

MCS-51系列单片机 实用子程序集锦

陈 伟 人



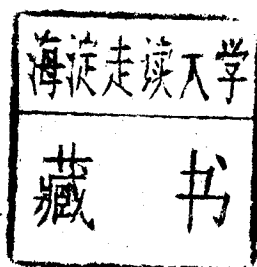
368.1
VR/1

清华大学出版社

TP368.1
CWR/1

MCS-51 系列单片机实用子程序集锦

陈伟人 编著



0023687
清华大学出版社

内 容 简 介

本书是编者近年来从事 MCS-51 单片机开发和应用的实际经验的总结。共有 160 多个实用子程序,包括科学计算、信息处理、查表技术、A/D、D/A、定时器、串行通信、键盘、显示和打印等,几乎涉及到了单片机应用的各个方面。每个子程序都有编制说明和程序框图,有助于读者理解;并注明入口条件和出口参数,便于读者直接调用。

本书作为一本实用的软件资料,可供从事单片机开发应用的广大科技工作者和大、中专学校师生参考;也可作为初学者自学单片机编程的一本入门指导书。

(京)新登字 158 号

JS283/09

MCS-51 系列单片机实用子程序集锦

陈 伟 人 编著

☆

清华大学出版社出版

北京·清华园

国防科工委印刷厂印刷

新华书店总店科技发行所发行

☆

开本: 787×1092 1/16 印张: 10.25 字数: 249 千字

1993 年 3 月第 1 版 1993 年 3 月第 1 次印刷

印数: 00001—10000

ISBN 7-302-01113-3/TP·420

定价: 7.00 元

前 言

近年来,随着超大规模集成电路的出现,单片微型计算机发展异常迅猛。在电子应用技术领域,如果说60年代面临的是晶体管技术;70年代面临的是集成电路技术;80年代面临的是单板机技术的话,那末,可以讲,单片机技术无疑将是90年代最活跃的新一代电子应用技术。在当前以应用电子技术为代表的高新技术改造传统工业方面,在机电一体化领域,单片机技术必将显示出强大的生命力。实际上,单片机以其功能强、体积小、使用方便、性能价格比高等许多优点,已经在实时控制系统、自动测试系统、智能仪器仪表、计算机终端、遥测通信、家用电器等许多方面得到了很好的应用。INTEL公司的MCS-51系列单片机则是国内应用最广泛的一种。

在单片机应用产品的研制过程中,软件的设计和调试往往是难度最大、工作量最多的一个部分,而对于不甚熟悉单片机的科技人员,要完成这部分的工作就更感困难了。编者在多次“单片机应用短训班”上介绍一些实用子程序,都深受学员的欢迎,并普遍要求更多的介绍。编者本人在开发科研项目时,也常常要借用一些现成的子程序。多年的实践体会,深感为了缩短单片机应用产品的开发周期,避免在应用软件设计中的重复劳动,有必要把多年实践中积累起来的子程序加以整理和系统化、标准化,作为一本软件资料以备工作中查阅。为了尽快推广、普及单片机应用技术,为科技人员在单片机软件开发过程中提供一条捷径;也应许多学员的迫切要求和热心关切,今将该软件资料整理编写成本书,希望能起到抛砖引玉的作用,求得相互交流、促进和提高的效果。。

全书分十三章,共有160多个子程序,内容包括二进制定点数/浮点数运算,BCD码运算,代码转换,数码变换,查表技术,信息处理,A/D、D/A转换,定时器,串行通信,键盘,显示,打印等,几乎涉及到单片机应用的所有方面。一方面,编者与广大单片机应用人员有着广泛的联系,深知各个层次的具体要求;另一方面,编者本人也一直在从事单片机科研项目和产品的研制开发,对于经常需要哪些子程序有自己的实际了解,因此,在本书内容的选取和编排上,做到层次清晰、注重实用、通俗易懂。为了有助于读者的理解,每个子程序都有编制说明和程序框图,并注明入口条件和出口参数,便于读者直接调用。

