

M6800微型计算机资料之七

M6800 驻 留 FORTRAN

编 译 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 驻留汇编程序参考手册。
- 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 EXORCism User's 指南补充。
- 九、打印机接口模板用户指南。
- 十、EXORDISK II 软磁盘控制口模板用户指南。
- 十一、M6800 磁盘操作系统预备用户指南。
- 十二、M6800 驻留 MPZ 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、用户系统评价口 USER 用户指南。
- 十六、系统分析口用户指南。

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。

目 录

第一章 序言 ----- 1

- 1.1 在EDOS中的FORT 命令
- 1.2 在EDOS 中抹去行和字符
- 1.3 在MDOS 中的FORT 命令

第二章 FORTRAN 编译程序 ----- 4

- 2.1 语句
- 2.2 编写FORTRAN语句
- 2.3 语言单元
- 2.4 常数
 - 2.4.1 整常数
 - 2.4.2 实常数
 - 2.4.3 文字常数
- 2.5 标识符
- 2.6 变量
 - 2.6.1 变量名
 - 2.6.2 变量类型和字长
 - 2.6.3 用于先定义的类型说明
- 2.7 数组
 - 2.7.1 说明一个数组的规模和类型
 - 2.7.2 在存储器中数组的排列
- 2.8 下标
- 2.9 表达式
 - 2.9.1 算术表达式
 - 2.9.2 算术运算符
 - 2.9.3 构成算术表达式的规则
 - 2.9.4 逻辑表达式

第三章 算术赋值语句-----18

3.1 一般形式

3.2 赋值语句

第四章 控制语句-----20

4.1 引言

4.2 GO TO 语句(转向语句)

4.2.1 无条件GO TO 语句

4.2.2 标号GO TO 语句

4.3 算术控制语句

4.3.1 算术IF语句

4.3.2 逻辑IF语句

4.3.3 DO语句

4.3.4 编写DO 循环注意事项

4.3.5 CONTINUE 语句

4.3.6 STOP语句

4.3.7 END 语句

第五章 输入/输出语句-----28

5.1 输入/输出表格

5.2 顺序的输入/输出语句

5.3 READ 语句

5.4 WRITE 语句

5.5 PRINT 语句

5.6 REWIND 语句(反绕语句)

5.7 FORMAT 语句

5.7.1 一个FORMAT 语句的各种形式

5.7.1.1 逗号

5.7.1.2 斜线

- 5.7.2 I 格式代码
- 5.7.3 E 和 F 格式代码
- 5.7.4 数值格式代码的一些例子
- 5.7.5 A 格式代码
- 5.7.6 X 格式代码
- 5.7.7 组格式说明
- 5.7.8 自由格式输入
- 5.8 EDOS的OPENF/CLOSEF语句
- 5.8.1 EDOS的OPENF/CLOSEF语句的参号
- 5.8.2 EDOS的OPENF/CLOSEF语句编程注意事项
- 5.8.3 EDOS的OPENF/CLOSEF语句的一些例子
- 5.8.4 MDOS的OPENF/CLOSEF语句
- 5.8.5 MDOS的OPENF/CLOSEF语句编程注意事项
- 5.8.6 MDOS的OPENF/CLOSEF 语句的一些例子

第六章	数据子置语句	48
-----	--------	----

第七章 说明语句 49

- 7.1 定义
- 7.2 DIMENSION 语句
- 7.3 COMMON 语句

第八章 子程序-----51

- 8.1 概述
- 8.2 给子程序命名
- 8.3 函数
- 8.3.1 函数定义
- 8.3.2 调用函数
- 8.4 函数子程序

5/5/20/02

8.5 在一个 FUNCTION 子程序中的 RETURN 语句

8.6 SUBROUTINE 子程序

8.6.1 CALL 语句

8.6.2 在一个 SUBROUTINE 子程序中的 RETURN 语句

8.7 在一个 FUNCTION 或 SUBROUTINE 子程序中的形参

附录 A 源程序字符

附录 B 编译时的错误信息

附录 C 执行时的错误信息

附录 D MOTOROLA 提供的函数

附录 E FORTRAN 的一些例子

附录 F 把 FORTRAN 和汇编程序连接起来

附录 G 建立一个库

附录 H 改变运行 I/O 地址

第一章 序言

以 Motorola M6800 EXORCiser 为基础的 FORTRAN 编译程序系统是一个为解答中小规模的科学和商业问题而设计的紧凑的计算机系统。系统由计算机的硬件和软件组成。最小的结构是：

- EXORCiser
- 16,384 字节的存储口
- 一个 EXORDISK 驱动单元 (262,244 字节)
- 与 TTY 或 RS-232C 兼容的终端
- M6800 驻留编辑程序
- M6800 连接装配程序 (Linking Loader)

