

AA MOV R2,
AA MOV R2,A
AB MOV R3,A
AC B4,A
AD C,A
AE MOV R6,A
AF MOV R7,A
B0 MOV @R0,data
B1 MOV @R1,data
B2 JB5
B3 JMPP @A
B4 CALL (page5)
B5 CPLF 1
B6 JF0

MCS-48 单片微型计算机 实用子程序

姚志石 徐时亮 等编

3
3/1

复旦大学出版社

312
y: 3/

MCS-48 单片微型计算机 实用子程序

姚志石 涂时亮 等编

复旦大学出版社

内 容 提 要

本书较详细地介绍了 MCS-48 单片机汇编语言程序设计方法和技巧。每个程序都附有详细流程图，还列举了实用的键盘、显示、A/D、D/A 等详细硬件接线图，书中所列举的程序设计方法具有普遍意义。

本书内容取材于实践，对于从事微机应用(特别在智能仪器仪表、工业自动化控制方面的工程技术人员)是一本实用的参考书，也可作为微机初学者自学入门的指导书。

本书还附上 FD-35 在线仿真器的监控仿真程序。对研制和开发新的微型机的工程技术人员有一定的参考价值。

3551/16

MCS-48 单片微型计算机 实用子程序

复旦大学出版社出版
新华书店上海发行所发行
复旦大学印刷厂印刷

字数 211 千 开本 850×1163 1/32 印张 9.25

1986 年 11 月第一版 1986 年 11 月第一次印刷

印数：1—6,000

书号：13253·045

定价：2.60 元

前 言

近几年来,微型电脑的应用发展很快。Intel 公司的 MCS-48 单片机系列,在一块芯片上就集成了 CPU, RAM, ROM, 定时/计数器和多种功能 I/O, 具有体积小、价格低、控制功能强, 稳定可靠等优点。它广泛应用于生产过程的自动检测、实时控制和智能仪器仪表中, 是目前微机应用产品化最合适和应用最广泛的一种机种, 受到广大用户的欢迎。

由于单片机内存容量小, 人们不可能用高级语言来编制各种应用程序, 同时在有些实时控制中高级语言难于完成快速的实时控制。这就要求用户使用汇编语言来编制程序。但这样做难度较大, 给单片机的推广普及带来困难。我们在多次举办单片机应用短训班的基础上, 根据广大用户要求, 编制了一些应用性的实用子程序。其中大部分程序是实际应用的例子。本书为初学者介绍了一些汇编语言程序设计的基础和技巧。每个程序都画出详细的流程图, 所提供的设计方法基本适用于 8080/8085, Z80, 6800 等各种微型机的汇编语言程序设计。

本书由姚志石、涂时光同志主编。参加编写的还有张友德, 高哲鸣同志。

谢铭培同志审校了全书。在编写过程中我们还得到徐君毅、陈章龙同志的热情帮助和指导, 在此一并表示感谢。

由于作者水平有限, 时间仓促, 书中必有不当之处, 敬请读者批评指正。

作者

1984.10

目 录

前言

第一部分 MCS-48 单片微型计算机概论

第一章 MCS-48 简介

- 1.1 运算部件 (3)
- 1.2 程序存储器 (4)
- 1.3 数据存储器 (4)
- 1.4 输入/输出 (5)
- 1.5 程序状态字 (5)
- 1.6 中断 (6)
- 1.7 复位与单步操作 (6)
- 1.8 MCS-48 处理机引脚功能 (9)

第二章 MCS-48 指令系统简介

- 2.1 引言 (12)
- 2.2 指令系统说明 (16)
- 2.3 指令系统详介 (17)

