

MCS—51
DAN PIAN JI
KAI FA
YU
YING YONG
ZHI NAN

MCS—51
单片机开发与
应用指南

● 孙永浩 王铁流 编著

MCS—51

单片机开发与应用指南

孙永浩 王铁流 编著

黑龙江科学技术出版社

责任编辑：郑 爽

封面设计：洪 冰

单片机开发与应用指南

孙永浩 王铁流 编著

黑龙江科学技术出版社出版
(哈尔滨市南岗区建设街35号)

哈尔滨新路印刷厂印刷

787×1092毫米16开本12.75印张250千字

1989年8月第1版·1989年8月第1次印刷

印数：1—4000册 定价：5.50元

ISBN7-5388-0844-2/TP·7

前 言

1976年以来, Intel公司生产了MCS—48系列单片机。80年代, 该公司在MCS—48基础上加以改进, 又推出了一种新的单片机系列——MCS—51。MCS—51的典型产品为8051, 它有4K×8ROM, 128字节RAM, 2个16位定时/计数器, 4个8位I/O口, 一个串行口, 可代替各种微处理器(它的功能可近似相当于由一片Z80CPU, 一片2732EPROM, 一片RAM, 一片Z80CTC, 两片Z80PIO和一片Z80SIO组成的系统), 运用于各种复杂的控制和仪器仪表中。

为了有助于了解MCS—51的性能和推广单片微型计算机的应用, 根据单片计算机教学及科技开发工作的需要, 我们编写了《MCS—51单片机开发与应用指南》这本书。它既可作为单片机教学的实验指导书, 也可作为工程科技人员学习和应用单片机的自学实践教材, 是一本较有实用价值的单片机程序设计和接口电路设计的应用指南。

全书的主要内容可分三部分。第一部分是程序设计练习。通过程序设计, 可进一步了解单片机的数据传送, 算术运算, 逻辑操作, 子程序调用和各种转移等指令的运用, 提高汇编语言的编程技巧。并通过程序的调试, 掌握单片机开发系统的各种功能和使用方法。因浮点运算的设计较复杂, 故在附录中列出。

第二部分为接口设计。单片机的应用离不开接口电路设计, 其中A/D和D/A转换器, 开关量的输入/输出, 键盘输入和字型显示等接口的设计是非常重要而普遍的。在接口设计中首先要了解所使用的芯片的引脚排列以及功能, 并在此基础上根据实际应用条件考虑如何与单片机进行配接。有时用软件可替代部分硬件的接口功能, 因此在接口设计中不仅要考虑使用什么功能的芯片和如何连接电路, 而且要考虑在这种电路中如何使用相应的指令来完成管理接口的任务。

第三部分为系统设计。目前, 国内开发系统种类很多。本书以黑龙江大学单片机开发与应用研究室设计研制的HDW—I型MCS—51单片机开发与应用兼容系统为例进行操作。不同的开发系统在信号的引出位置和标号方面可能有差异, 但是其设计思想和使用方法是完全相同的。因此, 本书的全部程序和接口电路对于其它类型的MCS—51单片机开发系统的应用具有普遍指导意义。在开发系统设计中详细介绍了HDW—I型开发系统的功能并提供了整个系统全部的实际电路图。开发系统本身的设计就是MCS—51单片机软件和硬件设计的最好的应用举例。因为其中包括了程序存储器的扩展, 数据存储器的扩展, 外围接口片8155和8255的扩展, 译码电路, EPROM固化方法及其电路, 键盘显示的硬件电路和软件设计等等。应用系统设计是以智能采样功率表作为一个实例给出。可以说这是单片机在智能仪器仪表中的典型应用, 因为其中包括数值计算、采样保持、A/D转换、定时/计数器编程和中断处理等软硬件设计的内容, 具有一定的普通意义。为了使读者了解整个方案的设计思想和具体实现过程, 这里给出了全部的硬件电路图和程序清单。其中有些子程序有一定的通用性可供随意调用。

