



高等专科学校教材

单片微型计算机技术

刘国荣 主编



机械工业出版社

高等专科学校教材

单片微型计算机技术

刘国荣 主编



机械工业出版社

本书从应用的角度出发,着重介绍了MCS-51单片机的硬件结构、指令系统、程序设计和系统扩展方法以及接口技术。全书共分十一章,内容包括:计算机基础知识;MCS-51单片机的结构及指令系统;汇编语言程序设计方法;输入输出和中断;定时/计数器;串行通信接口;MCS-51单片机系统的扩展;可编程接口芯片;A/D和D/A转换;人机联系接口技术和单片机的应用。

本书叙述由浅入深,通俗易懂,特别适用于微机原理的初学者。本书由基本内容和扩充内容两部分组成,既可作为高等院校大专非电类专业微机原理教材,也可作为大、中专电类专业微机原理教材,还可供有关工程技术人员参考。

单片微型计算机技术

刘国荣 主编

责任编辑:韩雪清 版式设计:霍永明

封面设计:姚毅 责任校对:刘志文

责任印制:路琳

机械工业出版社出版(北京阜成门外百万庄南街一号)

邮政编码:100037

(北京市书刊出版业营业许可证出字第117号)

北京市房山区印刷厂印刷

新华书店北京发行所发行·新华书店经售

开本 $787 \times 1092^{1/16}$ ·印张15.5·字数381千字

1997年1月第1版第1次印刷

印数 00 001—10 000 定价:18.00元

ISBN 7-111-05379-6/TP·418(课)

ISBN 7-111-05379-6



9 787111 053798 >

凡购本书,如有缺页、倒页、脱页,由本社发行部调换

前 言

为适应工程专科培养应用型人才的目标,促进教学改革,提高教学质量,湖南省高校电工学会于1987年9月组织省内部分工程专科学校编写了供专科使用的《电工技术》、《电子技术》和《微型计算机技术》三本教材,本书是湖南省高校电工学会组织编写的第二轮专科微机原理教材。

本书共有十一章,主要介绍微型计算机基础知识,MCS-51单片机的硬件结构、指令系统、程序设计方法及其系统的扩展方法,常用接口芯片及接口技术等。最后,作为对全书的总结,较详细地介绍了一个MCS-51单片机应用实例——单片机防盗报警系统。

本书的一个突出特点是采用符合教学规律的编写方法,将指令系统和程序设计融为一章,以程序设计为主线,讲完一类指令,便介绍一类程序的设计方法,利于读者记忆和理解指令。

本书重点突出,选材注重应用。原理性内容只作了一般介绍,重点放在程序设计、系统扩展、片内硬件资源应用和接口技术方面。对接口电路芯片、片内定时/计数器、串行口等,着重介绍它们的外部特性和使用方法,并列举了多个应用例子,从软硬件结合的角度作了详尽的分析。

本书是一本以MCS-51单片机为样机的微机原理教材,对未学过微机原理,而直接学习单片机技术的读者是非常适用的。全书叙述简单明了、通俗易懂,几乎每节之后都附有思考题,每章有小结和习题,便于教学和自学。

本书由基本内容和扩充内容(带*号部分)两部分组成。基本内容适用于专科非电类专业学生,教学时数为62~66学时(其中实验为10~14学时);基本内容加扩充内容,适用于专科电类专业学生,教学时数为86~90学时(其中实验为20~24学时)。

参加本书编写的有湖南省水利水电学校蔡立红(第一、二章)、湖南大学衡阳分校王杰文(第三章第一~五节)和何映(第三章第六、七节)、湘潭机电高等专科学校李晓秀(第四、五章)和刘国荣(第九、十章)、湖南省纺织专科学校杨子华(第六、七章)、长沙工业高等专科学校罗桂娥(第八、十一章)。刘国荣任主编、罗桂娥、王杰文任副主编。