陈谦同志参加了部分程序的调试,并为整理书稿和绘制图稿做了大量工作,在此深表谢意。

编 者

1992年2月

目 录

第一章 代码转换	1	2.2 BCD 码移位	18
1.1 十六进制数到 ASCII 码的转换	1	2.2.1 程序一(双字节循环左移一位)	18
1.1.1 程序一(方法一)	1	2.2.2 程序二(双字节循环右移一位)	18
1.1.2 程序二(方法二)	1	2.2.3 程序三(多字节循环左移一位)	19
1.2 ASCII 码到十六进制数的转换	2	2.2.4 程序四(多字节循环右移一位)	19
1.2.1 程序一(方法一)	2	2.3 二进制定点数求补	20
1.2.2 程序二(方法二)	2	2.3.1 程序一(单字节)	20
1.3 二进制数到 BCD 码的转换	3	2.3.2 程序二(双字节)	20
1.3.1 程序一(单字节整数)	3	2.3.3 程序三(多字节)	20
1.3.2 程序二(双字节整数)	3	2.4 BCD 码求补	21
1.3.3 程序三(多字节整数)	4	2.4.1 程序一(单字节)	21
1.3.4 程序四(多字节小数)	5	2.4.2 程序二(双字节)	21
1.4 BCD 码到二进制数的转换	6	2.4.3 程序三(多字节)	21
1.4.1 程序一(单字节整数)	6	2.5 二进制浮点数左规	22
1.4.2 程序二(双字节整数)	7	2.5.1 程序一(双字节无符号数)	22
1.4.3 程序三(多字节整数)	7	2.5.2 程序二(多字节无符号数)	23
1.4.4 程序四(多字节小数)	9	2.5.3 程序三(双字节有符号数)	23
1.5 BCD 码到 ASCII 码的转换	10	2.5.4 程序四(多字节有符号数)	24
1.6 ASCII 码到 BCD 码的转换	10	2.6 二进制浮点数对阶	25
1.7 二进制数到格雷码的转换	11	2.6.1 程序一(双字节无符号数)	25
1.7.1 程序一(方法一)	11	2.6.2 程序二(多字节无符号数)	26
1.7.2 程序二(方法二)	12	2.6.3 程序三(双字节有符号数)	27
1.8 格雷码到二进制数的转换	12	2.6.4 程序四(多字节有符号数)	28
1.9 二进制定点数到浮点 数的转换	13	第三章 二进制定点数运算	30
1.9.1 程序一(单字节)	13	3.1 加法	30
1.9.2 程序二(双字节)	14	3.1.1 程序一(多字节无符号数)	30
1.9.3 程序三(多字节)	14	3.1.2 程序二(多字节有符号数)	30
第二章 数据变换	16	3.2 减法	31
2.1 二进制定点数移位	16	3.2.1 程序一(多字节无符号数)	31
2.1.1 程序一(双字节循环左移一位)	16	3.2.2 程序二(多字节有符号数)	32
2.1.2 程序二(双字节循环右移一位)	16	3.3 乘法	33
2.1.3 程序三(多字节循环左移一位)	17	3.3.1 程序一(双字节无符号数)	33
2.1.4 程序四(多字节循环右移一位)	17	3.3.2 程序二(多字节无符号数)	34
		3.3.3 程序三(单字节有符号数)	36
		3.3.4 程序四(双字节有符号数)	37

