

微型计算机 在工业中的应用

美 斯特克哈恩 著
登 奥特

韩光翰 译
刘德辉

19

<1

电子工业出版社

74 39

微型计算机在工业中的应用

〔美〕 斯特克哈恩
登 奥特

韩光翰 刘德辉 译
乐毅 武晨 校



电子工业出版社

内 容 提 要

本书共分十个部分,从基本的数制讲起,直到能构成自己所需要的系统,并专门介绍了在连接系统的各部分中可能遇到的诸如:接口、噪声、接地等一般教科书中涉及不到的问题。书中围绕MC6800、MC6805E2和可编程控制器Modicon484的原理,列举了大量的针对工业环境中的具体问题的实例。对从事微型计算机控制系统工作的读者确是一本难得的教科书和实践指导书。特别是本书所讲的基本原理和方法很容易推广到其它型号的微型计算机系统中去。

微型计算机在工业中的应用

【美】斯特克哈恩 登 奥特

韩光翰 刘德辉 译

乐 毅 校

责任编辑 林 培

电子工业出版社出版(北京海淀区万寿路)

新华书店北京发行所发行 各地新华书店经售

山东电子工业印刷厂印刷(淄博市周村)

开本: 850×1168毫米 1/32 印张: 14.75 字数: 354千字

1988年10月第一版 1988年10月第一次印刷

印数: 1—4000册 定价: 5.20元

ISBN 7-5052-2137-1/TN·66

序 言

本书可供从事微处理机控制技术的学生、技术人员和工程师使用。书中讨论了用电机控制的微处理机的构成、程序设计和接口等各个方面。然而，任何一本书都不可能涉及存在的所有问题。但是，如果以一个在理论上成熟，实际应用成功的具体问题为例，会使我们很容易进入微处理机控制这样一个高深的领域。

在本书中介绍的大多数资料，在最近三年已经由埃德蒙顿北方艾伯塔技术学会的著者向来自工业界的人们阐述和讲授过。在环绕埃德蒙顿的地区有一个大的石油化工工业，就利用了微处理机和以微处理机为基础的可编程序控制器领域的最现代化的技术成就。

本书着重介绍 MC6800 和 MC6805E2 微处理机，以及 Medicon 484 可编程序控制器。介绍的全部例子都是工业问题，这些问题基于对工业界学生的教学经验和直接工业方面的经验。书中有许多成熟的教学举例，作者认为这是学习这种技术的方法。本书从基本的数制开始，即十进制、二进制、八进制、二进制编码的十进制和十六进制，以及从一种数制向另一种数制的转换。

一个题目在第一次介绍不必全面阐述，因为这可能太复杂，而且可能导致混乱；然而，在介绍题目的时候要介绍足够的信息，以便使读者充分了解下一部分。

第二章介绍通过逻辑门和 JK 触发器进行的电机控制系统的设计。从简单的串联或并联系统开始，包括起一停电机控制电路的设计，而以状态图和状态简化技术结束。这一章对于在微处理机应用中的地址译码和控制信号传递所需要的逻辑接口设计来说，是非常基本的。

另外，介绍了故障保险的重要性。如果对于在这一章和下一章援引这个题目感到厌烦，则作者表示歉意；但是，在工业应用中不能过于强调安全。旧式的工业应用采用磁激励器件，它有固有的故障保险，但是固体器件有可能发生故障；因此，必须使一切努力与故障保险技术结合起来，而且必须研究停止该过程的方法。设备的控制必须经常维护。

第三章向读者介绍MC6800微处理机及其外围电路，如RAM和ROM，介绍用以说明微处理机工作方法的定时图；还详细介绍PIA(并行数据接口)，ACIA(串行数据接口)，PIM(可编程程序定时器组件)，以及该处理机的寻址方式。提供如何读指令系统的详细情况，以及简单的程序设计例子。关于变址寄存器及其应用的比较全面的详细情况，将在后面几章介绍。

第四章向读者介绍结构化程序设计及其利用的许多例子。这种技术对于编制可工作的、无错误的程序来说是最基本的。并且，再次提供大量证明是有效的例子。

第五章进一步详细地介绍MC6800，并说明如何建立一个微处理机控制系例。所研究的系统是一个最小系统，所执行的程序是很基本的，它只不过使LED闪光。从这点来说，本章是基本的。但是，一旦读者了解这些，更复杂的任务就在转角处。许多学生认为建立一个微处理机控制的控制系统，是他们力所不及的；实际并非如此，正如本章说明的那样。

