

5.1 数制、码制和逻辑运算



5.1.1 数 制

1. 二进制数

二进制数的 1 位 (bit) 只能取 0 和 1 这两个不同的值，可以用来表示开关量 (或称数字量) 的两种不同的状态，例如触点的断开和接通、线圈的通电和断电等。

在编程软件中，位编程元件的 1 状态和 0 状态用 TRUE 和 FALSE 来表示。

计算机和 PLC 用多位二进制数来表示数字，二进制数基数为 2，遵循逢二进一的运算规则，从右往左的第 n 位 (最低位为第 0 位) 的权值为 2^n 。

用位置记数法表示为：

$$\begin{aligned} N &= (B_{n-1} 2^{n-1} + \dots + B_1 2^1 + B_0 2^0 + B_{-1} 2^{-1} \dots B_{-m} 2^{-m}) \\ &= \sum_{i=m}^{n-1} (D_i 2^i) \end{aligned}$$

二进制常数以 2# 开始， 用下式计算 2#1100 对应的十进制数：

$$1 \blacklozenge 2^3 + 1 \blacklozenge 2^2 + 0 \blacklozenge 2^1 + 0 \blacklozenge 2^0 = 12$$

2. 十六进制

多位二进制数的书写和阅读很不方便。为了解决这一问题，可以用十六进制数来取代二进制数，二进制整数从最低位（小数从最高位）起，每四位用一位十六进制数表示。十六进制数的 16 个数字是 0~9 和 A~F（对应于十进制数 10~15）。B#16#、W#16# 和 DW#16# 分别用来表示十六进制字节、字和双字常数，例如 W#16#13AF。在数字后面加“H”也可以表示十六进制数，例如：

$$B3H = 1011\ 0011B$$

$$1FB.02H = 0001\ 1111\ 1011.0000\ 0010B$$

$$F3ADH = 1111\ 0011\ 1010\ 1101B$$

3. 数制转换

1) 十进制 ∇ 二进制

整数部分：除基取余法——整数连续用基数 2 除，取各次余数，直到商为零。

小数部分：乘基取整法——小数连续乘基数 2，每次取整数部分，直到小数为零；若小数乘 2 无法使尾数为零，则可根据精度要求求出足够位数。

S7-1200 PLC 编程基础知识

例：分别将 237 和 0.6875 化为二进制数，其过程如下。

2	237	余数	低位		0.6875		高位
2	118	...	1=k ₀		×	2	整数
2	59	...	0=k ₁			1.3750	...
2	29	...	1=k ₂			0.3750	1=k ₁
2	14	...	1=k ₃		×	2	
2	7	...	0=k ₄			0.7500	...
2	3	...	0=k ₅		×	2	0=k ₂
2	1	...	1=k ₆			1.5000	...
0	...	1=k ₇	高位			0.5000	0=k ₃
					×	2	
						1.0000	...
						0=k ₄	低位

其结果是 (237)₁₀ =(1110,1101)₂ , (0.6875)₁₀ =(0.1011)

S7-1200 PLC 编程基础知识

2) 十进制 ∇ 十六进制

整数部分：除基取余法——整数连续用基数 16 除，取各次余数，直到商为零。

小数部分：乘基取整法——小数连续乘基数 16，每次取整数部分，直到小数为零；若小数乘 16 无法使尾数为零，则可根据精度要求求出足够位数。

例：分别将 237 和 0.5429 化为十六进制数。

$ \begin{array}{r} 16 \overline{) 237} \\ 16 \overline{) 14} \dots \\ \quad 0 \dots \end{array} $	余数 13(D)=k_0 14(E)=k_1	低位 ↑ 高位	$ \begin{array}{r} 0.5429 \\ \times 16 \\ \hline 8.6864 \dots \\ 0.6864 \\ \times 16 \\ \hline 10.9824 \dots \\ 0.9824 \\ \times 16 \\ \hline 15.7184 \dots \end{array} $	整数 8=k_1 10(A)=k_2 15(F)=k_3	高位 ↓ 低位
---	---	------------------------------	--	---	------------------------------