第二部分 MCS-48 实用子程序

第三章 代码转换

- 3.1 十六进制 $\leftrightarrow$ ASCII 码转换程序 $\overline{HEASC}$ (43)
- 3.2 ASCII 码 $\leftrightarrow$ 十六进制数转换程序 $\overline{ASCHE}$ (44)
- 3.3 ASCII 码 $\leftrightarrow$ BCD 码转换程序 $\overline{ABCHC}$ (45)
- 3.4 BCD 码 $\leftrightarrow$ ASCII 码转换 $\overline{BCDASC}$ (46)
- 3.5 BCD 码 $\leftrightarrow$ 二进制转换程序 $\overline{BCDB}$ (48)

- 3.6 十六位二进制数 $\rightarrow$ BCD码的转换程序 1, BINBCD.....(49)
 3.7 十六位二进制数 $\rightarrow$ BCD码转换程序 2, BBCD.....(51)

第四章 二进制定点运算

- 4.1 多字节加减法 AADD, SUBAB.....(56)
 4.2 单字节无符号数乘法 SFU.....(60)
 4.3 单字节有符号数乘法(整数) SYU1.....(61)
 4.4 单字节有符号数乘法(纯小数) SYU2.....(64)
 4.5 多字节无符号数乘法 MUX.....(67)
 4.6 多字节有符号数乘法 SFMUX.....(70)
 4.7 单字节无符号数除法 SDV.....(74)
 4.8 单字节有符号数除法 SYDIV.....(75)
 4.9 多字节无符号数除法 UXDV.....(79)
 4.10 多字节有符号数除法 UXDVF.....(84)
 4.11 求单字节的平方根 SQRS.....(89)
 4.12 求三字节的平方根的整数部分 SQRT.....(90)

第五章 BCD 码运算

- 5.1 BCD码定点运算(寄存器间运算) BCDA, BCDS.....(94)
 5.2 BCD码双字节加减法 SUBCD, ADDBCD.....(95)
 5.3 BCD码多字节加减法内存运算 AAADD, ASUB.....(96)
 5.4 一位数BCD码乘法 ANDI.....(101)
 5.5 单字节BCD码乘法 ABCD.....(102)
 5.6 BCD码双字节定点乘法 BCDMUL.....(103)
 5.7 单字节BCD数的除法 DIV.....(107)
 5.8 双字节十进制数除法 BUBCD.....(110)
 5.9 单字节BCD码整数平方根 SQRBI.....(115)
 5.10 求双字节BCD码整数平方根 SQRBI2.....(116)

第六章 A/D, D/A 部分

- 6.1 A/D, D/A 的使用例 I.....(119)
 6.2 A/D 转换使用例 II.....(128)

第七章 浮点运算子程序

7.1	对阶子程序 ADJ	(136)
7.2	右规子程序 1, RST1	(138)
7.3	右规子程序 2, RST2	(139)
7.4	左规子程序 LST	(140)
7.5	浮点加减法子程序	(142)
7.5.1	浮点加法子程序 FADD	(142)
7.5.2	浮点减法子程序 FSUB	(143)
7.6	浮点乘法子程序 FMUL	(144)
7.7	浮点除法子程序 FDIV	(145)
7.8	定点无符号小数双字节乘法子程序 DMUL	(147)
7.9	定点无符号小数双字节除法子程序 DDIV	(150)
7.10	正弦函数 SinX 子程序	(153)
7.11	多项式 $y=a_0x^n+a_1x^{n-1}+\dots+a_n$ 的计算方法	(156)
7.11.1	计算 $y=((a_0x+a_1)x+a_2)x+a_3$ 的主程序	(159)
7.11.2	取常数子程序 BA	(161)
7.11.3	数符阶符处理子程序 AH	(163)
7.11.4	乘法子程序(定浮点混合) ANBCD	(164)
7.11.5	加法子程序 BAND	(166)
7.11.6	四舍五入子程序 AAA	(169)
7.11.7	字节移位子程序 RRA	(171)
7.11.8	对阶处理程序 RABC	(173)
7.11.9	浮点加减程序 RMA	(177)
7.11.10	取数子程序 AHR	(181)
7.11.11	减法子程序 SUBCD	(182)
7.11.12	移位子程序 RLA	(185)
7.11.13	传递子程序 RHA	(187)
7.11.14	左规子程序 RAB	(188)
7.11.15	左规子程序 RLR	(190)

