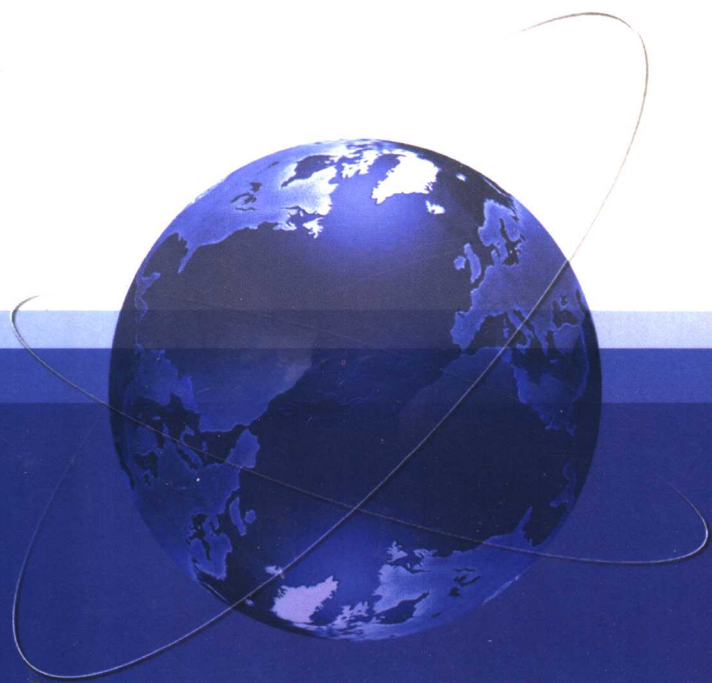




21世纪高职高专规划教材

(电工电子类)

单片机原理 及接口技术案例教程



李法春 主编



21 世纪高职高专规划教材
(电工电子类)

单片机原理及接口技术 案例教程

主 编 李法春
副主编 刘晓林 童成意 黄汉南
参 编 林 契 尹湛华 董晓倩 李 靖
主 审 陈仁森



机械工业出版社

本书是根据教育部教高〔2002〕2号文件精神编写的,以两个案例为主线,全面讲述掌握单片机应用所需的基础知识和基本技能,内容包括 MCS—51 单片机的组成结构、指令系统、程序设计、中断系统、定时器/计数器、串行接口、并行接口、存储器扩展、显示器与键盘、A/D 和 D/A 转换等内容。本书安排一定的实训内容,方便进行实践教学。还在附录中介绍了 C51 程序设计语言,方便学生掌握 C51,利用 C51 进行程序设计。

本书深入浅出、淡化理论、选材合理、通俗易懂,可作为 2 年制和 3 年制高职电子信息类、电气类及机电类各专业的教材,也可作为工程技术人员参考用书。

* 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

单片机原理及接口技术案例教程/李法春主编. —北京:机械工业出版社, 2006.1

21 世纪高职高专规划教材(电工电子类)

ISBN 7-111-18397-5

I. 单... II. 李... III. ①单片微型计算机—理论—高等学校:技术学校—教材②单片微型计算机—接口—高等学校:技术学校—教材

IV. TP368.1

中国版本图书馆 CIP 数据核字(2006)第 004364 号

机械工业出版社(北京市百万庄大街 22 号 邮政编码 100037)

策划编辑:余茂祚 责任编辑:余茂祚 版式设计:冉晓华

责任校对:刘志文 封面设计:饶薇 责任印制:李妍

北京中兴印刷有限公司印刷

2006 年 2 月第 1 版第 1 次印刷

787mm×1092mm $\frac{1}{16}$ ·15.75 印张·388 千字

0 001—4 000 册

定价:24.00 元

凡购本书,如有缺页、倒页、脱页,由本社发行部调换

本社购书热线电话(010)68326294

封面防伪标均为盗版

21 世纪高职高专规划教材 编 委 会 名 单

编委会主任 王文斌 郝广发

编委会副主任 (按姓氏笔画为序)

王建明	王明耀	王胜利	王寅仓	王锡铭
刘 义	刘晶磷	刘锡奇	杜建根	李向东
李兴旺	李居参	李麟书	杨国祥	余党军
张建华	茆有柏	秦建华	唐汝元	谈向群
符宁平	蒋国良	薛世山	储克森	

编委会委员 (按姓氏笔画为序, 黑体字为常务编委)

田建敏	成运花	曲昭仲	朱 强	刘 莹
刘学应	许 展	严安云	李连邨	李学锋
李选芒	李超群	杨 颀	杨群祥	杨翠明
吴 锐	何志祥	何宝文	佘元冠	沈国良
张 波	张 锋	张福臣	陈月波	陈向平
陈江伟	武友德	林 钢	周国良	宗序炎
赵建武	恽达明	俞庆生	夏春玲	晏初宏
倪依纯	殷建国	徐炳亭	徐铮颖	崔 平
崔景茂	焦 斌			

总 策 划 余茂祚

前 言

本书是根据教育部教高[2000]2号文件精神,由中国机械工业教育协会和机械工业出版社组织全国80多所高职高专院校联合编写的。

本书内容新颖、图文并茂,本着理论知识“必需、够用”为度的原则,以交通灯模拟控制系统和空调制冷控制系统两个案例为主线,以案例引入课题,以解决案例来讲解相关知识。本书适合作为2年制和3年制高职高专院校电子信息类、电气类、机电类各专业的教材。书中打“*”部分属于选学内容,各学校可根据专业实际要求进行讲授。