第六章介绍MC6805E2微处理机芯片；之所以选择这个器件，是因为它实际上是独立的芯片。将处理机与CRT连接起来的电路与起动程序一道说明。这一章说明工厂管理人员很容易为他们的特定应用建立一个以微处理机为基础的过程控制系统。

第七章介绍成熟的例子；这些例子比前面各章介绍的更复杂，并可应用于两个处理机6800和6805。选择决不是包罗无遗的，选择这些例子用以向读者说明，复杂的程序可以转化为比较简单的问题。故障保险总是包含所有设计中的一个因素。此外，

还包括连接CRT和处理进行通信的例子。

第八章向读者介绍可编程序的控制器(一个以处理机为基础的系统),它包含允许控制逻辑以类似于继电器梯形图的形式插入的电路。这些控制器在六十年代末引入汽车工业,以避免在产品变化时控制系统的昂贵的和消耗时间的重新布线。这种控制器随着更高级的固体器件的出现而迅速发展,现在在各种工业中都使用。

第九章介绍可编程序控制器的进一步发展,包括同MC6800微处理机的实际接口。这种处理机或计算机可用于几个控制器的总控制,以及用于重要数据的存储。

第十章介绍关于控制器和微处理机之间的接口的重要概念。在这个题目没有适当理解的情况下,不可能有可靠的过程控制。因此,这一章与本书其它各章都有关系,虽然它可以单独讨论。关于固体输入和输出电路的情况及噪声的考虑已经由艾伦·布雷德利提供;见会刊1772-820-1。

虽然本书集中讨论MC6800和MC6805处理机、以及484控制器,但本书的资料也可以用于MC6802D5、以及艾伦·布雷德利PLC。

作者感谢莫托罗拉公司和古尔—莫迪科部门在本书编写过程中给予的协作和好意,相互间的协作使工作很愉快。我们感谢我们的妻子玛丽和西纳在本书编写过程中的克制、理解和帮助。

A·D·斯特克哈恩

J·登奥特

译者后记

四化建设的发展，迫切需要我国广大工业企业采用先进的科学技术对现有设备、工艺过程进行技术改造和革新，充分挖掘工业企业现有潜力，努力提高工作效率。毫无疑问，微计算机和控制用微处理机是工业企业技术改造的强有力的工具。

我们翻译的“INDUSTRIAL APPLICATIONS FOR MICROPROCESSORS”一书是为工科院校的学生、工程师和技术人员编写的教科书，该书介绍了在工业生产过程中使用美国MOTOROLA公司生产的MC6800微处理机，工业控制用单片微处理机MC6805E2以及可编程控制器Modicon484的原理与实例。从方案设计想到用微处理器实现，书中列举的实例都是针对工业环境中的具体问题，具有很强的实用性。

围绕着如何使用微处理机，读者可以从两个方面入手，希望系统地学习微处理机的读者可以从数制与编程序学起，直到能构成自己所需的实际系统；已具有计算机基础的读者可以直接参考本书提供的方法，根据自己的需要构成系统。

本书的特色就是强调在实际的工业环境中应用微处理机，为此，专门介绍了在连接系统的各部分中可能会遇到的诸如：接口、噪声、接地等一般教科书中涉及不到的问题。

本书对从事微计算机控制系统研制工作的读者确是一本难得的教科书和实践指导书。书中所讲的基本原理和方法也容易地推广到其它型号的微计算机系统中。本书的第一章至第五章由韩光翰同志翻译，第六章以后由刘德辉同志翻译，最后由乐毅同志和武晨同志校对。由于我们水平有限，翻译过程中如有错漏之处，恳请读者指教。

译者 1985年9月

目 录

第一章 数制	(1)
1.1 十进制数制	(1)
1.2 二进制数制	(1)
1.3 八进制数制	(2)
1.4 二进制编码的十进制(BCD)数制	(2)
1.5 十六进制数制	(3)
1.6 十进制转换	(4)
1.7 二进制算术运算	(6)
1.8 带符号和不带符号的二进制数	(9)
1.9 二进制编码的十进制	(11)
1.10 指导例子	(11)
问题	(13)
第二章 逻辑设计	(14)
2.1 引言	(14)
2.2 AND、OR、NOT	(15)
2.3 真值表	(16)
2.4 布尔假定	(18)
2.5 布尔定理	(18)
2.6 逻辑门	(25)
2.7 逻辑电路	(28)
2.8 工业实践	(33)
2.9 状态图	(35)
2.10 时序电路	(37)
2.11 状态简化	(40)

