

NEC单片机用户手册

RA78KOS汇编程序包

(汇编语言)

日本电气(NEC)公司 编
张芳兰 杨华 孟力 译



电子工业出版社

PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
URL: <http://www.phei.co.cn>

NEC 单片机用户手册

RA78K0S 汇编程序包

(汇编语言)

日本电气 (NEC) 公司 编

张芳兰 杨 华 孟 力 译



电子工业出版社

Publishing House of Electronics Industry

内 容 简 介

本书对 RA78K0S 汇编程序包中各程序的基本功能与源程序的描述方法作了详细地介绍。其目的：使用户对 RA78K0S 汇编程序包中各程序的基本功能与源程序的描述方法有一个正确理解。

全书共分六章，包括：概述、怎样编写源程序、伪指令、控制指令、宏、产品的应用。另外，还有附录，给出了保留字表、伪指令表和最高性能。

本书可供从事小型通用微控制器软件开发的工程技术人员参考，也可作为专业培训人员的教材。

丛 书 名：NEC 单片机用户手册

书 名：RA78K0S 汇编程序包(汇编语言)

著 者：日本电气(NEC)公司

译 者：张芳兰 杨 华 孟 力

责任编辑：郭延龄

特约编辑：方 文

印 刷 者：北京科技印刷厂印刷

出版发行：电子工业出版社出版、发行 URL: <http://www.phei.co.cn>

北京市海淀区万寿路 173 信箱 邮编 100036 发行部电话：68214070

经 销：各地新华书店经销

开 本：787×1092 1/16 印张：8 字数：202 千字

版 次：1997 年 10 月第一版 1997 年 10 月第一次印刷

书 号：ISBN 7-5053-4341-6
TN·1103

定 价：45.30 元

凡购买电子工业出版社的图书，如有缺页、倒页、脱页者，本社发行部负责调换

版权所有·翻印必究

前 言

本手册的目的在于使用户对 RA78K0S 汇编程序包（简称为 RA78K0S）中各程序的基本功能与源程序的描述方法有一个正确的理解。

本手册并不介绍 RA78K0S 各程序的操作过程，因此当需要这方面内容时，在学完该手册后，应阅读 RA78K0S 汇编程序包用户手册的操作部分。

〔适用的设备〕

RA78K0S 能够用于 78K/0 系列小型通用微控制器的软件开发。为适应该设备（器件）的程序编写，需要相应于此设备类型的设备文件（可单独应用）。为找到设备类型，可查阅其设备文件中的产品说明。