全书分为13章,第1章介绍了微型计算机基础知识;第2章主要介绍MCS—51单片机的硬件结构;第3章介绍MCS—51单片机的寻址方式及指令系统;第4章介绍了汇编语言程序设计;第5、6、7章分别介绍MCS—51单片机的中断系统、定时器/计数器、串行接口;第8章介绍并行接口,包括MCS—51最小系统、并行接口扩展芯片及8255;第9章介绍存储器扩展技术;第10章介绍显示器与键盘接口,包括8279;第11章介绍A/D和D/A接口功能;第12章简要介绍进行单片机应用程序的开发以及两个案例的最终实现方案;第13章安排了实训,设置了12个实训项目,方便各学校进行实践教学。

本书还在附录中介绍了C51程序设计语言,方便学生在学习MCS—51单片机汇编语言的同时,也能学习C51,掌握利用C51进行程序设计。

本书由李法春担任主编,刘晓林、童成意、黄汉南担任副主编,李靖编写第1、8章,尹湛华编写了第2、11章,刘晓林编写了第3章及附录A,李法春编写了第4章及附录B,黄汉南编写了第5、7章及第13章的前6节,董晓倩编写了第6、9章及第13章后6节,林契编写了第10章,童成意编写了第12章,全书由李法春负责总体策划与统稿、定稿等工作。广州城市职业学院陈仁森副教授审阅了全书并提出了许多宝贵意见,谨此表示衷心感谢。本书的编写参考了国内外许多单片机原理及接口技术相关知识的书刊和资料,在此向有关作者表示感谢!

本书在编写和出版过程中得到了多方帮助和支持,机械工业出版社余茂祚教授给予了大力支持和指导,广东农工商职业技术学院电子与信息工程系高俊文主任及相关教师提了许多建议,在此一并向他们表示诚挚的谢意!

由于时间仓促和编者水平有限,书中错误或不妥之处在所难免,恳请各位专家、读者批评指正,作者的E-mail地址:1_fchun@gdaib.edu.cn。

编 者

目 录

前言

第1章 微型计算机基础知识	1
1.1 数制与编码	1
1.2 带符号数的表示	5
1.3 微型计算机系统的组成及 工作过程	8
1.4 单片机的发展	11
1.5 单片机的特点及应用	12
1.6 单片机应用案例	12
复习思考题	14
第2章 MCS—51 单片机的硬件 结构	15
2.1 MCS—51 单片机基本结构	15
2.2 中央处理器 CPU	17
2.3 存储器	21
2.4 并行输入 / 输出接口	27
2.5 单片机的引脚及其功能	29
复习思考题	32
第3章 MCS—51 单片机指令系统	34
3.1 指令系统概述	34
3.2 寻址方式	36
3.3 数据传送指令	40
3.4 算术运算指令	45
3.5 逻辑运算指令	49
3.6 控制转移指令	52
3.7 位操作指令	56
复习思考题	59
第4章 汇编语言程序设计	61
4.1 汇编语言程序设计概述	61
4.2 伪指令	62
4.3 基本程序结构	65
4.4 子程序及参数传递	73
4.5 常用程序举例	76
复习思考题	80

第5章 MCS—51 单片机的中断 系统	81
5.1 中断概述	81
5.2 MCS—51 中断系统	82
5.3 中断系统的应用	89
5.4 空调制冷控制系统温度预 置案例	93
复习思考题	94
第6章 定时器/计数器	95
6.1 定时器 / 计数器的结构及 工作原理	95
6.2 方式和控制寄存器	96
6.3 定时器 / 计数器的工作方式	98
6.4 定时器 / 计数器应用举例	105
6.5 空调定时检测温度案例	108
复习思考题	109
第7章 串行接口	110
7.1 串行通信基础	110
7.2 MCS—51 单片机的串行 接口	113
7.3 串行口应用编程实例	121
复习思考题	129
第8章 并行接口	130
8.1 MCS—51 单片机的最小应用 系统	130
8.2 MCS—51 单片机扩展基本 知识	131
*8.3 8255A 可编程并行 I / O 接口	135
*8.4 8255A 控制流水灯案例	144
复习思考题	146
第9章 存储器的扩展	147
9.1 程序存储器的扩展	147
9.2 数据存储器的扩展	152

9.3 存储器的综合扩展	155	复习思考题	205
复习思考题	155	第 13 章 实训	206
第 10 章 显示器及键盘接口	157	13.1 拆字程序实验	207
10.1 显示器及其接口	157	13.2 数据传送程序实验	208
10.2 键盘及其接口	162	13.3 循环程序实验 (查找相同 数个数)	208
*10.3 8279 接口芯片	166	13.4 多分支程序实验	209
*10.4 显示器和键盘接口实例	173	13.5 P1 口亮灯实验	209
10.5 空调预置温度显示案例	179	13.6 数码管定时循环显示实验	210
复习思考题	180	13.7 串行口的自发自收实验	210
第 11 章 A/D 和 D/A 接口	181	13.8 P3.3 控制 P1 口实验	211
11.1 A/D 转换器接口	181	13.9 8255APA 口控制 PB 口 实验	212
11.2 D/A 转换器接口	190	13.10 简单 I/O 口扩展实验	213
11.3 空调的温度控制案例	195	13.11 A/D 转换实验	213
复习思考题	196	13.12 D/A 转换实验	215
第 12 章 单片机应用程序的开发	198	附录	216
12.1 单片机应用系统开发的 一般方法	198	附录 A C51 程序设计	216
12.2 交通灯模拟控制系统 案例实现	200	附录 B MCS—51 系列单片机 指令系统表	238
12.3 空调制冷控制系统案 例实现	203	参考文献	242