2.12 定时时序电路	(42)
2.13 计数器	(46)
2.14 指导例子	(47)
问题	(50)
第三章 微处理机	(52)
3.1 微处理机系统概述	(52)
3.2 MPU定时	(62)
3.3 MC6800寻址方式	(64)
3.4 MC6800的程序设计	(66)
3.5 可编程序的外围接口	(70)
3.6 指导例子	(122)
问题	(129)
第四章 结构化程序设计	(132)
4.1 一个结构化程序的基本结构	(133)
4.2 屏蔽	(141)
4.3 “做一当”和“重复一直到”结构结合的效果	(147)
4.4 变址寄存器	(148)
4.5 子程序程序设计	(151)
4.6 定时循环	(157)
4.7 指导例子	(158)
问题	(166)
第五章 MC6800 微处理机	(169)
5.1 6800MPU 及其控制	(169)
5.2 MC6800系统时钟	(172)
5.3 堆栈和堆栈指示器(SP)	(173)
5.4 中断	(176)
5.5 建立一个MC6800系统	(183)
5.6 指导例子	(192)
问题	(195)

第六章	MC6805E2 微处理机	(199)
6.1	6805MPU 及其控制	(199)
6.2	存储空间	(204)
6.3	地址总线	(205)
6.4	寄存器	(206)
6.5	寻址方式	(207)
6.6	6805指令集	(212)
6.7	时钟	(213)
6.8	定时器	(213)
6.9	I/O端口	(218)
6.10	堆栈与堆栈指针	(220)
6.11	中断	(222)
6.12	具有MC6805E2芯片的最小微处理机系统	(226)
6.13	教学举例	(231)
	问题	(236)
第七章	程序设计举例	(238)
7.1	实际应用举例	(239)
	问题	(287)
第八章	可编程控制器	(291)
8.1	处理机	(291)
8.2	输入/输出	(293)
8.3	程序设计	(294)
8.4	定时器和计数器	(294)
8.5	序列发生器	(296)
8.6	算术功能	(298)
8.7	传送	(298)
8.8	转换	(300)
8.9	电源故障	(301)
8.10	跨接和旁路 I/O	(303)

8.11 教学举例	(304)
问题	(310)
第九章 可编程控制器接口	(311)
9.1 协议	(315)
9.2 教学举例	(334)
问题	(337)
第十章 连接中的电源控制器, 微处理机, 噪声和接地	(338)
10.1 数据采集和转换	(339)
10.2 数据转换器与微处理机的接口	(358)
10.3 固态电路	(361)
10.4 噪音	(369)
10.5 接地	(374)
10.6 示范举例	(377)
问题	(386)
附录A 6800 指令集和摘要	(387)
附录B 可编程控制器	(412)
附录C 6805指令集和摘要	(420)
附录D FCC 规则与细则	(444)
附录E 十六进制和十进制转换	(446)
附录F 有符号的二进制数	(447)
名词汇编	(449)

第一章 数 制

十进制数制是我们日常生活的组成部分，我们的货币是十进制的，而且十进制或公制已家喻户晓。因此，很难想象通过其他数制进行计数和计算；但是，实际上已由全部数字计算机、微处理机实现了。这些设备正是通过二进制数制完成它们日常工作的。因此，这种数制及其有关的形式，即二进制编码的十进制(BCD)、八进制和十六进制，以及进制乘法和加法，全部在本章讨论。在本书中将只考虑整数。

1.1 十进制数制

十进制数制基于十个符号或数字：0、1、2、3、4、5、6、7、8、9。任何数都能用一组有序的数字表示，例如， 5053_{10} （下标表示该数制的基数）。这个数说明如下：

$$\begin{array}{cccc} 10^3 & 10^2 & 10^1 & 10^0 \\ 1000 & 100 & 10 & 1 \\ 5 & 0 & 5 & 3 \end{array} \quad (1.1)$$
$$= 5 \times 1000 + 0 \times 100 + 5 \times 10 + 3 \times 1$$