本书承湖南大学熊秋思教授仔细审阅,并提出了许多宝贵意见,谨致以衷心感谢。

由于编者水平有限,书中缺点和错误在所难免,敬请读者批评指正。

编 者

1996年3月

目 录

前言		寄存器	109
第一章 微型计算机基础	1	第三节 定时/计数器的工作方式	110
第一节 微型计算机的发展和应用概况	1	第四节 定时/计数器应用举例	112
第二节 单片机的特点及常用系列简介	1	小结	115
第三节 数制与码制	2	习题	115
第四节 微型计算机的基本组成电路	11	第六章 串行通信接口	117
第五节 微型计算机的基本结构及工作		第一节 串行通信的一般概念	117
原理	17	第二节 MCS-51 单片机的串行通信	
小结	23	接口	119
习题	24	第三节 MCS-51 单片机串行接口应	
第二章 MCS-51 单片微型计算机		用举例	124
的结构	25	小结	137
第一节 MCS-51 单片机的基本组成	25	习题	138
第二节 MCS-51 单片机的内部结构	26	第七章 MCS-51 单片机系统的扩展	139
第三节 MCS-51 单片机引脚功能及		第一节 MCS-51 单片机最小应用系统	139
片外三总线结构	34	第二节 程序存储器的扩展	140
小结	37	第三节 数据存储器的扩展	145
习题	38	第四节 并行 I/O 口的扩展	148
第三章 程序设计基础	39	小结	152
第一节 指令及指令格式	39	习题	153
第二节 寻址方式	41	第八章 可编程接口芯片	154
第三节 简单程序的设计	44	第一节 8255A 可编程接口芯片	154
第四节 分支程序设计	57	第二节 8155 可编程接口芯片	165
第五节 循环程序设计	65	小结	172
第六节 堆栈和子程序	69	习题	172
第七节 汇编语言程序设计	76	第九章 数/模和模/数转换	173
小结	91	第一节 数/模 (D/A) 转换	173
习题	92	第二节 模/数 (A/D) 转换	179
第四章 输入输出和中断	96	小结	186
第一节 输入与输出	96	习题	187
第二节 中断	99	第十章 人机联系接口技术	188
小结	107	第一节 键盘接口	188
习题	107	第二节 LED 显示器接口	193
第五章 定时/计数器	108	第三节 可编程键盘、显示器接口——	
第一节 定时/计数器的结构和工作		Intel8279	199
原理	108	第四节 微型打印机接口	210
第二节 定时/计数器的方式和控制		小结	217

IV

习题.....	218	报警系统.....	226
第十一章 单片微型计算机的		小结.....	233
应用	219	习题.....	234
第一节 MCS-51 单片机应用系统		附录 A MCS-51 指令表	235
设计过程.....	219	附录 B ASCII (美国标准信息交	
第二节 硬件设计方法.....	221	换码) 表	240
第三节 软件设计方法.....	224	附录 C 常用芯片引脚图	241
第四节 应用实例——单片机防盗		参考文献	244

第一章 微型计算机基础

本章首先介绍微型计算机的发展和应用概况，讲述单片机的特点及学习微型计算机原理所需的基础知识，如计数制、不同数制之间的转换、二进制的运算规则、数的编码方法及微型计算机的基本组成电路等，简要介绍微型计算机的基本结构和基本工作原理。目的在于使读者掌握微型计算机的一些基本概念，对微型计算机有一个概括性的了解，为学习后面各章打下基础。

第一节 微型计算机的发展和应用概况

微型计算机是计算机技术与大规模集成电路发展相结合的产物。在计算机领域中，它是发展最快，应用最广的一个分支。

微型计算机的发展概括起来经历了四代：

第一代(1971~1973年)是以 Intel 公司的 4004 和 8008 为代表的 4 位、8 位低档微处理机。

第二代(1973~1978年)是以 I8080、M6800、Z80 为代表的 8 位机。

第三代(1978~1981年)是以 I8086、M68000、Z8000 为代表的 16 位机。

第四代(1981年至今)是以 IAPX432、HP32 为代表的 32 位机。

随着超大规模集成电路的发展以及计算机软件技术和应用技术的日益成熟完善，微型计算机正朝着以下方向发展：提高集成度；提高处理速度；增强功能（增加字长、扩大存储量、增加中断和数据交换能力等）；加强软件支持能力；提高性能价格比。

众所周知，微型计算机以其运算速度快、用途广、体积小、使用灵活、价格便宜等优点已广泛渗透到各个部门及日常生活中，并显示出强大的生命力。它的应用可概括为两大类型。

(1) 检测控制类 如将微型计算机用于过程控制、数值控制；把微机与检测仪器、仪表连接起来，可以存储测量信息，并对其进行分析、综合和各种处理；大大提高检测效率，实现检测自动化。

(2) 数据处理类 用微型计算机进行科学计算、辅助设计 (CAD)、企业管理等。

第二节 单片机的特点及常用系列简介

单片微型计算机简称为单片机。它是在一块芯片上集成的一台微型计算机，芯片内一般集成有微处理器（也叫中央处理单元 CPU）、存储器（RAM、ROM、或 EPROM）、输入/输出 (I/O) 接口，定时/计数器、时钟电路等。有些单片机除并行 I/O 接口外，还增添了串行接口、A/D 转换接口等。

单片机是微型计算机发展中的一个分支，以其独特的结构和优点，深受广大用户欢迎和重视，发展极为迅速，应用极为广泛。它的应用已渗透到国民经济和人民生活的各个领域，并发挥出很大的效益。

单片机已经历了 4 位单片机, 8 位低档单片机, 8 位高档单片机, 16 位单片机。现在正向高性能、高速度、高集成度、大容量、多功能、低功耗、加强 I/O 能力及结构兼容的 32 位和双 CPU 方向发展。

