



**The Application Examples
for Popular and Useful Subprogram
of Singlechip Computer**

流行单片机实用子程序 及应用实例

杨振江 杜铁军 李 群 编著



西安电子科技大学出版社
<http://www.xduph.com>

流行单片机实用子程序 及应用实例

杨振江 杜铁军 李群 编著

西安电子科技大学出版社
2002

内 容 简 介

本书从应用角度出发,精选了在我国最流行的MCS-51、PIC和MCS-96三种单片机的实用于程序和典型器件应用实例20多类100余种。内容包括单片机的基本运算、排序、数码转换、数字滤波、软件抗干扰、串行通信、中断处理、器件接口等实用程序。书中所涉及的内容大部分都经过上机验证,具有很强的实用性。有些例子直接来自于科学研究和生产实践,有些例子稍加修改就可解决工作中的实际问题。

本书对从事智能仪器设计、数据采集、自动控制、数字通信及计算机接口等领域工作的科技人员和广大的单片机爱好者都具有很高的使用和参考价值,也可作为工科大专院校有关课程的教学参考书。

图书在版编目(CIP)数据

流行单片机实用于程序及应用实例/杨振江等编著.

西安:西安电子科技大学出版社,2002.7

ISBN 7-5606-1137-0

I. 流… II. 杨… III. 单片微型计算机-程序设计 IV. TP368.1

中国版本图书馆 CIP 数据核字(2002)第 036127 号

责任编辑 云立实

出版发行 西安电子科技大学出版社(西安市太白南路2号)

电 话 (029)8227828 邮 编 710071

http://www.xduph.com E-mail: xdupfxb@pub.xaonline.com

经 销 新华书店

印 刷 西安兰翔印刷厂

版 次 2002年7月第1版 2002年7月第1次印刷

开 本 787毫米×1092毫米 1/16 印张 14.375

字 数 336千字

印 数 1~4 000册

定 价 19.00元

ISBN 7-5606-1137-0/TP·0578

XDUP 1408001-1

*** 如有印装问题可调换 ***

本书封面贴有西安电子科技大学出版社的激光防伪标志,无标志者不得销售。

前 言

单片机以其体积小、功能齐全、价格低廉、可靠性高等特点，在各个领域获得了广泛的应用，特别在工业控制、智能化仪器仪表、产品自动化、分布式控制系统中都已取得了可喜的成果。单片机已经成为衡量工业发展水平的标志之一，是产品更新换代、发展新技术、改造老产品的主要手段。目前，在众多的单片机产品中，MCS-51 系列、PIC 系列及 MCS-96 系列单片机是我国单片机应用的主流机种。

大家知道，在单片机系统设计中，程序设计是非常重要的环节，它的质量直接影响到整个系统的性能。用汇编语言进行程序设计的过程和用高级语言设计程序有相似之处，其设计过程大致可以分为以下几个步骤：

- (1) 明确课题对程序功能、运算精度、执行速度等方面的要求及硬件条件。
- (2) 把复杂问题分解为若干个模块，确定各模块的处理方法，画出程序流程图(简单问题可以不画)。如果各模块仍较为复杂，还应分别画出分模块流程图和总的流程图。
- (3) 正确分配存储器资源，如各程序段的存放地址、数据区地址、工作单元分配等。
- (4) 根据流程图精心组合合适的指令和编制源程序。

然而，由于单片机的特殊结构，在编写应用程序时，绝大部分还需要用汇编语言编写，特别是在尽量减少机器代码，提高执行速度，频繁操作输入输出和实时控制等方面更是如此。因此，从应用角度来看，随着单片机应用的普及，编写应用程序的问题愈来愈明显地摆在用户面前，成为当务之急。

编写本书的主要目的，就是为了解决用户的某些难题：

- (1) 使用户得心应手地调用某些子程序以组成具有特定功能的应用程序。
- (2) 使用户可以借鉴所介绍的程序实例和某些器件的功能，方便地编写具体应用程序，解决编写程序中的困难，减少那些不必要的重复性工作。
- (3) 在使用这些程序的过程中，取长补短，总结经验，找到编制程序的方法，进一步提高编制程序的水平和技能。

