

ECLIPSE MV系列机资料

AOS/VS 宏汇编

参 考 手 册

苏州电子计算机厂情报室

ECLIPSE MV 系列机资料译丛

AOS/VS宏汇编
参 考 手 册

陕西师大 计算机教研室 曹豫莪审校

ECLIPSE MV 系列机资料译丛编辑部

此文译自Data General公司资料

AOS/VS

Macroassembler

(MASM)

Reference Manual

093—000242—00

出版说明

ECLIPSE MV系列计算机是美国DG公司1980年4月宣布的32位超级小型机,时至今日先后完成生产和投放市场的有ECLIPSE MV/4000, MV/6000, MV/8000和MV/10000等4种机型。该系列无论在软件和硬件方面都有其特点,并在继续向前发展。

为了满足ECLIPSE MV系列机用户的需要,我们组织力量对MV系列机的资料进行了翻译,并取名为《ECLIPSE MV系列机资料译丛》,作为内部资料发行。

资料所涉及的范围十分广泛,说明清晰和具体。第一批资料共计30余种。本译丛并将追踪该系列的发展、新资料的发行,陆续翻译并发行。

第一批资料包括:

- (1) MV系列机的操作原理,并按机型有单行本的说明。
- (2) 先进的带虚拟存贮的操作系统(AOS/VS)。它包括操作系统的生成、运行和程序人员手册等。
- (3) 各种高级语言。包括各种常用的高级语言的新品种和新版本,如FORTRAN77、COBOL、PASCAL和PL/I以及其他一些常用高级语言。
- (4) 系统检查、诊断和维护手册。包括现场维修手册、系统练习程序及可靠性检查有关资料。

(5) 其他,包括各种服务性程序及用户手册等。

《ECLIPSE MV系列机译丛》不仅是ECLIPSE MV系列机的系统管理人员、系统操作人员程序设计人员和用户所必须阅读资料,而且对于从事计算机工作的科技人员和有关专业的大专院校师生都有很好的参考价值。

参加编译的单位有：计算机管理总局，清华大学，西南师范学院，成都科技大学，山东海洋学院，华中师范学院，陕西师大，东北师大，北京第七建筑设计院，上海冶金所，苏州电子计算机厂等。

编辑部成员：

总 编：房家国（清华大学）

副总编：李宏信（成都科技大学）

邱玉辉（西南师范学院）

编 委：吴 奇（计算机管理局）

李邦几（华中师院）

曹豫我（陕西师大）

刘长欢（东北师大）

徐 斯（山东海洋学院）

姚林声（上海冶金所）

责任编辑：金传祚（苏州电子计算机厂）

限于译者水平，加之时间紧促，缺点错误在所难免，敬请读者及时提出批评指正。

ECLIPSE MV系列机资料译丛编辑部

译 者 的 话

这本手册是介绍 Advanced Operating System/Virtual Storage (AOS/VS) 宏汇编实用程序的使用和操作。而且假使读者具有使用汇编语言进行程序设计的经验, 又熟悉 ECLIPSE MV/8000。

这本《AOS/VS 宏汇编程序参考手册》是根据 Data General 公司的《Advanced Operating System/Virtual Storage (AOS/VS) Macroassembler (MASM) Reference Manual》093-000242-00 翻译的。

这本书对MV/8000宏汇编程序的指令, 功能, 使用等各方面进行了全面地论述, 对使用宏汇编进行程序设计的科技人员确实是一本比较详尽而有用的工具书。也可以作为高等学校电子计算机专业的参考书。

在翻译过程中, 曾得到清华大学房家国教授的很多指导, 也得到过成都科技大学, 西南师范学院, 山东海洋学院, 华中师范学院, 东北师范大学等院校有关同志的帮助, 对此表示感谢!

