

M6800微型计算机资料之五

M6800 汇 编 程 序
参 考 手 册

广州市自动控制研究所译

前 言

我所引进美国 MOTOROLA 公司生产的 M6800 微型计算机发展系统资料一套，应有关单位要求，我们翻译了全文，进行内部交流。

这套资料共分十六册，其名称如下：

- 一、M6800 EXORCism 200 用户指南。
- 二、M6800 连接装配程序参考手册。
- 三、M6800 驻留编辑程序参考手册。
- 四、M6800 BASIC 解释程序参考手册。
- 五、M6800 汇编程序参考手册。
- 六、M6800 宏汇编程序参考手册。
- 七、M6800 驻留 FORTRAN 编译手册。
- 八、M6800 EXORCier USER's 指南补充。
- 九、打印机接口模板用户指南。
- 十、EXORdisk II 软磁盘控制器模板用户指南。
- 十一、M6800 磁盘操作系统预备用户指南。
- 十二、M6800 驻留 MPL 编译参考手册。
- 十三、M6800 MM01A 单板计算机用户指南。
- 十四、M6800 微型机发展系统电路图集。
- 十五、用户系统评价器 USER 用户指南。
- 十六、系统分析器用户指南

由于我们水平有限，加上时间仓促，虽然作了很大努力，但在翻译校正过程中还可能存在不少不妥和错误之处，恳请广大读者及时批评指正。

M6800 驻留汇编程序参考手册

第一章 概述

- 1、1 引言
- 1、2 M6800 驻留汇编程序语言
 - 1、2、1 机器操作码
 - 1、2、2 伪指令
- 1、3 M6800 驻留汇编程序
 - 1、3、1 汇编目标
 - 1、3、2 汇编操作
- 1、4 定货资料
- 1、5 使用环境
 - 1、5、1 设备要求
 - 1、5、2 软件要求

第二章 M6800 驻留汇编程序的编码

- 2、1 源语句格式
 - 2、1、1 行序号
 - 2、1、2 标号字段
 - 2、1、3 操作码字段
 - 2、1、4 操作数字段
 - 2、1、5 注释字段
- 2、2 表达式
 - 2、2、1 常数
 - 2、2、2 ASC II 文字符
- 2、3 符号
- 2、4 M6800 寻址方式
 - 2、4、1 隐含和累加器寻址方式
 - 2、4、2 立即寻址方式
 - 2、4、3 相对寻址方式
 - 2、4、4 变址寻址方式
 - 2、4、5 直接和扩展寻址方式
- 2、5 汇编程序格式
 - 2、5、1 汇编清单
 - 2、5、2 目标程序

第三章 伪指令

- 3、1 引言
- 3、2 End
- 3、3 EQU——使用符值相等
- 3、4 FCB——形成常数字节

- 3、5 FCC——形成常数字符
- 3、6 FDB——形成双字节常数
- 3、7 NAM——程序名
- 3、8 OPT——输出选项
- 3、9 ORG——起点
- 3、10 PAGE——页的顶部
- 3、11 RMB——保留存储器字节
- 3、12 SPC——空格

第四章

- 4、1 概述
- 4、2 纸带/磁带驻码汇编程序操作方法
 - 4、2、1 从纸带/磁带上装入驻码汇编程序
 - 4、2、1、1 把纸带/磁带装入 Exorjiser
 - 4、2、1、2 把纸带/磁带装入 Evaluation Module 存储器
 - 4、2、2
 - 4、2、3 纸带/磁带驻码汇编程序的操作
- 4、3 磁盘驻码汇编程序操作方法
 - 4、3、1 磁盘驻码汇编程序操作特点
 - 4、3、2 磁盘驻码汇编程序的操作

附录A：字符集

附录B：M6800 指令摘要

附录C：M6800 驻码汇编程序 伪指令摘要

附录D：汇编程序错误信息表

附录E：绝对目标记录格式

附录F：程序实例

附录G：使用 Mik Bug 文本的 M6800 驻码软件

附录H：其它外围设备和驻码软件一起使用

附录I：PROM 文本的驻码汇编程序

第一章

概述

1.1 引言

M6800 驻码汇编程序是一种用来处理用 M6800 汇编语言编写的源语句的程序，它把这些源语句编译成与 M6800 固件装载程序 (Firmware Loader) 相容的目标程序。并产生一个源程序的格式清单。M6800 驻码汇编程序与 MPCASM 和 M68SAM 交叉汇编程序相容，还能和 M6800 驻码编辑程序 (Editor) 一起驻码在存储器里。关于编辑程序的说明，请参看《M6800 驻码编辑程序参考手册》。