一、单片机的特点

单片机除了体积小、价格便宜外, 还有以下一些特点。

1) 具有通用性和灵活性。单片机与其他微型计算机一样, 按照用户给定的程序去工作。用户可以根据应用需要去开发它。单片机不仅可应用于生产各领域, 还可应用于普通生活中。若要改变一个单片机应用系统的功能, 硬件可以基本不变, 只需改变固化在存储器中的程序即可。

2) 面向控制。单片机的指令系统适合实时控制。它有丰富的转移指令、I/O 口逻辑操作指令, 位处理功能很强, 而且中央处理单元 CPU 逻辑控制功能也很强, 运行速度快。所以, 特别适合于工业控制、仪器仪表及数据采集诸领域。

3) 可靠性高、抗干扰能力强。由于单片机把各个功能部件集成在一块芯片上, 所以不仅大大缩短了系统内部信息传输的距离, 而且也有效地减少了外部干扰, 提高了可靠性。

4) 具有掉电保护。

5) I/O 接口功能很强, 便于系统扩展。

6) 应用研制周期短, 开发效率高。一般微型计算机应用研制周期时间较长, 而单片机应用研制周期较短。因为单片机开发较简单, 只要通过简易开发装置, 就可实现在线开发, 即直接在实际环境中开发。当它一旦开发成功, 就可付诸实际应用。

二、单片机常用系列简介

自 4 位单片机问世以来, 在短短的十多年间, 高性能产品不断问世, 典型的单片机有 TMS 系列、F8 系列、Z8 系列、MC 系列、MCS 系列等。MCS-51 系列是目前 8 位单片机的主流机型, 在实时控制、智能化仪器仪表等方面应用最广。MC6805 系列是具有多种专门用途的单片机大家族, 因而在家用电器及一些专用控制场合应用最为广泛。Z8 系列单片机的规模相对而言要小些。各种常见的单片微型计算机性能一览表见表 1-1。

第三节 数制与码制

一、数制及其互换

数制是计数的进位制, 最常用的数制是十进制, 十进制有 10 个不同的数字符号, 即 0, 1, 2, ..., 9。这些字符也叫数码。数制所用数码的个数称为基数。例如十进制有 10 个数码, 则基数为 10, 计数时逢 10 进 1, 除十进制外, 常用的数制还有二进制、八进制、十六进制等。表 1-2 列出了这几种数制的基数和字符。

基数小于 10 的数制, 用十进制的相应字符作为它的字符。如二进制的字符为 0、1, 八进制的字符为 0—7。基数大于 10 的数字, 采用十进制字符加上大写的英文字母 (从首字母 A 开始) 作为它的字符。如十六进制的字符为 0—9、A—F。

字符在数中的位置不同, 所表示的十进制数的值也不同。例如十进制数 123.4, 字符“1”处于百位, 其值为 100; “2”处于十位, 其值为 20; “3”处于个位, 其值为 3; “4”在小数点后一位, 即 1/10 位, 其值为 4/10。123.4 这个数可以展开为下面的多项式

表 1-1 常见单片微型计算机性能一览表

类别	公司	系列	典型品种型号	同系列产品型号	ROM (KB)	片内 RAM (B)	并行 I/O 线(根)	串行 I/O 口(个)	定时/计数器 (位)	中断源 (个)	典型晶振 (MHz)	D/A (位)	A/D (位)	其他
8 位 机	Intel	MCS -48	8048	8748	8035	64	27	1	1×8	2	8			
			8049	8749	8039	128	27	1	1×8	2	11			
		MCS -51	8051	8751	8031	128	32	1	2×16	5	12			8052AH-BASIC 片内固化有 BA-SIC 解释程序, 80C51 具有低功耗及待机工作方
			8052	8752	8032	256	32	1	2×16	6	12			式
			80C51	87C51	80C31	128	32	1	2×16	5	12			
			8044	8744	8344	192	32	1	2×16	5	12			
	Motorola	MC 6805	80C252	87C252	83C252	256	32	1	3×16	7	12			MC6805T ₂ 片内有锁相环, MC6805R ₂ 有一个 8 位 AD
				MC6805(P. U. R. T)	1~4	64~128	20~32	1	1×(8~16)	4	4			
				MC6805(G. F. E. C)										
16 位 机	Zilog	Z8	Z8601	Z8602	Z8603	128	32	1	2×8	6	8			Z8671 片内固化有 BASIC 解释程序
			Z8611	Z8612	Z8613	128	32	1	2×8	6	12			
		MCS -96	8394	8794	8094	232	32	1	2×16	8	12			4×16 位软件定时器, 1×16 位监视定时器, D/A 变换器用 PWM 实现, 具有高速输入输出器件 HSI、HSO, 8098 是准 16 位机
			8396	8796	8096	232	48	1	2×16	8	12	1	4×10	
	Intel		8395	8795	8095	232	32	1	2×16	8	12	1	8×10	
			8397	8797	8097	232	48	1	2×16	8	12	1	4×10	
			8398	8098		232	24	1	2×16	8	12	1	8×10	
			80C196			232	40	1	2×16	8	12	1	8×10	
32 位 机	Texas	TMS	TMS32020	TMS320C25 TMS320C30	4	544×2	32	1	1×16	3	32~40			TMS32020 是准 32 位 DSP, 指令周期 100ns