第 1 章 微型计算机基础知识

学习目标

1. 掌握二进制数和十六进制数的表示及与十进制间的相互转换。
2. 掌握二进制编码、符号数的补码表示、补码运算。
3. 了解定点数和浮点数。
4. 掌握微型计算机系统的组成,理解微型计算机的基本工作过程。
5. 了解单片机的发展、特点和应用。

重点难点

1. 补码运算及其溢出判断。
2. 微型计算机系统的组成。

1.1 数制与编码

计算机最基本的功能是进行数据的计算和处理加工,为了方便和可靠,计算机内部采用了二进制数字系统,因此,所有数值数据都必须采用二进制数表示,所有非数值数据,例如,字母、符号等也都必须采用二进制代码表示。

1.1.1 进位计数制

数制是进位计数制的简称。为区别不同的进位计数制,通常用字母来表示数制,D(Decimal)代表十进制(也可省略)、B(Binary)代表二进制、O(Octal)代表八进制、H(Hexadecimal)代表十六进制。

1. 十进制数 在日常生活中,人们最熟悉的是十进制数。十进制数有两个基本特点:

- 1) 每一位数是 0~9 十个数码中的一个数码,即基数为 10。
- 2) 逢十进一,借一当十。

任意一个十进制数可以用多项式表示为

$$N = K_n \times 10^n + K_{n-1} \times 10^{n-1} + \cdots + K_1 \times 10^1 + K_0 \times 10^0 + K_{-1} \times 10^{-1} + \cdots + K_{-m} \times 10^{-m} = \sum_{i=-m}^n K_i \times 10^i$$

式中, K_i 称为系数, 10^i 称为第 i 位的权, $K_i \times 10^i$ 即为加权系数,上式称为按权展开式。

例如,十进制数 687.25 按权展开为

$$687.25 = 6 \times 10^2 + 8 \times 10^1 + 7 \times 10^0 + 2 \times 10^{-1} + 5 \times 10^{-2}$$

2. 二进制数 二进制是计算机惟一能识别的机器语言。二进制数有两个基本特点:

- 1) 每一位数只能是 0 或 1 两个数码中的一个数码,即基数为 2。
- 2) 逢二进一,借一当二。

同十进制数,任意一个二进制数的按权展开式为:

$$N = K_n \times 2^n + K_{n-1} \times 2^{n-1} + \cdots + K_1 \times 2^1 + K_0 \times 2^0 +$$

$$K_{-1} \times 2^{-1} + \cdots + K_{-m} \times 2^{-m} = \sum_{i=-m}^n K_i \times 2^i$$

例如：二进制数 $1011.01\text{B} = 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 + 0 \times 2^{-1} + 1 \times 2^{-2}$

3. 十六进制数 二进制数书写冗长，为简化书写和阅读，常用十六进制（有时用八进制）代替二进制表示数据。十六进制数有两个基本特点：

1) 每一位数是 0~9、A~F 十六个数码中的一个数码，即基数为 16。

2) 逢十六进一，借一当十六。

其中，A 代表 10，B 代表 11，C 代表 12，依此类推。

同十进制数，任意一个十六进制数的按权展开式为

$$N = K_n \times 16^n + K_{n-1} \times 16^{n-1} + \cdots + K_1 \times 16^1 + K_0 \times 16^0 +$$

$$K_{-1} \times 16^{-1} + \cdots + K_{-m} \times 16^{-m} = \sum_{i=-m}^n K_i \times 16^i$$

例如：十六进制数 $8\text{C}2.\text{EH} = 8 \times 16^2 + 12 \times 16^1 + 2 \times 16^0 + 14 \times 16^{-1}$

1.1.2 数制间的转换

1. 其他进制数转换为十进制数 二进制数、十六进制数转换为十进制数时，只需将该数按权展开，求出各加权系数的和，便得到相应进制数对应的十进制数。如

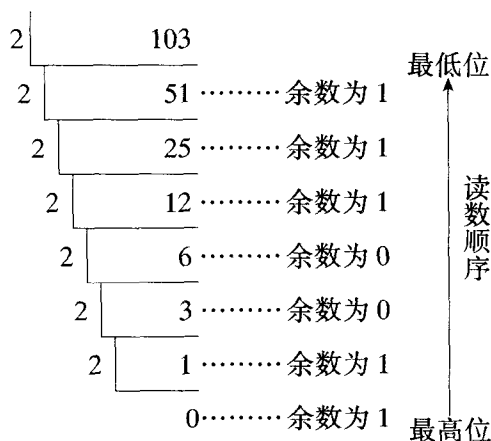
$$1101.011\text{B} = 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 + 0 \times 2^{-1} + 1 \times 2^{-2} + 1 \times 2^{-3} = 13.375$$

$$\text{B}3.\text{C}4\text{H} = 11 \times 16^1 + 3 \times 16^0 + 12 \times 16^{-1} + 4 \times 16^{-2} = 179.765625$$

