加之单片机技术发展迅速,书中难免存在错误和不足之处,敬请广大读者批评指正。

编 者
2010 年 9 月

目 录

第1章 单片机的基础知识	1
1.1 计算机中数据的表示方法	1
1.1.1 带符号数的表示方法	1
1.1.2 字符在计算机内的表示	5
1.2 微型计算机的基本结构	5
1.3 微处理器的组成及功能	7
1.4 单片机的概念及其特点	9
1.5 典型的单片机产品	10
1.6 单片机的应用	10
1.7 单片机的发展趋势	11
习题与思考	12
第2章 MCS-51 系列单片机结构	13
2.1 MCS-51 系列单片机结构与引脚	13
2.1.1 引脚及功能说明	13
2.1.2 内部结构及功能部件	15
2.2 中央处理器 CPU	16
2.3 单片机的时钟与时序	18
2.3.1 时钟电路	18
2.3.2 CPU 时序	18
2.4 MCS-51 单片机存储器及存储空间	20
2.4.1 程序存储器	21
2.4.2 内部数据存储器	22
2.4.3 外部数据存储器	25
2.5 MCS-51 单片机并行 I/O 口	26
2.6 单片机复位和复位电路	29
2.6.1 单片机复位功能	29
2.6.2 复位电路	30
2.7 MCS-51 系列单片机的工作方式	30
2.7.1 程序执行方式	30
2.7.2 掉电和节电方式	30

2.7.3 EPROM 编程和校验方式	31
习题与思考	32
第3章 指令系统	33
3.1 寻址方式	33
3.2 指令系统常用符号	35
3.3 MCS-51 单片机的指令系统	35
3.3.1 数据传送类指令	35
3.3.2 算术运算类指令	41
3.3.3 逻辑运算类指令	45
3.3.4 控制转移类指令	46
3.3.5 布尔操作类指令	50
习题与思考	51
第4章 汇编语言程序设计	55
4.1 MCS-51 单片机汇编语言的伪指令	55
4.2 程序流程图和程序结构	57
4.2.1 顺序结构程序设计	58
4.2.2 分支结构程序设计	59
4.2.3 循环结构程序设计	60
4.2.4 子程序结构程序设计	64
4.3 常用程序设计举例	66
4.3.1 查表程序设计	66
4.3.2 排序和检索程序设计	68
4.3.3 运算程序设计	71
4.3.4 数制转换程序设计	74
4.3.5 滤波程序设计	76
习题与思考	80
第5章 中断系统	81
5.1 中断的概念	81
5.2 MCS-51 中断系统	82
5.2.1 MCS-51 中断源	82
5.2.2 中断控制	83
5.2.3 中断处理	85
5.3 中断应用程序举例	87
习题与思考	90

第6章 内部定时器/计数器及串行接口	91
6.1 定时器/计数器	91
6.1.1 定时器/计数器结构与功能	91
6.1.2 定时器/计数器相关寄存器	92
6.1.3 定时器/计数器工作模式	93
6.1.4 定时器/计数器应用	95
6.1.5 定时器/计数器应用的其他问题	98
6.1.6 MCS-51 定时器/计数器 2 的工作方式	100
6.2 MCS-51 串行接口	103
6.2.1 串行通信概念	104
6.2.2 MCS-51 串行接口	106
6.2.3 串行口的工作方式	108
6.2.4 T2 作波特率发生器	110
6.2.5 MCS-51 多机通信原理	111
6.2.6 串行通信总线标准及接口	112
6.2.7 串行通信的应用举例	115
习题与思考	118
第7章 单片机系统扩展	120
7.1 单片机系统总线扩展原理	120
7.2 存储器扩展技术	121
7.2.1 总线扩展的地址译码方法	122
7.2.2 程序存储器的扩展	124
7.2.3 数据存储器的扩展	129
7.2.4 存储器综合扩展技术	130
7.3 MCS-51 系列单片机 I/O 口扩展	131
7.3.1 并行 I/O 口的简单扩展	131
7.3.2 采用可编程并行 I/O 接口芯片 8255A 扩展	132
7.3.3 采用可编程并行 I/O 接口芯片 8155 扩展	140
7.4 单片机 I/O 端口模拟时序扩展设备	145
7.4.1 SPI 串行接口总线技术	146
7.4.2 I ² C 串行接口总线技术	147
7.4.3 串行单总线技术	151
习题与思考	154
第8章 单片机应用接口技术	156
8.1 显示器接口	156