参加翻译的同志有: 曹豫莪、张步云、唐家益、冯德民、王四万、师军。曹豫莪同志对全书进行了校审。另外, 杨先桂、张玉琴同志作了许多辅助工作。

由于我们的水平有限, 漏误之处可能难免, 请读者不吝指教。

1983·7 陕西师大

附录A ASCLL字符集

附录B MASM伪指令一览表

附录C 汇编错误和命令行错误表及其说明

在本手册中, 我们讲解许多概念, 这些概念更进一步的内容在 Data General 其他一些出版物中讲述。在一些实例中, 关于进一步信息你必须参考下列文件之一。

- AOS/VS 调试程序和磁盘文件编辑用户手册(093-000246), 该手册讲述调试实用程序。调试程序在运行期间对用户程序进行检查。使用这个实用程序来查明用户程序中的错误。
- AOS/VS 连接和库文件编辑用户手册(093-000245), 本手册进一步讲述连接实用程序。汇编用户程序之后, 还必须再用连接实用程序产生一个可执行的程序文件。
- AOS/VS 程序员手册(093-000241), 本手册记录着 AOS/VS 系统调用和系统参数。系统调用是执行预定的通常使用的操作。
- ECLIPSE MV/8000操作原理(014-000648)它记录着 MV/8000全部汇编语言指令。这本资料还描述ECLIPSE MV/8000硬件方面的概念。

请读者注意:

在这本手册中, 我们采用的命令形式是: COMMAND ☐required (☐optional)

前 言

这本手册是介绍 Advanced Operating System/Virtual Storage (AOS/VS) 宏汇编实用程序的使用和操作。而且假设读者具有使用汇编语言进行程序设计的经验, 又熟悉 ECLIPSE MV/8000 汇编语言指令。

我们是从参考工具这个角度出发写这本手册的。下面概述各章的内容和附录:

- 第1章 AOS/VS 宏汇编程序导引。解释宏汇编程序的用途, 重点介绍它的特性。还说明 MASM 的简单使用。
- 第2章 描述对宏汇编程序的输入。这一章插入了3个不同部分: 语句成分(如: 数字, 符号, 表达式), 语句格式(如: 标号, 注释), 和语句类型(如: 指令, 伪指令, 赋值)。
- 第3章 解释 MASM 是怎样汇编用户程序的。要正确地使用 MASM, 也未必要通读本章, 然后, 你必须复习存储区和再定位部分。
- 第4章 描述 MASM 在汇编期间所产生的各种输出形式, 即, 目标文件, 汇编清单, 互相对照表和错误清单。
- 第5章 描述宏功能。这一章还解释怎样使用生成数字和符号。
- 第6章 描述在用户程序中可以使用的各种伪指令类型。
- 第7章 提供按字母顺序排列的伪指令的详细说明。
- 第8章 解释调用宏汇编程序的命令行。这一章还说明怎样连接和运行用户程序, 以及怎样使用永久符号表。

附录A ASCLL字符集

附录B MASM伪指令一览表

附录C 汇编错误和命令行错误表及其说明

在本手册中, 我们讲解许多概念, 这些概念更进一步的内容在 Data General 其他一些出版物中详述。在一些实例中, 关于进一步信息你必须参考下列文件之一。

- AOS/VS 调试程序和磁盘文件编辑用户手册(093—000246), 该手册讲述调试实用程序。调试程序在运行期间对用户程序进行检查。使用这个实用程序来查明用户程序中的错误。
- AOS/VS 连接和库文件编辑用户手册(093—000245), 本手册进一步讲述连接实用程序。汇编用户程序之后, 还必须再用连接实用程序产生一个可执行的程序文件。
- AOS/VS 程序员手册(093—000241), 本手册记录着 AOS/VS 系统调用和系统参数。系统调用是执行预定的通常使用的操作。
- ECLIPSE MV/8000操作原理(014—000648)它记录着 MV/8000全部汇编语言指令。这本资料还描述ECLIPSE MV/8000硬件方面的概念。

请读者注意:

在这本手册中, 我们采用的命令形式是: COMMAND ☐required ☐optional)

其中

COMMAND 必须输入的命令

required 必须输入的变元, 有时我们采用:

$\left\{ \begin{array}{l} \text{required 1} \\ \text{required 2} \end{array} \right\}$

其意义是: 必须输入这两个变元中的一个。不必输入大括弧, 它仅表示选择项。

< optional >

表示可以任意选取角括弧中的变元。不输入角括弧, 它仅表示可以选择的项。

...