为了使用方便, 附录中列有常用集成电路芯片引脚图, 同时附有MCS—51单片机指令速查表提供参考。此外还提供了一些较为复杂但实用的运算子程序。

本书是以本科生教学及单片机短训班中多次使用的实验讲义为基础，加上我们近年来在单片机应用成果中的一些实例而扩充和整理出来的。由于编者水平有限，书中定有错误或不妥之处，敬请读者批评指正。

本书由方振贤副教授校阅，丁群同志协助做了大量的工作，黑龙江大学万达电子公司对本书的出版给予了大力协助，在此表示感谢。

编者

1989年5月

目 录

第一部分 程序设计练习

一. 代码转换	(1)
1. 4位2进制数转换成ASCII码	(1)
2. 16进制数的ASCII码转换成4位2进制数	(3)
3. 多字节2进制整数转换成BCD码	(5)
4. 多位BCD码小数转换为2进制小数	(7)
5. 多字节2进制小数转换为BCD码	(9)
6. 多位整数BCD码转换成2进制整数	(12)
二. 算术运算	(15)
1. 10进制运算	(15)
2. 2进制运算方法	(17)
3. 多字节移位子程序	(19)
4. 双字节整数乘法子程序	(20)
5. 双字节整数除法子程序	(24)

第二部分 单片机的I/O接口扩展

一. 并行I/O接口硬件扩展及软件编程	(29)
(一) I/O接口芯片8255的扩展	(29)
1. 8255A芯片介绍	(29)
2. 8255A的输入/输出编程实验	(40)
(二) 8255A在EPROM固化板上的应用	(42)
1. 固化板编程	(42)
2. EPROM固化程序分析	(44)
(三) I/O接口与外部RAM扩展 (8155)	(46)
1. 8155芯片介绍	(46)
2. 8155中扩展RAM的读写	(54)

3. 8155的输入/输出编程	(54)
4. 8155的CTC定时编程	(55)
二、模拟/数字转换接口	(56)
(一) 8位A/D转换 (ADC0809)	(56)
1. ADC0809模拟/数字转换芯片介绍	(56)
2. 实验内容与步骤	(58)
(二) 高精度模/数转换 (ICL8068/ICL7104—16)	(59)
1. ICL8068/ICL7104—16CPL的主要特性	(60)
2. 16位A/D转换器的典型电路	(61)
3. ICL7104与MCS—51单片机的接口	(62)
(三) 三位半双积分A/D变换器 (5G14433)	(64)
1. 5G14433 的主要特性	(64)
2. 5G14433 的引脚功能说明	(64)
3. 编程操作	(67)
三、数字/模拟转换接口	(68)
(一) 8位D/A转换 (DAC0832)	(69)
1. DAC0832数模转换芯片介绍	(69)
2. 实验内容与步骤	(72)
(二) 双锁存D/A转换	(73)
1. 实验原理	(73)
2. 实验内容与步骤	(74)
(三) 高精度数字/模拟转换 (7520)	(74)
1. 转换原理	(75)
2. 实验操作	(75)
四、存储器的扩充	(76)
1. 单片机常用存储器芯片介绍	(76)
2. 实验内容	(80)
3. 实验步骤	(81)
五、定时器/计数器 (CTC) 应用	(83)
(一) 定时编程操作	(83)
1. CTC介绍	(83)
2. 实验内容及步骤	(84)

(二) 用CTC作实时时钟产生器.....	(87)
1. 实验内容.....	(87)
2. 实验步骤.....	(87)
3. 参考程序.....	(87)
六. 开发系统的用户自定义键盘及显示程序设计.....	(91)
1. 实验内容及步骤.....	(91)
2. 键盘分析与显示子程序的注释及使用.....	(94)
七. 单片机的中断	(99)
1. MCS—51系列单片机的中断系统概述	(99)
2. 实验内容	(101)
3. 实验步骤.....	(101)
八. 通用打印机接口设计	(104)
1. 通用打印机接口介绍	(104)
2. 用8255A扩充的打印机接口电路.....	(105)
3. 打印程序的编制	(105)
4. 其它接口方式	(107)
九. 串行口实验.....	(109)

