

CMC-80

微型电脑  
实用子程序手册



江苏工学院  
电子计算机教研室编

# 目 录

## 第一章 代码转换程序

- 1—1 十六进制→ASCII码转换程序…………… ( 1 )
- 1—2 ASCII码→十六进制数转换程序…………… ( 1 )
- 1—3 十六进制数的ASCII码→四位二进制数转换程序…………… ( 2 )
- 1—4 ASCII码→二进制转换程序…………… ( 4 )
- 1—5 ASCII码→BCD码转换程序…………… ( 4 )
- 1—6 BCD码→ASCII码转换程序…………… ( 5 )
- 1—7 BCD码→二进制转换程序…………… ( 7 )
- 1—8 十六位二进制数→BCD码数的转换程序—I…………… ( 8 )
- 1—9 二进制数→ASCII码字符串转换程序…………… ( 10 )
- 1—10 一位十六进制数→ASCII码转换程序…………… ( 11 )
- 1—11 十进制(BCC码)→七段代码转换程序…………… ( 11 )
- 1—12 十六进制→七段代码转换程序…………… ( 12 )
- 1—13 十六位二进制数→BCD码数转换程序—II…………… ( 13 )

## 第二章 二进制定点运算

- 2—1 多字节加法…………… ( 16 )
- 2—2 多字节减法…………… ( 16 )
- 2—3 单字节无符号数乘法…………… ( 17 )
- 2—4 单字节有符号数乘法(纯小数)…………… ( 18 )
- 2—5 单字节有符号数乘法(整数)…………… ( 19 )
- 2—6 多字节无符号数乘法…………… ( 20 )
- 2—7 多字节有符号数乘法…………… ( 23 )
- 2—8 单字节无符号数除法(整除)…………… ( 26 )
- 2—9 单字节有符号数除法(一)…………… ( 27 )
- 2—10 单字节有符号数除法(二)…………… ( 29 )

2—11	多字节无符号数除法(整除)	( 31 )
2—12	多字节有符号数除法(整除)	( 34 )
2—13	求单字节数的绝对值	( 37 )
2—14	求算术平均值	( 38 )

### 第三章 BCD码运算

3—1	单字节BCD码加法	( 41 )
3—2	多字节BCD码加法	( 41 )
3—3	单字节BCD码减法	( 42 )
3—4	多字节BCD码减法	( 42 )
3—5	一位BCD码乘法	( 43 )
3—6	单字节BCD码乘法	( 44 )
3—7	多字节BCD码乘法	( 46 )
3—8	单字节BCD码除法	( 48 )
3—9	多字节BCD码除法(一)	( 51 )
3—10	多字节BCD码除法(二)	( 54 )

### 第四章 浮点运算及函数

4—1	二进制数转化为浮点数FLOT	( 58 )
4—2	浮点数规格化子程序FNOR	( 59 )
4—3	浮点数加法子程序FADD	( 62 )
4—4	浮点数减法子程序FSUB	( 65 )
4—5	浮点数乘法子程序FMUL	( 65 )
4—6	浮点数除法子程序FDIV	( 69 )
4—7	浮点数平方根子程序FSQT	( 73 )
4—8	浮点数绝对值子程序FABS	( 75 )
4—9	存浮点数子程序FGET	( 76 )
4—10	浮点数退栈子程序FPOP	( 77 )
4—11	浮点数进栈子程序FPSH	( 78 )
4—12	浮点数求补子程序FNEG及FNGX	( 79 )
4—13	浮点数求积(商)符号子程序FSIG	( 81 )
4—14	浮点数对阶右移子程序FASR	( 82 )

4—15	多字节左移一位子程序FRAL及FRLO	( 84 )
4—16	初始化子程序	( 84 )
4—17	单字节带进位加法子程序FADS	( 84 )
4—18	单字节带进位减法子程序FSBS	( 85 )
4—19	B、C、D寄存器左移子程序RLBD	( 85 )
4—20	浮点数平方子程序	( 86 )
4—21	浮点数立方子程序FCUBE	( 86 )
4—22	取浮点数字子程序FPUT	( 87 )
4—23	角度化弧度子程序CDTR	( 88 )
4—24	弧度化角度子程序CRTD	( 88 )
4—25	角度余弦函数FCOD	( 88 )
4—26	角度正弦函数FSID	( 89 )
4—27	弧度余弦函数FCOS	( 89 )
4—28	弧度正弦函数FSIN	( 89 )
4—29	正切函数FTAN	( 93 )
4—30	常用对数函数FLOG	( 94 )
4—31	自然对数函数FLN	( 94 )
4—32	反正切函数FATN	( 99 )
4—33	指数函数FEXP	( 104 )
4—34	多项式计算子程序FCAL	( 108 )
4—35	取整函数FIXF	( 110 )
4—36	取整子程序FIXI	( 110 )
4—37	B、C、D寄存器右移子程序RRBD	( 112 )
4—38	浮点数比较子程序FCMP	( 112 )

## 第五章 分类与检索

5—1	找最大数(无符号)	( 114 )
5—2	找最大数(带符号)	( 115 )
5—3	顺序检索	( 117 )
5—4	对分搜索	( 118 )
5—5	八位排序(无符号)	( 121 )
5—6	八位排序(有符号)	( 122 )

5—7	十六位排序(无符号).....	( 125 )
-----	-----------------	---------

## 第六章 其他

6—1	数据块移动.....	( 127 )
6—2	内存数据块校验.....	( 128 )
6—3	CTC作为计数方式编程.....	( 130 )
6—4	CTC作为定时方式编程.....	( 132 )
6—5	1~255毫秒软件延时.....	( 133 )
6—6	1~255秒软件延时.....	( 134 )
6—7	1~255秒CTC中断延时.....	( 136 )
6—8	1~255分钟CTC中断延时.....	( 138 )
6—9	数据校验和(八位).....	( 139 )
6—10	数据校验和(十六位).....	( 141 )
6—11	条件选择转移程序(一).....	( 142 )
6—12	RAM唯一地址检查程序.....	( 143 )
6—13	RAM存贮器读写检查程序(一).....	( 145 )
6—14	RAM存贮器读写检查程序(二).....	( 148 )
6—15	显示子程序.....	( 151 )
6—16	七段显示器计数程序.....	( 153 )
6—17	唱歌程序.....	( 156 )
6—18	PIO方式3初始化程序.....	( 159 )
6—19	条件选择转移程序(二).....	( 159 )
6—20	A/D转换程序.....	( 161 )
6—21	一组无符号的十进制制数→二进制数.....	( 162 )

## 编 后

# 第一章 代码转换程序

## 1-1 十六进制→ASCII码转换程序

- 说明：1. 十六进制数放在累加器A中。  
2. 子程序执行后，该数的ASCII码放在累加器A中。  
3. 程序入口 HEXASC。

程序：

2000	FE0A	HEXASC: CP	10	
2002	3003	JR	NC, NEXT	; A ≥ 10, 转NEXT
2004	C630	ADD	A, 48	; A < 10, 加48
2006	C9	RET		
2007	C637	NEXT:	ADD	A, 55 ; A ≥ 10, 加55
2009	C9	RET		

框图：

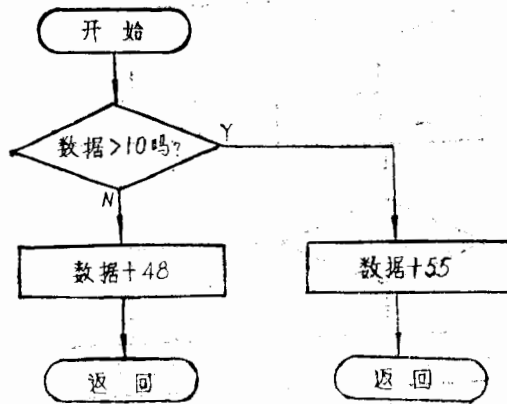


图 1-1

## 1-2 ASCII→十六进制数转换程序

- 说明：1. ASCII码放在累加器A中，转换后的十六进制数亦放A中。  
2. 如果A中放的是0—9的ASCII码，则减去30H后，即得0—9之间的数。如果A中放的是A—F的ASCII码，则减去41H后得0—5之间的数。如果A中的ASCII码不在0—9和A—F之间，则放错误标志FFH于A中。

程序:

2000	D630	ASHEX:	SUB	030H	
2002	FE0A		CP	0AH	; 检查该数是否在0-9之间
2004	D8		RET	C	
2005	D611		SUB	011H	; 不在0-9之间, 再减17
2007	FE06		CP	06H	; 该数是否在A-F之间
2009	3003		JR	NC,ERR	
200B	C60A		ADD	A,0AH	
200D	C9		RET		
200E	3EFF	ERR:	LD	A,0FFH	
2010	C9		RET		

框图:

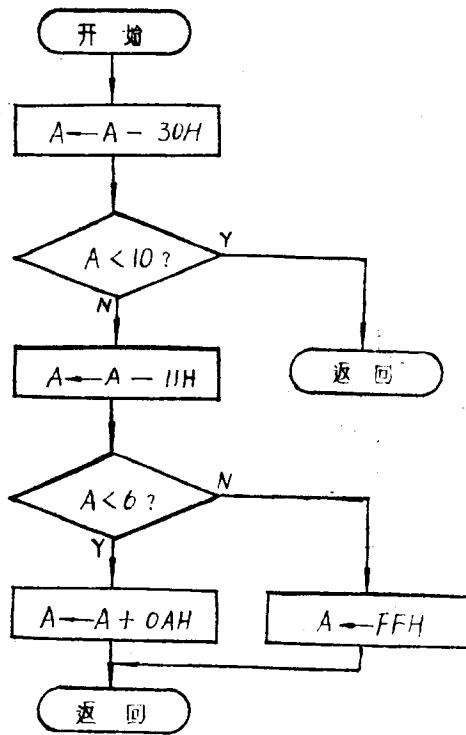


图 1-2

### 1-3 十六进制数的ASCII码→四位二进制数转换程序

说明: ASCII码放在A中, 转换后的二进制数仍存在A中。

程序:

3000	FE30	HVCK;	CP	30H
3002	3812		JR	C,NGH
3004	FE3A		CP	3AH
3006	3E0B		JR	C,OKH
3008	FE41		CP	41H
300A	380A		JR	C,NGH
300C	FE47		CP	47H
300E	F21630		JP	P,NGH
3011	D607		SUB	7
3013	E60F	OKH;	AND	0FH
3015	C9		RET	
3016	37	NGH;	SCF	
3017	C9		RET	

框图:

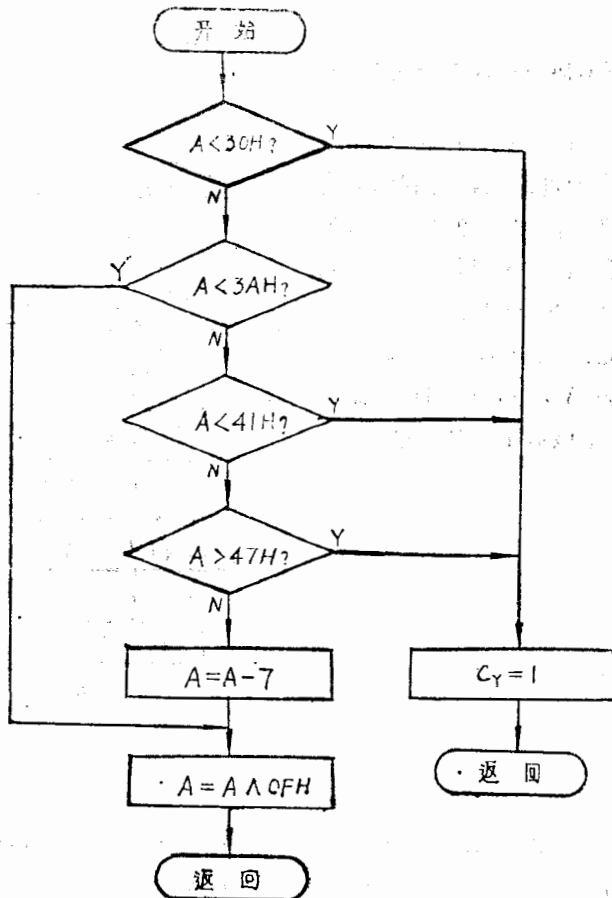


图 1-3



## 1-4 ASCII码→二进制转换程序

说明：将累加器内的ASCII码转换为二进制数。

程序：

2000	D630	UABIN:	SUB	030H
2002	FE0A		CP	0AH
2004	F8		RET	M
2005	D607		SUB	07H
2007	C9		RET	

框图：

## 1-5 ASCII码→BCD码转换程序

说明：1. 将2040H单元的内容从ASCII码字符转换为十进制数，并将结果存入2041H单元。如果2040H中放的不是十进制数ASCII码，则将2041H的内容置成FF。

2. 基本方法是：任一ASCII码十进制数字减去ASCII 0（16进制的30），便得出该数字的BCD数。

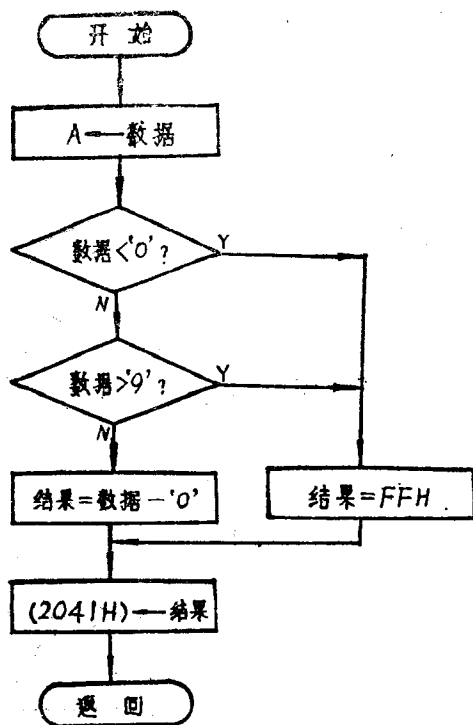


图 1-5

程序：

2000	06FF	ASCBCD:	LD	B,0FFH	；取错误标记
2002	3A4020		LD	A,(2040H)	

2005	D630		SUB	'0'	；数据<0吗？
2007	3805		JR	C,DONE	；是，则不是一个十进制数
2009	FE3A		CP	'9'+1	；数据>9吗？
200B	3001		JR	NC,DONE	；是，则不是一个十进制数
200D	47		LD	B,A	
200E	78	DONE:	LD	A,B	
200F	324120		LD	(2041H),A	
2012	C9		RET		

### 1—6 BCD码→ASCII码转换程序

- 说明：1. HL寄存器中存放被转换的BCD码数的首地址。  
 2. IY变址寄存器中存放转换后的ASCII首地址。  
 3. B寄存器中存放BCD码数的字节数。

程序：

2000	215020		LD	HL,2050H	
2003	FD210021		LD	IY,2100H	
2007	0605		LD	B,05H	
2009	CD0023		CALL	BCDA:SC	
200C	76		HALT		
2300	AF	BCDA:SC:	XOR	A	
2301	ED6F		RLD		；高位BCD→ASCII
2303	C630		ADD	A,30H	
2305	FD7700		LD	(IY),A	
2308	ED6F		RLD		；低位BCD→ASCII
230A	FD23		INC	IY	
230C	FD7700		LD	(IY),A	
230F	ED6F		RLD		；恢复原BCD码
2311	23		INC	HL	
2312	FD23		INC	IY	
2314	10EA		DJNZ	BCDA:SC	
2316	C9		RET		

框图:

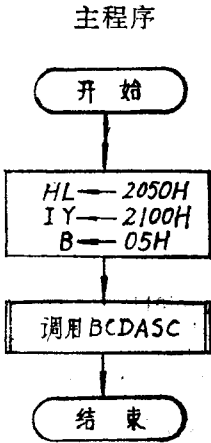


图 1-6-1

子程序

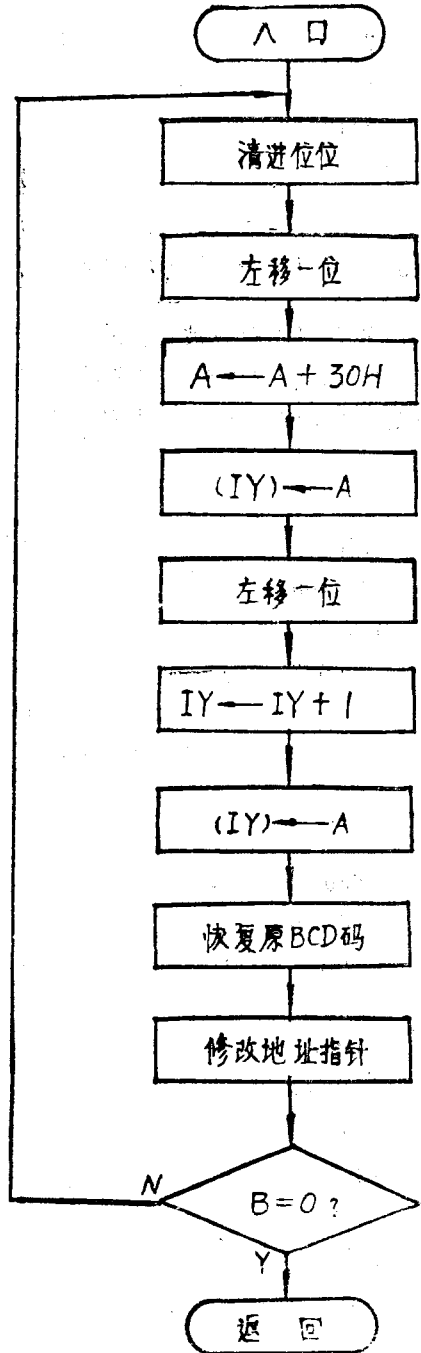


图 1-6-2

## 1-7 BCD码→二进制转换程序

说明：1. 将2040H和2041H单元中的两个BCD数转换成二进制数，放在2042H单元。

2. 转换的方法是：将BCD数的高位数乘以10，这可以用移位的方法来解决。

框图：

程序：

2000	214020	BCDBIN: LD	HL, (2040H)
2003	7E	LD	A, (HL)
2004	87	ADD	A, A
2005	47	LD	B, A
2006	87	ADD	A, A
2007	87	ADD	A, A
2008	80	ADD	A, B
2009	23	INC	HL
200A	86	ADD	A, (HL)
200B	23	INC	HL
200C	77	LD	(HL), A
200D	76	HALT	

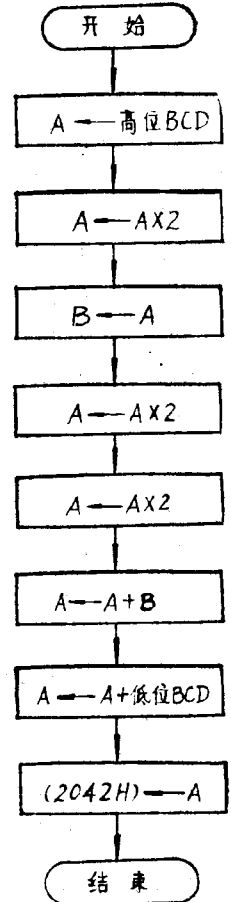


图 1-7

## 1-8 十六位二进制数→BCD数码的转换程序-I

- 说明: 1. 此方法是将二进制数从MSB开始逐位移入BCD寄存器的最低位来实现的。当最后一个二进制位已移入时, 转换的结果就存在BCD寄存器中了。
2. BCD寄存器实际上是以2050H开始的3个存贮单元, 二进制数存放在以2100H开始的2个存贮单元。
3. 若要转换32位二进制数, 只要修改二进制数的字节计数器(4)和BCD数字字节计数器(5)就行了。

程序:

2000	215020	BINBCD:	LD HL, 2050H	; 指向结果
2003	0603		LD B, 03H	; BCD字节计数
2005	3600	ILOOP:	LD (HL), 00H	
2007	23		INC HL	
2008	10FB		DJNZ ILOOP	
200A	0610		LD B, 10H	; 移位的位数
200C	210021	CLOOP:	LD HL, 2100H	; 指向二进制数的低位字节
200F	0E02		LD C, 02H	; 字节计数
2011	AF		XOR A	
2012	7E	RLOOP:	LD A, (HL)	; 取一字节
2013	17		RLA	
2014	77		LD (HL), A	; 存一字节
2015	23		INC HL	
2016	0D		DEC C	
2017	C21220		JP NZ, RLOOP	; C≠0, 转
201A	215020		LD HL, 2050H	; 指向存放结果的第一单元 (低位)
201D	0E03		LD C, 03H	; 字节计数值
201F	7E	BLOOP:	LD A, (HL)	
2020	8E		ADC (HL)	; 自加
2021	27		DAA	
2022	77		LD (HL), A	
2023	23		INC HL	
2024	0D		DEC C	
2025	C21F20		JP NZ, BLOOP	
2028	10E2		DJNZ CLOOP	
202A	C9		RET	

框图:

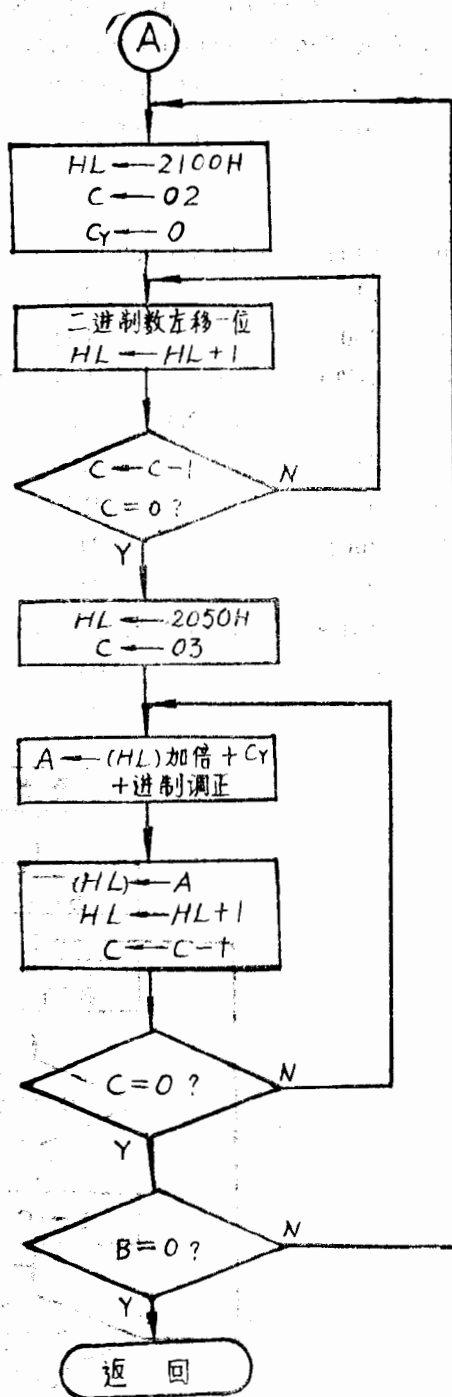
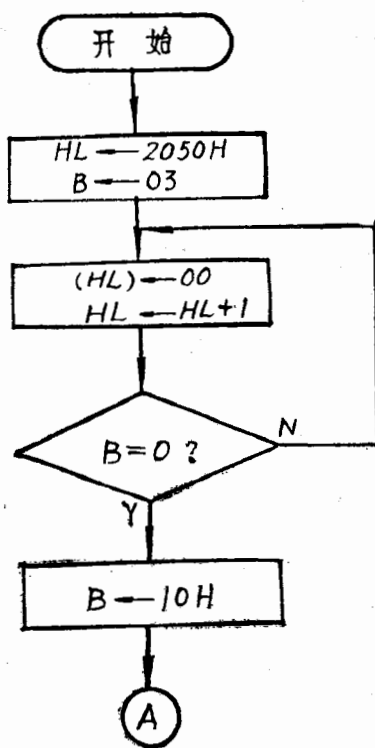


图1-8

### 1-9 二进制数→ASCII码字符串转换程序

- 说明：1. 将存在2041H单元的8位二进制数转换成8位ASCII字符(ASCII 0或ASCII 1)，存放到2042—2049H区域，高位在2042H单元。
2. 这种转换常用于ASCII码设备上以二进制形式打印数字。
3. 转换的方法是：在二进制数上加ASCII 0就行了。

程序：

```

2000  214120  BINASC; LD    HL, 2041H
2003  7E      LD    A, (HL)
2004  0608    LD    B, 08      ; 字的位数计数
2006  0E30    LD    C, '0'    ; C←ASCII 0
2008  23      CONV; INC  HL
2009  71      LD    (HL), C
200A  17      RLA
200B  3001    JR    NC,COUNT ; Cy=0? 否,转COUNT
200D  34      INC  (HL)
200E  10F8    COUNT; DJNZ CONV
2010  C9      RET
    
```

框图：

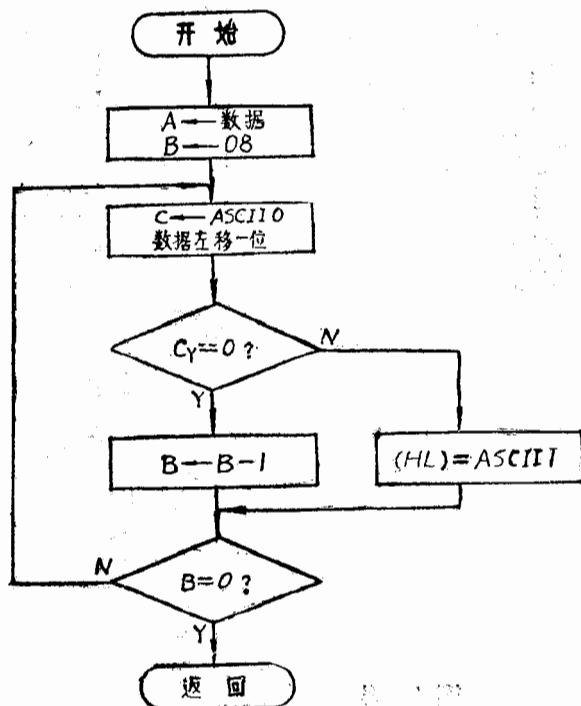


图 1-9

### 1—10 一位十六进制数→ASCII码转换程序

说明：将累加器里的一位十六进制数转换为ASCII码

程序：

```

2000 E60F UBASC: AND 0FH
2002 C690 ADD A, 90H
2004 27 DAA
2005 CE40 ADC A, 40H
2007 27 DAA
2008 C9 RET
    
```

### 1—11 十进制(BCD)→七段代码转换程序

说明：1. 将2080H单元的内容转换成七段代码，结果存入2082H单元。若2080H单元的内容不是一个BCD数，则2082H单元清0。

2. 七段代码为40H(0), 79H(1), 24H(2), 30H(3), 19H(4), 12H(5), 02H(6), 78H(7), 00H(8), 18H(9),

程序：

```

2000 0600 LD B, 0
2002 3A8020 LD A, (2080H)
2005 FE0A CP 10
2007 3008 JR NC, DONE
2009 6F LD L, A
200A 2600 LD H, 0
200C 112020 LD DE, SSEG
200F 19 ADD HL, DE
2010 46 LD B, (HL)
2011 78 DONE: LD A, B
2012 328220 LD (2082H), A
2015 76 HALT
2020 3F065B4F SSEG: DB 40H, 79H, 24H, 30H, 19H
66
2025 6D7D077F DB 12H, 02H, 78H, 00H, 18H
6F
    
```



框图:

七段显示器示意图:

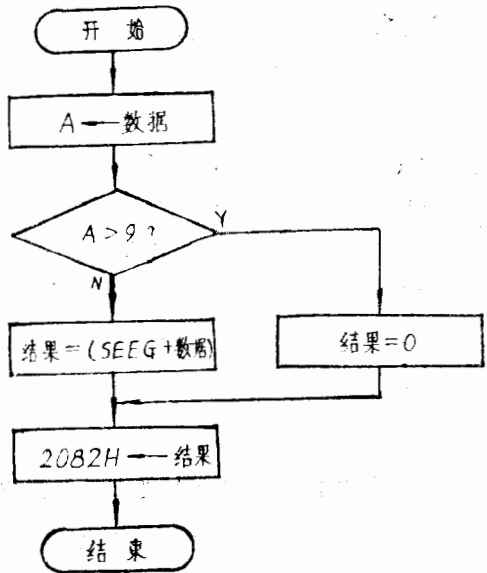
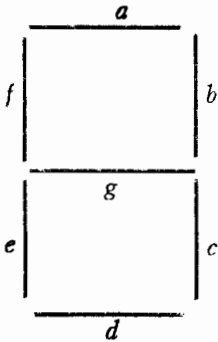


图 1-11

### 1-12 十六进制→七段代码转换程序

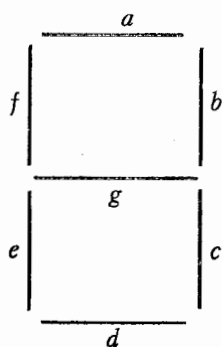
说明: 被转换的两位十六进制数存在累加器A中, 转换的结果存到HL寄存器中。

程序:

2000	0E02	START:	LD C, 02H
2002	1600		LD D, 00H
2004	47		LD B, A
2005	E60F		AND 0FH
2007	DD212220	LCOP:	LD IX, SEGPT
200B	5F		LD E, A
200C	DD19		ADD IX, DE
200E	DD7E00		LD A, (IX+0)
2011	0D		DEC C
2012	280C		JR Z, EPR
2014	6F		LD L, A
2015	78		LD A, B
2016	CB3F		SRL A
2018	CB3F		SRL A
201A	CB3F		SRL A
201C	CB3F		SRL A

201E	18E7		JR LOOP
2020	67	EPR:	LD H, A
2021	76		HALT
2022	40792430	SEGPT:	DB 40H, 79H, 24H, 30H
2026	19120278		DB 19H, 12H, 02H, 78H
202A	00180803		DB 00H, 18H, 08H, 03H
202E	4621060E		DB 46H, 21H, 06H, 0EH

七段显示器示意图:



框图:

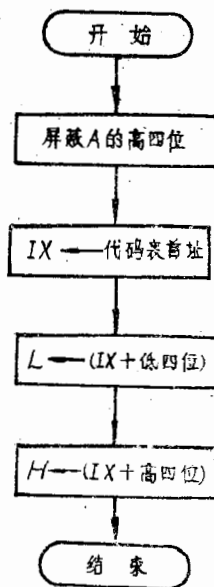


图1-12

### 1-13 十六位二进制数→BCD码数转换程序II

- 说明:
1. 16位二进制无符号数能表示的最大数为 65535，而本程序中只取 4 位十进制数，只有两个字节的BCD数。
  2. 即转换的方法是把16位二进制数分别除以 1000, 100, 10三个十进数的二进制数 (即03E8H, 0064H, 000AH), 所得的三个商和一个余数即四位BCD码。
  3. 被转换的二进制数存放到 2577H 单元和2578H单元中去, 将03E8H, 0064H 和 000AH分别存放到2595H~2598H单元中去, 转换的结果存放在 2591H 和2592H 单元中。

程序:

2000	2A7725	BBCD:	LD HL, (2577H)	; 被转换的二进制数送HL
2003	FD219525		LD IY, 2595H	; 放1000, 100, 10三个数首址

2007	DD219125		LD IX,2591H	; BCD数首址
200B	0E00	LC0:	LD C,00H	; C作计数器,记下减的次数
200D	FD5E00		LD E,(IY)	; 1000、100、10,三常数送DE
2010	FD23		INC IY	
2012	ED5600		LD D,(IY)	
2015	FD23		INC IY	
2017	B7	LC1:	OR A	; 从HL中分别减去1000、100、10,不够减时转LC2
2018	ED52		SBC HL,DE	
201A	3807		JR C,LC2	
201C	79		LD A,C	; 够减时,继续减,并记下减过的次数
201D	C601		ADD A,01H	
201F	27		DAA	
2020	4F		LD C,A	
2021	18F4		JR LC1	
2023	19	LC2:	ADD HL,DE	; 恢复HL,保存C
2024	DD7100		LD (IX),C	
2027	DD23		INC IX	
2029	7B		LD A,E	
202A	FE0A		CP 10	
202C	20DD		JR NZ,LC0	
202E	DD7500		LD (IX),L	
2031	0E02		LD C,02H	; 将4个单独的BCD数合并后放入2591H和2592H两单元之中
2033	DD219125		LD IX,2591H	
2037	FD219125		LD IY,2591H	
203B	DD7E00	LC3:	LD A,(IX)	
203E	DD23		INC IX	
2040	0604		LD B,04H	
2042	CB07	LC4:	RLC A	
2044	10FC		DJNZ,LC4	
2046	DD8600		ADD A,(IX)	
2049	DD23		INC IX	
204B	FD7700		LD (IY),A	
204E	FD23		INC IY	

2050 0D  
 2051 20E8  
 2053 C9

DEC C  
 JR NZ,LC3  
 RET

框图:

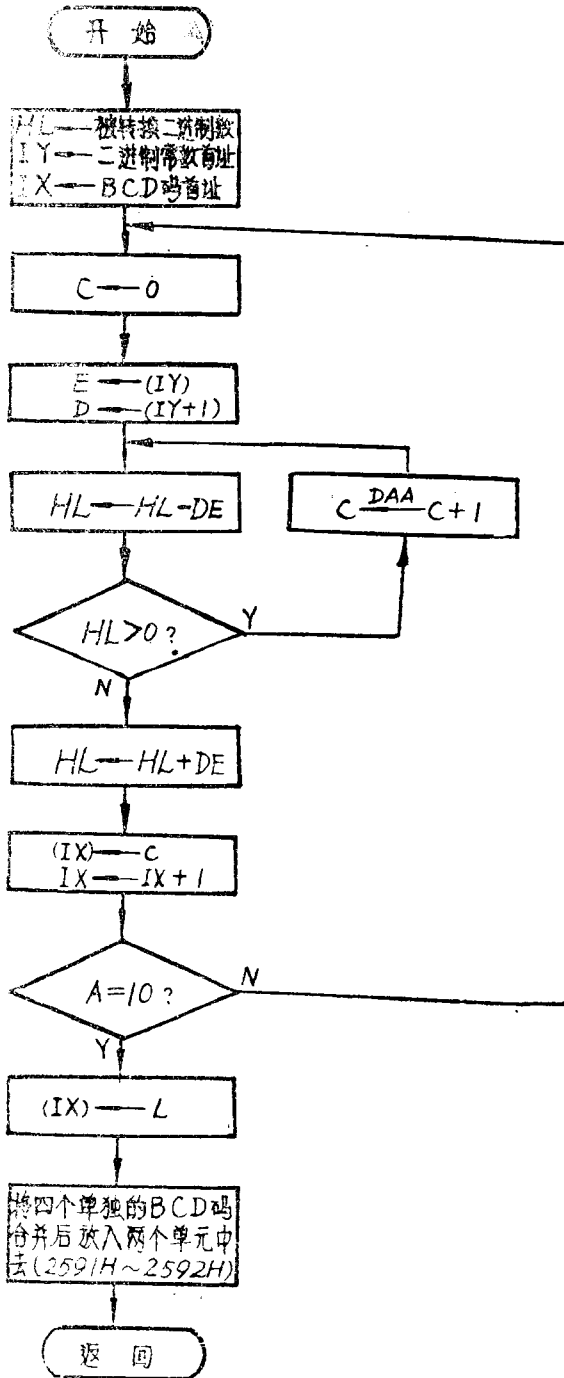


图 1—13

## 第二章 二进制定点运算

### 2—1 多字节加法

- 说明：1. HL指示被加数数据区末址；DE指示加数数据区末址；B指示数据区长度。  
2. 数据存放形式为低地址单元存高字节数据。  
3. 和值送往被加数数据区域。  
4. 本子程序使用A、BC、HL、DE，破坏AF

程序：

2000	E5	START:	PUSH HL	; 保护参数
2001	D5		PUSH DE	; 保护参数
2002	C5		PUSH BC	; 保护参数
2003	B7		OR A	; 清Cy
2004	7A	L1:	LD A, (DE)	
2005	8E		ADC A, (HL)	
2006	77		LD (HL), A	
2007	1B		DEC DE	
2008	2B		DEC HL	
2009	10F9		DJNZ L1	; 计算和
200B	C1		POP BC	; 恢复参数
200C	D1		POP DE	; 恢复参数
200D	E1		POP HL	; 恢复参数
200E	C9		RET	

### 2—2 多字节减法

- 说明：1. HL指示减数数据区首址；DE指示被减数数据区首址。  
2. 数据存放形式为字节数、数据（低地址存低字节）。  
3. 差值送往被减数区域。  
4. 本子程序使用A、B、HL、DE，破坏AF、B。

程序：

2000	E5	START:	PUSH HL	
2001	D5		PUSH DE	
2002	46		LD B, (HL)	; B作为字节计数器

2003	B7		OR A	, 清Cy
2004	23	L1:	INC HL	
2005	13		INC DE	
2006	1A		LD A, (DE)	
2007	9E		SBC A, (HL)	
2008	12		LD (DE), A	
2009	10F9		DJNZ L1	; 计算差值
200B	D1		POP DE	
200C	E1		POP HL	
200D	C9		RET	

### 2-3 单字节无符号数乘法

- 说明: 1. H放乘数, E放被乘数。  
 2. 积为双字节, 放HL。  
 3. 本子程序使用B、HL、DE, 破坏F、B、L、D。

程序:

2000	0608	START:	LD B, 8
2002	1600		LD D, 0
2004	6A		LD L, D
2005	29	L1:	ADD HL, HL
2006	3001		JR NC, L2
2008	19		ADD HL, DE
2009	10FA	L2:	DJNZ L1
200B	C9		RET

框图:

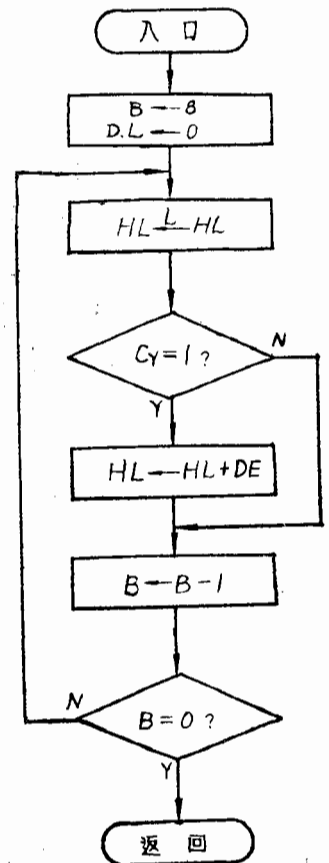


图2-3

## 2-4 单字节有符号数乘法(纯小数)

- 说明：1. L 放乘数；D 放被乘数。  
 2. 积为双字节，放HL。  
 3. 本子程序使用B、HL、DE，破坏B、H、E。

框图：

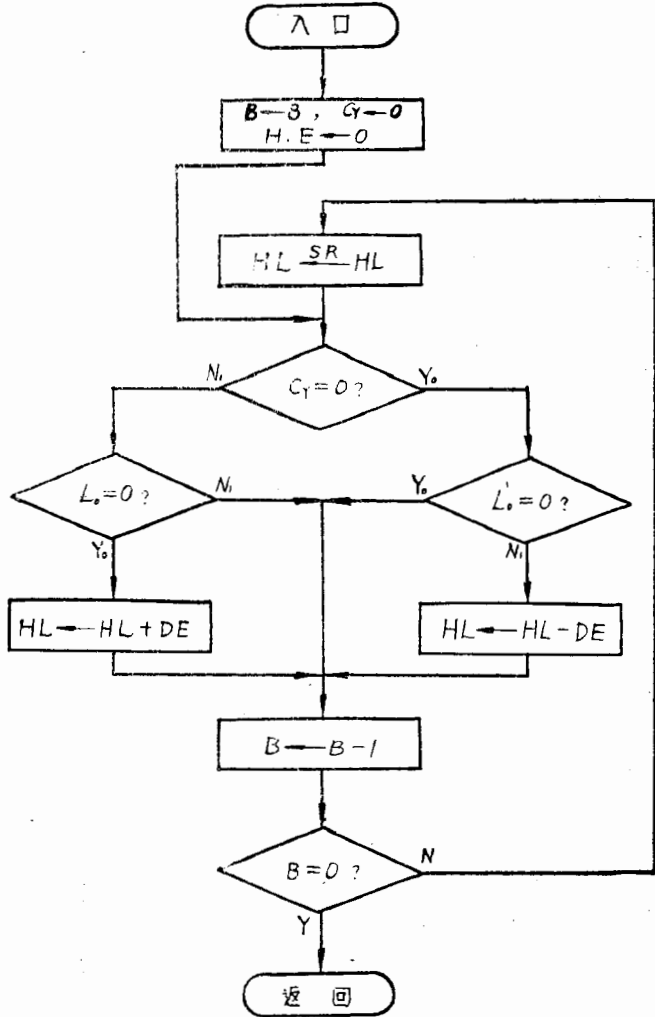


图2-4

程序：

2000	0608	START:	LD B, 8
2002	2600		LD H, 0
2004	5C		LD E, H

2005	B7		OR A
2006	1804		JR L2
2008	CB2C	L1:	SRA H
200A	CB1D		RR L
200C	3007	L2:	JR NC, L3
200E	CB45		BIT 0, L
2010	2009		JR NZ, L4
2012	19		ADD HL, DE
2013	1806		JR L4
2015	CB45	L3:	BIT 0, L
2017	2802		JR Z, L4
2019	ED52		SBC HL, DE
201B	10EB	L4:	DJNZ L1
201D	C9		RET