最小结构可以扩充到包括 65,536 个字节的存储口和总存储能力超过一百万字节的四个磁盘。此外，可以使用各种硬拷贝和 CRT 终端。

FORTRAN 语言主要用来书写数学计算和其他数据运算的程序。FORTRAN 的名称是取自 FORMULA TRAN slator 两个字的前半部分。

用 FORTRAN 语言写的源程序是由一组语句所组成。这些语句由程序员按本刊物中所叙述的语言单元构成。

在编译的过程中，被称为 FORTRAN 编译程序的程序分析源程序的语句，并把它翻译成称为目标程序的机口语言。在使用 M6800 连接装配程序 (Linking Loader) 把目标程序连接之后，这个程序能够在 Motorola M6800 微处理机上执行。此外，当 FORTRAN 编译程序在源程序中检测到错误时，它产生适当的诊断错误信息。注意：可以将 FORTRAN 和汇编程序连接在一起。

1.1 驻留 FORTRAN 编译程序用 EDOS-II FORT 命令调用。这个命令和它的参数定义如下：

格式：

FORT, list, object, source-1, source-2, ..., source-n

用途：

用 M6800 FORTRAN 编译程序来编译 FORTRAN 源文件，并产生一份清单（任选）和可重新定位的目标文件（任选）。

注释：

表格 (list) —— 表格文件或设备（允许设备：

#LP 一行式打印机，#CN 控制台）。

假如选择一个表格文件，那么就不能指定一个目标文件。假如项目是空白，就没有表格产生。

#LP8 或 #CN8 使状态编译记录被编译，参看 2.2 节。

#LP5 或 #CN5 将禁止页式源表格。

目标 (object) —— 可重新定位的目标文件。

假如一个表格文件被指定，则不能指定一个目标文件。

假如项目是空白，则没有目标文件产生。

源 (source)i —— 装有源程序文件。多于一个的程序或子例程子程序，可以被指定作为源输入，一个源文件可以包含一个程序的一部份或若干个程序。

例：

FORT, #LP, OBJ, PROG1

在行式打印机上产生一个包含源程序的表格和符号表，并产生一个对应于源程序文件“PROG1”的一个可重新定位的目标文件——“OBJ”

1.2 抹去行和字符

在制作源程序期间，一个在？行被错误送入的数据行可用先按下CTRL (Control) 键并保持，然后按X键抹去。这将抹去数据，在进行回车 (carriage return) 和进行 (line feed) 操作后即可重新输入。一个字符的抹去可用按下 CTRL (Control) 键并保持，然后按DEL或Rubout (错误) 键实现，每按一次DEL或rubout键抹去一个字符。

1.3 驻留FORTRAN编译程序由MDOS FORT命令调用，这个命令调用，这个命令简单定义如下：

格式：

FORT, SOURCE-1, SOURCE-2, ---, SOURCE-N

用途：

为了编译用M6800 FORTRAN 语言写的源程序语句，并产生一个表格 (任选) 和可重新定位目标文件 (任选)。

注释：

SOURCE-i --- 装有源程序文件。

多于一个的程序或子例程序，可以被指定为源输入。一个源文件可以包括程序的一部分或若干个程序。一个可重新定位的目标文件和源 (SOURCE-i) 文件有相同的名字。而其后缀为 "R0"。假如源文件不只一个时，输出将串在一起。

对于MDOS FORT命令的完整的叙述，请参看MDOS命令手册。

例：

FORT PROG1

这是在行式打印机上产生一个包括源程序的清单，并对源程序文件 PROG1 产生一个可重新定位的目标文件 PROG1 R0。

第二章 FORTRAN 编译程序

2.1 语句

源程序由一组语句组成，编译程序根据这些语句产生机内指令、常数和存储区域。一个给出的FORTRAN 语句有效地执行下面三种功能中的一种：

1. 完成一定的操作（即加法，乘法，转移。
2. 指定正在被处理的数据的性质。
3. 指定源程序的特性。

FORTRAN 语句通常是由用一定的FORTRAN 关键字连同基本语言单元：常数，变数和表达式所组成。FORTRAN 语句有如下的类型：

1. 算术赋值语句：

这些语句可进行运算及检查条件。其结果置换一个被指定的变数或带下标变数的现行值。

2. 控制语句：

这些语句使用户有可能控制目标程序的流程和结束目标程序的执行。

3. 输入/输出语句：

这些语句使用户有可能在内存与终端或磁盘之间传送数据。

4. FORMAT 语句：

这些语句和输入/输出语句一起用来指定一个FORTRAN 记录的形式。

5. DATA 预置语句：

该语句给变数赋予初始值。

6. 说明语句：

这些语句是用来说明变数和数组的性质。

7. 子程序语句：

这些语句使用户能够命名和定义函数及子例程子程序。子例程子程序作为一个源文件而跟在主程序后面进行编译或做为一个独立的文件，不与主程序在一起。

8. OPENF/CLOSEF语句：

这些语句与磁盘 I/O 一起用来给一个文件名赋予 FORTRAN 文件标号并设置 EDOS 指示字。

2.2 编写 FORTRAN 语句：

一个 FORTRAN 流程序的语句能以每行显示 72 列的方式送入 CRT 终端。假如一个语句对一行来说是太长了，则可在后面接续的每一行的第一列上使用符号 & 来继续这个语句。

