机舱自动化与
微型计算机及其管理

曹善文 编著

人民交通出版社

机舱自动化与微型计算机及其管理

Jicang Zidonghua Yu Weixing ji suanji jiqi Guanli

曹善文　编著

人民交通出版社

（京）新登字091号

机舱自动化与微型计算机及其管理
曹 善 文 编著
插图设计：陈 竞 正文设计：乔文平 责任校对：戴瑞萍
人民交通出版社出版
（100013 北京和平里东街10号）
各地新华书店经销
北京市顺义县小店印刷厂印刷

开本：850×1168 $\frac{1}{32}$ 印张：8.375 插页：3 字数：225千

1993年4月 第1版
1993年4月 第1版 第1次印刷
印数：0001—1000册 定价：9.75元
ISBN7-114-01557-7

TP·00004

内 容 提 要

本书共分五章。第一章为微型计算机的基础知识。第二、三、四章分别为机舱巡回检测系统、柴油机主机遥控系统、船舶电站自动化。这三章对三大子系统的基本原理、微机在系统中的应用及参数调整、故障检查与排除等进行了较深入地分析和论述。第五章为机舱自动化系统的技术管理。

本书可供船舶轮机、电气管理人员和学校、科研、机务部门、修造船厂及有关技术管理人员学习参考。

前　言

机舱自动化是船舶自动化系统极其重要的组成部分，是一门理论性和实践性很强的综合性科学技术。近几年，微型计算机在船舶自动化中广泛应用，无人机舱自动化船舶不仅在数量上剧增，而且船舶自动化的发展已达到了机—驾合一的程度。

本书主要讲述机舱自动化中主要的三大子系统，即机舱巡回检测、柴油机主机遥控系统、电站自动化。为处理好普及与提高的关系，第一章介绍了微型计算机的一些基础知识。第二、三、四章分别对三大子系统的基本原理，微机在各子系统中的应用、参数调整、故障检查与处理等进行了较为深入地分析。第五章为机舱自动化系统的技术管理。

作者通过多年教学和在自动化船舶上的实践，力图将微型计算机、电气自动化和机械自动化三者之间的相互渗透、相互制约、相互影响的错综复杂关系有机地统一起来。书中用了一些典型实例，理论联系生产实际，由浅入深，由一般到特殊，再回到一般，以定性分析为主，也进行了一些必要地数学推导和定量分析，所以本书具有很强的实用性。

由于作者水平所限，不当之处在所难免，恳切希望广大读者批评指正。

编著者

目　录

第一章　微型计算机的基础知识

数字计算机是一种能对数字进行运算和处理的电子设备。其数字是以“0”或“1”的二进制代码来表示的。也就是说，计算机只认得二进制数，所有需要计算机加以处理的数、字母和符号都要用二进制编码来表示，它有一套独特的运算规则。要实现对二进制数的运算和处理，计算机又靠大量的数字逻辑部件来进行。从微型计算机应用的角度出发，本章主要介绍数制与编码、基本逻辑部件、逻辑运算以及微型计算机的基本原理。这是应用和管理计算机的最基本、最重要的理论知识。

§1-1　数制与编码

一、进位计数制

按进位的方法进行计数，称为进位计数制。在日常生活中，我们最熟悉最常用的是十进制数，还有廿四进制、六十进制等。在计算机中，常用的却是二进制、八进制和十六进制。我们不妨先从十进制计数入手，分析其特点及计数规律。

1．十进制数制（*Decimal Number System*）

一个十进制数有两个主要特点：

1）它有十个不同的数字符号：0、1、2、3、4、5、6、7、8、9和一个小数点符号。

2）它是逢“十”进位的。同一个数字符号处于不同的数位（称为“权”），所代表的数值是不同的。例如，在256.24这个数中，小数点左边第一位6代表个位，其值为6×10^0；左边第二位5代表十位，其值为5×10^1；左边第三位2代表百位，其值为