表 1-2 计数制的基数和字符表

数制	基 数	字 符
二进制	2	0 1
十进制	10	0 1 2 3 4 5 6 7 8 9
十六进制	16	0 1 2 3 4 5 6 7 8 9 A B C D E F

$$123.4 = 1 \times 10^2 + 2 \times 10^1 + 3 \times 10^0 + 4 \times 10^{-1}$$

式中, 10^2 、 10^1 、 10^0 、 10^{-1} 称为相应位的“权”, 每一位上的字符与该位“权”的乘积, 就是该位数以十进制表示的值, 上式称为按权展开式。

其他进制的数, 也可按权展开, 其一般形式为

$$N_x = D_{n-1}X^{n-1} + D_{n-2}X^{n-2} + \cdots + D_0X^0 + D_{-1}X^{-1} + \cdots + D_{-m}X^{-m} = \sum_{i=n-1}^{-m} D_i X^i$$

式中, D_i 为第 i 位的字符; X 为基数; n 为整数的位数; m 为小数的位数。

例如, 二进制数 $1100.1 = 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 0 \times 2^0 + 1 \times 2^{-1}$

十六进制数 $2FA4 = 2 \times 16^3 + 15 \times 16^2 + 10 \times 16^1 + 4 \times 16^0$

在计算机中, 主要采用二进制。

1. 二进制

(1) 计算机中为什么采用二进制

1) 二进制只有 0 和 1 两个数码, 很容易用电子元器件来实现。如电压的高和低、脉冲的有和无、电灯的亮和灭均可以用 0 和 1 来代表, 且这种简单的状态工作可靠, 抗干扰能力强。

2) 二进制数的运算公式简单, 加法有四个, 乘法有四个, 即

$$0+0=0 \quad 0+1=1 \quad 1+0=1 \quad 1+1=10$$

$$0 \times 0 = 0 \quad 0 \times 1 = 0 \quad 1 \times 0 = 0 \quad 1 \times 1 = 1$$

显然它比十进制运算要简单得多。因此, 计算机的运算电路也可以大大简化。

下面通过几个四则运算的例子来领会二进制运算的规则。

① 加 (进位规则是逢二进一)

$$\begin{array}{r} 1011 \\ +1010 \\ \hline 10101 \end{array}$$

$$\begin{array}{r} 01010111 \\ +10011111 \\ \hline 11110110 \end{array}$$

② 减 (借位规则是借一当二)

$$\begin{array}{r} 1101 \\ -0110 \\ \hline 0111 \end{array}$$

$$\begin{array}{r} 11010111 \\ -01011110 \\ \hline 01111001 \end{array}$$

③ 乘

$$\begin{array}{r} 1101 \\ \times 101 \\ \hline 1101 \\ 0000 \\ 1101 \\ \hline 1000001 \end{array}$$

$$\begin{array}{r} 1111 \\ \times 1101 \\ \hline 1111 \\ 0000 \\ 1111 \\ 1111 \\ \hline 11000011 \end{array}$$

④ 除

$$\begin{array}{r}
 0001101 \\
 110 \overline{) 1001110} \\
 \underline{110} \\
 111 \\
 \underline{110} \\
 110 \\
 \underline{110} \\
 0
 \end{array}$$

由上可见,在加、减、乘、除四则运算中,乘法实质上是做移位加法,除法实质上是做移位减法。后面将要介绍,减法也可化做加法。因此,在计算中只要用一加法器就可完成四则运算,从而使运算电路大大简化。

(2) 二进制(或其他进制)转换为十进制

将二进制(或其他进制)数转换为十进制数的基本方法是对二进制数(或其他进制数)按权展开求和得到十进制数。

例 1-1 将二进制数 1101, 11.101 和十六进制数 64 分别转换为等值的十进制数。

解 $(1101)_2 = 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 = 8 + 4 + 1 = (13)_{10}$

$(11.101)_2 = 2^1 + 2^0 + 2^{-1} + 2^{-3} = 2 + 1 + 0.5 + 0.125 = (3.625)_{10}$

$(64)_{16} = 6 \times 16^1 + 4 \times 16^0 = 96 + 4 = (100)_{10}$

(3) 十进制转换为二进制

1) 将十进制整数转换为等值的二进制整数的基本方法是除 2 取余逆排法,即将十进制整数不断用 2 除,直到商为 0 为止,历次余数从后往前排,得到等值的二进制数。