总之，编写本书就是要使用户视其所需，取之即来，为其所用，使单片机的潜在能力充分发挥作用，真正为用户解决一些工作中的实际问题，为单片机应用的普及和推广创造一些现实条件。

本书共分 5 章，内容包括 MCS-51 系列、PIC 系列和 MCS-96 系列单片机的基本运算、排序、数码转换、软件抗干扰、数字滤波、串口、中断等实用程序。

本书主要由杨振江编写，另外参加部分章节编写的还有杜铁军、李群等同志。在编写过程中，始终得到了云立实编辑的大力支持和帮助；王曙梅同志为程序的整理付出了心血；杨京武同志也提供了不少资料。在此，笔者谨向他们表示由衷的感谢。

由于作者水平所限，书中错误或不妥之处在所难免，诚恳希望读者提出宝贵意见。

编著者

2002 年 4 月

于西安电子科技大学

MS 74/03

目 录

第 1 章 MCS-51 单片机基本运算子程序	1
1.1 二进制基本运算子程序	1
1.1.1 两个半字节数合并成一个一字节数子程序 RZD_YZD	1
1.1.2 双字节数取补子程序 D_QBSUB	1
1.1.3 双字节补码(或原码)加法子程序 D_BYJSUB	2
1.1.4 两个无符号双字节数相加子程序 DW_ADD1	2
1.1.5 多字节无符号加法子程序 D_WADD1	3
1.1.6 多字节无符号整数加法子程序 D_WADD2	3
1.1.7 双字节补码(或原码)减法子程序 D_QYJSUB	4
1.1.8 多字节无符号整数减法子程序 D_NISUB1	4
1.1.9 多字节无符号减法子程序 D_NISUB2	5
1.1.10 双字节无符号乘法子程序 D_MULSUB	5
1.1.11 双字节乘法子程序 H2MUL_SUB	6
1.1.12 双字节无符号快速乘法子程序 D_MULSUB2	7
1.1.13 多字节数乘 10 子程序 D_MUL10	8
1.1.14 三字节乘二字节乘法子程序 H3MUL_SUB	9
1.1.15 四字节乘四字节乘法子程序 H4MUL_SUB	10
1.1.16 多字节无符号数乘法子程序 D_WMUL1	12
1.1.17 多字节无符号整数乘法子程序 D_NIMUL2	14
1.1.18 双字节平方根子程序 SQRT_SUB	15
1.1.19 双字节无符号除法子程序 D_WFHDIV	16
1.1.20 多字节无符号除法子程序 D_NIDIV	17
1.1.21 带符号双字节除单字节除法子程序 SHF_1DIV	18
1.1.22 双字节除以双字节除法子程序 SHZ_2DIV	20
1.1.23 三字节除双字节商为三字节除法子程序 SH_32DIV	22
1.1.24 四字节除双字节商为二字节除法子程序 SH_42DIV	23
1.1.25 五字节除五字节除法子程序 SHZ_5DIV	25
1.1.26 程序转入相应的分支子程序 JMP_XX	26
1.2 二进制符点运算子程序	27
1.2.1 原码有符号双字节小数乘法子程序 YD_2MUL	27
1.2.2 双字节带符号小数乘法子程序 SHF_2MUL	28
1.2.3 原码有符号双字节小数除法子程序 YD_DIVX	30
1.2.4 尾数用原码表示的双字节浮点数乘法子程序 WS_2FMUL	31
1.2.5 尾数用原码表示的双字节除单字节浮点数除法子程序 WS_1FDIV	33
1.2.6 浮点数比较大小程序 FD_CMPP	35
1.3 十进制(BCD 码)运算子程序	36
1.3.1 单字节 BCD 码转换成压缩 BCD 码子程序 D_BCD_Y	36
1.3.2 十进制求补运算子程序 BCD_QMP	36

1.3.3	多字节十进制加法子程序 D-BCDADD	37
1.3.4	多字节十进制减法子程序 D-BCDSSUB	38
1.3.5	多字节十进制 BCD 码减法子程序 D-BCDSUB	38
1.3.6	无符号十进制数乘法子程序 W-BCDCMUL	39
1.3.7	无符号十进制数除法子程序 W-BCDDIV	40
1.3.8	四字节十进制除法子程序 DEV-BCD4	42
1.3.9	多字节无符号 BCD 码除法子程序 D-WBCDDIV	44
1.3.10	单字节 BCD 码平方根子程序 D-1SQR-SUB	46
1.3.11	双字节 BCD 码平方根子程序 D-2SQR-SUB	47
1.3.12	三字节 BCD 码平方根子程序 D-3SQR-SUB	48