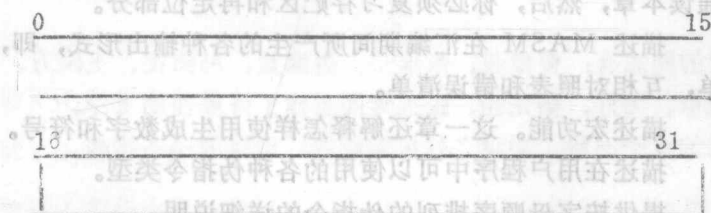
可以重复前面的输入, 这将正确地告诉你应重复些什么。

□

代表空格, 横向列表(tab), 或逗号或者它们之间的任意组合。我们通常使用□把命令和它的变元分开。

另外, 我们用/代表一个语句的结束字符。回来, 换页和换行都是语句结束符。

我们用下列形式表示位域



0位是第1位, 我们称这个位为最有效位, 在32位的情况, 31位是最后有效位。

注意: 我们把位域划分成16位的分量, 每个16位段称为一个字。

32位和16位程序

32位程序包括ECLIPSE MV/8000汇编语言指令和在32位AOS/VS操作系统下运行的程序。16位程序使用16位ECLIPSE指令, 即通常是在16位的Advanced Operating System (AOS)支持下运行。如果对16位和32位程序进行正确的汇编和连接以后, 则它们两个都可以在AOS/VS支持下运行。

用AOS/VS宏汇编实用程序去汇编32位程序。本手册深入讲述AOS/VS宏汇编程序。

在AOS/VS操作系统支持下运行16位程序时, 要用16位AOS宏汇编程序去汇编该程序(在AOS宏汇编程序参考手册中叙述, 该手册号码为093—000192)。

汇编32位或16位程序之后, 用AOS/VS连接实用程序产生一个程序文件。这就是说, 不管用的宏汇编程序是AOS还是AOS/VS, 但是, 必须使用AOS/VS连接程序就产生一个AOS/VS程序文件(即, 产生一个在AOS/VS支持下运行的程序)。AOS/VS连接和库文件编辑用户手册中, 对连接32位和16位程序方面有更详细的叙述。

目 录

3-4.3.6	代码和数据属性	3-4.3-2
3-4.3.7	有回送信息的重写和无回送信息的重写	3-4.3-2
3-4.3.8	存储区各类型	3-4.3-2
3-4.3.9	绝对区	3-4.3-2
3-4	第一章 宏汇编程序导引	3-4.3-2
3-4.3.11	NREL区	3-4.3-2
1-1	汇编语言程序开发	3-4.3-2
1-2	MASM概述	3-4.3-2
1-2.1	宏汇编程序输入	3-4.3-2
1-2.2	汇编地址和绝对值的对照	3-4.3-2
1-2.3	宏汇编程序输出表达式	3-4.3-2
1-3	MASM 的特点	3-4.3-2
1-4	MASM 最简单的应用	3-4.3-2
1-4.1	最少而又必要的伪指令	3-4.3-2
1-4.2	基本的 MASM 命令行	3-4.3-2
3-4.5.1	提供位移量和变址	3-4.3-2
3-4	第二章 宏汇编程序的输入	3-4.3-2
2-1	字符集	3-4.3-2
2-2	源语句	3-4.3-2
2-3	语句成分	3-4.3-2
2-3.1	终止符和定界符	3-4.3-2
2-3.2	数据	3-4.3-2
2-3.2.1	单精度整数	3-4.3-2
2-3.2.2	双精度整数	3-4.3-2
2-3.2.3	特殊整数生成格式	3-4.3-2
2-3.2.4	单精度浮点常数	3-4.3-2
2-3.3	符号	3-4.3-2
2-3.3.1	符号名	3-4.3-2
2-3.3.2	符号类型	3-4.3-2
2-3.3.3	永久符号	3-4.3-2
2-3.3.4	指令符号	3-4.3-2
2-3.3.5	指令符号的重新定义	3-4.3-2
2-3.3.6	宏符号	3-4.3-2
2-3.3.7	宏符号的重新定义	3-4.3-2
2-3.3.8	用户符号	3-4.3-2
2-3.3.9	用户符号的重新定义	3-4.3-2
2-3.4	表达式	3-4.3-2
2-3.4.1	运算符	3-4.3-2
2-3.4.2	一元运算符	3-4.3-2