注意，每位数字的权，随着该数字组从右向左移而增加恒定的倍数(10倍)。我们讨论的全部数制都是基于这种位置加权制。

1.2 二进制数制

二进制数制是基于2个数字，0和1。任何数都可用这两个数字表示，例如， 1101_2 ；这个数可以解释如下：

$$\begin{array}{cccc}
 2^3 & 2^2 & 2^1 & 2^0 \\
 8 & 4 & 2 & 1 \\
 1 & 1 & 0 & 1
 \end{array} = 1 \times 8 + 1 \times 4 + 0 \times 2 + 1 \times 1 = 13_{10} \quad (1.2)$$

注意，每位数字的权，随着该数字从右向左移而增加恒定的倍数(2倍)。因此，数 13_{10} 是 1101_2 的十进制等值。

1.3 八进制数制

八进制数制是基于8个数字，0、1、2、3、4、5、6、7。任何一个数都可以用这八个数字表示，例如， 675_8 ，这个数可以看作：

$$\begin{array}{ccc}
 8^2 & 8^1 & 8^0 \\
 64 & 8 & 1 \\
 6 & 7 & 5
 \end{array} = 6 \times 64 + 7 \times 8 + 5 \times 1 = 445_{10} \quad (1.3)$$

注意，数字的权增加的方式与前面的例子相同；而且，很容易地转换成十进制形式，并用 $445_{10} = 675_8$ 表示。

二进制数用八进制数的形式表示如下：

4	2	1	4	2	1	4	2	1	二进制权	(1.4)
1	1	0	1	1	1	1	0	1	二进制表示	
6			7			5			八进制表示	

因此， $675_8 = 445_{10} = 110111101_2$ 。

1.4 二进制编码的十进制(BCD)数制

BCD数制是基于8421权的一种二进制数制，任何十进制数都能用这四个数字表示，例如：

8421	8421	8421	权	
1001	0111	0000	BCD表示	(1.5)
9	7	0	十进制表示	

因此, $970_{10} = 100101110000_2$ (BCD表示法)。

用BCD数制的缺点是它的最大数是1001, 因此, 下述代码被浪费:

二进制	十进制	二进制	十进制	
1010 =	10	1101 =	13	
1011 =	11	1110 =	14	(1.6)
1100 =	12	1111 =	15	

然而, 该数制广泛用于数字仪表。

1.5 十六进制数制

十六进制数制基于16个数字和符号, 0、1、2、3、4、5、6、7、8、9, A、B、C、D、E、F。该数制按下述方法利用8421加权制的全部组合。数字0~9分别代表二进制数0000~1001, 其余的二进制组合采取字母字符。

十进制	8421代码	十六进制代码	
10	1010	A	(1.7)
11	1011	B	
12	1100	C	
13	1101	D	
14	1110	E	
15	1111	F	

用十六进制表示法表示的数可能出现下述形式: $30F6_{16}$ 。这个数可用与前面例子相同的方法转换成二进制和十进制表示。

二进制转换:

8421	8421	8421	8421	权	
0011	0000	1111	0110	二进制表示	(1.8)
3	0	F	6	十六进制表示	

因此, $30F6_{16} = 0011\ 0000\ 1111\ 0110_2$ 。

十进制转换:

$$16^3 \quad 16^2 \quad 16^1 \quad 16^0 \quad (1.9)$$

$$4096 \quad 256 \quad 16 \quad 1$$

$$3 \quad 0 \quad F \quad 6 = 3 \times 4096 + 0 \times 256 + F \times 16 + 6 \times 1 = 12534_{10}$$

(记住 $F = 15$)

因此, $12534_{10} = 30F6_{16} = 0011\ 0000\ 1111\ 0110_2$ 。

前面介绍的二进制数制是计算机和微处理机进行各种计算的基础。另外, 前面讨论的各种数制及数制间的转换是很重要的, 读者应该掌握。

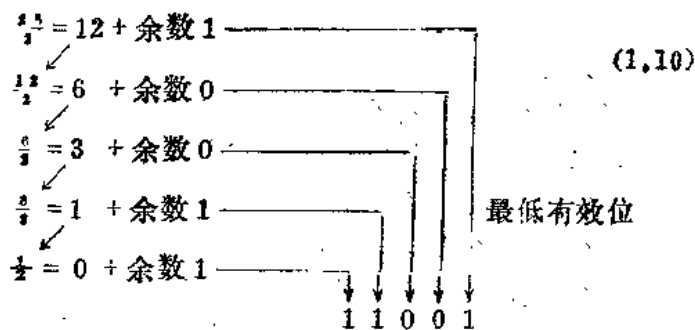
从每一种数制向十进制数制转换的方法已经说明了; 而从十进制向二进制等数制的转换是比较简单的, 下面进行介绍。