第八章 键盘和显示部分

- 8.1 键盘扫描和显示程序 (1).....(192)
- 8.2 键盘扫描和显示程序 (2).....(203)
- 8.3 键盘扫描和显示程序 (3).....(215)
- 8.4 8035 和 GP 16 打印机的接口(221)

第九章 其他程序

- 9.1 实时钟程序 TCT(234)
- 9.2 双字节定点数比较子程序 DCMF(237)
- 9.3 数字(多字节数)移位子程序 RRRR(237)
- 9.4 多寄存器内存移位子程序 RLR(239)
- 9.5 找最大数(无符号)子程序 SCH(241)
- 9.6 顺序检索子程序 STE(243)

第三部分 FD-35- II 型开发仿真程序

第十章 FD-35- I 型原理与程序

- 10.1 EPROM 仿真器的结构原理.....(247)
- 10.2 FD-35- II 程序(250)

附录: MCS-48 单片微计算机指令手册

第一部分

MCS-48 单片微型计算机概论



第一章 MCS-48 简介

8048/8748 是单片微型计算机。它在一个芯片上包括了时钟电路, CPU, RAM, ROM 以及 I/O 等各部分, 构成了一个完整的计算机。

它封装在一个 40 脚的芯片内, 具有以下功能:

8 位的 CPU; 1K $\times$ 8 ROM 程序存储器; 64 $\times$ 8 RAM 数据存储器; 27 根 I/O 线; 8 位定时/计数器。

它的周期时间为 2.5 μ s $\sim$ 5 μ s, 有 90 多条指令, 其中的 70% 是单周期指令。

8048/8748 的结构框图如图 1.1 所示。下面就有关的内部寄存器与标志位, 分别加以详细说明。

1.1 运算部件

运算部件分成如下模块

运算器: ALU 在指令译码器的控制下, 接收一个或二个八位数据, 产生八位结果, 它具有:

1. 带进位或不带进位的加法
2. 与, 或, 异或
3. 加 1 或减 1
4. 按位取反
5. 左移或右移(带或不带进位)
6. 半字交换
7. BCD 码十进制位调整。

如果 ALU 执行的操作产生八位以上的结果(最高位溢出),则在程序状态字 (PSW) 中进位标志置 1。

ALU 的核心是累加器 A。它是 ALU 的一个输入源,经常又是操作结果目的地。ALU 与 I/O 口与内存贮器的相互传送一般也通过累加器 A。

1.2 程序存贮器

内部程序存贮器为 1024(8048)或 2048(8049)字节。8035/8039 内部无程序存贮器,必须使用外部器件。程序存贮器中,有三个单元特别重要。

1. 0 号单元

处理机复位 (RESET) 时,计算机总是从 0 号单元开始执行指令。

2. 3 号单元

在允许外部中断时,当中断输入线上有中断信号时计算机总是转到 3 号单元执行中断服务程序。

3. 7 号单元

当允许 定时/计数器 中断时,若定时器计数器溢出,则产生 定时/计数器 中断,计算机转到 7 号单元执行 定时/计数器 中断服务程序。

在使用时,0 号单元应该放一条转移指令,转到用户主程序;3 号单元放转移指令,使有中断时,转到外部中断服务子程序上;7 号单元的转移指令转到 定时/计数器 中断服务子程序上。

1.3 数据存贮器

内部数据存贮器有 64(8048,8035)或 128(8049,8039)字节。

64×8 RAM 有三个地址范围 (0~7, 8~23, 24~31), 有专门的使用方法。

当初始通电时,状态寄存器中的寄存器开关 BS 为 0。BS=0, 则 0~7 就作为直接可寻址的高速暂存寄存器 $R_0 \sim R_7$ 。这样,指令 MOV A, R_7 即把 RAM 中的第 7 号单元内容送到累加器 A。