### 2—5 单字节有符号数乘法(整数)

- 说明: 1. L放乘数, D放被乘数。  
 2. 积为双字节, 放HL。  
 3. 本子程序使用B、HL、DE, 破坏B、H、E。

程序:

2000	0608	START:	LD B, 8
2002	2600		LD H, 0
2004	5C		LD E, H
2005	B7		OR A
2006	3007	L1:	JR NC, L2
2008	CB45		BIT 0, L
200A	200D		JR NZ, L3
200C	19		ADD HL, DE
200D	180A		JR L3
200F	CB45	L2:	BIT 0, L
2011	2806		JR Z, L3
2013	ED52		SBC HL, DE
2015	CB2C		SRA H
2017	CB1D		RR L
2019	10EB	L3:	DJNZ L1
201B	C9		RET



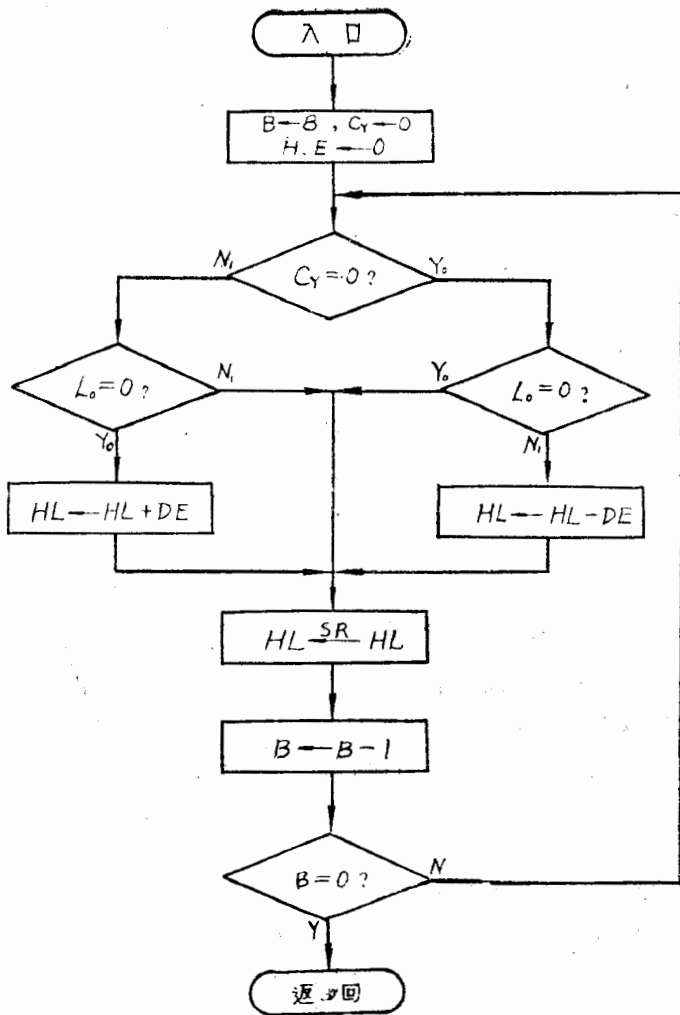


图 2-5

## 2—6 多字节无符号数乘法

- 说明：
1. HL 指示乘数数据区首址；DE 指示被乘数数据区首址；C 指示数据区长度。
  2. 数据存放形式为低地址存高字节数。
  3. 积为两倍因子的字节数，由 C 指出。
  4. 本子程序使用 A、BC、HL、DE，破坏 AF、B。

程序：

```

2000 E5      START: PUSH HL
2001 D5      PUSH DE
2002 AF      XOR A
  
```

2003	41		LD B, C
2004	1B	L1:	DEC DE
2005	12		LD (DE), A
2006	10FC		DJNZ L1
2008	09		ADD HL, BC
2009	41		LD B, C
200A	77	L2:	LD (HL), A
200B	23		INC HL
200C	10FC		DJNZ L2
200E	CB21		SLA C
2010	EB		EX DE, HL
2011	09		ADD HL, BC
2012	EB		EX DE, HL
2013	79		LD A, C
2014	CB27		SLA A
2016	CB27		SLA A
2018	F5	L3:	PUSH AF
2019	E5		PUSH HL
201A	41		LD B, C
201B	2B	L4:	DEC HL
201C	CB16		RL (HL)
201E	10FB		DJNZ L4
2020	E1		POP HL
2021	300D		JR NC, L6
2023	41		LD B, C
2024	B7		OR A
2025	2B	L5:	DEC HL
2026	1B		DEC DE
2027	1A		LD A, (DE)
2028	8E		ADC A, (HL)
2029	77		LD (HL), A
202A	10F9		DJNZ L5
202C	09		ADD HL, BC
202D	EB		EX DE, HL
202E	09		ADD HL, BC
202F	EB		EX DE, HL
2030	F1	L6:	POP AF
2031	3D		DEC A

2032 20E4  
 2034 D1  
 2035 E1  
 2036 C9

JR NZ, L3  
 POP DE  
 POP HL  
 RET

框图:

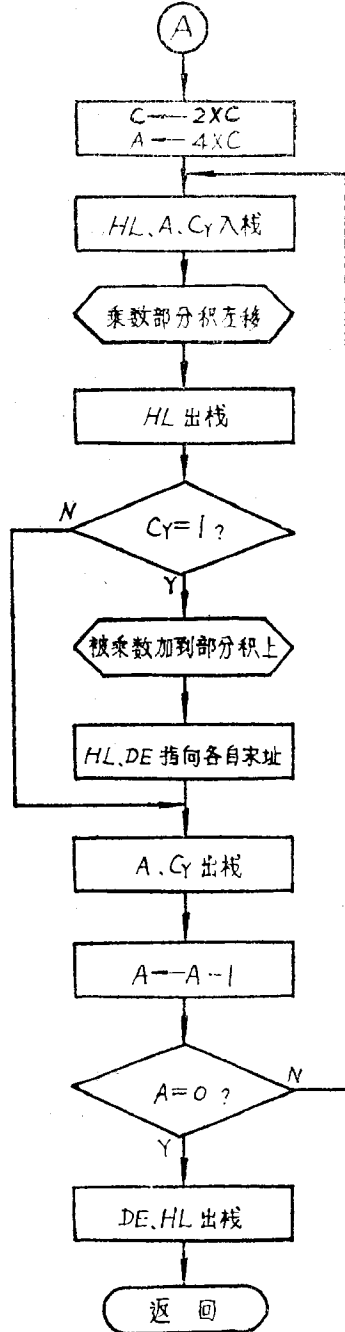
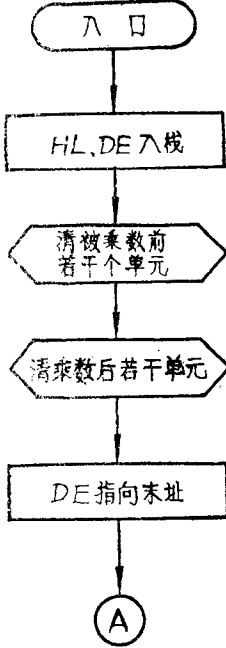


图 2-6

## 2—7 多字节有符号数乘法

说明：数据是以补码形式存放的，其它同2—6说明。

程序：

2000	D5	START:	PUSH DE
2001	E5		PUSH HL
2002	3E00		LD A, 0
2004	CD4C20		CALL FC2
2007	EB		EX DE, HL
2008	CD4C20		CALL FC2
200B	EB		EX DE, HL
200C	F5		PUSH AF
200D	AF		XOR A
200E	41		LD B, C
200F	1B	L1:	DEC DE
2010	12		LD (DE), A
2011	10FC		DJNZ L1
2013	09		ADD HL, BC
2014	41		LD B, C
2015	77	L2:	LD (HL), A
2016	23		INC HL
2017	10FC		DJNZ L2
2019	CB21		SLA C
201B	EB		EX DE, HL
201C	09		ADD HL, BC
201D	EB		EX DE, HL
201E	79		LD A, C
201F	CB27		SLA A
2021	CB27		SLA A
2023	F5	L3:	PUSH AF
2024	E5		PUSH HL
2025	41		LD B, C
2026	2B	L4:	DEC HL
2027	CB16		RL (HL)
2029	10FB		DJNZ L4
202B	E1		POP HL
202C	300D		JR NC, L6
202E	41		LD B, C

202F	B7		OR A
2030	2B	L5:	DEC HL
2031	1B		DEC DE
2032	1A		LD A, (DE)
2033	8E		ADC A, (HL)
2034	77		LD (HL), A
2035	10F9		DJNZ L5
2037	09		ADD HL, BC
2038	EB		EX DE, HL
2039	09		ADD HL, BC
203A	EB		EX DE, HL
203B	F1	L6:	POP AF
203C	3D		DEC A
203D	20E4		JR NZ, L3
203F	F1		POP AF
2040	E1		POP HL
2041	D1		POP DE
2042	CD4620		CALL FC1
2045	C9		RET
2046	CB47	FC1:	BIT 0, A
2048	281A		JR Z, LL5
204A	1805		JR LL1+1
204C	CB7E	FC2:	BIT 7, (HL)
204E	2814		JR Z, LL5
2050	3C	LL1:	INC A
2051	#5		PUSH AF
2052	0600		LD B, 0
2054	09		ADD HL, BC
2055	41		LD B, C
2056	B7		OR A
2057	2B	LL2:	DEC HL
2058	7E		LD A, (HL)
2059	3804		JR C, LL3
205B	ED44		NEG
205D	1801		JR LL4
205F	2F	LL3:	CPL
2060	77	LL4:	LD (HL), A
2061	10F4		DJNZ LL2

2063 F1

2064 C9

POP AF

LL5; RET

框图:

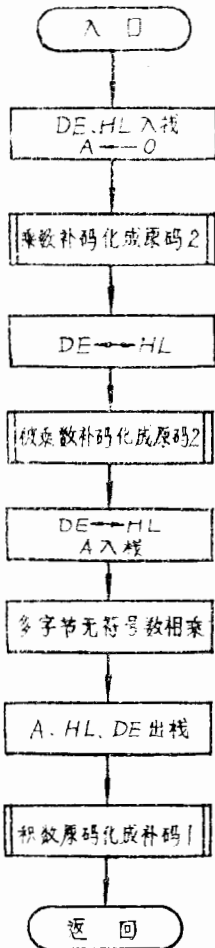


图 2-7-1

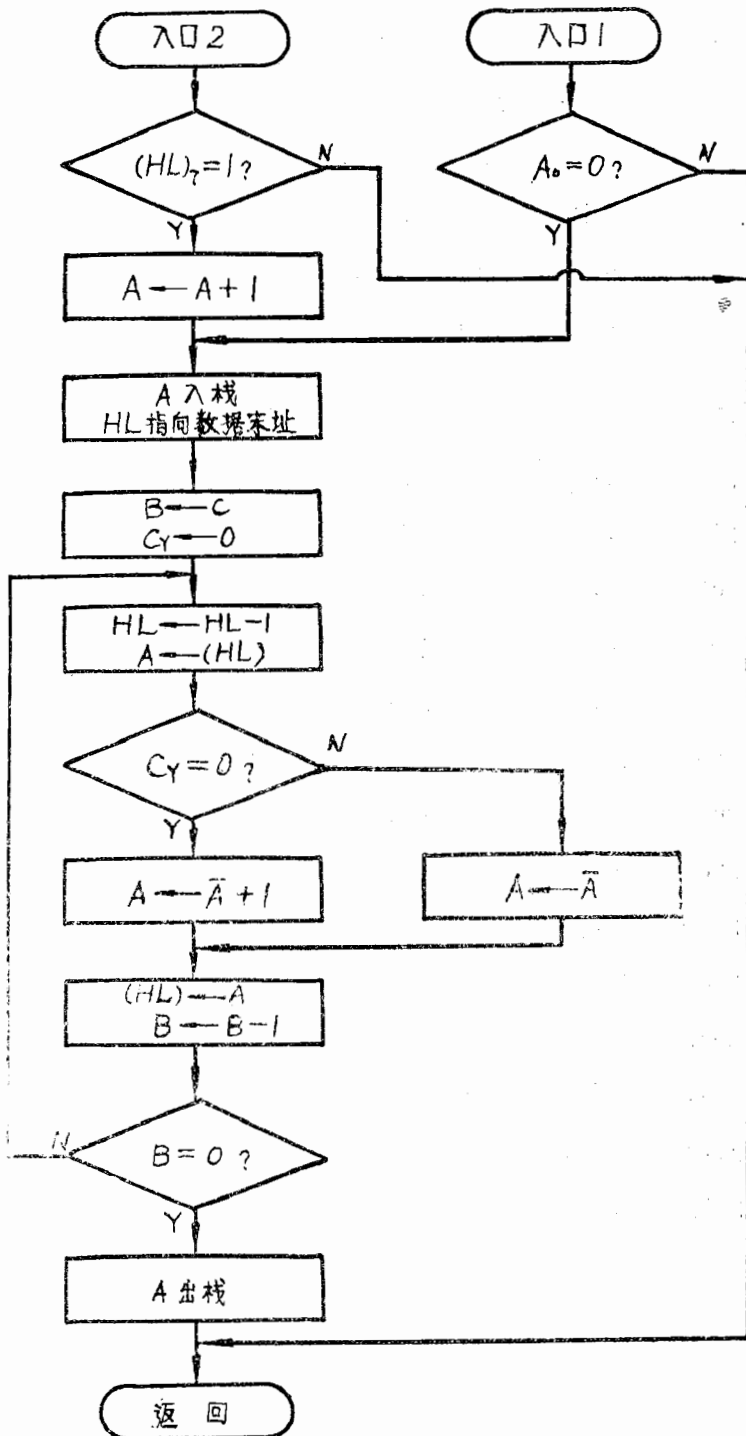


图 2-7-2

## 2-8 单字节无符号数除法(整除)

框图:

- 说明: 1. C放除数; L放被除数。  
 2. 出口时L放商; H放余数。  
 3. 当除数为0时, 出错(A=00表示) 返回, 否则A=FFH。  
 4. 本子程序使用A、BC、HL, 破坏AF、B、H。

程序:

2000	AF	XOR A
2001	B9	CP C
2002	280E	JR Z, L3
2004	67	LD H, A
2005	0608	LD B, 8
2007	29	L1: ADD HL, HL
2008	7C	LD A, H
2009	91	SUB C
200A	3802	JR C, L2
200C	67	LD H, A
200D	2C	INC L
200E	10F7	L2: DJNZ L1
2010	3EFF	LD A, 0FFH
2012	C9	L3: RET

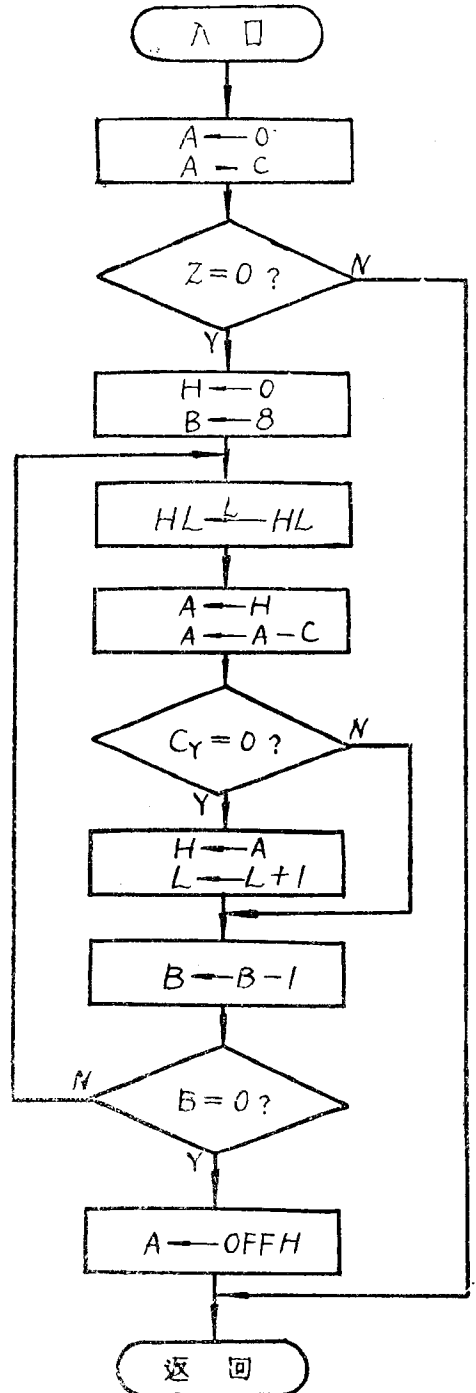


图 2-8

## 2-9 单字节有符号数除法（一）

- 说明：
1. HL放被除数；C放除数。
  2. 出口时L放商；H放余数。
  3. 当除数为0时，出错返回（以A = 00H表示），否则A = FFH。
  4.  $|被除数H| < |除数|$ ，否则结果错误。
  5. 本子程序使用A、BC、HL，破坏AF、B。

程序：

```

2000  AF          START: XOR  A
2001  81          ADD  A, C
2002  2823       JR   Z, L6
2004  0608       LD   B, 8
2006  7C         LD   A, H
2007  A9         XOR  C
2008  CB7F       BIT  7, A
200A  200A       JR  NZ, L3
200C  1804       JR  L2
200E  CB45       L1:  BIT  0, L
2010  2804       JR  Z, L3
2012  7C         L2:  LD   A, H
2013  91         SUB  A, C
2014  1802       JR  L4
2016  7C         L3:  LD   A, H
2017  81         ADD  A, C
2018  67         L4:  LD   H, A
2019  A9         XOR  C
201A  CB7F       BIT  7, A
201C  2001       JR  NZ, L5
201E  37         SCF
201F  ED6A       L5:  ADC  HL, HL
2021  10EB       DJNZ L1
2023  3EFF       LD   A, 0FFH
2025  CB3C       SRL  H
2027  C9         L6:  RET

```



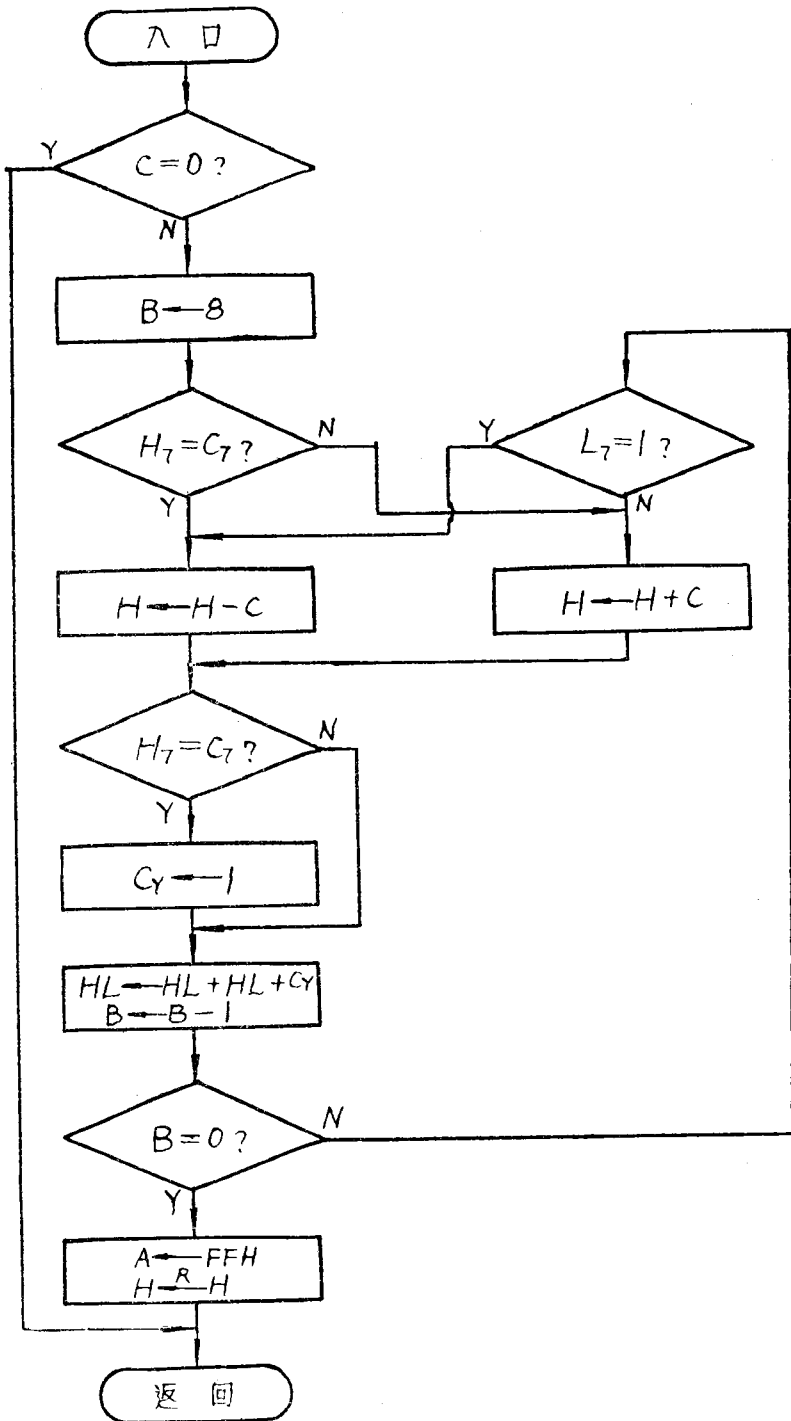


图 2-9

## 2-10 单字节有符号数除法(二)

- 说明:
1. HL放被除数;C放除数。
  2. 出口时L放商;H放余数。
  3. 当除数为0时,不运算直接返回。
  4. 要求 $|被除数H| < |除数|$ 。
  5. 本子程序使用A、BC、HL、D,破坏AF、B、D。

程序:

```

2000  AF      START: XOR  A
2001  81          ADD  A, C
2002  2835      JR   Z, L9
2004  7C          LD   A, H
2005  A9          XOR  C
2006  CB7F      BIT   7, A
2008  7C          LD   A, H
2009  2003      JR   NZ, L1
200B  91          SUB  A, C
200C  1801      JR   L2
200E  81      L1:  ADD  A, C
200F  67      L2:  LD   H, A
2010  0607          LD   B, 7
2012  1600          LD   D, 0
2014  180C      JR   L4
2016  AF      L3:  XOR  A
2017  84          ADD  A, H
2018  2008      JR   NZ, L4
201A  29          ADD  HL, HL
201B  82          ADD  A, D
201C  2012      JR   NZ, L7
201E  2C          INC  L
201F  14          INC  D
2020  180E      JR   L7
2022  7C      L4:  LD   A, H
2023  A9          XOR  C
2024  CB7F      BIT   7, A
2026  29          ADD  HL, HL

```

(下转第31页)

框图:

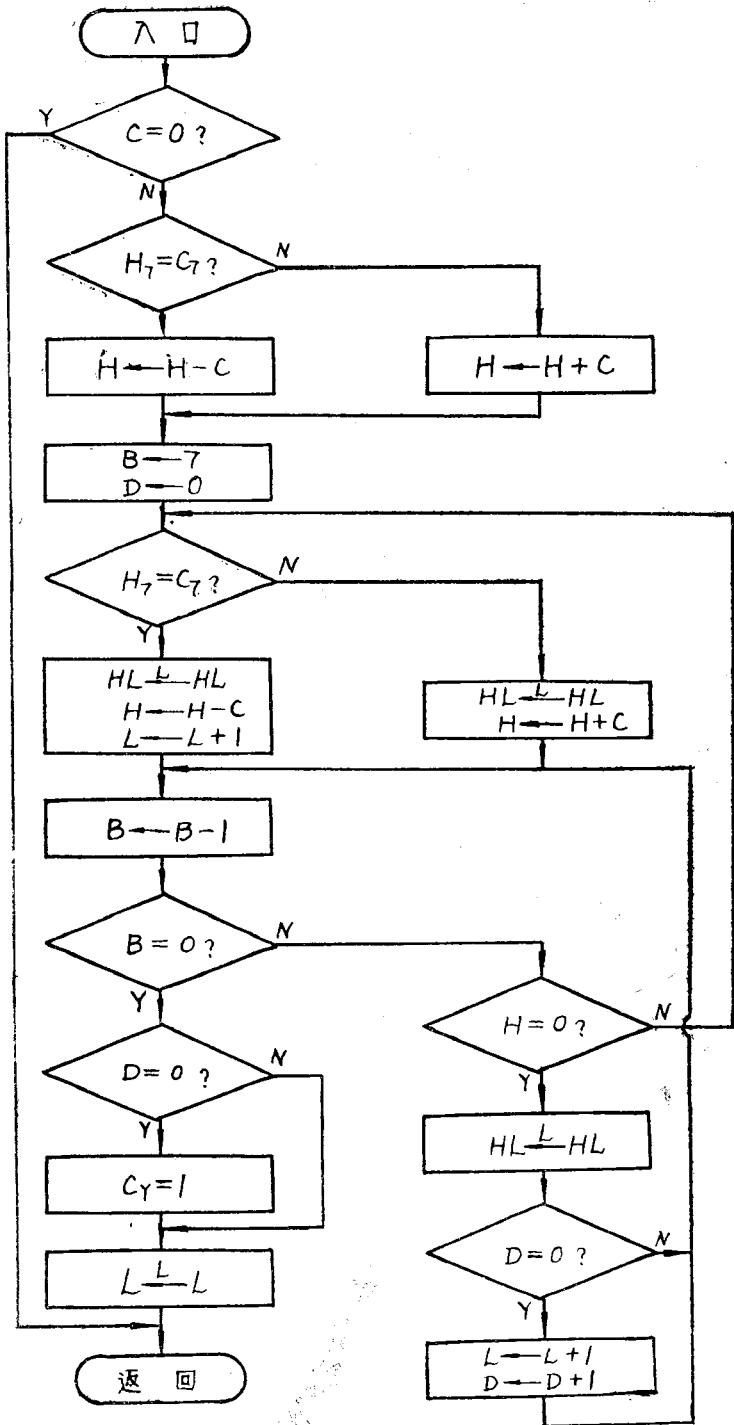


图 2-10

(上接第29页)

2027	7C		LD A, H
2028	2004		JR NZ, L5
202A	91		SUB A, C
202B	2C		INC L
202C	1801		JR L6
202E	81	L5:	ADD A, C
202F	67	L6:	LD H, A
2030	10E4	L7:	DJNZ L3
2032	AF		XOR A
2033	82		ADD A, D
2034	2001		JR NZ, L8
2036	37		SCF
2037	CB15	L8:	RL L
2039	C9	L9:	RET

#### 2—11 多字节无符号数除法(整除)

- 说明:
1. HL指示被除数数据区首址; DE指示除数数据区首址; C指示数据区长度。
  2. 数据存放形式为低地址存高字节数。
  3. 出口时HL指示商数据区首址; 余数在商数据区的前若干单元区域中; 长度均由C指出。
  4. 当除数为0时, 出错(B=00H表示)返回, 否则B=FFH。
  5. 本子程序使用A、BC、HL、DE, 破坏AF、B。

程序:

2000	D5	START:	PUSH DE
2001	41		LD B, C
2002	1A	L1:	LD A, (DE)
2003	13		INC DE
2004	B7		OR A
2005	10FB		DJNZ L1
2007	2847		JR Z, L9
2009	AF		XOR A
200A	41		LD B, C
200B	2B	L2:	DEC HL
200C	77		LD (HL), A
200D	10FC		DJNZ L2

200F	CB21		SLA C
2011	09		ADD HL, BC
2012	41		LD B, C
2013	CB20		SLA B
2015	CB20		SLA B
2017	C5	L3:	PUSH BC
2018	E5		PUSH HL
2019	D5		PUSH DE
201A	41		LD B, C
201B	2B	L4:	DEC HL
201C	CB16		RL (HL)
201E	10FB		DJNZ L4
2020	CB39		SRL C
2022	09		ADD HL, BC
2023	41		LD B, C
2024	EB		EX DE, HL
2025	2B	L5:	DEC HL
2026	1B		DEC DE
2027	1A		LD A, (DE)
2028	9E		SBC A, (HL)
2029	12		LD (DE), A
202A	10F9		DJNZ L5
202C	EB		EX DE, HL
202D	300F		JR NC, L7
202F	F5		PUSH AF
2030	09		ADD HL, BC
2031	EB		EX DE, HL
2032	09		ADD HL, BC
2033	EB		EX DE, HL
2034	41		LD B, C
2035	B7		OR A
2036	2B	L6:	DEC HL
2037	1B		DEC DE
2038	1A		LD A, (DE)
2039	8E		ADC A, (HL)
203A	77		LD (HL), A
203B	10F9		DJNZ L6

(下转第34页)

框图:

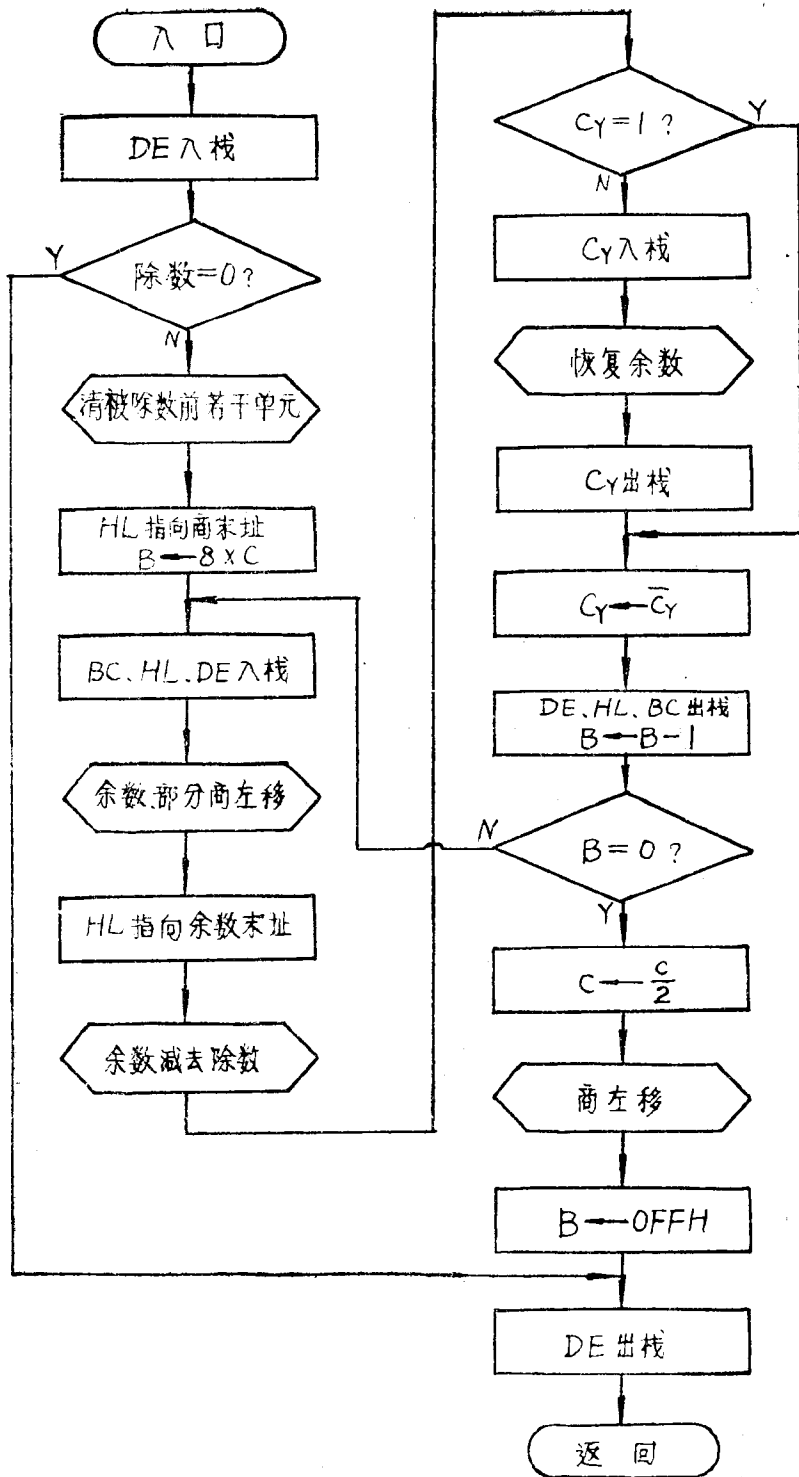


图 2-11

(上接第32页)

203D	F1		POP AF
203E	3F	L7:	CCF
203F	D1		POP DE
2040	E1		POP HL
2041	C1		POP BC
2042	10D3		DJNZ L3
2044	F5		PUSH AF
2045	CB39		SRL C
2047	F1		POP AF
2048	41		LD B, C
2049	2B	L8:	DEC HL
204A	CB16		RL (HL)
204C	10FB		DJNZ L8
204E	06FF		LD B, 0FFH
2050	D1	L9:	POP DE
2051	C9		RET

## 2—12 多字节有符号数除法(整除)

说明: 数据以补码形式存放, 其它同2—11说明。补码 $\longleftrightarrow$ 原码的程序框图参考图2—7—2。

程序:

2000	D5	START:	PUSH DE
2001	41		LD B, C
2002	1A	L1:	LD A, (DE)
2003	13		INC DE
2004	B7		OR A
2005	10FB		DJNZ L1
2007	2857		JR Z, L9
2009	D1		POP DE
200A	D5		PUSH DE
200B	AF		XOR A
200C	CD6820		CALL FC2
200F	EB		EX DE, HL
2010	CD6820		CALL FC2
2013	<del>BEEB</del>		EX DE, HL
2014	F5		PUSH AF

2015	AF		XOR A
2016	41		LD B, C
2017	2B	L2:	DEC HL
2018	77		LD (HL), A
2019	10FC		DJNZ L2
201B	CB21		SLA C
201D	09		ADD HL, BC
201E	41		LD B, C
201F	CB20		SLA B
2021	CB20		SLA B
2023	C5	L3:	PUSH BC
2024	E5		PUSH HL
2025	D5		PUSH DE
2026	41		LD B, C
2027	2B	L4:	DEC HL
2028	CB16		RL (HL)
202A	10FB		DJNZ L4
202C	CB39		SRL C
202E	09		ADD HL, BC
202F	41		LD B, C
2030	EB		EX DE, HL
2031	2B	L5:	DEC HL
2032	1B		DEC DE
2033	1A		LD A, (DE)
2034	9E		SBC A, (HL)
2035	12		LD (DE), A
2036	10F9		DJNZ L5
2038	EB		EX DE, HL
2039	300F		JR NC, L7
203B	F5		PUSH AF
203C	09		ADD HL, BC
203D	EB		EX DE, HL
203E	09		ADD HL, BC
203F	EB		EX DE, HL
2040	41		LD B, C
2041	B7		OR A
2042	2B	L6:	DEC HL
2043	1B		DEC DE



2044	1A		LD A, (DE)
2045	8E		ADC A, (HL)
2046	77		LD (HL), A
2047	10F9		DJNZ L6
2049	F1		POP AF
204A	3F	L7:	CCF
204B	D1		POP DE
204C	E1		POP HL
204D	C1		POP BC
204E	10D3		DJNZ L3
2050	F5		PUSH AF
2051	CB39		SRL C
2053	F1		POP AF
2054	41		LD <sub>x</sub> B, C
2055	2B	L8:	DEC HL
2056	CB16		RL (HL)
2058	10FB		DJNZ L8
205A	F1		POP AF
205B	CD6220		CALL FC1
205E	06FF		LD B, 0FFH
2060	D1	L9:	POP DE
2061	C9		RET
2062	CB47	FC1:	BIT 0, A
2064	281A		JR Z, LL5
2066	1805		JR LL1+1
2068	CB7E	FC2:	BIT 7, (HL)
206A	2814		JR Z, LL5
206C	3C	LL1:	INC A
206D	F5		PUSH AF
206E	0600		LD B, 0
2070	09		ADD HL, BC
2071	41		LD B, C
2072	B7		OR A
2073	2B	LL2:	DEC HL
2074	7E		LD A, (HL)
2075	3804		JR C, LL3
2077	ED44		NEG
2079	1801		JR LL4

207B	2F	LL3:	CPL
207C	77	LL4:	LD (HL), A
207D	10F4		DJNZ LL2
207F	F1		POP AF
2080	C9	LL5:	RET

### 2-13 求单字节数的绝对值

说明：1. 设单字节数存放在2010H单元中。求出的绝对值仍放回该单元中。  
 2. 方法是利用判符号标志位S。S=0为正，S=1为负。为负则对负数求补码，得其绝对值。

程序：

2000	3A1020	LD A, (2010H)
2003	B7	OR A
2004	F20920	JP P, END
2007	ED44	NEG
2009	321020	END: LD (2010H), A
200C	76	HALT

框图：

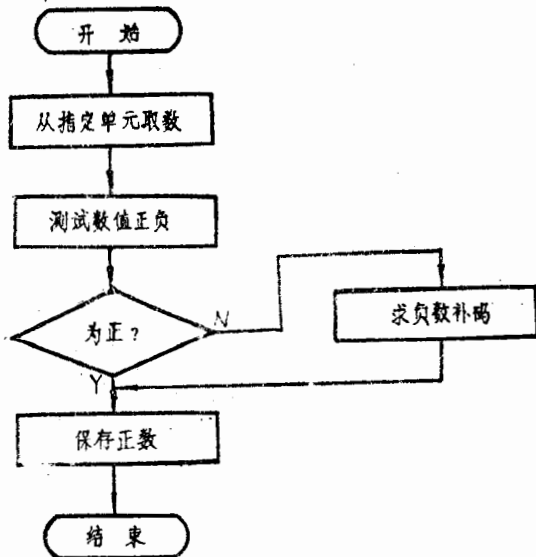


图 2-13

框图：

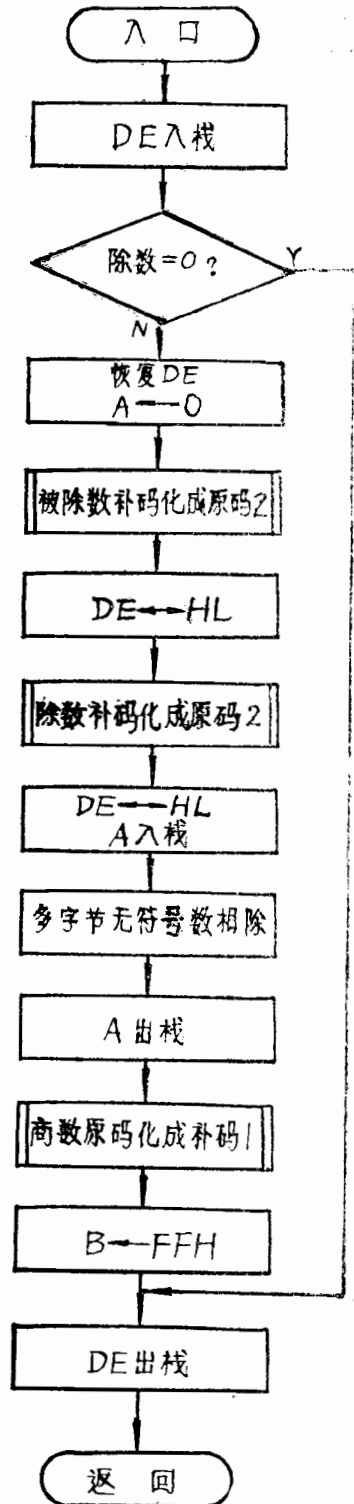


图 2-12

## 2-14 求算术平均值

- 说明：1. 本程序可以用来求若干个 ( $\leq 256$ 个) 八位二进制数的算术平均值，其和数  $\leq 16$ 位二进制数。平均值取小数点后二位十六进制数，并且有四舍五入的功能。
2. 进入程序前，所求平均值的个数存在SIZE中，而各个数存放在以BLOCK开始的连续单元中。
3. 程序执行完后，所求得平均值的整数部分存在MEANI单元中，小数部分存放在MEAN单元中。
4. 程序具有嵌套子程序DIV (除法子程序) 和MULT (乘法子程序)。使用的寄存器有AF、BC、HL、DE、IX、A'F'。

程序：

```

2000 DD215C20   START: LD IX, BLOCK      ; 置数据地址
2004 3A5B20           LD A, (SIZE)      ; 计数等
2007 47             LD B, A
2008 4F             LD C, A
2009 210000          LD HL, 0
200C 55             LD D, L
200D DD5E00   LOOP: LD E, (IX)        ; 数据送入E
2010 DD23           INC IX            ; 取下一个数
2012 19             ADD HL, DE        ; 求和
2013 10F8           DJNZ LOOP         ; 求完没有?
2015 E5             PUSH HL          ; 完, 记总数
2016 59             LD E, C
2017 CD3820          CALL DIV          ; 调用除法子程序
201A 7C             LD A, H
201B 08             EX AF, AF'
201C 65             LD H, L
201D 1E64           LD E, 64H
201F CD4F20          CALL MULT         ; 调用乘法子程序
2022 59             LD E, C
2023 CD3820          CALL DIV          ; 调用除法子程序
2026 7D             LD A, L
2027 85             ADD A, L
2028 BB             CP E
2029 3801           JR C, JMP
202B 24             INC H
202C 08             JMP; EX AF, AF'

