

MCS-51 宏汇编程序  
用户指南

余梅笙 译

订书号：9800937-03



中国英特尔计算机用户协会

3

## 前 言

本书试图作为一本参考读物，但它也有一些说明性的资料。全书以下面内容组织。

- 第1章——“绪言”，叙述汇编语言编程方法及对8051硬件的概述。
- 第2章——“操作数及表达式”，叙述每一类操作数并讨论绝对的及可以重新定位的表达式。
- 第3章——“指令集”，依字母顺序完整地叙述每一种指令的操作。
- 第4章——“汇编程序伪指令”，叙述如何定义符号及如何使用各种伪指令。
- 第5章——“宏处理语言”，定义并叙述对宏处理语言的用法。
- 第6章——“汇编程序的操作及控制”，叙述如何引用汇编程序及如何控制汇编程序的操作。
- 第7章——“汇编程序输出：出错消息及列表文件格式”，叙述如何解释出错消息及列表文件。

在您对MCS-51任何一种微计算机编程之前，您应当读MCS-51 User's Manual，订书号121517。

### 有关的文献

为帮助您使用这本手册，您应当使自己了解下面的手册：

- MCS-51 Utilities User's Guide，订书号121737（叙述RL51重新定位程序及连接处理程序和LIB51程序库）
- MCS-51 Family of Single-Chip Microcomputers User's Manual，订书号121517
- ISIS-11 User's Guide，订书号9800306
- MCS-51 Macro Assembly Language Pocket Reference，订书号9800935
- MCS-51 Assembler and Utilities Pocket Reference，订书号121817
- ICE-51 In-Circuit Emulator Operating Instructions for ISIS-II Users，订书号9801004
- Universal PROM Programmer User's Manual，订书号9800819
- Universal PROM Programmer Reference Manual，订书号9800133

# 前 言

本手册叙述如何用汇编语言对MCS-51单片微计算机编制程序。也叙述对MCS-51宏汇编程序的操作说明。

术语MCS-51指的是整个单片微计算机系列，它们按相同的设计构造基本处理器。它们使用相同的指令集，但在存储器映象的特殊功能寄存器及片上的ROM与RAM方面稍有不同。例如，8051有128字节RAM；8044为192字节；而8052则有256字节。

本手册通篇将用术语8051代表整个MCS-51系列。要单指8051处理器时将有明确的说明。

## 第1章

### 绪 言

- 什么是汇编程序
- 如何开发一个程序
  - 模块化编程的好处
  - 高效地开发程序
  - 子程序的重复使用
  - 易于调试及修改
- MCS-51模块程序的开发过程
  - 段、模块及程序
  - 程序送入及编辑
  - 汇编
    - 目标文件
    - 列表文件
  - 重新定位及连接
  - 转换为十六进制格式
  - 保留文件的演变记录
- 对一个MCS-51程序的编写, 汇编及调试

### 硬件概述

- 存储器地址
- 数据单位
- 算术的和逻辑的功能
- 通用寄存器
- 堆栈
- 可作符号寻址的寄存器
- 位寻址
- 程序状态字
- 定时器及计数器
- I/O口
- 串行I/O口
- 中断控制
- 复位

## 第2章

### 操作数及表达式

#### 操作数

特殊的汇编程序符号

简接寻址

立即数

数据寻址

位寻址

代码寻址

相对跳转(SJMP)及条件跳转

在块内跳转及调用(AJMP及ACALL)

长跳转及调用(LJMP及LCALL)

类属跳转及调用(JMP及CALL)