8.1.1	LED 显示器接口	156
8.1.2	点阵式发光显示屏的接口技术	162
8.1.3	LCD 显示器接口	165
8.2	键盘接口技术	170
8.2.1	独立式键盘接口设计	171
8.2.2	矩阵式键盘接口设计	172
8.3	A/D 转换器接口	177
8.3.1	A/D 转换器概述	177
8.3.2	A/D 转换器芯片 ADC0809 与单片机接口	177
8.3.3	AD574 与单片机接口	181
8.4	D/A 转换器接口	184
8.4.1	D/A 转换器概述	184
8.4.2	8 位 D/A 转换器 DAC0832 与单片机接口	184
	习题与思考	188
第 9 章	C51 程序基础	189
9.1	C51 简介	189
9.2	C51 程序结构	189
9.3	C51 数据类型	191
9.4	常量与变量	192
9.4.1	常量	192
9.4.2	变量	192
9.5	运算符与表达式	195
9.5.1	赋值运算	196
9.5.2	算术运算	196
9.5.3	关系运算	197
9.5.4	逻辑运算	197
9.5.5	位运算	198
9.5.6	自增减运算及复合运算	198
9.5.7	逗号表达式	199
9.6	C51 程序结构	199
9.6.1	分支结构	199
9.6.2	循环结构	200
9.7	数组、结构体、联合体	201
9.7.1	数组	201
9.7.2	结构体	203
9.7.3	联合体	206
9.8	指针	206

9.8.1 指针的定义与引用	206
9.8.2 指针和数组	207
9.8.3 指针和结构体、联合体	209
习题与思考	209
第 10 章 函数及 C51 程序设计	211
10.1 函数的定义	211
10.2 函数的调用	212
10.3 变量的作用域与存储方式	212
10.4 中断函数	213
10.5 C51 库函数和头文件	214
10.6 C51 程序举例	215
10.7 C51 与汇编混合编程	231
10.7.1 模块内接口	231
10.7.2 模块间接口	232
10.8 C51 的启动文件	237
习题与思考	240
第 11 章 单片机应用系统设计技术	242
11.1 单片机应用系统的基本结构	242
11.2 单片机应用系统的设计原则	243
11.3 单片机应用系统的设计过程	244
11.3.1 总体设计	244
11.3.2 硬件设计	244
11.3.3 软件设计	247
11.3.4 系统调试	248
11.4 可靠性设计	249
11.4.1 干扰的来源	249
11.4.2 硬件抗干扰技术	249
11.4.3 软件抗干扰技术	253
习题与思考	255
第 12 章 柴油发电机组测控系统设计举例	256
12.1 系统基本原理	256
12.2 主要技术指标要求	256
12.3 系统总体设计	257
12.4 系统的硬件设计	262

12.5 软件设计	264
附录1 MCS-51 指令系统	280
附录2 常用字符的 ASCII 码(用十六进制数表示)	283
参考文献	284

第 1 章 单片机的基础知识

1.1 计算机中数据的表示方法

在计算机中,能直接表示和使用的有数值数据和符号数据两大类。数值数据用来表示数值的大小,并且还带有表示数值正负的符号位。符号数据又称非数值数据,用来表示一些符号标记,包括英文大小字母、数字符号 0~9、汉字和图像信息等。由于计算机中的数据都采用二进制编码形式,因此,讨论数据的表示方法就是讨论它们在计算机中的组成格式和编码规则。

1.1.1 带符号数的表示方法

在计算机中,数值有大小,也有正负,用什么方法表示数值的符号呢?通常用一个数的最高位表示符号位,若字长为 8 位,则 D7 为符号位,D6~D0 为数值位。符号位用 0 表示正数,用 1 表示负数。例如:

$$X = (01011011)_2 = +91$$

$$X = (11011011)_2 = -91$$

这种连同同一个符号位在一起的数称为机器数,它的数值称为机器数的真值。机器数的表示如图 1-1 所示。

为了运算方便,机器数在计算机中有 3 种表示法:原码、反码和补码。

原码、反码和补码都是带符号数在机器中的表示方法。在介绍这 3 种编码方法之前,先介绍模的概念和性质。

把一个计量器的容量称为模或模数,记为 M 或 $\text{mod } M$ 。例如:一个 n 位二进制计数器,它的容量是 2^n ,所以它的模为 2^n (即可表示 2^n 个不同的数)。又如:时钟可表示 12 个钟点,它的模是 12。

模具有这样的性质,当模为 2^n 时, 2^n 和 0 的表现形式是相同的。例如:一个 n 位的二进制计数器,可以从 0 计数到 2^{n-1} ,如果再加 1,计数器就会变为 0。同样,时钟的 0 点和 12 点在钟表上的表现形式也是相同的。