```

2021D	6C		LD L, H	; 送平均值整数部分
202E	67		LD H, A	; 送平均值小数部分
202F	C1		POP BC	
2030	ED430022		LD (SUML), BC	; 送出和数
2034	220222		LD (MEAN), HL	; 送出平均值
2037	C9		RET	
2038	0608	DIV:	LD B, 08H	; 除法子程序
203A	AF		XOR A	
203B	ED6A	LOOP1:	ADC HL, HL	
203D	7C		LD A, H	
203E	3803		JR C, JMP1	
2040	BB		CP E	
2041	3803		JR C, JMP2	
2043	93	JMP1:	SUB E	
2044	67		LD H, A	
2045	AF		XOR A	
2046	3F	JMP2:	CCF	
2047	10F2		DJNZ, LOOP1	
2049	CB15		RL L	
204B	7C		LD A, H	
204C	65		LD H, L	
204D	6F		LD L, A	
204E	C9		RET	
204F	1600	MULT:	LD D, 00	; 乘法子程序
2051	6A		LD L, D	
2052	0608		LD B, 8H	
2054	29	LOOP2:	ADD HL, HL	
2055	3001		JR NC, JMP3	
2057	19		ADD HL, DE	
2058	10FA	JMP3:	DJNZ, LOOP2	
205A	C9		RET	
205B	n	SIZE:	DB n	; 求平均值数的个数
205C		BLOCK:	DB	; 存放待求平均值的数
2200	(0001)	SUML:	DS 1	; 和的低位
2201	(0001)	SUMH:	DS 1	; 和的高位
2202	(0001)	MEAN:	DS 1	; 平均值小数部分
2203	(0001)	MEANI:	DS 1	; 平均值整数部分

框图:

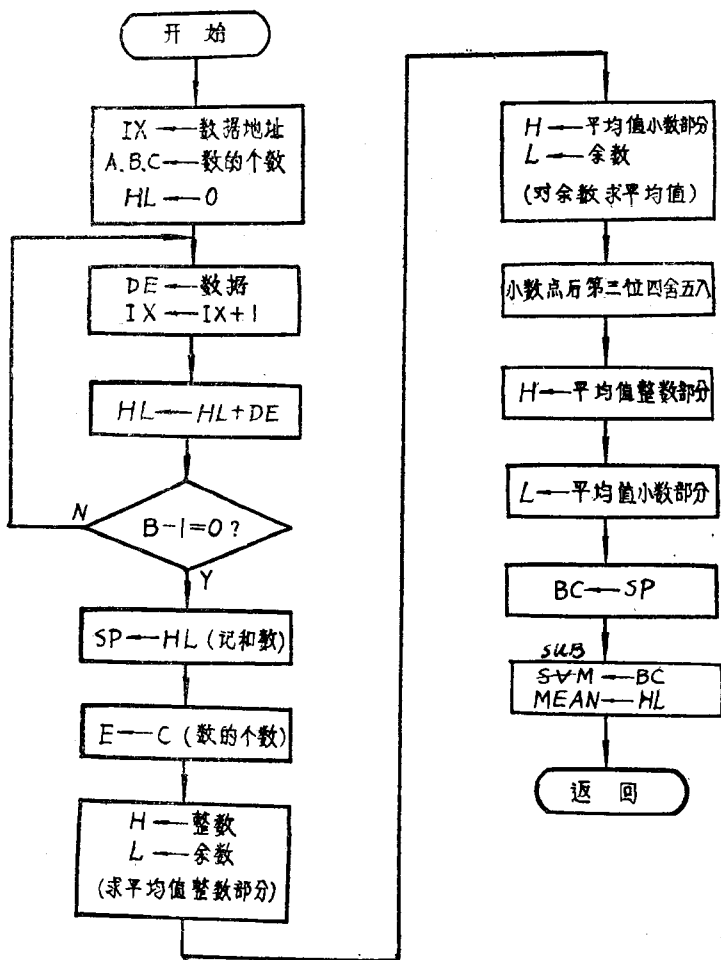


图 2-14

# 第三章 BCD 码运算

## 3-1 单字节BCD码加法

说明：2040和2041单元中分别存放两个加数。相加的和存放在2042单元中。

地址	操作数	指令
2001	3A4020	LD A,(2040H)
2004	214120	LD HL,2041H
2007	86	ADD A,(HL)
2008	27	DAA
2009	324220	LD (2042H),A
200C	76	HALT

## 3-2 多字节BCD码加法

说明：字节数存放在2040单元中。两个加数分别从2041和2051单元开始存放（先放最低位）。而和则代替从2041单元开始存放的那个加数。

地址	操作数	指令
2000	214020	LD HL,2040H
2003	46	LD B,(HL)
2004	23	INC HL
2005	1151 0	LD DE, 2051H
2008	A7	AND A
2009	1A	DECAD: LD A,(DE)
200A	8E	ADC A,(HL)
200B	27	DAA
200C	77	LD (HL),A
200D	13	INC DE
200E	23	INC HL
200F	10F8	DJNZ DECAD
2011	76	HALT

框图：

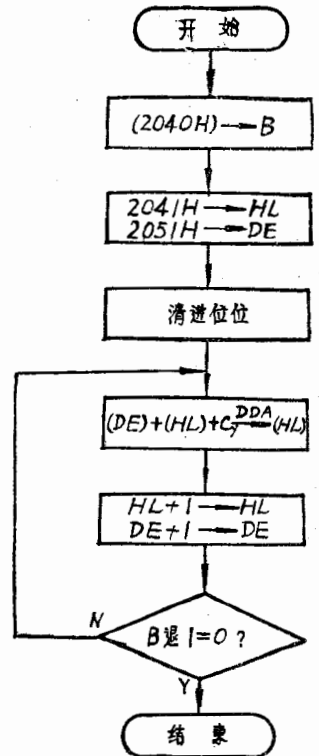


图 3-2

### 3-3 单字节BCD码减法

说明：2040和2041单元分别存放被减数和减数。相减的差存放在2042单元中。

程序：

2000	A7	AND A
2001	3A4020	LD A,(2040H)
2004	214120	LD HL,2041H
2007	96	SUB A,(HL)
2008	27	DAA
2009	324220	LD (2042H), A
200C	76	HALT

框图：

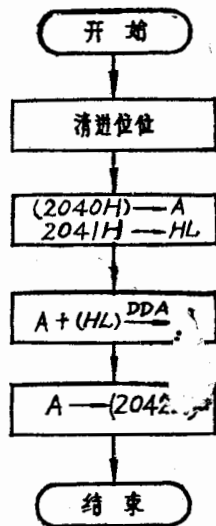


图 3-3

### 3-4 多字节BCD码减法

说明：字节数存放在2040单元中。减数和被减数分别从2041和2051单元开始存放（先放低位）。而相减的差则代替从2041单元开始存放的减数。

程序：

2000	214020	LD HL, 2040H
2003	46	LD B, (HL)
2004	23	INC HL
2005	115120	LD DE, 2051H
2008	A7	AND A
2009	1A	DECSB: LD A, (DE)
200A	9E	SBC A, (HL)
200B	27	DAA
200C	77	LD (HL), A
200D	23	INC HL
200E	13	INC DE
200F	10F8	DJNZ DECSB
2011	76	HALT

框图：

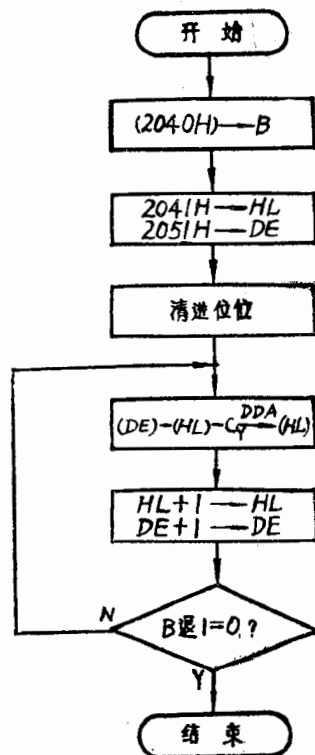


图 3-4

### 3-5 一位BCD码乘法

说明：2040单元的高四位和低四位分别存放两个乘数。而乘积取代两个乘数存放在2040单元中。

程序：

2000	AF	XOR A
2001	214020	LD HL,2040H
2004	ED67	RRD
2006	2806	JR Z,L1
2008	47	LD B,A
2009	AF	XOR A
200A	86	L0: ADD A, (HL)
200B	27	DAA
200C	10FC	DJNZ L0
200E	77	L1: LD (HL), A
200F	76	HALT

框图：

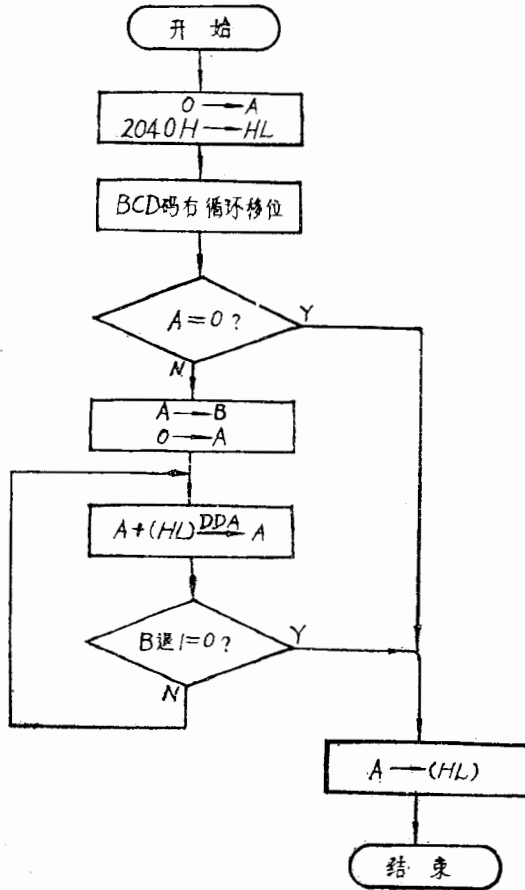


图3-5



### 3-6 单字节BCD码乘法

说明：2040和2041单元分别存放两个乘数。而积则从高位到低位取代两个乘数存放在2040和2041单元中。

程序：

2000	3A 4020	LD A, (2040H)
2003	5F	LD E, A
2004	1603	LD D, 3
2006	AF	XOR A
2007	324020	LD (2040H), A
200A	214020	X0: LD HL, 2040H
200D	ED67	RRD
200F	23	INC HL
2010	ED67	RRD
2012	280F	JR Z, X3
2014	47	LD B, A
2015	2B	DEC HL
2016	0E00	LD C, 0
2018	7E	LD A, (HL)
2019	B7	X1: OR A
201A	83	ADD A, E
201B	27	DAA
201C	3001	JR NC, X2
201E	0C	INC C
201F	10F8	X2: DJNZ X1
2021	77	LD (HL), A
2022	79	LD A, C
2023	15	X3: DEC D
2024	20E4	JR NZ, X0
2026	76	HALT

框图:

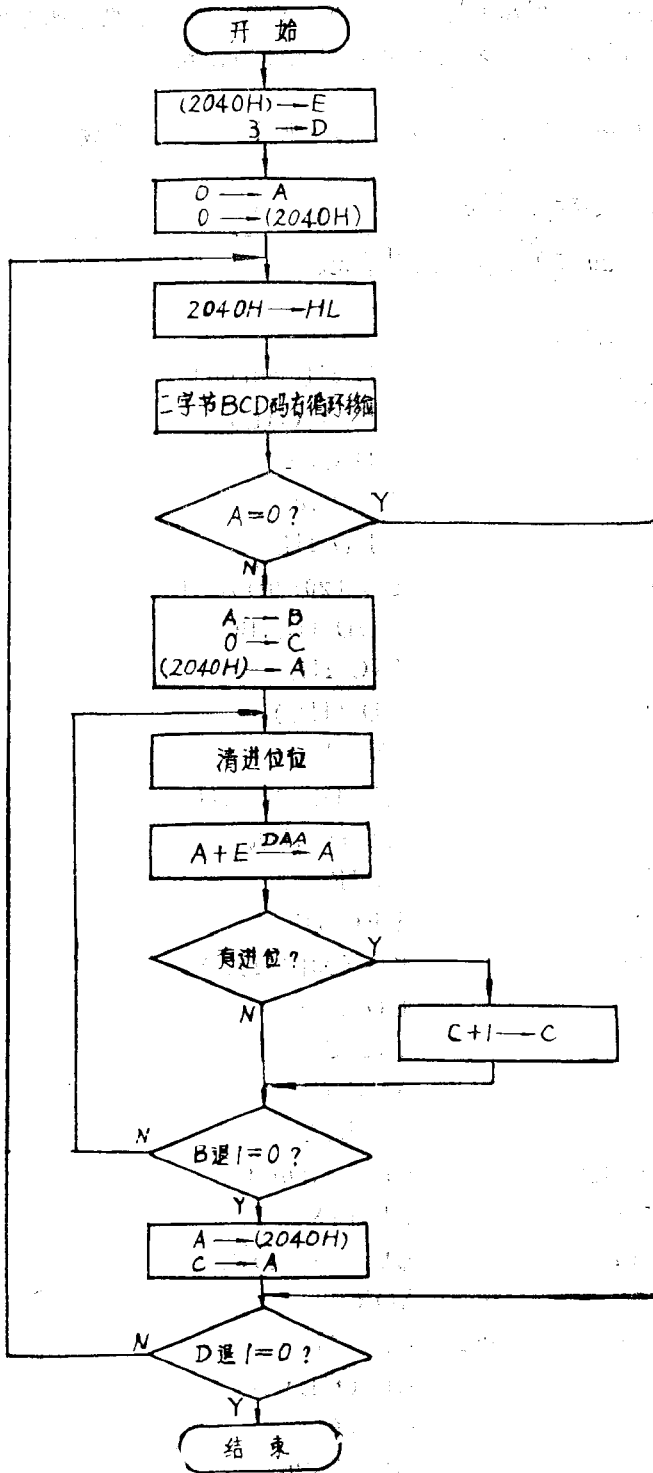
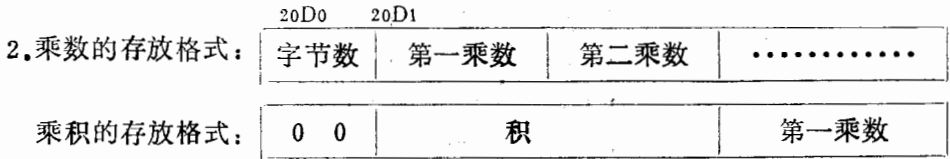


图3-6

3 ~ 7 多字节BCD码乘法

说明：1. 字节数存放在20D0单元中。从20D1单元开始存放两个乘数（从高位到低位存放）。而积取代两个乘数从20D1单元开始存放。



3. 20F0——20F5单元作为工作单元。

程序：

```

2000  21D020      LD HL,20D0H
2003  7E          LD A,(HL)
2004  4F          LD C,A
2005  0600       LD B,0
2007  09          ADD HL,BC
2008  22F020     LD (20F0H),HL
200B  09          ADD HL,BC
200C  23          INC HL
200D  3600       LD (HL),0
200F  23          INC HL
2010  11D120     LD DE,20D1H
2013  EB          EX DE,HL
2014  EDB0       LDIR
2016  1B          DEC DE
2017  ED53F220  LD (20F2H),DE
201B  4F          LD C,A
201C  0C          INC C
201D  81          ADD C
201E  47          LD B,A
201F  DD21F420  LD IX,20F4H
2023  DD7700     LD (IX),A
2026  DD7101     LD (IX+1),C
2029  AF          XOR A
202A  2B          P0: DEC HL
202B  77          LD (HL),A
202C  0D          DEC C
202D  20FB       JR NZ,P0
    
```

(下转第48页)

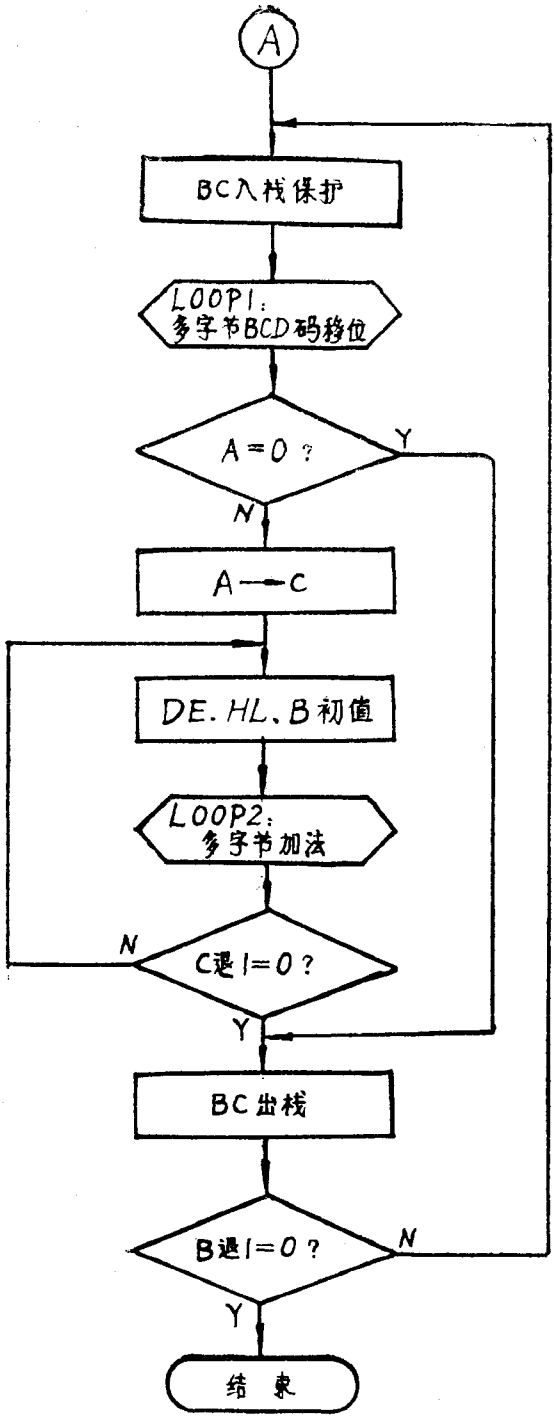
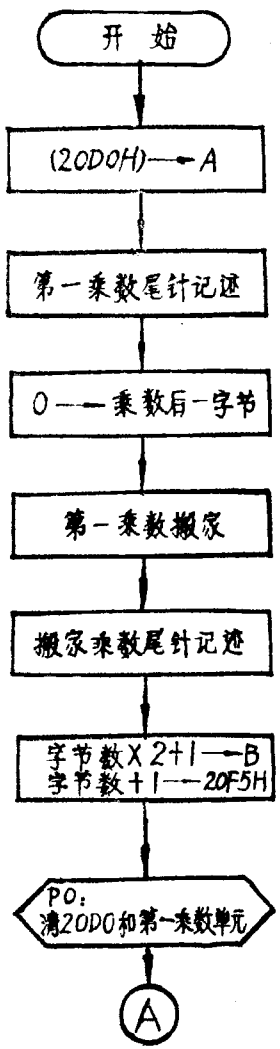


图 3-7

(上接第46页)

202F	C5	P1:	PUSH BC
2030	21D020		LD HL,20D0H
2033	DD4600		LD B,(IX)
2036	AF		XOR A
2037	ED67	LOOP1:	RRD
2039	23		INC HL
203A	10FB		DJNZ LOOP1
203C	FE00		CP 0
203E	2817		JR Z,P3
2040	4F		LD C,A
2041	ED5BF020	P2:	LD DE,(20F0H)
2045	2AF220		LD HL,(20F2H)
2048	DD4601		LD B,(IX+1)
204B	B7		OR A
204C	1A	LOOP2:	LD A,(DE)
204D	8E		ADC (HL)
204E	27		DAA
204F	12		LD (DE),A
2050	1B		DEC DE
2051	2B		DEC HL
2052	10F8		DJNZ LOOP2
2054	0D		DEC C
2055	20EA		JR NZ,P2
2057	C1	P3:	POP BC
2058	10D5		DJNZ P1
205A	76		HALT

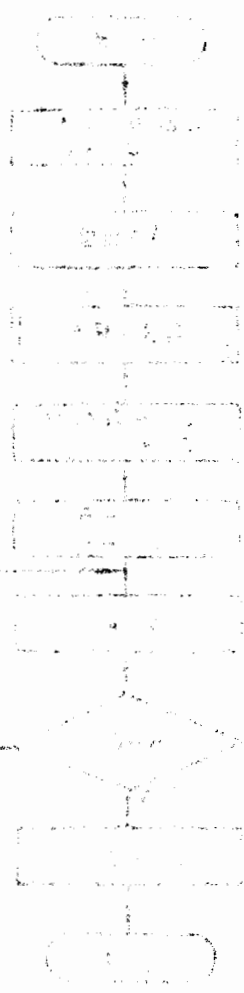
### 3—8 单字节BCD码除法

说明：2060和2061单元分别存放被除数和除数。其商的整数部分存放在2060单元，商保留两位小数（一字节）存放在2061单元中。

程序：

2000	DD216020	LD IX,2060H
2004	DD7E00	LD A,(IX)
2007	0E00	LD C,0
2009	B7	OR A

200A	CD3120		CALL BX
200D	DD7100		LD (IX),C
2010	0E00		LD C,0
2012	47		LD B,A
2013	AF		XOR A
2014	5F	B0:	LD E,A
2015	AF		XOR A
2016	80		ADD B
2017	2814		JR Z,B1
2019	78		LD A,B
201A	D601		SUB 1
201C	27		DAA
201D	47		LD B,A
201E	7B		LD A,E
201F	DD9601		SUB (IX+1)
2022	27		DAA
2023	CD4220		CALL BZ
2026	B7		OR A
2027	CD3120		CALL BX
202A	C31420		JP B0
202D	DD7101	B1:	LD (IX+1),C
2030	76		HALT
2031	DD9601	BX:	SUB (IX+1)
2034	27		DAA
2035	3806		JR C,BY
2037	CD4220		CALL BZ
203A	C33120		JP BX
203D	DD8601	BY:	ADD (IX+1)
2040	27		DAA
2041	C9		RET
2042	08	BZ:	EX AF,AF
2043	79		LD A,C
2044	C601		ADD 1
2046	27		DAA
2047	4F		LD C,A
2048	08		EX AF,AF
2049	C9		RET



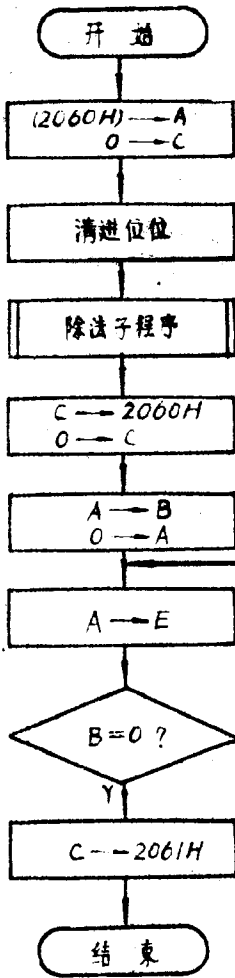


图3-8-1

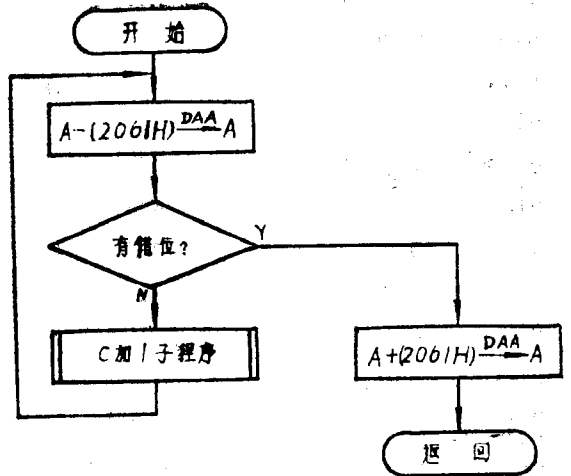


图3-8-2

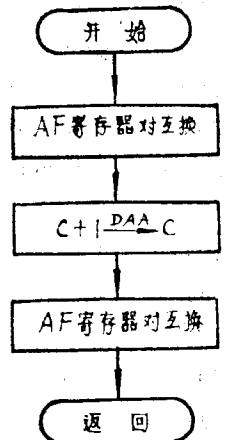


图3-8-3

### 3-9 多字节BCD码除法(一)

- 说明: 1. 字节数存放在20D0单元中。从20D1单元开始存放被除数和除数(先高位后低位)。而商接在除数后面存放, 保留两位(一字节)小数。
2. 被除数、除数的存放格式:

20D0	20D1		
字节数	被除数	除数	.....

计算后商的存放格式:

20D0	20D1				
00	余数	除数	商整数	商小数	

3. 20F0——20F6单元为工作单元。

4. 此程序适于两个相近的数相除。如果很大的数除以较小的数, 运算时间较长。例如:  $100000 \div 2 = 50000$  要进行五万次减法, 显然此算法不可取。对于这种情况, 可以采用下面除法(二), 进行被除数和除数不同字节数的除法。上述例子仅需进行八次减法。

程序:

2000	DD21F020		LD IX,20F0H
2004	21D020		LD HL,20D0H
2007	4E		LD C,(HL)
2008	0600		LD B,0
200A	09		ADD HL,BC
200B	22F120		LD (20F1H),HL
200E	09		ADD HL,BC
200F	22F320		LD (20F3H),HL
2012	09		ADD HL,BC
2013	22F520		LD (20F5H),HL
2016	DD7100		LD (IX),C
2019	41		LD B,C
201A	AF		XOR A
201B	77	S0:	LD (HL),A
201C	2B		DEC HL
201D	10FC		DJNZ S0
201F	CD8320	S1:	CALL XX
2022	3813		JR C,S4
2024	2AF520		LD HL,(20F5H)
2027	B7	S2:	OR A



2028	7E		LD A,(HL)
2029	C601		ADD 1
202B	27		DAA
202C	3005		JR NC,S3
202E	77		LD (HL),A
202F	2B		DEC HL
2030	C32720		JP S2
2033	77	S3:	LD (HL),A
2034	C31F20		JP S1
2037	ED5BF120	S4:	LD DE,(20F1H)
203B	2AF320		LD HL,(20F3H)
203E	DD4600		LD B,(IX)
2041	B7		OR A
2042	1A	S5:	LD A,(DE)
2043	8E		ADC (HL)
2044	27		DAA
2045	12		LD (DE),A
2046	2B		DEC HL
2047	1B		DEC DE
2048	10F8		DJNZ S5
204A	1E02		LD E,2
204C	DD4600	S6:	LD B,(IX)
204F	04		INC B
2050	2AF120		LD HL,(20F1H)
2053	AF		XOR A
2054	ED6F	S7:	RLD
2056	2B		DEC HL
2057	10FB		DJNZ S7
2059	1D		DEC E
205A	20F0		JR NZ,S6
205C	0E00		LD C,0
205E	CD8320	S8:	CALL XX
2061	3808		JR C,S10
2063	79	S9:	LD A,C
2064	C601		ADD 1
2066	27		DAA
2067	4F		LD C,A
2068	C35E20		JP S8

206B 3AD020  
 206E C600  
 2070 2809  
 2072 D601  
 2074 27

S10: LD A,(20D0H)  
 ADD 0  
 JR Z,S11  
 SUB 1  
 DAA

框图:

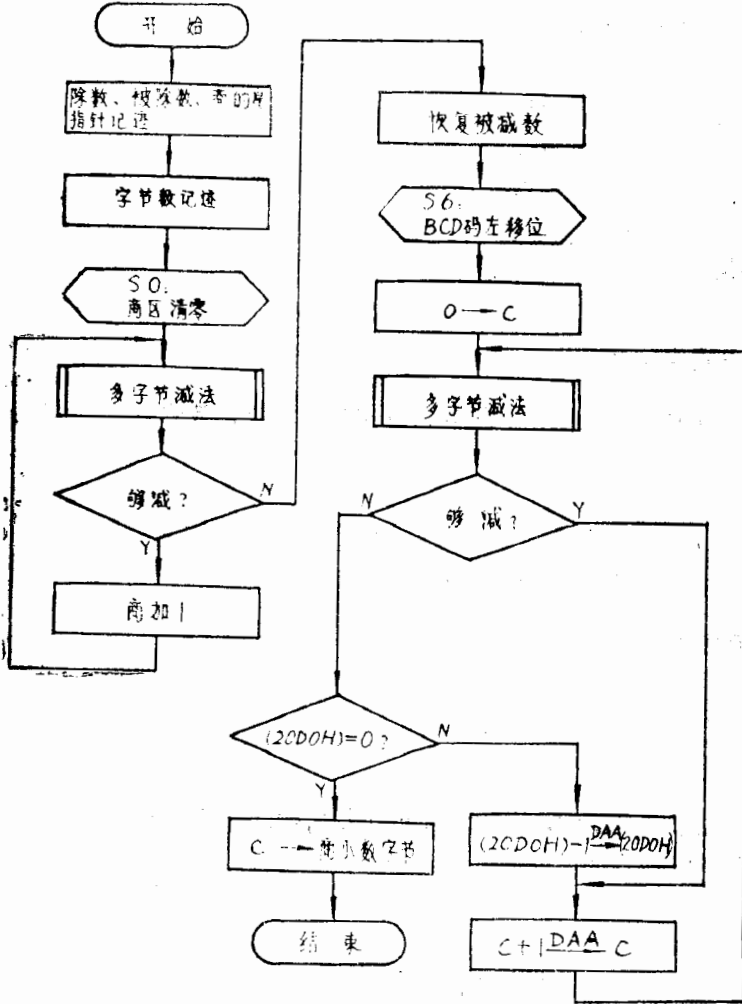


图3-9-1

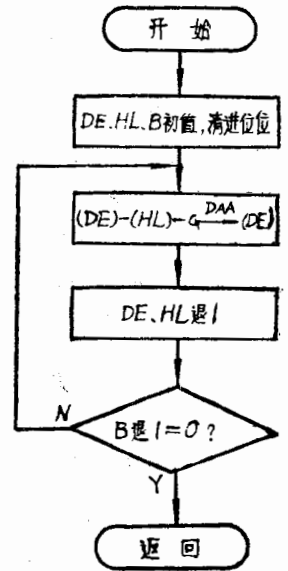


图3-9-2

2075 32D020 LD (20D0H),A  
 2078 C36320 JP S9  
 207B DD2AF520 S11: LD IX,(20F5H)  
 207F DD7101 LD (IX+1),C

2082	76		HALT
2083	ED5BF120	XX:	LD DE,(20F1H)
2087	2AF320		LD HL,(20F3H)
208A	DD4600		LD B,(IX)
208D	B7		OR A
208E	1A	X0:	LD A,(DE)
208F	9E		SBC (HL)
2090	27		DAA
2091	12		LD (DE),A
2092	2B		DEC HL
2093	1B		DEC DE
2094	10F8		DJNZ X0
2096	C9		RET

### 3-10 多字节BCD码除法(二)

说明: 1. 从20D0单元开始依次存放除数字节数、除数、被除数字节数、被除数。而商接在被除数后面存放。

2. 程序执行前数的存放格式:

20D0	20D1				
字节数	除数	字节数	被除数	.....	

计算后结果数的存放格式:

20D0	20D1				
00	除数	00	余数	商	

3. 20F0—20F6单元为工作单元。

4. 该算法未保留小数。如果精度不够, 可以增加被除数倍数, 计算后的商相应减小倍数。这样可以得到所需任意精度的商。

程序:

2000	DD21F020	LD IX,20F0H
2004	21D020	LD HL,20D0H
2007	4E	LD C,(HL)
2008	0600	LD B,0
200A	70	LD (HL),B
200B	09	ADD HL,BC
200C	DD7100	LD (IX),C
200F	22F120	LD (20F1H),HL

2012	23		INC HL
2013	4E		LD C,(HL)
2014	70		LD (HL),B
2015	09		ADD HL,BC
2016	41		LD B,C
2017	AF		XOR A
2018	23	V0:	INC HL
2019	77		LD (HL),A
201A	10FC		DJNZ V0
201C	ED42		SBC HL,BC
201E	ED42		SBC HL,BC
2020	79		LD A,C
2021	DD4E00		LD C,(IX)
2024	09		ADD HL,BC
2025	22F320		LD (20F3H),HL
2028	4F		LD C,A
2029	09		ADD HL,BC
202A	22F520		LD (20F5H),HL
202D	DD9600		SUB (IX)
2030	3C		INC A
2031	DD3400		INC (IX)
2034	4F		LD C,A
2035	ED5BF320	V1:	LD DE,(20F3H)
2039	2AF120		LD HL,(20F1H)
203C	DD4600		LD B,(IX)
203F	B7		OR A
2040	1A	LOOP1:	LD A,(DE)
2041	9E		SBC (HL)
2042	27		DAA
2043	12		LD (DE),A
2044	1B		DEC DE
2045	2B		DEC HL
2046	10F8		DJNZ LOOP1
2048	380B		JR C,V2
204A	2AF520		LD HL,(20F5H)
204D	7E		LD A,(HL)
204E	C601		ADD 1
2050	27		DAA

2051	77		LD (HL),A
2052	C33520		JP V1
2055	ED5BF320	V2:	LD DE,(20F3H)
2059	2AF120		LD HL,(20F1H)
205C	DD4600		LD B,(IX)
205F	B7		OR A
2060	1A	LOOP2:	LD A,(DE)
2061	8E		ADC (HL)
2062	27		DAA
2063	12		LD (DE),A
2064	1B		DEC DE
2065	2B		DEC HL
2066	10F8		DJNZ LOOP2
2068	0D		DEC C
2069	2001		JR NZ,V3
206B	76		HALT
206C	DD3403	V3:	INC (IX+3)
206F	DD3405		INC (IX+5)
2072	C33520		JP V1

框图:

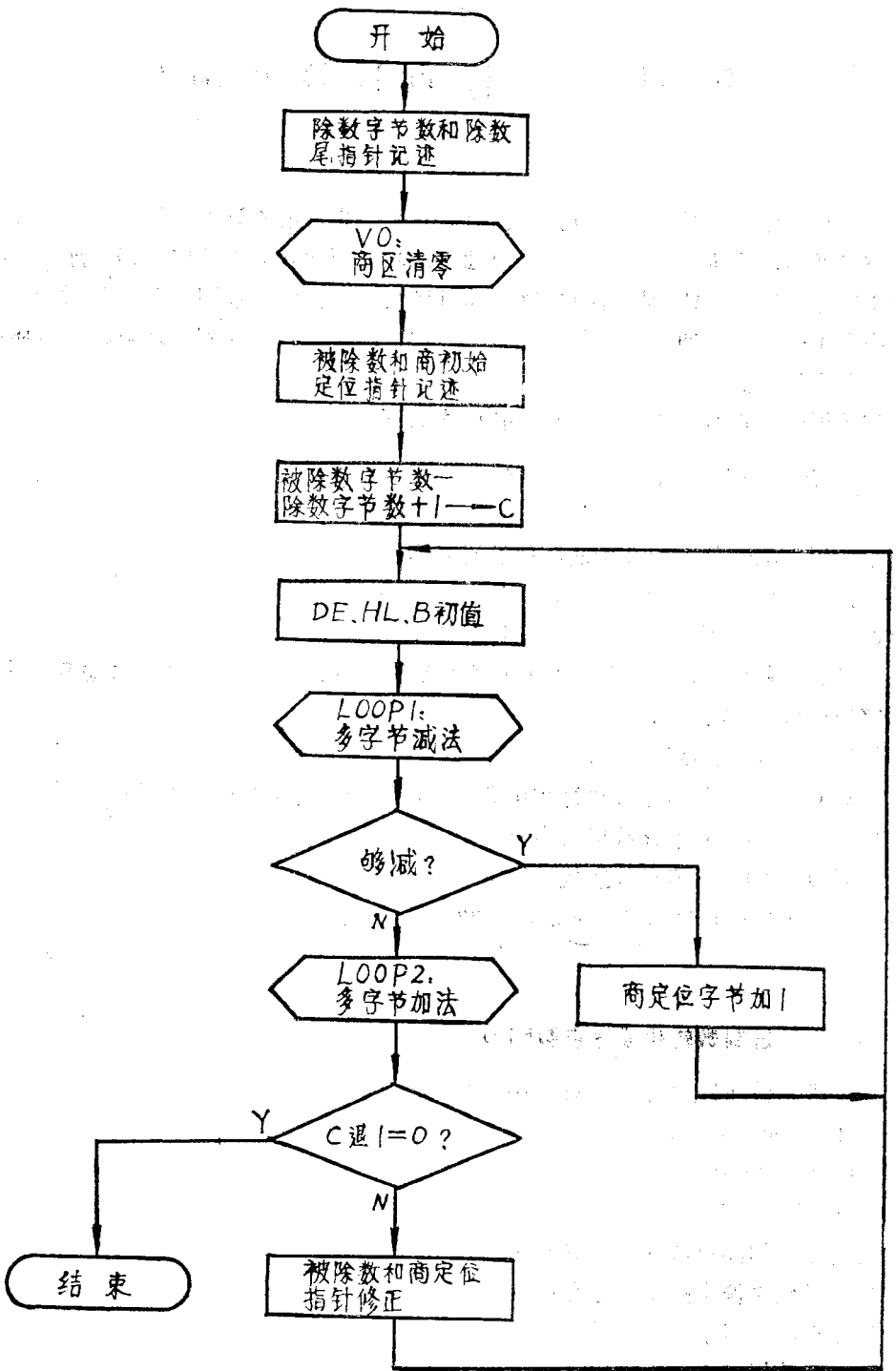
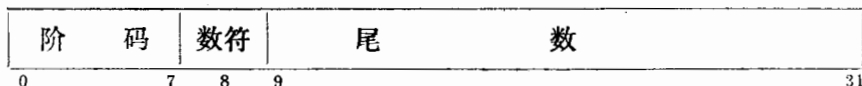


图3-10

## 第四章 浮点运算及函数

本章主要介绍浮点数的加、减、乘、除运算及一些常用的算术函数和三角函数。这些子程序交叉调用，很难把它们一个一个地独立编制，因此我们把这些子程序连成一组，从2000H地址开始统一编址。对于这些子程序的目的、功能、程序框图及清单分别予以说明，同时列出在本子程序中调用的其他子程序名，这样既使读者容易阅读理解，又方便用户正确选用这些子程序。

我们采用的浮点数字长32位，其存贮格式如下：



其中阶码：0到7位，以2为底，用补码表示。

数符：第8位，0表示正，1表示负。

尾数：9到31，用补码表示（连同符号位）。

在本章的子程序中，一般浮点数存贮在FACC开始的四个字节中，或寄存在E、D、C、B寄存器中。一般说来，

(FACC)或E中存放阶码；

(FACC+1)或D中存放数符和尾数高位，我们称为尾数高位字节；

(FACC+2)或C中存放尾数中位字节；

(FACC+3)或B中存放尾数低位字节。

我们这里采用的FACC的对应地址为2288H。

### 4-1 二进制数转化成浮点数FLOT

说明：1.二进制数在B、C、D寄存器中，其中：

B为二进制低位，

C为二进制中位，

D为二进制高位。

2.转化成的浮点数存(FACC)。

3.本子程序调用的子程序有：

浮点数退栈子程序FPOP，

浮点数规格化子程序FNOR。

地址	操作码	操作数	注释
2000	FLOT: LD	E, 23	; 把DCB中二进制作为尾数
2002	PUSH	BC	; 阶码为23存E, 把E、D、C、B中的浮点数推入堆栈
2003	PUSH	DE	
2004	LD	HL, FAC	; 浮点数存贮地址FACC送HL
2007	CALL	FPOP	; 把栈中浮点数送(FACC)
200A	JP	FNOR	; (FACC)中浮点数规格化

框图:

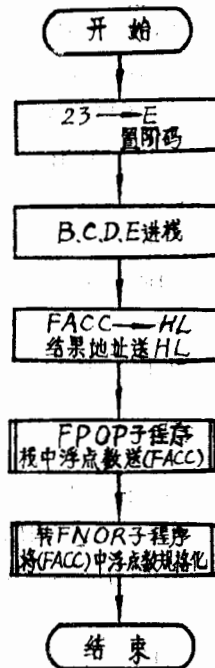


图4-1

#### 4-2 浮点数规格化子程序FNOR

说明: 1. 为了保持计算精度, 浮点数运算之后, 都要根据情况进行规格化。因为浮点数的尾数是用补码表示的, 其规格化的形式是: 当浮点数  $> 0$  时, 其尾数(包括符号位)是  $S = 01XX \dots X$ ; 当浮点数  $< 0$  时, 其尾数是  $S = 10X \dots X$ 。其中  $X$  可以是1或0。

2. 本子程序的功能是使 (FACC) 中的浮点数规格化后仍送 (FACC)。

3. 本子程序调用的子程序有:



浮点数进栈子程序FPSH, 浮点数退栈子程序FPOP, 浮点数取补子程序FNGX. B、C、D三字节左移子程序RLBD.