3.3.5 程序五(多字节有符号数)	37	5.3.2 程序二(单字节)	69
3.4 除法	39	5.3.3 程序三(双字节)	70
3.4.1 程序一(双字节无符号数)	39	5.3.4 程序四(多字节)	71
3.4.2 程序二(多字节无符号数)	40	5.4 除法	73
3.4.3 程序三(单字节有符号数)	42	5.4.1 程序一(单字节)	73
3.4.4 程序四(双字节有符号数)	43	5.4.2 程序二(双字节)	74
3.4.5 程序五(多字节有符号数)	44	5.4.3 程序三(多字节)	76
3.5 开方	45	5.5 开方	78
3.5.1 程序一(单字节)	45	5.5.1 程序一(单字节)	78
3.5.2 程序二(双字节)	46	5.5.2 程序二(双字节)	79
3.5.3 程序三(多字节)	47	5.5.3 程序三(多字节)	80
第四章 二进制浮点数运算	49	第六章 模-数和数-模转换	82
4.1 加法	49	6.1 单片机最小基本系统	82
4.1.1 程序一(双字节无符号数)	49	6.2 模-数转换	82
4.1.2 程序二(多字节无符号数)	50	6.2.1 程序一(单路 8 位 A/D 芯片 ADC0804)	82
4.1.3 程序三(双字节有符号数)	51	6.2.2 程序二(8 路 8 位 A/D 芯片 ADC0809)	83
4.1.4 程序四(多字节有符号数)	52	6.2.3 程序三(单路 12 位 A/D 芯片 AD574)	84
4.2 减法	53	6.2.4 程序四(单路 3 1/2 位 A/D 芯片 5G14433)	84
4.2.1 程序一(双字节无符号数)	53	6.2.5 程序五(单路 4 1/2 位 A/D 芯片 5G7135)	86
4.2.2 程序二(多字节无符号数)	54	6.3 数-模转换	87
4.2.3 程序三(双字节有符号数)	55	6.3.1 程序一(单路 8 位 D/A 芯片 AD1408)	87
4.2.4 程序四(多字节有符号数)	56	6.3.2 程序二(单路 8 位 D/A 芯片 DAC0832)	88
4.3 乘法	56	6.3.3 程序三(2 片单路 8 位 D/A 芯片 DAC0832)	89
4.3.1 程序一(双字节无符号数)	56	6.3.4 程序四(单路 10 位 D/A 芯片 AD7520)	89
4.3.2 程序二(多字节无符号数)	57	6.3.5 程序五(单路 12 位 D/A 芯片 DAC1210)	90
4.3.3 程序三(双字节有符号数)	59	第七章 定时器	91
4.3.4 程序四(多字节有符号数)	60	7.1 简单延时	91
4.4 除法	61	7.1.1 程序一(2ms 延时)	91
4.4.1 程序一(双字节无符号数)	61	7.1.2 程序二(0.5ms 延时)	91
4.4.2 程序二(多字节无符号数)	61	7.2 多重循环延时	92
4.4.3 程序三(双字节有符号数)	62	7.2.1 程序一(一重循环)	92
4.4.4 程序四(多字节有符号数)	63	7.2.2 程序二(二重循环)	93
第五章 BCD 码运算	65	7.2.3 程序三(三重循环)	94
5.1 加法	65		
5.1.1 程序一(单字节)	65		
5.1.2 程序二(双字节)	65		
5.1.3 程序三(多字节)	66		
5.2 减法	66		
5.2.1 程序一(单字节)	66		
5.2.2 程序二(双字节)	67		
5.2.3 程序三(多字节)	67		
5.3 乘法	68		
5.3.1 程序一(半字节)	68		