1.2 M6800 驻码汇编程序语言

由汇编程序处理的——用符号语言编辑的源程序，称为 M6800 驻码汇编程序语言。

这个语言由以下助记符的集合来表达：

- 操作符

- M6800 机器指令操作码

- M6800 驻码汇编程序伪指令

- 符号名 (标号)

- 运算符

- 特定符号

1.2.1 机器操作码

汇编语言提供了在 M6800 指令系统中所有的机器指令的助记代码。这些机器指令均在《M6800 程序参考手册》中详细说明。M6800 指令摘要见附录 B。

1.2.2 伪指令

汇编语言还包括一些由汇编程序来执行规定的辅助功能的助

记符伪指令。这些伪指令并不全部翻译成机器语言，（伪指令的说明见第三章，它们的摘要见附录C）。

1、3 M6800 驻留汇编程序

M6800 驻留汇编程序把用 M6800 汇编语言编写的源语句编译成机器语言，对指令和数据分配存储单元，以及执行一些由编程者指定的辅助汇编操作。

1、3、1 汇编目标

M6800 驻留汇编程序有两个基本的目标：

- 把源程序编译成目标代码，这些目标代码的格式符合 M6800 驻留装配程序或 Exorciser 相容的装配程序的要求。

- 提供一个包括有输入的源语言汇编出来的目标代码以及一些有助于分析程序的附加信息（例如错误信息，如果有的话）的打印清单。

1、3、2 汇编操作

汇编程序分两次读入源程序：第一次，建立一个符号表；第二次，参照第一次扫描时建立的符号表来汇编目标程序。在第二次扫描周期，产生目标代码和汇编清单。在下一行源语句被读入以前，每一行的源语句已被处理完毕。

当每一行的源语句被处理时，汇编程序检测其存储地址，操作码，操作数字段。为寻找和操作字段相匹配的代码，扫描操作码表格。如果正被处理的是标准的机器操作码，则正确的数据被打入目标代码中去。如果规定的是一伪指令，则产生相应的动作。为了准备输出，目标代码和汇编清单都被形成，同时，在含有错误的行被打印以前，任何检测到的实际的和潜在的错误标志也被形成。

1、4 定货资料

M6800 驻留汇编程序可以在 M6800 Exorciser、评价模板 I、评价模板 II，以及评价包等装置上使用。表 1-1 列出了可选择的汇编程序，它们的部件号，以及与它们一起工作的

硬件。

1、5、使用环境

1、5、1、设备要求

M6800 驻留汇编程序要求最少的设备为：

- Exorciser 、评价模板 I、评价模板 II、或评价包。
- 8K字节的 RAM。
- 带有 TTY (20 mA 中性电流) 或 RS-232C 接口的终端，并装有自动阅读/穿孔的控制。

1、5、2 软件要求

M6800 驻留汇编程序和 Exbug 固件，Mikbug 固件、Minibug 固件一起工作，还可以和 Exordisk 及 Edos II 软件操作系统一起使用。

注 意：

当驻留软件和评价模板 I 或评价包一起使用时，要按照附录 G 来修改一些硬件。

表 1—1 驻留汇编程序

硬 件	软件名称	软件部件号*
1、Exorciser (Exbug)	驻留汇编程序	M68 ASMRO13A B, D
2、评价模板 I (Mikbug)	驻留汇编/编辑 程序	M68 ASM6813 A, B
3、评价模板 II (Minibug)	驻留汇编程序	M68 ASM R213 A, B

* A = 磁带, B = 纸带, D = 磁盘

第二章

M6800 汇编程序语言的编码

2.1 源语句格式

用汇编语言写的程序由一系列源语句组成。而每个源语句又由以“CR”(Carriage Return)结尾的一系列ASCII字符组成,请参看附录A的ASCII字符集。

每个源语句最多能包含以下五个字段:

- 行序号
- 标号(如是*,则表示是注释行)
- 操作码
- 操作数
- 注释

2.1.1 行序号

行序号字段是任意选择的,它提供给编程者的一个方便之处。行序号字段在一行源语句的起始位置,由最多5位十进制数组成(其值必须小于65536)。它后面必须跟有一个空格。

虽然行序号是任意的,但它们的使用要有一致性,即在整个程序中要么都用,要么都不用。如果第一个源语句有行序号,则所有的源语句也都要有行序号,如果第一个源语句无行序号,则其它源语句都应无行序号。在这种情况下,汇编程序将在汇编清单中提供连续的行序号。