2. 十进制数转换为其他进制数 十进制数转换为其他进制数时，要将整数部分和小数部分分别进行转换，再将结果排列在一起。其中，整数部分转换采用的方法是“除基数取余逆序”排列；小数部分转换采用的方法是“乘基数取整顺序”排列。下面举例说明：

例 1-1 将十进制数 103.625 转换成二进制数。

解：（1）将十进制数的整数部分转换为二进制数采用“除 2 取余逆序”方法，即将整数部分逐次除以 2，依次记下余数，直到商为 0。其中，第一个余数为二进制数的最低位，最后一个余数为最高位。



所以， $103 = 1100111\text{B}$

（2）将十进制小数部分转换为二进制数采用“乘 2 取整顺序”方法，即将小数部分连续乘

以 2，依次取积的整数部分作为二进制数的小数。

$$\begin{array}{rcl}
 0.625 & & \\
 \times 2 & & \text{最高位} \\
 \hline
 \boxed{1}.250 & \cdots \cdots \cdots & \text{整数为 1} \\
 \times 2 & & \\
 \hline
 \boxed{0}.500 & \cdots \cdots \cdots & \text{整数为 0} \\
 \times 2 & & \\
 \hline
 \boxed{1}.000 & \cdots \cdots \cdots & \text{整数为 1 最低位}
 \end{array}
 \quad \begin{array}{c} \downarrow \\ \text{读取顺序} \end{array}$$

所以， $0.625 = 0.101\text{B}$

由此可得 $103.625 = 1100111.101\text{B}$

需要指出的是，并非所有十进制小数都能用精确的二进制数表示。若十进制小数连续乘以 2 取整后，结果仍不为 0，此时应根据机器的精度，取一定的位数作为近似值。

例 1-2 将十进制数 301.359375 转换成十六进制数。

解：(1) 整数部分转换：将十进制数的整数部分转换为十六进制数采用“除 16 取余逆序”方法。

$$\begin{array}{rcl}
 16 \overline{) 301} & & \\
 16 \overline{) 18} & \cdots \cdots \cdots & \text{余数为 D} \\
 16 \overline{) 1} & \cdots \cdots \cdots & \text{余数为 2} \\
 & 0 & \cdots \cdots \cdots \text{余数为 1}
 \end{array}
 \quad \begin{array}{c} \uparrow \\ \text{最低位} \\ \text{读取顺序} \\ \uparrow \\ \text{最高位} \end{array}$$

所以， $301 = 12\text{DH}$

(2) 小数部分转换：将十进制小数部分转换为十六进制数采用“乘 16 取整顺序”方法。

$$\begin{array}{rcl}
 0.359375 & & \\
 \times 16 & & \text{最高位} \\
 \hline
 \boxed{5}.75 & \cdots \cdots \cdots & \text{整数为 5} \\
 \times 16 & & \\
 \hline
 \boxed{12}.00 & \cdots \cdots \cdots & \text{整数为 C} \\
 & & \text{最低位}
 \end{array}
 \quad \begin{array}{c} \downarrow \\ \text{读取顺序} \end{array}$$

所以， $0.359375 = 0.5\text{CH}$

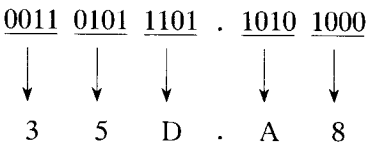
由此可得 $301.359375 = 12\text{D}.5\text{CH}$

3. 二进制数与十六进制数间的相互转换

(1) 二进制数转换成十六进制数：由于十六进制数的基数 $16 = 2^4$ ，故每位十六进制数用四位二进制数构成。因此，二进制数转换成十六进制数的方法为：以小数点为界分别向左、向右，每 4 位二进制数为一组用一个十六进制数码表示，不足 4 位的用 0 补足，其中整数部分左补 0，小数部分右补 0。

例 1-3 将二进制数 1101011101.10101B 转换成十六进制数。

解：因为

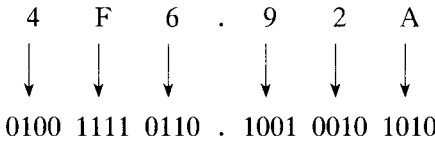


所以，1101011101.10101B=35D.A8H

(2) 十六进制数转换成二进制数：将每个十六进制数码用四位二进制数来表示。

例 1-4 将十六进制数4F6.92AH转换成二进制数。

解：因为



所以，4F6.92AH=10011110110.10010010101B

1.1.3 计算机中的常用编码

在计算机中常需要把二进制代码按一定规则排列起来表示某种特定含义，这就是编码。下面介绍在计算机中常使用的BCD码和ASCII码。

1. 二一十进制编码 二一十进制编码是指将十进制数的0~9十个数字用二进制数表示的编码，即BCD(Binary Coded Decimal)编码。

由于十进制数有十个不同的数码，因此需用4位二进制数来表示。而4位二进制编码有16种不同的组合，从中取出10种组合来表示0~9十个数有多种方案，所以BCD码也有多种方案。最常用的编码是8421BCD码，它是一种恒权码，8(2³)、4(2²)、2(2¹)、1(2⁰)分别是4位二进制数的权值。表1-1列出了8421BCD码十进制数和二进制编码的对应关系。