第 2 章 MCS-51 单片机排序、数码转换等子程序 51

2.1	数据排序子程序	51
2.1.1	单字节无符号数排序子程序 SQRT-1WP	51
2.1.2	单字节无符号数据块排序子程序 SQRT-2WP	51
2.1.3	单字节带符号数按由小到大排序子程序 SQRT-3WP	52
2.1.4	双字节无符号数由小到大排序子程序 SQRT-4WP	53
2.1.5	存储器无符号数据排序子程序 DATA-SORT	54
2.1.6	单字节顺序检索子程序 DZ-1FIND	55
2.1.7	测试字符串的长度子程序 T-CHAR	56
2.1.8	比较两个无符号数大小子程序 CMP-DX	56
2.1.9	大于 256 个字节表的双字节查表子程序 D-FIND	56
2.2	数码转换子程序	57
2.2.1	二进制数转换成 ASCII 码字符串子程序 BIN-ASCII	57
2.2.2	四位二进制数转换为 ASCII 码子程序 SBIN-ASCII	58
2.2.3	十六进制数转换成 ASCII 码子程序 HEX-ASCII	58
2.2.4	十六进制数转换成二位 ASCII 码子程序 HEX-ASCII	59
2.2.5	ASCII 码转换为四位二进制数子程序 ASII-SBIN	60
2.2.6	ASCII 码转换成十六进制数子程序 ASCII-HEX	60
2.2.7	二进制数转换成十六进制数子程序 BIN-16BCH	61
2.2.8	BCD 码转换成二进制数转换子程序 BCD-BIN	61
2.2.9	四字节非压缩 BCD 码数据转换成二进制数子程序 S-BCD-BIN	62
2.2.10	四位十进制数(BCD 码)转换为二进制码子程序 S-BCDBIN	62
2.2.11	四位 BCD 码转换成十六位二进制数子程序 SZ-16BIN-S	64
2.2.12	双字节二进制数转换成三字节 BCD 码子程序 SBIN-BCD	65
2.2.13	双字节二进制小数转换成四字节 BCD 码子程序 DXS-SBCD	66
2.2.14	三字节二进制整数转换成四字节 BCD 码子程序 SBIN-SBCD	67
2.2.15	三字节小数 BCD 码转换成四字节二进制数子程序 SXB-SBIN	68
2.2.16	带符号小数 BCD 码转换成二进制数子程序 DX-BINSUB	69
2.3	其它实用子程序	70
2.3.1	读取汉字字模子程序 LOAD-ZM	70
2.3.2	点与点画图子程序 POINT-P	73
2.3.3	几种散转程序 PRG-JMP1/2/3/4/5	76