2.1.2 标号段

标号段直接跟在顺序号字段(如果有的话)之后,或者作为源语句行的第一个字段。标号段可以采用以下形式之一:

(1) 一个星号(*)作为第一个字符,表示这个源语句行是一个注释行,在汇编过程中将被忽略(除用作表列清单外)。

(2) 空格(b)作为第一个字符,表示标号字段是空的(这行源代码行不是注释,也没有标号)。

(3) 一个符号

这个符号的特征是:

- 由1—6个字符组成
- 符号中有效的字符是字母A~Z, 数字0~9。
- 符号的第一个字符必须是字母。
- “A”、“B”、“X”是汇编程序中的一些特定符号,绝对不要单独用在标号段中。

标号段中的只能出现一次。如果某个标号出现在多个标号段内,那么在每次引用这个标号时,就会引起错误。

通常,把被汇编的指令或数据的第一字节的程序计数器值赋给标号(指在标号段内的符号)。(译注: EQU 伪指令除外)。

把操作数段中的表达式的值赋给 EQU 伪指令的标号。

某些伪指令的标号段必须没有标号,这些伪指令是: ORG, NAM, END, OPT, PAGE, SPC。

在符号表中,程序里的每个符号被分配给8字节的位置。

2. 1. 3 操作码字段

在汇编语言的源语句中,操作码字段直接跟在标号段之后。这个字段由3或4个字符的操作码组成,有关符号的规则也适用于操作码字段。

在操作码字段中的所有操作码是以下两种类型之一:

- 机器助记符操作码 —— 它们直接对应于 M6800 的机器指令。这个操作码字段包括在双寻址和累加器寻址方式时所用的“A”或“B”字符。为了与其它 M6800 汇编程序兼容,可把一个空格打入结果累加器和操作码之中(即: LDA A 和 LDAA 是一样的)。

- 伪指令 —— 是被汇编程序所读取的特定的操作码,用于汇编过程的控制,而不直接地编译成机器语言。

在汇编过程中，汇编程序在机器操作码和为指令的表格里检索操作码，如果找不到，就打印出错误信息。

2. 1. 4 操作数字段

操作数字段的内容是从属于操作码字段的。对于 M6800 的机器指令，它的操作数字段必须要规定寻址方式。操作数字段的格式和所对应的寻址方式如下：

操作数格式 M6800 机器指令寻址方式

无操作数 —— 隐含寻址和累加器寻址

表达式 —— 直接寻址或扩展寻址（如有可能则用直接寻址）

<表达式> —— 立即寻址

<表达式> X —— 变址寻址

寻址方式和表达式在《M6800 程序手册》中加以说明。汇编程序伪指令是采用另一种形式，详见第三章。

2. 1. 5 注释段

源语句最后一个字段是注释段，这个字段是任意的，并且除了用在汇编清单以外，汇编程序对它不处理。要用一个或多个空格把注释段和操作数字段隔开（若无操作数字段，则和操作码字段隔开）。此字段可由任何 ASCII 字符组成，它的重要之处在于用来说明程序的运行情况。

2. 2 表达式

一个表达式是由数字运算符（+、-、*、/）分隔开的一系列符号与/或数字的组合。

汇编程序对代数表达式的求值是从左到右执行，没有括号组合，也没有数字运算符的优先问题。在对表达式求值过程中得到的小数或中间值，都将会舍去小数部分而成为整数。

2. 2. 1 常数

十进制数：<数字>

十六进制数：\$<数字> 或 <数字> H

八进制数: @ <数字> 或 <数字> O 或 <数字> Q

二进制数: % <数字> 或 <数字> B

2. 2. 2 ASCII 字符

· <字符> (撇号(,) 后面的是 — ASCII 字符) 其结果变为 ASCII 字符的数值。

2. 3 符号

在表达式中的符号和标号段中的符号相类似, 只是这些表达式符号的值是被引用的, 而不是被定义的, 星号(*) 是被汇编程序所识别的一个特殊符号, 它代表现行的指令计数器的值(被用在符号的上下文时, 表示指令的第一字节的值)。

每个符号对应于十六位整数, 在表达式求值时, 这个值被用来代替这个符号。

M6800 驻留汇编程序是二次扫描汇编程序。在第一次扫描时建立一个符号表, 第二次扫描产生目标记录和汇编清单。在第一次扫描周期, 某些表达式不能完全求值, 是因为它们含有向前引用的而又没有被定义的符号。在某些情况下, 在做第二遍扫描前一个符号没有被定义, 由于汇编程序不能对这种符号求值, 把这些情况作为错误处理, 符号仅允许一次向前引用。(Only one level of forward referencing is allowed)