7.3 脉冲宽度测量	95	11.1.1 程序一(打印字符)	126
第八章 查表	96	11.1.2 程序二(打印数字)	128
8.1 自变量为单字节规则量	96	11.1.3 程序三(打印图形)	129
8.1.1 程序一(函数值为单字节)	96	11.2 LASER PP40 四色描绘器	130
8.1.2 程序二(函数值为双字节)	96	11.2.1 程序一(文本模式)	130
8.1.3 程序三(函数值为多字节)	97	11.2.2 程序二(图形模式)	131
8.2 自变量为双字节规则量	97	第十二章 信息处理	134
8.2.1 程序一(函数值为单字节)	97	12.1 数据移入	134
8.2.2 程序二(函数值为双字节)	98	12.1.1 程序一(单字节移入 片内 RAM 区)	134
8.2.3 程序三(函数值为多字节)	99	12.1.2 程序二(双字节移入 片内 RAM 区)	134
8.3 函数值为单字节规则量	100	12.1.3 程序三(单字节移入 扩展 RAM 区)	135
8.3.1 程序一(自变量为单字节)	100	12.1.4 程序四(双字节移入 扩展 RAM 区)	135
8.3.2 程序二(自变量为双字节)	101	12.1.5 程序五(多字节移入 扩展 RAM 区)	135
8.4 函数值为双字节规则量	102	12.2 数据块复制	136
8.4.1 程序一(自变量为单字节)	102	12.2.1 程序一(片内 RAM→ 片内 RAM)	136
8.4.2 程序二(自变量为双字节)	102	12.2.2 程序二(片内 RAM→ 扩展 RAM)	136
8.5 自变量和函数值均为 非规则量	103	12.2.3 程序三(扩展 RAM→ 扩展 RAM)	136
8.5.1 程序一(单字节)	103	12.3 顺序检索	137
8.5.2 程序二(双字节)	104	12.4 找最大数	138
第九章 串行通信	105	12.4.1 程序一(单字节无符号数)	138
9.1 奇偶校验	105	12.4.2 程序二(单字节有符号数)	138
9.2 双机通信	106	12.4.3 程序三(双字节无符号数)	139
9.3 多机通信	108	12.4.4 程序四(双字节有符号数)	139
第十章 键盘与显示	110	12.5 排序	140
10.1 键盘	110	12.5.1 程序一(单字节无符号数)	140
10.1.1 程序一(程控扫描)	110	12.5.2 程序二(单字节有符号数)	141
10.1.2 程序二(定时扫描)	112	12.5.3 程序三(双字节无符号数)	142
10.1.3 程序三(中断扫描)	114	12.5.4 程序四(双字节有符号数)	143
10.2 LED 显示	117	12.6 求平均值	144
10.2.1 程序一(BCD 码直接显示)	117	12.6.1 程序一(双字节无符号数)	144
10.2.2 程序二(串行口扩展显示)	118	12.6.2 程序二(双字节有符号数)	145
10.2.3 程序三(程控扫描之一: 字形代码并行输出)	119	12.6.3 程序三(单字节 BCD 码)	148
10.2.4 程序四(程控扫描之二: 字形代码串行输出)	120	12.6.4 程序四(双字节 BCD 码)	149
10.2.5 程序五(程控扫描之三: BCD 码译码输出)	121	第十三章 其它	151
10.3 CRT 显示	123	13.1 散转	151
10.3.1 程序一(字符显示)	123		
10.3.2 程序二(汉字显示)	125		
第十一章 打印	126		
11.1 LH-GP16 打印机	126		

13.1.1 程序一(在 2KB 范围内, 散转数为单字节)	151	13.1.4 程序四(在 64KB 范围内, 散转数为双字节)	152
13.1.2 程序二(在 64KB 范围内, 散转数为单字节)	151	13.1.5 程序五(直线式键盘散转)	153
13.1.3 程序三(在 2KB 范围内, 散转数为双字节)	152	13.2 软件计数器	153
		13.3 编辑器	154
		13.4 屏幕上字符跑动	156

第一章 代码转换

1.1 十六进制数到 ASCII 码的转换

1.1.1 程序一(方法一)

1. 编制说明

凡大于等于 10 的十六进制数加 37H, 凡小于 10 的十六进制数加 30H, 便得到相应的 ASCII 码。

(1) 入口: R_2 (高 4 位为 0000, 低 4 位为 0000~1111 的一个十六进制数 0~F)。

(2) 出口: R_2 (相应的 ASCII 码)。

2. 程序清单

```
HASC1:  MOV  A, R2
        ADD  A, *0F6H
        MOV  A, R2
        JNC  AD30H
        ADD  A, *07H
AD30H:  ADD  A, *30H
        MOV  R2, A
        RET
```

3. 程序框图

见图 1-1。

1.1.2 程序二(方法二)

1. 编制说明

凡大于等于 10 的十六进制数按 BCD 码加法加 31H, 凡小于 10 的十六进制数按 BCD 码加法加 30H, 便得到相应的 ASCII 码。

(1) 入口: R_2 (高 4 位为 0000, 低 4 位为 0000~1111 的一个十六进制数 0~F)。

(2) 出口: R_2 (相应的 ASCII 码)。

2. 程序清单

```
HASC2:  MOV  A, R2
        ADD  A, *90H
        DA   A
        ADDC A, *40H
        DA   A
        MOV  R2, A
        RET
```