2.3.4 子程序调用时的几种参数传递程序 PRG_CALL1/2/3/4	80
第3章 MCS-51 单片机软件抗干扰、串口等子程序	85
3.1 单片机执行过程中的软件抗干扰措施	85
3.1.1 采用指令冗余对程序“跑飞”的一种软件处理	85
3.1.2 采用软件陷阱对程序“跑飞”的一种软件处理	85
3.1.3 使用“看门狗”对程序“跑飞”的软件处理	87
3.1.4 采用 8155 芯片对单片机“死机”的处理	89
3.1.5 通过“复位”使系统恢复正常	90
3.2 数字滤波子程序	93
3.2.1 中值数字滤波子程序 ZZ_FILLE	93
3.2.2 防脉冲干扰数字滤波子程序 FG_FILIE	94
3.2.3 循环冗余检测(CRC)子程序 LOP_CRCs	96
3.3 中断服务子程序	97
3.3.1 利用定时/计数器 T0、T1 端作为外部中断源子程序 TO_T0SUB	97
3.3.2 精确定时子程序 TIMES_P	98
3.3.3 扩展外部中断源子程序 IO_POOP	101
3.4 串行通信子程序	102
3.4.1 全双工串行通信子程序 QS_COMMA	102
3.4.2 多机串口通信子程序 D_COMMA	105
3.4.3 用外部中断配合查询接收串行数据子程序 YW_READCX	107
3.4.4 PC 机与 51 单片机的快速串行通信程序 COMUN_PC/COMUN_51	110
3.4.5 带纠错的串行通信子程序 JERR_COM	115
第4章 MCS-51 单片机应用程序	119
4.1 步进电机控制子程序 MOT_DRVb	119
4.1.1 硬件描述	119
4.1.2 应用程序	120
4.2 八位串行输出 A/D 转换器 TLC548/549 及应用程序	121
4.2.1 硬件描述	121
4.2.2 数据采集程序	124
4.3 11 路 12 位串行 A/D 转换器 TLC2543 及应用程序	124
4.3.1 硬件描述	124
4.3.2 数据采集程序	131
4.4 10 位串行 D/A 转换器 TLC5615 及应用程序	133
4.4.1 硬件描述	133
4.4.2 应用程序	135
4.5 八位串行 LED 显示驱动器 MAX7219/7221 及应用程序	136
4.5.1 硬件描述	136
4.5.2 应用程序	142
4.6 液晶显示驱动器 HD61202 及应用子程序	143
4.6.1 硬件描述	143
4.6.2 驱动程序的编写	146

4.7 液晶显示控制器 SED1330/1335/1336/E1330 及应用程序	149
4.7.1 硬件描述	149
4.7.2 应用程序	154
4.8 点阵液晶显示控制器 MSM6255 及应用程序	162
4.8.1 硬件描述	162
4.8.2 驱动及应用程序	168
4.9 高性能低功耗带 RAM 实时时钟芯片 DS1302 及应用程序	172
4.9.1 硬件描述	172
4.9.2 应用程序	177
4.10 跨越 2000 年的时钟芯片 DS12C887 及应用程序	182
4.10.1 硬件描述	182
4.10.2 器件的软件编程	186
4.11 二总线串行 EEPROM AT24C01/02/04/08/16 及应用程序	188
4.11.1 硬件描述	188
4.11.2 应用程序	193
4.12 51 单片机中两种典型的查表程序设计	195
4.12.1 RAM 表的程序设计	195
4.12.2 ROM 表的程序设计	196
4.13 μ P-16B 微打印机与 51 单片机的硬件接口与应用程序	198
4.13.1 硬件接口描述	198
4.13.2 软件编程	199
第 5 章 PIC 单片机与 MCS-96 单片机应用程序	201
5.1 PIC 单片机应用子程序	201
5.1.1 单字节无符号数 BCD 码加法子程序 WBCD-ADD	201
5.1.2 单字节无符号数 BCD 码减法子程序 WBCD-SUBB	202
5.1.3 双精度加法运算子程序 DJD-ADD	203
5.1.4 双精度减法运算子程序 DJD-SUBB	203
5.1.5 双精度乘法运算子程序 D-MULSUB	204
5.1.6 双精度除法运算子程序 DJD-DIV	206
5.1.7 二进制数转换成 BCD 码子程序 PICBIN-BCD	208
5.1.8 BCD 码转换成二进制数子程序 PICBCD-BIN	209
5.2 MCS-96 单片机应用程序	210
5.2.1 单字节 ASCII 码转换成十六进制数子程序 DASCII-HEX	210
5.2.2 单字节十六进制数转换成 ASCII 子程序 DHEX-ASCII	210
5.2.3 二通道数据采集程序 SCD2-SUB	211
5.2.4 转速系统的测量与控制程序 ZHUAN-CTR	213
5.2.5 用定时器 0 定时中断发送 ASCII 码字符串程序 T0-INTESUB	216
5.2.6 通过 HSO0 中断发送 ASCII 码字符串程序 HSO0-INTE	218
5.2.7 对应用程序进行加密程序 PRG-SAVE	220