可以使用寄存器区开关指令使 BS=1。当 BS=1 时,则把另 8 个单元 24~31 定义为 $R_0 \sim R_7$, 此时 MOV A, R_7 即把内存 31 号单元内容送到累加器 A。8~23 号单元既可作一般的存储器,也可作堆栈使用,在调用子程序时用作记迹。可容纳 8 个返回地址,故子程序的嵌套最大可以 8 级,除了 0~7, 24~31 可以直接寻址外,其它的内存单元的寻址只能通过 R_0 或 R_1 间接寻址。

1.4 输入/输出

有 27 根可用作输入或输出的线。它们是三个 8 位口, P1 口、P2 口、BUS 口。P1、P2 具有相同的特性,写到这些口的数据被锁存起来。输入时无锁存。

P1, P2 为准双向口,允许每一根线作为输入、输出或同时作输入、输出。由于它的内部特殊结构,在输入时必须先写一个“1”,否则输入要出错。当不是同时作输入/输出时,无此要求。

P1, P2, BUS 口还可以和立即数进行与/或操作。BUS 口是真正的双向口,输入/输出不能混用。

另有三个输入/输出线是: T_0 , T_1 , $\overline{INT}$ 。能用条件转移指令来测试。

此外, T_0 可作时钟输出; T_1 作为计数输入; $\overline{INT}$ 作中断输入。

1.5 程序状态字

程序状态字 (PSW) 是一个 8 位状态字。当出现子程序调用或中断时, PSW 高四位和程序计数器 PC 一起被存入堆栈中(内存 8~23 号单元),并能用 RETR 指令来恢复。

PSW 各位定义如下:

位 0~2 栈指针,初态为 000,指向 RAM8 和 RAM9 号单元;
第一个子程序调用或中断时把程序计数器的内容和
PSW 的高四位存入栈中;然后栈指针加 1,指向 10
和 11 号单元。

子程序末尾应为返回指令,可使栈指针减 1。

位 3 不使用。

位 4 工作寄存器区开关 (BS)。0=区 0,
1=区 1。

位 5 标志 F₀。它是用户控制的标志位。

位 6 辅助进位 (AC)。由 ADD 指令产生,在十进制位调整
指令 DA A 时使用。

位 7 进位位 (CY)。指示前面操作时累加器 A 溢出情况。

1.6 中 断

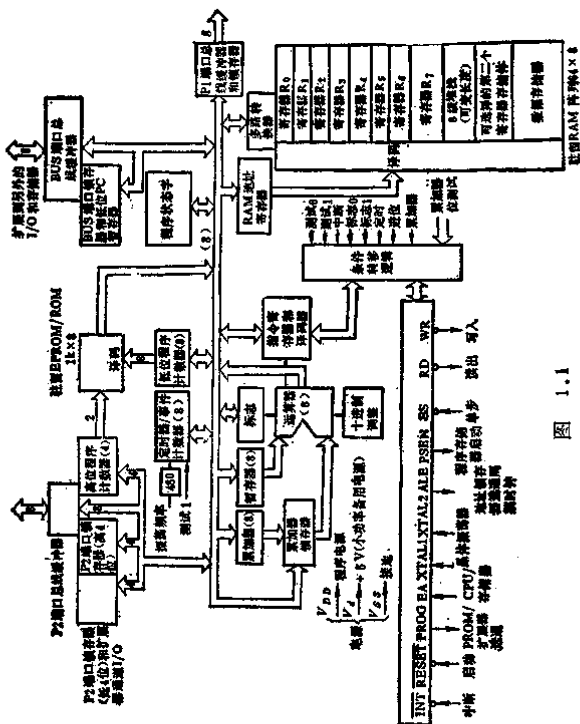
可有两种中断,在 $\overline{\text{INT}}$ 上加一个低电平输入,可产生中断
(在允许中断时),程序转到 003 号单元。当 定时/计数器 发生溢
出时,也可产生中断,程序转到 007 号单元。