其结果是 $(237)_{10} = (ED)_{16}$ ， $(0.5429)_{10} = (0.8AF)_{16}$

3) 十六进制 ∇ 十进制

整数部分：按权展开法

$$\begin{aligned}\text{例： } 3D7BH &= 3 \times 16^3 + 13 \times 16^2 + 7 \times 16^1 + 11 \times 16^0 \\ &= 3 \times 4096 + 13 \times 256 + 7 \times 16 + 11 \times 1 \\ &= 12288 + 3328 + 112 + 11 \\ &= 15739\end{aligned}$$

小数部分：一般不用；使用时可化成二进制小数再作换算。

二进制 ∇ 十进制：一般二进制先换成十六进制，再转换成十进制。

十进制 ∇ 十六进制：一般十进制先转换成二进制，再转换成十六进制。

S7-1200 PLC 编程基础知识

表 5.1 给出了不同进制数的表示方法。

表 5.1 给出了不同进制数的表示方

十进制 数	十六进 制数	二进制 数	BCD 码	十进制 数	十六进制 数	二进制 数	BCD 码
0	0	00000	0000 0000	9	9	01001	0000 1001
1	1	00001	0000 0001	10	A	01010	0001 0000
2	2	00010	0000 0010	11	B	01011	0001 0001
3	3	00011	0000 0011	12	C	01100	0001 0010
4	4	00100	0000 0100	13	D	01101	0001 0011
5	5	00101	0000 0101	14	E	01110	0001 0100
6	6	00110	0000 0110	15	F	01111	0001 0101
7	7	00111	0000 0111	16	10	10000	0001 0110
8	8	01000	0000 1000	17	11	10001	0001 0111

5.1.2 码制

带符号数的表示方法：二进制数最高位表示数符，其余位表示数值。

最高位 0 : + 1 : —

例： 00000100 表示 +4 10000011 表示 -3

1. 原码

尾数部分直接表示数值本身绝对值：此称原码表示法。

表达式：

$$[X]_{\text{原}} = \begin{matrix} x & (x \geq 0) \\ 2^{n-1} - x & (x < 0) \end{matrix}$$

例： $n=8$ ， $x=+4$ ， 则 $[x]_{\text{原}} = 00000100$

$n=8$ ， $x=-3$ ， 则 $[x]_{\text{原}} = 2^8 - (-00000011) = 10000011$

存在二种表示方式：

$[x]_{\text{原}} = 00000000$ 或 $x=+0$

$[x]_{\text{原}} = 10000000$ 或 $x=-0$

2. 补码

二数相减，用电路实现减法，结构极为复杂。实际实现时，用加法器就可完成减法。加法器容易制作，故运算更加方便。用 8 位二进制数表示一个带符号数，最高位是符号位，剩下 7 位表示数值。

