家用电器控制电路

徐崇庶　吴晴　主编

人民邮电出版社

前　言

我国的家用电器工业在改革开放中得到了迅速发展，取得了举世瞩目的成绩。现在国产家用电器中的洗衣机和电风扇的产量已跃居世界首位，电冰箱的产量也居于世界前列。以微处理器为核心的控制技术（本书中为了名词的统一，将具有智能作用的芯片统称为微处理器，包括单片机和专用芯片等），已开始应用到家用电器中，它使产品的性能、结构、质量都得到了提高。

本书在内容安排上是以家用电器产品为对象，以应用为主线，力求使读者在阅读本书之后，对进行家用电器控制的设计、调试和维修的能力有所帮助。全书共分两大部分：第一篇是基础知识，首先介绍了作为计算机知识基础的数字电路，然后以MCS—51系列单片机为重点，讲述了其硬件结构，指令系统和接口电路，对家用电器产品中常用的几种单片机也作了介绍，并介绍了近年来在家用电器中正在兴起的模糊控制技术。第二篇是家用电器控制电路，对遥控器、音响设备、电视机、录像机、摄像机、照相机、电冰箱、空调器、电饭锅、微波炉、洗衣机和电风扇等家用电器中的微处理器控制系统作了详细介绍、并从设计角度对其硬件电路和软件系统进行了分析，这些系统都是近来国内外家用电器新产品的实际系统，可供读者参考。

参加本书编写的有徐崇庶（第四章、8.3，10.5），鹿树理（第二、三章），吴晴（第一章、10.6），李晓毅（10.1，10.2，10.3，10.4），俞必忠（第五、六、七、八、九章），全书经徐崇庶、吴晴校改。书中有些章节引用了参考文献的内容，对其作者谨表衷心感谢。

由于编者水平有限，经验不足，书中缺点和错误一定不少，热忱欢迎读者批评和指正。

编者

一九九四年五月

目　录

第一篇　基础知识

第二篇　家用电器控制电路

第　一　篇

基础知识

第一章

数字电路基础知识

1.1 概述

随着电子技术的迅速发展，新型电子器件和电路层出不穷，种类也与日俱增。根据电路中工作信号的不同，可以将电子电路分为两大类：模拟电子电路和数字电子电路。

电子电路中的工作信号分为数字信号和模拟信号两类。数字信号是指在时间上和幅值上都是离散的信号。其特点是：

1. 数字信号的变化在时间上不连续，总是发生在一系列离散的瞬间；
2. 数字信号的数值大小和增减变化是不连续的，是某一个最小数量单位的整数倍。

工作于数字信号下的电子电路称为数字电路。

除数字信号之外的所有其它形式的信号统称为模拟信号。模拟信号的变化在时间上和幅值上都是连续的，工作于模拟信号下的电子电路即为模拟电路。

1.2 计算机的运算基础

1.2.1 数制和码制

一、几种常用的计数制

1. 十进制

十进制中，每一位数都可以用 0～9 这十个数码中的一个来表示，称“计数基数”为十。低位和相邻高位数之间的关系是“逢十进一”。

例如 $123.45 = 1\times10^2+2\times10^1+3\times10^0+4\times10^{-1}+5\times10^{-2}$

即任何一个十进制数都可以表示为各位的数码 Ki 与位权 10^i 之积再求和的形式：

$$S_{(十)}=\Sigma Ki\times10^i \tag{1-1}$$

由十进制数的构成规律不难推得任意进制（N 进制）数展开式的一般形式为

$$S_{(N)}=\Sigma Ki\times N^i \tag{1-2}$$

式中 K_i 为第 i 位的系数；N^i 称为第 i 位的权。

2. 二进制

二进制数中，每一位仅有 0 和 1 两种可能的数码，因此计数基数 $N=2$，相邻两位的进位关系为“逢二进一”，即有

$$S_{(二)}=\Sigma Ki\times 2^i \tag{1-3}$$

表 1-1 列出了四位二进制数。

表1-1　　四位二进制数

十进制数	二进制数	十进制数	二进制数
0	0000	8	1000
1	0001	9	1001
2	0010	10	1010
3	0011	11	1011
4	0100	12	1100
5	0101	13	1101
6	0110	14	1110
7	0111	15	1111