#### 汇编时表达式求值

带说明的数字

ASM51数的表示

表达式中的字符串

符号的用法

在表达式中操作符的用法

算术操作符

逻辑操作符

特殊汇编操作符

关系操作符

操作符优先权

在表达式中段类型

可重新定位的表达式求值

简单的可重新定位的表达式

一般的可重新定位的表达式

## 第3章

### 指令集

引言

注释

## 第4章

### 汇编程序伪指令

#### 引言

位置计数器  
符号名称  
语句标号

#### 符号定义

SEGMENT伪指令  
EQU伪指令  
SET伪指令  
BIT伪指令  
DATA伪指令  
XDATA伪指令  
IDATA伪指令  
CODE伪指令

#### 存储器初始化及保留

DS伪指令  
DBIT伪指令  
DB伪指令  
DW伪指令

#### 程序的连接

PUBLIC伪指令  
EXTRN伪指令  
NAME伪指令

#### 汇编程序状态控制

END伪指令  
ORG伪指令

#### 段选择为指令

USING伪指令

## 第5章

### 宏处理语言

#### 引言

#### 宏处理器概述

#### 介绍生成及调用宏

生成简单的宏

带参数的宏

LOCAL 符号清单

#### 宏处理器的内在功能

内在的注释、换码、括号及METACHAR功能

注释功能

换码功能

括号功能

METACHER功能

MPL中的数和表达式

宏SET

EVAL功能

MPL中的逻辑表达式及串比较

控制流功能

IF功能

WHILE功能

REPEAT功能

EXIT功能

串处理内在功能

LEN功能

SUBSTR功能

MATCH功能

控制台I/O内在功能

#### 先进的MPL概念

宏分隔符

隐空白分隔符

标识符分隔符  
文字分隔符  
文字方式与正常方式的比较  
对宏调用求值的算法

## 第6章

### 汇编程序操作及控制

#### 如何引用MCS—51宏汇编程序

#### 汇编程序控制项

## 第7章

### 汇编程序输出 出错提示及列表文件格式

#### 出错提示及恢复

控制台出错提示

I/O出错

ASM51内部的错误

引用行出错

列表文件出错提示

源文件出错提示

宏出错提示

控制项出错提示

特殊的汇编程序出错提示

致命的出错提示

#### 汇编程序列表文件格式

列表文件题头

源清单

宏及INCLUDE文件的格式

符号表

列表文件尾

附录A 汇编语言BNF语法(略)

附录B 指令集总结

附录C 汇编程序伪指令总结



- 附录D 汇编程序控制项总结
- 附录E MPL内在功能
- 附录F 被保留的符号
- 附录G 示范程序
- 附录H 参考表
- 附录J 出错提示
- 附录K 把绝对程序改为可重新定位的程序
- 附录L 目标处理器的选择

## 表

### 表

1-1 寄存器组选择.....	( 11 )
1-2 8051可作符号寻址的硬件寄存器.....	( 12 )
1-3 加电后8051的状态.....	( 16 )
2-1 特殊的汇编程序符号.....	( 19 )
2-2 8051预先定义了了的位地址.....	( 24 )
2-3 汇编语言数的表示法.....	( 26 )
2-4 表示数的例子.....	( 27 )
2-5 数表示方法的解释.....	( 27 )
2-6 8051预先定义了的数据地址.....	( 29 )
2-7 汇编时的算术操作符.....	( 30 )
2-8 汇编时的逻辑操作符.....	( 31 )
2-9 汇编时的特殊操作符.....	( 31 )
2-10 汇编时的关系操作符.....	( 32 )
3-1 所用的缩写及记号.....	( 37 )
6-1 汇编程序控制项.....	( 169 )
B-1 指令集总结(略).....	( 200 )
B-2 十六进制表示的指令操作码(略).....	( 200 )
C-1 汇编程序伪指令.....	( 201 )
D-1 汇编程序控制项.....	( 203 )

## 图

### 图

1-1 汇编程序及连接程序/重新定位程序的输出.....	( 3 )
------------------------------	-------

1-2	MCS-51程序开发过程 .....	( 4 )
1-3	示范程序清单 .....	( 6 )
1-4	8051方块图 .....	( 8 )
1-5	MCS-51代码地址空间及外部的数据 .....	
	地址空间 .....	( 9 )
1-6	MCS-51数据地址空间及位地址空间 .....	( 9 )
1-7	MCS-51数据单位 .....	( 10 )
1-8	程序状态学的位描述 .....	( 13 )
1-9	TCON的位描述 .....	( 14 )
1-10	口3的位描述 .....	( 14 )
1-11	串行口控制的位描述 .....	( 15 )
1-12	允许中断及中断优先级的位描述 .....	( 16 )
2-1	8051硬件寄存器地址区 .....	( 21 )
2-2a	RAM中可作位寻址的字节 .....	( 22 )
2-2b	8051硬件寄存器地址区中可作位寻址的字节 .....	( 22 )
3-1	指令定义的格式 .....	( 36 )
5-1	宏处理器与汇编程序的比较——一个源文件的两种不同的看法.....	( 146 )
7-1	列表文件格式的例子.....	( 194 )
7-2	题头举例.....	( 194 )
7-3	源清单举例.....	( 195 )
7-4	宏列表方式举例.....	( 196 )
7-5	符号表清单举例.....	( 198 )
G-1	可重新定位的示范程序.....	( 218 )
K-1	示范的绝对程序.....	( 242 )

