

单片机实用子程序库

马共立 彭喜源 张毅刚 编

哈尔滨工业大学出版社

前 言

在单片机的应用产品开发中，主要有两个难点，一是接口电路的设计，二是程序设计。由于单片机有一批可编程的大规模电路接口芯片可供选用，因而使接口电路的设计得以简化。但由于程序设计涉及面很广，处理的内容繁杂多样，目前还无统一的工程设计方法。怎样才能较快地完成自己的单片机应用程序的设计，并使其更加高效可靠？除了尽快提高自己的程序设计水平以外，还有一个非常好的方法，那就是借用经检验过的程序，根据所要完成的程序设计任务，尽可能寻找合适的现成的程序模块（有的可能要作局部修改），实在找不到的才自己编写，然后将这些程序块有机地组合起来，即可得到所需要的程序。这就是目前常采用的“组装程序”的设计方法。采用这种方法不仅能较快地完成程序设计任务，而且所编制的应用程序也更高效可靠，显然尽量利用现有的经过检验的程序模块，是迅速可靠的完成程序设计任务的一条捷径。

为了帮助从事单片机应用设计的科技人员尽快地实现上述目的，我们在搜集国内外大量资料的基础上，结合自己的科研及教学实践，考虑了选材的典型性、先进性、实用性，精选出了近100个单片机应用的实用程序，编写了这本书。和其它单片机资料相比，本书有几个重要特点：一是本书的选材直接来自单片机测量控制的应用实践。它们描述了一些典型环境的程序实现，具有很强的实用性和通用性，读者可以象利用具有一定功能的集成电路芯片来设计各种实用电路那样，利用这些程序模块来设计自己的应用程序，从而有助于

加速完成自己的单片机应用开发任务。二是本书没有象其它单片机资料那样介绍每个程序的设计思想及过程，而是从应用角度出发象集成电路芯片手册那样介绍了每个程序的功能、入口、出口参数、资源占用情况以及与其它程序的联系。为了便于读者理解和使用各个子程序，还给出程序框图。三是本书所有子程序都是调试通过的，并存于磁盘上。

作为单片机实用于程序库，作者希望本书能象集成电路芯片手册那样，成为从事单片机开发和应用的科技人员的有力助手，为加快单片机在我国的普及起到推动作用，

全书从计算、转换、处理、控制四部分出发分八章，第一章~第四章给出了定点算术运算、浮点算术运算、开平方及基本函数运算，第五、六章介绍了码制转换排序查找等常用程序。为了便于程序设计的比较，以适合于不同应用场合的需要，同一种运算选用了不止一个程序。为了满足对实时性的要求，还提供了一些新颖的高效快速的运算、转换程序。第七章选择了一些在数据采集和处理时常用的程序，如数字滤波、标度变换以及等精度测量数据处理等等。最后一章介绍了单片机典型应用时键盘处理及显示程序的设计。给出了计算机控制最常用的PID算法的程序实现。

在本书的编写过程中，作者始终得到孙圣和教授的热情指导，得到哈尔滨工业大学航天学院电子仪器及测量技术教研室的许多同志大力协助，在此表示衷心地感谢。

由于作者水平有限，错误难免，欢迎批评指正。

编 者

一九八九年七月

目 录

第一章 二进制定点数运算程序	(1)
§ 1.1 双字节数取补子程序 (CMPT)	(1)
§ 1.2 双字节原码左移一位子程序 (DRL)	(2)
§ 1.3 双字节原码右移一位子程序 (DRR)	(2)
§ 1.4 双字节补码右移一位子程序 (CRR)	(3)
§ 1.5 双字节补码加法子程序 (NADD)	(4)
§ 1.6 双字节补码减法子程序 (NSUB)	(4)
§ 1.7 双字节原码加减法子程序 (DSUB)	(5)
§ 1.8 采用重复加法的双字节无符号数乘法子程序 (NMUL)	(7)
§ 1.9 原码有符号双字节数乘法子程序 (IMUL)	(10)
§ 1.10 原码有符号双字节小数乘法子程序 (DMUL)	(11)
§ 1.11 无符号双字节数快速乘法子程序 (QMUL)	(14)
§ 1.12 采用比较法的无符号双字节数除法子程序 (NDIV)	(15)
§ 1.13 原码有符号双字节整数除法子程序 (IDIV)	(18)
§ 1.14 原码有符号双字节小数除法子程序 (DDIV)	(20)
§ 1.15 多字节无符号整数加法子程序 (NIADD)	(22)