例 1-5 十进制数和BCD码相互转换。

十进制数86.5转换为8421BCD码为

86.5=(1000 0110.0101)_{8421BCD}

BCD码1001 0011.0100转换为十进制数为

(1001 0011.0100)_{8421BCD}=93.4

表 1-1 8421BCD 码

十进制数	8421BCD 码	十进制数	8421BCD 码
0	0000	5	0101
1	0001	6	0110
2	0010	7	0111
3	0011	8	1000
4	0100	9	1001

注意，同一二进制代码，当把它看作是二进制数和把它看作是二进制编码的十进制数，其数值是不同的。例如1000 0110作为二进制数时，其值为134，作为BCD码时，其值为83。

2. 字符编码 在计算机内，任何信息都是用代码表示的，字母、数字和符号(以后简称

为字符)也是用二进制代码表示的。目前使用的字符编码系统有许多种，计算机中普遍采用的是美国国家信息交换标准字符码，即 ASCII 码(American Standard Code for Information Interchange)，如表 1-2 所示。

表 1-2 ASCII 码表

字 符	高位	000(0H)	001(1H)	010(2H)	011(3H)	100(4H)	101(5H)	110(6H)	111(7H)
低位									
0000(0H)		NUL	DLE	SP	0	@	P	,	p
0001(1H)		SOH	DC ₁	!	1	A	Q	a	q
0010(2H)		STX	DC ₂	"	2	B	R	b	r
0011(3H)		ETX	DC ₃	#	3	C	S	c	s
0100(4H)		EOT	DC ₄	\$	4	D	T	d	t
0101(5H)		ENQ	NAK	%	5	E	U	e	u
0110(6H)		ACK	SYN	&	6	F	V	f	v
0111(7H)		BEL	ETB	'	7	G	W	g	w
1000(8H)		BS	CAN	(	8	H	X	h	x
1001(9H)		HT	EM	)	9	I	Y	i	y
1010(AH)		LF	SUB	*	:	J	Z	j	z
1011(BH)		VT	ESC	+	;	K	[	k	{
1100(CH)		FF	FS	,	<	L	\	l	
1101(DH)		CR	GS	-	=	M	]	m	~
1110(EH)		SO	RS	.	>	N	↑	n	~
1111(FH)		SI	HS	/	?	O	←	o	DEL

ASCII 码采用 7 位二进制代码对字符进行编码。它包括 52 个大、小写英文字母，10 个阿拉伯数字，32 个通用控制符号，34 个专用符号，共 128 个字符。例如 0~9 对应的 ASCII 码为 30H~39H，A~Z 对应的 ASCII 码为 41H~5AH。

通常，7 位 ASCII 码在最高位添加一个“0”组成 8 位代码，因此字符在计算机内部存储正好占一个字节。在存储和传送时，最高位常用作奇偶校验位，用于检查代码传输过程是否出现差错。偶校验时，每个二进制编码中应有偶数个 1。奇校验时，每个二进制编码中应有奇数个 1。例如，字母“F”的 ASCII 码为 1000110，因有 3 个 1，若用偶校验传送该字符，则奇偶校验位应为 1，传送的代码将为 11000110；若用奇校验传送，则奇偶校验位应为 0，传送的代码将为 01000110。

1.2 带符号数的表示

在很多数据处理过程中需要把二进制数作为无符号数看待，即所有二进制数位均为数值位。然而实际中的数会有正有负，这就存在一个带符号数的表示问题。通常把数的最高位作为符号位。这样符号位和数值位一起完整地表示带符号二进制数，就称为机器数，其数值就称为机器数的真值。在计算机中带符号数有三种表示方法——原码、反码和补码。

1.2.1 原码、反码和补码

1. 原码 将正数的符号位用 0 表示，负数的符号位用 1 表示，就称为原码表示法。例如：

$$X_1 = +91 = +1011011, \text{ 则 } [X_1]_{\text{原}} = 01011011$$

$$X_2 = -91 = -1011011, \text{ 则 } [X_2]_{\text{原}} = 11011011$$

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

8 位二进制原码所能表示的数值范围是：-127 ~ +127。

2. 反码 正数的反码与其原码相同；负数的反码是将其原码的符号位保持不变，数值位按位取反。例如：

$$[+91]_{\text{原}} = 01011011, \text{ 则 } [+91]_{\text{反}} = [+91]_{\text{原}} = 01011011$$

$$[-91]_{\text{原}} = 11011011, \text{ 则 } [-91]_{\text{反}} = 10100100$$

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

$$[-0]_{\text{原}} = 10000000, \text{ 则 } [-0]_{\text{反}} = 11111111$$

8 位二进制反码所能表示的数值范围为：-127 ~ +127。

3. 补码 正数的补码与其原码相同；负数的补码为其反码加 1。例如：

$$[+91]_{\text{补}} = [+91]_{\text{原}} = 01011011, [-91]_{\text{补}} = [-91]_{\text{反}} + 1 = 10100101$$