其中	2-3.4.3	逻辑运算符	
COM	2-3.4.4	关系运算符	
requir	2-3.4.5	位定位运算符	
	2-3.4.6	B运算符	
	2-3.4.7	S运算符	
	2-3.4.8	位定运意符与符号一起使用	
	2-3.4.9	运算符的优先级	
opti	2-3.4.10	绝对表达式与浮动表达式的比较	
	2-3.5	殊特原子	
	2-3.5.1	间址符@	
□	2-3.5.2	数符(*)	
	2-3.5.3	星号(**)	
另外,	2-4	语句格式	
	2-4.1	标号	
	2-4.2	语句体	
	2-4.3	注释	
	2-5	语句类型	
	2-5.1	汇编语言指令	
	2-5.2	宏调用和系统调用	
	2-5.3	伪指令语句	
	2-5.4	赋值语句	
	2-5.5	数据语句	

第三章 汇编过程

(AOS)交	3-1	符号解释	
AOS/VS	3-3.1	符号表	
用AO	3-2	语法检查	
在AO	3-3	处理宏命令和系统调用	
序(在AOS	3-3.1	宏定义的处理	
汇编	3-3.2	展开宏命令和系统调用	
管用的宏	3-4	分配存贮单元	
AOS/VS	3-1.1	存贮器	
件编辑用	3-4.2	单元计数器	
	3-4.3	划分存贮区断	
	3-4.3.1	存贮区断属性	
	3-4.3.2	绝对、ZREL和NREL属性	
	3-4.3.3	共享和外共享属性	
	3-4.3.4	标准基区和公用基区属性	
	3-4.3.5	定位属性	

3-4.3.6	代码和数据属性	命令宏用属元变寄	2.8-2
3-4.3.7	有回送信息的重写和无回送信息的重写性	命令宏用属元变寄	2.8-2
3-4.3.8	存贮区各类型	存贮区各类型	1.8.8-2
3-4.3.9	绝对区	绝对区	2.8.8-2
3-4.3.10	ZREL区	绝对区中单部除工	1-2
3-4.3.11	NREL区	命令宏用属元变寄	2-2
3-4.4	浮动性	命令宏用属元变寄	2-2
3-4.4.1	浮动基地址	命令宏用属元变寄	7-2
3-4.4.2	浮动基地址和符号	命令宏用属元变寄	8-2
3-4.4.3	绝对地址和绝对值的对照	命令宏用属元变寄	0-2
3-4.4.4	浮动基区地址和表达式	命令宏用属元变寄	1-2
3-4.4.5	绝对表达式	命令宏用属元变寄	1-2
3-4.4.6	浮动表达式	命令宏用属元变寄	1-2
3-4.4.7	求解浮动表达式	命令宏用属元变寄	1-2
3-4.5	MRI指令中的地址定位	命令宏用属元变寄	1-2
3-4.5.1	提供位移量和变址	命令宏用属元变寄	1-2
3-4.5.2	仅提供一位移值	命令宏用属元变寄	1-2

第四章 宏汇编程序输出形式

4-1	目标文件	目标文件	7-0
4-2	汇编清单	汇编清单	8-0
4-2.1	汇编清单控制	汇编清单控制	0-0
4-2.1.1	控制清单伪指令	控制清单伪指令	01-0
4-2.1.2	全号(**)	全号(**)	11-0
4-2.1.3	汇编统计表	汇编统计表	11-0
4-2.1.4	页面大小	页面大小	11-0
4-3	交叉对照表	交叉对照表	11-0
4-3.1	交叉对照表控制	交叉对照表控制	11-0
4-4	错误清单	错误清单	11-0
4-4.1	错误清单控制	错误清单控制	11-0
4-5	输出功能开关	输出功能开关	11-0

第五章 宏命令以及产生的数和符号

5-1	宏命令	宏命令	1-7
5-2	宏定义	宏定义	1-7
5-2.1	宏定义串中的变元	宏定义串中的变元	1-7
5-3	宏调用	宏调用	1-7
5-3.1	无变元调用宏命令	无变元调用宏命令	1-7

5-3.2	有变元调用宏命令	3-4.3.6
5-3.3	把专用字符和空变元传递给宏命令	3-4.3.7
5-3.3.1	专用字符	3-4.3.8
5-3.3.2	空变元	3-4.3.9
5-4	汇编清单中的宏展开	3-4.3.10
5-5	与伪指令有关的宏命令	3-4.3.11
5-5	宏命令中的循环和条件	3-4.4
5-7	宏命令举例	3-4.4.1
5-8	系统调用	3-4.4.2
5-9	产生的数和符号	3-4.4.3