3. 八进制

八进制数中，每一位均用 0～7 这八个数码中的任一个表示，计数基数为八；相邻位的进位关系为“逢八进一”。八进制数均可按下式展开

$$S_{(八)}=\Sigma Ki\times 8^i \tag{1-4}$$

例如　$(23.45)_{(八)}=2\times8^1+3\times8^0+4\times8^{-1}+5\times8^{-2}$

4. 十六进制

十六进制数的每一位都应有十六种可能出现的数码，通常用 0～9，A、B、C、D、E、F 来表示。任何一个十六进制数均可展开为

$$S_{(十六)}=\Sigma Ki\times(16)^i \tag{1-5}$$

例如　$(2A.4E)_{(十六)}=2\times16^1+A\times16^0+4\times16^{-1}+E\times16^{-2}$

二、几种常用计数制间的转换

1. 二进制转化为十进制数（二－十进制转换）

将一个二进制数转换为等值的十进制数，只要将二进制数按式（1-3）展开，即“按位按权展开求和”即可。

例如

$$\begin{aligned}(10110)_{(二)}&=1\times2^4+0\times2^3+1\times2^2+1\times2^1+0\times2^0\\&=16+0+4+2+0\\&=(22)_{(十)}\end{aligned}$$

$$\begin{aligned}(110.01)_{(二)}&=1\times2^2+1\times2^1+0\times2^0+0\times2^{-1}+1\times2^{-2}\\&=4+2+0+0.5+0.25\\&=(6.75)_{(十)}\end{aligned}$$

2. 十进制转换为二进制（十－二转换）

十进制数转换为二进制数、应将整数部分和小数部分分开进行。

(1) 整数部分的转换。设十进制整数为 $S_{(十)}$，对应的二进制数为 $(K_nK_{n-1}\cdots\cdots K_0)_{(二)}$，则

$$\begin{aligned} S_{(十)} &= K_n\times 2^n+K_{n-1}\times 2^{n-1}+\cdots+K_1\times 2^1+K_0\times 2^0 \\ &= K_n\times 2^n+K_{n-1}\times 2^{n-1}+\cdots+K_1\cdot 2^1+K_0 \\ &= 2\,(K_n\times 2^{n-1}+K_{n-2}\times 2^{n-2}+\cdots+K_1)\ +K_0 \end{aligned}$$

将上式两边同除以 2，两边得到的商和余数必然对应相等，即

$$\frac{S_{(十)}}{2}=\underbrace{(K_n\times 2^{n-1}+K_{n-1}\times 2^{n-2}+\cdots+K_1)}_{商}\cdots\cdots\cdots\cdots\underbrace{K_0}_{余数} \tag{1-6}$$

显然，只要将十进制数除以 2 得到的余数即为 K_0。若将得到的商再除以 2，得到的余数为 K_1。因此可以用这种“除 2 求余”的方法将十进制数连续除以 2，便可依次得到对应二进制数的每位系数 K_0，K_1，……。

例如，将 $(23)_{(十)}$ 化为二进制数

$$\begin{array}{l} 2\,\underline{|\,23} \\ \quad 2\,\underline{|\,11}\ \cdots\cdots 余数为\ 1\cdots\cdots K_0=1 \\ \qquad 2\,\underline{|\,5}\ \cdots\cdots 余数为\ 1\cdots\cdots K_1=1 \\ \qquad\quad 2\,\underline{|\,2}\ \cdots\cdots 余数为\ 1\cdots\cdots K_2=1 \\ \qquad\qquad 2\,\underline{|\,1}\ \cdots\cdots 余数为\ 0\cdots\cdots K_3=0 \\ \qquad\qquad\qquad 0\ \ \cdots\cdots 余数为\ 1\cdots\cdots K_4=1 \end{array}$$

因此，　$(23)_{(十)}=(10111)_{(二)}$

(2) 小数部分的转换。设十进制小数为 $S_{(十)}$，对应的二进制小数为 $(0.\ K_{-1}K_{-2}K_{-3}\cdots)_{(二)}$，则

$$S_{(十)}=K_{-1}\times 2^{-1}+K_{-2}\times 2^{-2}+\cdots+K_{-m}\times 2^{-m}$$

若将上式两边同乘以 2，则有

$$2\times S_{(十)}=\underbrace{K_{-1}}_{整数部分}+\underbrace{K_{-2}\times 2^{-1}+K_{-3}\times 2^{-2}+\cdots+K_{-m}\times 2^{-m+1}}_{小数部分} \tag{1-7}$$

因此，将十进制小数乘 2，得到的整数部分即为 K_{-1}，显然，将上式剩余的小数部分再乘 2，得到的整数部分应为 K_{-2}，依此类推，便可用这种“乘 2 求整”的方法依次求出十进制小数所对应的二进制小数的每一位系数。