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

$$[-127]_{\text{反}} = 10000000, \text{ 则 } [-127]_{\text{补}} = 10000001$$

8 位二进制补码所能表示的数值范围为：-128 ~ +127。

1.2.2 补码的运算

1. 模的概念 模是指一个系统的量程或此系统所能表示的最大的数。以钟表为例，其模为 12。若标准时间为 6 点，而时针指向 10 点，调整时间的方法有两种，一是将时针倒拨 4 小时，即 $10 - 4 = 6$ ；二是将时针正拨 8 小时，即 $10 + 8 = 12 + 6 = 6$ 。这是因为在以 12 为模的钟表中，(+8)和(-4)互为补数，因此加 8 和减 4 的效果一样，即 $10 - 4 = 10 + 8 \pmod{12}$ 。由此不难得出：对于某一确定的模，某数减去小于模的数，可以用加上该数的补数来代替。所以引进了补码后，减法就可以转换为加法了。

2. 补码的运算 补码运算的几个公式：

$$[[X]_{\text{补}}]_{\text{补}} = [X]_{\text{原}}$$

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

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

其中， $[-Y]_{\text{补}}$ 为 $[Y]_{\text{补}}$ 的每一位(包括符号位)都按位取反，再加 1。

例 1-6 用补码进行 $X - Y$ 运算：

(1) 若 $X = 43$ $Y = 19$

$$\text{则 } [X]_{\text{补}} = [X]_{\text{原}} = 00101011\text{B} \quad [Y]_{\text{补}} = [Y]_{\text{原}} = 00010011\text{B} \quad [-Y]_{\text{补}} = 11101101\text{B}$$

所以有

$$\begin{array}{r} 0010 \ 1011\text{B} \quad [+43]_{\text{补}} \\ + \quad 1110 \ 1101\text{B} \quad [-19]_{\text{补}} \\ \hline [1]0001 \ 1000\text{B} \quad [+24]_{\text{补}} \\ \uparrow \\ \text{——进位, 丢掉} \end{array}$$

(2) 若 $X = 43$ $Y = -19$

$$\text{则 } [X]_{\text{补}} = [X]_{\text{原}} = 00101011\text{B} \quad [Y]_{\text{补}} = 11101101\text{B} \quad [-Y]_{\text{补}} = 00010011\text{B}$$

有

$$\begin{array}{r}
 0010\ 1011\text{B} \quad [+43]_{\text{补}} \\
 + \quad 0001\ 0011\text{B} \quad [+19]_{\text{补}} \\
 \hline
 0011\ 1110\text{B} \quad [+62]_{\text{补}}
 \end{array}$$

(3) 若 $X = -43$ $Y = -19$

则 $[X]_{\text{补}} = 11010101\text{B}$ $[Y]_{\text{补}} = 11101101\text{B}$ $[-Y]_{\text{补}} = 00010011\text{B}$

有

$$\begin{array}{r}
 1101\ 0101\text{B} \quad [-43]_{\text{补}} \\
 + \quad 0001\ 0011\text{B} \quad [+19]_{\text{补}} \\
 \hline
 1110\ 1000\text{B} \quad [-24]_{\text{补}}
 \end{array}$$

(4) 若 $X = -43$ $Y = 19$

则 $[X]_{\text{补}} = 11010101\text{B}$ $[Y]_{\text{补}} = [Y]_{\text{原}} = 00010011\text{B}$ $[-Y]_{\text{补}} = 11101101\text{B}$

有

$$\begin{array}{r}
 1101\ 0101\text{B} \quad [-43]_{\text{补}} \\
 + \quad 1110\ 1101\text{B} \quad [-19]_{\text{补}} \\
 \hline
 [1]1100\ 0010\text{B} \quad [-62]_{\text{补}}
 \end{array}$$

如上所示,一旦采用补码进行运算,所有参加运算的数和运算的结果都是用补码表示的,要得到真值还需转换。计算机里的实际情况就是这样。

需要指出的是,对任意给定的二进制数,可以把它看作有符号数,也可以看作无符号数,其差别就是如何看待最高位,所以对两个无符号数的加减运算,利用前面的有符号数补码运算规则得到的结果也仍然是正确的。

1.2.3 带符号数运算时的溢出问题

两个有符号数进行加减运算,若运算结果超出了机器所允许表示的范围,得出了错误结果,这种情况称为溢出。例如,8位字长的计算机所能表示的有符号数的范围为 $-128 \sim +127$,若运算结果超出此范围,就会发生溢出。

判断的方法:对加(减)法运算,判断最高位与次高位的进(借)位情况是否相同,若相同,则无溢出;若不同,则有溢出。

例 1-7 判断下列运算的溢出情况。

(1) $(+93) + (+54)$

$$\begin{array}{r}
 0101\ 1101\text{B} \quad [+93]_{\text{补}} \\
 + \quad 0011\ 0110\text{B} \quad [+54]_{\text{补}} \\
 \hline
 1001\ 0011\text{B} \quad [-109]_{\text{补}}
 \end{array}$$

有溢出,结果出错

(2) $(-63) + (+70)$

$$\begin{array}{r}
 1100\ 0001\text{B} \quad [-63]_{\text{补}} \\
 + \quad 0100\ 0110\text{B} \quad [+70]_{\text{补}} \\
 \hline
 [1]0000\ 0111\text{B} \quad [+7]_{\text{补}}
 \end{array}$$