1.6 十进制转换

十进制数向二进制数或向与二进制相关数制的转换, 可以象方程(1.10)所表示的那样进行。

例 1 将 25_{10} 转换成二进制形式。

将该十进制数除以 2, 并记录余数, 如下所示:

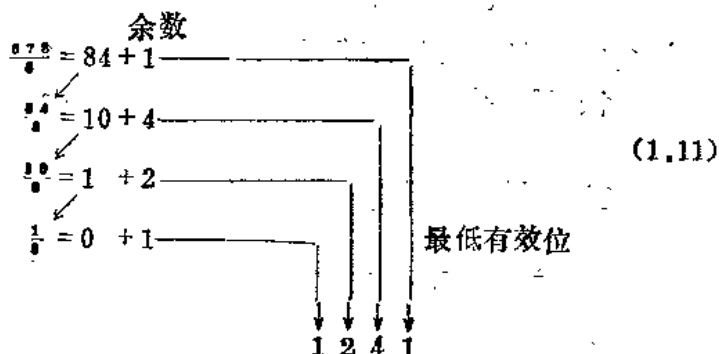


这可以用 8 位形式写成 00011001_2 。因此, $25_{10} = 00011001_2$ 。

读者现在应该练习一下将这个二进制数转换成原来的十进制数。同样,我们可以从十进制向八进制代码转换,如下所述。

例 2 将 673_{10} 转换成八进制形式。

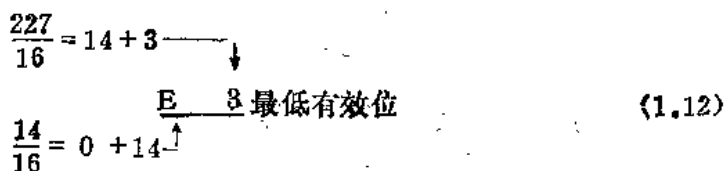
将该十进制数除以 8 并记录余数,如下所示:



所得到的八进制数是 1241_8 。

十进制数向十六进制转换的方法与例 1 和例 2 所示的方法相同,只是除数是 16。

例 3 将 227_{10} 转换成十六进制数。



所得的十六进制数是E3₁₆，建议读者练习将这个结果和例2的结果转换成原来的十进制形式。

读者很自然会问，为什么微处理机以二进制表示法工作呢？这是因为，二进制数制可以表示仅有二个状态的任何系统，例如电信号，它或是存在的（比如对地或对参考线为+5伏），或者是不存在的（比如为0伏）。因此，如果某条线上的电压为高电平，+5伏（对地），则该线的状态可以用二进制1来表示；如果该线为0伏（对地），则该线的状态可用二进制0表示。选择5伏作为高电平电压，是因为微处理机通常用5伏电源工作；因此，电信号的两个电平是比较容易检测的。但是，在0和5伏之间的哪一点是0变成1呢？下面是一般标准：

$$0 < \text{二进制} 0 < 1.5\text{V} \quad (1.13)$$

$$3.5\text{V} < \text{二进制} 1 \leq 5\text{V}$$

在1.5和3.5伏之间的电压是灰色区，构成微处理机的电子器件的响应在这个区域是不能预料的，因此可能导致误操作。

1.7 二进制算术运算

在本书中仅讨论两种形式的算术运算，乘法和加法。这是因为电子逻辑电路和微处理机基本上仅利用这两个概念。逻辑设计和逻辑系统的详细情况将在第二章介绍。

二进制乘法与普通的算术运算的乘法相同：

$$1 \cdot 0 = 1 \times 0 = 0 \quad (1.14)$$

$$0 \cdot 1 = 0 \times 1 = 0$$

$$0 \cdot 0 = 0 \times 0 = 0$$

$$1 \cdot 1 = 1 \times 1 = 1$$

二进制加法，在二进制算术运算中，仅在一种情况下与普通的算术运算的加法不同，如：