§ 1.16	多字节无符号整数减法子程序 (NISUB)	(24)
§ 1.17	多字节无符号整数乘法子程序 (NIMUL)	(26)
§ 1.18	多字节数取补子程序 (NINORM)	(30)
§ 1.19	多字节有符号整数乘法子程序 (SNIMUL)	(32)
§ 1.20	多字节数判零子程序 (NIZERO)	(36)
§ 1.21	多字节无符号整数除法子程序 (NIDIV)	(37)
§ 1.22	多字节有符号整数除法子程序 (SNIDIV)	(42)
第二章	十进制数运算程序	(47)
§ 2.1	十进制数求补运算符子程序 (CCMP)	(47)
§ 2.2	多字节十进制数加法子程序 (CADD)	(48)
§ 2.3	多字节十进制数减法子程序 (CSUB)	(50)
§ 2.4	多字节十进制数原码加减法子程序 (CDSUB)	(52)
§ 2.5	无符号十进制数乘法子程序 (CMUL)	(55)
§ 2.6	无符号十进制数除法子程序 (CDIV)	(59)
第三章	二进制浮点数运算程序	(63)
§ 3.1	左规格化子程序 (NORM)	(63)
√ § 3.2	通用规格化子程序 (FSDT)	(66)
√ § 3.3	浮点数加减法处理子程序 (FABP)	(69)
√ § 3.4	浮点数取数子程序 (FMLD)	(73)

√§ 3.5	浮点数加法子程序 (FADD)	(75)
√§ 3.6	浮点数减法子程序 (FSUB)	(76)
√§ 3.7	浮点数存数子程序 (FSTR)	(77)
√§ 3.8	浮点数乘法子程序 (FMUL)	(78)
§ 3.9	浮点数除法子程序 (FDIV)	(81)
第四章	常用函数运算程序	(86)
§ 4.1	双字节整数开平方子程序 (DSQR)	(86)
§ 4.2	浮点数开平方子程序 (FSQR)	(87)
§ 4.3	定点小数正弦函数计算子程序 (DSIN)	(91)
§ 4.4	三字节原码表示浮点数多项式计算子程序 (FPLN)	(94)
§ 4.5	浮点数正弦函数子程序 (FSIN)	(98)
§ 4.6	浮点数指数函数子程序 (FEXP)	(103)
§ 4.7	浮点数反正切函数子程序 (FATN)	(107)
§ 4.8	浮点数自然对数函数子程序 (FLNX)	(111)
§ 4.9	纯小数快速开平方子程序 (SQRF)	(116)
§ 4.10	快速对数计算子程序 (LOGQ)	(118)
第五章	码制转换程序	(123)
§ 5.1	单字节BCD码转换成压缩的BCD码子程序 (BCDC)	(123)
§ 5.2	压缩的BCD码转换成单字节BCD码子程序 (CBCD)	(125)
§ 5.3	双字节整数转换成三字节原码表示的浮 点数子程序 (INTF)	(126)
§ 5.4	浮点数取整子程序 (FINT)	(128)

§ 5.5	四位十进制整数转换成二进制整数子程序 (IDTB)	(129)
§ 5.6	四位十进制小数转换成二进制小数子程序 (PDTB)	(132)
§ 5.7	双字节整数二翻十子程序 (IBTD)	(134)
§ 5.8	双字节小数二翻十子程序 (PBTD)	(135)
§ 5.9	三字节浮点数二翻十子程序 (FBTD)	(138)
§ 5.10	十进制数转换成三字节浮点数子程序 (FDTB)	(146)
§ 5.11	多字节十进制整数转换为二进制整数子程序 (NIDTB)	(151)
§ 5.12	多字节二进制整数转换为十进制整数子程序 (NIBTD)	(155)
§ 5.13	多字节十进制小数转换为二进制小数子程序 (NPDTB)	(157)
§ 5.14	多字节二进制小数转换为十进制小数子程序 (NPBTD)	(159)
§ 5.15	一位十六进制数转换为ASCII码子程序 (HEXASC1)	(163)
§ 5.16	多位十六进制数转换为ASCII码子程序 (HEXASC2)	(164)
§ 5.17	多位十六进制数转换为ASCII码子程序 (HEXASC3)	(166)
§ 5.18	格雷码转换子程序 (CRAY)	(168)
第六章	表格处理程序	(171)
§ 6.1	检查数据出现次数子程序 (FRQ)	(171)
§ 6.2	串比较子程序 (CMP)	(173)