3. 程序框图

见图 1-2。

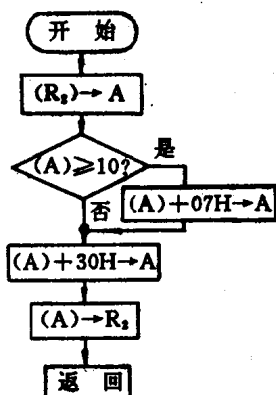


图 1-1

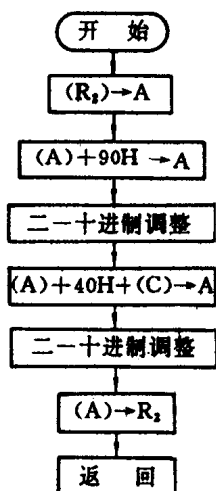


图 1-2

1.2 ASCII 码到十六进制数的转换

1.2.1 程序一(方法一)

1. 编制说明

若为 0~9 的 ASCII 码,则减去 30H;若为 A~F 的 ASCII 码,则减去 37H,便得到相应的十六进制数 0~F。

(1) 入口: R_2 (0~9 或 A~F 的 ASCII 码)。

(2) 出口: R_2 (高 4 位为 0000,低 4 位为 0000~1111)。

2. 程序清单

```

ASCH1:  MOV A,R2
        ADD A,*0D0H
        MOV R3,A
        ADD A,*0F6H
        JNC RET1
        MOV A,R2
        ADD A,*0F9H
        MOV R3,A
RET1:   RET
  
```

3. 程序框图

见图 1-3。

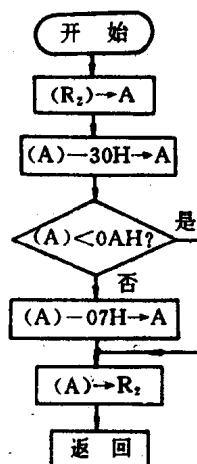


图 1-3

1.2.2 程序二(方法二)

1. 编制说明

原理、入口、出口、程序框图同方法一。

2. 程序清单

```

ASCH2:  MOV A,R1
        CLR C
        SUBB A,*30H
        MOV R2,A
        SUBB A,*0AH
        JC RET2
        MOV A,R2
        SUBB A,*07H
        MOV R2,A
RET2:   RET
    
```

1.3 二进制数到 BCD 码的转换

1.3.1 程序一(单字节整数)

1. 编制说明

将二进制数除以 100 得百位 BCD 码,余数除以 10 得十位 BCD 码,再得余数为个位 BCD 码。

(1) 入口: R_2 (8 位无符号二进制整数)。

(2) 出口: R_4 (百位)、 R_5 (十位、个位) 存放 3 位 BCD 码。

2. 程序清单

```

B8BCD:  MOV A,R2
        MOV B,*64H
        DIV AB
        MOV R4,A
        MOV A,*0AH
        XCH A,B
        DIV AB
        SWAP A
        ADD A,B
        MOV R5,A
        RET
    
```

3. 程序框图

见图 1-4。

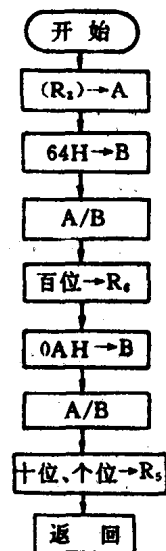


图 1-4

1.3.2 程序二(双字节整数)

1. 编制说明

因为

$$(a_{15}a_{14}\cdots a_1a_0)_2 = (\cdots(0 \times 2 + a_{15}) \times 2 + a_{14}\cdots) \times 2 + a_0$$

所以,将二进制数从最高位逐次左移入 BCD 码寄存器的最低位,并且每次都实现 $(\cdots) \times 2 + a_i$ 的运算,共循环 16 次,由 R_7 控制。