例如，将 $(0.625)_{(十)}$ 化为二进制数

$$\begin{array}{lr l} & 0.625 & \\ & \times\qquad 2 & \\ \hline & 1.250 & \cdots\cdots\cdots\cdots 整数部分为\ 1=K_{-1} \\ 剩余小数 & 0.25 & \\ & \times\qquad 2 & \\ \hline & 0.5 & \cdots\cdots\cdots\cdots 整数部分为\ 0=K_{-2} \\ 剩余小数 & 0.5 & \\ & \times\qquad 2 & \\ \hline & 1.0 & \cdots\cdots\cdots\cdots 整数部分为\ 1=K_{-3} \end{array}$$

所以　$(0.625)_{(十)}=(0.101)_{(二)}$。

3. 二进制数转换为八进制数（二—八转换）

三位二进制数共有八个状态，即 000～111 对应为 0～7，再加 1 则向高位进位，因此三位二进制数恰好相当于一个一位八进制数。于是将二进制数化为八进制数时，整数部分可以从 2^0 位开始，依次将每三位二进制数划为一组，每组写成一位八进制数，小数部分则从 2^{-1}位开始，每三位写成一个八进制数即可。

例如，将 $(100110.111001)_{(二)}$化为八进制数，

$$(\underbrace{100}_{4}\ \underbrace{110}_{6}\cdot\underbrace{111}_{7}\ \underbrace{001}_{1})_{(二)} = (46.71)_{(八)}$$

4. 八进制转换为二进制（八－二转换）

八－二转换与二－八转换相反，只要按原顺序将每一位八进制数写为三位二进制数即可，

例如将 $(37.25)_{(八)}$化为二进制数

$$(37.25)_{(八)} = (\underline{011}\ \underline{111}.\ \underline{010}\ \underline{101})_{(二)}$$
$$= (11111.\ 010101)_{(二)}$$

5. 二进制转换为十六进制（二－十六转换）

将四位二进制数看成一位十六进制数时，它向高位的进位，正好是“逢十六进一”。因此可将二进制数从小数点开始，将整数和小数部分分别每四位写成一位十六进制数即可。

例如　将 $(1110\ 0010.\ 1101\ 0011)_{(二)}$化为十六进制数。

$$(\underbrace{1110}_{E}\ \underbrace{0010}_{2}\cdot\underbrace{1101}_{D}\ \underbrace{0011}_{3})_{(二)} = (E2.\ D3)_{(十六)}$$

6. 十六进制数转换为二进制（十六－二转换）

只要将十六进制中的每位对应写为四位二进制数即可。

例如　将 $(4EC)_{(十六)}$化为二进制数

$$\frac{(4}{0100}\ \frac{E}{1110}\ \frac{C)}{1100}_{(十六)} = (100\ 1110\ 1100)_{(二)}$$

三、二进制数的运算

1. 二进制加法

规则　0＋0＝0

0＋1＝1＋0＝1

1＋1＝0　　进位 1

1＋1＋1＝1　进位 1

例如　1101＋1011 的具体过程如下

被加数	1 1 0 1
加　数	1 0 1 1
进　位	＋1 1 1 1
	1 1 0 0 0

2. 二进制减法

规则　0－0＝0

$1-1=0$

$1-0=1$

$0-1=1$ 有借位 1

例如 1100－0101 具体过程如下

$$
\begin{array}{lr}
\text{被减数} & 1\ 1\ 0\ 0 \\
\text{减　数} & -0\ 1\ 0\ 1 \\
\text{借　位} & \underline{1\ 1\ 1\ \ } \\
 & 0\ 1\ 1\ 1
\end{array}
$$

3．二进制乘法

规则 $0\times0=0$

$0\times1=0$

$1\times0=0$

$1\times1=1$

例如 1111×1101 具体过程如下

$$
\begin{array}{lr}
\text{被乘数} & 1\ 1\ 1\ 1 \\
\text{乘　数} & \times\quad 1\ 1\ 0\ 1 \\
\hline
 & 1\ 1\ 1\ 1 \\
 & 0\ 0\ 0\ 0\ \ \ \\
 & 1\ 1\ 1\ 1\ \ \ \ \ \ \ \\
 & +)\ 1\ 1\ 1\ 1\ \ \ \ \ \ \ \ \ \ \ \\
\hline
 & 1\ 1\ 0\ 0\ 0\ 0\ 1\ 1
\end{array}
$$

上述过程与十进制乘法完全类似，不难看出，若乘数位为 1，乘得的中间结果（又称部分积）即与被乘数相同；若乘数位为 0，乘得的中间结果为 0。显然，只要判断一下乘数位，就可以知道该位乘得的中间结果，最后将各中间结果相加。这样就可以用加法器来实现乘法运算。