〔手册的组织结构〕

本手册包括下列 6 章和附录：

第一章 概述

概要介绍 RA78K0S 的基本功能。

第二章 怎样编写源程序

介绍如何描述源程序，说明汇编程序的操作符。

第三章 伪指令

介绍编写和利用伪指令的方法及其实例。

第四章 控制指令

介绍编写和利用控制指令的方法及其实例。

第五章 宏

介绍所有宏的功能，包括宏的定义、引用与扩展。

有关宏伪指令的内容，在第三章中说明。

第六章 产品的应用

介绍几种描述源程序建议的度量方法。

附 录 包括保留字表、伪指令表和 RA78K0S 的最高性能。

在本手册中没有详细介绍指令集，若需了解可参阅开发软件所用微控制器的用户手册。有关结构的说明，可参阅相应微控制器的用户手册（硬件版本）。

〔使用手册的建议〕

对于第一次使用汇编程序的读者，应从手册的第一章着手学习。若已有了汇编程序的一般知识，则可跳过第一章，但无论如何也应阅读第一章的 1.2 节和第二章的内容。

对于希望了解汇编程序伪指令和控制指令方面内容的读者，应分别学习第三章与第四章，在这两章中，详细介绍了每种伪指令或控制指令的格式、功能、使用和应用实例。

〔图标符号〕

本手册中使用下列符号和缩写代号：

[]：表示方括号内的参数可以省略。

{ }：表示 { } 内的字符可以选择之一。

' '：表示 ' '（单引号）内为字符。

△：表示一个空格或 TAB。

⋮：表示这部分程序可以省略。

" "：指示引用处。

⋯：表示这部分程序可以省略。

插 图 清 单

图 号	标 题	页 码
1—1	RA78K0S 汇编程序包	1
1—2	汇编程序的流程	2
1—3	利用微控制器的产品的开发过程	2
1—4	再汇编过程	3
1—5	利用已有模块开发新程序	4
2—1	源模块的组成	6
2—2	源模块的总体结构	7
2—3	源模块结构示例	8
2—4	范例程序的结构	8
2—5	组成语句的字段	11
3—1	段的存储器分配	45
3—2	代码段的再定位	46
3—3	数据段的定位	48
3—4	位段的再定位	51
3—5	绝对段的定位	54
3—6	两个模块间符号的关系	64

表 格 清 单

表 号	标 题	页 码
1—1	汇编程序的最高性能	4
1—2	连接程序的最高性能	5
2—1	模块头中可以使用的指令	7
2—2	符号的类型	14
2—3	符号的属性与值	14
2—4	由汇编程序自动产生的段名	16
2—5	数值常数的表示方法	17
2—6	可以在操作数字段描述的专用字符	18
2—7	运算符的分类	21
2—8	运算符的优先次序	22
2—9	再定位属性的类型	32
2—10	项与运算符按再定位属性的组合 (二元运算符)	32
2—11	项与运算符按再定位属性的组合 (一元运算符)	33
2—12	项与运算符按再定位属性的组合 (专用运算符)	34
2—13	项与运算符按再定位属性的组合 (专用一元运算符)	34
2—14	运算中符号属性的类型	34
2—15	项与运算符按符号属性的组合 (二元运算符)	35
2—16	项与运算符按符号属性的组合 (一元运算符)	35
2—17	项与运算符按符号属性的组合 (专用运算符)	36
2—18	项与运算符按符号属性的组合 (专用一元运算符)	36
2—19	第 1 项 (X) 与第 2 项 (Y) 的组合	37
2—20	第 1 和第 2 项按再定位属性的组合	38
2—21	比特符号的值	38
2—22	指令操作数值的范围	39
2—23	伪指令操作数值的范围	40
2—24	可作为操作数的符号的特征	41
2—25	可作为伪指令操作数描述的符号的特征	41

(续表)

表 号	标 题	页 码
3—1	伪指令表	43
3—2	段的定义方法及存储器地址分配	44
3—3	CSEG 的再定位属性	46
3—4	CSEG 的缺省段名	46
3—5	DSEG 的再定位属性	48
3—6	DSEG 的缺省段名	49
3—7	BSEG 的再定位属性	51
3—8	BSEG 的缺省段名	52
3—9	表示比特值的操作数的表达格式	56
4—1	控制指令表	82
4—2	控制指令与汇编程序选项	83

目 录

第一章 概述	(1)
1.1 汇编程序概述	(1)
1.1.1 什么是汇编程序	(1)
1.1.2 什么是可再定位汇编程序	(2)
1.2 程序开发前的注意事项	(3)
1.2.1 RA78K0S 的最高性能	(3)
1.3 RA78K0S 的特性	(5)
第二章 怎样编写源程序	(6)
2.1 源程序的基本组成	(6)
2.1.1 模块头	(6)
2.1.2 模块体	(6)
2.1.3 模块尾	(7)
2.1.4 源程序的总体结构	(7)
2.1.5 源程序举例	(7)
2.2 源程序的格式	(11)
2.2.1 语句的结构	(11)
2.2.2 字符集	(11)
2.2.3 组成语句的字段	(13)
2.3 表达式和操作数	(21)
2.3.1 运算符的功能	(22)
2.3.2 对运算的一些限制	(32)
2.4 比特位置分类符	(37)
2.5 操作数的特性	(38)
2.5.1 操作数值的长度和地址范围	(39)
2.5.2 用于指令的操作数长度	(39)
2.5.3 操作数的符号属性和再定位属性	(40)
第三章 伪指令	(43)
3.1 伪指令概述	(43)
3.2 段定义伪指令	(43)
3.3 符号定义伪指令	(55)
3.4 存储器初始化及保留存储区伪指令	(58)
3.5 连接伪指令	(64)
3.6 目标模块名说明伪指令	(68)
3.7 分支指令自动选择伪指令	(69)
3.8 宏伪指令	(70)
3.9 汇编终止伪指令	(80)
第四章 控制指令	(82)
4.1 控制指令概述	(82)
4.2 处理器类型说明控制指令	(83)