第六章 伪指令分类

6-1	单元计数器和存贮器管理	3-4.4.7
6-2	文件结束	3-4.5
6-3	重复和条件汇编	3-4.5.1
6-4	宏命令	3-4.5.2
6-5	数据布局	
6-6	模块间的通讯	
6-7	列表控制	
6-8	堆栈控制	
6-9	基数控制	
6-10	文本串	
6-11	符号删除	
6-12	其它伪指令	

第七章 伪指令说明

7-1	编写代码的辅助手段	3-4.5.1
7-2	一般参考章节	3-4.5.2
(.)	存贮单元	3-4.5.3
• ARGCT		3-4.5.4
• BLK		3-4.5.5
• COMM		3-4.5.6
3-4.3.1 • CSIZ	断属性	3-4.5.7
3-4.3.2 • DO	2REL和NREL属性	3-4.5.8
3-4.3.3 • DUSR	外共享属性	3-4.5.9
3-4.3.4 • DWORD	公用基区属性	3-4.5.10
3-4.3.5 • EJEC	性	3-4.5.11

• ELSE

• ENABLE

• END

• ENDC

• ENT

• EOF

• EXT D

• EXTL

• EXT N

• FORC

• GLOC

• IF E, • IF G, IF L, • IF N

• LOC

• MACRO

• MCALL

• NOCON

• NOLOC

• NOMAC

• NREL

• OB

• PART

• PASS

• POP

• PUSH

• RDX

• RDXO

• REV

• SWORD

• TITLE, • TITL

• TSK

• TXT

• TXTM

• TXTN

• UWORD

• WORD

• XPNG

• ZREL

第 8 章 汇编语言

语言实际上就是由一些定义的一些表示方法、约定的符号和语法规则的集合。最低级的计算机语言是由硬件操作的数字代码组成的，计算机语言就是这种称为机器语言的数字语言。

例如，下面的机器语言代码表示将地址 377 的转移指令，
000500 是转移指令而 377 是内存单元。

用 ECLIPSE 汇编器将机器语言代码翻译成汇编语言代码，汇编语言代码的格式如下，
因此，为使程序设计方便，我们为每条机器语言指令，设计一种汇编语言代码。

例如，机器语言代码 `000500` 翻译成汇编语言代码 `MCALL 377`，
这条指令读起来简单得多。

与指令一样，我们也可以用符号来表示地址。例如，地址 377 可以用符号 `LOC` 表示，
可以给 377 号单元赋以符号 `LOC`，于是，该指令成为：

• NOMAC LOC
这条指令就比较容易理解，即：将程序计数器 PC 移到 `LOC` 单元。
这种由符号而不是由数字组成的程序设计语言叫做汇编语言。

因此，汇编语言提供了与机器语言相同的操作集，而用起来却要简单得多。

与机器语言不同，计算机不容易直接理解汇编语言。因此，必须设法把用符号表示的汇编语言程序翻译成机器语言程序。宏汇编程序（即 `MACRO`）正是执行这种翻译操作的系统程序。

我们先来简单地讨论一下编写、执行和调试汇编语言程序的五个必不可少的步骤，然后我们再讨论宏汇编程序的编写。

1. 编写和编辑源程序
当然，编写程序的第一步是把适当的汇编语言指令输入到计算机中去。通常，是从控制台上输入。采用 `DG` 公司的一种文章编辑程序来输入程序，汇编语言程序叫做源模块，若存入磁盘，则以源文件的形式驻留。（按照约定，源文件名以后缀 `SR` 结尾。）

2. 对程序进行汇编
源程序输入以后，就可以调用宏汇编程序，把由符号组成的汇编语言源模块，翻译成由数字表示的机器语言。宏汇编程序把这种称为目标模块的机器语言程序，存入叫做目标文件（宏汇编程序总是用后缀 `OB` 作目标文件的结尾）。如果宏汇编程序在源模块中遇到错误，则必须对源文件重新进行编辑和汇编，然后再继续程序的开发过程。

3. 对程序进行连接
汇编通过后，还必须用连接实用程序来产生可执行程序。连接程序把目标模块链接，然后重新排列，形成运行时程序会在内存中出现的图象。连接程序把这种图象存贮起来。（连接程序总是用后缀 `PR` 作为程序文件名的结尾。）与宏汇编程序一样，连接