2×10^2；小数点右边第一位2代表十分位，其值为2×10^{-1}，右边第二位4代表百分位，其值为4×10^{-2}，所以这个数可以写成：

$$256.24=2\times10^2+5\times10^1+6\times10^0+2\times10^{-1}+4\times10^{-2}$$

一般地说，任意一个十进制数N都可以表示为：

$$N=K_{n-1}\times10^{n-1}+K_{n-2}\times10^{n-2}+\cdots+K_1\times10^1+K_0 10^0$$
$$+K_{-1}\times10^{-1}+K_{-2}\times10^{-2}+\cdots+K_{-m}\times10^{-m}$$
$$=\sum_{i=n-1}^{-m}(K_i\times10^i)$$

式中K_i表示第i位数码，可以是0～9个数字符号中的任何一个，由具体的数N来确定。m、n为正整数，n为小数点左面的位数，m为小数点右面的位数。10^{n-1}、10^{n-2}、$\cdots10^0$、$10^{-1}\cdots10^{-m}$称为十进制的"权"，而"10"为十进制数的基数（即在该进制中可能用到的数码的个数），它表示这种计数制为十进制。

从十进制第二个特点中还可以看出，各数字符号每向小数点左移一位，数的值就增长10倍，而向右移一位，则缩小10倍。

2． 二进制数制（*Binary Number System*）

与十进制类似，二进制也有两个主要特点：

1）它只有两个不同的数码"0"和"1"，同样也有一个小数点符号。

2）它是逢"二"进位的。例如，对十进制数$1+1=2$，而对于二进制数$1+1=10$，这里是逢二进一变成了"10"。不同的数码在不同的数位上所代表的值也是不同的。例如：

$$(1111.11)_2=1\times2^3+1\times2^2+1\times2^1+1\times2^0+1\times2^{-1}$$
$$+1\times2^{-2}=(15.75)_{10}$$

一般地说，任意一个二进制数N，都可以表示为：

$$N=K_{n-1}\times2^{n-1}+K_{n-2}\times2^{n-2}+\cdots+K_0\times2^0$$
$$+K_{-1}\times2^{-1}+\cdots+K_{-m}\times2^{-m}$$

$$= \sum_{i=n-1}^{-m} (K_i \times 2^i)$$

式中：K_i只能取“0”或“1”，由具体数N确定，m、n为正整数；2是二进制基数；2^{n-1}、2^{n-2}、…2^0、…2^{-m}是二进制的“权”。由二进制的特点还可以看出：各数码每向小数点左移一位，数的值增长两倍；而每向右移一位，则缩小两倍（数值除2）。

3．十六进制数制（*Hexadecimal Number System*）

其主要特点是：

1）它有十六个不同的数码，即0～9、A、B、C、D、E、F以及一个小数点符号。

2）它是逢“十六”进位的。

如上所述，任意一个十六进制数N，可表示为：

$$N = K_{n-1} \times 16^{n-1} + K_{n-2} \times 16^{n-2} + \cdots + K_0 \times 16^0$$

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

式中：K_i可以是0～9、A…F之中的一个，取决于数N；n、m为正整数；“16”为基数。

4．任意进制数制

通过以上对三种进制特点的分析可知，它们都有类似的一些特征，这些特征可归纳为：

1）R进制数具有R个基本的有序数码和一个小数点符号。R为该进制数的基数。其中最大数码的值为基数R减1。例如，十进制为9，二进制为1，八进制为7，十六进制为F（相当于十进制15）。

2）逢“R”进位。在减法运算中则按“借一当R”的原则，即每向小数点左移一位，其数值增大R倍，向右移一位，则减小R倍。一个R进制数N的多项式表示形式为：

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

$$+K_{-1}\times R^{-1}+\cdots+K_{-m}R^{-m}=\sum_{i=n-1}^{-m}(K_iR^i)$$

式中：K_i为第i位数码，由具体的数N来决定。R^i为R进制数第i位的“权”。

二、数制之间的转换

数制之间的转换很重要，特别是二进制、十六进制怎样转换为十进制或反之。

1. 任意进制数转换成十进制数

这种转换十分方便，只要将任意进制数按“权”展开相加，即可得到十进制数。

为了区分不同进制，在数的右下角加一个字母。如B(*Binary*) 表示二进制数，O (*Octal*) 表示八进制数，H (*Hexadecimal*) 表示十六进制数，不加字母的表示十进制数。