第三部分 单片机系统设计

一. 开发系统设计.....	(114)
(一) 开发系统概述	(114)
(二) 开发系统的键盘操作	(116)
1. 返回“P”状态及存储器读写.....	(116)
2. 寄存器及外存检查.....	(118)
3. 设置断点和运行程序.....	(120)
4. 插入、删除与偏移量.....	(123)
5. EPROM固化与检查.....	(126)
6. 数据块搬家及比较.....	(127)
(三) 开发系统的结构与设计.....	(128)
1. 主机板的设计.....	(129)
2. 键盘/显示电路的设计.....	(130)

3. EPROM固化板的设计	(131)
4. 实验板的设计	(131)
二. 应用系统设计	(136)
(一) 系统设计原理	(136)
(二) 全部程序清单	(140)
附录A 浮点运算子程序	(155)
A. 1 多字节2进制整数转换成浮点数	(155)
A. 2 浮点数取补子程序	(155)
A. 3 浮点数规格化子程序	(156)
A. 4 浮点数对阶子程序	(159)
A. 5 浮点数加法子程序	(161)
A. 6 浮点数减法子程序	(164)
A. 7 多字节浮点数乘法	(164)
A. 8 浮点数除法子程序	(169)
附录B MCS—51单片机汇编及指令表	(176)
附录C 常用集成电路引脚图	(188)

第一部分 程序设计练习

一. 代码转换

在计算机内部直接参与运算的数据是2进制码,但有时需要用ASCII码与外部设备进行信号交换;有时要用BCD码完成10进制运算。本实验的目的是了解2进制数与ASCII码或BCD码互相转换的基本方法。同时对初学编程的人可通过操作练习熟悉一下主程序的编制与子程序的调用。

1. 4位2进制数转换成ASCII码

实验的目的:了解2进制数转换成ASCII码的基本方法。

说明:2进制数与ASCII码的对应关系如表1—1所示。

表1—1

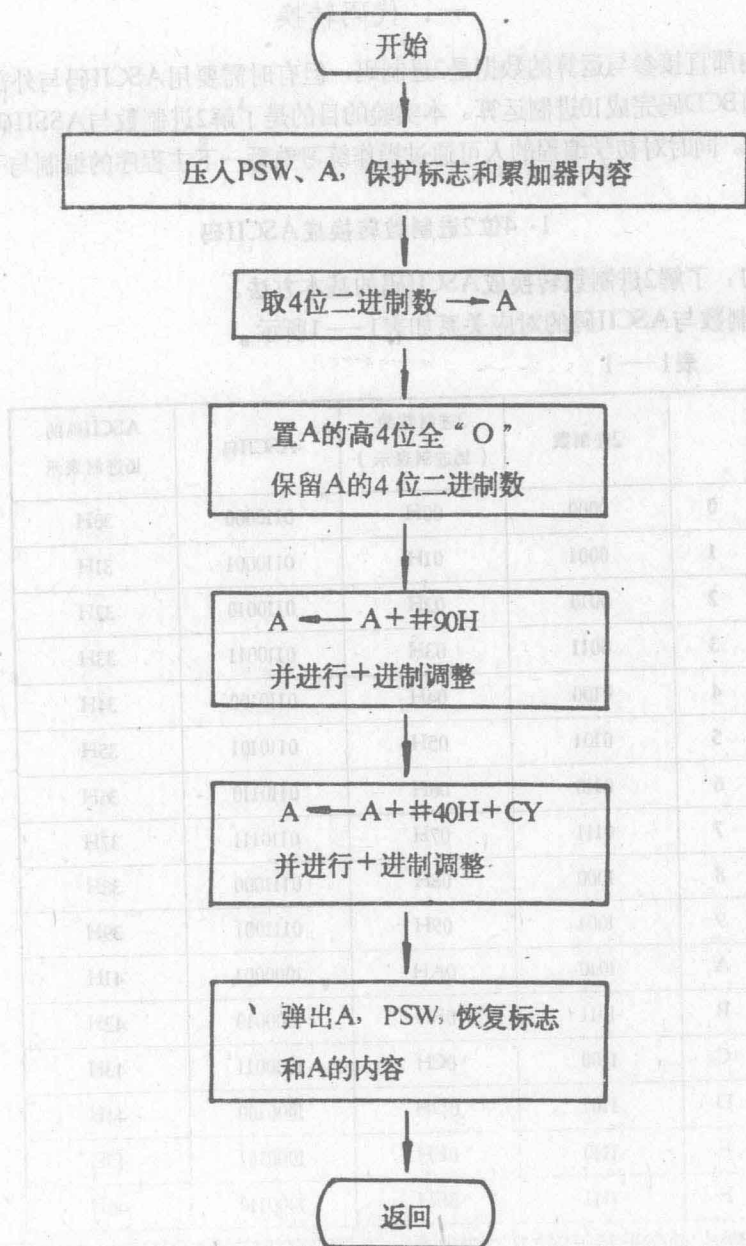
	2进制数	2进制数的 (16进制表示)	ASCII码	ASCII码的 16进制表示
0	0000	00H	0110000	30H
1	0001	01H	0110001	31H
2	0010	02H	0110010	32H
3	0011	03H	0110011	33H
4	0100	04H	0110100	34H
5	0101	05H	0110101	35H
6	0110	06H	0110110	36H
7	0111	07H	0110111	37H
8	1000	08H	0111000	38H
9	1001	09H	0111001	39H
A	1010	0AH	1000001	41H
B	1011	0BH	1000010	42H
C	1100	0CH	1000011	43H
D	1101	0DH	1000100	44H
E	1110	0EH	1000101	45H
F	1111	0FH	1000110	46H