为了便于阅读，在关键字和变量名之间可写上许多空格，每一个关键字必须至少跟有一个空格。并在文字数据中的空格作为数据内部的空格对待并被保留着。在变量名、关键字及数据中不能包含空格。

~~解释~~程序的注释可写在第一列放有 C 字母的那一行的第 2 - 72 列中，注释可出现在任何 FORTRAN 语句之间；一个注释行不能直接放在一个继续行之前。FORTRAN 编译程序是不理会注释的，为了便于阅读，空格可以在所希望的地方插入。

表示一个注释记录的 C，表示语句连续的字符 &，和作为条件编译的 X 都必须开始于第一列。假如 X 在第一列，调用编译时，此记录被看作一个注释。但在用到 #LP8 或 #CNS 指令这种情况下，第一列中有 X 的记录将被编译。语句标号的范围是从 1 到 99999，标号也在第一列开始，它的后面至少跟着一个空格，所有其他语句可以在第 2 - 72 列的任何地方开始。

2.3 语言单元：

语言的基本单元在下面各节中讨论。下章讨论这些单元在实际的 FORTRAN 语句中的使用。

程序单元这个词指的是一个主程序或一个子程序。

FORTRAN程序单元的次序如下:

1. 子程序语句, 假如有的话。
2. COMMON 和 DIMENSION 语句, 假如有的话, 它们可以互相混合。
3. DATA语句, 假如有的话。
4. 执行语句, 至少必须有一个。
5. END 语句。

FORMAT 语句和 DATA 语句可以出现在END语句之前的任何地方。但 DATA 语句必须跟在任何包含有同样变量或数组名字的说明语句后面。

2.4 常数

常数是固定的, 不变的量。常数有两类——一种表示数(数值常数), 另一种表示文字数据(文字常数)。

数值常数可以是整数或实数; 而文字常数可以是一串字母数字或是特殊的字符。

2.4.1 整常数

定义: 一个整常数是指整个数没有引进小数点。它占有存储空间的二个字节。最大值是32,767($2^{15}-1$)。

一个整常数可以是正, 0 或负; 假如没有符号, 就假设它是正。它的数值必须不大于最大值和必须不嵌入逗号。

例:

正确的整常数

0

90

173

-21474

不正确的整常数

27. (包含1个小数点)

3145903612 (超过允许范围)

5.936 (包含一个嵌入逗号)

2.4.2 实常数

定义: 一个实常数有如下三种形式中的一种: 基本实常数, 基本实常数后面跟着一个十进制指数, 或一个整常数跟着一个小数点。一个实常数有四个存储字节。

一个基本实常数是一个带有小数点, 可多达八位的十进制数的数字串。基本实常数占有四个存储单元(字节)。

实常数的值是0, 或是 10^{-05} (近似 10^{-78}) 到 10^{63} (近似 10^{75})。

采用四个字节的数据精度是六位十六进制数字(近似7.2个十进制数字)。

一个实常数可以是正, 0 或负(假如无符号, 就设定为正)和必须是允许的值。它不能嵌入逗号。十进制指数允许把一个实常数的表达式当作一个基本实常数或整常数乘以10的所需要的幂的取积。

例:

正确的实常数

+0

-999.9999

7.0E0 (即 $7.0 \times 10^0 = 7.0$)

19761.25E1 (即 $19761.25 \times 10 = 197612.5$)

7.E3

7.0E3 (即 $7.0 \times 10^3 = 7000.0$)

7E-03 (即 $7.0 \times 10^{-3} = 0.007$)

不正确的实常数

1 (漏了1个小数点或一个十进制指数)

3,471.1 (嵌入一个逗号)

1.E (在E后面, 漏了1或2位整常数。注意: 它不能解释为 1.0×10^0)。

1.2E113 (E后面跟着三位整数)

23.5E97 ($23.5 \times 10^{97} > 10^{63}$, 数值在允许范围外)

21.3E-90 ($21.3 \times 10^{-90} < 10^{-65}$, 数值在允许范围外)