# 第1章

## 绪言

本手册叙述MCS-51宏汇编程序及讲解以汇编语言开发MCS-51系列处理器软件的过程。8051是本手册主要叙述的处理器。

汇编语言程序直接翻译成机器指令，它告诉处理器应当执行什么操作。因此，汇编语言程序员应当既了解微计算机结构又了解汇编语言。本章介绍MCS-51宏汇编程序及其用法的概貌，以及简要叙述8051结构及硬件特性。

### 什么是汇编程序？

汇编程序是一种软件工具——程序——设计来简化写计算机程序的任务。它把符号代码翻译成可以执行的目标代码。这种目标代码从而有可能被编写到一种MCS-51处理器内并执行之。如果您曾经直接用机器可识别的形式，如二进制或十六进制代码穿过计算机程序，您就会尝识到用符号汇编语言编写程序的优越性了。

汇编语言操作代码（助记符）很易记忆（MOV为传送指令，ADD为加指令）。您也能用符号来表示在指令操作数域中被寻址的地址及数值。由于您指定了这些名称，您便能够把它们作为指令的有意义的助记符来使用。例如，如果您的程序要把一个日期作为数据来处置，您可把它指派为符号名DATE（日期）。如果您的程序有一组指令用做一个定时循环（一组重复执行至规定时间的指令）您能把该指令组命名为TIMER\_LOOP（定时器—循环）。

汇编程序有三个组成部分：

机器指令

汇编程序伪指令

汇编程序控制项

机器指令是能被机器执行的机器代码。对机器指令的详细讨论在第3章中提供。

汇编程序伪指令用于定义程序结构及符号和产生不可执行的代码（数据、消息等等）。见第4章有全部汇编伪指令的详尽材料。

汇编程序控制项设置汇编方式及指引汇编流程。第6章内对全部的汇编控制项有全面的介绍。

### 如何开发一个程序

ASM51能让用户以模块方式编程。下面各节解释模块程序开发的概要。

### 模块化编程的好处

有许多程序因太长或太复杂不能写成一个简单的程序。若把程序分成小的功能单位则编程就相当简单了。模块化编程通常比单块的程序更易编写，调试与修改。

采用模块方法编程类似于对复杂电路的硬件设计。把设备或程序从逻辑上分成具有特殊输入与输出值的“黑盒子”。一旦这些单元之间的接口定义后，每一单元的详细设计便能分头进行。

## 高效地开发程序

用模块方法之所以能较快地开发程序是由于小的子程序比大程序更易明瞭、设计及测试。根据模块的输入及输出的定义，程序员能够提供所需的输入并通过检查输出来校验模块的正确性。而后把分开的各模块连接并定位到一个程序模块中。最后，测试全模块。

## 子程序的重复使用

为一个程序所写的程序常常对其他程序有用。模块化编程允许把这些部分的程序保存起来供以后使用。因为这些程序是可以重新定位的，所保存的模块能够连接到满足其输入及输出要求的任何程序。用单一模块编程方法中，这样的程序部分被淹没在程序内从而不便为其他程序所用。

## 易于调试及修改

模块程序通常比单块程序较易调试。由于程序有良好定义了模块接口，可以把问题隔离到特殊的模块内。一旦识别出有毛病的模块，解决这问题就相当简单了。当必须修改一个程序时，模块编程使工作简化。您能有把握地把新的或调试了的各模块连接到已有程序去而不必修改程序的其余部分。