第八章 宏汇编程序操作过程

8-1 宏汇编程序命令行

8-1.1 命令行开关

8-2 连接和执行用户程序

8-3 文件名

8-3.1 通用文件名

8-4 宏汇编程序符号表

8-4.1 符号识别

8-4.2 永久符号表

8-4.2.1 建立永久符号表

8-4.2.2 确定汇编符号表

8-4.2.3 永久符号表的长度

8-4.2.4 符号长度

符录A ASCLL字符景(略去)

符录B 伪指令一览表

符录C 各种错误信息

7-5 汇编 MASM 命令行错误
7-7 列表 汇编错误

7-8 堆栈控制

7-9 基数控制

7-10 文本串

7-11 符号删除

7-12 其它伪指令

第七章 伪指令说明

7-1 编写代码的辅助手段

7-2 一般参考章节

7-3 一般参考章节

7-4 ARGCT

7-5 BLK

7-6 COMM

7-7 CSIZ

7-8 DO

7-9 DUSE

7-10 DWORD

7-11 EJECT

第一章 宏汇编程序导引

语言实际上就是经过严格定义的一些表示方法、约定,以及传送信息的规则的集合。最低级的计算机语言是由代表机器硬件操作的数字代码组成的,计算机容易理解这种称为机器语言的数字语言。

例如,下面的机器语言代码代表转向地址377的转移指令:

000 377

000000是转移指令的基值,而377是内存单元。

用ECLIPSE MV/8000的名目繁多的机器指令给程序编码,是很费时间的烦琐工作,因此,为使程序设计方便起见,我们为每条机器语言指令,设计了一个有效的符号名。

例如,机器语言指令 000000 用符号名 JMP 代替,于是前面的转移语句成为:

JMP 377

这条指令读起来简单很多,因为符号 JMP 类似于英文单词 JUMP,而具隐含着对应的机器级令所执行的操作。

与指令一样,我们还可以用符号表示地址来进一步简化机器语言。上一个例子中,我们可以给377号单元赋以符号名 LOCI。于是,该指令成为:

JMP LOCI

这条指令就比较容易阅读和理解,即:“转移到LOCI单元”。

这种由符号而不是由数字组成的程序设计语言叫做汇编语言。一般地说,每一条汇编语言指令对应一条机器语言指令。因此,汇编语言提供了与机器语言相同的操作集,而用起来却要简单得多。

与机器语言不同的是,计算机不容易直接理解汇编语言。因此,必须设法把用符号表示的汇编语言程序翻译成等价的机器语言程序。宏汇编程序(即MASM)正是执行这种翻译操作的系统程序。

1-1 汇编语言程序开发

我们先来简单地浏览一下编写、执行和调试汇编语言程序的五个必不可少的步骤,然后再讨论宏汇编程序的操作。

1.编写和编辑源程序——当然,编写程序的第一步是把适当的汇编语言指令输入到计算机中去。通常,是从控制台上,采用DG公司的一种文章编辑程序来输入程序。汇编语言程序叫做源模块,若存入磁盘,则以源文件的形式驻留。(按照约定,源文件名以后缀SR结尾。)

2.对程序进行汇编——源程序输入以后,就可以调用宏汇编程序,把由符号组成的汇编语言源模块,翻译成等价的用数字表示的机器语言。宏汇编程序把这种称为目标模块的机器语言程序,存入叫做目标文件的新文件(宏汇编程序总是用后缀OB作目标文件的结尾)。如果宏汇编程序在源模块中测试出错误,则必须对源文件重新进行编辑和汇编,然后再继续程序的开发过程。

3.对程序进行连接——汇编通过后,还必须用连接实用程序来产生可执行程序。连接程序把目标模块拉断,然后重新排列,形成运行时程序会在内存中出现的图象。连接程序把这种图象存贮起来。(连接程序总是用后缀·PR作为程序文件名的结尾。)与宏汇编程序一样,连接

实用程序也会测试出错误。如果有错误，就必须再对程序进行编辑，并重新进行汇编和连接等步骤。以便改正错误。

4. 运行程序——对程序进行连接成功之后，就可以通过在控制台上键入以下命令来运行该程序：

XEQ Program—filename

若程序运行顺利并能执行适当的动作，则程序开发过程就完成了。

5. 调试该程序——起初程序往往不能正常运行。有以下两种可能的情况：

- 或者引起运行时错误

- 或者不完成所要求的动作

无论属于哪种情况，都必须对程序进行调试（即纠正错误（即“bug”））。有时很容易发现错误，在这些情况下，只要改正源模块中的错误，然后重新进行汇编，连接就行了。如果确定不了源模块中的错误，则可以使用 AOS/VS Debugger 实用程序，在运行中检测程序的各个部分。找到错误以后，还必须再对源程序进行编辑，然后重新对它进行汇编和连接。