(1) 入口: R_3R_2 (16 位无符号二进制整数)。

(2) 出口: R_6 (万位)、 R_5 (千位、百位)、 R_4 (十位、个位)存放 5 位 BCD 码。

2. 程序清单

```

B16BCD: CLR A           ;BCD 码寄存器清零
        MOV R4,A
        MOV R5,A
        MOV R6,A
        MOV R7,#10H      ;设循环指针
LP0:    CLR C             ;左移一位,移入 C
        MOV A,R2
        RLC A
        MOV R2,A
        MOV A,R3
        RLC A
        MOV R3,A
        MOV A,R4          ;实现 $(\dots) \times 2 + a_i$ 运算
        ADDC A,R4
        DA A
        MOV R4,A
        MOV A,R5
        ADDC A,R5
        DA A
        MOV R5,A
        MOV A,R6
        ADDC A,R6
        DA A
        MOV R6,A
        DJNZ R7,LP0
        RET
    
```

3. 程序框图

见图 1-5。

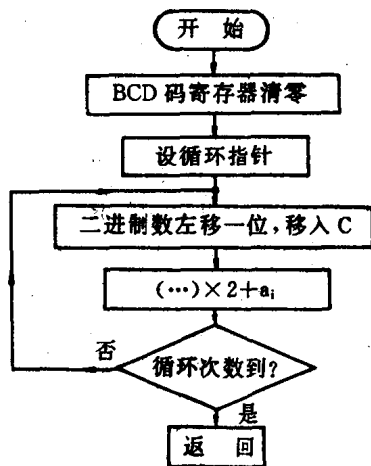


图 1-5

1.3.3 程序三(多字节整数)

1. 编制说明

原理、程序框图同程序二。

(1) 入口: R_0 (无符号二进制整数低位字节地址指针); R_7 (字节数)。

(2) 出口: R_1 (BCD 码结果低位字节地址指针)。

2. 程序清单:

```

BMBCD: MOV A,R0          ;复制保存地址指针
        MOV R5,A
        MOV A,R1
        MOV R6,A
        MOV A,R7          ;BCD 码字节数一般比二进制数要多一个字节
        MOV R3,A
        INC R3
        CLR A
CLBCD:  MOV @R1,A          ;BCD 码结果单元清零
        INC R1
        DJNZ R3,CLBCD
    
```

```

MOV A,R7          ;算出二进制数的位数作为循环指针
MOV B,*08H
MUL AB
MOV R3,A
LP0: MOV A,R3
MOV R0,A
MOV A,R7
MOV R2,A
CLR C
LP1: MOV A,®R0      ;左移一位,移入 C
RLC A
MOV ®R0,A
INC R0
DJNZ R2,LP1
MOV A,R6
MOV R1,A
MOV A,R7
MOV R2,A
INC R2
LP2: MOV A,®R1      ;实现(…)×2+ai 运算
ADDC A,®R1
DA A
MOV ®R1,A
INC R1
DJNZ R2,LP2
DJNZ R3,LP0
MOV A,R6
MOV R1,A
RET

```

1.3.4 程序四(多字节小数)

1. 编制说明

将二进制小数乘 10 得到相应位的 BCD 码,循环运算的次数由所需 BCD 码的位数确定。

(1) 入口: R₀(无符号二进制小数低位字节地址指针); R₇(字节数); R₆(所需 BCD 码结果字节数)。

(2) 出口: R₁(BCD 码结果低位字节地址指针)。

2. 程序清单

```

BMDBCD: MOV A,R0      ;复制保存地址指针
MOV R4,A
MOV A,R1
MOV R5,A
MOV A,R6
MOV R3,A
CLR A
CLBCD:  MOV ®R1,A      ;BCD 码结果单元清零
INC R1
DJNZ R3,CLBCD
MOV A,R6          ;所需 BCD 码的位数作为循环指针