程序:

200D	218822	FNOR: LD HL,FACC	; 浮点数地址FALL→HL
2010	E5	FNRX: PUSH HL	
2011	CDD421	CALL FPSH	; 浮点数进栈
2014	F5	PUSH AF	
2015	FC EE21	CALL M,FNGX	; 浮点数为负时调用取补子程序取补
2018	F1	POP AF	
2019	D1	POP DE	; 浮点数退栈
201A	C1	POP BC	
201B	F5	PUSH AF	
201C	78	LD A,B	; 检测尾数, 为0时转FNR2
201D	B1	OR C	
201E	B2	OR D	
201F	CA2F20	JP Z,FNR2	
2022	7A	FNR1: LD A,D	; 检测D中的尾数首位为1, 转FNR3
2023	17	RLA	
2024	A7	AND A	
2025	FA3020	JP M,PNR3	
2028	CD5622	CALL RLBD	; 尾数首位为0, D、C、B左移1位
202B	1D	DEC E	; E中阶码减1, 转FNR1
202C	D32220	JP PNR1	
202F	5F	FNR2: LD E,A	; A→E, 即0→E, 阶码置0
2030	F1	PNR3: POP AF	
2031	E1	POP HL	
2032	E5	PUSH HL	
2033	F5	PUSH AF	
2034	C5	PUSH BC	

(下转第62页)

框图:

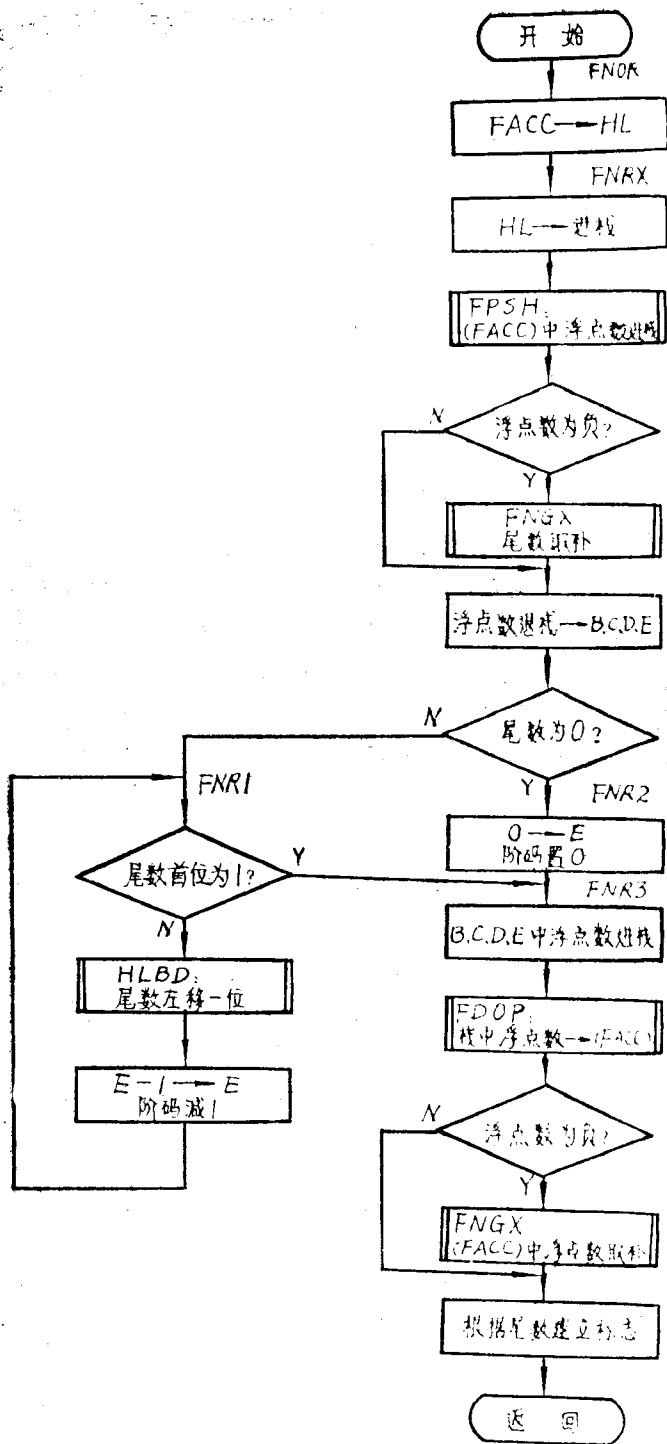


图 4-2

(上接第60页)

2035	D5	PUSH DE	
2036	CDC621	CALL FPOP	; 浮点数送(FACC中)
2039	F1	POP AF	
203A	E1	POP HL	
203B	FC EE21	CALL M,FNGX	; 浮点数为负则取补
203E	23	INC HL	; 根据尾数置标志
203F	7E	LD A,(HL)	
2040	2B	DEC HL	
2041	A7	AND A	
2042	C9	RET	; 返回

#### 4-3 浮点数加法子程序FADD

- 说明: 1. 本子程序的功能是实现(FACC)+(HL)→(FACC)。即被加数在(FACC)中, 加数在HL指示的存贮区(4字节)中, 它们的和仍存贮在(FACC)中。
2. 当加数尾数首字节为0, 不作加法运算, (FACC)中仍旧是被加数。
3. 当被加数与加数的阶差大于或等于24时, 不作加法运算, (FACC)中存大的数。
4. 本子程序调用的子程序有浮点数进栈子程序FPSH, 浮点数出栈子程序FPOP, 浮点数对阶右规子程序FASR, 浮点数规格化子程序FNOR。

程序:

2043	CDD421	FADD: CALL FPSH	; (HL)中的加数进栈
2046	CA8D20	JP Z,FAD5	; 加数为0转FAD5处理
2049	1A	LD A,(DE)	
204A	96	SUB (HL)	; 被加数阶码—加数阶码→A
204B	CA6C20	JP Z,FAD3	; A=0转FAD3处理
204E	F5	PUSH AF	
204F	F25420	JP P,FAD1	; A>0转FAD1处理
2052	2F	CPL	; A<0时取补码存A
2053	3C	INC A	
2054	4F	FAD1: LD C,A	
2055	FE18	CP 24	
2057	F28820	JP P,FAD4	; A≥24转FAD4
205A	F1	POP AF	
205B	F5	PUSH AF	
205C	F26020	JP P,FAD2	; 阶码大的数的存贮地址在DE中, 小的在HL中。

205F	EB	EX DE,HL	
2060	CD1622	FAD2: CALL FASR	; (HL) 中的数右规一位, 阶差减1, 阶差不为0时转 FAD2继续右规对阶。
2063	0D	DEC C	
2064	C26020	JP NZ,FAD2	
2067	F1	POP AF	
2068	F26C20	JP P,FAD3	
206B	EB	EX DE,HL	; 恢复被加数地址在 DE 中, 加数地址在HL中
206C	CD1622	FAD3: CALL FASR	; 两加数都右规一位, 并使被加数地址在HL中, 即HL = FACC
206F	EB	EX DE,HL	
2070	CD1622	CALL FASR	
2073	C640	ADD 40H	; 为四舍五入准备进位位, 然后两加数尾数相加送(FACC)
2075	23	INC HL	
2076	23	INC HL	
2077	23	INC HL	
2078	D1	POP DE	
2079	C1	POP BC	
207A	78	LD A,B	
207B	8E	ADC (HL)	
207C	77	LD (HL),A	
207D	2B	DEC HL	
207E	79	LD A,C	
207F	8E	ADC (HL)	
2080	77	LD (HL),A	
2081	2B	DEC HL	
2082	7A	LD A,D	
2083	8E	ADC (HL)	
2084	77	LD (HL),A	
2085	C30D20	JP FNOR	; (FACC)中浮点数规格化并返回
2088	F1	FAD4: POP AF	
2089	EB	EX DE,HL	; 两加数交换地址即HL = FACC

(下转第65页)

框图:

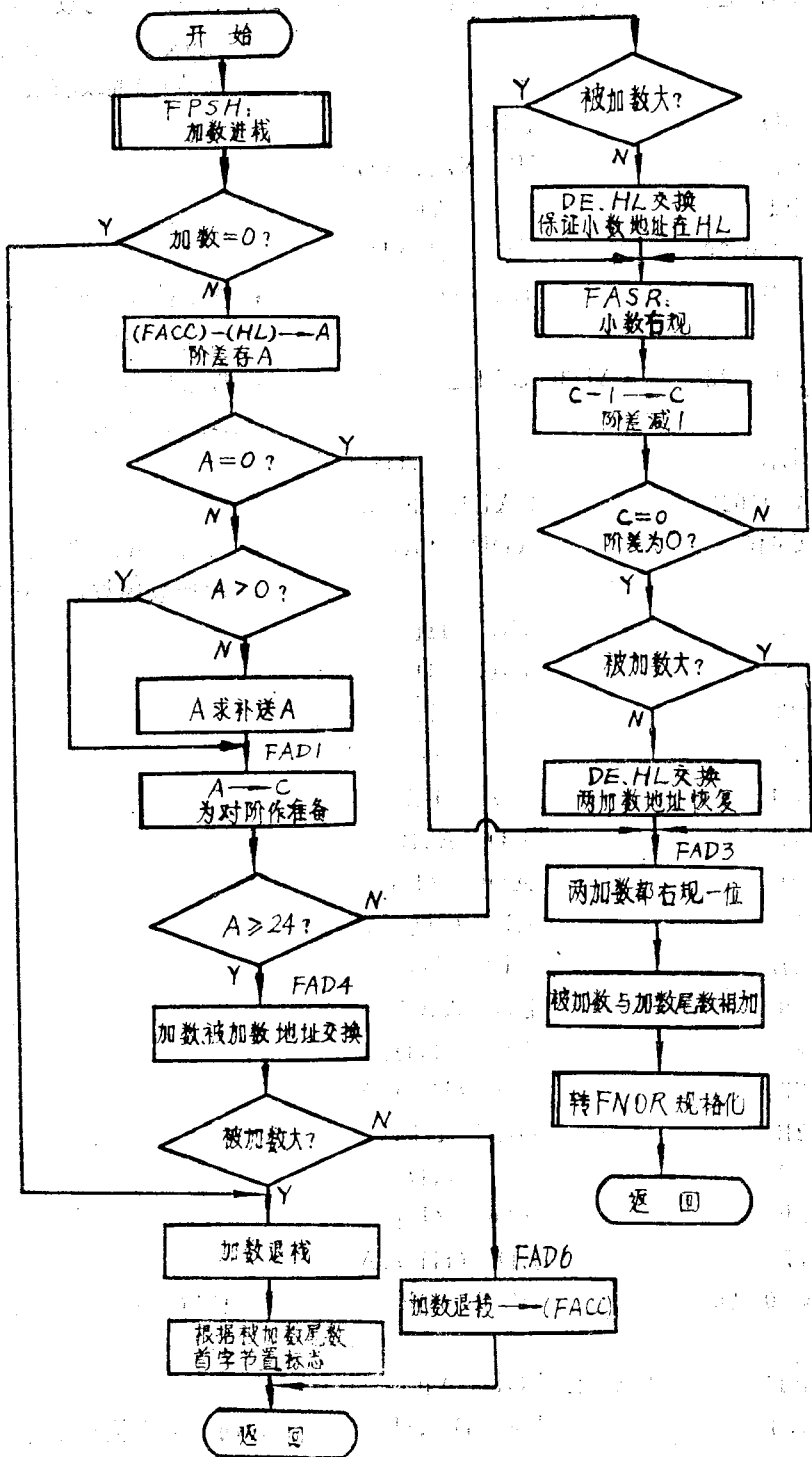


图 4-5

(上接第63页)

```
208A FA9620 JP M,FAD6 ; 加数阶码大转 FAD6, 被加数
           ; 阶码大, 栈中加数退掉
208D D1 FAD5: POP DE
208E C1 POP BC
208F 218922 FCFE: LD HL,FACC+1
2092 7E LD A,(HL)
2093 2B DEC HL ; 根据尾数置标志, 返回
2094 A7 AND A
2095 C9 RET
2096 CDC621 FAD6: CALL FPOP ; 栈中加数退至(FACC)
2099 C9 RET
```

#### 4-4 浮点数减法子程序FSUB

- 说明: 1. 本子程序实现的功能是  $(FACC) - (HL) \rightarrow (FACC)$ 。即被减数在 (FACC) 中, 减数在 HL 指示的存贮区中, 差存 (FACC)。
2. 实现算法是减数取补加被减数而得到差。
3. 调用的子程序有浮点数进栈子程序FPSH, 浮点数取补子程序FNGX, 浮点数加法子程序FADD。

程序:

```
209 A CDD421 FSUB: CALL FPSH ; 减数进栈
209 D CDEE21 CALL FNGX ; 减数取补
20A 0 CD4320 CALL FADD ; 调用加法子程序
20A 3 D1 POP DE ; 减数退栈
20A 4 C1 POP BC
20A 5 C9 RET ; 返回
```

#### 4-5 浮点数乘法子程序FMUL

- 说明: 1. 本子程序实现的功能是  $(FACC) * (HL) \rightarrow (FACC)$ 。即被乘数在 (FACC) 中, 乘数在 HL 指示的存贮区中, 积仍存 (FACC)。
2. 首先处理积的阶码和符号, 在处理尾数相乘时, 取绝对值 (对负数则取补) 进行相乘。
3. 本子程序调用的子程序有:  
浮点数进栈子程序 FPSH, 浮点数取补子程序 FNEG, 浮点数规格化子程序 FNOR, 积商符号子程序 FSIG, 单字节带进位加法子程序 FADS, 六字节左移

一位子程序FRAL,初始化子程序FSPS。

程序:			
20A6	CDD421	FMUL: CALL FPSH	; 乘数进栈
20A9	1A	LD A,(DE)	; 被乘数阶码+乘数阶码+1乘(DE),即存(FACC)
20AA	81	ADD C	
20AB	3C	INC A	
20AC	12	LD (DE),A	
20AD	CDFF21	CALL FSIG	; 形成积的符号,对负的运算量取补
20B0	E1	POP HL	; 乘数出栈
20B1	D1	POP DE	
20B2	7D	LD A,L	
20B3	6B	LD L,E	
20B4	010000	LD BC,00	
20B7	59	LD E,C	
20B8	E5	PUSH HL	; H、L、D中为乘数高、中、低三个尾数字节, E、B、C为三个'0'字节一道进栈,占有六字节工作区
20B9	D5	PUSH DE	
20BA	C5	PUSH BC	
20BB	47	LD B,A	; 乘积符号暂存B
20BC	0E18	LD C,24	; 移位次数24暂存C
20BE	CD4122	FMU1: CALL FSPS	; 初始化,使HL指示部分积低位字节地址(即EBC中的C在栈中地址),DE = FACC + 3,部分积(六字节工作区)左移一位,无进位转FMU3
20C1	CD2C22	CALL FRAL	
20C4	D2DC20	JP NC,FMU3	
20C7	CD4122	CALL FSPS	; 有进位,再初始化,进位清0,被乘数尾数+部分积尾数→部分积

20CA	A7	AND A	
20CB	CD4A22	CALL FADS	
20CE	CD4A22	CALL FADS	
20D1	CD4A22	CALL FADS	
20D4	D2DC20	JP NC,FMU3	; 相加后无进位转FMU3, 有进位, 部分积高位加 进位
20D7	34	FMU2: INC (HL)	
20D8	23	INC HL	
20D9	CAD720	JP Z,FMU2	
20DC	0D	FMU3: DEC C	; C-1→C, C≠0时转 FMU1继续作乘法, C=0时, 作善后处理
20DD	C2BE20	JP NZ,FMU1	
20E0	218922	LD HL,FACC+1	
20E3	D1	POP DE	; 部分积低位退栈, 高位 三字节存(FACC+1)
20E4	D1	POP DE	
20E5	4A	LD C,D	
20E6	D1	POP DE	
20E7	72	LD (HL),D	
20E8	23	INC HL	
20E9	73	LD (HL),E	
20EA	23	INC HL	
20EB	71	LD (HL),C	
20EC	78	LD A,B	; 积的符号为负, 部分积 取补
20ED	A7	AND A	
20EE	C4EB21	CALL NZ,FNEG	
20F1	C30D20	JP FNOR	; 转规格化子程序



框图:

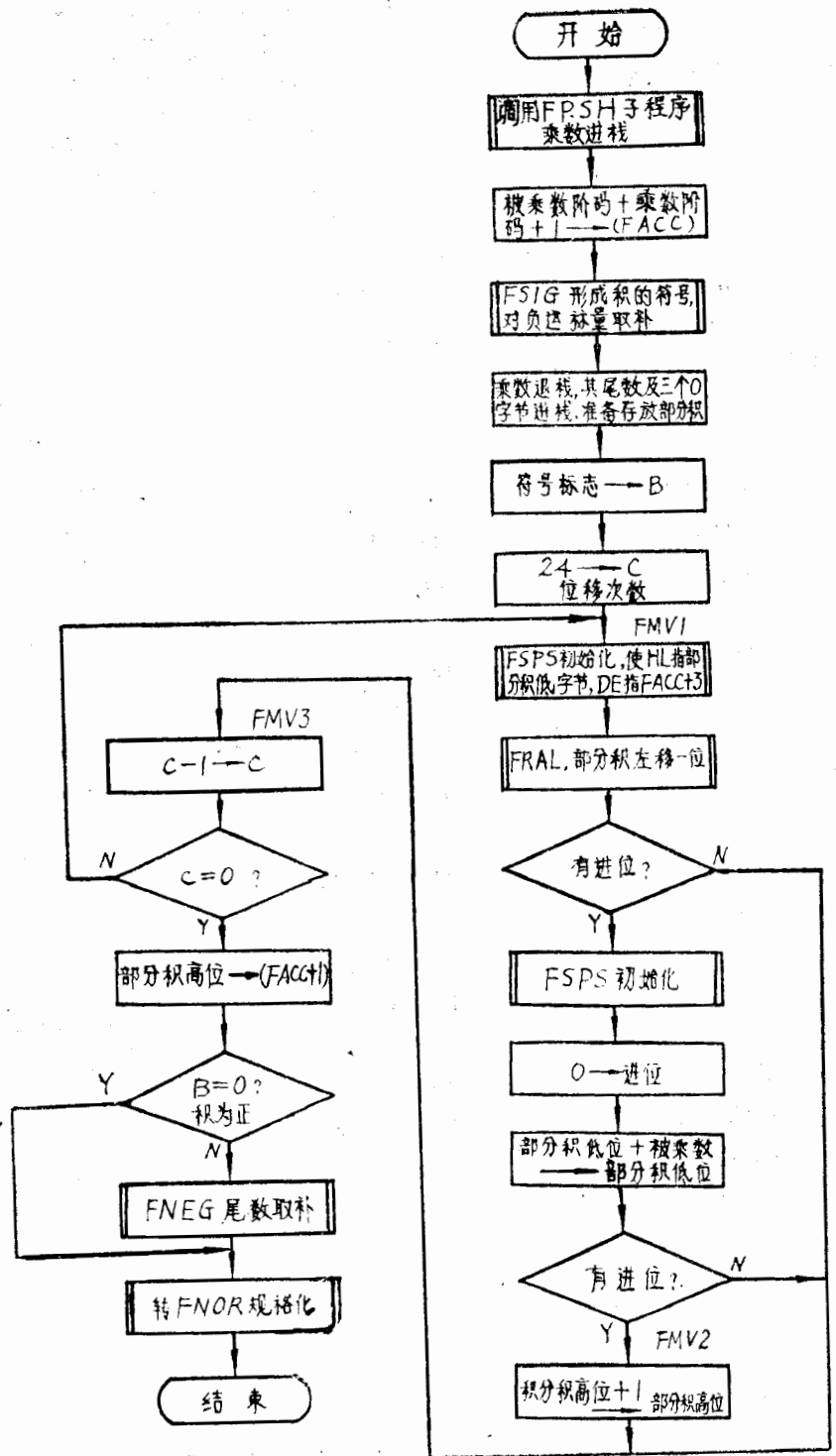


图4-5

#### 4-6 浮点数除法子程序FDIV

- 说明: 1. 本子程序实现  $(FACC)/(HL) \rightarrow (FACC)$  功能。被除数在  $(FACC)$  中, 除数在  $HL$  指示的存贮区中, 结果商仍存  $(FACC)$ 。
2. 被除数为 0, 立即返回且进位为 0; 除数为 0, 立即返回, 但进位为 1, 以示除运算不合法; 而正常的除运算以后进位亦为 0。
3. 浮点数除运算时, 首先进行商的阶码和符号运算, 并且尾数取绝对值 (负数则取补), 然后进行除运算。被除数尾数在  $(FACC+1)$  中, 在运算时应称为余数, 但在框图中我们仍称其为被除数。除数以负数补码形式存在  $HL$  指示的三字节中, 这样被除数减除数, 只要进行被除数加除数补码就行了。商存  $HL+3$  指示的三字节中。而  $HL$  至  $HL+5$  这六个字节都在堆栈中。
4. 本子程序调用了浮点数进栈子程序  $FPSH$ , 积商符号子程序  $FSIG$ , 浮点数取补子程序  $FNEG$ , 单字节减法子程序  $FSBS$ , 三字节左移一位子程序  $FRLO$ , 浮点数规格化子程序  $FNOR$ , 初始化子程序  $FSPS$ 。

程序:

```

20F4  118922  FDIV: LD  DE,FACC+1    ; 检测被除数为 0 吗?
20F7  1A      LD  A,(DE)
20F8  A7      AND  A
20F9  C8      RET  Z                ; 为 0 时返回, 此时进位为 0
20FA  23      INC  HL                ; 检测除数为 0 吗?
20FB  7E      LD  A,(HL)
20FC  A7      AND  A
20FD  37      SCF                    ; 进位置 1
20FE  2B      DEC  HL
20FF  C8      RET  Z                ; 除数为 0 返回
2100  CDD421  CALL FPSH                ; 除数不为 0 进栈
2103  1A      LD  A,(DE)
2104  91      SUB  C                    ; 被除数阶码 - 除数阶码 + 1
                                   存 (FACC)
2105  3C      INC  A
2106  12      LD  (DE),A
2107  CDF21  CALL FSIG                ; 求商的符号, 并使负运算量
                                   取补
210A  CDEE21  CALL FNGX                ; 除数取补, 为减除数作准备
210D  D1      POP DE                ; 除数退栈 → D、C、B
210E  C1      POP BC
210F  7B      LD  A,E                ; 符号标志送 A

```

2110	5A	LD E,D	
2111	61	LD H,C	
2112	68	LD L,B	
2113	010000	LD BC,00	; 三个‘0’字节准备存商
2116	51	LD D,C	
2117	C5	PUSH BC	; 三个‘0’字节及除数尾数进栈
2118	D5	PUSH DE	; 占有六个字节的工作区
2119	E5	PUSH HL	
211A	47	LD B,A	; 商的符号送B暂存
211B	0E17	LD C,23	; 移位次数23送C
211D	CD4122	FDV1;CALL FSPS	; 初始化,使HL指示工作区中 除数尾数的低位字节地址, DE = FACC + 3,
2120	1A	LD A,(DE)	; (DE) + (HL), 即被除数 - 除数
2121	86	ADD (HL)	
2122	1B	DEC DE	
2123	23	INC HL	
2124	1A	LD A,(DE)	
2125	8E	ADC (HL)	
2126	1B	DEC DE	
2127	23	INC HL	
2128	1A	LD A,(DE)	
2129	8E	ADC (HL)	
212A	23	INC HL	
212B	D23A21	JP NC,FDV2	; 无进位,也就是不够减,转 FDV2
212E	CD4122	CALL FSPS	; 有进位,先初始化,然后实 行被除数 - 除数 → 被除数 (余数),并产生进位
2131	CD5022	CALL FSBS	
2134	CD5022	CALL FSBS	
2137	CD5022	CALL FSBS	
213A	CD3622	FDV2;CALL FRL0	; 商左移一位,进位位移入商
213D	218B22	LD HL,FACC+3	; 被除数(即余数)左移一位
2140	7E	LD A,(HL)	
2141	17	RLA	
2142	77	LD (HL),A	

2143	2B	DEC HL	
2144	7E	LD A,(HL)	
2145	17	RLA	
2146	77	LD (HL),A	
2147	2B	DEC HL	
2148	7E	LD A,(HL)	
2149	17	RLA	
214A	77	LD (HL),A	
214B	0D	DEC C	， C-1→C, C≠0 转FDV1
			继续作除法
214C	C21D21	JP NZ,FDV1	
214F	D1	POP DE	， C = 0 时，除数退栈，商送
2150	D1	POP DE	(FACC)
2151	4A	LD C,D	
2152	D1	POP DE	
2153	72	LD (HL),D	
2154	23	INC HL	
2155	73	LD (HL),E	
2156	23	INC HL	
2157	71	LD (HL),C	
2158	78	LD A,B	
2159	A7	AND A	
215A	C4EB21	CALL NZ,FNEG	，如果商符为负，取补
215D	C30D20	JP FNOR	，转规格化子程序对商规格化

框图:

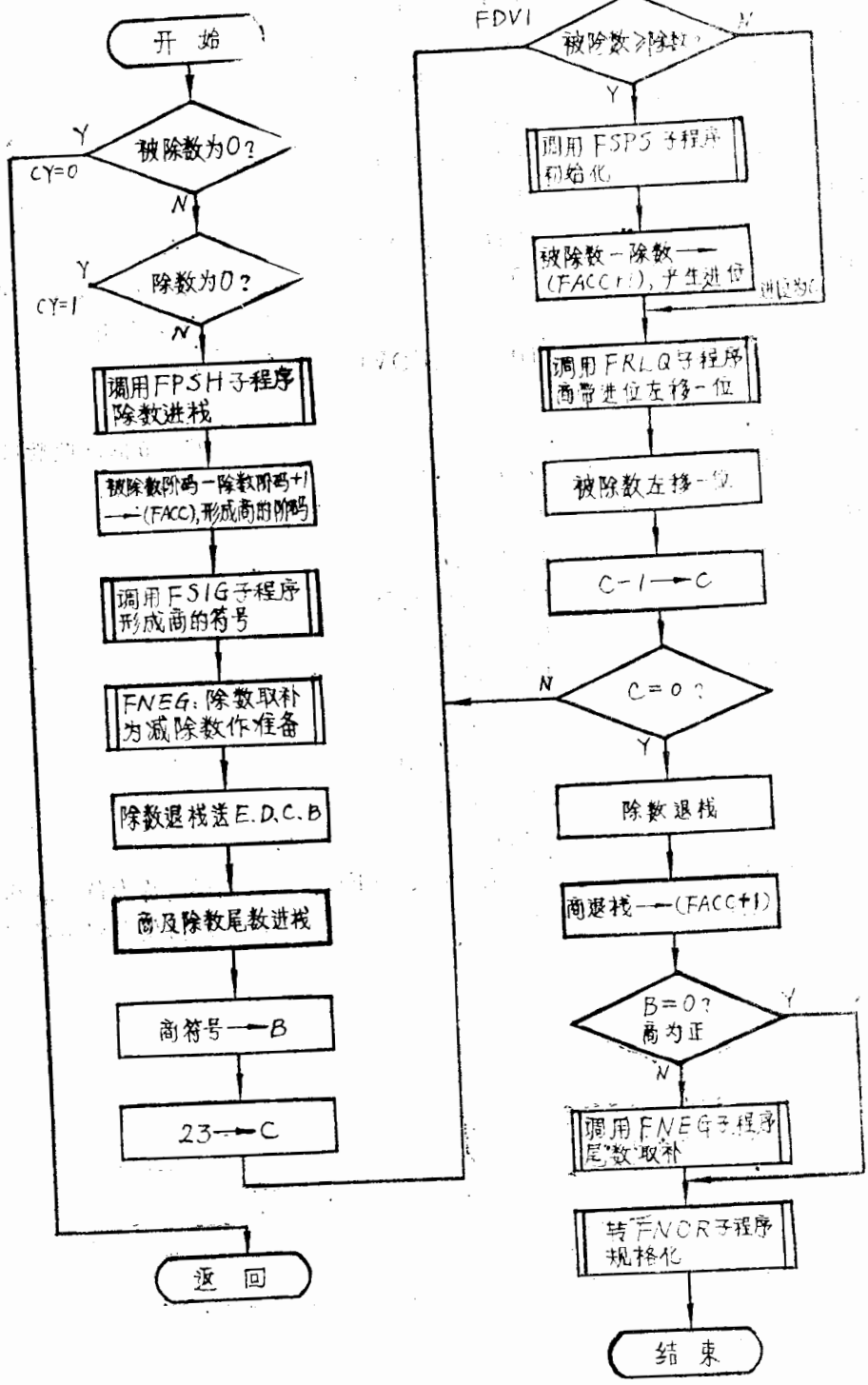


图4-6

#### 4-7 浮点数平方根子程序FSQT

- 说明: 1. 被开方数在(FACC)中, 计算结果仍送(FACC)。  
 2. 被开方数为0, 则不进行开方运算而返回。被开方数为负数, 取绝对值后进行开方运算。  
 3. 本子程序采用牛顿迭代法进行计算。设  $x = \sqrt{a}$ , 即被开方数为 a, 按公式  $x_{k+1} = \frac{1}{2}(x_k + a/x_k)$  进行迭代。  
 本子程序中选取的初值  $x_0$  为: 尾数 = 60D000H, 阶码 = [a的阶码 + 1] / 2, 当  $x_{k+1}$  与  $x_k$  的阶码相等并且尾数的高位字节及中位字节相等时, 就把  $x_{k+1}$  作为结果值。  
 4. 本子程调用的子程序有: 浮点数取补子程序FNEG, 浮点数加法子程序FADD, 浮点数除法子程序FDIV, 浮点数进栈子程序FPSH, 存浮点数子程序FGET。

程序:

2160	218922	FSQT:	LD HL, FACC + 1	; 检测被开方数为0时, 返回
2163	7E		LD A, (HL)	
2164	A7		AND A	
2165	C8		RET Z	
2166	17		RLA	
2167	F5		PUSH AF	
2168	2B		DEC HL	
2169	FCEB21		CALL M, FNEG	; 被平方数 < 0 时, 取补
216C	CDD421		CALL FPSH	; 被开方数进栈
216F	7E		LD A, (HL)	; 阶码/2 + 进位 → E, 阶码符号位不变, 尾数
2170	17		RLA	60D000H送D、C、B, 形成初值
2171	7E		LD A, (HL)	
2172	1F		RRA	
2173	CE00		ADC 00	
2175	5F		LD E, A	
2176	1660		LD D, 60H	
2178	01D000		LD BC, 0D0H	
217B	C5		PUSH BC	; 初值进栈
217C	D5		PUSH DE	

217D	210400	FSQ1:	LD HL, 4	; 被开方数a存(FACC)
2180	39		ADD HL, SP	
2181	CDBE21		CALL FGET	
2184	210000		LD HL, 0	; 形成迭代初值地址 →HL
2187	39		ADD HL, SP	
2188	E5		PUSH HL	; HL进栈
2189	CDF420		CALL FDIV	; (FACC)/ $x_k$ →(FACC)
218C	E1		POP HL	
218D	CD4320		CALL FADD	; (FACC) + $x_k$ → (FACC)
2190	218822		LD HL, FACC	; (FACC)/2→ FACC
2193	35		DEC (HL)	; 成成一次迭代, 计算 出 $x_{k+1}$
2194	D1		POP DE	; 迭代初值退栈
2195	C1		POP BC	
2196	7B		LD A, E	; $x_{k+1}$ 与 $x_k$ 的阶码比较
2197	BE		CP (HL)	
2198	C2AB21		JP NZ, FSQ3	; 不等转FSQ3
219B	23		INC HL	; $x_{k+1}$ 与 $x_k$ 的尾数 高位 字节比较
219C	7A		LD A, D	
219D	BE		CP (HL)	; 不等转FSQ3
219E	C2AB21		JP NZ, FSQ3	
21A1	23		INC HL	; $x_{k+1}$ 与 $x_k$ 的尾数中位 字节比较
21A2	79		LD A, C	
21A3	BE		CP (HL)	
21A4	C2AB21		JP NZ, FSQ3	; 不等转FSQ3
21A7	D1		POP DE	; $x_{k+1}$ 与 $x_k$ 比较相等, 被开方数退栈
21A8	C1		POP BC	
21A9	F1		POP AF	
21AA	69 69		RET	; 返回

21AB 218822 FSQ3: LD HL, FACC

;  $x_{k+1}$ 进栈, 作为新的  
 $x_k$ 转FSQ1继续迭代

21AE CDD421 CALL FPSH

21B1 C37D21 JP FSQ1

框图:

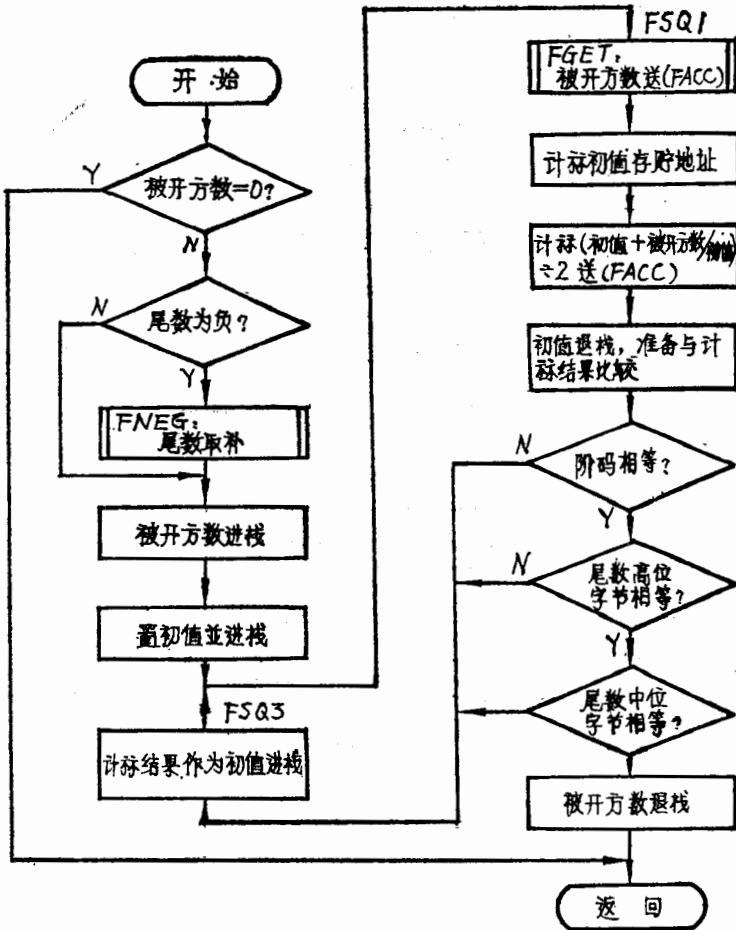


图4-7

#### 4-8 浮点数绝对值子程序FABS

- 说明:
1. 浮点数在 (FACC) 中, 其绝对值仍旧存贮在 (FACC) 中。
  2. 原浮点数为正时, 出口时进位位为 0, 原浮点数为负时, 出口时进位位为 1。
  3. 本子程序调用了浮点数取补子程序 FNEG。



程序:

21B4	3A8 <sup>9</sup> 22	FABS: LD A,(FACC+1); 取浮点数尾数首字节
21B7	A7	AND A
21B8	F0	RET P ; 尾数为正返回
21B9	CDEB21	CALL FNEG ; 为负调用求补子程序
21BC	37	SCF ; 进位置1, 返回
21BD	C9	RET

框图:

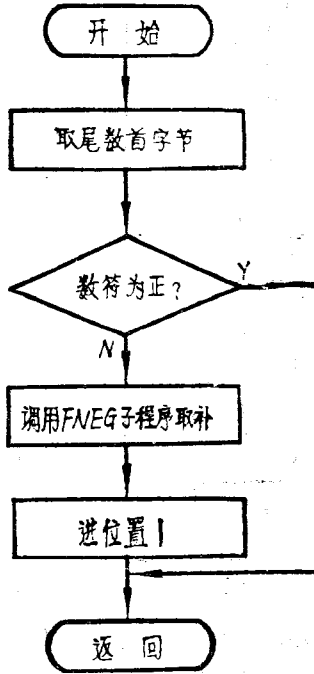


图 4-8

#### 4-9 存浮点数字程序FGET

说明: 1. 本子程序的功能是将由HL指示的存储区中的浮点数存到(FACC)中。

2. 本子程序调用了浮点数进栈子程序FPSH及浮点数出栈子程序FPOP。

程序:

21BE	CDD421	FGET: CALL FPSH ; (HL)中浮点数进栈
21C1	EB	EX DE,HL ; HL中置FACC
21C2	CDC621	CALL FPOP ; 栈中浮点数退出存(FACC)
21C5	C9	RET

## 4—10 浮点数退栈子程序FPOP

- 说明：1. 堆栈中的浮点数退出送HL 指示的存储区中。  
 2. 根据尾数首字节置标志  
 3. 调用本子程序，寄存器内容改变。

程序：

21C6	D1	FPOP: POP DE	； 返回地址退栈送DE
21C7	C1	POP BC	； 退浮点数I、II字节，送(HL)及(HL + 1)
21C8	71	LD (HL),C	
21C9	23	INC HL	
21CA	70	LD (HL),B	
21CB	23	INC HL	
21CC	78	LD A,B	； 根据II字节(即尾数高位字节)置标志
21CD	A7	AND A	
21CE	C1	POP BC	； 退浮点数III、IV字节送(HL + 2)及(HL + 3)
21CF	71	LD (HL),C	
21D0	23	INC HL	
21D1	70	LD (HL),B	
21D2	EB	EX DE,HL	； 返回地址→HL, 返回调用处
21D3	E9	JP (HL)	

框图：

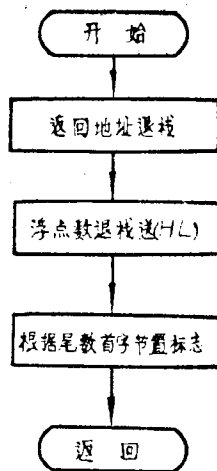


图 4-10

#### 4—11 浮点数进栈子程序FPSH

- 说明：1.把HL指示的存贮区中的浮点数压入堆栈中寄存。  
 2.根据浮点数尾数首字节置A、F，浮点数I、II字节送BC寄存。  
 3.本子程序出口时HL中寄存浮点数在堆栈中的地址，DE中寄存FACC。

程序：

21D4	5E	FPSH: LD E,(HL)	
21D5	23	INC HL	
21D6	56	LD D,(HL)	
21D7	23	INC HL	
21D8	4E	LD C,(HL)	
21D9	23	INC HL	
21DA	46	LD B,(HL)	; (HL)中浮点数→E、D、C、B
21DB	7A	LD A,D	
21DC	A7	AND A	; 根据尾数首字节置标志
21DD	E1	POP HL	; 弹出返回地址
21DE	C5	PUSH BC	
21DF	D5	PUSH DE	; 浮点数压入堆栈
21E0	C1	POP BC	
21E1	C5	PUSH BC	; 浮点数I、II字节送BC
21E2	E5	PUSH HL	; 压入返回地址
21E3	210200	LD HL,2	; 浮点数在堆栈中的地址送HL
21E6	39	ADD HL,SP	
21E7	118222	LD DE,FACC	; FACC送DE
21EA	C9	RET	

框图:

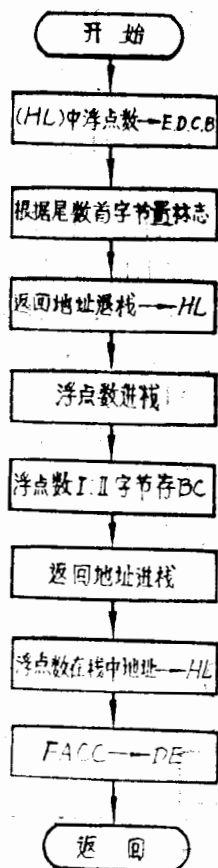


图4-11

#### 4-12 浮点数求补子程序FNEG及FNGX

- 说明: 1. 本子程序有两个入口, FNEG入口使FACC中的浮点数取补, 而FNGX入口使HL指示的存贮区中的浮点数取补。
2. 这里的浮点数求补, 是指浮点数的II、III、IV三个字节看成无符号的数求反加1。
3. FNG1是本子程序的内部子程序, 其功能是实现单字节求反加进位, 因此本子程序中二次调用FNG1, 第三次直接进入并实现返回。

程序:				
21EB	218222	FNEG:	LD HL, FACC	; 浮点数地址送HL
21EE	23	FNGX:	INC HL	; HL + 3 → HL, 使HL 指向浮点数末字节
21EF	23		INC HL	
21F0	23		INC HL	
21F1	37		SCF	; 进位置1
81F2	CDF821		CALL FNG1	; 低位字节求反加1
21F5	CDF821		CALL FNG1	; 中位字节求反加进位
21F8	7E	FNG1:	LD A, (HL)	; 高位字节求反加进位
21F9	2F		CPL	
21FA	CE00		ADC 0	
21FC	77		LD (HL), A	
21FD	2B		DEC HL	
21FE	C9		RET	; 返回