第1章 MCS-51 单片机 基本运算子程序

1.1 二进制基本运算子程序

1.1.1 两个半字节数合并成一个一字节数子程序 RZD_YZD

1. 功能

将两个半字节数存放于内部 RAM 的 40H、41H 单元，两个半字节数合并成一个一字节数后，结果存入 50H 单元中。

该程序使用了 A 累加器和 R1 寄存器等资源。

2. 程序清单

```
RZD_YZD:  MOV    R1, #40H      ;设置 R1 为数据指针
           MOV    A, @R1    ;取出第一个单元中的内容
           ANL    A, #0FH   ;取第一个数的低半字节
           SWAP   A         ;移至高半字节
           INC    R1        ;修改数据指针
           MOV    A, @R1    ;取第二个单元中的内容
           ANL    A, #0FH   ;取第二个数的低半字节
           ORL    A, @R1    ;拼字
           MOV    50H, A    ;存放结果
           RET
```

1.1.2 双字节数取补子程序 D_QBSUB

1. 功能

把 R0 寄存器所指的 NUMB1 二字节数取补后返送到 R0 寄存器所指的二单元中。

该程序使用了 A 累加器和 R0 寄存器等资源。

2. 程序清单

```
D_QBSUB:  MOV    R0, #NUMB1  ;设置起始地址
           MOV    A, @R0    ;读低位字节且取反加 1
           CPL    A
           ADD    A, #01H
           MOV    @R0, A
           INC    R0
           MOV    A, @R0    ;读高位字节且取反加进位
           CPL    A
           ADDC   A, #00H
```

```
MOV    @R0, A
RET
```

1.1.3 双字节补码(或原码)加法子程序 D_BYJSUB

1. 功能

把 R0 寄存器所指的 NUMB1 二字节补码(或原码)数与 R1 寄存器所指的 NUMB2 二字节补码(或原码)数对应相加后,送回 R0 寄存器所指的二单元中。

该程序使用了 A 累加器和 R0、R1 寄存器等资源。

2. 程序清单

```
D_BYJSUB:  MOV    R0, #NUMB1    ;设置被加数起始地址(低位字节)
            MOV    R1, #NUMB2    ;设置加数起始地址(低位字节)
            MOV    A, @R0        ;读被加数低位字节
            ADD    A, @R1        ;加低位字节
            MOV    @R0, A
            INC    R0
            INC    R1
            MOV    A, @R0        ;读被加数高位字节
            ADDC   A, @R1        ;加高位字节及进位位
            MOV    @R0, A
            RET
```

1.1.4 两个无符号双字节数相加子程序 DW_ADD1

1. 功能

将被加数存放于内部 RAM 的 40H(高位字节)、41H(低位字节)单元中,加数存放于 50H(高位字节)、51H(低位字节)单元中,结果(和数)回送到 40H(高位字节)和 41H(低位字节)单元中。

该程序使用了 A 累加器和 R0、R1 寄存器等资源。

2. 程序清单

```
DW_ADD1:  CLR    C            ;进位标志位 CY 清 0
            MOV    R0, #41H    ;被加数地址送数据指针 R0
            MOV    R1, #51H    ;加数地址送数据指针 R1
            MOV    A, @R0      ;被加数低字节的内容送入 A
            ADD    A, @R1      ;两个低字节相加
            MOV    @R0, A      ;低字节的和存入被加数低字节中
            DEC    R0          ;指向加数高位字节
            DEC    R1          ;指向加数高位字节
            MOV    A, @R0      ;被加数高位字节送入 A
            ADDC   A, @R1      ;两个高位字节带 CY 相加
            MOV    @R0, A      ;高位字节的和送被加数高位字节
            RET
```

1.1.5 多字节无符号加法子程序 D_WADD1

1. 功能

在调用程序前,被加数放在以 R0 为起始地址的单元(低字节在前)中,加数放在以 R1 为起始地址的单元(低字节在前)中,字节数放在 R2 中,结果回送到以 R0 为起始地址的单元(低字节在前)中,字节数放在 R3 中。

该程序使用了 A 累加器和 R0、R1、R3 寄存器等资源。

2. 程序清单

```
D_WADD1:  PUSH    PSW                ;保护标志寄存器内容
           CLR     C                ;进位位清 0
           MOV     R3, #00H
ADD10:    MOV     A, @R0            ;相加
           ADDC    A, @R1
           MOV     @R0, A
           INC     R0                ;地址值增 1
           INC     R1
           INC     R3                ;字节数增 1
           DJNZ    R2, ADD10        ;所有字节未加完继续,否则向下执行
           JNC     ADD20            ;无进位去 ADD20,有进位向下执行
           MOV     @R0, #01H        ;和的最高字节地址内容为 01H
           INC     R3                ;字节数增 1
ADD20:    POP     PSW              ;恢复标志寄存器内容
           RET                     ;返回主程序
```