例 1-2 将十进制数 7、13 转换为等值的二进制数。

解

2/7	...	余 1		
2/3	...	余 1		
2/1	...	余 1		
0				

1
1
1

即 $(7)_{10} = (111)_2$

2/13	...	余 1		
2/6	...	余 0		
2/3	...	余 1		
2/1	...	余 1		
0				

1
1
0
1

即 $(13)_{10} = (1101)_2$

如将基数 2 改为 N, 则上述方法适用于将十进制数转换成等值的 N 进制整数。

2) 十进制小数转换为二进制小数的基本方法是乘 2 取整顺排法: 将十进制小数不断用 2 乘, 直到积为 0 或转换位数达到要求为止, 每次的整数部分取出, 从先往后排, 即得相应的二进制小数。

例 1-3 将十进制小数 0.875, 0.325 转换成等值的二进制小数。

解

$$\begin{array}{r|l}
 0.875 & \xrightarrow{\quad} 0.75 \\
 \times 2 & \times 2 \\
 \hline
 1.750 & 1.50 \\
 \downarrow & \downarrow \\
 1 & 1
 \end{array}
 \quad
 \begin{array}{r|l}
 0.75 & \xrightarrow{\quad} 0.5 \\
 \times 2 & \times 2 \\
 \hline
 1.50 & 1.0 \\
 \downarrow & \downarrow \\
 1 & 1
 \end{array}$$

即 $(0.875)_{10} = (0.111)_2$

$$\begin{array}{r|l}
 0.325 & \xrightarrow{\quad} 0.65 \\
 \times 2 & \times 2 \\
 \hline
 0.650 & 1.30 \\
 \downarrow & \downarrow \\
 0 & 1
 \end{array}
 \quad
 \begin{array}{r|l}
 0.65 & \xrightarrow{\quad} 0.3 \\
 \times 2 & \times 2 \\
 \hline
 1.30 & 0.6 \\
 \downarrow & \downarrow \\
 1 & 0
 \end{array}
 \quad
 \begin{array}{r|l}
 0.3 & \xrightarrow{\quad} 0.6 \\
 \times 2 & \times 2 \\
 \hline
 0.6 & 1.2 \\
 \downarrow & \downarrow \\
 0 & 1
 \end{array}$$

即 $(0.325)_{10} = (0.0101)_2$

2. 十六进制

二进制的缺点是位数太多, 书写起来麻烦, 且不易识别。因此, 在编写计算机的程序时, 常将它们写成十六进制数。表 1-3 是部分二进制、十进制、十六进制数的数码对照表。

表 1-3 二进制、十进制、十六进制数码对照表

十进制数	十六进制数	二进制数	十进制数	十六进制数	二进制数
0	0	0 0 0 0	8	8	1 0 0 0
1	1	0 0 0 1	9	9	1 0 0 1
2	2	0 0 1 0	10	A	1 0 1 0
3	3	0 0 1 1	11	B	1 0 1 1
4	4	0 1 0 0	12	C	1 1 0 0
5	5	0 1 0 1	13	D	1 1 0 1
6	6	0 1 1 0	14	E	1 1 1 0
7	7	0 1 1 1	15	F	1 1 1 1
			16	10	1 0 0 0 0

(1) 十六进制与二进制的转换 由表 1-3 可知, 每位十六进制数可由 4 位二进制数表示。因此, 将十六进制数的各位用相应的 4 位二进制数表示, 即可转换成等值的二进制数, 这种方法称为“一拉四”。

例 1-4 将十六进制数 $(3AF.C8)_{16}$ 转换成等值的二进制数。

解 $(3AF.C8)_{16} = 0011\ 1010\ 1111.\ 1100\ 1000 = (1110101111.11001)_2$

二进制数转换为十六进制数的方法为“四合一”, 即将二进制整数部分从右至左, 小数部分从左至右, 每 4 位为一组, 每一组用等值的十六进制数表示即可。

例 1-5 将二进制数 $(1011001.01011)_2$ 转换成等值的十六进制数。

解 $(1011001.01011)_2 = 0101\ 1001.0101\ 1000 = (59.58)_{16}$

(2) 书写规定 在前面的书写中,是用基数作下标来区别各种进制的。在书写计算机程序时,一般用相应的英文字母作后缀来标识各种进制的数。例如:

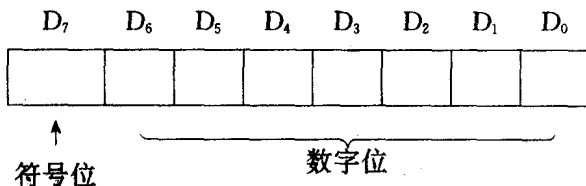
B (Binary) 表示二进制数。可将 $(1010)_2$ 写成 $1010B$ 。