## MCS-51模块程序的开发过程

本节简要讨论使用可重新定位的MCS51汇编程序（ASM51），连接/重新定位程序（RL51）及代码转换程序来开发程序的过程。

## 段、模块及程序

在开始设计阶段，定义了由程序执行的各任务而后把它分割成子程序。这里简要介绍随MCS-51汇编程序及连接/重新定位程序使用的各种子程序。

段是一个分程序代码或数据存储器。段能够是可重新定位的或是绝对的。一个可重新定位的段有一个名称、类型及其别的各属性。具有相同名称，来自不同模块的各个段，被认为是相同段的一部分，并叫做“部分段”。各部分段由RL51把它们结合到各段。绝对段没有名称也不能与其余段结合。

一个模块具有一个或多个段或部分段。模块有用户指派名称。模块的定义决定局部符号的作用域。一个目标文件有一个或多个模块。您能把别的目标文件简单地添到一个文件后面来把多个模块加入该文件（例如，COPY file1, file 2 TO file3）。

把全部输入模块各个绝对的及可重新定位的段汇集起来，这个程序便由单一的绝对模块组成。

## 程序送入及编辑

在完成了这种设计之后，把每一个模块的源程序使用文本编辑程序送入磁盘文件。若在开发过程中检测到错误，文本编辑程序可以用于改正源程序。

## 汇编

汇编程序(ASM51)把源程序翻译成目标代码。汇编程序产生一个目标文件(当至少有一个输入段是可重新定位的段时称为可重新定位的目标文件，否则是绝对的目标文件)，以及表明汇编结果的一个列表文件。(图1-1总结了汇编与连接及重新定位输出文件)。当ASM51的引用中有DEBUG控制项，该目标文件也收到符号表及其他用于该程序作符号调试的调试信息。

**目标文件。**目标文件内有机语言指令及数据，它们可被装入存贮器供执行或翻译。此外，它还有支配其装入过程的控制信息。

汇编程序能以可重新定位的目标代码格式产生各目标文件。不过，如果该模块仅有绝对段而无外部访问，从汇编中形成的该目标文件是绝对的。它可以不必通过RL51来装载。

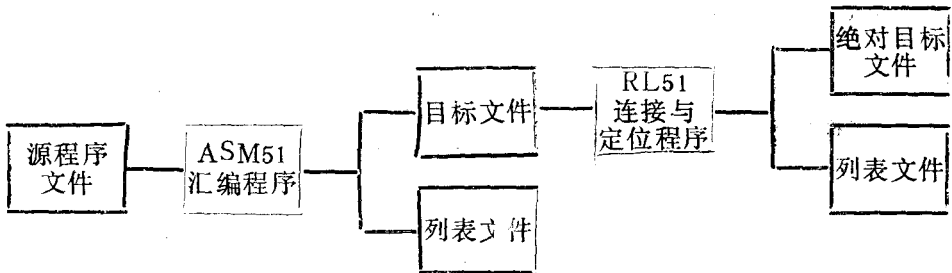


图 1-1 汇编程序与连接/重新定位程序的输出

**列表文件。**列表文件提供的正常记录既有源程序又有目标代码。汇编程序也在该列表文件中提供作语法及其他编码出错的诊断信息。举例说，如果您为一条仅能使用八位值的指令规定一个16位值，汇编程序会告诉您该值超出了许用范围。第7章叙述列表文件的格式。此外，您也能要求把一个符号表添加到列表文件后面。该符号表列出全部符号及其属性。

## 重新定位及连接

在对程序的各个模块汇编之后，用RL51处理各目标模块文件。RL51程序通过把具有相同名称及类型的各段结合起来的方法，把绝对存贮器位置分配给各可重新定位的段。RL51也解决所有模块间的访问。RL51输出一个绝对模块文件，它具有完整的程序，并汇总一个表明连接/定位过程的结果的列表文件。

## 转换为十六进制格式

由RL51产生的绝对目标代码能被放入存贮器并由目标处理器执行而不必作进一步

的修改。不过，某些MCS-51支持的产品(如SDK-51)要求十六进制的目标代码格式。为使用这些产品，绝对目标文件必须用OBJHEX代码转换程序处理。参看ISIS-II System User's Guide (9800306)。