计算机中的乘法正是转变为加法来实现的，因而使计算机中的运算器大为简化。

常用的方法有乘数左移和部分积右移两种。现在简单介绍一下部分积右移的方法。

乘数	被乘数	部分积
1101	1111	0000 |
①乘数最低位为 1，加被乘数		+1111 |
		1111 |
部分积右移一位		0111 | 1
②第 2 位乘数为 0，不加被乘数部分积右移一位		0011 | 11
③第 3 位乘数为 1，加被乘数		+ 1111 |
部分积右移一位		10010 | 11
		1001 | 011
④第 4 位乘数为 1，加被乘数		+ 1111 |
		11000 | 011
部分积右移一位		1100 | 0011

在这种方法中，若相乘的数为n位，部分积的相加也只有n位，只需n个加法器和移位电路便可进行乘法，这在计算机中很容易实现。

4. 二进制除法

除法是乘法的逆运算，运算过程与十进制除法完全类似。

例如　　　　1001 110÷110=1101

$$
\begin{array}{r}
1101 \\
110\overline{)1001110} \\
\underline{110} \\
111 \\
\underline{110} \\
110 \\
\underline{110} \\
0
\end{array}
$$

四、码制

1. 代码

在二进制数字系统中，每一位二进制数只有0和1两个数码，因此只能表示两种不同的信号，为了能用二进制数码表示更多的信号，常把0和1按一定的规律编排在一起，从而构成各种不同的0，1的组合，称之为代码。

根据代码构成的规律不同，可以有若干种代码，常用的有：二一十进制代码，循环码 *ISO* 码，*ASCII* 码等。

2. 二一十进制代码

用四位二值代码（0，1）表示一位十进制数0～9这十个状态，得到的代码称为二一十进制代码，即BCD（Banary-Coded Decimal）码。由于四位二值代码可以有十六种不同的状态，所以任取其中十种，并按不同的状态排列，能够组成多种不同的代码。

8421码是最常用的二一十进制代码之一。它从左至右的每一位上的1分别代表8、4、2、1，这就是对应位的位权，因此称之为8421码。它属于有权码，即每一位的权是固定不变的。在8421码中，把每一代码中各位的1代表的权加起来，得到的结果就是它所表示的十进制数。

有权代码还有2421（*A*），2421（*B*），5211码等，只是每位的权不同而已。常用的二一十进制代码见表1-2。

表1-2　　　　**常用的二-十进制编码**

编码种类 / 十进制数	8421码	余3码	2421（*A*）码	2421（*B*）码	5211码	余3循环码	步进码
0	0000	0011	0000	0000	0000	0010	00000
1	0001	0100	0001	0001	0001	0110	10000
2	0010	0101	0010	0010	0100	0111	11000
3	0011	0110	0011	0011	0101	0101	11100
4	0100	0111	0100	0100	0111	0100	11110
5	0101	1000	0101	1011	1000	1100	11111
6	0110	1001	0110	1100	1001	1101	01111

续表

十进制数 \ 编码种类	8421码	余3码	2421(A)码	2421(B)码	5211码	余3循环码	步进码
7	0111	1010	0111	1101	1100	1111	00111
8	1000	1011	1110	1110	1101	1110	00011
9	1001	1100	1111	1111	1111	1010	00001
权	8421		2421	2421	5211		

3. 循环码

循环码是格雷码（Gray Code）中最常用的一种。在格雷码中，相邻两个代码之间只有一位状态不同。格雷码的形式又有多种。四位循环码见表 1-3。

表1-3　　四位循环码编码表

十进制数	循环码	十进制数	循环码
0	0000	8	1100
1	0001	9	1101
2	0011	10	1111
3	0010	11	1110
4	0110	12	1010
5	0111	13	1011
6	0101	14	1001
7	0100	15	1000

循环码中，每一位的 1 并不表示固定的数值，所以是一种变权码。

4. 奇偶校验编码

电子计算机在传输数据的过程中，有时会发生错误，对这样的错误，可利用许多编码方式加以发现，即附加一个校验位来进行检测。其中最简单最广泛使用的方法是奇偶法。这种方法一般是通过在编码信息上附加一个校验位来实现的。它分两种类型：奇校验和偶校验。若某位为 1，认为有信息；若为 0，认为无信息，则

奇校验-附加校验位＋原信息位

　　　＝奇数个信息位

偶校验-附加校验位＋原信息位

　　　＝偶数个信息位

表 1-4 列出 10 个 BCD 码的奇偶校验附加校验位的取值。

表1-4　　8421 编码的奇、偶校验

十进制数	BCD 编码 8421	奇偶校验位	
		奇校验	偶校验
0	0000	1	0
1	0001	0	1
2	0010	0	1
3	0011	1	0
4	0100	0	1
5	0101	1	0
6	0110	1	0
7	0111	0	1
8	1000	0	1
9	1001	1	0