框图:

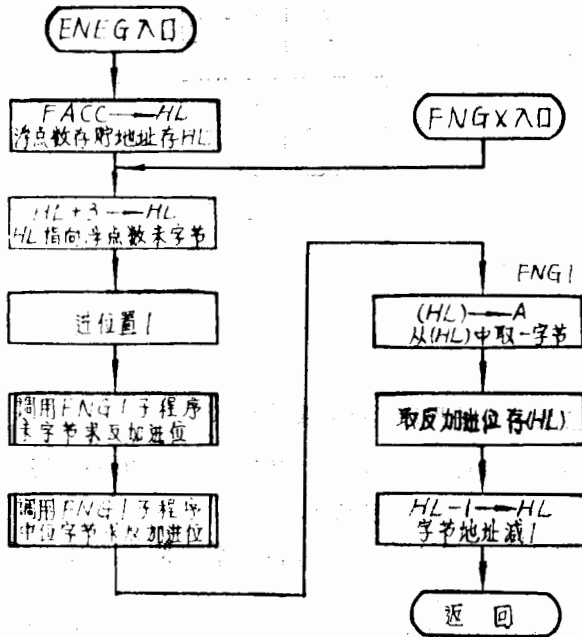


图4-12

#### 4-13 浮点数求积(商)符号子程序FSIG

说明: 1. 浮点数的符号位在尾数首字节的第0位。其值为1, 表示负数; 其值为0, 表示正数。

2. 在浮点数乘(除)运算确定符号时调用本子程序。本子程序的功能是根据两运算量的符号确定积(商)的符号。同时对负的运算量取补(即取其绝对值), 为进行尾数乘除作准备。

3. 本子程序入口时HL指示乘(除)数的地址, DE指示被乘(除)数的地址, B中已是乘(除)数的尾数首字节。

4. 出口时乘(除)数的第一字节(即阶码字节)置入符号标志。其意义

为: (HL) =  $\left\{ \begin{array}{ll} 0 & \text{两运算量同号, 结果为正} \\ 1 & \text{只有乘(除)数为负} \\ -1 & \text{只有被乘(除)数为负} \end{array} \right\}$  结果为负

5. 本子程序调用了浮点数取补子程序FNGX。

程序:

21FF	3600	FSIG:	LD (HL), 0	; 乘(除)数阶码置0
2201	78		LD A, B	; 检测乘(除)数符号, 为 正时转FSG1, 为负时乘 (除)数取补, 标志置1
2202	A7		AND A	
2203	F20A22		JP P, FSG1	
2206	CDEE21		CALL FNGX	
2209	34		INC (HL)	
220A	13	FSG1:	INC DE	; 检测被乘(除)数符号
220B	1A		LD A, (DE)	
220C	1B		DEC DE	
220D	A7		AND A	
220E	F0		RET P	; 为正, 时返回
220F	EB		EX DE, HL	; 为负时, 取补
2210	CDEE21		CALL FNGX	
2213	EB		EX DE, HL	
2214	35		DEC (HL)	; 标志-1
2215	C9		RET	; 返回

框图:

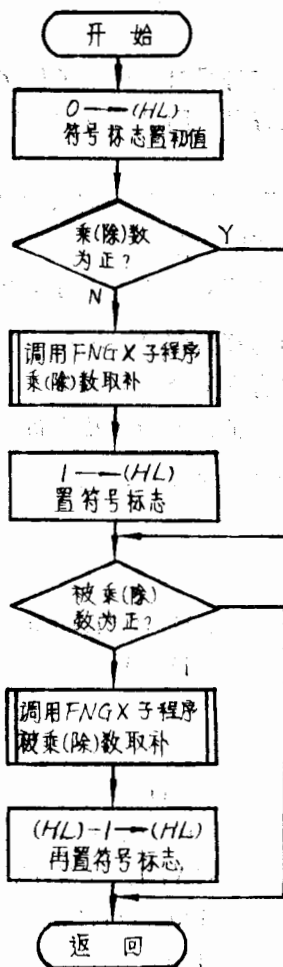


图 4-13

#### 4-14 浮点数对阶右移子程序 FASR

- 说明:
1. 在进行浮点数加减法运算时, 一般需要进行对阶并对阶码小的数尾数右移, 阶码增大。本子程序使尾数右移一位, 阶码增 1。
  2. 尾数移出部分存 A、B 寄存器的第七位, 为进行尾数舍入作准备,
  3. 入口时 HL 指示浮点数的存贮区, 出口时 HL 值恢复。

程序:

2216	34	FASR:	INC (HL)	; 阶码 + 1
2217	23		INC HL	; 尾数高位字节左移一位, 移出
2218	7E		LD A, (HL)	; 符号位
2219	17		RLA	
221A	7E		LD A, (HL)	; 尾数三字节带进位右移一位,
221B	1F		RRA	; 符号位不变
221C	77		LD (HL), A	
221D	23		INC HL	
221E	7E		LD A, (HL)	
221F	1F		RRA	
2220	77		LD (HL), A	
2221	23		INC HL	
2222	7E		LD A, (HL)	
2223	1F		RRA	
2224	77		LD (HL), A	
2225	78		LD A, B	; 移出位移入B、A寄存器高位
2226	1F		RRA	; 为舍入作准备
2227	47		LD B, A	
2228	2B		DEC HL	; HL中的值恢复
2229	2B		DEC HL	
222A	2B		DEC HL	
222B	C9		RET	; 返回

框图:

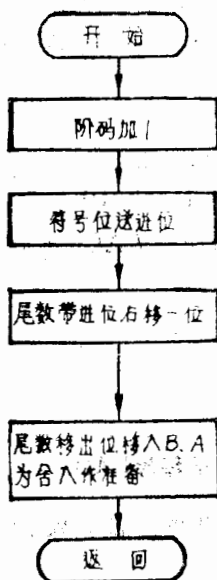


图4-14



#### 4—15 多字节左移一位子程序FRAL及FRL0

- 说明：1.本子程序有两个入口，FRAL子程序是使HL指示的六个字节左移一位，FRL0子程序是使HL指示的三个字节左移一位。
- 2.入口时HL指示的是低位字节地址，出口时HL值变化了。

程序：

222C	A7	FRAL:	AND A	；进位位置0
222D	CD3C22		CALL FRL1	；HL指示的三个字节左移
2230	CD3C22		CALL FRL1	；一位
2233	CD3C22		CALL FRL1	；HL + 3 → HL
2236	CD3C22	FRL0:	CALL FRL1	；HL指示的三个字节左移
2239	CD3C22		CALL FRL1	；一位
223C	7E	FRL1:	LD A,(HL)	；HL + 3 → HL
223D	17		RLA	
223E	77		LD (HL),A	
223F	23		INC HL	
2240	C9		RET	；返回

#### 4—16 初始化子程序

- 说明：在进行乘除运算时需要调用初始化子程序来恢复工作区指针。被乘(除)数在FACC~FACC+3存储区中，所以出口时DE=FACC+3，指向尾数的低位字节地址。子程序中，SP指示的是本子程序的返回地址，SP+2才是乘除工作区的始址，因此出口时HL中指示的是乘除工作区的首地址，工作区末址为HL+5。

程序：

2241	210200	FSPS:	LD HL, 2	；形成工作区首址送HL
2244	39		ADD HL, SP	
2245	118522		LD DE, FACC+3	；被乘(除)数尾数低位字节地址 存DE
2248	A7		AND A	
2249	C9		RET	；返回

#### 4—17 单字节带进位加法子程序FADS

- 说明：1.本子程序是供多字节加法调用的。被加数字节由DE指示，且DE递减；加数字节由HL指示，且HL递增。
- 2.本子程序实现(DE)+(HL)+CY→(HL)。

程序:

224A	1A	FADS:	LD A, (DE)	; 取(DE)
224B	1B		DEC DE	; DE减1
224C	8E		ADC (HL)	; (DE)+(HL)+CY→(HL)
224D	77		LD (HL), A	
224E	23		INC HL	; HL+1
224F	C9		RET	; 返回

#### 4—18 单字节带进位减法子程序FSBS

- 说明: 1. 本子程序实现的功能是 $(DE) + (HL) + CY \rightarrow (DE)$ , 且DE减1, HL增1。  
2. 本子程序的功能与FADS子程序功能相仿, 但是本子程序是供多字节减法调用的。减数取补后才调用本子程序进行单字节加, 而子程序名仍以减法命名。

程序:

2250	1A	FSBS:	LD A, (DE)	; (DE)+(HL)+CY→(DE)
2251	8E		ADC (HL)	
2252	12		LD (DE), A	
2253	1B		DEC DE	; DE减1
2254	23		INC HL	; HL加1
2255	C9		RET	; 返回

#### 4—19 B、C、D寄存器左移子程序RLBD

- 说明: 1. 在进行浮点数运算时, 其尾数一般寄存在B、C、D寄存器中。并且D寄存器中为尾数高位字节, C为中位字节, B为低位字节。  
2. 本子程序实现寄存在D、C、B中的尾数左移一位。

程序:

2256	A7	RLBD:	AND A	; 进位清0
2257	78		LD A, B	; 低位字节带进位左移一位
2258	17		RLA	
2259	47		LD B, A	
225A	79		LD A, C	; 中位字节带进位左移一位
225B	17		RLA	
225C	4F		LD C, A	
225D	7A		LD A, D	; 高位字节带进位左移一位

225E	17	RLA	
225F	57	LD D, A	
2260	C7	RET	; 返回

#### 4—20 浮点数平方子程序

- 说明：1. 本子程序实现 $(EACC)^2 \rightarrow (FACC)$ 的功能。即(FACC)中的浮点数平方运算后仍送(FACC)。
2. 本子程序调用了浮点数乘法子程序FMUL。

程序：

2261	218822	FSQU: LD HL, FACC	; FACC→HL
2264	C3A620	JP FMUL	; (FACC)×(HL)→(FACC)

#### 4—21 浮点数立方子程序FCUBE

- 说明：1. 本子程序实现的功能是 $(FACC)^3 \rightarrow (FACC)$ ，即(FACC)中的浮点数立方运算后仍送(FACC)。
2. 本子程序调用了浮点数进栈子程序FPSH及浮点数乘法子程序FMUL。

程序：

2267	218822	FCUBE: LD HL, FACC	; 底数进栈暂存
226A	CDD421	CALL FPSH	
226D	E5	PUSH HL	; 底数在栈中地址进栈
226E	CDA620	CALL FMUL	; (FACC)×(HL) →(FACC), 实现平方
2271	E1	POP HL	; 底数地址退栈送HL
2272	CDA620	CALL FMUL	; (FACC)×(HL) →(FACC), 实现立方
2275	D1	POP DE	; 栈中底数退栈
2276	D1	POP DE	
2277	C9	RET	; 返回

框图:

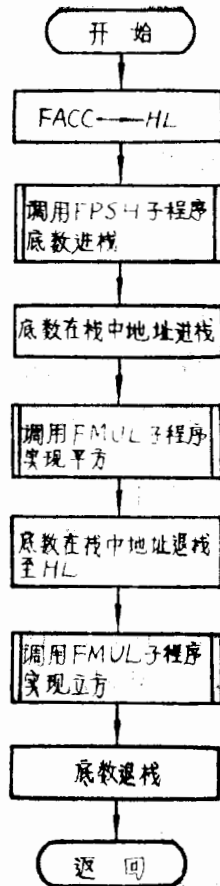


图 4-21

#### 4-22 取浮点数字程序FPUT

- 说明: 1. 本程序的功能是将(FACC)中的浮点数取到HL指示的存贮区中。  
 2. 本程序调用的子程序有浮点数进栈子程序FPSH和浮点数出栈子程序FPOP

程序:

```

2278 E5      FPUT; PUSH HL          ; HL进栈, 暂存结果地址
2279 218822 LD HL, FACC          ; (FACC)中浮点数进栈
227C CDD421  CALL FPSH
227F D1      POP DE              ; 浮点数退栈送E、D、C、B
2280 C1      POP BC
  
```

2281	E1	POP HL	；结果地址退栈送HL
2282	C5	PUSH BC	；浮点数再进栈
2283	D5	PUSH DE	
2284	CDC621	CALL FPOP	；调用FPOP子程序，将栈中浮点数退至HL指示的存贮区，返回
2287	C9	RET	
2288	0000	FACC: DEFW 00, 00	
	0000		

#### 4—23 角度化弧度子程序CDTR

- 说明：1. 本子程序的功能是将(FACC)中的角度化成弧度。  
 2. 本子程序使用了1°的弧度数0.0174532这一常数，其地址在254DH，标号为FP18。  
 3. 本子程序调用了乘法子程序FMUL。

程序：

228C	214D25	CDTR: LD HL, FP18	；取1度的弧度数地址至HL。
228F	C3A620	JP FMUL	；(FACC) × (HL) → (FACC) 得到弧度数

#### 4—24 弧度化角度子程序CRTD

- 说明：1. 本子程序的功能是将(FACC)中的弧度化成角度。  
 2. 本子程序使用了1°的弧度数0.0174532这一常数，其存贮地址为254DH，标号为FP18。  
 3. 本子程序中调用了除法子程序FDIV。

程序：

2292	214D25	CRTD: LD HL, FP18	；取1度的弧度数地址至HL
2295	C3F420	JP FDIV	；(FACC)/(HL) → (FACC)， 得到角度数。

#### 4—25 角度余弦函数FCOD

- 说明：1. 本子程序的功能是实现 $\cos(\text{FACC}) \rightarrow (\text{FACC})$ 。即(FACC)中是角度，将其余弦值送(FACC)。  
 2. 本子程序的执行过程是首先将(FACC)中的角度再加90°，然后再转角度正弦

函数FSID处理。90°值存2565H, 标号为FP90。

3. 本子程序的清单和框图请看4—28。

#### 4—26 角度正弦函数FSID

- 说明: 1. 本子程序的功能是实现 $SIN(FACC) \rightarrow (FACC)$ 。即(FACC)中是角度, 将其正弦值送(FACC)。
2. 本子程序的执行过程是首先将角度转化成弧度, 然后再转FSIN处理。
3. 本子程序的清单和框图请看4—28。

#### 4—27 弧度余弦函数FCOS

- 说明: 1. 本子程序是计算(FACC)的余弦值, 并将计算结果送(FACC)。
2. 本子程序首先将(FACC)中的弧度加 $\pi/2$ 后送(FACC), 然后转FSIN子程序处理。
3. 本子程序的清单和框图请看4—28。

#### 4—28 弧度正弦函数FSIN

- 说明: 1. 本子程序实现计算(FACC)中的弧度数的正弦函数值并送(FACC)。
2. 本子程序按切比雪夫近似多项式计算, 即

$$\sin \frac{\pi}{2} x = 1.5707963x - 0.64596336x^3 + 0.079688475x^5 - 0.0046722203x^7 + 0.00015081716x^9$$

3. 本子程序调用的子程序有: 浮点数出栈子程序FPOP, 浮点数加法子程序FADD, 浮点数取补子程序FNEG, 浮点数减法子程序FSUB, 浮点数进栈子程序FPSH, 浮点数乘法子程序FMUL, 存浮点数字子程序FGET, 浮点数除法子程序FDIV, 取浮点数字子程序FPUT, 浮点数比较子程序FCMP, 多项式计算子程序FCAL。

程序: 下列程序清单中有四个入口, 即FCOD、FSID、FCOS、FSIN。

```

2298 216525 FCOD: LD HL, FP90 ; 取90°的存贮地址送HL
229B CD4320 CALL FADD ; (FACC)+90°→(FACC)
229E 214D25 FSID: LD HL, FP18 ; 取1°的弧度数存贮地址
22A1 CDA620 CALL FMUL ; 角度化成弧度
22A4 C3AD22 JP FSIN ; 转FSIN求正弦值
22A7 215125 FCOS: LD HL, FPI2 ; 取π/2存贮地址
22AA CD4320 CALL FADD ; (FACC)+π/2→(FACC)

```

22AD	AF	FSIN:	XOR A	; A清0作为标志初值进栈
22AE	F5		PUSH AF	
22AF	3A8922		LD A,(FACC+1)	; 检测弧度的符号
22B2	A7		AND A	
22B3	F2BC22	FSI1:	JP P, FSI2	; 弧度为正转FSI2
22B6	CDEB21		CALL FNEG	; 弧度为负取补, 并设标志-1
22B9	F1		POP AF	
22BA	3D		DEC A	
22BB	F5		PUSH AF	
22BC	215925	FSI2:	LD HL, F2PI	; (FACC)<2 $\pi$ 转FSI3
22BF	CDA525		CALL FCMP	
22C2	FACF22		JP M, FSI3	
22C5	218822		LD HL, FACC	; (FACC) $\geq$ 2 $\pi$ 时
22C8	CDD421		CALL FPSH	
22CB	D5		PUSH DE	; (FACC)进栈且FACC进栈
22CC	C31F23		JP FSCM	转FSCM处理
22CF	215525	FSI3:	LD HL, FPAI	; 取 $\pi$ 地址 $\rightarrow$ HL
22D2	E5		PUSH HL	; HL进栈
22D3	CDA525		CALL FCMP	
22D6	FAE122		JP M, FSI4	; (FACC)< $\pi$ 转FSI4
22D9	E1		POP HL	; (FACC) $\geq$ $\pi$ 时
22DA	F1		POP AF	
22DB	3C		INC A	; 标志加1
22DC	F5		PUSH AF	
22DD	E5		PUSH HL	
22DE	CD9A20		CALL FSUB	; (FACC)- $\pi$ $\rightarrow$ (FACC)
22E1	215125	FSI4:	LD HL, FPI2	; 取 $\pi/2$ 地址 $\rightarrow$ HL
22E4	CDA525		CALL FCMP	; (FACC)与 $\pi/2$ 比较
22E7	E1		POP HL	; $\pi$ 地址退栈 $\rightarrow$ HL
22E8	FAF322		JP M, FSI5	; (FACC) $\leq$ $\pi/2$ 转FSI5
22EB	E5		PUSH HL	; $\pi$ 地址进栈
22EC	CDEB21		CALL FNEG	; $\pi$ -(FACC) $\rightarrow$ (FACC)
22EF	E1		POP HL	
22F0	CD4320		CALL FADD	
22F3	215125	FSI5:	LD HL, FPI2	; (FACC)/( $\pi/2$ ) $\rightarrow$ (FACC), 即x
22F6	CDF420		CALL FDIV	
22F9	218822		LD HL, FACC	; (FACC)进栈(x进栈)

22FC	CDD421	CALL FPSH	
22FF	E5	PUSH HL	; x在栈中的指示地址进栈
2300	EB	EX DE, HL	; (FACC)→HL
2301	CDA620	CALL FMUL	; (FACC) <sup>2</sup> →(FACC) 即 $x^2 \rightarrow (FACC)$
2304	114223	LD DE, FSIT	; 参数地址FSIT→DE
2307	3E04	LD A, 04H	; 多项式项数-1→A
2309	CD2325	CALL FCAL	; 调用多项式计算子程序
230C	E1	POP HL	; 取x, 与多项式结果相乘
230D	CDA620	CALL FMUL	
2310	D1	POP DE	; 退x
2311	D1	POP DE	
2312	F1	POP AF	; 退标志
2313	C4EB21	CALL NZ, FNEG	; 标志非0, 结果取补
2316	23	INC HL	; 根据x设标志
2317	7E	LD A, (HL)	
2318	2B	DEC HL	
2319	A7	AND A	
231A	C9	RET	; 返回
231B	(0004)	SARE, DEFS 4	
231F	215925	FSCM: LD HL, F2PI	; $2\pi \rightarrow (FACC)$
2322	CDBE21	CALL FGET	
2325	211B23	LD HL, SARE	; (FACC)→SARE, 暂存
2328	CD7822	CALL FPUT	2 $\pi$ 值
232B	E1	POP HL	; FACC出栈HL
232C	CDC621	CALL FPOP	; 原(FACC)中的值恢复
232F	211B23	LD HL, SARE	
2332	E5	FSM1: PUSH HL	; SARE进栈
2333	34	INC (HL)	; (SARE)中的值×2, 与
2334	CDA525	CALL FCMP	(FACC)比较
2337	E1	POP HL	; SARE出栈→HL
2338	F23223	JP P, FSM1	; (FACC)≥(SARE) 转FSM1
233B	35	DEC (HL)	; (SARE)/2→(SARE)
233C	CD9A20	CALL FSUB	; (FACC)-(SARE)
233F	C3BC22	JP FS12	→(FACC)

(下转第93页)



框图:

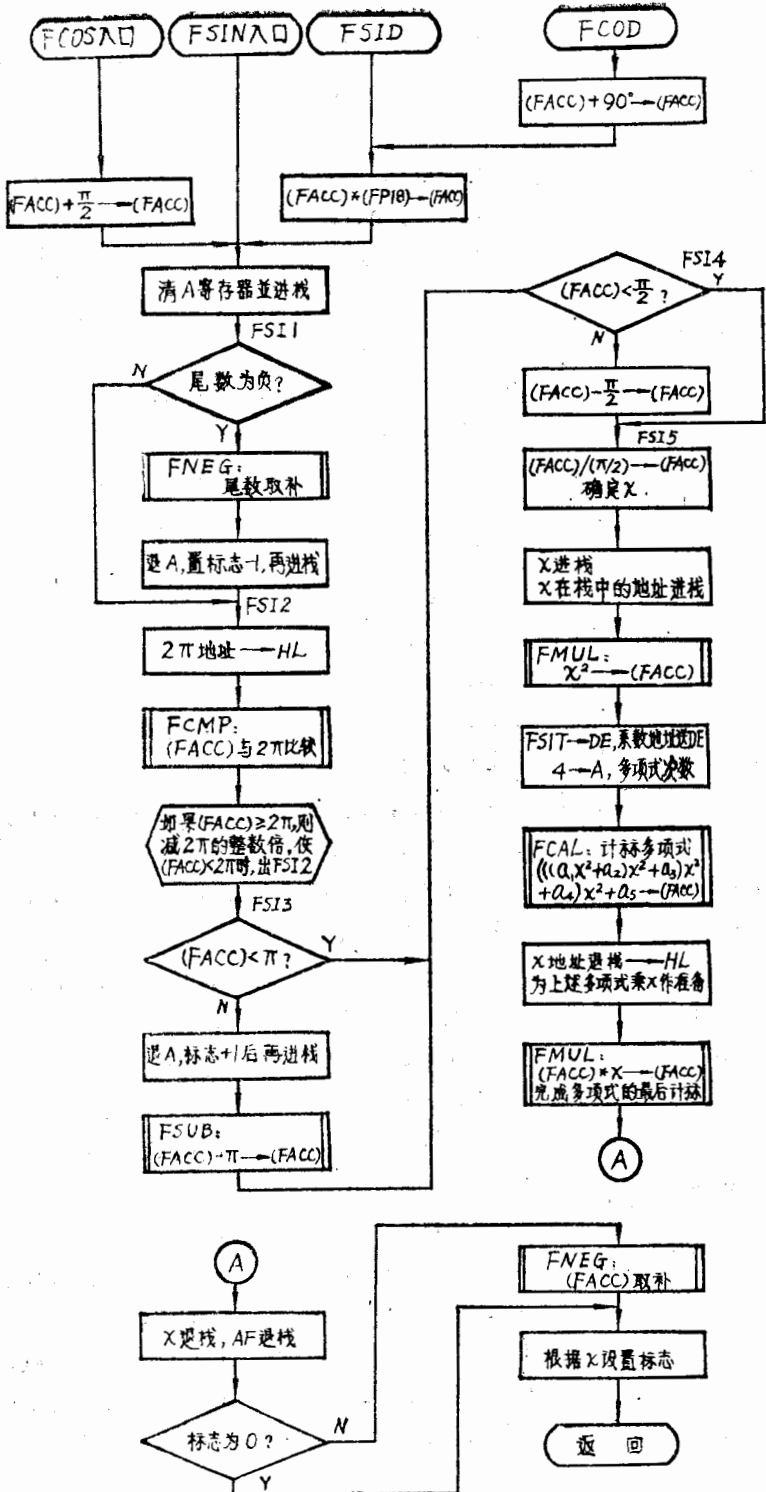


图 4-28

(上接第91页)

					多项式系数:
2342	F44F6BDD	FSIT; DEFB	0F4H, 04FH, 06BH, 0DDH		
					; 0.00015081716
2346	F9B36CCF	DEFB	0F9H, 0B3H, 06CH, 0CFH		
					; -0.0046722203
234A	FD519A2C	DEFB	0FDH, 051H, 09AH, 02CH		
					; 0.079688475
324E	00AD5110	DEFB	000H, 0ADH, 051H, 010H		
					; -0.64596336
2325	016487F0	DEFB	001H, 064H, 087H, 0F0H		
					; 1.5707963

#### 4-29 正切函数FTAN

- 说明:
1. (FACC)中的初值为弧度数, 结果为其正切值。
  2. 当(FACC)中的弧度数 =  $n \cdot \pi + \pi/2$  ( $n$ 为整数)时, 无意义, 出口进位位为1; 正常出口时进位位为0。
  3. 本子程序调用了取浮点数字子程序FPUT, 存浮点数字子程序FGET, 弧度正弦函数FSIN, 弧度余弦函数FCOS, 浮点除法子程序FDIV。

程序:

2356	(0004)	TARE, DEFS	04H		
235A	215623	FTAN; LD	HL, TARE		; (FACC)→TARE
235D	CD7822	CALL	FPUT		
2360	CDA722	CALL	FCOS		; 计算COS(x), 且进栈
2363	218822	LD	HL, FACC		
2366	CDD421	CALL	FPSH		
2369	E5	PUSH	HL		; 在栈中的地址进栈
236A	215623	LD	HL, TARE		; (TARE)→(FACC)
236D	CDBE21	CALL	FGET		
2370	CDAD22	CALL	FSIN		; 计算SIN(x)
2373	E1	POP	HL		; COS存贮地址→HL
2374	CDF420	CALL	FDIV		; 计算SIN(x)/COS(x)
2377	D1	POP	DE		; COS(x)退栈。
2378	D1	POP	DE		
2379	C9	RET			; 返回

框图:

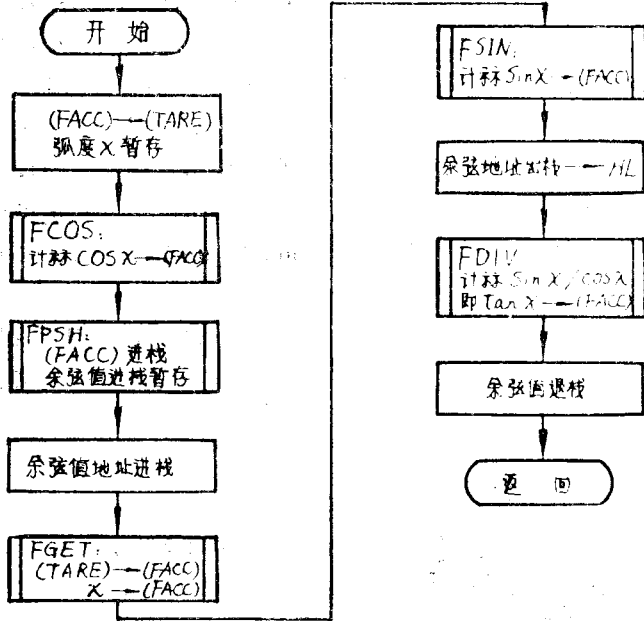


图 4-29

#### 4—30 常用对数函数FLOG

- 说明:
1. 本子程序的功能就是求 $\lg x$ 。初值 $x$ 存贮在(FACC)中, 计算结果亦存(FACC)。
  2.  $\lg x = (\ln x) / (\ln 10)$ , 所以计算 $x$ 的常用对数只要先求出 $x$ 的自然对数再除以10的自然对数即可。本子程序在设立标志后就进入FLN子程序处理, 在处理完后再根据标志作除以 $\ln 10$ 的运算。 $\ln 10$ 存贮在2561H, 标号为FL10。
  3. 本子程序的清单和框图请看4—31。

#### 4—31 自然对数函数FLN

- 说明:
1. 本子程序的功能是计算自然对数 $\ln x$ , 存贮在(FACC), 结果 $\ln x$ 亦存(FACC)。
  2. 当 $x = 1$ 时, 直接置0(FACC)。当 $x \leq 0$ 时, 进位位置1返回。因此在返回时, 进位为1表示不正常计算, 进位为0表示有正常结果。
  3. 下面对子程序算法作简单介绍:

为计算 $\ln x$ , 设 $x = 2^r \cdot t$  其中 $r$ 为整数, 是 $x$ 的阶码;  $t$ 是小数, 是 $x$ 的

尾数，规格化后的尾数 $0.5 \leq t < 1$ 。

再设  $u = 2t - 1$ ，显然有  $0 \leq u < 1$

因此  $\ln x = \ln(2^{r-1} \cdot 2t) = \ln 2^{r-1} + \ln(2t) = (r-1) \times \ln 2 + \ln(1+u)$

前一项 $(r-1) \times \ln 2$ ，可把 $r-1$ 转换成浮点数乘以 $\ln 2$ 这一常数求得。

后一项 $\ln(1+u)$ 采用近似多项式计算，即

$$\ln(1+u) = u \times (0.999981 - 0.499470u + 0.328233u^2 - 0.225873u^3 \\ + 0.134639u^4 - 0.055150u^5 + 0.010757u^6)$$

两项相加就得到了 $\ln x$ 的值。

4. 本子程序调用的子程序有：存浮点数字子程序FGET，取浮点数字子程序FPUT，浮点数比较子程序FCMP，浮点数减法子程序FSUB，浮点数加法子程序FADD，浮点数乘法子程序FMUL，浮点数除法子程序FDIV（除法为FLOG入口时调用）。

程序：

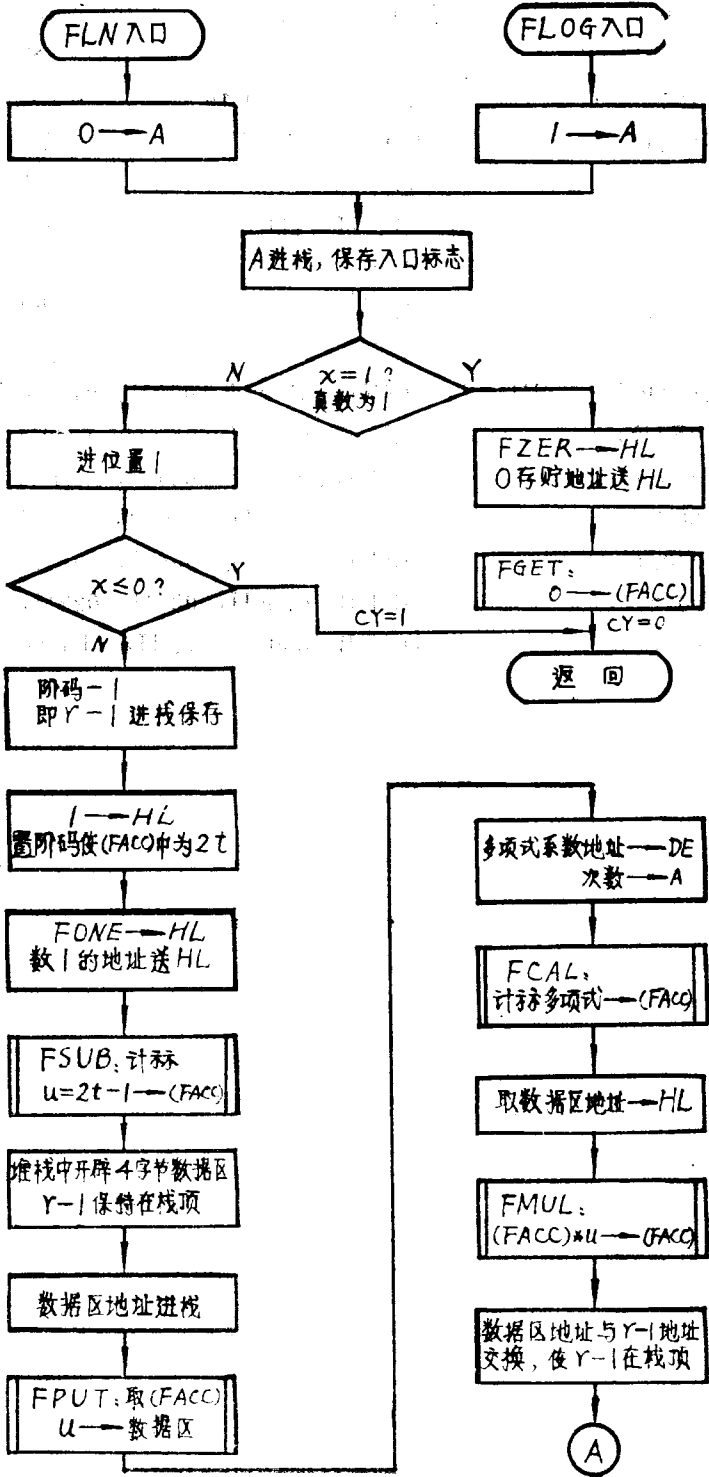
本程序清单中有两个入口，即常用对数入口FLOG及自然对数入口FLN。

```
237A 3E01      FLOG: LD A, 001H      ; 常用对数在A中置入口标志1
237C A7                AND A
237D C38123       JP FLN+1            ; 转FLN+1
2380 AF          FLN: XOR A                ; 自然对数在A中置入口标志0
2381 F5                PUSH AF           ; 入口标志A进栈保存
2382 21C225       LD HL, FONE          ; 真数与1比较，不等于1转
2385 CDA525       CALL FCMP          FLN1,等于1(FACC)置0返回
2388 C29323       JP NZ, FLN1
238B 21BE25       LD HL, FZER
238E CDBE21       CALL FGET
2391 E1          FLN0: POP HL
2392 C9                RET
2393 218922       FLN1: LD HL, FACC+1 ; 检测尾数
2396 7E                LD A, (HL)
2397 A7                AND A
2398 3F                CCF                ; 进位置1
2399 2B                DEC HL
239A CA9123       JP Z, FLN0         ; 真数≤0转FLN0返回
239D FA9123       JP M, FLN0
23A0 7E                LD A, (HL)        ; 阶码(r-1)进栈保存
23A1 3D                DEC A
23A2 F5                PUSH AF
23A3 3601         LD (HL), 001H     ; 计算u=2t-1, 存(FACC)
```

23A5	21C225	LD HL, FONE	
23A8	CD9A20	CALL FSUB	
23AB	F1	POP AF	; 在堆栈中开辟4个字节区,用来
23AC	21FCFF	LD HL, -004H	暂存u, 并且其始址进栈,为多
23AF	39	ADD HL, SP	项式最后乘u作准备
23B0	F9	LD SP, HL	
23B1	F5	PUSH AF	
23B2	E5	PUSH HL	
23B3	CD7822	CALL FPUT	
23B6	11F723	LD DE, FLGT	; 取系数地址
23B9	3E06	LD A, 006H	; 多项式次数存A
23BB	CD2325	CALL FCAL	; 计算多项式
23BE	E1	POP HL	; 取u地址
23BF	E5	PUSH HL	
23C0	CDA620	CALL FMUL	; 多项式乘u,完成计算
23C3	E1	POP HL	$\ln(1+u)$ , 即计算
23C4	F1	POP AF	$u \cdot (a_1u^6 + a_2u^5 + a_3u^4 + a_4u^3$
23C5	E5	PUSH HL	$a_5u^2 + a_6u + a_7)$
23C6	F5	PUSH AF	
23C7	CD7822	CALL FPUT	; $\ln(1+u)$ 送栈暂存(使用u占用的4个字节)
23CA	F1	POP AF	; 阶码(r-1)退栈,并送B
23CB	47	LD B, A	
23CC	F2D523	JP P, FLN2	; $r-1 \geq 0$ 转FLN2处理
23CF	0EFF	LD C, 0FFH	; $r-1 < 0$ 则C、D置值FFH
23D1	51	LD D, C	
23D2	C3D823	JP FLN3	; 转FLN3
23D5	AF	FLN2: XOR A	; C、D都置0
22D6	4F	LD C, A	
23D7	57	LD D, A	
23D8	CD0020	FLN3: CALL FLOT	; 将(r-1)转化成浮点数送
23DB	215D25	LD HL, FL2C	(FACC)
23DE	CDA620	CALL FMUL	; 计算 $(r-1) \times \ln 2$
23E1	E1	POP HL	; $\ln(1+u)$ 存贮地址退栈送HL
23E2	CD4320	CALL FADD	; 计算 $(r-1) \times \ln 2 + \ln(u+1)$
23E5	E1	POP HL	; $\ln(u+1)$ 退栈
23E6	E1	POP HL	
23E7	F5	PUSH AF	; 设置FLN出口标志

23E8	E1		POP HL	
23E9	F1		POP AF	; 入口标志退栈→AF
23EA	CAF423		JP Z, FLN4	; 自然对数A=0, 转 FLN4
23ED	216125		LD, HL, FL10	
23F0	CDF420		CALL FDIV	; 常用对数作除ln10运算
23F3	C9		RET	; 常对数返回
23F4	E5	FLN4,	PUSH HL	; 取出口标志送AF
23F5	F1		POP AF	
23F6	C9		RET	; 自然对数返回 ; 多项式系数:
23F7	FA581FD6	FLGT;	DEFB 0FAH, 058H, 01FH, 0D6H	
23FB	FC8F1D45		DEFB 0FCH, 08FH, 01DH, 045H	
23FF	FE44EF70		DEFB 0FEH, 044H, 0EFH, 070H	
2403	FE8C5A57		DEFB 0FEH, 08CH, 05AH, 057H	
2407	FF540715		DEFB 0FFH, 054H, 007H, 015H	
240B	FF8022BA		DEFB 0FFH, 080H, 022H, 0BAH	
240F	007FFE60		DEFB 000H, 07FH, 0FFH, 060H	

框图:



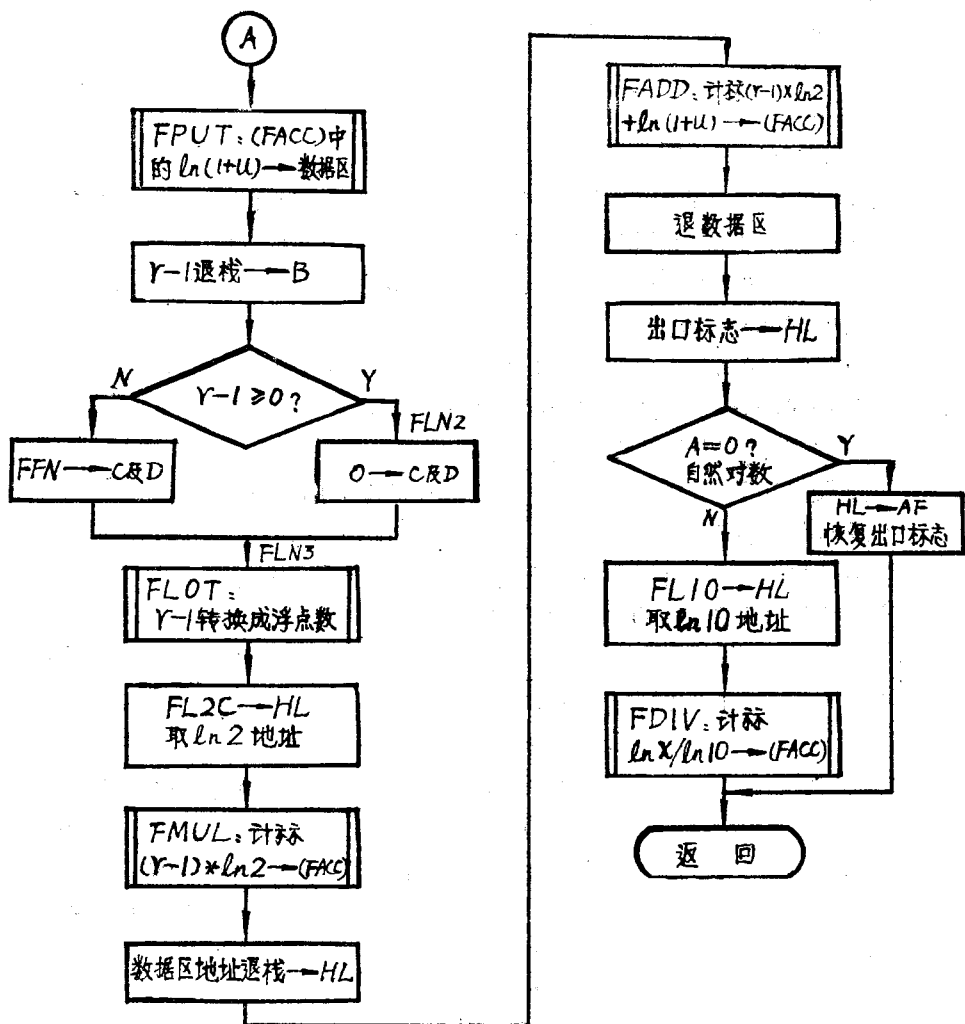


图 4-31

#### 4-32 反正切函数 FATN

说明: 1. 本子程序用来计算  $\arctan x$ , 初值  $x$  在 (FACC) 中, 其结果为弧度仍存 (FACC)。

2. 当  $x=0$  时, 结果亦为 0, 不必计算就返回。

当  $x < 0$  时, 结果亦为负, 但计算时,  $x$  取补 (即取其绝对值) 后按  $x > 0$  一样计算, 不过计算前设标志 2, 以便在计算后判别结果是否取补。