4.3	符号控制指令	(84)
4.4	调试信息输出控制指令	(85)
4.5	交叉引用列表输出说明控制指令	(87)
4.6	包含控制指令	(88)
4.7	汇编列表控制指令	(90)
4.8	条件汇编控制指令	(100)
4.9	汉字代码控制指令	(105)
4.10	其它控制指令	(106)
第五章	宏	(107)
5.1	关于宏的概述	(107)
5.2	宏的应用	(108)
5.2.1	宏定义	(108)
5.2.2	宏引用	(108)
5.2.3	宏展开	(109)
5.3	宏内的符号	(109)
5.4	宏运算符	(111)
第六章	产品应用	(113)
附录 A	保留字表	(114)
A.1	保留字表	(114)
附录 B	伪指令表	(115)
附录 C	最高性能	(116)

第一章 概 述

本章介绍 RA78K0S 在微控制器软件开发中的作用及其特性。

1.1 汇编程序概述

RA78K0S 汇编程序包是一组程序的总称,这些程序把以 78K/0 系列小型通用微控制器的汇编语言编写的源程序转换成机器语言代码。

RA78K0S 包括 6 个程序:结构化的汇编预处理程序、汇编程序、连接程序、目标码转换程序、库管理程序和表转换程序。

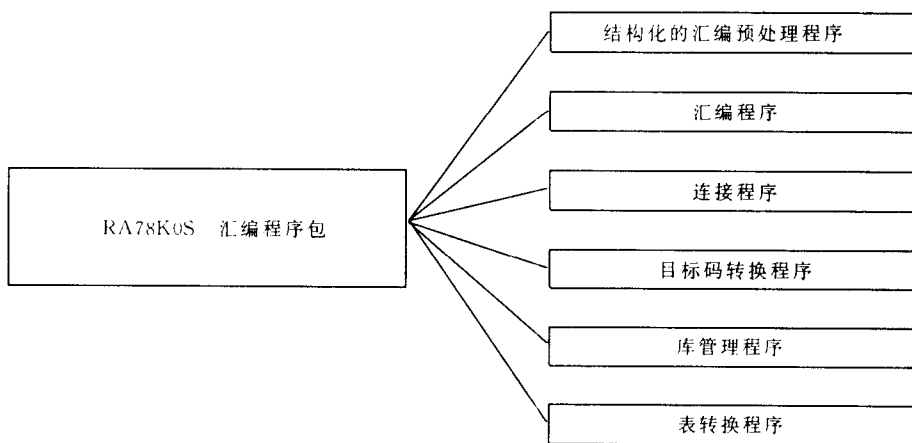


图 1-1 RA78K0S 汇编程序包

1.1.1 什么是汇编程序?