1. 原码

用原码表示时,最高位为符号位,正数用 0 表示,负数用 1 表示,其余的位用于表示数的绝对值。原码的表示如图 1-2 所示。

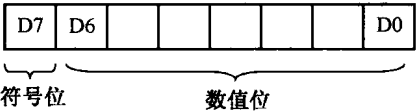


图 1-1 机器数的表示

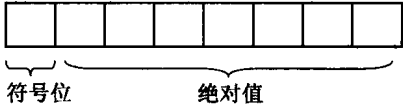


图 1-2 原码的表示

对于一个 n 位的二进制数,其原码定义为

$$[X]_{\text{原}} = \begin{cases} 2^n + X, & 0 \leq X < 2^{n-1} \\ 2^{n-1} + X, & -2^{n-1} < X \leq 0 \end{cases}$$

式中: X 为真值的 $(n-1)$ 位绝对值; n 为机器可表示的二进制位数。

【例 1.1】求 $+67$ 、 -25 的原码(机器字长 8 位)。

解: 因为

$$|+67| = 67 = 01000011\text{B}$$

$$|-25| = 25 = 00011001\text{B}$$

所以

$$[+67]_{\text{原}} = 01000011\text{B}$$

$$[-25]_{\text{原}} = 10011001\text{B}$$

用原码表示时,对于 -0 和 $+0$ 的编码不一样。假设机器字长为 8 位,则

$$[-0]_{\text{原}} = 10000000$$

$$[+0]_{\text{原}} = 00000000$$

2. 反码

用反码表示时,最高位为符号位,正数用 0 表示,负数用 1 表示。正数的反码与原码相同,而负数的反码可在原码的基础之上,符号位不变,其余位取反得到。反码数的表示范围与原码相同,对于一个 n 位的二进制,其反码的定义为

$$[X]_{\text{反}} = \begin{cases} 2^n + X, & 0 \leq X < 2^{n-1} \\ (2^{n-1} - 1) + X, & -2^{n-1} < X \leq 0 \end{cases}$$

从定义可以看出, X 为正数时, $[X]_{\text{反}}$ 与 X 的差别只是用 0 代替符号位, X 为负数时,用 1 表示符号位,其他各位取反。

【例 1.2】求 $+67$ 、 -25 的反码(机器字长 8 位)。

解: 因为

$$[+67]_{\text{原}} = 01000011\text{B}$$

$$[-25]_{\text{原}} = 10011001\text{B}$$

所以

$$[+67]_{\text{反}} = 01000011\text{B}$$

$$[-25]_{\text{反}} = 11100110\text{B}$$

用反码表示时,对于 -0 和 $+0$ 的编码也不一样。假设机器字长为 8 位,则

$$[-0]_{\text{反}} = 11111111$$

$$[+0]_{\text{反}} = 00000000$$

3. 补码

用补码表示时,最高位为符号位,正数用 0 表示,负数用 1 表示。正数的补码与原码相同,而负数的补码可在原码的基础之上,符号位不变,其余位取反,末位加 1 得到。对于

一个负数 X , 其补码也可用 $2^n - |X|$ 得到, 其中 n 为计算机字长。

【例 1.3】求 $+67$ 、 -25 的补码(机器字长 8 位)。

解: 因为

$$[+67]_{\text{原}} = 01000011\text{B}$$

$$[-25]_{\text{原}} = 10011001\text{B}$$

所以

$$[+67]_{\text{补}} = 01000011\text{B}$$

$$[-25]_{\text{补}} = 11100111\text{B}$$

另外, 对于计算补码, 也可用一种求补运算方法求得。

求补运算: 一个二进制数, 符号位和数值位一起取反, 末位加 1。

求补运算具有以下特点:

对于一个数 X , 有

$$\bullet [X]_{\text{补}} \xrightarrow{\text{求补}} [-X]_{\text{补}} \xrightarrow{\text{求补}} [X]_{\text{补}}$$

那么, 已知正数的补码, 则可通过求补运算求得对应负数的补码, 已知负数的补码, 相应也可通过求补运算求得对应正数的补码, 也就是说, 在用补码表示时, 求补运算可得到数的相反数。