例如，将二进制数10111_B转换成十进制数：

$$10111_B=1\times2^4+0\times2^3+1\times2^2+1\times2^1+1\times2^0$$
$$=16+4+2+1=23$$

对于四位二进制数，为计算方便，只要记住它的权数分别为8、4、2、1，然后用心算加起来就可以立即求出二进制数转换成的十进制数。

例如：1000_B代表8，因1位的权数为8，其余位均为零；0111_B代表7，因为三个1位的权数分别为4、2、1，加起来得7；依次类推。

其他进制数也是按权展开相加后就可得到十进制数。

例如，将十六进制数$F3D_H$转换成十进制数为：

$$F3D_H=15\times16^2+3\times16^1+13\times16^0=3840+48+13$$
$$=3901$$

2. 以2^n为基数的进位制之间的转换

在微型计算机中常常使用以2^n为基数的进位制，如2^1进制、

$2^3=8$进制和 $2^4=16$ 进制等。任何以2^n为基数的进位制之间的转换都可以用直接的简便方法，因为这些进制都和二进制有直接的关系。例如十六进制与二进制的关系：

$$16^1=2^4$$

因此，一位十六进制数相当于四位二进制数。根据这种对应关系，二进制数与十六进制数之间转换十分简单，例如：

3	A	B	·	4	A	（十六进制数）
↓	↓	↓		↓	↓	
0011	1010	1011	·	0100	1010	（二进制数）

结果：$3AB\cdot4A_H=0011\ 1010\ 1011.0100\ 1010_B$

反之，将二进制数转换成十六进制数，可将二进制数从小数点分别向左和向右分为每四位一组用一位十六进制数码表示即可，例如：

0111	0110	·	1110	（二进制数）
↓	↓			
7	6	·	E	（十六进制数）

结果：$1110110\cdot111_B=76\cdot E_H$

根据上述分析，任何以2^n为基数的进制数都可以方便地转换成其他以 2^n 为基数的进制数。通常是把某进制数转换成二进制数，然后再将二进制数转换成另一要转换的进制数。例如，要将八进制数转换成十六进制数，可先将八进制数转换为二进制数，然后再将二进制数转换成十六进制数。

3．十进制数转换成任意进制数

这种转换要把整数部分和小数部分分开进行，然后再相加。

1）整数部分转换：将十进制整数部分除待转换数制的基数，直至商为0止，以最后余数为最高数位，依次排列余数，即可得所转换的进制数。

例如，将十进制数215转换成二进制数。二进制基数为2，所以用除 2 取余的方法，即用下式表示整个计算过程：

$$
\begin{array}{r|l}
2 & 215 \\ \hline
2 & 107 \quad \cdots\cdots\cdots\cdots \text{余 } 1=K_0 \text{（最低位）} \\ \hline
2 & 53 \quad \cdots\cdots\cdots\cdots \text{余 } 1=K_1 \\ \hline
2 & 26 \quad \cdots\cdots\cdots\cdots \text{余 } 1=K_2 \\ \hline
2 & 13 \quad \cdots\cdots\cdots\cdots \text{余 } 0=K_3 \\ \hline
2 & 6 \quad \cdots\cdots\cdots\cdots \text{余 } 1=K_4 \\ \hline
2 & 3 \quad \cdots\cdots\cdots\cdots \text{余 } 0=K_5 \\ \hline
2 & 1 \quad \cdots\cdots\cdots\cdots \text{余 } 1=K_6 \\ \hline
 & 0 \quad \cdots\cdots\cdots\cdots \text{余 } 1=K_7 \text{（最高位）}
\end{array}
$$