3. 当  $x > 1$  时, 取  $u = 1/x$  进行多项式计算, 并预先使标志 +1, 计算后根据标志进行  $\pi/2 - \arctan u$  计算, 得到  $\arctan x$ 。

当  $x \leq 1$  时, 直接进行多式计算, 为统一起见, 我们不仿设  $u = x$ 。



下面我们进一步来讨论多项式的计算。

我们知道 $\arctan u$ 的幂级数展开式为:

$$\arctan u = u - u^3/3 + u^5/5 - u^7/7 + u^9/9 - u^{11}/11 + \dots + (-1)^k x^{2k+1}/(2k+1) + \dots$$

其中 $u$ 在区间 $[-1, 1]$ 上。

按上述公式,取前几项进行近似计算,在 $u$ 比较接近1或-1时,其误差比较大,

在本子程序中采用了切比雪夫级数多项式进行近似计算,其计算多项式为:

$$\begin{aligned} \arctan u &= 0.99996u - 0.333174u^3 + 0.198079u^5 - 0.132334u^7 + 0.079622u^9 \\ &\quad - 0.033606u^{11} + 0.013573u^{13} \\ &= u \cdot ((((((0.013573u^2 - 0.033606) \cdot u^2 + 0.079622) \cdot u^2 - 0.132334) \cdot u^2 \\ &\quad + 0.198079) \cdot u^2 - 0.333174) \cdot u^2 + 0.99996) \end{aligned}$$

#### 4. 本子程序中调用的子程序有:

浮点数取补子程序FNEG, 浮点数比较子程序FCMP,

存浮点数子程序EGET, 取浮点数子程序FPUT,

浮点数加法子程序FADD, 浮点数乘法子程序FMUL,

浮点数除法子程序FDIV, 多项式计算子程序FCAL。

程序:

```
2413 AF FATN: XOR A ; A清0, 作为标志进栈
2414 F5 PUSH AF
2415 3A8922 LD A, (FACC+1) ; 取x尾数首字节检测
2418 A7 AND A
2419 C21E24 JP NZ, FAT1 ; x≠0转 FAT1, x=0
; 退栈取标志返回

241C F1 POP AF
241D C9 RET
241E F22824 FAT1: JP P, FAT2 ; x>0转FAT2
2421 CDEB21 CALL FNEG ; x<0取补
2422 F1 POP AF
2425 C602 ADD 002H ; 修改栈中的标志为2
2427 F5 PUSH AF
2428 21FCFF FAT2: LD HL, -004H ; 栈中开辟4个字节数据
; 区, 其栈地址进栈

242B 39 ADD HL, SP
242C F9 LD SP, HL
242D E5 PUSH HL
242E CD7822 CALL FPUT ; x送数据区
2431 21C225 LD HL, FONE ; 取常数1的地址并进栈
2434 E5 PUSH HL
```

2435	CDA525	CALL FCMP	; 与1比较
2438	E1	POP HL	
2439	FA5524	JP M, FAT3	; $x \leq 1$ 转 FAT3
243C	CA5524	JP Z, FAT3	
243F	CDBE21	CALL FGET	; 1送(FACC)
2442	E1	POP HL	; 取x在数据区地址
2443	E5	PUSH HL	
2444	CDF420	CALL FDIV	; $1/x \rightarrow (FACC)$
2447	E1	POP HL	
2448	E5	PUSH HL	
2449	CD7822	CALL FPUT	; $1/x \rightarrow$ 数据区
244C	E1	POP HL	
244D	C1	POP BC	
244E	D1	POP DE	
244F	F1	POP AF	
2450	3C	INC A	; 修改栈中标志, 标志 + 1
2451	F5	PUSH AF	
2452	D5	PUSH DE	
2453	C5	PUSH BC	
2454	E5	PUSH HL	
2455	E1	FAT3: POP HL	; 取数据区地址
2456	E5	PUSH HL	
2457	CDA620	CALL FMUL	; 计算 $u^2 \rightarrow (FACC)$
245A	118824	LD DE, FATT	; 取系数地址
245D	3E06	LD A, 006H	; 取多项式次数
245F	CD2325	CALL FCAL	; 计算多项式
2462	E1	POP HL	; 数据区地址退栈
2463	CDA620	CALL FMUL	; 多项式 $\times u \rightarrow (FACC)$
2466	F5	PUSH AF	
2467	C1	POP BC	
2468	E1	POP HL	
2469	E1	POP HL	
246A	F1	POP AF	
246B	1F	RRA	; 标志右移一位
246C	F5	PUSH AF	
246D	D27B24	JP NC, FAT4	; 没有进位, 即 $x < 1$ 转 FAT4
2470	CDEB21	CALL FNEG	; 有进位, $\pi/2 - \arctan u \rightarrow \arctan x$

2473	215125		LD HL, FPI2	
2476	CD4320		CALL FADD	
2479	F5		PUSH AF	; 标志送BC
247A	C1		POP BC	
247B	F1	FAT4:	POP AF	; 标志退栈, 再右移一位
247C	1F		RRA	
247D	D28524		JP NC, FAT5	; 无进位, 即 $x > 0$ , 转F AT5
2480	CDEB21		CALL FNEG	; 有进位, $x < 0$ , $\arctan x$ 取补
2483	A7		AND A	; 置标志返回
2484	C9		RET	
2485	C5	FAT5:	PUSH BC	; 标志送AF 返回
2486	F1		POP AF	
2487	C9		RET	

; 多项式系数:

2488	F96F9D52	FATT:	DEFB 0F9H, 06FH, 09DH, 052H
248C	FCBB2CA3		DEFB 0FCH, 0BBH, 02CH, 0A3H
2490	FD51898F		DEFB 0FDH, 051H, 089H, 08FH
2494	FEBC3E94		DEFB 0FEH, 0BCH, 03EH, 094H
2498	FE656A91		DEFB 0FEH, 065H, 06AH, 091H
249C	FFAAB520		DEFB 0FFH, 0AAH, 0B5H, 020H
24A0	007FFFDF		DEFB 000H, 07FH, 0FFH, 0DFH

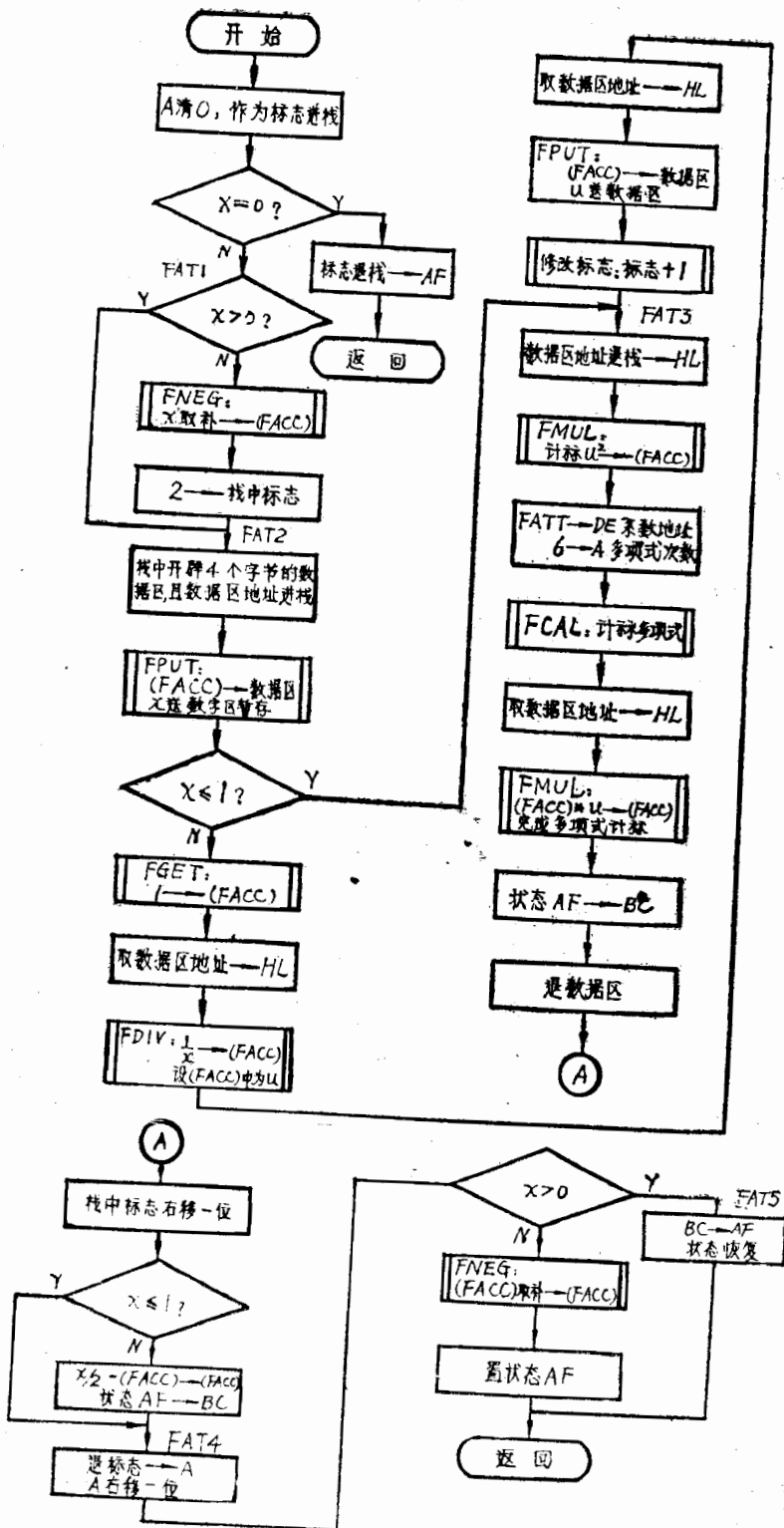


图 4-32

#### 4—33 指数函数FEXP

说明：1.本子程序计算指数函数  $e^x$ ，其中  $x$  初值存贮在 (FACC)，计算结果亦存 (FACC)。

2.由于浮点数格式位数限制了数的表示范围（不超过  $10^{38}$ ），因此对于指数函数的自变量  $x$  也有限制：

当  $x \geq 87.3365$  时，进位位置 1，返回；

当  $x < -87.3365$  时，结果置 0，正常返回。

3.指数函数的计算方法：

$$\because e^x = 2^{x \log_2 e} = 2^{x / \ln 2}$$

设  $x / \ln 2 = r + t$ ，其中  $r$  为整数部分， $t$  为小数部分，且  $|r| < 256$ ，用一个字节存贮。

$$\therefore e^x = 2^r \cdot 2^t = 2^{r+1} \cdot 2^{t-0.5} / \sqrt{2}$$

设  $t - 0.5 = u$ ，则上式可写成  $e^x = 2^{r+1} \cdot 2^u / \sqrt{2}$

我们知道如果求出  $2^u / \sqrt{2}$ ，则在其阶码部分加上  $r + 1$  就能得到  $e^x$  的值。下面我们进一步考察  $2^u / \sqrt{2}$  的幂级数展开。

$$2^u / \sqrt{2} = (1 + \ln 2 \cdot u + (\ln 2)^2 / 2! \cdot u^2 + \dots + (\ln 2)^n / n! \cdot u^n + \dots) / \sqrt{2}$$

如果我们取前几项进行近似计算，当然是  $u$  越小，误差越小，这就是为什么  $u = t - 0.5$  的原因。为了进一步提高精度，本子程序中采用了切比雪夫多项式进行近似计算，其公式为：

$$2^u / \sqrt{2} = 0.707107 + 0.490129u + 0.169866u^2 + 0.039246u^3 + 0.006801u^4 + 0.000948u^5 + 0.000109u^6$$

4.本子程序调用的子程序有：

存浮点数字子程序 FGET，取浮点数字子程序 FPUT，

浮点数比较子程序 FCMP，浮点数取补子程序 FNEG，

二进制数转换成浮点数字子程序 FLOT，浮点数取整子程序 FIXI，

浮点数加法子程序 FADD，浮点数减法子程序 FSUB，

浮点数除法子程序 FDIV，多项式计算子程序 FCAL。

程序：

```

24A4  3A8922  FEXP:  LD  A, (FACC+1)
24A7  A7      AND  A           ; 检测x
24A8  FAB624  JP  M, FEX0     ; x < 0 转 FEX0
24AB  216925  LD  HL, FP87
24AE  CDA525  CALL FCMP       ; x 与 87.3365 比较
24B1  37      SCF           ; 进位位置 1
24B2  F0     RET  P         ; x ≥ 87.3365 返回
24B3  C3C624 JP  FEX1       ; 否则转 FEX1
    
```

24B6	216D25	FEX0:	LD HL, FM87	
24B9	CDA525		CALL FCMP	; x与-87.3365比较
24BC	F2C624		JP P, FEX1	; $x \geq -87.3365$ 转 FEX1
24BF	21BE25		LD HL, FZER	; 否则置0(FACC), 然后返回
24C2	CDBE21		CALL FGET	
24C5	C9		RET	
24C6	21FCFF	FEX1:	LD HL, -004H	; 栈中开辟4字节数 据区
24C9	39		ADD HL, SP	
24CA	F9		LD SP, HL	
24CB	E5		PUSH HL	; 数据区地址进栈
24CC	215D25		LD HL, FL2C	
24CF	CDF420		CALL FDIV	; $x/ln2 \rightarrow (FACC)$
24D2	CD7825		CALL FIXI	; (FACC)取整 $\rightarrow D$ 、 C、B,实际上 $[x/ln2]$ 只要用B寄存器, D、C都为0
24D5	E1		POP HL	
24D6	C5		PUSH BC	; D、C、B进栈暂存
24D7	E5		PUSH HL	
24D8	D5		PUSH DE	
24D9	CD7822		CALL FPUT	; $x/ln2 \rightarrow$ 数据区
24DC	D1		POP DE	
24DD	E1		POP HL	
24DE	C1		POP BC	
24DF	C5		PUSH BC	; 取暂存的整数 $\rightarrow D$ 、 C、B,将其转化成浮 点数
24E0	E5		PUSH HL	
24E1	CD0020		CALL FLOT	
24E4	CDEB21		CALL FNEG.	
24E7	E1		POP HL	
24E8	CD4320		CALL FADD	; 计算 $t = x/ln2 -$ $[x/ln2] \rightarrow (FACC)$ , 规格化的t为 $0.5 \leq$ $t < 1$

24EB	21C625		LD HL, FP50	
24EE	CD9A20		CALL FSUB	; 计算 $u = t - 0.5 \rightarrow$ (FACC)
24F1	110725		LD DE, FEXT	; 取系数地址
24F4	3E06		LD A, 06H	; 取多项式次数
24F6	CD2325		CALL FCAL	; 计算多项式 $2^n / \sqrt{2}$ $\rightarrow$ (FACC)
24F9	C1		POP BC	
24FA	04		INC B	; $r + 1 \rightarrow B$
24FB	218822		LD HL, FACC	
24FE	7E		LD A, (HL)	
24FF	80		ADD B	; 阶码加 $(r + 1)$
2500	77		LD (HL), A	
2501	23		INC HL	
2502	7E		LD A, (HL)	; 置状态
2503	A7		AND A	
2504	E1		POP HL	; 退数据区
2505	E1		POP HL	
2506	C9		RET	; 多项式系数:
2507	F372B427	FEXT;	DEFB 0F3H,072H,0B4H,027H	
250B	F67C3280		DEFB 0F6H,07CH,032H,080H	
250F	F96F6D57		DEFB 0F9H,06FH,06DH,057H	
2513	FC506098		DEFB 0FCH,050H,060H,098H	
2517	FE56F8A6		DEFB 0FEH,056H,0F8H,0A6H	
251B	FF7D7919		DEFB 0FFH,07DH,079H,019H	
251F	005A8279		DEFB 000H,05AH,082H,079H	

框图:

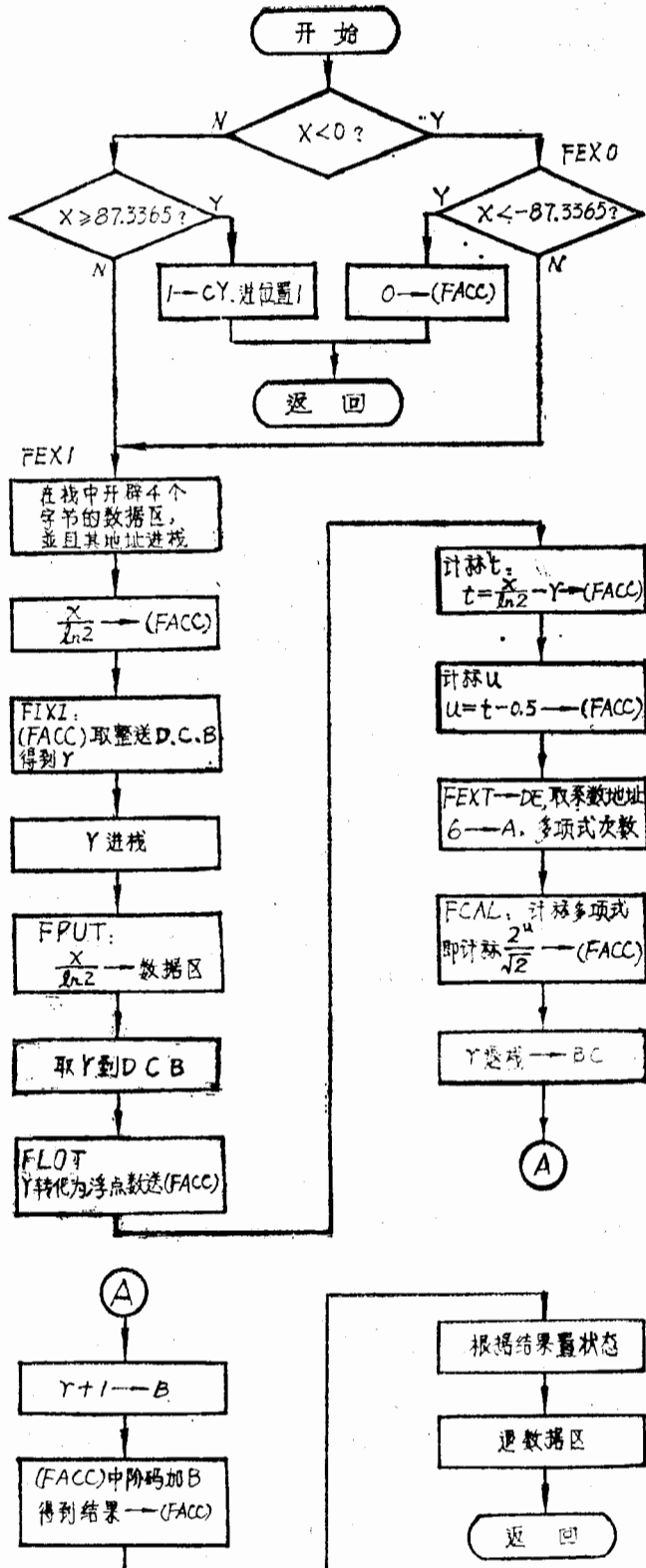


图 4-33



#### 4-34 多项式计算子程序FCAL

说明: 1. 子程序实现计算多项式  $a_1x^n + a_2x^{n-1} + a_3x^{n-2} + \dots + a_nx + a_{n+1}$  的值。入口时,  $x$  在 (FACC) 中,  $n$  在 A 中,  $a_1$  的地址在 DE 中, 每个  $a_i$  占有 4 个字节,  $a_1, a_2, \dots, a_{n+1}$  顺序存贮。

出口时, 多项式的值存贮在 (FACC) 中。

2. 多项式的计算采用秦九韶算法即按下列顺序计算:

$$(\dots((a_1x + a_2)x + a_3)x + \dots)x + a_{n+1}$$

3. 本子程序调用了下列子程序: 取浮点数字子程序 FPUT, 浮点数加法子程序 FADD, 浮点数乘法子程序 FMUL。

程序:

2523	E5	FCAL:	PUSH HL	; 栈指针下移4字节,作为数据区
2524	E5		PUSH HL	
2525	F5		PUSH AF	; 多项式次数A进栈
2526	210200		LD HL,002H	
2529	39		ADD HL,SP	; 数据区地址进栈
252A	E5		PUSH HL	
252B	D5		PUSH DE	; DE进栈
252C	CD7822		CALL FPUT	; (FACC) → 数据区
252F	E1		POP HL	
2530	E5		PUSH HL	; 取DE → HL; 系数地址
2531	CDA620	FCL1:	CALL FMUL	; (FACC) × (DE) → (FACC)
2534	E1		POP HL	
2535	23		INC HL	; DE + 4, 指向下一系数地址
2536	23		INC HL	
2537	23		INC HL	
2538	23		INC HL	
2539	E5		PUSH HL	
253A	CD4320		CALL FADD	; (FACC) + 下一系数 → (FACC)
253D	D1		POP DE	
253E	E1		POP HL	
253F	F1		POP AF	
2540	3D		DEC A	; 取次数 - 1
2541	F5		PUSH AF	
2542	E5		PUSH HL	
2543	D5		PUSH DE	
2544	C23125		JP NZ,FCL1	; 次数不为0转FCL1

2547	E1	POP HL	; 退系数地址
2548	D1	POP DE	; 退数据区地址
2549	D1	POP DE	; 退多项式次数
254A	D1	POP DE	; 退数据区
254B	D1	POP DE	
254C	C9	RET	
254D	FB477D1B	FP18;DEFB 0FBH,047H,07DH,01BH	; 1°的弧度数
2551	016487EC	FPI2;DEFB 001H,064H,087H,0ECH	; $\pi/2$
2555	026487ED	FPAI;DEFB 002H,064H,087H,0EDH	; $\pi$
2559	036487EC	F2PI;DEFB 003H,064H,037H,0ECH	; $2\pi$
255D	0058B90B	FL2C;DEFB 000H,058H,0B9H,00BH	; $\ln 2$
2561	0249AEC7	FL10;DEFB 002H,049H,0AEH,0C7H	; $\ln 10$
2565	075A0000	FP90;DEFB 007H,05AH,000H,000H	; 90
2569	07575627	FP87;DEFB 007H,057H,056H,027H	; 87.3365
256D	07A8A9D9	FM87;DEFB 007H,0A8H,0A9H,0D9H	; -87.3365

框图:

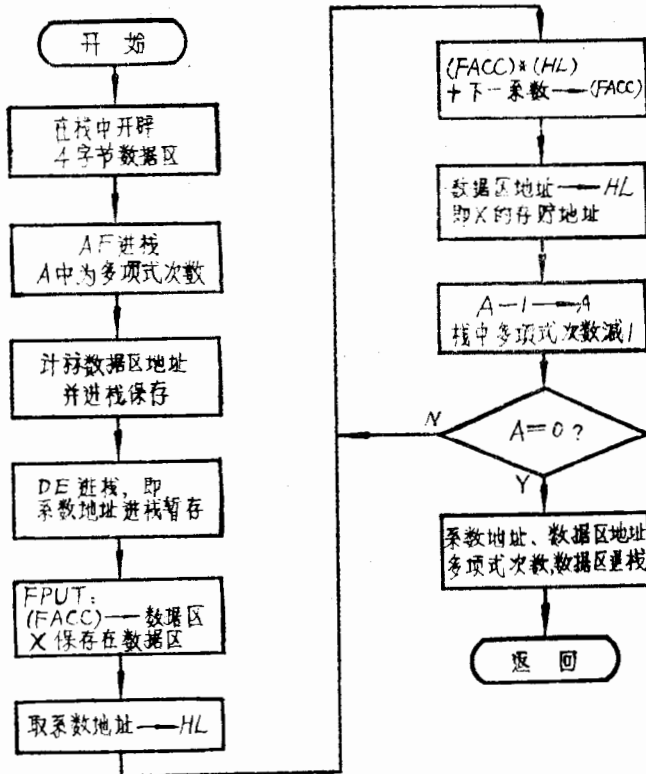


图4-34

#### 4-35 取整函数FIXF

说明: 1. 本子程序实现浮点数取整功能。设初值X在(FACC)中, 运行本程序后, 得到不大于X的最大整数的浮点形式存(FACC)中。

例 X = 3.5, 取整后, (FACC)中为3的浮点格式: 02600000H。

X = -3.5, 取整后, (FACC)中为-4的浮点格式: 03C00000H。

2. 本子程序调用了取整子程序FIXI和二进制转化成浮点数字子程序FLOT。

程序:

2571	CD7825	FIXF:	CALL FIXI	; (FACC)中浮点数取整 →B、C、D寄存器
2574	D8		RET C	; 当浮点数阶码 $\geq 24$ 时, 超过整 数格式范围, 立即返回
2575	C30020		JP FLOT	; 否则再将B、C、D中的整数 转化成浮点数存(FACC)

#### 4-36 取整子程序FIXI

说明: 1. 本子程序的功能只是将(FACC)中的浮点数取不大于它的整数存D、C、B寄存器中, 而不改变(FACC)中的初值。

2. 当初值 $|X| < 1$ 时, D、C、B中为0, 立即返回。当X的阶码 $\geq 24$ 时, 该数的整数部分用24位无法表示, 进位置1, 表示不正常返回。

3. 本子程序调用了浮点数进栈子程序FPSH和B、C、D寄存器右移子程序RRBD。

程序:

2578	218822	FIXI:	LD HL, FACC	; 浮点数X地址→HL
257B	010000		LD BC, 00H	; B、C、D、E寄存器清0
257E	C5		PUSH BC	
257F	D1		POP DE	
2580	7E		LD A, (HL)	; 取阶码检测
2581	A7		AND A	
2582	F8		RET M	; X为小数返回
2583	C8		RET Z	
2584	D618		SUB 24	; 阶码-24→A
2586	37		SCF	; 进位置2
2587	F0		RET P	; 阶码 $\geq 24$ 返回
2588	F5		PUSH AF	; A进栈
2589	CDD421		CALL FPSH	; X进栈

258C	D1		POP DE	; X退栈→E、D、C、B
258D	C1		POP BC	
258E	F1		POP AF	; 阶差退栈→E
258F	5F		LD E, A	
2590	A7	FIX1:	AND A	; 进位清0
2591	1C		INC E	; E + 1 → E
2592	C8		RET Z	; E = 0返回
2593	7A		LD A, D	; 符号位→c <sub>y</sub>
2594	17		RLA	
2595	CD9B25		CALL RRBD	; D、C、B带符号右移一位
2598	C39025		JP FIX1	; 转FIX1

框图:

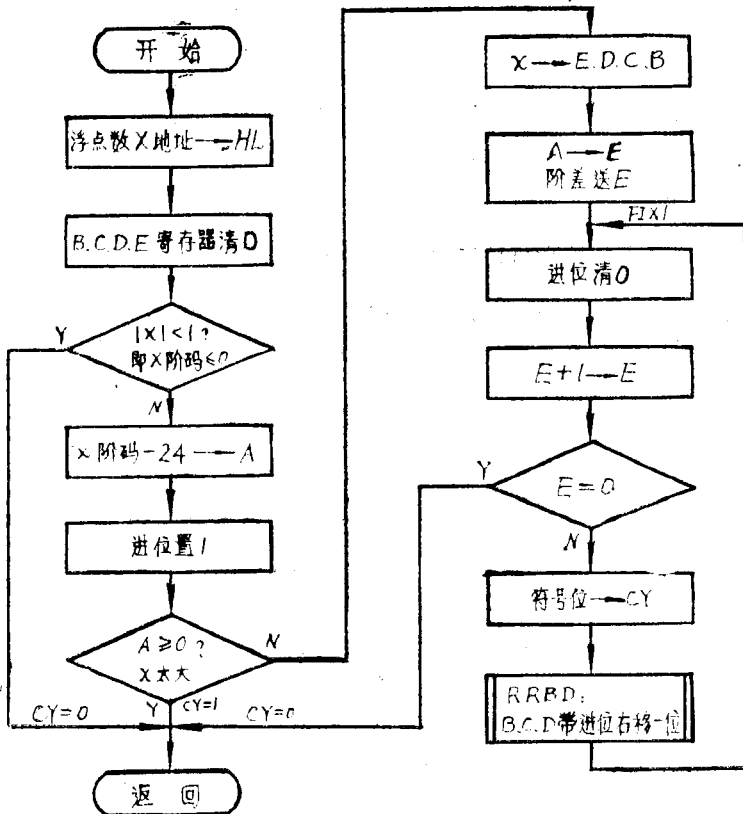


图 4-36

#### 4—37 B、C、D寄存器右移子程序RRBD

- 说明：1.本子程序实现D、C、B带进位右移一位。原进位移入D的最高位，B的最低位移入进位位。
- 2.一般D、C、B组成二十四位二进制数，其中D为高位字节，B为低位字节。

程序：

```
259B 7A RRBD: LD A, D ; D带进位右移一位
259C 1F RRA
259D 57 LD D, A
259E 79 LD A, C ; C带进位右移一位
259F 1F RRA
25A0 4F LD C, A
25A1 78 LD A, B ; B带进位右移一位
25A2 1F RRA
25A3 47 LD B, A
25A4 C9 RET
```

#### 4—38 浮点数比较子程序FCMP

- 说明：1.本子程序实现(FACC)中的浮点数与HL指示的存贮区中的浮点数比较。即(FACC)-(HL)，不送结果，只置标志：  
当 (FACC)>(HL)时，有P标志；  
当 (FACC)<(HL)时，有M标志；  
当 (FACC)=(HL)时，有Z标志、P标志。
- 2.本子程序中调用了存浮点数字子程序FGET，取浮点数字子程序FPUT和浮点数减法子程序FSUB。
- 3.本子程序清单中列有三个常数，即 FZER: 0, FONE: 1, FP50: 0.5。这三个常数是为其子程序调用的，与本子程序无关。

程序：

```
25A5 E5 FCMP: PUSH HL ; 开辟四个字节的数区
25A6 E5 PUSH HL
25A7 E5 PUSH HL ; HL进栈
25A8 210200 LD HL, 02H ; 计算数据区地址并进栈
25AB 39 ADD HL, SP
25AC E5 PUSH HL
25AD CD7822 CALL FPUT ; (FACC)→数据区
```

25B0	E1	POP HL	; 数据区地址出栈
25B1	E3	EX (SP), HL	; 比较数地址交换到HL中
25B2	CD9A20	CALL FSUB	; (FACC) - (HL) → (FACC)
25B5	E1	POP HL	; 数据区地址出栈
25B6	F5	PUSH AF	; 标志进栈
25B7	CDBE21	CALL FGET	; 数据区的数 → (FACC), 恢复被比较数
25BA	F1	POP AF	; 标志退栈
25BB	D1	POP DE	; 数据区退栈
25BC	D1	POP DE	
25BD	C9	RET	; 返回
25BE	00000000	FZFR: DEFB 00, 00, 00, 00	; 0
25C2	01400000	FONE: DEFB 01H, 40H, 00H, 00H	; 1
25C6	00400000	FP50: DEFB 00H, 40H, 00H, 00H	; 0.5

框图:

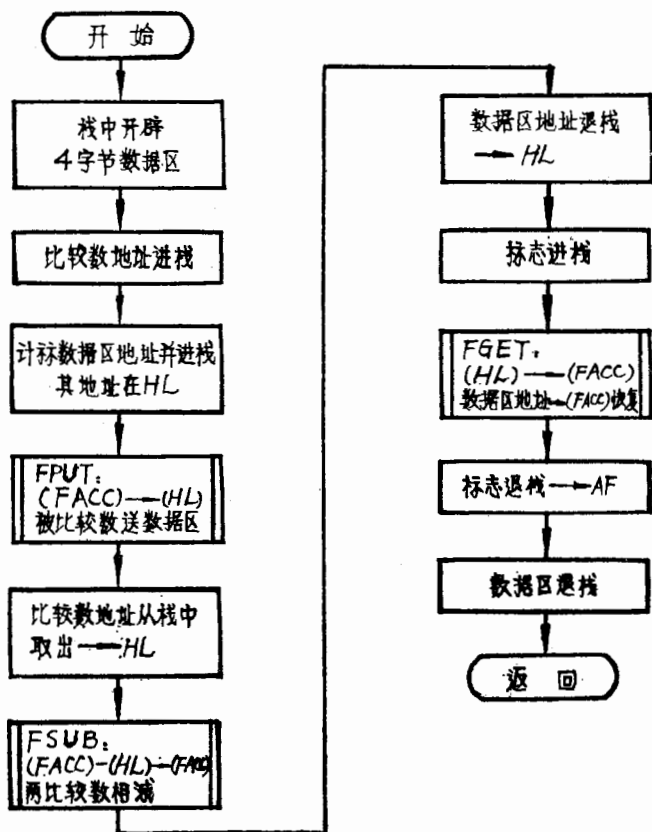


图4-38

## 第五章 分类与检索

### 5-1 找最大数(无符号)

- 说明: 1. 数据块存放在以BLOCK开始的连续单元中, 并假设数据块中的数全是8位不带符号的二进制数。  
2. 数据块的长度放在存储单元 SIZE 中, 程序完成将最大值放在存储单元 MAX 中。

程序:

2000	211920	START: LD HL, BLOCK	; 置数据地址
2003	DD211820	LD IX, SIZE	
2007	DD4600	LD B, (IX)	; 置计数器
200A	7E	LD A, (HL)	
200B	23	INC HL	
200C	BE	LOOP: CP (HL)	; 比较
200D	3001	JR NC, NEXT	
200F	7E	LD A, (HL)	
2010	23	NEXT: INC HL	; 不要交换
2011	05	DEC B	
2012	20F8	JR NZ, LOOP	; 计算完否
2014	322320	LD (MAX), A	; 完,最大数存 MAX 单元
2017	C9	RET	
2018	0A	SIZE: DB 0AH	; 数据个数
2019	(000A)	BLOCK: DS 0AH	; 数据
2023	(0001)	MAX: DS 1	; 最大值

框图:

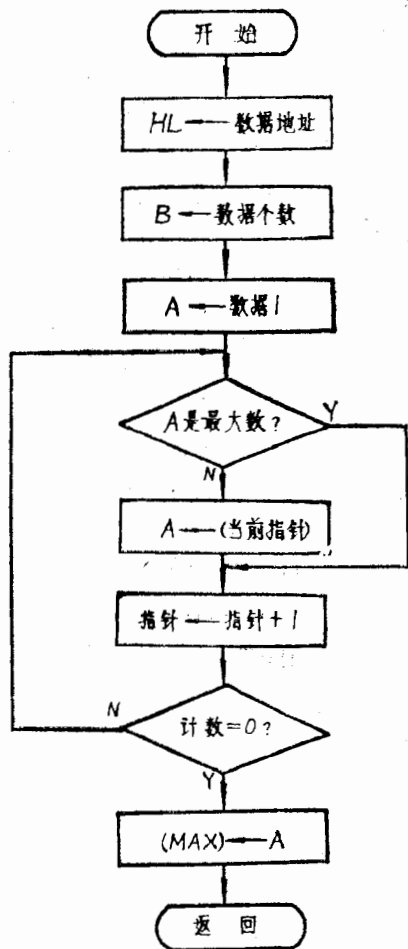


图 5-1

## 5-2 找最大数 (带符号)

- 说明:
1. 本程序功能是找出一个数据块中的最大元素。数据块的长度放在 COUNT 单元中。
  2. 要找的数据块存放在以 DATA 开始的连续单元中, 并且均为八位带符号的二进制数。
  3. 程序执行完后将最大值放在存储单元 MAX 中。



程序:

2000	212320	START:	LD HL, DATA	; 置数据地址
2003	112D20		LD DE, MAX	
2006	FD212220		LD IY, COUNT	
200A	FD4600		LD B, (IY)	; 置循环次数初值
200D	05		DEC B	
200E	7E		LD A, (HL)	
200F	23		INC HL	
2010	BE	LOOP:	CP (HL)	; 数据比较
2011	FA1920		JP M, SIGN	
2014	EA1C20		JP PE, EXCH	
2017	1804		JR UNEX	
2019	EA1D20	SIGN:	JP PE, UNEX	
201C	7E	EXCH:	LD A, (HL)	; 把当前值送A,
201D	23	UNEX:	INC HL	; 不交换修改指针
201E	10F0		DJNZ LOOP	; 未完则循环
2020	12		LD (DE), A	; 最大值存MAX单元。
2021	C9		RET	
2022	0A	COUNT:	DB 0AH	; 数据个数
2023	(000A)	DATA:	DS 0AH	; 数据
202D	(0001)	MAX:	DS 1	; 最大值

框图: 参考前页无符号最大值程序框图。

### 5-3 顺序检索

- 说明：1.本程序功能是对指定内存区域进行线性搜索，判断要搜索的表格中是否有给定的关键字。
- 2.表格存放在以TABLE开始的连续单元中，表格的长度存放在 COUNT单元中。
- 3.给出的关键字存放在KEY单元中，查出的关键字序号放在SEARCH单元中，若未查到则SEARCH单元置零。

程序：

2000	211E20	START:	LD HL, TABLE	; 置地址指针
2003	010600		LD BC, COUNT	; 置计数器
2006	1600		LD D, 0	
2008	3A1C20		LD A, (KEY)	; 送入关键字
200B	14	LOOP:	INC D	; 搜索次数
200C	EDA1		CPI	; 比较
200E	2807		JR Z, OUT2	
2010	E21520		JP PO, OUT1	
2013	18F6		JR LOOP	
2015	1600	OUT1:	LD D, 0	; 未找到, D置零
2017	7A	OUT2:	LD A, D	
2018	321D20		LD (SEARCH), A	; 输出搜索次数
201B	C9		RET	
	(0006)	COUNT:	EQU 06H	; 数据个数
201C	08	KEY:	DB 08	; 关键字
201D	(0001)	SEARCH:	DS 1	; 搜索次数
201E	38E8085A	TABLE:	DB 56, -24, 08, 90,	; 数据
	2237		34, 55	

框图:

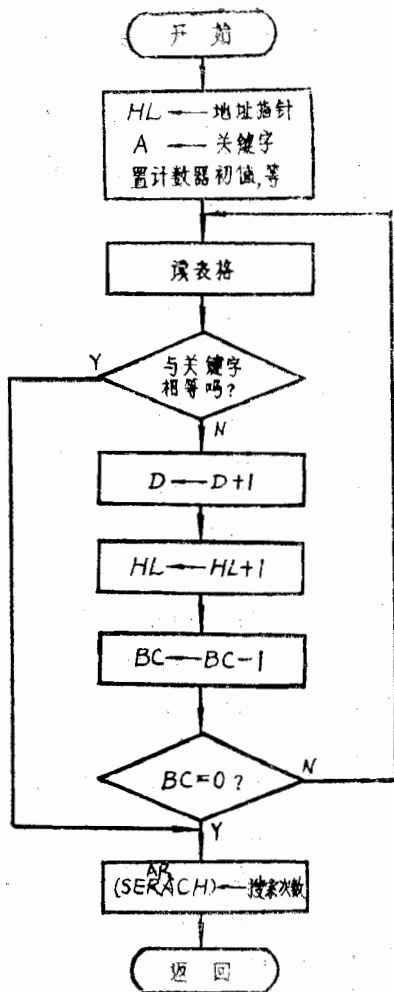


图 5-3

#### 5-4 对分搜索

- 说明:
1. 本程序实现对指定内存区域中已按大小次序排好的数组, 进行对分搜索。对于较大的数组能有效地减少搜索次数。
  2. 数组元素的个数存放在COUNT单元中, 数组存放在以TABLE单元开始的连续单元中。
  3. 要搜索的关键字存放在KEY单元中, 若被搜索的字找到, 则搜索次数存放在SEARCH单元中, 否则SEARCH单元置成0FFH。

程序:

2000	0600	START:	LD B, 0	; 区间上限置零
2002	50		LD D, B	
2003	260A		LD H, COUNT	; 设区间下限
2005	3A3C20		LD A, (KEY)	; 输入关键字
2008	4F		LD C, A	
2009	2E01		LD L, 01H	; 搜索次数
200B	DD213E20	LOOP:	LD IX, TABLE	; 置表的首地址
200F	78		LD A, B	
2010	84		ADD A, H	
2011	CB3F		SRL A	
2013	B8		CP B	
2014	281F		JR Z, OUT1	; 未找到
2016	5F		LD E, A	
2017	DD19		ADD IX, DE	
2019	DD7E00		LD A, (IX)	
201C	B9		CP C	; 与关键字比较
201D	CA3720		JP Z, OUT2	; 相等, 则找到
2020	FA2820		JP M, SIGN	
2023	EA2B20		JP PE, SMALL	
2026	1808		JR BIG	
2028	EA3020	SIGN:	JP PE, BIG	
202B	43	SMALL:	LD B, E	; 关键字 > (IX)
202C	2C		INC L	
202D	30B20		JP LOOP	
2030	63	BIG:	LD H, E	; 关键字 < (IX)
2031	2C		INC L	
2032	C30B20		JP LOOP	
2035	2EFF	OUT1:	LD L, 0FFH	; 未找到
2037	7D	OUT2:	LD A, L	; 找到关键字
2038	323D20		LD (SEARCH), A	
203B	C9		RET	
	(000A)	COUNT:	EGU 0AH	; 数据个数
203C	(0001)	KEY:	DS 1	; 关键字
203D	(0001)	SEARCH:	DS 1	; 搜索次数
203E	(000A)	TABLE:	DS 0AH	; 表