当 $x \geq 0$ 时， $[x]$ 补是 x 的本身值，最高位为 0；

当 $x < 0$ 时， $[x]$ 补 $= 2^8 + x$

例：① $x = +0000101B$ $[x]$ 补 $= 00000101B$

② $x = -0000101B$ $[x]$ 补 $= 100000000B + (-0000101B) = 11111011B$

对于负数 x ，求 $[x]$ 补的简单方法：绝对值变反加 1，再在最高

S7-1200 PLC 编程基础知识

例：二进制补码的简明方法。

设 $y = -0000010B$ （-2 十） $[y]_{补} = 11111101B$ ，再加 1，得：

11111110B

设 $y = -0000110B$ （-6 十） $[y]_{补} = 11111010B$

已知 $[X]_{补}$ ，求 X 。 正数： X 即 $[X]_{补}$ ，只是最高位 0 变成 +；

负数：数值变反加 1，再添负号（-）。

例：已知 $[X]_{补} = 11111010B$

首先，数值变反：0000101B，然后加 1：10000110B，最后添加符号：-

0000110B

例：计算 $14-32=14+(-32)=-18$

$x=14=+0001110B$ $[x]$ 补 $=00001110B$

$y=-32=-0100000B$ $[y]$ 补 $=11100000B$

$[x+y]$ 补 $=[x]$ 补 $+ [y]$ 补 $=00001110B+11100000B=11101110B$

$x+y=-0010010B=- (16+2) =-18$

对 n 位字长的二进制数，最高位仍为符号位。 -128 是特殊情况

：最高位既是数符位又是数值位。 8 位二进制数可表示带符号数数值范围为：

$00000000B \sim 01111111B$

$10000000B \sim 11111111B$

3. BCD 码

实际应用中一般计算问题的原始数据大多是十进制数，人们为计算机设计了一种用二进制数为它编码，该编码称 BCD 码

（ Binary Coded Decimal ）。BCD 是二进制编码的十进制数的缩写，BCD 码用四位二进制数表示一位十进制数（见表 5.1）

一个字节的 BCD 码可表示数值范围为 $0 \sim 99$ ，16 位 BCD 码的范围为 $-999 \sim +999$ ，32 位 BCD 码的范围为 $-99999999 \sim +99999999$ 。

S7-1200 PLC 编程基础知识

BCD 码运算时，每半个字节结果不超过 9 不修正，超过 9 则加 6

例：0001 0101 13+

+) 0010 0011 23+

0011 1000 38+

0100 1000 48+

+) 0110 1001 69+

1011 0001 ← 低半字节有进位，修正

↑ 高半字节 > 9，修正

+) 0110 0110

0001 0111 117+ CY ← 1

注意 BCD 码处理 一般均为正数。

S7-1200 PLC 编程基础知识

5.1.3 逻辑运算

在数字量（或称开关量）控制系统中，变量仅有两种相反的工作状态，例如高电平和低电平、继电器线圈的通电和断电，可以分别用逻辑代数中的 1 和 0 来表示这些状态，在波形图中，用高电平表示 1 状态，用低电平表示 0 状态。使用数字电路或 PLC 的梯形图都可以实现数字逻辑运算，图 5.1 的上面是 PLC 的梯形图，下面是对应得数字门电路。

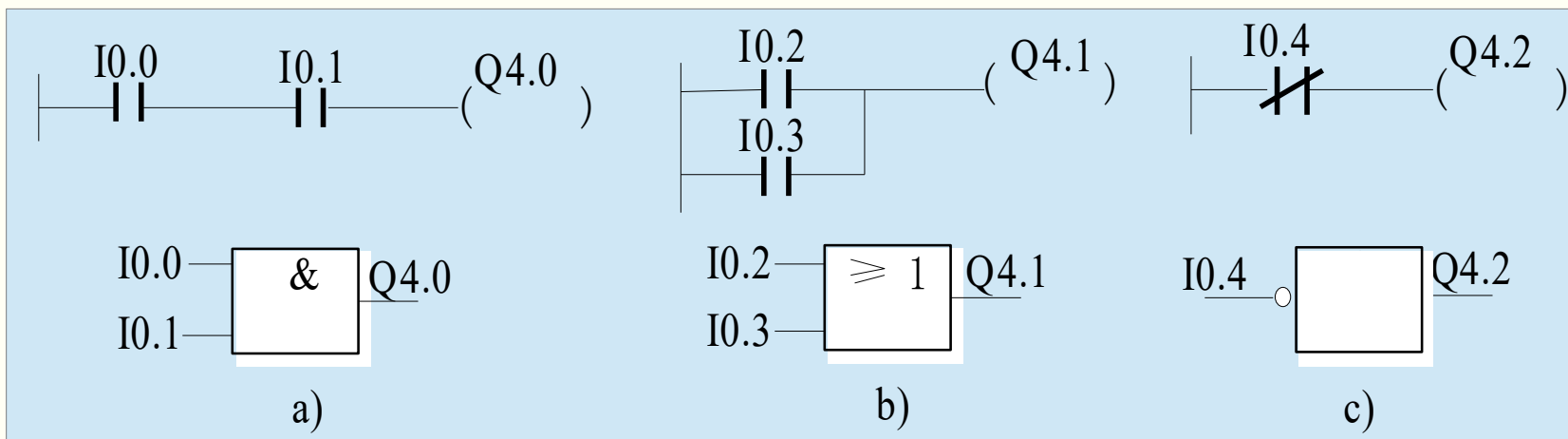


图 5.1 基本逻辑运算

S7-1200 PLC 编程基础知识

图 5.1 中的 I0.0~I0.4 为数字输入变量， Q4.0~Q4.2 为数字量输出变
它们之间的“与”、“或”、“非”逻辑运算关系如表 5.2 所示。

表 5.2 逻辑运算关

与			或			非	
Q4.0 = I0.0 · I0.1			Q4.1 = I0.2 + I0.3			Q4.2 = $\overline{I0.4}$	
I0.	I0.1	Q4.0	I0.2	I0.3	Q4.1	I0.4	Q4.2
0	0	0	0	0	0	0	1
0	1	0	0	1	1	1	0
1	0	0	1	0	1		
1	1	1	1	1	1		