由以上关系可看出对小于等于9的4位2进制数加30H即得相应的ASCII码,对大于9的4位2进制数,加37H得到相应的ASCII码。

实验内容与步骤：

- (1) 编制把4位2进制数转换成ASCII码的子程序。
- (2) 把程序代码输入到2000H开始的存储器中。
- (3) 编制主程序，调用首地址为2000H的子程序。
- (4) 用“单步”键执行该程序，观察程序执行过程及转换结果。

参考子程序流程图：



参考程序：
入口参数：4位2进制数存放于R₂寄存器中
出口参数：ASCII码存放于R₂寄存器中

主程序

```

ORG 2800H
2800 12 20 00    LCALL 2000
2803 01 03        AJMP 2803
  
```

子程序

```

ORG 2000H
2000 C0 D0        PUSH PSW
2002 C0 E0        PUSH A
2004 EA          MOV A,R2
2005 54 0F        ANL A,#0FH
2007 24 90        ADD A,#90H
2009 D4          DA A
200A 34 40        ADDC A,#40H
200C D4          DA A
200D FA          MOV R2,A
200E D0 E0        POP A
2010 D0 D0        POP PSW
2012 22          RET
  
```

运算实例：

- | | |
|---------------------------|----------------------------|
| ① (R ₂) = 03H | 结果 (R ₂) = 33H |
| ② (R ₂) = 0AH | 结果 (R ₂) = 41H |