上面概述的轮廓仅仅是汇编语言程序设计的一个很简略的梗概。本手册的其余部分，主要介绍第二步即汇编过程。前言中列出了介绍程序设计的其它步骤的手册清单，可供参考。

1—2 MASM 概述

下面几节简要讨论对于宏汇编程序的输入，汇编过程，以及宏汇编程序产生的输出。对这些题目的更详尽的讨论是第二章、第三章和第四章的任务。

1—2.1 宏汇编程序输入

输入宏汇编程序的源模块由字符组成，这些字符组合成一系列的源模块语句。源语句一般有以下功能：

- 运行时完成一些操作

- 包含数据或

- 控制宏汇编程序的操作

1—2.2 汇编

宏汇编程序阅读两遍源模块中的语句。在两遍阅读源程序的过程当中，MASM 的任务是：

- 解释所有符号

- 检有源语句是否合乎语法

- 进行内存分配

- 展开宏功能和系统调用（在本章的稍后叙述）。

1—2.3 宏汇编程序输出

宏汇编程序可以产生五种类型的输出：

- 目标文件

- 汇编清单

- 交叉访问表
- 错误清单
- 永久符号表

目标文件保存源模块经过汇编生成的机器语言版本。汇编清单可以用来将宏汇编程序的输入与它产生的输出进行比较。该清单中包含有原始的源语句加上 MASM 产生的数值的机器语言值。

交叉访问表提供程序中所使用的符号字母表，后面跟着各自的数值。

错误清单中列出源模块中引起汇编错误的所有语句的信息。

永久符号表保存符号定义，以便将来汇编时使用。如果使用永久符号表，则每次汇编时无需对那些经常使用的符号重新定义。高级操作系统/虚拟存贮 (AOS/VS) 软件包为用户提供了标准的永久符号表。

1—3 MASM 的特点

我们已经看出，宏汇编程序的主要功能，是把由符号组成的汇编语言程序翻译成等价的机器语言程序。在此过程中，宏汇编程序可以完成各种提高程序设计能力的操作，同时还可以简化源代码。

下面列出宏汇编程序的部分特点，其详细解释则在本手册的其它部分。

符号地址名 如前所述，宏汇编程序允许给内存单元赋以符号名，叫做标号。

数的表示 有三种不同的内部数表示法(单精度整数，双精度整数和浮点数)可以自由使用。

表达式计算 宏汇编程序为数值计算提供了算术、逻辑和关系运算符。

存贮管理 宏汇编程序既可以给目标代码赋绝对地址，又可以赋以在连接方最后确定的浮动地址。

重复汇编 可以控制 MASM 对一些源语句汇编一定的次数，就是说，汇编时可以实现 DO 循环。

条件汇编 宏汇编程序允许根据对表达式的计算结果，这源代码的一个部分进行条件汇编或实行旁路 (by-pass)；换句话说，汇编时可以实现 IF—THEN—ELSE 结构。

16位和32位数 宏汇编程序对源模块中的各个数据，既可分配16位存贮单元，又可分配32位存贮单元。

文本字符串的存贮 可以控制宏汇编程序对于任何字符串均以 ASCII 码存贮。

基数控制 可以改变转入宏汇编或从宏汇编输出的数值的基数。

汇编程序堆栈 宏汇编程序提供一个在汇编时操作的下推堆栈。

模块间的通信 可以单独汇编各个源模块，然后把它们连接成一个程序文件。这些单独汇编的模块可以共享数据和符号定义。

宏功能 宏功能允许给许多源语句赋以符号名。然后，每当需要插入源语句时，只要输入所赋的符号名就行了。汇编时，宏汇编程序可以正确地把这些宏名展开成原来的源语句。

宏汇编程序提供上述这些以及其它一些程序设计工具，以便帮助你进行程序开发。注意，装配实用程序和 AOS/VS 系统调用(预定义的宏功能)，则进一步提供对汇编语言程序设计

(汇编语言程序设计)