2. 4 M6800 寻址方式

2. 4. 1 隐含寻址和累加器寻址

M6800 的某些指令仅需要一个操作码字节。这些指令使隐含寻址和累加器寻址方式, 在写 M6800 汇编语言时, 它们不需要操作数段。

2. 4. 2 立即寻址方式

用一或两个字节的立即数放在指令操作码的后面作为操作数, 就称为立即寻址。在源语句行中, 立即寻址方式由操作数段前面的“#”符号来选定。表达式跟在“#”后, 需要一或二个字节, 这取决于不同的指令。

2. 4. 3 相对寻址方式

相对寻址用于分枝(转移)指令。分枝转移仅能够在相对于分枝指令第一字节的地址从-126到+129的范围里进行:

$$(PC + 2) - 126 \leq D \leq (PC + 2) + 127$$

其中: PC = 分枝指令第一字节的地址

D = 分枝指令的目的地址

实际的地址偏移量放在分枝指令的第二字节里,它是紧跟在分枝指令之后的地址和目的地址之差的补码。

2. 2. 4 变址寻址方式

变址寻址是相对于 M6800的变址寄存器而言的。在执行指令时,把存放在指令第二字节中的一字节的偏移量和存放在16位变址寄存器中的内容相加,得到新的执行地址。因为所执行的是无符号数运算,所以没有负偏移量。

变址寻址通常用跟在操作数字段中表达式之后的字符“,X”来表示。(在特殊情况下:“,X”或一个单独的“X”都等同于“@,X”。)

2. 4. 5 直接寻址和扩展寻址方式

直接寻址和扩展寻址用一个字节(直接)或二个字节(扩展)来形成所需要的操作数的地址。直接寻址的范围被限制在存储器的前256个字节(0~255)之中。直接寻址和扩展寻址的选用是简单的,把地址直接写入操作数字段的表达式即可。如果可能的话,应采用直接寻址方式。在源程序中,如果可直接寻址的变量在还没有定义之前就引用,就会产生寻址错误(Phasing error)。为了避免这种问题,对于这些可直接寻址的变量要先定义再引用。

2. 5 汇编程序清单

汇编程序的输出包括汇编清单和目标程序。

2. 5. 1 汇编清单

汇编清单既包括源程序，又包括在汇编过程中产生的附加信息。在这个表中，大多数行直接对应于源语句，而不直接对应于源语句的行有：

- 页标题行
- 错误行（请参看附录D中的错误号）
- FCC、FDB、FCB 伪指令的扩展行

在表中大多数行都符合表2—1中的标准格式。

2、5、2 目标程序

在附录E中，详细地说明了绝对的和可再定位的目标程序的格式。

表2—1 标准格式

（特殊情况可以使用不同的格式）

列	内 容
1 ~ 5	源程序行序号# — 由汇编程序按五位十进制值进行计数。
7 ~ 10	现行的地址计数器值（十六进制）
12 ~ 13	机器操作码（十六进制）
15 ~ 16	操作数第一字节（十六进制）
17 ~ 18	操作数第二字节（十六进制）
20 ~ 25	标号字段
27 ~ 31	操作码字段
34 ~ 41	操作数字段（若操作数字段较长，可扩展到注释字段中去）
43 ~ 最后一列	注释字段

第 三 章

汇 编 伪 指 令

3.1 引 言

汇编伪指令是进行汇编时用的指令，它并不直接编译成目标代码。这一章对 M6800 驻留汇编程序的汇编伪指令的结构进行说明。

表 3-1 中的伪指令是按功能来分组。以后的各条伪指令的详细说明按字母排列。

表 3-1 伪指令

伪 指 令	功 能
<u>汇编控制</u>	
NAM	程 序 名
ORG	起 点
END	程序结束
<u>汇编清单控制</u>	
PAGE	页的顶部
SPC	跃过“n”行
OPT NOO	无目标带
OPT O	汇编程序将产生一个目标带（隐含选择
（目标带）	（Selected by default）。
OPT M	汇编程序将把机器码写入内存中去。
（内存文件）	

· 续上表 ·

伪指令	功能
OPT NOM	不产生内存文件 (隐含选择)
OPT S (打印符号)	汇编程序在第二次扫描结束时印出符号。
OPT NOS	不印出符号 (隐含选择)。
OPT NOL (无清单)	汇编程序将不印出汇编数据清单。
OPT L	印出汇编数据清单 (隐含选择)。
OPT NOP (无页)	汇编程序将禁止汇编清单使用页格式。
OPT P	汇编清单按页格式 (隐含选择)。
OPT NOG (不产生)	在使用伪指令 FCC、FCB、FDB 时, 在汇编清单中仅产生一行的数据。
OPT G	印出由伪指令 FCC、FCB、FDB 产生的所有数据 (隐含选择)
<u>数据的定义</u>	
<u>存储分配</u>	
FCC	字符串数据
FCB	一字节数据
FDB	双字节数据
RMB	保留存储字节
<u>符号定义</u>	
BQU	赋给固定数值