1.1.6 多字节无符号整数加法子程序 D_WADD2

1. 功能

将(R0)指出的 n 字节无符号整数和(R1)指出的 n 字节无符号整数相加,和数存放于(R0)所指的 RAM 单元中,其中低位字节在前,R3 中为字节数 n。

该程序使用了 A 累加器与 R0、R1、R3、R4、R7 寄存器及进位标志位 CY 等资源。

2. 程序清单

```
D_WADD2:  MOV     A, R0                ;将指针 R0 暂存到 R4 中
           MOV     R4, A
           MOV     A, R3
           MOV     R7, A
           CLR     C
SA20I:    MOV     A, @R0            ;执行加法运算
           ADDC    A, @R1
           MOV     @R0, A
           INC     R0
           INC     R1
           DJNZ    R7, SA20I        ;循环 n 次
```

```

CLR      A
MOV      ACC.0, C
MOV      @R0, A
INC      R3          ;字节数加1
MOV      A, R4
MOV      R0, A        ;恢复 R0 值并返回
RET

```

1.1.7 双字节补码(或原码)减法子程序 D_QYJSUB

1. 功能

把 R0 寄存器所指的 NUMB1 二字节补码(或原码)数与 R1 寄存器所指的 NUMB2 二字节补码(或原码)数对应相减后,送回 R0 寄存器所指的二单元中。

该程序使用了 A 累加器和 R0、R1 寄存器等资源。

2. 程序清单

```

D-QYJSUB:  MOV      R0, #NUMB1      ;设置被减数起始地址(低位字节)
           MOV      R1, #NUMB2      ;设置减数起始地址(低位字节)
           MOV      A, @R0          ;读被减数低位字节
           CLR      C
           SUBB     A, @R1          ;减低位字节
           MOV      @R0, A
           INC      R0
           INC      R1
           MOV      A, @R0          ;读被减数高位字节
           SUBB     A, @R1
           MOV      @R0, A
           RET

```

1.1.8 多字节无符号整数减法子程序 D_NISUB1

1. 功能

将(R0)指向的内部 RAM 中 n 字节无符号整数减去(R1)指向的 n 字节无符号整数,它们的差存放于(R0)指向的 n 个字节单元中,其中,低位字节在前。

该程序使用了 A 累加器和 R0、R1、R3、R4、R7 寄存器,以及进位标志位 CY 等资源。

2. 程序清单

```

D-NISUB1:  MOV      A, R0          ;将 R0 指针保护到 R4 中
           MOV      R4, A
           MOV      A, R3
           MOV      R7, A
           CLR      C
SB20H:     MOV      A, @R0          ;执行减法运算
           SUBB     A, @R1
           MOV      @R0, A

```

```

INC      R0
INC      R1
DJNZ     R7, SB20H      ; 循环 n 次
MOV      A, R4
MOV      R0, A          ; 恢复 R0 并返回
RET

```

1.1.9 多字节无符号减法子程序 D_NISUB2

1. 功能

在调用程序前,被减数放在以 R0 为起始地址的单元(低字节在前)中,减数放在以 R1 为起始地址的单元(低字节在前)中,字节数放在 R2 中。结果回送到以 R0 为起始地址的单元(低字节在前)中,字节数放在 R3 中。07H 为符号位,“0”表示差为正,“1”表示差为负。负数是补码形式,若要得其原码,须再求补一次。

该程序使用了 A 累加器和 R0、R1、R3 寄存器等资源。

2. 程序清单

```

D_NISUB2:  PUSH    PSW                ; 标志寄存器内容进栈
           CLR     C                  ; 进位位清 0
           CLR     07H                ; 符号位清 0
           MOV     R3, #00H           ; 差字节计数器清 0
SUB10:     MOV     A, @R0
           SUBB    A, @R1             ; 相减
           MOV     @R0, A
           INC     R0                 ; 地址值增 1
           INC     R1
           INC     R3                 ; 差字节数增 1
           DJNZ    R2, SUB10          ; 未减完继续,减完向下执行
           JNC     SUB20              ; 差为正,去 SUB20
           SETB    07H                ; 差为负,置“1”符号位
SUB20:     POP     PSW                ; 恢复标志寄存器内容
           RET