### 保留文件的演变记录

可以方便地用文件名称的扩展名来表明每一个文件内容所代表的处理阶段。因此，源程序文件能用像，.SRC或.A51(表示该程序用于向ASM51输入)那样的扩展名。目标文件由系统设定接收.OBJ扩展名，或者用户也能规定其他的扩展名。可执行的文件通常没有扩展名。列表文件能用.LST，这是由汇编程序给的扩展名。RL51内部设定总结清单文件扩展名为.M51。

用扩展名.TMP要当心，因为许多ISIS-II实用程序所生成的暂存文件都用这个扩展名。如果该文件具有与它们生成的暂存文件相同的名称及扩展名的话，这些实用程序将复盖您的文件。

### 对一个MCS-51程序的编写，汇编及调试

还有一些步骤在您应用MCS-51微计算机时是必须的。图1-2表明的流程图包括准备代码的各个步骤。如果您除了开发软件还开发应用硬件的话，请查阅MCS-51 User's Manual。

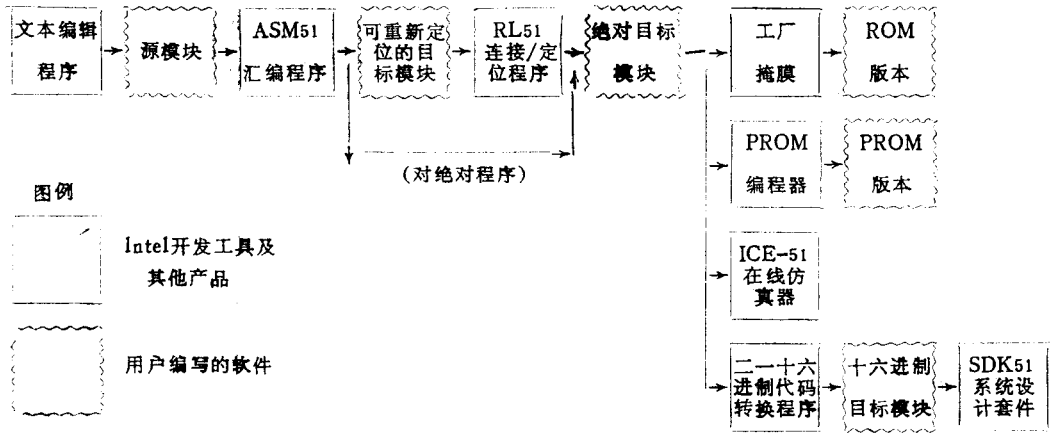


图1-2MCS-51程序开发过程

图1-3有一个示范程序的汇编清单。该程序用下面的命令引用：

```
—ASM51 : F1 : DEMO.A51
```

其输出为：

```
ISIS-II MCS-51 MACRO ASSEMBLER, v2.0
ASSEMBLY COMPLETE, NO ERRORS FOUND
```

图1-3续表示一个示范模块的文件清单，它是没有在此列出的大程序的一部分。在附录H内有一个大程序的例子。

在用ASM51汇编程序之后，下一步是用RL51把全部块模连接为一个程序。RL51

MCS-51 MACRO ASSEMBLER 8051-BASED MONITOR  
 ISIS-II MCS-51 MACRO ASSEMBLER V2.0  
 OBJECT MODULE PLACED IN : F1: DEMO.OBJ  
 ASSEMBLER INVOKED BY: ASM51 : F1 : DEMO.A51  
 LOC OBJ LINE SOURCE

(地址)	(目标码)	(行号)	(源程序)
		1	\$TITLE(8051-BASED MONITOR)
		2	;The main module of an 8051-based monitor
		3	
		4	;symbol definitions
		5	PROG_S SEGMENT COCE
			; Contains the exectutable program
		6	TABLE_S SEGMENT CODE
			; Contain tables and other constant data
000D		7	CR EQU 13
			; Carriage-Return character(ASCII)
000A		8	LF EQU 10 ; Line-feed character (ASCII)
		9	EXTRN(CONSOL_OUT,MONITOR)
			; Defined elsewhere
		10	
		11	;The main program
----		12	CSEG AT 0
			; Skip interrupt vectors if any
0000 020000	F	13	JMP START
		14	
----		15	RSEG PROG_S
		16	START:
0000 900000	F	17	MOV DPTR,#SIGNON
			; Print signon message
0003 120000	F	18	CALL CDNSOL_OUT
0006 020000	F	19	JMP MONITOR
			; Enter the monitoring loop
		20	
----		21	RSEG TABLE_S
0000 1A		22	SIGNON: DB LEN, '8051-BASED MONITOR

```

    ,V1.0',CR,LF
0001 38303531
0005 2D424153
0009 4544204D
000D 4F4E4954
0011 4E522C20
0015 56312E30
0019 0D
001A 0A
001A      23      LEN      EQU      $-SIGNDN-1
           ;Compute message length
           24
           25      END