框图:

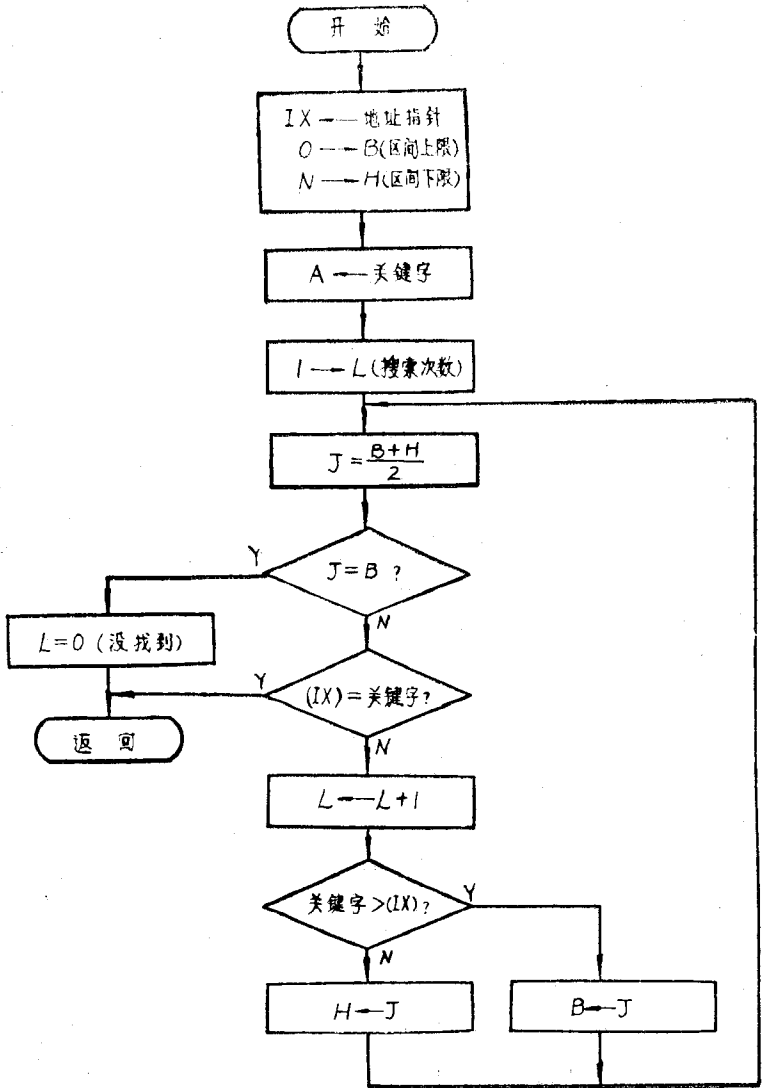


图 5-4

## 5—5 八位排序（无符号）

说明：1.本程序将一无符号二进制数数组按降序排序。所采用的排序方法为“气泡分类法”。

2.数组的长度放在存贮单元SIZE中，而数组本身从存贮单元DATA开始存放。

程序：

2000	0E00	SORT:	LD C, 0	; 对换标志清零
2002	211C20		LD HL, SIZE	
2005	46		LD B, (HL)	; 置计数器
2006	05		DEC B	
2007	211D20		LD HL, DATA	
200A	7E	NEXT:	LD A, (HL)	; 从数组中取一个元素
200B	23		INC HL	
200C	BE		CP (HL)	; 该元素<下一个元素吗?
200D	3007		JR NC, CNT	; 否。不需对换
200F	56		LD D, (HL)	; 是。两个元素对换
2010	77		LD (HL), A	
2011	2B		DEC HL	
2012	72		LD (HL), D	
2013	23		INC HL	
2014	0E01		LD C, 1	; 置对换标志
2016	10F2	CNT:	DJNZ NEXT	
2018	0D		DEC C	
2019	28E5		JR Z, SORT	; 有交换，再做一遍
201B	C9		RET	
201C	05	SIZE:	DB 05	
201D	247F2D02	DATA:	DB 36, 127, 45, 02,	
	43			
			67	

框图:

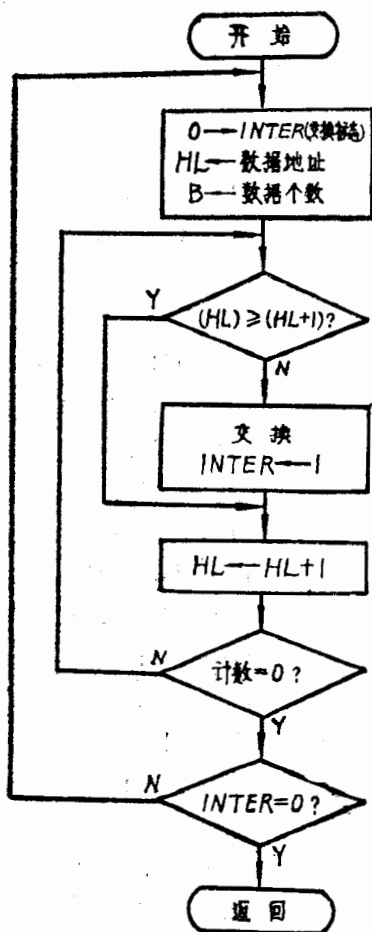


图 5-5

### 5-6 八位排序（带符号）

说明：1. 若有一数组具有N个带符号的二进制数，希望按升序将它们排列，则可调用本程序。

2. 数组的长度放在存储单元 SIZE 中，而数组本身从存储单元 DATA 开始连续存放。

程序:

2000	D9	SORT:	EXX	
2001	1E02	START:	LD E, 02H	
2003	DD213E20	LOOP:	LD IX, DATA	; 置地址指针
2007	3A3D20		LD A, (SIZE)	; 数据个数送A
200A	4F		LD C, A	
200B	6F		LD L, A	
200C	57		LD D, A	
200D	0600		LD B, 0	
200F	DD09		ADD IX, BC	
2011	DD2B		DEC IX	; 指针指向数据底部
2013	DD7E00	LOOP1:	LD A, (IX)	
2016	DD46FF		LD B, (IX-1)	
2019	B8		CP B	; 比较
201A	FA2220		JP M, SIGN	
201D	EA2520		JP PE, EXCH	
2020	180A		JR UNCH	
2022	EA2C20	SIGN:	JP PE, UNCH	
2025	DD77FF	EXCH:	LD (IX-1), A	; 交换
2028	DD7000		LD (IX+0), B	
202B	4A		LD C, D	; 改变交换标志
202C	DD2B	UNCH:	DEC IX	; 修改地址指针
202E	15		DEC D	
202F	7A		LD A, D	
2030	BB		CP E	
2031	F21320		JP P, LOOP1	; 小循环未结束
2034	79		LD A, C	
2035	BD		CP L	
2036	2803		JR Z, DONE	; 数组排好, 转结束
2038	1C		INC E	
2039	18C8		JR LOOP	
203B	D9	DONE:	EXX	
203C	C9		RET	
203D	0A	SIZE:	DB 0AH	; 数据个数
203E	(000A)	DATA:	DS 0AH	; 数据



框图:

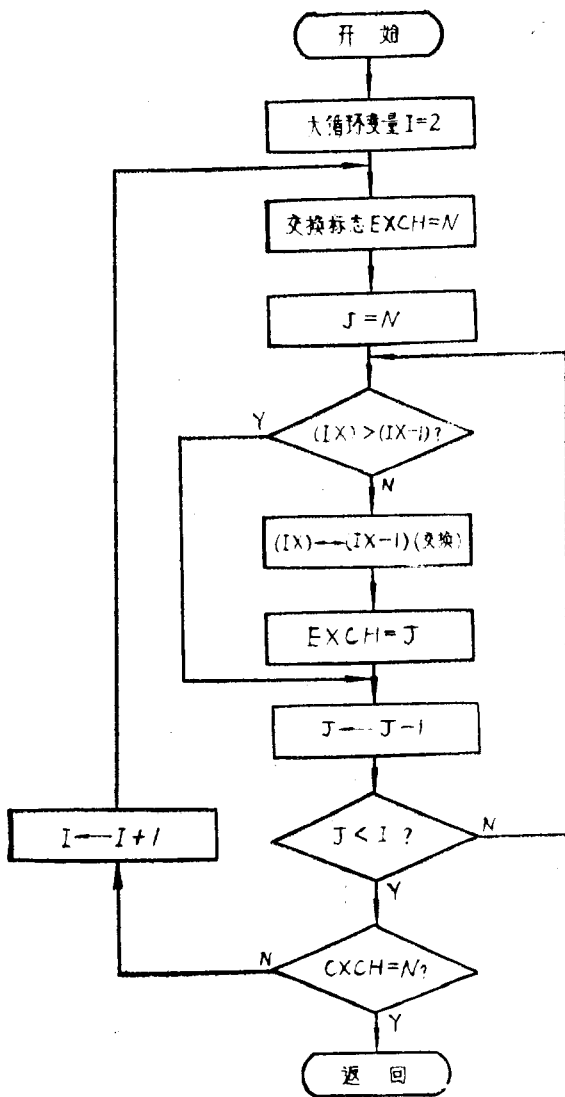


图 5-5

## 5-7 十六位排序（无符号）

- 说明：1. 本程序将一无符号二进制数数组按降序排序，数组的每个元素均为十六位无符号的二进制数，用相邻的两个字节表示，元素的高八位放在较高地址单元，低八位放在较低地址单元。
2. 数组元素的个数放在存贮单元 SIZE 中，而数组本身从存贮单元 DATA 开始存放。
3. 程序使用寄存器 IX、IY，而对其它寄存器没有影响。

程序：

```

2000 08          SORT:  EX AF, AF'
2001 D9          EXX
2002 0E00        START: LD C, 0          ; 置交换标志
2004 3A4C20      LD A, (SIZE)        ; 置数据长度
2007 47          LD B, A
2008 05          DEC B
2009 DD214D20    LD IX, DATA        ; 置指针1
200D FD214F20    LD IY, DATA + 2    ; 置指针2
2011 DD6E00      LOOP:  LD L, (IX)    ; 取数
2014 DD23        INC IX
2016 DD6600      LD H, (IX)
2019 FD5E00      LD E, (IY)
201C FD23        INC IY
201E FD5600      LD D, (IY)
2021 B7          OR A
2022 3E02        LD A, 2
2024 ED52        SBC HL, DE          ; HL - DE > 0 ?
2026 3014        JR NC, UNEX
2028 DD2B        EXCH:  DEC IX        ; 数据交换
202A FD2B        DEC IY
202C 3E01        LD A, 1
202E DD6E00      EXCH1: LD L, (IX)
2031 FD5E00      LD E, (IY)
2034 FD7500      LD (IY), L
2037 DD7300      LD (IX), E
203A 0E01        LD C, 1          ; 置交换标志
203C DD23        UNEX:  INC IX        ; 修改指针
203E FD23        INC IY
2040 3D          DEC A

```

2041	CA2E20	IP Z, EXCH1	
2044	10CB	DJNZ LOOP	; 小循环结束否
2046	0D	DEC C	
2047	28B9	JR Z, START	; 有交换则继续循环
2049	08	EX AF, AF'	
204A	D9	EXX	
204B	C9	RET	
204C	03	SIZE: DB 03H	; 数据个数
204D	D119603F	DATA: DB 0D1H,19H,60H,	; 数据
	2AB5	3FH,2AH,0B5H	

框图:

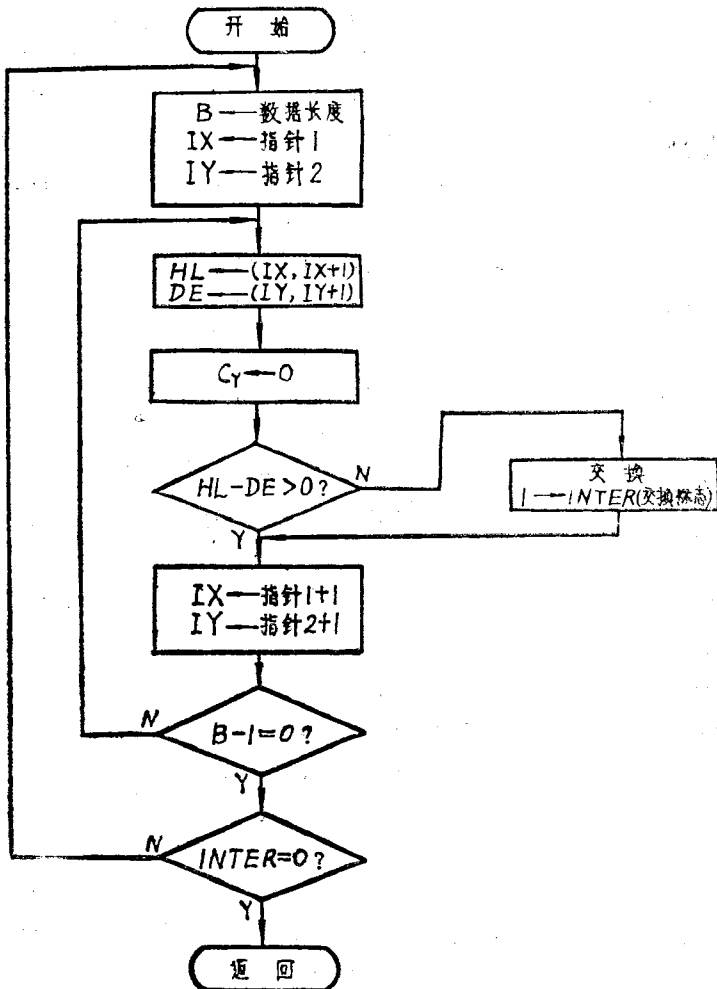


图 5-7'

# 第六章 其它

## 6-1 数据块移动

说明：将内存中某一数据（程序）块移动到用户指定的单元。

程序执行前置入：DE←源始地址

HL←源末地址

BC←目的地址

程序：

2000	C5	MOVE:	PUSH BC	; 保存目标地址
2001	B7		OR A	; Cy清0
2002	ED52		SBC HL, DE	
2004	23		INC HL	
2005	44		LD B, H	
2006	4D		LD C, L	
2007	E1		POP HL	
2008	EB		EX DE, HL	
2009	EDB0		LDIR	; 循环传送
200B	44		LD B, H	
200C	4D		LD C, L	
200D	0D		DEC C	
200E	C34C07		JP DISPY	; 转单板机显示
2011	76		HALT	; 以CMC机为例
	(074C)	DISPY	EQU 074CH	

框图：

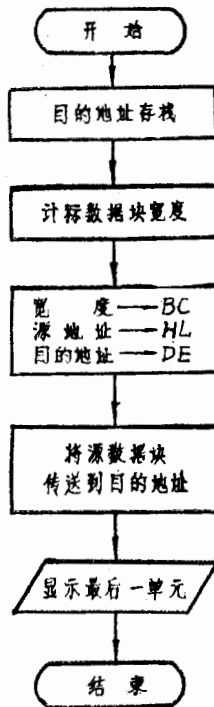


图 6-1

## 6—2 内存数据块校验

说明：1. 对两个数据块进行校验检查，若全相等停在源数据块的末址上，若不相等则显示当前源地址及相应单元内容。

2. 程序执行前置入：  
DE ← 源始地址  
HL ← 源末地址  
BC ← 目的地址

程序：

2000	CD0720	CALL VRFY	
2003	CD1A20	CALL PLAY	
2006	76	HALT	
2007	C5	VRFY: PUSH BC	；保存目标地址
2008	B7	OR A	；Cy清0
2009	ED52	SBC HL, DE	；求数据块长度

200B	23		NC HL	
200C	44		LD B, H	
200D	4D		LD C, L	
200E	E1		POP HL	
200F	EB		EX DE, HL	
2010	1A	LOOP:	LD A, (DE)	; 取相应目的单元内容
2011	EDA1		CPI	; 与源内容比较
2013	C41A20		CALL NZ, PLAY	; 不等转显示处理
2016	13		INC DE	; 相等产生下一条地址
2017	E0		RET PO	; 结束返回
2018	18F6		JR LOOP	
201A	44	PLAY:	LD B, H	
201B	4D		LD C, L	
201C	0D		DEC C	
201D	C37C04		JP DISPY	; 转单板机显示
	(047C)	DISPY	EQU 047CH	

框图:

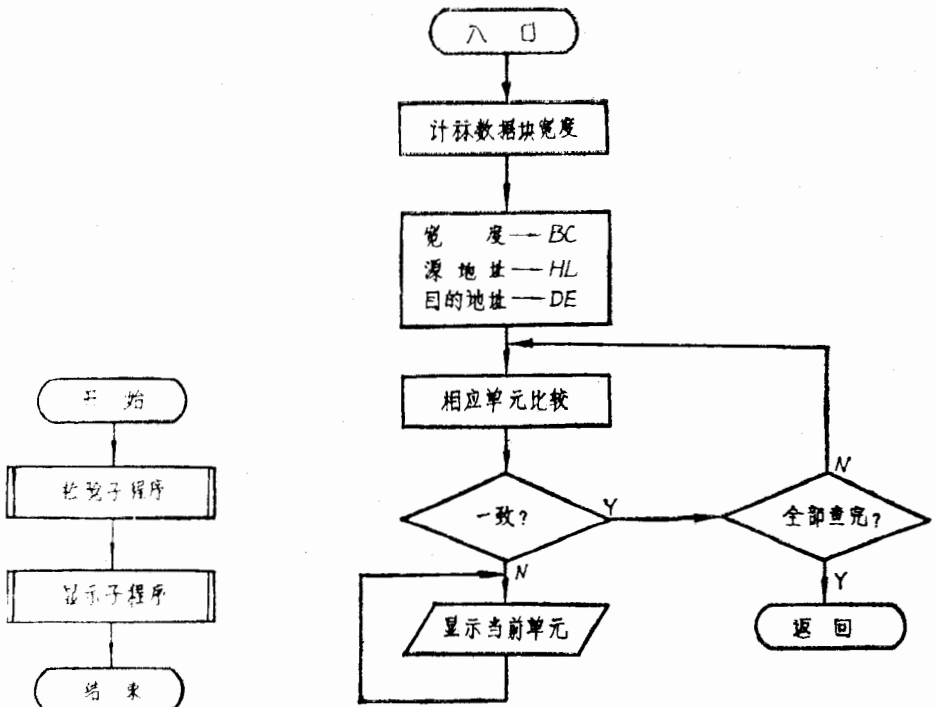


图 6-2-1

图 6-2-2

### 6-3 CTC作为计数器方式编程

说明：CTC通道0作为计数器，并设：

- (1) 每计数60H次发一个中断请求；
- (2) 由CLK/TRG的下降沿触发减1计数器；
- (3) 允许中断。

0通道控制字：

D <sub>7</sub>	D <sub>6</sub>	D <sub>5</sub>	D <sub>4</sub>	D <sub>3</sub>	D <sub>2</sub>	D <sub>1</sub>	D <sub>0</sub>
1	1	0	0	0	1	0	1

- 1 表示控制字
- 0 使之继续运行
- 1 下一个控制字是时间常数
- 0 此位不用
- 0 CLK/TRG的下降沿触发减1计数器
- 0 计数器不用这一位
- 1 计数器方式
- 1 允许中断

程序：

```

00002
2000 3E23      START: LD  A, 23H
2002 ED47          LD  I, A           ; 设定中断矢量高字节
2004 3E00          LD  A, 00H
2006 D38C          OUT (CTC0), A       ; 外设提供中断矢量低
                                     字节
2008 3EC5          LD  A, 0C5H
200A D38C          OUT (CTC0), A       ; 设定 CTC0工作方式
                                     输入控制字
200C 3E60          LD  A, 60H
200E D38C          OUT (CTC0), A       ; 把计数值60H输出到
                                     0通道
2010 ED5E          IM2
2012 FB           LOOP: EI
2013 76           HALT
2014 18FC          JR  LOOP
2016              ORG 2300H
    
```

2200	0024	DW 2400H	; 中断服务程序入口地址表
2302		ORG 2400H	; 中断服务程序入口
2400	ED4D	RETI	; 中断返回
	(008C)	EQU 8CH	; CTC0口地址

框图:

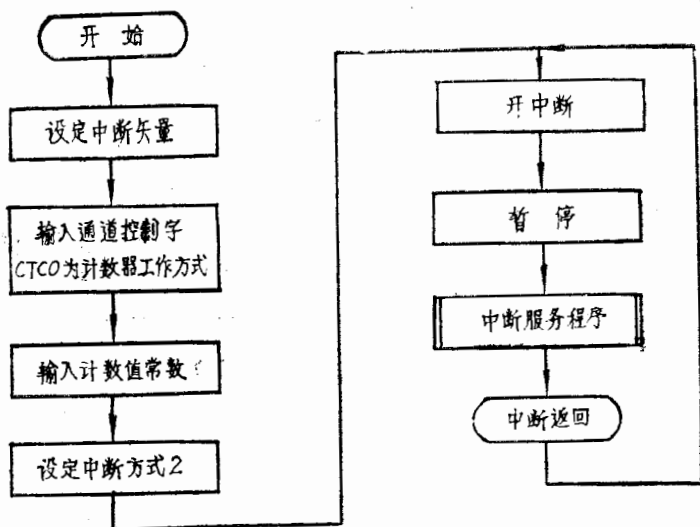


图6-3



## 6-4 CTC作为定时器方式编程

说明：CTC通道0作为定时器，并设：

- (1) 设时钟 $\phi$ 的周期为500毫微秒；
  - (2) 每送16个时钟脉冲( $\phi$ )使减1计数器减1；
  - (3) 使减1计数器回零时间为 $500 \times 16 \times 60\text{H} = 760\mu\text{s}$ ；
  - (4) 定时器由时钟 $\phi$ 启动；
  - (5) 允许中断
- 0通道控制字：

D <sub>7</sub>	D <sub>6</sub>	D <sub>5</sub>	D <sub>4</sub>	D <sub>3</sub>	D <sub>2</sub>	D <sub>1</sub>	D <sub>0</sub>
1	0	0	1	0	1	0	1

- 1 表示控制字
- 0 使之运行
- 1 下一字节为计时值
- 0 由装入时间常数启动定时器工作
- 1 由CLK/TRG的上升沿触发
- 0 每16个时钟脉冲减1
- 0 定时器工作方式
- 1 允许中断

程序：

```

2000 3E23    START: LD  A, 23H
2002 ED47                LD  I, A           ; 设定中断矢量高字节
2004 3E00                LD  A, 00H
2006 D38C                OUT (CTC0), A     ; 外设提供中断矢量低字节
2008 3E95                LD  A, 95H
200A D38C                OUT (CTC0), A     ; 设定CTC0为定时器方式
200C 3E60                LD  A, 60H
200E D38C                OUT (CTC0), A     ; 把定时值60H输出到0通道
2010 ED5E                IM2
2012 FB          LOOP: EI
2013 76                  HALT
2014 18FC                JR  LOOP
2016                    ORG 2300H
2300 0024                DW 2400H           ; 中断服务程序址入口地址表
    
```

2302

ORG 2400H

2400 ED4D

RETI

; 中断返回

(008C) CTC0

EQU 8CH

; CTC0口地址

框图

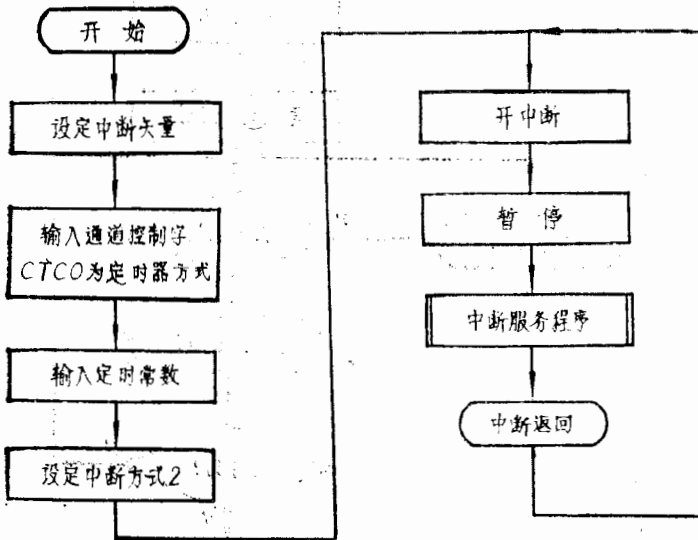


图 6-4

6-6E

## 6-5 1~255MS软件延时

说明：利用寄存器循环计数软件延时，延时范围从1~255ms。事先应把延时数化为16进制数，存放在2020H单元。该程序以子程序方式，保存原BC、DE、HL内容。

程序：

```

2000 D9      MTIME: EXX          ; 保存现场
2001 3A2020 LD A, (ADDRSS)     ; 取延时数
2004 0EF9   DLP0: LD C, 0F9H   ; 1ms时间常数
2006 0D     WFL: DEC C
2007 20FD   JR NZ, WFL
2009 3D     DEC A
200A 20F8   JR NZ, DLP0       ; 延时未到, 则跳转
200C D9     EXX                ; 恢复现场
200D C9     RET
(2020) ADDRSS EQU 2020H      ; 存延时ms数
  
```

框图：

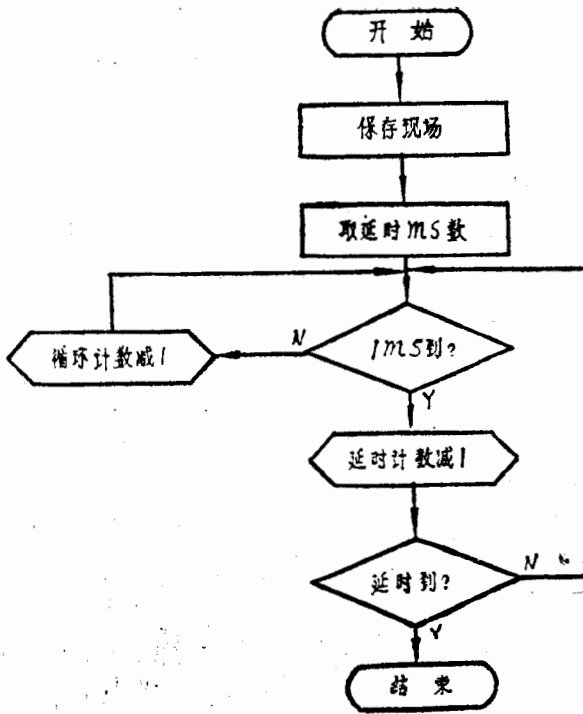


图6-5

### 6-6 1秒~255秒软件延时

说明：利用寄存器进行循环计数执行软件程序延时，延时范围从1~255秒。事先将延时数化为16进制数，存放在 2020H 单元。该程序子程序方式，保存原 BC、DE、HL 内容。

程序：

2000	D9	STIME:	EXX	;	保护现场
2001	3A2020		LD A, (SECOND)	;	取延时数
2004	2604	DST:	LD H, 04H	;	置时间常数
2006	2EFA	LOOPH:	LD L, 0FAH		
2008	0EF9	DLP1:	LD C, 0F9H		
200A	0D	WTL:	DEC C		
200B	20FD		JR NZ, WTL	;	延时计数
200D	2D		DEC L		
200E	20F8		JR NZ, DLP1		
2010	25		DEC H		

2011	20F3	JR NZ, LOOPH	; 1秒到? 未到转
2013	3D	DEC A	
2014	20EE	JR NZ, DST	; 延时到?
2016	C9	RET	; 时间到返主程序
	(2020)	SECOND EQU 2020H	; 存延时秒数

框图:

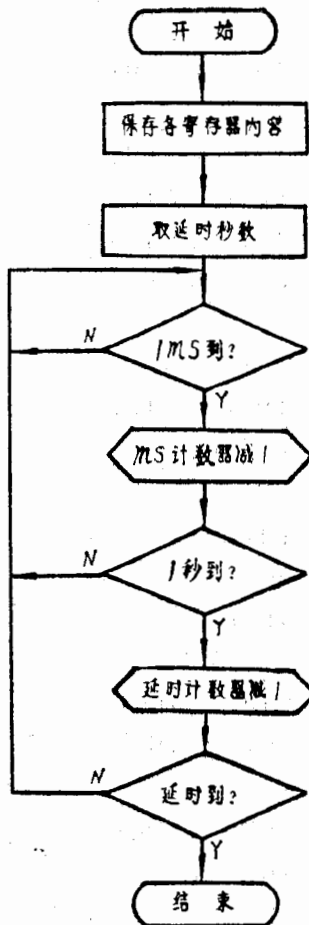


图 6-6

## 6-7 1~255秒CTC中断延时

- 说明：1.在调用软件延时情况下，往往不能满足对实时要求较高的场合，而利用CTC进行实时中断延时，则不会影响主程序的运行。
- 2.延时范围从1~255秒。事先将延时数先化为16进制数，存放在2020H单元。在每次执行前，应按RESET键使CTC接口片复位。

程序：

2000	3E20	LD A, 20H	
2002	ED47	LD I, A	; 设定中断矢量高字节
2004	3E40	LD A, 40H	
2006	D38C	OUT (CTC0), A	; 外设提供中断矢量低字节
2008	3EA5	LD A, 0A5H	
200A	D38C	OUT (CTC0), A	; 输入控制字, 设定CTC0工作方式
200C	3E9C	LD A, 9CH	
200E	D38C	OUT (CTC0), A	; 输入时间常数
2010	0E32	LD C, 032H	; 1秒计时常数
2012	ED5E	IM2	
2014	3A2020	LD A, (2020H)	; 取延时数
2017	FB	MAIN: EI	; 开中断
2018	76	HALT	; 等待中断
2019	18FC	JR MAIN	
201B		ORG 2040H	
2040	5020	DW 2050H	; 中断服务程序入口地址表
2042		ORG 2050H	; 中断服务程序
2050	0D	DEC C	
2051	2005	JR NZ, INTRET	; 未到1秒中断返回
2053	3D	DEC A	
2054	28C1	JR Z, MAIN	; 延时到, 转MAIN
2056	0E32	LD C, 032H	
2058	ED4D	INTRET: RETI	; 中断返回
	(008C)	CTC0 EQU 8CH	

框图:

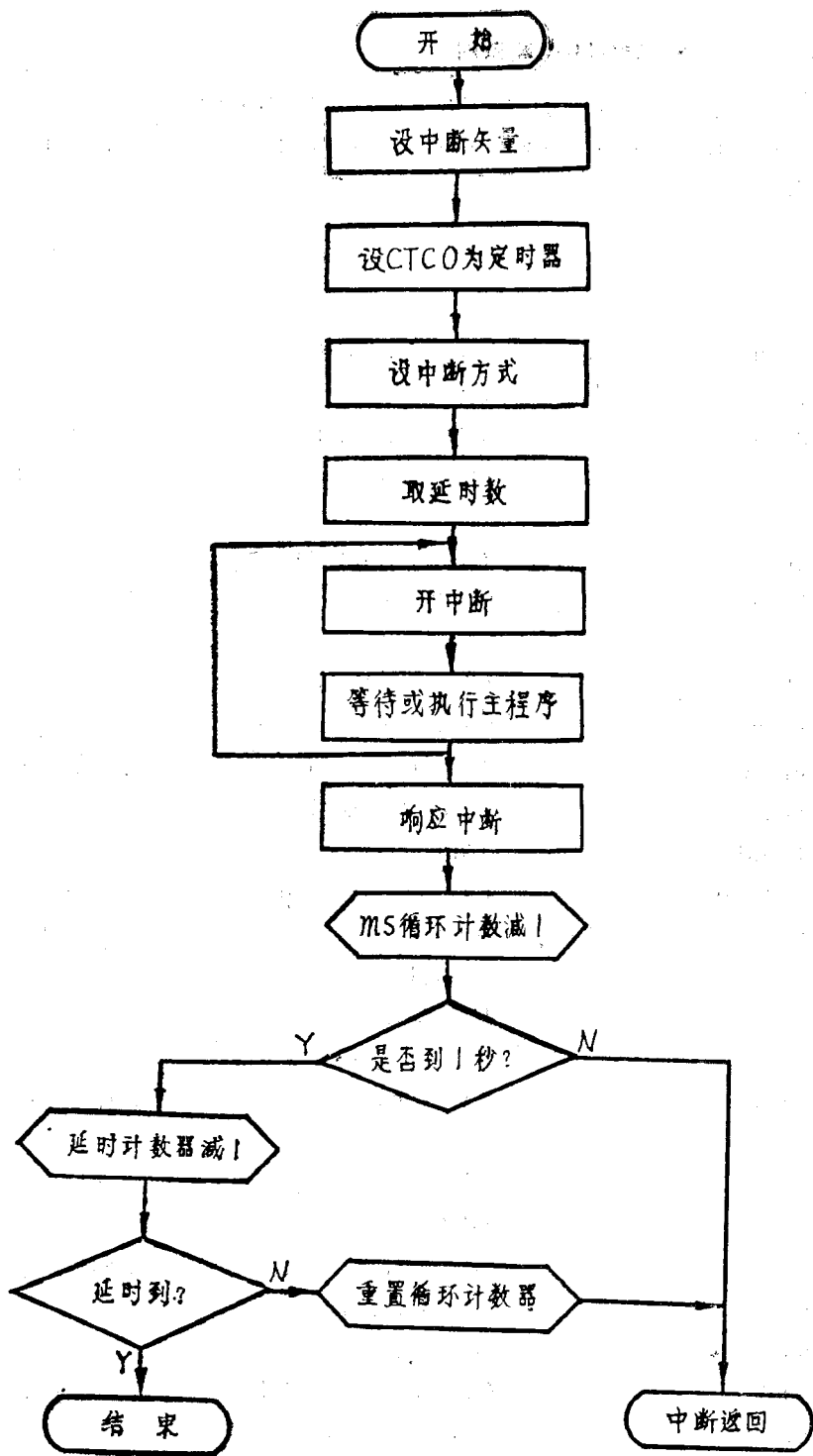


图6-7

## 6-8 1~255分钟CTC中断延时

说明：此延时程序利用CTC通道0进行中断延时，延时范围从1~255分钟。运行前，将延时数转换成16进制数存放在2020H单元。在每次执行前，应按 RESET 键使CTC接口片复位。

程序：

2000	3E20	LD A, 20H	
2002	ED47	LD I, A	
2004	3E40	LD A, 40H	； 设定中断矢量高字节
2006	D38C	OUT (CTC0), A	； 外设提供中断矢量低字节
2008	3EA5	LD A, 0A5H	
200A	D38C	OUT (CTC0), A	； 输入控制字设定 CTC0 工作方式
200C	3E9C	LD A, 9CH	
200E	D38C	OUT (CTC0), A	； 输入时间常数
2010	0E32	LD C, 032H	； 置计数常数
2012	063C	LD B, 03CH	
2014	ED5E	IM2	； 开中断
2016	3A2020	LD A, (2020H)	； 取延时数
2019	FB	MAIN: EI	； 等待中断
201A	76	HALT	
201B	18FC	JR MAIN	
201D		ORG 2040H	
2040	5020	DW 2050H	； 中断服务程序入口地址表
2042		ORG 2050H	； 中断服务程序
2050	0D	DEC C	
2051	200C	JR NZ, INTRET	； 未到1秒返回中断
2053	0E32	LD C, 32H	
2055	05	DEC B	
2056	2007	JR NZ, INTRET	； 未到1分返回中断
2058	0E32	LD C, 32H	
205A	063C	LD B, 03CH	
205C	3D	DEC A	

205D	28BA	JR Z, MAIN	；延时到转MAIN
205F	ED4D	INTRET; RETI	；中断返回
	(008C)	CTC0 EQU 8CH	

框图：

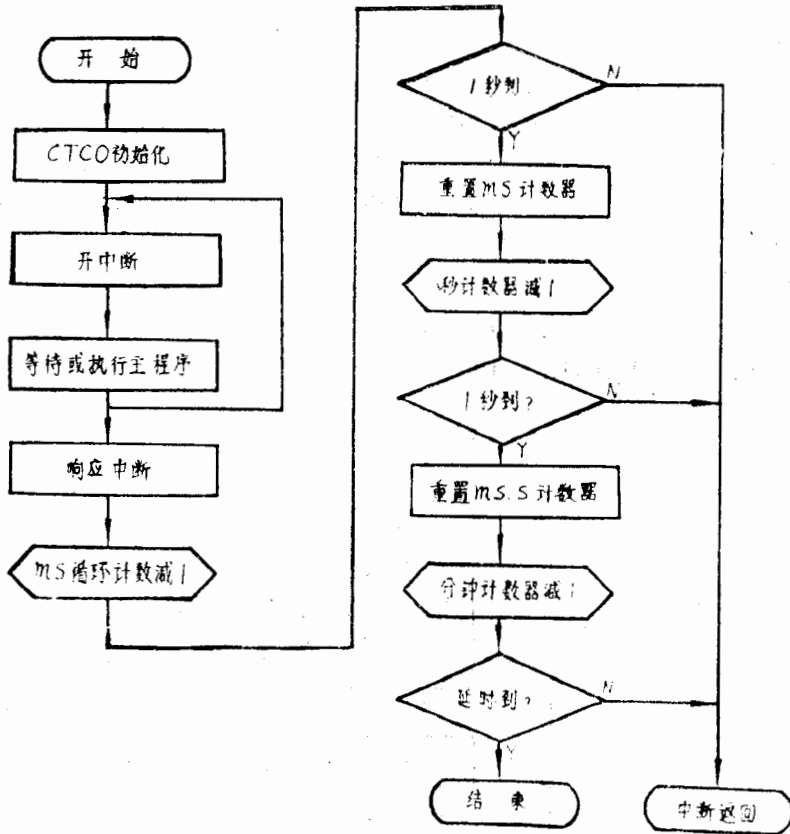


图6-8

### 6-9 数据的校验和 (八位)

- 说明：
1. 这种校验和常用于纸带与盒式磁带系统，以保证正确地读出数据。
  2. 程序要求将要校验的数存放在以 DATA 单元开始的连续存储单元中，而这一串数的长度放在单元 SIZE 中。校验和存入存储单元 DONE 中。
  3. 校验和是靠将这串数全部进行逻辑“异”来产生的。使用寄存器 A、B 和 HL。



程序:

2000	211020	START: LD HL, DATA-1	; 置地址指针
2003	3A1020	LD A, (SIZE)	; 置计数器
2006	47	LD B, A	
2007	97	SUB A	; A清零
2008	23	LOOP: INC HL	
2009	AE	XOR (HL)	; 逻辑“异”
200A	10FC	DJNZ LOOP	; 循环完否?
200C	321420	LD (DONE), A	; 完, 送校验和
200F	C9	RET	
2010	03	SIZE: DB 03H	; 数据个数
2011	285526	DATA: DB 28H, 55H, 26H	; 数据
2014	(0001)	DONE: DS 1	; 校验和

框图:

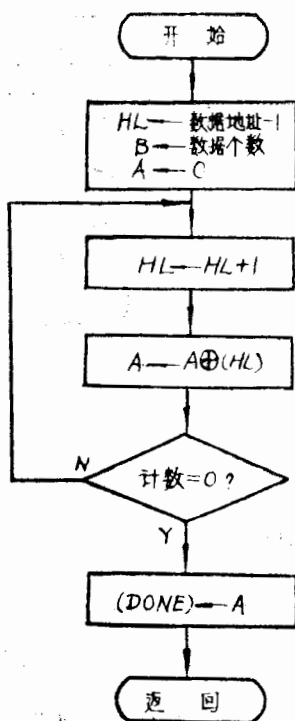


图6-9

## 6—10 数据校验和（十六位）

- 说明：1. 本程序对一组十六位的二进制数进行逻辑“异”，从而产生相应的校验和每个数的高八位放在两相邻字节的高地址单元，低八位放在低地址。
2. 要进行校验和操作的数组存放在以DATA单元开始的连续单元中，数据个数放在SIZE单元中。
3. 程序运行后将校验和的低八位放在 XORL 单元中，高八位放在存储单元XORH 中。
4. 程序使用寄存器A、B、C和HL，但不影响各寄存器的内容。