§ 6.3	顺序查找子程序 (SER)	(175)
§ 6.4	查找字符串子程序 (RST)	(178)
§ 6.5	插入排序子程序 (IST)	(181)
§ 6.6	快速排序子程序 (SST)	(184)
§ 6.7	查表子程序 (RTBA)	(187)
§ 6.8	大于256个字节表的查表子程序 (RTBB)	(188)
§ 6.9	找出输入字符串在表中位置序号子程序 (RTBC)	(190)
第七章	数据处理程序	(194)
§ 7.1	实用滤波子程序	(194)
§ 7.1-1	算术平均值法滤波子程序 (FAVG)	(194)
§ 7.1-2	防脉冲干扰平均值法滤波子程序 (DAVG)	(197)
§ 7.1-3	低通滤波子程序 (LPAS)	(200)
§ 7.1-4	移动平均滤波子程序 (SMFM)	(202)
§ 7.2	标度变换子程序	(205)
§ 7.2-1	线性标度变换子程序 (UNEX1)	(205)
§ 7.2-2	非线性标度变换子程序 (UNEX2)	(208)
§ 7.3	等精度直接测量的数据处理子程序	(212)
§ 7.3-1	算术平均值的计算子程序 (MEAN)	(212)
§ 7.3-2	均方差 σ 的计算子程序 (DMSRM)	(214)

§ 7.3-3 剔除可疑测量值子程序 (EDMD)	(225)
§ 7.4 数字滤波子程序	(228)
§ 7.4-1 无限长单位脉冲响应数字滤波子程序 (IIR)	(226)
§ 7.4-2 有限长单位脉冲响应数字滤波子程序 (FIR)	(240)
§ 7.5 快速傅里叶变换子程序 (FFT)	(250)
第八章 控制程序	(274)
§ 8.1 通用显示子程序 (DISP1)	(274)
§ 8.2 键盘处理子程序 (KEYS)	(277)
§ 8.3 利用串行口显示子程序 (DISP2)	(280)
§ 8.4 增量式PID算法子程序 (PID1)	(281)
§ 8.5 位置式PID算法子程序 (PID2)	(287)
附表1 按字母顺序排列的指令表	(292)
附表2 按功能分类的指令表	(296)
附表3 按代码顺序排列的指令表	(306)
附表4 影响标志位设置的指令	(314)
附表5 特殊功能寄存器地址表	(315)
附表6 8031内部RAM中20H~2FH的位地址表	(316)
参考文献	(317)

第一章 二进制定点数运算程序

定点数就是小数点固定的数，它可分为整数、小数。另外按数的正、负可分为无符号数和有符号数。有符号数的表示方法有原码和补码两种。本章包括定点整数的加、减、乘、除运算，定点小数的加、减、乘、除运算，以及一些移位和取补子程序等。

§ 1.1 双字节数取补子程序(CMPT)

功能： $(R4R5)$ 取补 $\rightarrow (R4R5)$ ，其中 $R4$ 为高位， $R5$ 为低位。

使用寄存器： A ， $R4$ ， $R5$ 及标志 CY 。

例如： $(R4R5) = 1234$

执行后得： $(R4R5) = EDCC$

程序清单：

```
CMPT:    MOV     A, R5
          CPL     A
          ADD     A, #1
          MOV     R5, A
          MOV     A, R4
          CPL     A
          ADDC    A, #0
```

```
MOV    R4,A
RET
```

§ 1.2 双字节原码左移一位子程序(DRL)

功能： (R2R3) 左移一位→ (R2R3) ；不改变符号位，不考虑溢出。其中R2 为高位，R3 为低位。

使用寄存器： A, R2,R3及标志CY。

例如： (R2R3) = 8002

执行后得： (R2R3) = 8004

程序清单：

```
DRL:      MOV     A,R3
           CLR     C
           RLC     A
           MOV     R3,A
           MOV     A,R2
           RLC     A
           MOV     A.7,C
           MOV     R2,A
           RET
```

§ 1.3 双字节原码右移一位子程序(DRR)

功能： (R2R3) 右移一位→ (R2R3) ，不改变符号位。其中R2为高位，R3为低位。

使用寄存器： A, R2, R3及标志CY。

例如： (R2R3) = 8002

执行后得: $(R_2R_3) = 8001$

程序清单:

```
DRR:      MOV      A,R2
           MOV      C,A.7
           CLR      A.7
           RRC      A
           MOV      R2,A
           MOV      A,R3
           RRC      A
           MOV      R3,A
           RET
```

§ 1.4 双字节补码右移一位子程序(CRR)

功能: (R_2R_3) 右移一位 $\rightarrow (R_2R_3)$, 不改变符号位。

其中 R_2 为高位, R_3 为低位。

使用寄存器: A, R_2, R_3 及标志 CY 。

例如: $(R_2R_3) = 8002$

执行后得: $(R_2R_3) = C001$

程序清单:

```
CRR:      MOV      A,R2
           MOV      C,A.7
           RRC      A
           MOV      R2,A
           MOV      A,R3
           RRC      A
           MOV      R3,A
           RET
```

000010000000 0000 0010
01100000 00000001

§ 1.5 双字节补码加法子程序(NADD)