```

```

      RL    A
      MOV   R3, A
LP0:  MOV   A, R4
      MOV   R0, A
      MOV   A, R7
      MOV   R2, A
      MOV   R1, *00H
      CLR   C
LP1:  MOV   A, *R0      ;乘 10 得到相应位的 BCD 码,存于 R1 低 4 位
      MOV   B, *0AH
      PUSH  PSW
      MUL   AB
      POP   PSW
      ADDC  A, R1
      MOV   *R0, A
      MOV   R1, B
      INC   R0
      DJNZ  R2, LP1
      MOV   A, R1
      ADDC  A, *00H
      MOV   R1, A
      MOV   A, R4      ;本次所得 BCD
      MOV   R2, A      ;码移入最低位
      MOV   A, R5      ;结果单元
      MOV   R0, A
LP2:  MOV   A, *R0
      SWAP  A
      MOV   *R0, A
      XCH   A, R1
      XCHD  A, *R0
      INC   R0
      DJNZ  R2, LP2
      DJNZ  R3, LP0
      MOV   A, R5
      MOV   R1, A
      RET

```

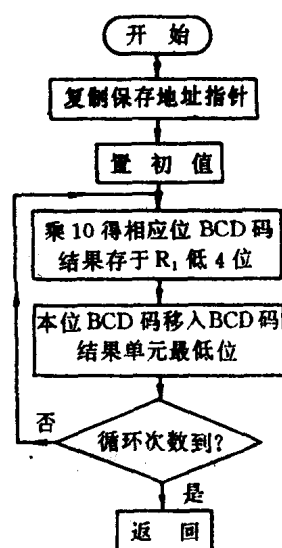


图 1-6

3. 程序框图

见图 1-6。

1.4 BCD 码到二进制数的转换

1.4.1 程序一(单字节整数)

1. 编制说明

高位 BCD 码乘 10 加低位 BCD 码。

(1) 入口: R₂(BCD 码)。

(2) 出口: R₂(8 位无符号二进制整数)。

2. 程序清单

BCD2B: MOV A, R₂

```

ANL    A,*0F0H
SWAP   A
MOV    B,*0AH
MUL    AB
MOV    R3,A
MOV    A,R3
ANL    A,*0FH
ADD    A,R3
MOV    R2,A
RET

```

3. 程序框图

见图 1-7。

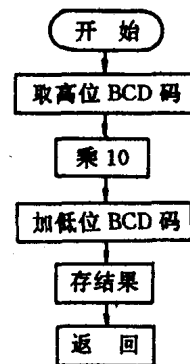


图 1-7

1.4.2 程序二(双字节整数)

1. 编制说明

因为

$$(d_3d_2d_1d_0)_{\text{BCD}} = (d_3 \times 10 + d_2) \times 100 + (d_1 \times 10 + d_0)$$

所以, $d_{i+1} \times 10 + d_i$ 运算可由子程序 BCD2B 完成。

(1) 入口: R_5 (千位、百位)、 R_4 (十位、个位)为 BCD 码。

(2) 出口: R_5, R_4 (16 位无符号二进制整数)。

2. 程序清单

```

BCD4B:  MOV    A,R5
        MOV    R3,A
        ACALL  BCD2B    ;参见程序一
        MOV    B,*64H
        MUL    AB
        MOV    R4,A
        XCH    A,B
        MOV    R5,A
        MOV    A,R4
        MOV    R2,A
        ACALL  BCD2B    ;参见程序一
        ADD    A,R5
        MOV    R4,A
        MOV    A,R4
        ADDC   A,*00H
        MOV    R5,A
        RET

```

3. 程序框图

见图 1-8。

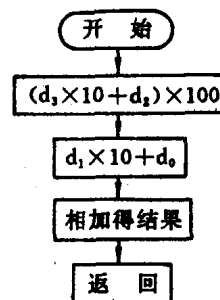


图 1-8

1.4.3 程序三(多字节整数)

1. 编制说明

因为

$$(d_n d_{n-1} \dots d_1 d_0)_{\text{BCD}} = (\dots (0 \times 10 + d_n) \times 10 + d_{n-1} \dots) \times 10 + d_0$$