```

1.1.10 双字节无符号乘法子程序 D_MULSUB

1. 功能

将 R3、R4 寄存器和 R5、R6 寄存器中的双字节无符号数对应相乘,结果送往以 NUMB3 为起始地址的四个单元中,其中低位字节在前。

该程序使用了 A 累加器和 R0、R1、R3、R6 寄存器等资源。

2. 程序清单

```

D_MULSUB:  MOV     R0, #NUMB3        ; 置结果起始地址
           MOV     A, #00H
           MOV     R1, #04H
CLR_01:    MOV     @R0, A            ; 结果区先清 0

```

```

                INC      R0
                DJNZ     R1, CLR_01
CON_MUL1:      MOV      R0, #NUMB3      ;置结果起始地址
                MOV      A, R4          ;读乘数低位字节数
                ADD      A, @R0
                MOV      @R0, A        ;放到(NUMB3+0)单元中
                INC      R0
                MOV      A, R3          ;读乘数高位字节数
                ADDC     A, @R0
                MOV      @R0, A        ;放到(NUMB3+1)单元中
                INC      R0
                MOV      A, #00H
                ADDC     A, @R0        ;加进位位
                MOV      @R0, A        ;放到(NUMB3+2)单元中
                INC      R0
                ADDC     A, @R0        ;加进位位
                MOV      @R0, A        ;放到(NUMB3+3)单元中
                MOV      A, R6          ;判断是否加完
                CLR      C
                SUBB     A, #01H
                MOV      R6, A
                MOV      A, R5
                SUBB     A, #01H
                MOV      R5, A
                JNC      CON_MUL1
                RET

```

1. 1. 11 双字节乘法子程序 H2MUL_SUB

1. 功能

在调用程序前,被乘数和乘数分别放在内部 RAM 的 20H、21H、22H 和 23H 单元中,乘数位数放在 2AH 单元中。执行后,积数放在 R4、R5、R6 和 R7 寄存器中。

该程序使用了 A 累加器和 R0、R1、R4、R5、R6、R7 寄存器等资源。

2. 程序清单

```

H2MUL_SUB:     MOV      R0, 20H        ;装入被乘数低位数据到 R0 中
                MOV      R1, 21H        ;装入被乘数高位数据到 R1 中
                MOV      R2, 22H        ;装入乘数低位数据到 R2 中
                MOV      R3, 23H        ;装入乘数高位数据到 R3 中
                MOV      R4, #00H       ;将存放积的单元(R4、R5、R6、R7)清 0
                MOV      R5, #00H
                MOV      R6, #00H
                MOV      R7, #00H
                MOV      2AH, #10H     ;设置乘数位数

```

```

LOOP-D-M:  MOV      A, R4
           ADD      A, R4           ;从低位开始将乘积乘 2, 相当于积左移
           MOV      R4, A           ;将积的最低位放回 R4 中
           MOV      A, R5
           ADDC     A, R5
           MOV      R5, A
           MOV      A, R6
           ADDL     A, R6
           MOV      R6, A
           MOV      A, R7
           ADDC     A, R7
           MOV      R7, A           ;将积的最高位放回 R7 中
           MOV      A, R2
           ADD      A, R2
           MOV      R2, A
           MOV      A, R3
           ADDC     A, R3
           MOV      R3, A           ;乘数左移
           JNC      TO-LOOP0        ;若无进位则转 TO-LOOP0 执行
           MOV      A, R4
           ADD      A, R0
           MOV      R4, A
           MOV      A, R5
           ADDC     A, R1
           MOV      R5, A
           MOV      A, R6
           ADDC     A, #00H         ;考虑进位位
           MOV      R6, A
           MOV      A, R7
           ADDC     A, #00H         ;考虑进位位
           MOV      R7, A
TO-LOOP0:  DJNZ     2AH, LOOP-D-M   ;若乘数位数不为 0, 继续循环, 否则退出
           RET