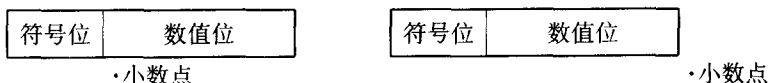
无溢出,结果正确

1.2.4 定点数和浮点数

在计算机中，用二进制表示实数的方法有两种，即定点表示法和浮点表示法。

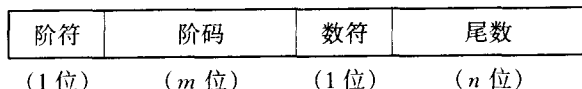
1. 定点数 定点表示法就是规定一个固定的小数点位置，并把用这种方法表示的数称为定点数。

一般地说，小数点规定在哪个位置上并没有限制，但为了方便，通常把数化为纯小数或纯整数，那么定点数就有下面两种表示方法：



定点表示法运算方便，但所有原始数据都要用比例因子化成纯小数或纯整数，计算后的结果又要用比例因子折算成实际值。并且，这种方法所表示的数的范围小、精度低。

2. 浮点数 一个十进制数可以写成多种表示形式，如 123.45 可以写成 0.12345×10^3 ， 12345×10^{-2} 等等。同样，一个二进制数也可以写成多种表示形式，如 1011.01 可以写成 0.101101×2^4 ， 101101×2^{-2} 等等。因此，任一个二进制数总可以表示为 $\pm d \times 2^{\pm p}$ 的形式，其中， d 称为尾数，它前面的符号称为数符； p 称为阶码，它前面的符号称为阶符。其表示格式如下：



像这种用阶码和尾数表示的数，称为浮点数。它能表示的数的范围大。

1.3 微型计算机系统的组成及工作过程

1.3.1 微型计算机系统的组成

计算机系统由硬件和软件两部分组成。

经过半个多世纪的发展，计算机系统结构已经发生了重大变化，性能已经有了极大提高，但其本质采用的仍然是冯·诺依曼的“存储程序”的思想。

冯·诺依曼的主要思想如下：

- 1) 采用二进制代码表示指令和数据。
- 2) 存储程序的工作方法。程序和数据事先放在内部存储器中，计算机运行时自动、连续地从存储器中取出指令并执行。
- 3) 计算机硬件系统由运算器、控制器、存储器、输入设备和输出设备五大部分组成。其中，运算器和控制器合称为中央处理器(即 CPU)。

1. 微型计算机系统的硬件组成 微型计算机系统的硬件组成如图 1-1 所示，它由微处理器、存储器、I/O 接口电路、I/O 设备、系统总线组成。

(1) 微处理器：在微型计算机中，把运算器、控制器和寄存器组集成在一个芯片上，称为微处理器，它是微型计算机的核心部件，具有运算器和控制器的功能。其结构如图 1-2 所示。

运算器由算术逻辑单元 ALU、累加器 A、暂存寄存器、标志寄存器 F 等组成，用来进行算术、逻辑运算。其中算术逻辑单元 ALU 由加法器和控制逻辑电路组成，是运算器的核