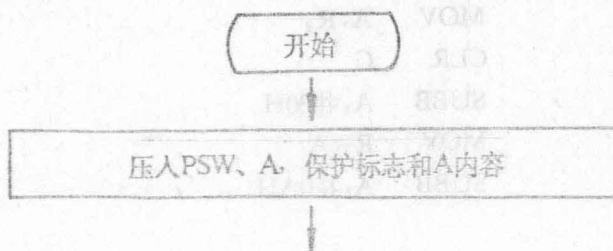
2. 16进制数的ASCII码转换成4位2进制数

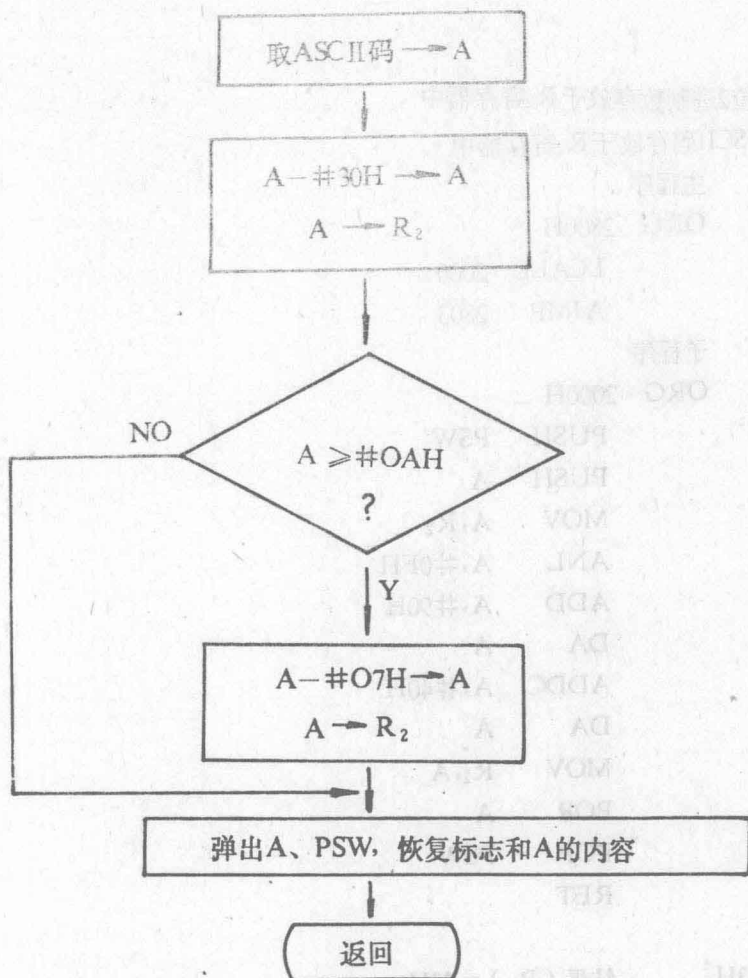
目的：了解ASCII码转换成4位2进制数的基本方法

实验内容与步骤：

- (1) 编制16进制数的ASCII码转换成4位2进制数的子程序。
- (2) 把程序代码输入到2015H开始的存储器中。
- (3) 编制主程序调用首地址为2015H的子程序。
- (4) 分别用“连续”和“单步”键执行程序，并检查转换结果。

参考子程序流程图





参考程序：

入口参数：ASCII码存放于R₂中。

出口参数：4位2进制数结果存放于R₂中。

主程序

ORG 2800H

2800 12 20 15

LCALL 2015H

2803 01 03

AJMP 2803H

子程序

ORG 2015H

2015 C0 D0

PUSH PSW

2017 C0 E0

PUSH A

2019 EA

MOV A, R₂

201A C3

CLR C

201B 94 30

SUBB A, #30H

201D FA

MOV R₂, A

201E 94 0A

SUBB A, #0AH

2020	40 04		JC	EXT
2022	CA		XCH	A, R ₂
2023	94 07		SUBB	A, #07H
2025	FA		MOV	R ₂ , A
2026	D0 E0	EXT:	POP	A
2028	D0 D0		POP	PSW
202A	22		RET	

说明：对于 ≤ 9 的数的ASCII码减去30H得4位2进制数；对于 > 9 的16进制的ASCII码减去37H得2进制数。

运算实例：

- | | |
|---------------------------|----------------------------|
| ① (R ₂) = 37H | 结果 (R ₂) = 07H |
| ② (R ₂) = 45H | 结果 (R ₂) = 0EH |