【例 1.4】已知 $+25$ 的补码为 00011001B , 用求补运算求 -25 的补码。

解: 因为

$$[25]_{\text{补}} \xrightarrow{\text{求补}} [-25]_{\text{补}}$$

所以

$$[-25]_{\text{补}} = 11100110 + 1 = 11100111\text{B}$$

对于一个 n 位的二进制数, 用补码表示时, 对于 -0 和 $+0$, 其补码是相同的, 假设机器字长为 8 位, 则

$$[+0]_{\text{补}} = 00000000$$

$$[-0]_{\text{补}} = 00000000$$

4. 补码的加减运算

在现在的计算机中, 有符号数的表示都用补码表示, 用补码表示时运算简单。补码的加减法运算规则为

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

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

即求两个数之和的补码, 直接用两个数的补码相加; 求两个数之差的补码, 用被减数的补码加减数的相反数的补码($[-Y]_{\text{补}}$)。

【例 1.5】假设计算机字长为 8 位, 完成下列补码运算。

(1) $(+25) + (+32)$

解: 因为

$$[+25]_{\text{补}} = 00011001\text{B} \quad [+32]_{\text{补}} = 00100000\text{B}$$

$$\begin{array}{r} [+25]_{\text{补}} = 00011001 \\ + [+32]_{\text{补}} = 00100000 \\ \hline 00111001 \end{array}$$

所以

$$[(+25) + (+32)]_{\text{补}} = [+25]_{\text{补}} + [+32]_{\text{补}} = 00111001\text{B} = [+57]_{\text{补}}$$

$$(2) (+25) + (-32)$$

解: 因为

$$[+25]_{\text{补}} = 00011001\text{B} \quad [-32]_{\text{补}} = 11100000\text{B}$$

$$\begin{array}{r} [+25]_{\text{补}} = 00011001 \\ + [-32]_{\text{补}} = 11100000 \\ \hline 11111001 \end{array}$$

所以

$$[(+25) + (-32)]_{\text{补}} = [+25]_{\text{补}} + [-32]_{\text{补}} = 11111001\text{B} = [-7]_{\text{补}}$$

$$(3) (+25) - (+32)$$

解: 因为

$$[+25]_{\text{补}} = 00011001\text{B} \quad [+32]_{\text{补}} = 00100000\text{B}$$

$$[-32]_{\text{补}} = \{ [+32]_{\text{补}} \}_{\text{求补}} = 11100000\text{B}$$

$$\begin{array}{r} [+25]_{\text{补}} = 00011001 \\ + [-32]_{\text{补}} = 11100000 \\ \hline 11111001 \end{array}$$

所以

$$[(+25) - (+32)]_{\text{补}} = [+25]_{\text{补}} + \{ [+32]_{\text{补}} \}_{\text{求补}} = 11111001\text{B} = [-7]_{\text{补}}$$

$$(4) (+25) - (-32)$$

解: 因为

$$[+25]_{\text{补}} = 00011001\text{B} \quad [-32]_{\text{补}} = 11100000\text{B}$$

$$[+32]_{\text{补}} = \{ [-32]_{\text{补}} \}_{\text{求补}} = 00100000\text{B}$$

$$\begin{array}{r} [+25]_{\text{补}} = 00011001 \\ + [+32]_{\text{补}} = 00100000 \\ \hline 00111001 \end{array}$$

所以

$$[(+25) - (-32)]_{\text{补}} = [+25]_{\text{补}} + \{ [-32]_{\text{补}} \}_{\text{求补}} = 00111001\text{B} = [+57]_{\text{补}}$$

从以上可以看出,通过补码进行加减运算非常方便,而且能把减法转换成加法,得到正确的结果。

5. 十进制数的表示

计算机内部对信息是按二进制方式进行处理,但人们生活中习惯使用十进制。为

了处理方便,在计算机中,对于十进制数也提供了十进制编码形式。

十进制编码又称为 BCD 码,分为压缩 BCD 码和非压缩 BCD 码。压缩 BCD 码又称为 8421 码,它用 4 位二进制编码来表示 1 位十进制符号。十进制数符号有 10 个: 0~9, 编码情况如表 1-1 所列。

表 1-1 压缩 BCD 编码表

十进制符号	压缩 BCD 编码	十进制符号	压缩 BCD 编码
0	0000	5	0101
1	0001	6	0110
2	0010	7	0111
3	0011	8	1000
4	0100	9	1001

用压缩 BCD 码表示十进制数,只要把每个十进制符号用对应的 4 位二进制编码代替即可。例如,十进制数 124 的压缩 BCD 码为 0001 0010 0100。十进制数 456 的压缩 BCD 码为 0100 0101 0110。