(1) 汇编语言与机器语言

汇编语言是微控制器最基本的编程语言。

程序和数据是微控制器中的微处理器正常工作所必需的。这些程序和数据必须由人(程序员)编写并存在微控制器的存储器中。微控制器所处理的程序和数据是二进制数的集合,称为机器语言。然而,对于程序员来说,机器语言代码难记,经常产生错误。幸好,可以用英文缩写或助记符来表示机器语言代码的意义,这样便于人的记忆与理解。使用这种符号编码的编程语言系统叫作汇编语言。

由于微控制器处理的必须是机器语言的程序,因此还需要有另外的程序,把用汇编语言建立起来的程序转换成机器语言,这种程序就称为汇编程序。

(2) 利用微控制器的产品的开发及 RA78K0S 的作用

图 1-3 表示了汇编语言编程在(软件)产品开发过程中所处的位置。



图 1-2 汇编程序的流程

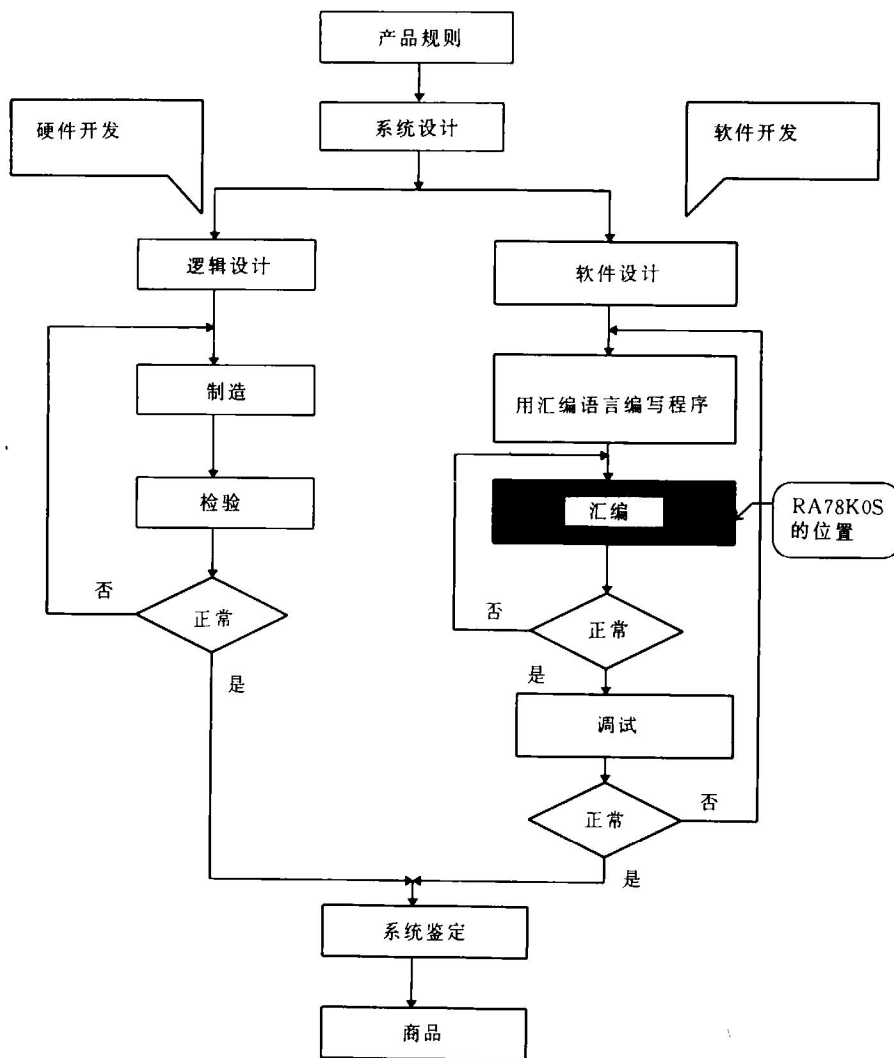


图 1-3 利用微控制器的产品的开发过程

1.1.2 什么是可再定位汇编程序？

利用汇编程序从源程序转换来的机器语言在使用前存在微控制器的存储器内。为此，每条机器语言指令在存储器中存放的位置必须是已确定了的。所以汇编程序产生的机器语言中需

要有附加信息,来说明每条机器语言指令存储的位置。

根据对机器语言指令分配地址的不同方法,可粗略地将汇编程序分为绝对汇编程序与可再定位汇编程序。

- 绝对汇编程序

绝对汇编程序把汇编语言产生的机器语言指令存放到绝对地址中。

- 可再定位汇编程序

在可再定位汇编程序中,为汇编语言产生的机器语言的指令所确定的地址是暂时的,其绝对地址要由此后的连接程序确定。

当利用绝对汇编程序编制程序时,原则上需一次完成整个编程。然而,要一次完成一个大程序,可能很复杂,程序的分析和维护也很困难。为此,可将这种大程序分成几个子程序,称之为模块,每个模块对应于一个功能单元。这种编程技术称作为模块化编程。

可再定位汇编程序适合于模块化编程。使用可再定位汇编程序进行模块化编程有以下优点:

(1)提高开发效率

要一次编制一个大程序是困难的。在这种情况下,可将它划分成若干功能模块,让多人同时编制各子程序,从而提高开发效率。

若程序出现错误,也不必重新汇编整个程序,只需重新汇编需要改正的模块,这就缩短了汇编时间。

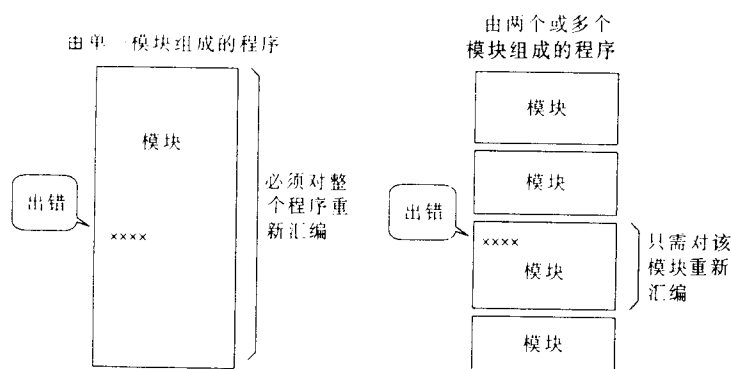


图 1-4 再汇编过程

(2)有效利用资源

可以把以前开发的高可靠性、高通用性的程序模块用到其它程序中去。积累一些这样的高通用性模块,可以减少新程序开发的时间和工作量。

1.2 程序开发前的注意事项

在着手开发程序之前,应考虑以下各点。

1.2.1 RA78K0S 的最高性能

(1)汇编程序的最高性能

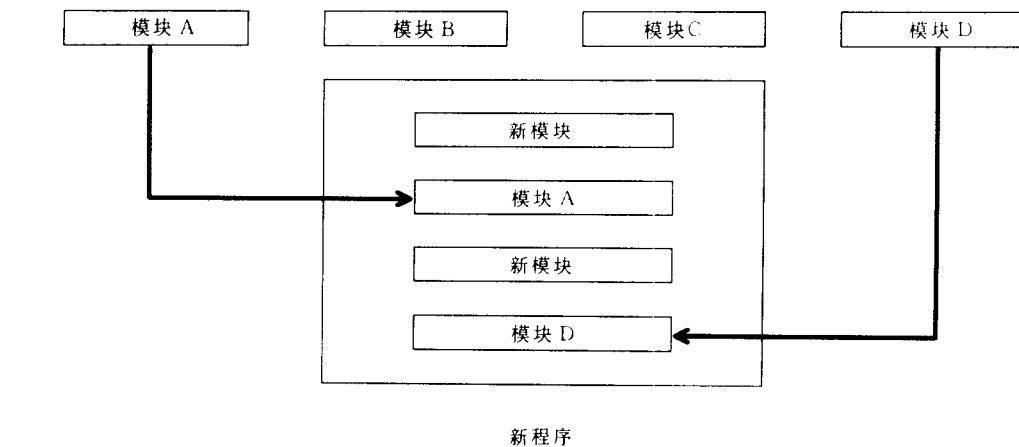


图 1-5 利用已有模块开发新程序

表 1-1 汇编程序的最大性能

项 目		最 高 性 能	
		PC 版	WS 版
符号数		无限制(注 1)	无限制(注 2)
交叉参考列表能输出的符号数		6000 符号	32000 符号
一次宏引用的宏体大小		32K 字节	32K 字节
一个文件中可以描述的宏		65535(注 3)	450K 字节—α(注 4)
一个文件中的段数		100	100
一个文件中可以引用的宏和可以指定的包含文件的总数		300	10000
一个包含文件中可以引用的宏和可以指定的包含文件的总数		300	10000
可再定位数据(注 5)		65535 项	65535 项
每行字符数		218 字符(注 6)	218 字符(注 6)
符 号 长 度	带 — NS 选项	8 字符	8 字符
	不带 — NS 选项	31 字符	31 字符

注:1. 使用 XMS,若不用 XMS,可用文件。

2. 使用存储器。若不用存储器,可用文件。

3. 因为宏控制表和自变量数据保存在常规内存中,所以常规内存的上限会依宏的数目变化。

4. “α”是用于宏控制表和自变量数据的存储区的大小。

5. “再定位数据”是当汇编程序不能确定符号的值时传送给连接程序的数据。

例如,当 MOV 指令引用外部引用符号时,在 .rel 文件中产生两项再定位数据。

6. 这里不包括回车和换行符。如果一行中有 219 个或更多的字符,则输出警告信息,并把多余的字符忽略。