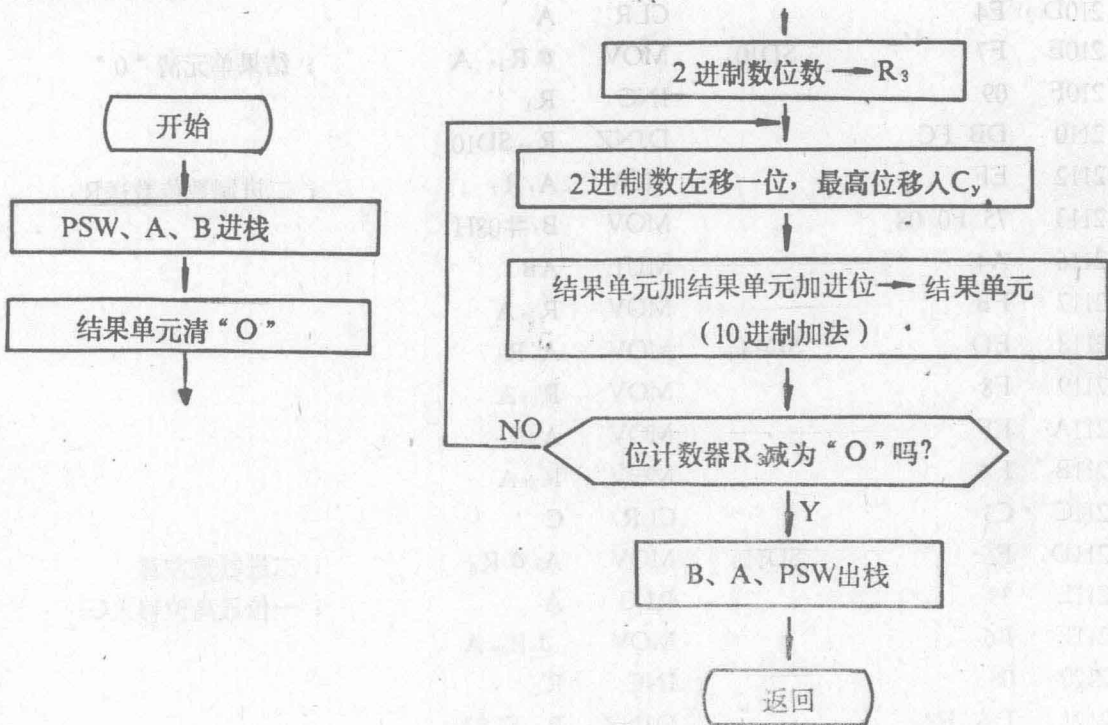
3. 多字节2进制整数转换成BCD码

目的：了解2进制整数转换成BCD码的基本方法。

实验内容与步骤：

- (1) 编制多字节2进制整数转换成BCD码子程序。
- (2) 编制主程序调用其子程序，观察转换结果。

参考子程序流程图



参考程序：

入口参数：2进制数低位字节地址指针存放于R₀，2进制数字字节数存放于R₇。

出口参数：BCD码结果低位字节地址指针存放于R₁，BCD码字节数为R₇的内容加1。

计算方法：将2进制数从高位开始逐次移入结果寄存器的最低位，当最低位移入时，以10进制方式乘2。BCD结果就存放于结果单元之中。

主程序

ORG 2800H

2800 12 21 00 LCALL 2100

2803 01 03 AJMP 2803

子程序

ORG 2100H

2100 C0 D0 SUBD1: PUSH PSW

2102 C0 E0 PUSH ACC

2104 C0 F0 PUSH B

2106 E8 MOV A, R₀

2107 FD MOV R₅, A

2108 E9 MOV A, R₁

2109 FE MOV R₆, A

210A EF MOV A, R₇

210B 04 INC A

210C FB MOV R₃, A

210D E4 CLR A

210E F7 SD10: MOV @R₁, A

; 结果单元清“0”