非压缩 BCD 码是用 8 位二进制编码来表示 1 位十进制符号,其中低 4 位二进制编码与压缩 BCD 码相同,高 4 位任取。例如,下面介绍的数字符号的 ASCII 码就是一种非压缩的 BCD 码。用非压缩 BCD 码表示十进制数,1 位十进制符号须用 8 位二进制数表示。例如,十进制数 124 的非压缩 BCD 码为 0011 0001 0011 0010 0011 0100。

1.1.2 字符在计算机内的表示

在计算机信息处理中,除了处理数值数据外,还涉及大量的字符数据。例如,从键盘上输入的信息或打印输出的信息都是以字符方式输入/输出的,字符数据包括字母、数字、专用字符及一些控制字符等,这些字符在计算机中也是用二进制编码表示的。现在的计算机中字符数据的编码通常采用的是美国信息交换标准代码(American Standard Code for Information Interchange, ASCII)。基本 ASCII 码标准定义了 128 个字符,用 7 位二进制来编码,包括英文 26 个大写字母、26 个小写字母、10 个数字符号 0~9,还有一些专用符号(如“:”、“!”、“%”)及控制符号(如换行、换页、回车等)。常用字符的 ASCII 码见附录 1。

计算机中一般以 8 位二进制表示 1 个字节,字符 ASCII 码通常放于低 7 位,高位一般补 0,在通信时,最高位常用作奇偶校验位。

1.2 微型计算机的基本结构

微型计算机主要由微处理器、存储器、I/O 接口和 I/O 设备组成。各组成部分之间通过系统总线联系在一起(图 1-3)。

1. 微处理器

微处理器 CPU 是计算机的核心部件,它的性能在很大程度上决定了计算机的性能。

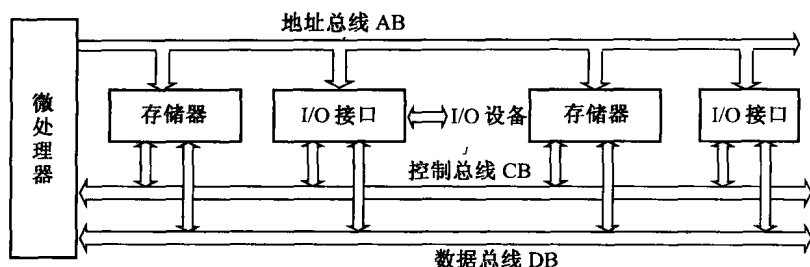


图 1-3 微型计算机结构

2. 系统总线 (System Bus)

总线 (Bus) 就是将多个装置或部件连接起来并传送信息的公共通道。总线实际上是一组传输信号的线路。系统总线一般分为 3 种类型, 即地址总线、数据总线和控制总线, 有时也称为三大总线。

1) 地址总线 (Address Bus, AB)

地址总线主要用来传输 CPU 发出的地址信息, 选择需要访问的存储单元和 I/O 接口电路。地址总线是单向的, 只能由 CPU 向外传送地址信息。地址总线的位数决定了可以直接访问的存储器的单元数目。

2) 数据总线 (Data Bus, DB)

数据总线用来在微处理器和存储器以及输入/输出 (I/O) 接口之间传送数据, 如从存储器中取数据到 CPU, 把运算结果从 CPU 送到外部输出设备等。数据总线是双向的, 即数据可从 CPU 传出, 也可以从外部送入 CPU, 微处理器的位数和外部数据总线的位数应该一致。

3) 控制总线 (Control Bus, CB)

控制总线可以是 CPU 的控制信号或状态信号, 也可以是外部设备的请求信号或联络信号。对于每一条具体的控制线, 信号的传送方向是固定的, 个别信号线还兼有双向功能。

系统总线是传送信息的通道, 非常繁忙, 其使用特点如下:

(1) 在某一时刻, 只能由一个总线主控设备控制总线, 其他总线主控设备必须放弃总线的控制权。

(2) 在连接系统总线的各个设备中, 同时只能有一个发送者向总线发送信号, 但可以多个设备同时从总线上获取信号。

3. 存储器 (Memory)

存储器就是存放程序和数据的部件。有了存储器, 计算机才能进行程序的运行和数据的处理。微型计算机上的存储器分为“主存”和“辅存”两类, 当前它们主要由半导体存储器和磁盘、光盘存储器等分别构成。

半导体存储器造价高、速度快, 但容量小, 主要用来存放当前正在运行的程序和正在等待处理的数据。它分为只读存储器 (Read Only Memory, ROM) 和随机存取存储器 (Random Access Memory, RAM) 两种。ROM 只允许读操作, 即在正常工作时只能读取其中的信息; RAM 可进行读/写操作, 除读出外也可写入, 所以又称为读写存储器。一般的 RAM