图 5.2 中继电器电路实现的逻辑运算可以用逻辑代数表达式表示为

KM=(SB1+KM) $\overline{SB2}$ $\overline{FR}$

S7-1200 PLC 编程基础知识

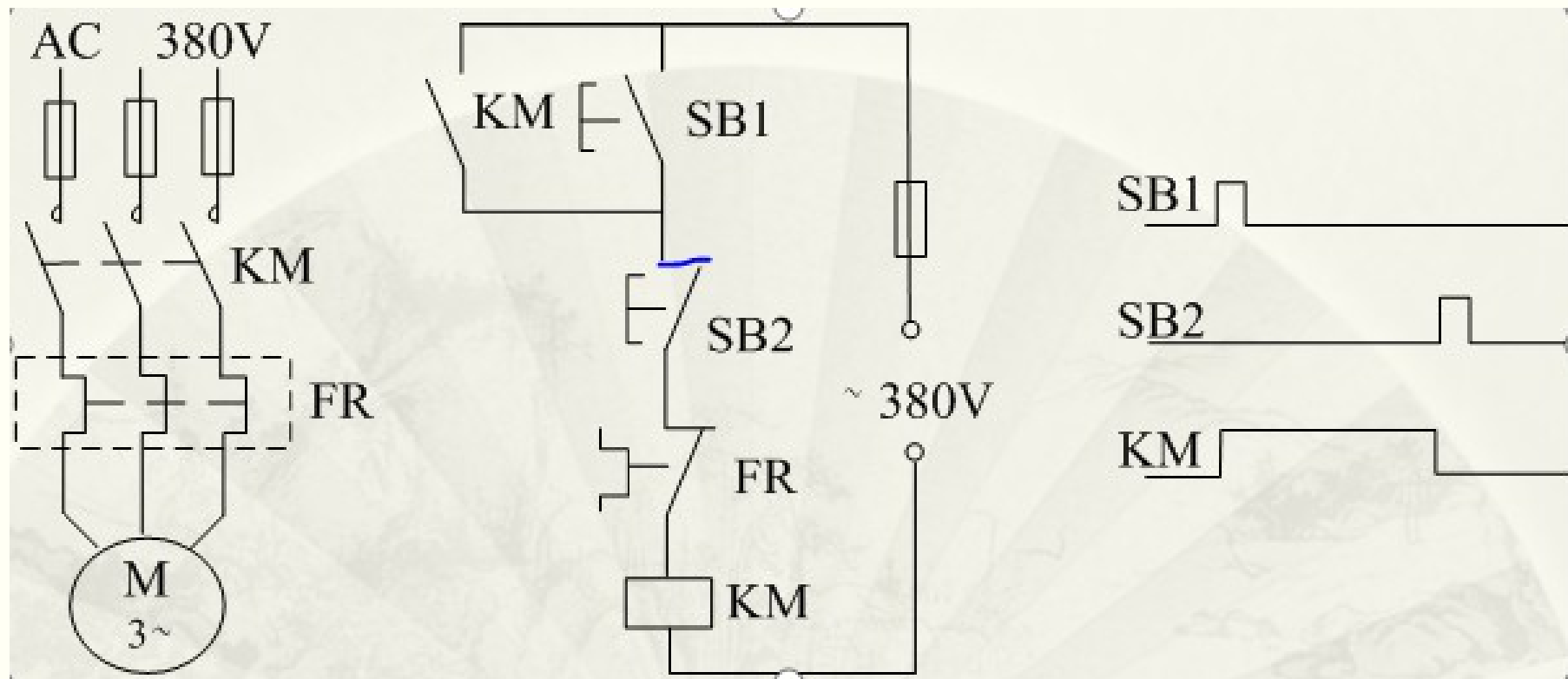


图 5.2 继电器控制电路

谢谢聆听