210F 09 INC R₁

2110 DB FC DJNZ R₃, SD10

2112 EF MOV A, R₇

; 二进制数位送R₃

2113 75 F0 08 MOV B, #08H

2116 A4 MUL AB

2117 FB MOV R₃, A

2118 ED SD14: MOV A, R₅

2119 F8 MOV R₀, A

211A EF MOV A, R₇

211B FA MOV R₂, A

211C C3 CLR C

211D E6 SD11: MOV A, @R₀

; 二进制数左移

211E 33 RLC A

; 一位最高位移入C_Y

211F F6 MOV @R₇, A

2120 08 INC R₀

2121 DA FA DJNZ R₂, SD11

2123 EE MOV A, R₅

2124	F9		MOV	R ₁ , A	
2125	EF		MOV	A, R ₇	
2126	FA		MOV	R ₂ , A	
2127	0A		INC	R ₂	
2128	E7	SD13;	MOV	A, $\odot$ R ₁	C _y 加移入结果单元
2129	37		ADDC	A, $\odot$ R ₁	
212A	D4		DA	A	
212B	F7		MOV	$\odot$ R ₁ , A	
212C	09		INC	R ₁	
212D	DA F9		DJNZ	R ₂ , SD13	
212F	DB E7		DJNZ	R ₃ , SD14	
2131	D0 F0		POP	B	
2133	D0 E0		POP	A	
2135	D0 D0		POP	PSW	
2137	22		RET		

运算实例:

① (R ₀) = 20H	(R ₁) = 24H	(R ₇) = 02H
(20H) = 64H	(21H) = 00H	
结果: (24H) = 00H	(25H) = 01H	(26H) = 00H
即0064H的10进制数为100。		
② (R ₀) = 20H	(R ₁) = 24H	(R ₇) = 02H
(20H) = FFH	(21H) = FFH	
结果: (24H) = 35	(25H) = 55	(26H) = 06
即FFFF的10进制整数为65535。		

4. 多位BCD码小数转换为2进制小数

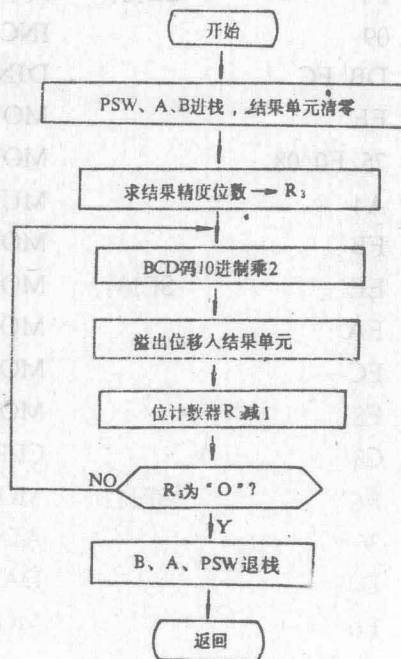
目的: 了解多位BCD码小数转换为2进制小数的基本方法,

学会利用此子程序进行转换。

实验内容与步骤:

- (1) 编制多位BCD码小数转换为2进制的子程序。
- (2) 编制主程序调其子程序进行小数转换。

参考子程序流程图:



参考程序：

入口参数：BCD码低位字节地址指针存放于R₀，BCD码字节数存放于R₃，“2进制数的字节数”存放于R₇。

出口参数：2进制小数低位字节地址指针存放于R₁，2进制小数字节数存放于R₇。

计算方法：将BCD码小数进行10进制乘2，溢出部分即为相应的小数位，R₇的内容由精度确定。

主程序

ORG 2800H

2800 12 22 80

LCALL 2280

2803 01 03

AJMP 2803

子程序

ORG 2280H

2280 C0 D0

SUBE1: PUSH PSW

2282 C0 E0

PUSH A

2284 C0 F0

PUSH B

2286 E8

MOV A, R₀

2287 FC

MOV R₄, A

2288 E9

MOV A, R₁

2289 FD

MOV R₅, A

228A EF

MOV A, R₇

228B FB

MOV R₃, A

228C E4

CLR A

228D F7

SE10: MOV @R₁, A

228E 09

INC R₁

228F DB FC

DJNZ R₃, SE10

2291 EF

MOV A, R₇

2292 75 F0 08

MOV B, #08H

2295 A4

MUL AB

2296 FB

MOV R₃, A

2297 EE

SE13: MOV A, R₆

2298 FA

MOV R₂, A

2299 EC

MOV A, R₄

229A F8

MOV R₀, A

229B C3

CLR C

229C E6

SE11: MOV A, R₀

229D 36

ADDC A, @R₁

229E D4

DA A

229F F6

MOV @R₀, A