(2)连接程序的最高性能

表 1-2 连接程序的最高性能

项 目	最 高 性 能	
	PC 版	WS 版
LOCAL 符号数	4676 符号×模块数	无限制
PUBLIC 符号数	4680	无限制
段 数	(注)	无限制
输入模块数	256	1024
输入库数	50	128

注:该值因常规内存不同而变化。如以下各例。

- 当所有段不同名时
 - 常规内存=420K 字节 482 段
 - 常规内存=450K 字节 556 段
 - 常规内存=500K 字节 630 段
- 当段名为 100 种类型时
 - 常规内存=420K 字节 1005 段
 - 常规内存=450K 字节 1179 段
 - 常规内存=500K 字节 1443 段

1.3 RA78K0S 的特性

RA78K0S 具有以下特性:

(1)宏功能

当源程序中多次重复使用同一组指令时,可为这组指令指定一个宏名,从而将其定义为宏。借助这种宏功能,可提高编程效率并增强程序的可读性。

(2)分支指令的优化功能

RA78K0S 有一条能自动选择分支指令的伪指令(即 BR 伪指令)。

为使程序能高效率地使用存储器,需要根据分支指令分支的目的范围选用字节分支指令。但对每次分支都要花精力去考虑分支的目的范围,从而选择分支指令,就会给程序员带来很大麻烦。利用 BR 伪指令,汇编程序能根据分支的目的范围产生合适的分支指令。这就是对分支指令的优化功能。

(3)条件汇编功能

利用这一功能,可根据预先指定的条件确定是否应对源程序的某部分进行汇编。如果源程序中有调试语句,那么,通过设置条件汇编的开关可以选择是否将该调试语句转换成机器语言。当不再需要调试语句时,无需对程序作大的修改即可对源程序进行汇编。

(4)选择通用寄存器的伪指令

可以用绝对名(R0、R1、PRO 等)或功能名(X、A、AX 等)表示通用寄存器。

第二章 怎样编写源程序

本章介绍源程序的编写方法、格式及源程序中的表达式与操作数。

2.1 源程序的基本组成

当把一个源程序分成几个模块时,则每个模块都是输入到汇编程序的一个单元,这种模块称为源模块(如果源程序只含一个模块,那么“源程序”与“源模块”意义相同)。

作为输入到汇编程序的一个单元的源模块主要由以下三部分组成:

- 〈1〉模块头
- 〈2〉模块体
- 〈3〉模块尾

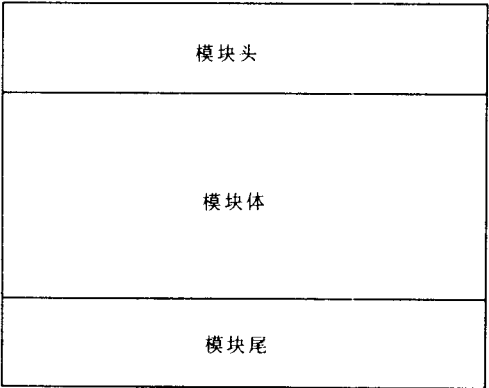


图 2-1 源模块的组成

2.1.1 模块头

在模块头中,可以包括表 2-1 中说明的控制指令。这些控制指令只能在模块头中使用。另外模块头可以省略。

2.1.2 模块体

在模块体中,不能使用以下指令:

- 与汇编程序选项具有同样功能的控制指令

其它所有伪指令、控制指令和机器指令都可以在模块体中使用。

模块体必须分成一些单元来描述,这些单元称作“段”。

用户可使用相应的伪指令定义以下四种段:

- 〈1〉代码段 必须用 CSEG 伪指令定义
- 〈2〉数据段 必须用 DSEG 伪指令定义

表 2-1 模块头中可以使用的指令

可使用的项目	说 明	在本手册中的章节
与汇编程序选项具有同样功能的控制指令	与汇编程序选项具有同样功能的控制指令包括： PROCESSOR, XREF/NOXREF, DEBUG/NODEBUG/ DEBUGA/NODEBUGA, TITLE, SYML EN/NOSYM LEN, CAP/NOCAP, SYMLIST/NOSYMLIST, FORM- FEED/NOFORMFEED, WIDTH, LENGTH, TAB, KANJI CODE	见“第四章控制指令”。
专用控制指令, 这是由 C 编译程序和结构化汇编预处理程序等高层程序输出的控制指令。	由 C 编译程序和结构化汇编预处理程序等高层程序输出的专用控制指令包括： TOL _INF, DGS, DGL	

〈3〉位段 必须用 BSEG 伪指令定义

〈4〉绝对段 利用 CSEG、DSEG 或 BSEG 伪指令且必须指定单元地址为再定位属性(AT 单元地址), 来定义绝对段。这种段也可用 ORG 伪指令定义。

模块体可由各段任意组合而成。但是数据段和位段应在代码段之前定义。

2.1.3 模块尾

模块尾表示源模块结束。这部分必须有 END 伪指令。

在 END 伪指令后, 除注释、空格、列表符或换行符外, 若有其它字符, 汇编程序将输出警告信息, 并忽略 END 伪指令后的字符。

2.1.4 源程序的总体结构

源模块(源程序)的总体结构如下所示。

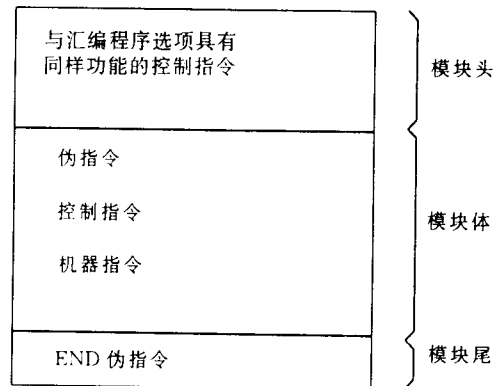


图 2-2 源模块的总体结构

一个简单源模块结构的例子示于图 2-3。

2.1.5 源程序举例

本节给出 78K/0 系列低档通用微控制器的一个源模块作为范例程序。这个程序的结构示于图 2-4。