```

1.1.12 双字节无符号快速乘法子程序 D_MULSUB2

1. 功能

把 R2、R3 寄存器和 R6、R7 寄存器中的无符号双字节数对应相乘，其乘积送回 R4、R5、R6、R7 寄存器中。

该程序使用了 A 累加器和 B、R2、R3、R4、R5、R6、R7 寄存器等资源。

2. 程序清单

```

D_MULSUB2: MOV      A, R3           ;读乘数低位字节
           MOV      B, R7           ;读被乘数低位字节

```

MUL	AB	;R3×R7
XCH	A, R7	;R7 中放低字节
MOV	R5, B	;R5 中放高字节
MOV	B, R2	
MUL	AB	;R2×R7
ADD	A, R5	
MOV	R4, A	
CLR	A	
ADDC	A, B	
MOV	R5, A	;R5(高字节)=R2×R7
MOV	A, R6	
MOV	B, R3	
MUL	AB	;R3×R6
ADD	A, R4	
XCH	A, R6	
XCH	A, B	
ADDC	A, R5	
MOV	R5, A	
MOV	PSW.5, C	;保存进位位
MOV	A, R2	
MUL	AB	;R2×R6
ADD	A, R5	
MOV	R5, A	
CLR	A	
MOV	ACC.0, C	
MOV	C, PSW.5	
ADDC	A, B	
MOV	R4, A	
RET		

1.1.13 多字节数乘 10 子程序 D_MUL10

1. 功能

在应用程序设计中,有时需要对一个多字节数做乘 10 运算,这时可用乘法指令 MUL,利用循环程序实现。由于低位字节乘 10,其积可能会超过八位。所以把本次乘积之低八位与上次(低位的字节)乘积的高八位相加作为本次之积存入。在进行相加时,有可能产生进位,因此可使用 ADDC 指令。这就要求进入循环之前 CY 必须清 0(第一次相加无进位),在循环体内未执行 ADDC 之前 CY 必须保持。由于执行 MUL 指令总是清除 CY,所以在该指令的前后可安排保护和恢复标志位的指令。程序实际上是逐字节进行这种相乘相加运算的,直到整个字节完毕,结束循环。程序调用前,多字节数低位字节地址存放于 R1 寄存器中,字节数存于 R7 寄存器中。

该程序使用了 A 累加器和 R1、R2、R7、B 寄存器等资源。

2. 程序清单

```

D_MUL10:  PUSH    PSW           ;保护现场
           PUSH    A
           PUSH    B
           CLR     C           ;清进位位
           MOV     R2, #00H     ;R2 清 0
SH10:     MOV     A, @R1       ;低字节送 A
           MOV     B, #0AH     ;10 送 B
           PUSH    PSW
           MUL     AB          ;字节乘 10
           POP     PSW
           ADDC    A, R2       ;上次积高八位与本次积低八位加得本次积
           MOV     @R1, A      ;送原存储单元
           MOV     R2, B
           INC     R1
           DJNZ    R7, SH10    ;未乘完转 SH10, 否则向下执行
           MOV     @R1, B
           POP     B
           POP     A
           POP     PSW
           RET

```

1.1.14 三字节乘二字节乘法子程序 H3MUL_SUB

1. 功能

在调用程序前,被乘数放在 R0(最低位)、R1、R2(最高位)寄存器中,乘数放在 R3(低位)、R4(高位)寄存器中,乘数位数放在 R7 寄存器中。执行后,积数放在内部 RAM 的 25H~29H 单元中。其中,29H 单元为最高位,25H 单元为最低位。

该程序使用了 A 累加器和 R0、R1、R2、R3、R4、R6、R7 寄存器等资源。

2. 程序清单

```

H3MUL_SUB: LCALL    CLR_MUL_5   ;调用运算区清 0 子程序
           MOV     R7, #10H     ;设置乘数位数
TO_LOOPG:  MOV     69H, R0      ;保护 R0 的内容
           MOV     6AH, #05H
           MOV     R0, #25H     ;从最低位开始将积左移一次
           CLR     C
LOOP_RLC:  MOV     A, @R0
           ADDC    A, @R0
           MOV     @R0, A
           INC     R0
           DJNZ    6AH, LOOP_RLC
           MOV     R0, 69H      ;恢复 R0 的值
           MOV     A, R3        ;将乘数位左移

```