所以,每个字节进行二次 $(\dots) \times 10 + d_i$ 运算,循环次数由 BCD 码字节数控制。

(1) 入口: R_0 (BCD 码高位字节地址指针); R_7 (BCD 码字节数)。

(2) 出口: R_1 (无符号二进制整数低位字节地址指针)。

2. 程序清单

```
BCDMB:  MOV    A, R1      ;复制保存地址指针
        MOV    R6, A
        MOV    A, R7
        MOV    R3, A
        CLR    A
CLBIN:  MOV    @R1, A     ;二进制数结果单元清零
        INC    R1
        DJNZ   R3, CLBIN
        MOV    A, R7     ;设循环指针
        MOV    R3, A
LP:     ACALL  MU10
        MOV    A, @R0
        ANL    A, #0F0H
        SWAP   A
        ACALL  SUM
        ACALL  MU10
        MOV    A, @R0
        ANL    A, #0FH
        ACALL  SUM
        DEC    R0
        DJNZ   R3, LP
        MOV    A, R6
        MOV    R1, A
        RET
MU10:   MOV    A, R7     ;( $\dots$ ) $\times 10$ 
        MOV    R4, A
        MOV    A, R6
        MOV    R1, A
        CLR    C
        MOV    R2, #00H
LP1:    MOV    A, @R1
        MOV    B, #0AH
        PUSH  PSW
        MUL   AB
        POP   PSW
        ADDC  A, R2
        MOV    @R1, A
        MOV    R2, B
        INC    R1
        DJNZ   R4, LP1
        RET
SUM:    MOV    R5, A     ;加  $d_i$ 
        MOV    A, R6
        MOV    R1, A
        MOV    A, R7
        MOV    R4, A
        DEC    R4
```

```

MOV    A,R5
ADD    A,®R1
MOV    ®R1,A
INC    R1
LP2:   MOV    A,®R1
        ADDC  A,*00H
        MOV    ®R1,A
        INC    R1
        DJNZ  R4,LP2
RET

```

3. 程序框图

见图 1-9。

1.4.4 程序四(多字节小数)

1. 编制说明

将 BCD 码小数乘 2 得到相应位的二进制小数,循环运算的次数由所需二进制小数的位数确定。

(1) 入口: R₆(BCD 码小数低位字节地址指针); R₅(字节数); R₇(所需二进制小数结果字节数)。

(2) 出口: R₁(无符号二进制小数结果低位字节地址指针)。

2. 程序清单

```

BCDMDB:  MOV    A,R6           ;复制保存地址指针
          MOV    R4,A
          MOV    A,R1
          MOV    R5,A
          MOV    A,R7
          MOV    R3,A
          CLR    A
CLBIN:   MOV    ®R1,A         ;二进制小数结果
          INC    R1           单元清零
          DJNZ  R2,CLBIN
          MOV    A,R7         ;所需二进制小数的
          MOV    B,®08H       位数作为循环指针
          MUL    AB
          MOV    R3,A
LP0:     MOV    A,R5
          MOV    R2,A
          MOV    A,R4
          MOV    R0,A
          CLR    C
LP1:     MOV    A,®R0         ;乘 2 得到相应位的
          ADDC  A,®R0       二进制小数存于 C
          DA    A
          MOV    ®R0,A
          INC    R0
          DJNZ  R2,LP1
          MOV    A,R5
          MOV    R1,A

```

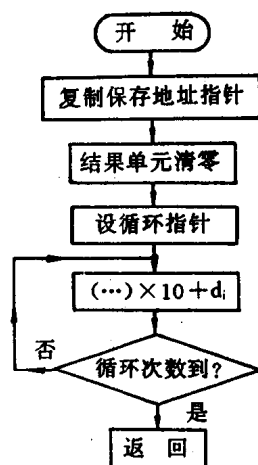


图 1-9

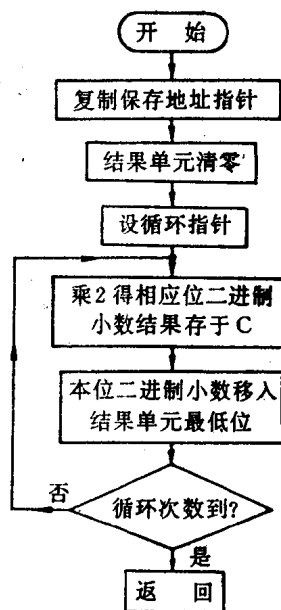


图 1-10