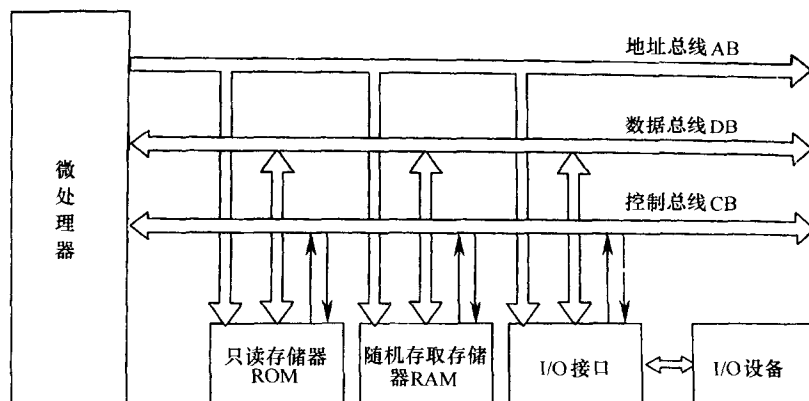


图 1-1 微型计算机系统的硬件组成框图

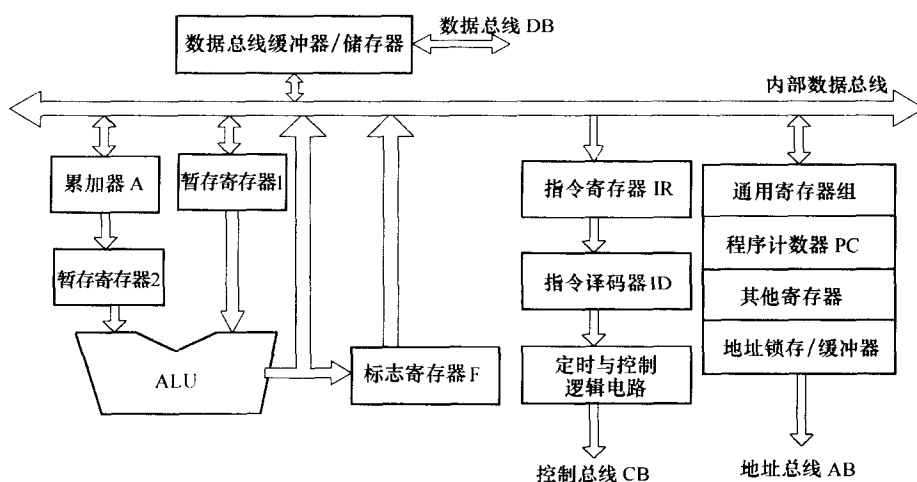


图 1-2 微处理器的结构框图

心。在控制器的指挥下，ALU 能够对分别来自两个暂存寄存器的操作数进行算术以及逻辑、移位等运算。运算后的结果经数据总线送至累加器 A，同时影响状态标志寄存器 F 的状态。

控制器由指令寄存器 IR、指令译码器 ID、定时与控制逻辑电路和程序计数器 PC 等组成，它是 CPU 的“指挥机关”，能够完成指令的读入、寄存、译码和执行操作。其中指令寄存器 IR 用来保存从存储器中读入的当前要执行的指令，它包括了指令的操作码和操作数两部分，其中，操作码指出指令进行的操作，操作数指出参加操作的数据或其地址；指令译码器 ID 对 IR 中保存的指令进行译码分析；定时与控制逻辑电路用来产生取指令和执行指令所需要的各种操作控制信号；程序计数器 PC 也叫指令地址计数器，用来保存下一条将要执行的指令的地址，也就是，由它提供一个存储器地址，从该地址对应的存储器单元中取出的内容就是将要执行的指令。

(2) 存储器：存储器是计算机存放程序和数据的部件。程序和数据在存储器中都必须以二进制的形式表示。这里介绍的存储器通常是指主存储器，也称为主存或内存。它由若干存储单元组成，每个存储单元按顺序都有一个惟一的编号，称为存储单元地址。一个存储单元

可以存放若干个二进制位(bit), 8个二进制位称为一个字节(Byte, 简称为B)。存储器就是以字节为单位存储信息的。

(3) I/O 接口电路: I/O 接口电路是CPU与I/O设备交换数据的通道和桥梁。由于I/O设备的种类繁多, CPU在与它们进行信息交换时, 会遇到速度不匹配、数据格式不同、工作方式差异等方面的问题, 这就必须由I/O接口电路进行转换和协调。

(4) I/O 设备: 输入设备是计算机从外部获取信息的装置, 其功能是将人们所熟悉的各种形式数据转换成计算机能识别的信息形式。常见的输入设备有键盘、鼠标等。

输出设备是将计算机的运算结果转换成人们或设备所能识别的形式, 如: 字符、文字、图像、声音等。常见的输出设备有显示器、打印机等。

通常, 将外存储器、I/O设备统称为外部设备, 简称外设。

(5) 系统总线: 系统总线是计算机中一组公共的信息交换通道。根据总线上传递的信息种类不同, 总线分为地址总线(Address Bus)、数据总线(Data Bus)、控制总线(Control Bus)。

三种总线中, CPU通过地址总线输出地址码用来选择某一存储器单元或某一I/O端口的寄存器; 数据总线用于CPU和存储器或I/O接口之间传送数据; 而控制总线用来传送自CPU发出的控制信息或送到CPU的状态信息。

2. 微型计算机的软件系统 软件是用于指挥计算机硬件系统工作的程序。软件系统包括计算机运行时所需要的程序、数据和文档。按其功能分类, 软件分为系统软件和应用软件。

系统软件是指负责管理、监控和维护计算机资源, 支持应用软件开发与运行的一类软件。包括操作系统、程序设计语言、语言处理程序、数据库管理系统、工具软件等。

应用软件是指为满足用户的不同应用目的而编制的程序。如字处理软件、过程控制软件、图书管理系统等。

1.3.2 微型计算机的基本工作过程

计算机的工作过程就是运行程序的过程。

任何复杂的操作, 都可以分解为一系列的基本运算, 计算机通过执行这一系列的基本运算后, 实现相应的功能。这里每种基本运算或操作就称为一条指令。所有指令的集合就是指令系统。一个微处理器的指令系统是设计微处理器时所决定的, 体现了该微处理器的功能强弱。把指令按功能要求排列起来就是程序。

本课程编写的程序是用字母或符号这些助记符表示的汇编语言程序。例如, 指令MOV A, #50H的功能是将50H传送到累加器A中。但是计算机中任何信息都必须以二进制代码形式存在, 因此汇编语言程序必须经过汇编, 翻译成二进制代码的机器码语言程序, 才能在计算机内执行。如上例, MOV A, #50H对应着机器代码0111010001010000, 即74H 50H, 其中74H为操作码部分; 50H为操作数部分。

根据冯·诺依曼存储程序的思想, 机器码语言程序准备执行时, 存放于存储器(内存)中, 计算机就会自动完成取指令和执行指令的任务了。

结合前面所述的计算机硬件结构, 下面以MOV A, #50H指令为例, 简要说明指令在CPU中的执行过程:

假设该指令的机器代码74H 50H存放于存储器00H和01H单元中。

1) 程序执行时, 程序计数器PC装入00H, 并将其送到地址总线上, 然后PC自动加1