所以，$215=K_7K_6K_5K_4K_3K_2K_1K_0=11010111_B$

最后还可以按“权”展开相加来验证，即：

$1\times2^7+1\times2^6+1\times2^4+1\times2^2+1\times2^1+1\times2^0=215$

2）小数部分转换：将十进制小数部分乘待转换数制的基数，直到满足所要求的精度或小数部分等于零为止，把每次乘积的整数，以最初整数为最高数位，依次排列，即得到所转换的进位数。

例如：将十进制小数 0.625 转换成二进制小数。将十进制小数部分乘 2 取整，其整个转换过程如下：

$$
\begin{array}{rl}
 & 0.625 \\
\times & 2 \\ \hline
 & 1.250 \quad \cdots\cdots\cdots\cdots \text{整数部分 } 1=K_{-1} \text{（最高位）} \\
 & 0.250 \\
\times & 2 \\ \hline
 & 0.500 \quad \cdots\cdots\cdots\cdots \text{整数部分 } 0=K_{-2} \\
\times & 2 \\ \hline
 & 1.000 \quad \cdots\cdots\cdots\cdots \text{整数部分 } 1=K_{-3} \text{（最低位）}
\end{array}
$$

所以，$0.625=0.101_B$

十进制数转换成八进制或十六进制数，方法同上。也可以将十进制数先转换为二进制数，再把二进制数转换成八进制或十六进制数，这样做会更方便些。

三、二进制编码

如前所述，计算机只能识别二进制数，因此在计算机中，所有的数字、字母和符号等都要以特定的二进制码来表示，这就是二进制编码。

1．十进制数的二进制编码（BCD码）

用二进制数对每一位十进制数字编码，这种数制称为二—十进制，简称*BC* 码（*Binary Coded Decimal*）。

这种编码方式的特点是保留了十进制的“权”，而数字则用二进制数码即0和1的组合来表示。BCD码有8421码、余3码、格雷码等，这里介绍最常用的8421码。

在这一编码系统中，用四位二进制数表示一位十进制数字。它是逢“十”进位的。四位二进制数的每一位都有特定的“权”，从左到右各位的“权”依次为$2^3=8$、$2^2=4$、$2^1=2$、$2^0=1$故称8421BCD码。

十进制数0～9用8421 BCD码表示为：

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

又如：十进制数256可表示为：

2	5	6	（十进制）
0010	0101	0110	（BCD码）

但要注意，00100101 0110并非等效于256的二进制数，十进制数256转换为二进制数为100000000B。

此外，4位二进制数可表示十六种状态，而十进制数只有0～9共十种状态，即0000～1001。余下六种状态1010～1111在BCD编码中为非法码或称冗余码。若在BCD码运算中出现了非法码，则需作某些修正，才能得到正确的结果。

2．字符的编码

在计算机中，除了处理数字信息外，还必须处理一系列字母和符号，如英文26个字母、标点符号及一些常用运算符号等。这些字母和符号统称为字符。它们也必须按特定的规则用二进制编码才能在计算机中表示。

目前在微型计算机中，普遍采用美国标准信息交换码又称ASCII码（*American Standard Code for Information Interchange*），见表1-1。

表中二进制代码按顺序$b_6b_5b_4b_3b_2b_1b_0$排列。

ASCII码(7位码)　　　　表1-1

$LSD\ b_3b_2b_1b_0$		$MSD\ b_6b_5b_4$							
		0	1	2	3	4	5	6	7
		000	001	010	011	100	101	110	111
0	0000	NUL	DLE	SP	0	@	P	`	p
1	0001	SOH	DC1	!	1	A	Q	a	q
2	0010	STX	DC2	"	2	B	R	b	r
3	0011	ETX	DC3	#	3	C	S	c	s
4	0100	EOT	DC4	$	4	D	T	d	t
5	0101	ENQ	NAK	%	5	E	U	e	u
6	0110	ACK	SYN	&	6	F	V	f	v
7	0111	BEL	ETB	'	7	G	W	g	w
8	1000	BS	CAN	(	8	H	X	h	x
9	1001	HT	EM	)	9	I	Y	i	y
A	1010	LF	SU	*	:	J	Z	j	z
B	1011	VT	ESC	+	;	K	[	k	{
C	1100	FF	FS	,	<	L	\	l	|
D	1101	CR	GS	—	=	M	]	m	}
E	1110	SO	RS	.	>	N	Ω	n	~
F	1111	SI	US	/	?	O	—	o	DEL