译注: 在某些伪指令后注明“隐含选择”(Selected by default)时,即表示: 当没有选择其它指令方式时, 汇编程序则自动选择这种指令。

3.2 END

格式: END

说明: END 伪指令向汇编程序表明源程序的结束。在它以后的源语句不予处理。当第一次扫描结束,而遇到END伪指令时,即开始汇编程序的第二次扫描。

3.3 EQU — 表示与符号值相等(Equate Symble Value)

格式: <标号> EQU <表达式> [*<注释>*]

说明: EQU 伪指令把在操作数字段中的表达式的值与赋给标号字段中的符号。标号和表达式要遵守以前各节所说明的规则。注意 EQU 是一个操作符,它把一个与程序计数器不同的值赋给标号。这个伪指令的标号段和操作数字段都是需要的,并且标号不能在程序的任何其它地方定义。

在 EQU 伪指令的操作数字段的表达式中,不能有无定义或尚未定义的符号(如符号不是向前引用,则允许有)。

3.4 FCB — 形成一字节常数(Form constant byte)

格式: [*<标号>*] FCB

$$\left\{ \begin{array}{l} \left\{ \begin{array}{l} \langle \text{表达式} \rangle, \\ \langle \text{空} \rangle, \end{array} \right\} \left[\begin{array}{l} \langle \text{表达式} \rangle, \\ , \end{array} \right]^{00} \left[\langle \text{表达式} \rangle \right] \\ \langle \text{表达式} \rangle \\ \langle \text{注释} \rangle \end{array} \right\}$$

说明: FCB 伪指令可以有一个或多个操作数(中间用逗号隔开),对应于每个操作数的值的是一个8位无符号二进制数,它被存放在目标程序的一个字节中。如果操作数多于一个,则它们存放在目标程序中连续相邻的字节中。操作数字段可以有实际的值(十进制、十六进制、八进制或二进制),也可以是由汇编程序赋给数值的符号或表达式。

当一条 FCB 伪指令后面跟着有一个或多个用逗号隔开的空操作数时,这些空操作数即存储零值。

3.5 FCC — 形成常数字符(Form constant character)

格式: [< 标号 >] FCC

$$\left\{ \begin{array}{l} d < \text{ASCII 字符串} > d \\ < \text{十进制数} >, < \text{ASCII 字符串} > \\ < \text{注释} > \end{array} \right\}$$

注: 1. “d”是任意非数値字符(作为分隔符用)

2. ASCII 字符串不可包括“CR”(Carriage return)

说明: FCC 指令把字符串翻译成七位的 ASCII 码。对应于 ASCII 十六进制码 20(SP) 到 5F(-) 中的任何字都能够用这条指令来处理。

1. 计数、逗号、文本(Count、comma、text), 计数规定要产生多少 ASCII 字符, 而文本开始于操作数的第一个逗号之后。若计数比文本长, 则(译注: 在目标程序数据栏内)填入空格, 使计数达到要求的值。最大计数为 255。

2. 文本包含在两个分隔符中间, 每个分隔符可为任何的单字符(如果分隔符是数, 则文本就不能以逗号开始)。

3. 6 FDB —— 形成双字节常数(Form double constant byte)

格式: < 标号 > FDB

$$\left\{ \begin{array}{l} \left\{ \begin{array}{l} < \text{表达式} >, \\ < \text{空} >, \end{array} \right\} \left[\begin{array}{l} < \text{表达式} >, \\ , \end{array} \right]_0^{00} \left[< \text{表达式} > \right] \\ < \text{表达式} > \\ < \text{注释} > \end{array} \right\}$$

说明: FDB 伪指令可以有一个或多个用逗号隔开的操作数。每个操作数的值对应于一个 16 位无符号的二进制数。这个数存储在目标程序的两个字节中。如果操作数多于一个, 则它们所对应的 16 位二进制数存放在目标程序中连续的相邻字节中。操作数字段可以含有实际的值(十、十六、八、二进制), 也可以是由汇编程序赋给数值的符号或表达式。

当一条 FDB 伪指令后面跟有一个或多个被逗号隔开的空操