```

图1-3 示范指序清单

## MCS-51 MACRO ASSEMBLER 8051-BASED MONITOR

### SYMBOL TABLE LISTING

NAME (名称)	TYPE (类型)	VALUE (值)	ATTRIBUTES (属性)
CONSOL-OUT	C ADDR	---	EXT
CR . . . . .	NLMB	000DH	A
LEN . . . . .	NUMB	001AH	A
LF . . . . .	NUMB	000AH	A
MONITOR . .	C ADDR	---	EXT
PROG__S . .	C SEG	0009H	REL=UNIT
SIGNON . .	C ADDR	0000H	R SEG=TABLE__S
START . . .	C ADDR	0000H	R SEG=PROG__S
TACLE__S . .	C SEG	001BH	REL=UNIT

REGISTER BANK(S) USED: 0, TARGET MACHINE(S): 8051

ASSEMBLY COMPLETE, NO ERRORS FOUND

图1-3续, 示范程序清单

产生一个清单文件的总结, 包含存储器映象表及符号表 (参看MCS-51 Utilities User's Guide, 1271737)。

调试您的程序的下一步骤, 是把程序写入EPROM8751并在样机环境下对它测试。(进一步的测试可用ICE-51完成)。要把您的程序编写入8751, 您必须有一个与



开发系统连接的UPP(通用PROM编程机)。要了解如何使用UPP及UPM(通用PROM编程模块)的完整叙述, 请阅Universal PROM Programmer Referece Manual, 订书号9800133及Universal PROM Programmer User's Manual, 订书号9800819。

## 硬件概述

有关MCS-51系列各种芯片的完整的硬件说明, 参看附录L及描述您的处理器的文献。8051的主要特性为:

- 内部4K字节ROM或EPROM程序存贮器(在8031内没有程序存贮器), 可以扩展为64K字节。
- 内部128字节RAM存贮器, 它包括四组八个通用寄存器及一个作子程序及中断例程调用的堆栈。
- 64K字节外部RAM地址空间
- 16位程序计数器提供对64K字节存贮器作直接访问。
- 八位堆栈指针, 它能被置为芯片上RAM的任意地址
- 两个可编程16位定时器/计数器
- 可编程全双工串行I/O口
- 四个八位双向并行I/O口
- 具有两级优先级的定时器及I/O中断
- 具有51种功能(包括存贮器对存贮器传送的111条指令)
- 具有128个软件标志的布尔功能、很多硬件标志及12位操作数指令
- 算术及逻辑单元, 它包括加、减、乘及除算术功能, 还有与、或、异或及求补的逻辑功能。

图4-1是8051处理器的框图。它表明了数据通路及可被程序员访问的主要功能单元。

## 存贮存地址

8051有五种地址空间:

- 程序地址空间——64K, 其中4K在片上(8031没有片上ROM)。
- 可直接寻址的内部数据地址空间——128字节RAM(0-127)及128字节硬件寄存器地址空间(128-255; 仅20个地址可以使用); 可以通过直接寻址访问它们。
- 可简接寻址的内部数据地址空间——128字节(0-127), 全部可作简接寻址访问。
- 外部的数据地址空间——由用户增加的外存贮器多达64K。
- 位地址空间——共享数据地址空间中可访问的地址; 可由直接寻址访问。

三个实际不同的存贮器对应于, 程序地址空间, 内部数据地址空间(包括直接与简接可寻址的空间和位地址空间两), 和外部的数据空间都由不同的机器指令寻址。这是重要的区分, 对明白如何编写8051程序是关键。

在您把一个操作数规定给具有错误的属性的符号时, ASM51产生一个出错消息警告您这种不一致。第2章及第3章表明每一条指令期待的段类型属性而第4章叙述如何