功能: $(R2R3) + (R6R7) = (R4R5)$

入口: R2R3中存放被加数, R6R7中存放加数。

出口: 和在R4R5中, OV = 1表示溢出。

使用寄存器: A, R2~R7。

例如: $(R2R3) = 1234; (R6R7) = 1111$

执行后得: $(R4R5) = 2345$

程序清单:

```
NADD:    MOV     A, R3
          ADD     A, R7
          MOV     R5, A
          MOV     A, R2
          ADDC    A, R6
          MOV     R4, A
          RET
```

§ 1.6 双字节补码减法子程序(NSUB)

功能: $(R2R3) - (R6R7) = (R4R5)$

入口: R2R3中存放被减数, R6R7中存放减数。

出口: 差存放在R4R5中, OV = 1表示溢出。

使用寄存器: A, R2~R7及标志CY。

例如 $(R2R3) = 2345; (R6R7) = 1111$

执行后得: $(R4R5) = 1234$

程序清单:

```
NSUB:    MOV     A,R3
          CLR     C
          SUBB    A,R7
          MOV     R5,A
          MOV     A,R2
          SUBB    A,R6
          MOV     R4,A
          RET
```

§ 1.7 双字节原码加减法子程序(DSUB)

功能: $(R2R3) \pm (R6R7) = (R4R5)$

入口: R2R3中存放被加数(或被减数), R6R7中存放加数(或减数)。

出口: 和(或差)存放在R4R5中, CY=1 发生溢出, CY=0正常。

使用寄存器: A, R2~R7及标志CY, F0。

调用程序: NADD, NSUB, CMPT。

说明: 数据均为原码表示的数, 最高位为符号位。

DADD为原码加法子程序入口, DSUB 为原码减法子程序入口。

例 1 DADD入口 $(R2R3) = 8234; (R6R7) = 0111$

执行后得: $(R4R5) = 8123$

例 2 DSUB入口 $(R2R3) = 8234; (R6R7) = 0111$

执行后得: $(R4R5) = 8345$

程序清单:

DSUB:	MOV	A,R6
	CPL	A.7
	MOV	R6,A
DADD:	MOV	A,R2
	MOV	C,A.7
	MOV	F0,C
	XRL	A,R6
	MOV	C,A.7
	MOV	A,R2
	CLR	A.7
	MOV	R2,A
	MOV	A,R6
	CLR	A.7
	MOV	R6,A
	JC	DAB2
	ACALL	NADD
	MOV	A,R4
	JB	A.7,DABE
DAB1:	MOV	C,F0
	MOV	A.7,C
	MOV	R4,A
	RET	
DABE:	SETB	C
	RET	
DAB2:	ACALL	NSUB
	MOV	A,R4
	JNB	A.7,DAB1

LCALL CMPT
CPL F0
SJMP DAB1
RET

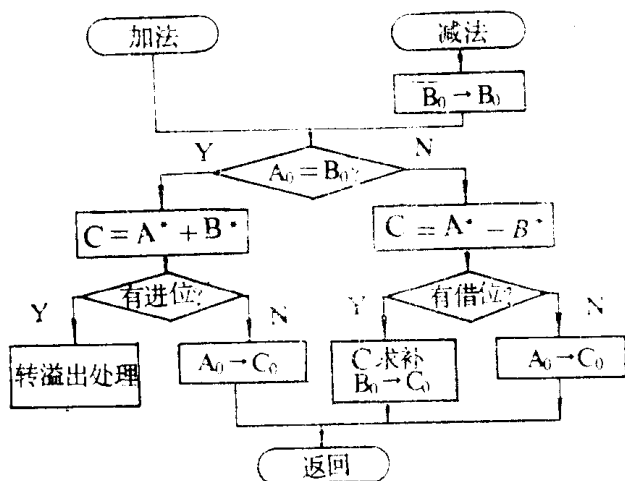


图 1-1 原码加减法运算程序框图

§ 1.8 采用重复加法的双字节无符号数乘法子程序(NMUL)

功能： $(R2R3) * (R6R7) = (R4R5R6R7)$

入口： R2R3中存放被乘数，R6R7中存放乘数。

出口： 积存放在R4R5R6R7中。

使用寄存器： A, R2~R7及标志CY。

例如： $(R2R3) = 0011; (R6R7) = 0011$

执行后得： $(R4R5R6R7) = 00000121$

程序清单：

NMUL:	MOV	R4, #0
	MOV	R5, #0
	MOV	R0, #16
	CLR	C
NMLP:	MOV	A, R4
	RRC	A
	MOV	R4, A
	MOV	A, R5
	RRC	A
	MOV	R5, A
	MOV	A, R6
	RRC	A
	MOV	R6, A
	MOV	A, R7
	RRC	A
	MOV	R7, A
	JNC	NMLN
	MOV	A, R5
	ADD	A, R3
	MOV	R5, A
	MOV	A, R4
	ADDC	A, R2
	MOV	R4, A
NMLN:	DJNZ	R0, NMLP
	MOV	A, R4
	RRC	A