程序：

```

2000 08      START: EX AF, AF'
2001 D9      EXX
2002 212620 LD HL, DATA-2      ; 置地址指针
2005 0E01    LD C, 01          ; 置低字节运算标志
2007 3A2720 HIGH: LD A, (SIZE) ; 计数
200A 47      LD B, A
200B 97      SUB A
200C 23      LOOP: INC HL
200D 23      INC HL
200E AE      XOR (HL)
200F 10FB    DJNZ LOOP
2011 0D      DEC C
2012 CA1B20  JP Z, JMP          ; 判运算标志
2015 322F20 LD (XORH), A          ; 送校验和的高字节
2018 C32420 JP DONE
201B 322E20 JMP: LD (XORL), A        ; 送校验和的低字节
201E 212720 LD HL, DATA-1
2021 C30720 JP HIGH          ; 转高字节运算
2024 08      DONE: EX AF, AF'
2025 D9      EXX
2026 C9      RET
2027 (0001) SIZE: DS 1        ; 数据个数
2028 (0006) DATA: DS 06     ; 数据
202E (0001) XORL: DS 1      ; 校验和的低字节
202F (0001) XORH: DS 1      ; 校验和的高字节

```

框图:

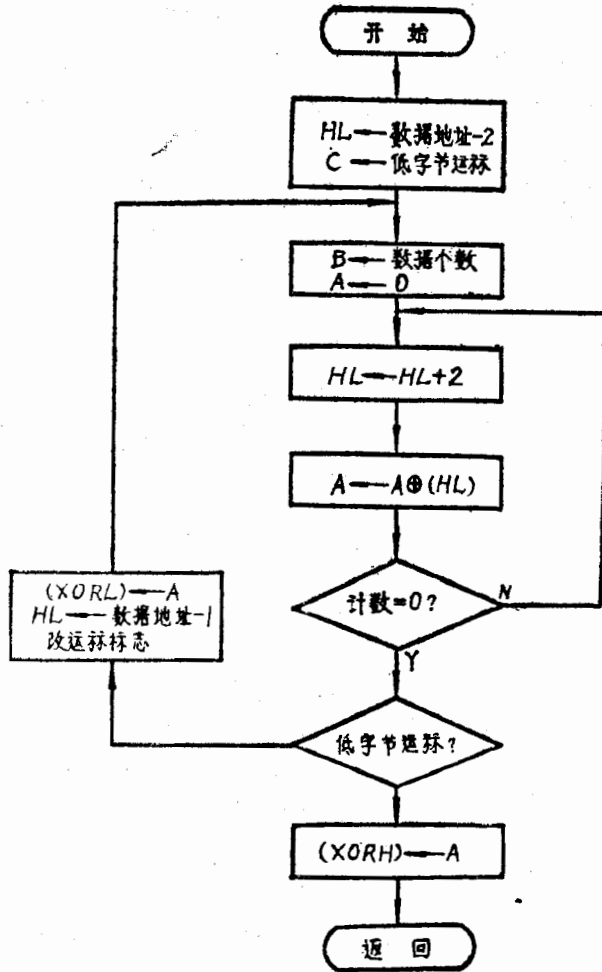


图 6-10

### 6-11 条件选择转移程序 (一)

说明: 1. 功能号存放在A累加器中, 本程序中功能从0~7。  
2. R0~R7分别对应着0~7号功能的程序入口。

程序:

```
2000 210D20 START: LD HL, FTAB
2003 87 ADD A, A
2004 5F LD E, A
```

2005	1600		LD R0, 0	
2007	19		ADD HL, DE	
2008	5E		LD E, (HL)	
2009	23		INC HL	
200A	56		LD D, (HL)	
200B	EB		EX DE, HL	
200C	E9		JP (HL)	
200D	0000	FTAB:	DW R0	; 八部功能的程序入口
			DW R1	
			DW R2	
			DW R3	
			DW R4	
			DW R5	
			DW R6	
			DW R7	

#### 6-12 RAM唯一地址检查程序

- 说明: 1. 将被检查内存区的首地址存到STAADD和STAADD + 1单元中去; 末地址存到ENDADD和ENDADD + 1单元中去。
2. 方法: 将被检查的地址的低八位存到该单元中去, 然后依次检查内容是否与地址一致, 不一致说明CPU只与某几个单元打交道, 这时显示出错的首地址, 否则显示监控程序提示符。

程序:

2000	2AC23F	START:	LD HL, (ENDADD)
2003	ED5BC03F		LD DE, (STAADD)
2007	ED52		SBC HL, DE
2009	D5		PUSH DE
200A	E5		PUSH HL
200B	C1		POP BC
200C	E1		POP HL
200D	E5		PUSH HL
200E	C5		PUSH BC
200F	75	LOOP:	LD (HL), L
2010	EDA1		CPI
2012	EA0F20		JP PE, LOOP
2015	C1		POP BC

2016	E1		POP HL
2017	7D	LOOP1:	LD A, L
2018	EDA1		CPI
201A	2004		JR NZ, LOOP2
201C	EA1720		JP PE, LOOP1
201F	C7		RST 0
2020	2B	LOOP2:	DEC HL
2021	7C		LD A, H
2022	DD21F73F		LD IX, DSMEM
2026	CD3C20		CALL UFOR1
2029	7D		LD A, L
202A	DD21F93F		LD IX, DSMEM2
002E	CD3C20		CALL UFOR1
2031	7E		LD A, (HL)
2032	DD21FB3F		LD IX, DSMEM4
2036	CD3C20		CALL UFOR1
2039	C38100		JP DISUP
203C	47	UFOR1:	LD B, A
203D	E60F		AND 0FH
203F	DD7701		LD (IX+1), A
2042	78		LD A, B
2043	CB3F		SRL A
2045	CB3F		SRL A
2047	CB3F		SRL A
2049	CB3F		SRL A
204B	DD7700		LD (IX+0), A
204E	C9		RET
	(0081)	DISUP:	EQU 0081H
	(3FF7)	DSMEM:	EQU 3FF7H
	(3FF9)	DSMEM2:	EQU 3FF9H
	(3FFB)	DSMEM4:	EQU 3FFBH
	(3FC0)	STAADD:	EQU 3FC0H
	(3FC2)	ENDADD:	EQU 3FC2H

框图:

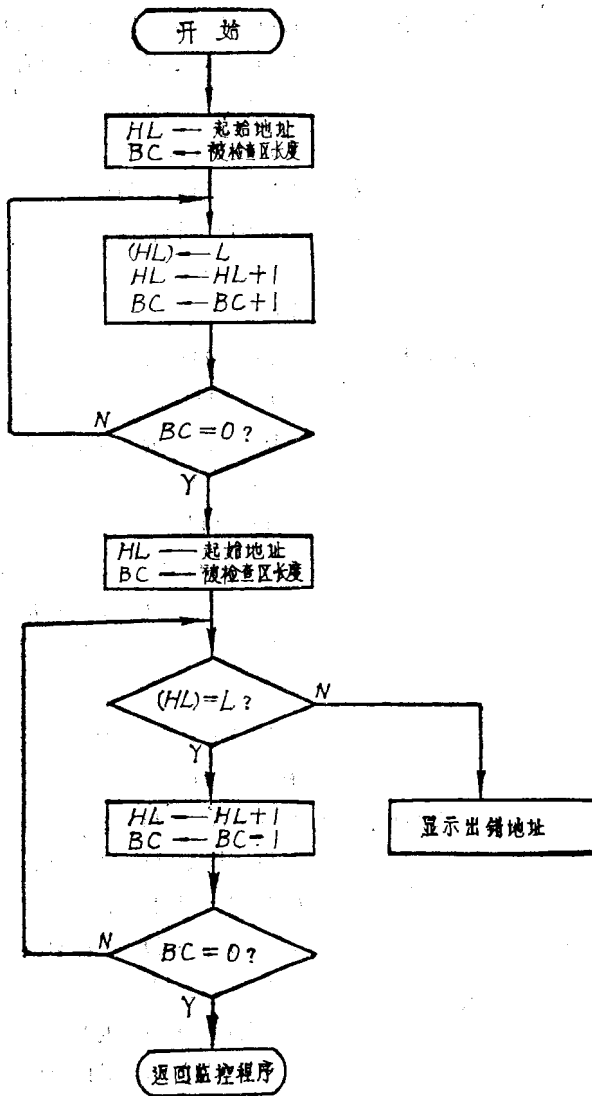


图 6-12

### 6-13 RAM存储器读写检查程序(一)

说明: 1. 被检查的RAM的首址存到STAADD和STAADD + 1 单元中, 末址存到ENDADD和ENDADD + 1 单元中。

2. 方法: 将检测字55H和AAH写入RAM的单元中, 然后读出与检测字比较, 如果正确则继续检查, 否则显示出错地址。如果全部检查完毕未发现错误, 则显示出错程序提示符。

程序:

2000	2AC23F	START:	LD HL, (ENDADD)
2003	ED5BC03F		LD DE, (STAADD)
2007	D5		PUSH DE
2008	3E55		LD A, 55H
200A	ED52		SBC HL, DE
200C	E5		PUSH HL
200D	C1		POP BC
200E	E1		POP HL
200F	77	LOOP:	LD (HL), A
2010	BE		CP (HL)
2011	200C		JR NZ, RAMERR
2013	2F		CPL
2014	77		LD (HL), A
2015	BE		CP (HL)
2016	2007		JR NZ, RAMERR
2018	2F		CPL
2019	EDA1		CPL
201B	EA0F20		JP PE, LOOP
201E	C7		RST 0
201F	7E	RAMERR:	LD A, (HL)
2020	DD21FB3F		LD IX, DSMEM4
2024	CD3A20		CALL UFOR1
2027	DD21F73F		LD IX, DSMEM0
202B	7C		LD A, H
202C	CD3A20		CALL UFOR1
202F	DD21F93F		LD IX, DSMEM2
2033	7D		LD A, L
2034	CD3A20		CALL UFOR1
2037	C38100		JP DISUP
203A	47	UFOR1:	LD B, A
203B	E60F		AND 0FH
203D	DD7701		LD (IX+1), A
2040	78		LD A, B
2041	CB3F		SRL A
2043	CB3F		SRL A
2045	CB3F		SRL A

2047	CB3F		SRL	A
2049	DD7700		LD	(IX+0), A
204C	C9		RET	
(3FF7)	DSMEM0:	EQU	3FF7H	
(3FF9)	DSMEM2:	EQU	3FF9H	
(3FFB)	DSMEM4:	EQU	3FFBH	
(0081)	DISUP:	EQU	0081H	
(3FC0)	STAADD:	EQU	3FC0H	
(3FC2)	ENDADD:	EQU	3FC2H	

框图:

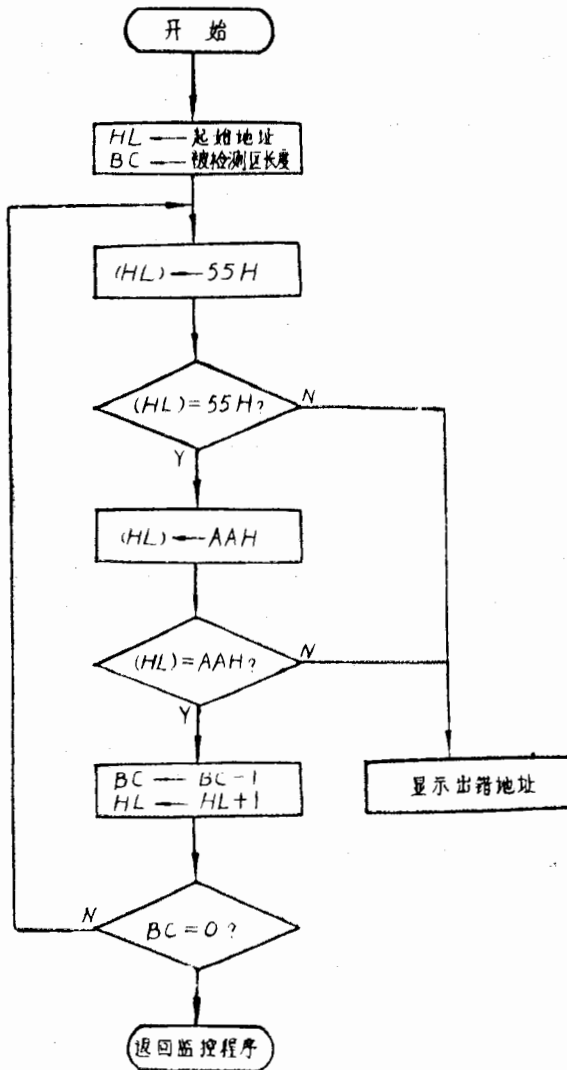


图 6-13



## 6—14 RAM存储器读写检查程序(二)

- 说明: 1. 被检查的RAM的首址存到STAADD和STAADD+1单元中; 末址存放到ENDADD和ENDADD+1单元中。
2. 方法: 依次用256个检测字(00H~FFH)对检测区的存储单元进行256遍扫描。发现错误则转去显示出错地址, 否则返回监控程序。

程序:

```

2000 2AC23F      START:  LD HL, (ENDADD)
2003 ED5BC03F   LD DE, (STAADD)
2007 D5         PUSH DE
2008 AF        XOR A
2009 ED52      SBC HL, DE
200B E5        PUSH HL
200C C1        LOOP1:  POP BC
200D E1        POP HL
200E E5        PUSH HL
200F C5        PUSH BC
2010 77        LOOP2:  LD (HL), A
2011 BE        CP (HL)
2012 2009      JR NZ, RAMERR
2014 EDA1      CPI
2016 EA1020    JP PE, LOOP2
2019 3C        INC A
201A 20F0      JR NZ, LOOP1
201C C7        RST 0
201D 7E        RAMERR: LD A, (HL)
201E DD21FB3F  LD IX, DSMEM4
2022 CD3820    CALL UFOR1
2025 DD21F73F  LD IX, DSMEM0
2029 7C        LD A, H
202A CD3820    CALL UFOR1

```

202D	DD21F93F		LD IX, DSMEM2
2031	7D		LD A, L
2032	CD3820		CALL UFOR1
2035	C3F400		JP DISUP
2038	47	UFOR1:	LD B, A
2039	E60F		AND 0FH
203B	DD7701		LD (IX+1), A
203E	78		LD A, B
203F	CB3F		SRL A
2041	CB3F		SRL A
2043	CB3F		SRL A
2045	CB3F		SRL A
2047	DD7700		LD (IX+0), A
204A	C9		RET
	(3FF7)	DSMEM0:	EQU 3FF7H
	(3FF9)	DSMEM2:	EQU 3FF9H
	(3FFB)	DSMEM4:	EQU 3FFBH
	(00F4)	DISUP:	EQU 00F4H
	(3FC0)	STAADD:	EQU 3FC0H
	(3FC2)	ENDADD:	EQU 3FC2H

框图:

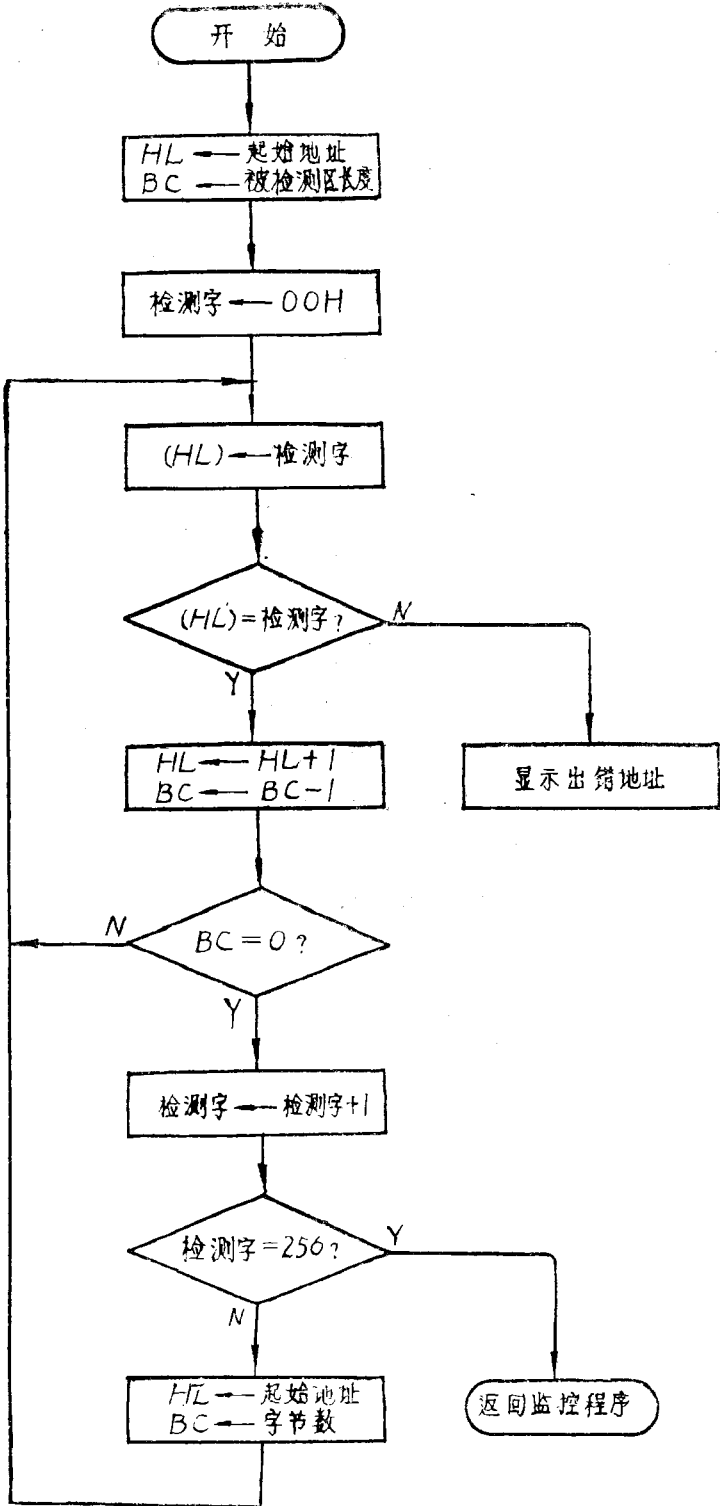


图 6-14

## 6—15 显示子程序

说明：1. DISMEM单元到DISMEM+5单元为显示缓冲区，SEGPT单元到SEGPT+18单元为显示字形表。

2. 方法：根据显示缓冲区的内容到字形表取出字形并送到LED显示器上去，显示1ms后返回主程序。

程序：

```

2000 213B20  DISUP:  LD HL, DISMEM
2003 0620      LD B, 20H
2005 5E      DISUP1: LD E, (HL)
2006 AF      XOR A
2007 57      LD D, A
2008 D394      OUT (DIGLH), A
200A DD212820 LD IX, SEGPT
200E DD19      ADD IX, DE
2010 DD7E00    LD A, (IX)
2013 D390      OUT (SEGLH), A
2015 78      LD A, B
2016 D394      OUT (DIGLH), A
2018 1E2D      LD E, 2DH
201A 1D      DISUP2: DEC E
201B AF      XOR A
201C BB      CP E
201D 20FB      JR NZ, DISUP2
201F 23      INC HL
2020 CB38      SRL B
2022 30E1      JR NC, DISUP1
2024 AF      XOR A
2025 D394      OUT (DIGLH), A
2027 C9      RET
2028 40792430 SEGPT:  DB 40H, 79H, 24H, 30H
202C 19120278  DB 19H, 12H, 02H, 78H
2030 00180803  DB 00H, 18H, 08H, 03H
2034 4621060E  DB 46H, 21H, 06H, 0EH
2038 7F3F5F    DB 7FH, 3FH, 5FH
    
```

203B (0006)

DISMEM: DS 6

(0094)

DIGLH: EQU 94H

(0090)

SEGLH: EQU 90H

框图:

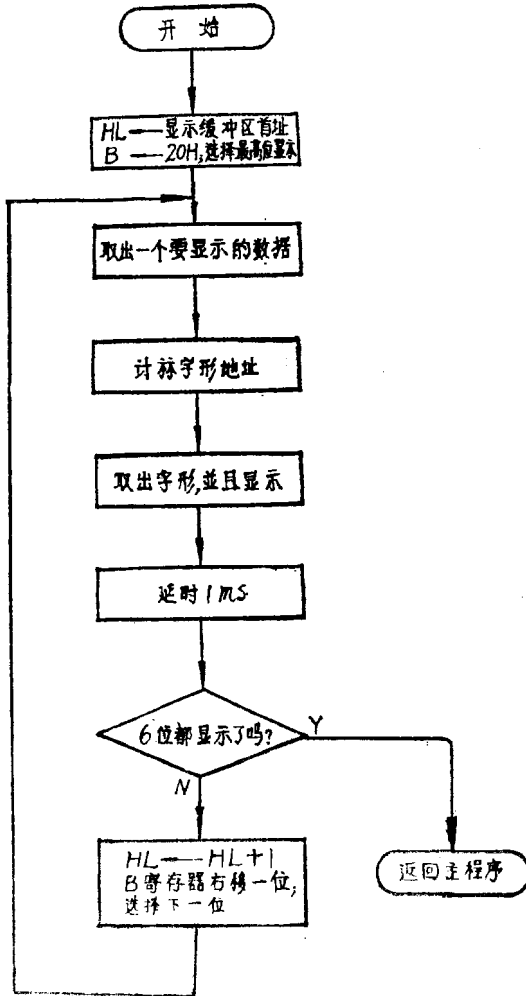


图 6-15

## 6-16 七段显示器计数程序

- 说明: 1. DSMEM到DSMEM + 5 单元为显示缓冲区, CMEM到CMEM + 2 单元为计数单元, 计数范围从 000000~999999, UFOR子程序的功能是将 A 累加器内容写到显示缓冲区。
2. 将CTC0通道设为定时方式, 然后运行显示程序, CTC每次中断都将计数单元的内容加 1, 然后再转去显示。

程序:

```

2000  3E2F      START;  LD  A, 2FH
2002  ED47                LD  I, A
2004  3E00                LD  A, 00H
2006  D38C                OUT (8CH), A
2008  3EA5                LD  A, 0A5H
200A  D38C                OUT (8CH), A
200C  3EF7                LD  A, 0FH
200E  D38C                OUT (8CH), A
2010  ED5E                IM2
2012  FB                  EI
2013  217F20      DIS;   LD  HL, CMEM
2016  7E                  LD  A, (HL)
2017  DD218620     LD  IX, DSMEM4
201B  CD5920       CALL UFOR
201E  23           INC  HL
201F  7E           LD  A, (HL)
2020  DD218420     LD  IX, DSMEM2
2024  CD5920       CALL UFOR
2027  23           INC  HL
2028  7E           LD  A, (HL)
2029  DD218220     LD  IX, DSMEM
202D  CD5920       CALL UFOR

```

2030	218220	DISUP:	LD HL, DSMEM
2033	0620		LD B, 20H
2035	5E	DISUP1:	LD E, (HL)
2036	AF		XOR A
2037	57		LD D, A
2038	D394		OUT (DIGLH), A
203A	DD216C20		LD IX, SEGPT
203E	DD19		ADD IX, DE
2040	DD7E00		LD A, (IX)
2043	D390		OUT (SEGLH), A
2045	78		LD A, B
2046	D394		OUT (DIGLH), A
2048	1E2D		LD E, 2DH
204A	1D	DISUP2:	DEC E
204B	AF		XOR A
204C	BB		CP E
204D	20FB		JR NZ, DISUP2
204F	23		INC HL
2050	CB38		SRL B
2052	30E1		JR NC, DISUP1
2054	AF		XOR A
2055	D394		OUT (DIGLH), A
2057	18BA		JR DIS
2059	47	UFOR:	LD B, A
205A	E60F		AND 0FH
205C	DD7701		LD (IX+1), A
205F	78		LD A, B
2060	CB3F		SRL A
2062	CB3F		SRL A
2064	CB3F		SRL A
2066	CB3F		SRL A
2068	DD7700		LD (IX+0), A

206B	C9		RET
206C	40792430	SEGPT:	DB 40H, 79H, 24H, 30H
2070	19120278		DB 19H, 12H, 02H, 78H
2074	00180803		DB 00H, 18H, 08H, 03H
2078	4621060E		DB 46H, 21H, 06H, 0EH
207C	7F3F5F		DB 7FH, 3FH, 5FH
207F	(0003)	CMEM:	DS 3
2082	(0002)	DSMEM:	DS 2
2084	(0002)	DSMEM2:	DS 2
2086	(0002)	DSMEM4:	DS 2
2088			ORG 2200H
2200	FB	ROUTIN:	EI
2201	08		EX AF, AF'
2202	D9		EXX
2203	217F20		LD HL, CMEM
2206	0603		LD B, 03H
2208	3E01		LD A, 01H
220A	A7		AND A
220B	8E	LOOP3:	ADC A, (HL)
220C	27		DAA
220D	77		LD (HL), A
220E	23		INC HL
220F	3E00		LD A, 00H
2211	10F8		DJNZ LOOP3
2213	08		EX AF, AF'
2214	D9		EXX
2215	ED4D		RETI
	(0094)	DIGLH:	EQU 94H
	(0090)	SEGLH:	EQU 90H



框图:

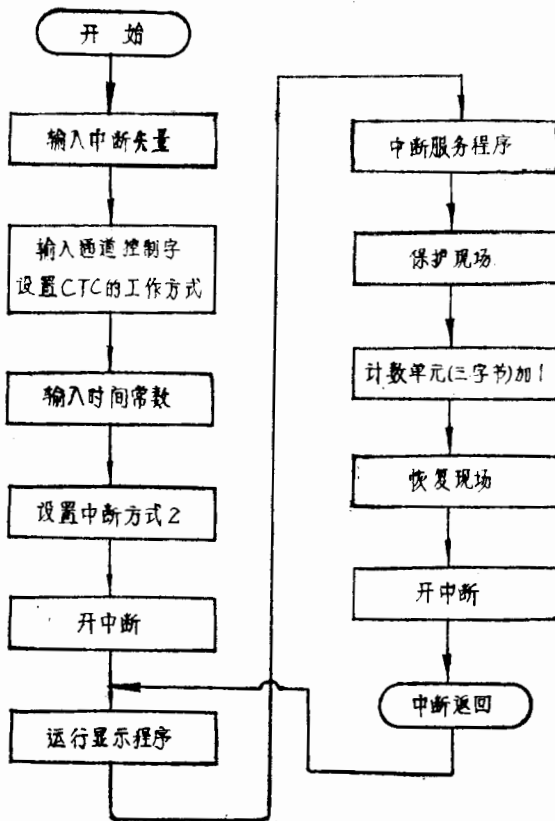


图6-16

## 6-17 唱歌程序

- 说明: 1. 将一歌曲的简谱存到起始地址为2050H的存贮区, 半拍打一个数字, 一拍重复打两个数字, 歌曲结束打入00H。低音的高位用0填充。高音的高位用1填充如: 1、2、3、4、5、6、7、1̇、2̇、3̇、4̇、5̇、6̇、7̇分别用数字01、02、03、04、05、06、07、11、12、13、14、15、16、17表示。程序执行后便可在CTC 1通道(录音机转贮端口)发出音频信号, 通过耳塞机或外接扬声器可以听到歌曲。
2. 改变2006H单元的内容便可改变歌曲的速度。

程序:

2000	215020	START:	LD HL, 2050H
2003	1600		LD D, 00H
2005	0610	LOOP:	LD B, 10H
2007	3E15		LD A, 15H
2009	D385		OUT (85H), A
200B	7E		LD A, (HL)
200C	B7		OR A
200D	2824		JR Z, HHHH
200F	4F		LD C, A
2010	E610		AND 10H
2012	2806		JR Z, HS
2014	DD213F20		LD IX, TST2
2018	1804		JR SS
201A	DD213820	HS:	LD IX, TST1
201E	79	SS:	LD A, C
201F	E60F		AND 0FH
2021	5F		LD E, A
2022	DD19		ADD IX, DE
2024	DD7EFF		LD A, (IX-1)
2027	D385		OUT (85H), A
2029	E5		PUSH HL
202A	CD3406	MUP:	CALL D20MS
202D	10FB		DJNZ MUP
202F	E1		POP HL
2030	23		INC HL
2031	18D2		JR LOOP
2033	3E03	HHHH:	LD A, 03H
2035	D385		OUT (85H), A
2037	76		HALT
2038	F4D8C0B4	TST1:	DB 0F4H, 0D8H, 0C0H, 0B4H
203C	A49383		DB 0A4H, 93H, 83H
203F	7B6C605A	TST2:	DB 7BH, 6CH, 60H, 5AH
2043	50463E		DB 50H, 46H, 3EH
	(0634)	D20MS:	EQU 0634H

图框:

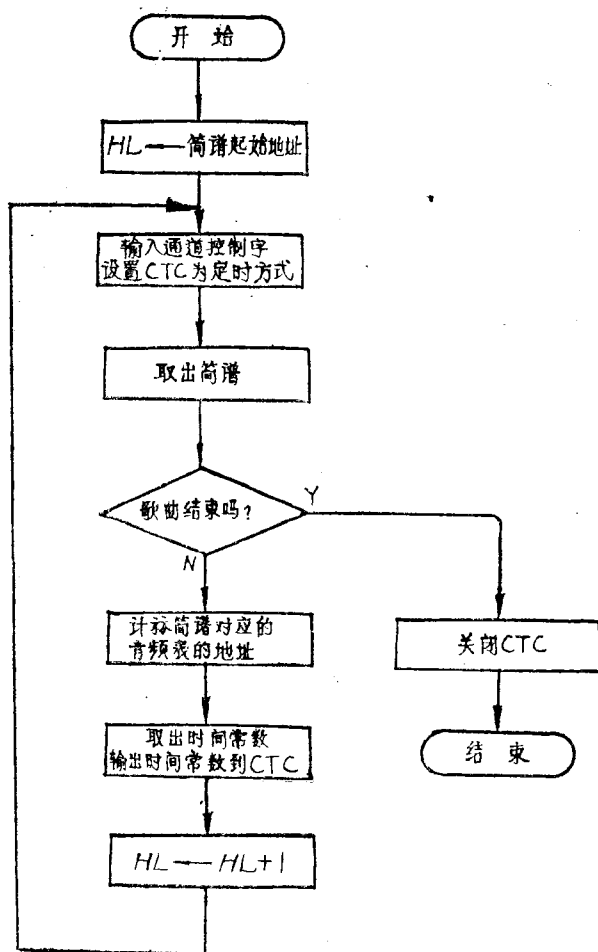


图6-17

## 6-18 PIO方式3初始化程序

说明：将中断服务程序的入口地址存到 2FC0H 和 2FC1H 单元中。执行此程序后便可对 PIOA 端口的第 0 位输入输出线 (PA<sub>0</sub>) 进行监视，当该线为高电平时，发出中断请求信号，转向中断服务程序。服务程序可根据用户的不同的要求编写。

程序：

2000	3E2F	START: LD A, 2FH	
2002	ED47	LD I, A	(形成中断向量地址)
2004	3EC0	LD A, 0C0H	
2006	D382	OUT (PIOADD), A	使 PA0 向 PPI
2008	3ECF	LD A, 0CFH	设置 PPI 寄存器
200A	D382	OUT (PIOADD), A	通知 PPI 开始操作
200C	3EFE	LD A, 0FEH	如选择 PPI 则 LDA
200E	D382	OUT (PIOADD), A	
2010	3EB7	LD A, 0B7H	
2012	D382	OUT (PIOADD), A	
2014	3E01	LD A, 01H	
2016	D382	OUT (PIOADD), A	
2018	ED5E	IM2	中断方式 =
201A	FB	EI	允许中断
201B	76	LOOP: HALT	
201C	18FB	JP LOOP	
	(0082)	PIOADD: EQU 82H	

## 6-19 条件选择转移程序 (二)

说明：1. 对于多个分支情况下，常采用转移表进行条件选择转移。将各例行程序的首地址以表的形式存放在存储器中，并采用间接寻址方式。

2. 假定有八个例行程序，它们的首址分别是 TAB、TAB + 2、TAB + 4……TAB + 14。在累加器中首先存放例行程序的标志位，根据哪一位是 1 决定转去执行那个程序。

例如：累加器为 00000001，程序转 TAB。

累加器为 00000010，程序转 TAB + 2。

累加器为 00000100，程序转 TAB + 4 等等。

程序:

2000	211120	LD HL,TAB	; 转移表起始地址→HL
2003	1F	GTBIT: RRA	; A的最低位移入进位位
2004	DA0C20	JP C,GETAD	; 有进位转GETAD
2007	23	INC HL	; 无进位, 转移表地址加 2
2008	23	INC HL	
2009	C30320	JP GTBIT	
200C	5E	GETAD: LD E,(HL)	; 转移地址送DE
200D	23	INC HL	
200E	56	LD D,(HL)	
200F	EB	EX DE,HL	
2010	E9	JP (HL)	; 按转移表地址转到该例行程序
2011	(0010)	TAB: DS 10H	; 保留16字节存放 8 个例行程序首地址

框图:

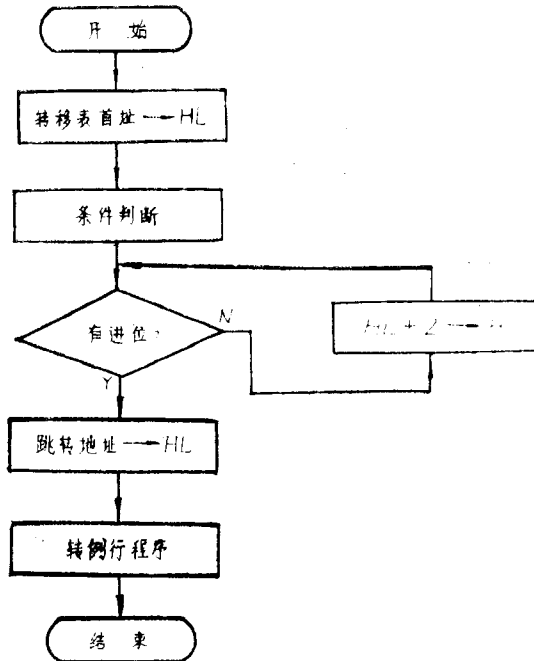


图 6-19

## 6—20 A/D转换程序

- 说明：1.在ADC 0809的输入通道0输入一个0—5V的模拟电压。  
 2.连续采样16次。  
 3.转换后的数字量存3000H开始的内存区。

程序：

2000	1610	LD D, 10H	； 转换次数
2002	210030	LD HL, 3000H	； 存贮区首址
2005	E7	RST 20H	
2006	76	HALT	
3FCE	C32020	JP 2020H	； 转中断服务程序入口
2020	FB	EI	； 中断服务程序开始
2021	D398	OUT (98H), A	； 选0通道
2023	1E18	LD E, 18H	； 延时
2025	00	NOP	
2026	1D	DEC E	
2027	C22520	JP NZ, 2025H	
202A	DB98	IN A, (98H)	； 读入
202C	77	LD (HL), A	
202D	23	INC HL	
202E	15	DEC D	
202F	C22020	JP NZ, 2020H	
2032	C9	RET	

框图:

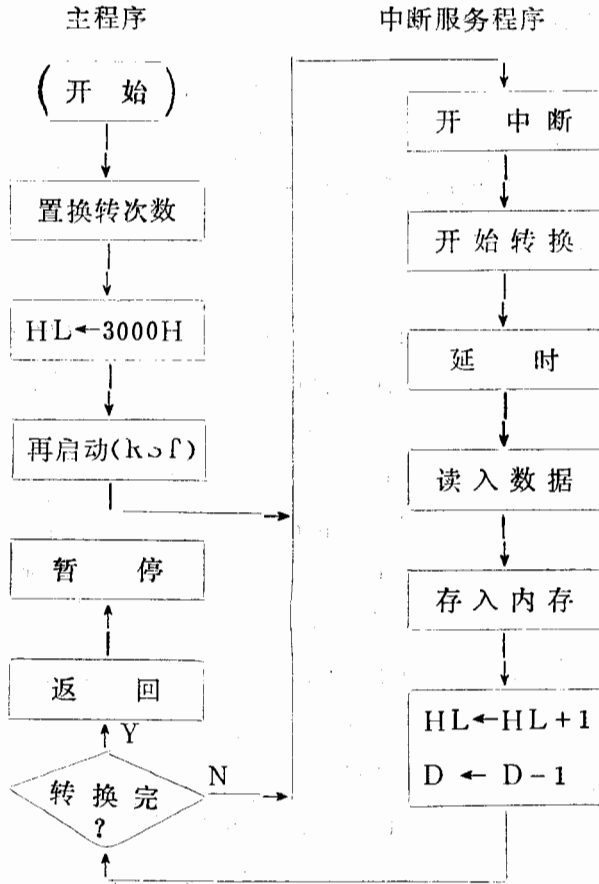


图 6—20

### 6—2 1 一组无符号的十进制数→二进制数

说明: 1.十进制数是以ASCII码的形式存放在内存中。

2.十进制数的首地址放在BC中。

3.转换后的二进制数在HL中。

程序:

2000	21000	ADEC:	LD HL, 0	
2003	0A	ADEC1:	LD A, (BC)	
2004	D630		SUB 30H	
2006	F8		RET M	; A < 0, 返回
2007	FE0A		CP 0AH	
2009	D0		RET C	; A > 9, 返回
			NC	

200A	54	LD D, H	
200B	5D	LD E, L	
200C	29	ADD HL, HL	; HL ← HL × 10
200D	29	ADD HL, HL	
200E	19	ADD HL, DE	
200F	29	ADD HL, HL	
2010	5F	LD E, A	
2011	1600	LD D, 0	
2013	19	ADD HL, DE	
2014	03	INC BC	
2015	C30320	JP ADEC1	

框图:

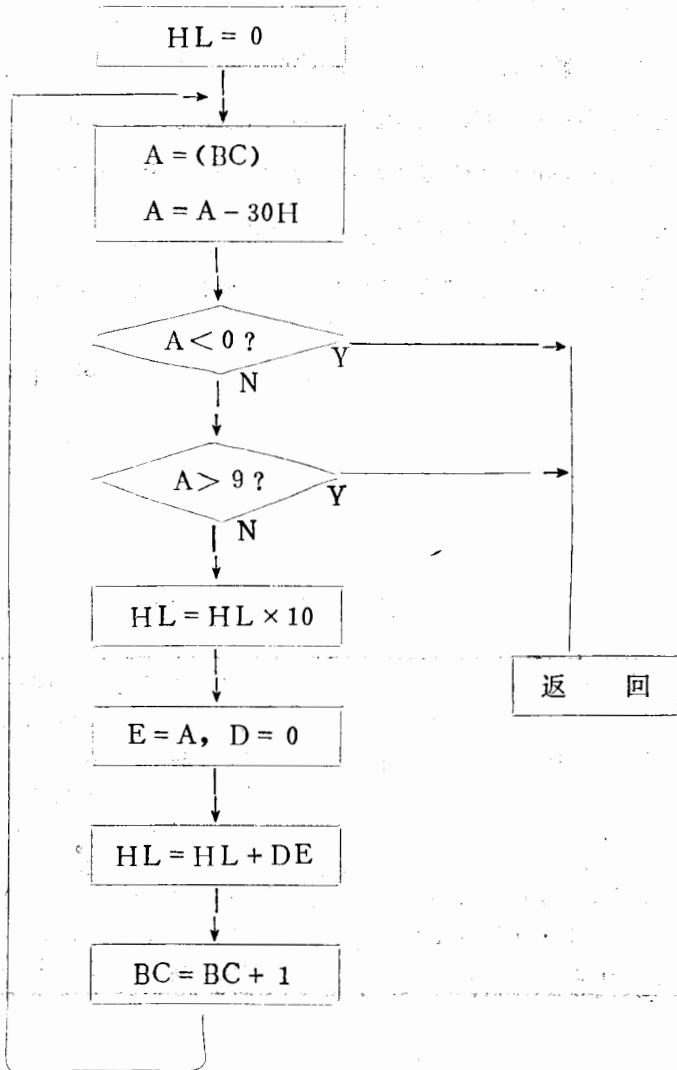


图 6-21



# 編 后

近几年来，国内微型电脑的应用有了很大的发展，特别是以Z—80为中央处理器的各类单板微型计算机，由于体积小，功耗低，可靠性高和价格低等显著的特点，深受广大用户的欢迎。

为了满足CMC—80和其它Z—80类型各单板微型计算机用户的需要，我们收集和研制了各种典型的实用子程序103个，编辑成“CMC—80微型电脑实用子程序手册”作为内部资料出版，供广大用户在实际工作中选用。

“手册”中所收集的程序，除作了扼要的说明之外，都附有详细的框图，结构清楚，阅读方便。全部程序都在CMC—80微型电脑上调试通过，因此内容正确可靠。

本“手册”对Z—80类型单板计算机用户有一定的参考价值。亦可供高等、中等学校上微型机课的师生、有关短训班的师生阅读参考。

由于水平有限，书中内容难免有不当之处，敬请读者批评指正。

编 者

一九八三年六月

## 勘 误 表

页数	行数	错误	正确	页数	行数	错误	正确
16	11	—A	1 A	59	7	CDDF21	CDC621
21	倒3	DE, HC	DE, HL	74	倒1	09	C 9
34	倒2	BE	EB	76	2	3A8322	3A8922
36	14	LD, B	LD B	84	倒8	118522	118B22
40	图2-14	SVB←BC	SUB←BC	93	11	2 <sup>2</sup> 25	2352
41	图3-2	HL-1→HL DE-1→DE	HL+1→HL DE+1→DE	118	图5-3	(SERACH)	(SEARCH)
59	6	219E22	218822	130	18	0000'	
				162	倒1	D8 REC C	DO REC NC