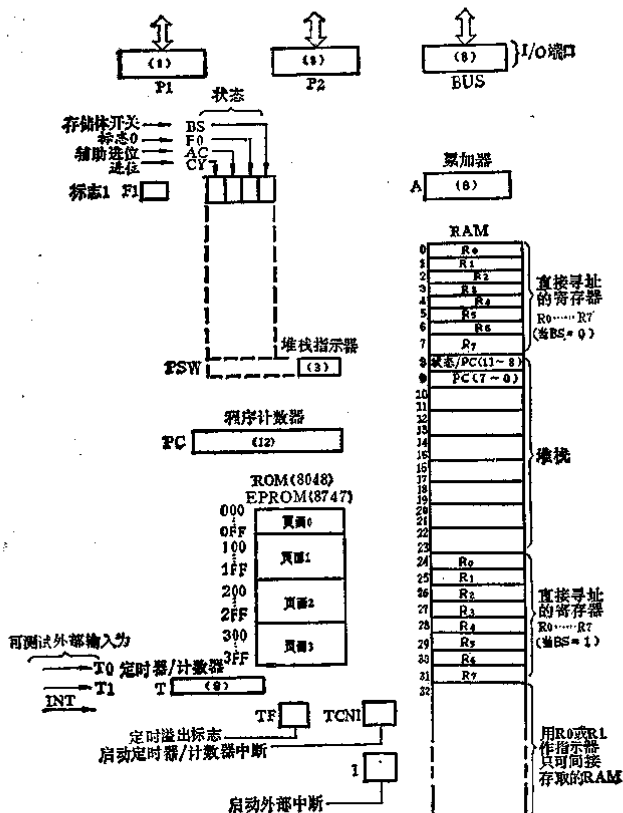
中断系统为单级中断,当检测到一个中断后,将不再理睬其他
中断;只有执行 RETR 指令后,才重新允许中断输入。当中断系
统检测到外部中断和 定时/计数器 中断同时存在时,则先响应外
部中断。

1.7 复位与单步操作

• 复位时具有以下功能

1. 清 0 程序计数器,程序从 0 号单元开始执行。
2. 清 0 栈指针。
3. 清 0 寄存器区开关 BS=0。





Intel 8048/8748 程序设计模型

图 1.2

4. 清 0 存储器区开关 DBF=0.
5. 关中断(定时和外部中断).
6. P1 口, P2 口为输入方式.
7. 清 0 F₀, F₁ 标志位.
8. 不允许 CLK 从 T0 输出.
9. BUS 口为高阻状态.
10. 停止计时.
11. 清 0 定时器标志.

单步功能给用户调试手段, $\overline{SS}$ 上低电平时, 要求停止处理机运行, 当 $\overline{SS}$ 升为高电平时, 退出停止方式取下一条指令。(见图 1.1, 图 1.2)

1.8 MCS-48 处理机引脚功能

MCS-48 微处理机(除 8021 外)均为 40 脚双列直杆封装(DIP), 下表给出每一个脚的功能, 除非特别声明, 每个输入端均与 TTL 电平相容, 每个输出端均可驱动一个标准 TTL 负载.

符 号	引 脚 号	功 能
V _{SS}	20	地电位。
V _{DD}	26	编程电源, 在编写 EPROM 时为 +25V, 在 ROM 和 PROM 读出时为 +5V. 在 ROM 形式 8048 中(包括 8035L)为掉电备用电源脚。
PROG	25	8748 编程时的编程脉冲 (+25V) 输入脚. 8243 I/O 扩展器的输出选通。
V _{CC}	40	主电源, 在操作和 8748 编程时均为 +5V。