D (Decimal) 表示十进制数。一般 D 可省略,即无后缀的数为十进制数。 $100D=100$ 。

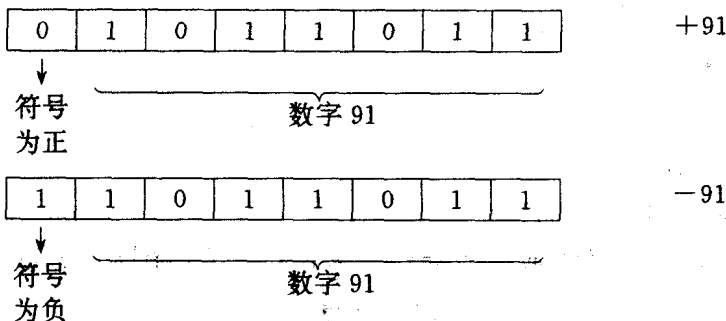
H (Hexadecimal) 表示十六进制数。如 $(100)_{16}=100H$ 。

二、带符号数的表示法

前面提到的二进制数,没有涉及符号问题,称为无符号数。但是实际上碰到的数会有正有负。那么,如何标识这些正负数的符号呢?通常规定:“一个数的最高位为符号位,用 0 表示正,1 表示负”。例如,在 8 位微型计算机中,一个 8 位二进制数,它的最高位 D_7 为符号位,后面 7 位 $D_6 \sim D_0$ 为数字位,其标准格式为



例如: $+91D = +1011011B$ $-91D = -1011011B$ 在机器中分别表示为



这种连同符号位一起作为一个数的数,称为机器数。它所表示的数值称为机器数的真值。

三、原码、反码、补码

在计算机中,机器数常用的三种表示方法是原码、反码、补码。下面分别加以介绍。

1. 原码

定义: 设 $|x| = x_1x_2 \cdots x_n$

$$[x]_{\text{原}} = \begin{cases} 0 & x_1x_2 \cdots x_n \quad x \geq 0 \\ 1 & x_1x_2 \cdots x_n \quad x < 0 \end{cases}$$

式中, $[x]_{\text{原}}$ 为原码,即机器数; x 为机器数的真值。

如果已知真值,则可由定义写出它的原码。例如

$$x = +5 = +0000101B$$

$$[x]_{\text{原}} = [+5]_{\text{原}} = 00000101B$$

$$x = -5 = -0000101B$$

$$[x]_{\text{原}} = [-5]_{\text{原}} = 10000101\text{B}$$

2. 反码

定义：设 $|x| = x_1x_2\cdots x_n$

$$[x]_{\text{反}} = \begin{cases} [x]_{\text{原}} & x \geq 0 \\ \overline{0x_1x_2\cdots x_n} & x < 0 \end{cases}$$

当 x 为正数时，其反码与原码相同。当 x 为负数时，其反码等于相应正数的原码按位取反。即 0 变成 1，1 变成 0。

例 1-6 求 $[+13]_{\text{反}}$ 、 $[-13]_{\text{反}}$ 、 $[+0]_{\text{反}}$ 、 $[-0]_{\text{反}}$

解 $[+13]_{\text{反}} = [+13]_{\text{原}} = 00001101\text{B}$

$[-13]_{\text{反}} = \overline{[+13]_{\text{原}}} = 11110010\text{B}$

$[+0]_{\text{反}} = [+0]_{\text{原}} = 00000000\text{B}$

$[-0]_{\text{反}} = \overline{[+0]_{\text{原}}} = 11111111\text{B}$

3. 补码

定义：设 $|x| = x_1x_2\cdots x_n$

$$[x]_{\text{补}} = \begin{cases} [x]_{\text{原}} & x \geq 0 \\ [x]_{\text{反}} + 1 & x < 0 \end{cases}$$

正数的补码与原码相同，负数的补码等于其反码加 1（亦即等于相应正数的原码按位取反后再加 1）。

例 1-7 求 $[+13]_{\text{补}}$ 、 $[-31]_{\text{补}}$ 、 $[-4]_{\text{补}}$ 。

解 $[+31]_{\text{补}} = [31]_{\text{原}} = 00011111\text{B}$

$[-31]_{\text{反}} = 11100000\text{B}$

$[-31]_{\text{补}} = [-31]_{\text{反}} + 1 = 11100001\text{B}$

$[-4]_{\text{反}} = 11111011\text{B}$

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

8 位二进制数为原码、反码、补码时所表示的真值见表 1-4，它们有以下特点：

表 1-4 8 位二进制数的真值表

真值 二进制数	码制	无符号二进制数	原码	反码	补码
00 00 00 00		0	+0	+0	+0
00 00 00 01		1	+1	+1	+1
00 00 00 10		2	+2	+2	+2
⋮		⋮	⋮	⋮	⋮
01 11 11 10		126	+126	+126	+126
01 11 11 11		127	+127	+127	+127
10 00 00 00		128	-0	-127	-128
10 00 00 01		129	-1	-126	-127
⋮		⋮	⋮	⋮	⋮
11 11 11 01		253	-125	-2	-3
11 11 11 10		254	-126	-1	-2
11 11 11 11		255	-127	-0	-1