表中：NUL—空；VT—垂直制表；SYN—空转同步；SOH—标题开始；FF—走纸控制；STX—正文结束；ETB—信息组传送结束；CR—回车；CAN—作废；EOT—传输结果；SO—移位输出；EM—纸尽；ETX—本文结束；SI—移位输入；SUB—减；ENQ—询问；DLE—数据链换码；ACK—承认；DC1～DC4—设备控制1～4；FS—文字分隔符；BEL—报警符；GS—组分隔符；BS—退一格；RS—记录分隔符；HT—横向列表；US—单元分隔符；LF—换行；NAK—否定；SP—空格；DEL—作废(删除)。

ASCII 码用七位二进制编码表示一个字符。如阿拉伯数字0～9的二进制代码分别是0110000～0111001（用十六进制表示为30H～39H）；大写英文字母A～Z用1000001～1011010(41H～5AH)表示；小写英文字母a～z用1100001～1111010（61H～7AH）表示。

§1-2　基本逻辑运算

电子计算机是由成千上万个数字逻辑电路和一些机械部件组成的复杂的数字逻辑装置。它可以对二进制数和逻辑关系进行处理和运算。

在逻辑运算中，常用的数学工具是逻辑代数，又称布尔代数。逻辑代数和普通代数有相似地方，它们都是用函数式来表达函数与变量之间的关系。如初等代数中的式子$y=ax+b$，其中y是函数，x是变量。在逻辑代数中 $P=A+B$，式中P 为逻辑函数，A、B是逻辑变量。但两者又有显著区别，前者y与x有很宽的取值范围，且是连续量，后者 P和 A、B 的取值只有“0”和“1”两个，且是开关量。它不仅能反映单个元件的两种状态“0”和“1”，而且能反映各逻辑元件之间的复杂逻辑关系。

各个元件之间的逻辑关系反映在数学上是几种最基本的运算关系，即逻辑或（逻辑加）、逻辑与（逻辑乘）和逻辑非运算。人们依据这三种最基本的逻辑运算关系，把许多逻辑线路组合成

庞大的电子数字计算机系统。下面介绍几种最基本的逻辑电路的工作原理和逻辑功能。

一、逻辑与（AND）

逻辑与是对两个或几个逻辑变量执行逻辑乘的运算。其实现电路称与门（*AND gate*）。图1-1a）、b）表示与门逻辑电路，旧的与逻辑符号见图1-1 c），IEC与逻辑符号见图1-1d）。

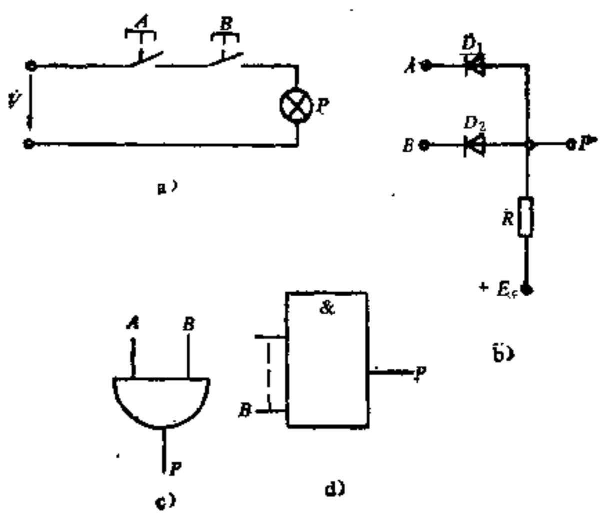

图 1-1 与门

逻辑与运算特点是只有当全部逻辑变量都为“1”时，输出逻辑函数才为“1”，否则为“0”。

图1-1中A、B为输入变量，输出为P。由图 1-1 a) 可知，