1) 正负符号数字化，规定数字的最高位为符号位，用 0 表示正，1 表示负。

2) 数值为正时, $[x]_{\text{原}} = [x]_{\text{反}} = [x]_{\text{补}}$ 。

3) 数值为负时, 三种机器数的符号位均为 1, 但后面的数值部分各不相同。原码的数值部分与原数值相同; 反码的数值部分为原数值按位取反; 补码的数值部分为原数值按位取反后加 1。

4) 8 位二进制反码所能表示的范围是 $+127 \sim -127$ 。8 位二进制补码所能表示的范围是 $+127 \sim -128$ 。

利用补码, 可将减法运算变为加法运算, 符号位也可当作数值来处理, 一起参加运算, 非常方便。因此, 在计算机中, 带符号数一般用补码表示。且运算的结果也用补码表示。现举例说明如下。

例 1-8 $35 - 24 = 11$

解 用减法

$$\begin{array}{r} 00\ 10\ 00\ 11\quad 35 \\ -00\ 01\ 10\ 00\quad 24 \\ \hline 00\ 00\ 10\ 11\quad 11 \end{array}$$

用加法

$$\begin{array}{r} 00\ 10\ 00\ 11\quad 35 \\ +11\ 10\ 10\ 00\quad [-24]_{\text{补}} \\ \hline 1\ 00\ 00\ 10\ 11\quad [11]_{\text{补}} \end{array}$$

自然丢失

在 8 位机中, 最高位的进位已超出机器字长的范围, 故自然丢失。减法运算与补码的加法运算结果相同。

例 1-9 $(-5) + (-7) = (-12)$

解

$$\begin{array}{r} 11\ 11\ 10\ 11\quad [-5]_{\text{补}} \\ +11\ 11\ 10\ 11\quad [-7]_{\text{补}} \\ \hline 1\ 11\ 11\ 01\ 00\quad [-12]_{\text{补}} \end{array}$$

自然丢失

在和中符号位为 1, 故结果为负, 将结果的数值部分按位取反加 1, 求得原码为 10001100B, 即 -12。在微型计算机中, 带符号数一般用补码表示, 自然, 其运算结果一般也用补码表示。

四、BCD 码 (Binary Coded Decimal)

计算机中均用二进制数进行内部运算, 但人们习惯用的是十进制数, 故常要进行十进制与二进制之间的转换。对较大的数, 十进制与二进制之间的转换既麻烦又费时。因此, 我们另外采用一种把二进制和十进制的特点结合到一起的十进制数的编码方法。

如果十进制数中的每一个数码均用其等效的四位二进制数来表示, 这样产生的代码就称为 BCD 码, 又叫二-十进制码。

用 4 位二进制码可以写出 16 种组合 (0000~1111), 用来表示十进制数码 0, 1, 2, ..., 9 只需其组合中的 10 种, 最常用的是取前 10 个组合 (0000~1111), 这种编码称为 8421 码。我们后面谈到 BCD 码时, 均指 8421 码。用 BCD 码表示的数称 BCD 数, 即二进制编码的十进

制数。

十进制数码与BCD码的对应关系为

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

如十进制数 94.3 用 BCD 码表示为 10010100.0011。

要注意，同一个十进制数化成 BCD 数和二进制数后，结果是不同的。即 BCD 数与二进制数是两回事，BCD 数是二进制数的形式，但表示的是十进制数。如

$$127 = (01111111)_2$$

$$127 = (000100100111)_{\text{BCD}}$$

下面介绍 BCD 数加法运算步骤：

1) 用普通的二进制加法将两个 BCD 数相加。

2) 若和的低位 BCD 码向高位 BCD 码有进位或低位大于 9，则在低位加 6 调整；若和的高位 BCD 码向前有进位或大于 9，则在高位加 6 进行调整。

例 1-10 用 BCD 码表示十进制数完成以下运算。

$$\begin{array}{r}
 \begin{array}{r} 16 \\ +11 \\ \hline \end{array}
 \qquad
 \begin{array}{r} 47 \\ +35 \\ \hline \end{array}
 \qquad
 \begin{array}{r} 81 \\ +98 \\ \hline \end{array}
 \end{array}$$

解

$$\begin{array}{r}
 00\ 01\ 01\ 10\ 16 \\
 +00\ 01\ 00\ 01\ 11 \\
 \hline
 00\ 10\ 01\ 11\ 27
 \end{array}$$

运算结果正确。不需调整。

$$\begin{array}{r}
 \begin{array}{r} 01\ 00\ 01\ 11\ 47 \\ +\ 00\ 11\ 01\ 01\ 35 \\ \hline \end{array}
 \qquad
 \begin{array}{l}
 01\ 11\ 11\ 00\ 低位\ BCD\ 码\ 为\ 非法\ 码\ (>9) \\
 +\ \quad\ 01\ 10\ 对\ 低位\ 加\ 6\ 调整 \\
 \hline
 10\ 00\ 00\ 10\ 82 \\
 10\ 00\ 00\ 01\ 81 \\
 +\ 10\ 01\ 10\ 00\ 98 \\
 \hline
 00\ 01\ 00\ 01\ 10\ 01\ 高位\ BCD\ 码\ 向前\ 有\ 进位 \\
 +\ 01\ 10\ 对\ 高位\ 数\ 加\ 6\ 进行\ 调整 \\
 \hline
 00\ 01\ 01\ 11\ 10\ 01\ 179
 \end{array}
 \end{array}$$

对减法运算也有同样的情况，只是把运算步骤中的“加”改为“减”，“和”改为“差”，“进位”改为“借位”。例如 $75 - 38 = 37$ 的运算过程为

$$\begin{array}{r}
 \begin{array}{r} 01\ 11\ 01\ 01\ 75 \\ -\ 00\ 11\ 10\ 00\ 38 \\ \hline \end{array}
 \qquad
 \begin{array}{l}
 00\ 11\ 11\ 01\ 低位\ BCD\ 码\ 大于\ 9\ 且\ 向\ 高位\ 有\ 借位 \\
 -\ \quad\ 01\ 10\ 对\ 低位\ 减\ 6\ 调整 \\
 \hline
 00\ 11\ 01\ 11\ 37
 \end{array}
 \end{array}$$

MCS-51 单片机对 BCD 数进行加法运算时,有专门的十进制调整指令 DA A,它对运算结果自动进行加 6 调整。不过,DA A 指令只适用于 BCD 数的加法运算。对于减法运算,MCS-51 单片机指令系统中没设置专门的调整指令。

第四节 微型计算机的基本组成电路

● 无论多么复杂的计算机,都是由若干基本电路组成,这些基本电路也就是电子技术中的逻辑电路。所以,不管计算机多么复杂,它的直接功能就是执行逻辑运算,所有其他的问题,包括从简单的算术到高深的数学,都要先经由硬件或软件把它们化为逻辑问题来处理。本节对微型计算机的基本组成电路作一简要介绍。这些电路是组成计算机硬件的基础。

一、常用逻辑电路

逻辑电路是计算机执行运算、控制功能所必需的电路,是计算机的基本单元电路。

常用的逻辑电路有与门、或门、非门、与非门、或非门和异或门。与门、或门、非门、与非门、或非门和异或门电路的结构和基本原理在电子技术课程中详细讨论过,这里只简单介绍它们的逻辑符号及功能。

1. 与门

与门是一个具有多端输入,单端输出的逻辑电路。图 1-1a 所示是一个二输入与门,其中 A、B 为输入,F 为输出。其输入输出关系如表 1-5 所示,逻辑表达式为

$$F = A \times B \quad \text{或} \quad F = A \wedge B = A \cdot B = AB$$

功能:只有当输入端 A 和 B 全为高电平“1”时,输出端 F 才为高电平“1”。

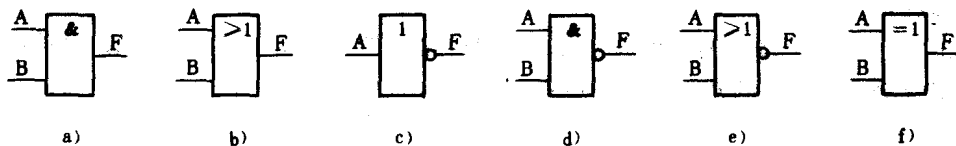


图 1-1 常用逻辑电路图形符号

a) 与门 b) 或门 c) 非门 d) 与非门 e) 或非门 f) 异或门

表 1-5 常用逻辑电路真值表

输 入		输 出 ΨF					
A	B	与门	或门	非门	与非门	或非门	异或门
0	0	0	0	1	1	1	0
0	1	0	1	1	1	0	1
1	0	0	1	0	1	0	1
1	1	1	1	0	0	0	0

2. 或门

或门是一个具有多端输入单端输出的逻辑电路。图 1-1b 所示是一个二输入或门,其输入输出关系如表 1-5 所示,逻辑表达式为

$$F = A + B \quad \text{或} \quad F = A \vee B$$

功能:如果一个以上的输入端为“1”时,输出端 F 为“1”;只有当所有输入端全为