2.4.3 文字常数

定义: 一个文字常数是串括在撇号中的字母数字或特殊的字符。

这一串字符可以包括任何字符, 各个字符需要存储四一个字节, 串中的字符数所包括的空格不能多于 72 个。

文字可用在 FORMAT, DATA 和赋值语句中, 文字也可以在 CALL 语句中用作实际变元, 但用在 DATA 和赋值语句的文字和作为 CALL 语句实际变元的文字被限制在两个字节内, 并只能与整数一起使用。

例:

'DATA'

'X-COORDINATE Y-COORDINATE Z-COORDINATE'

'5.14'

K='AB'

2.5 标识符

定义: 标识符可用 1-6 个字母数字字符 (即 0-9) 数字和 A 到 Z 字母) 表示, 它的第一个字符必须是字母。

例如: GOTO, IF, FORMAT 等之元素都不能用做标识符, 所有这些元素都被认为是保留字。

说明: 标识符用在一个程序段 (即一个主程序或一个子程序) 中, 以区别下列各种元素。

- 数组和该数组的元素 (见“数组”)
- 变量 (见“变量”)
- 内部函数
- 函数子程序 (见“函数子程序”)
- 子例子程序 (见“子例子程序”)

在一个程序段中的一种类型里，标识符必须是唯一的，并除了下面情况外，一个标识符只能识别一种类型的元素。

一个FUNCTION子程序名字必须是一个在FUNCTION子程序中的变量名。

在一个可执行程序的任何一段中用了一个标识符或一个外部程序名字，则在那个可执行程序的其他段中就不能以任何其他方法使用那个名字来识别这些类型中的一个实体。

2.6 变量

一个FORTRAN变量是一个占有一定存储区域的量的符号表示法。用变量名指定的值总是存储在該区域中的一个现行值。

例如：在语句中

$$A = 5.0 + B$$

A和B两者都是变量名。B的值取决于前面某一个语句，并且往往是可以改变的。无论什么时候执行了这个语句都将计算出A值，当B值改变时A值也跟着改变。

2.6.1 变量名

使用有意义的变量名能有助于编制程序。

即是说：使除了程序员以外的其他人也可以看懂程序并且明瞭它的功能。例如计算一辆汽车在给定的时间内，以给定的速率行驶的距离。其计算方程式可写成：

$$X = Y * Z$$

其中，*表示乘号。但假如程序员写成：

$$DIST = RATE * TIME$$

对这个方程式的某个人将更有意义。

例：在下面的程序段中，变量名是：

在循环的变量名

2925

RATE

VAR

不正确的变量名

B292704 (含有多于6个的字符)

4ARRAY (第一个字符不是字母)

SI X (含有一个特殊字符)

2.6.2 变量类型和字长

一个变量的类型对应于表示变量数据的类型。因此，一个整型变量表示整型数据或文字数据，一个实变量表示一个实型数。

对于不同类型的变量，有对应的保存变量的存储单元(字节)数目。下面表示出各变量类型的字长：

变量类型	字长
整 数	2
实 数	4

2.6.3 预先定义的类型说明

通常用预先定义变量来说明变量是整数或实数。

1. 假如变量名的第一个字符是 I, J, K, L, M 或 N, 则变量字长为2的整数。
2. 假如变量名的第一个字符是其他字母字符, 则变量是字长为4的实数。

这个规定是含蓄地说明一个变量是整型数还是实型数的传统FORTRAN 方法。在本刊物后面所有的例子中都应用这个规定。

2.7 数组

说明：一个FORTRAN 数组是一组变量，这组变量用单一的变量名来识别。在数组中，某一个变量可以按它在数组中的位置来称呼(例如，第一个变量，第三个变量，第七个变量等等)。例如命名为 NEXT 的数组，是由五个变量组成。每个变量的现行值如下：273, 41, 8976, 59, 和 2。

NEXT (I) 是装有273 的单元

NEXT(2) 是装有 41 的单元
 NEXT(3) 是装有 8976 的单元
 NEXT(4) 是装有 59 的单元
 NEXT(5) 是装有 2 的单元

在这个数组中，各个变易（元素）由数组的名字（即 NEXT）和后面一个在括号中的数（称为下标变易）所组成。数组内的变易称为下标变易。因此，下标变易 NEXT(1) 具有值 273，下标变易 NEXT(2) 具有 41 等等。

下标变易 NEXT(I) 系指在数组中第 I 个下标变易，其中，I 是一个整型变易，它的值可假设为 1, 2, 3, 4 或 5。若涉及数组中的任何元素，则数组名必须带下标。特别指出的是单独数组名不代表第一个元素。

考虑下面命名为 LIST 的数组，它用二个下标变易来描述。第一个下标从 1 到 5，第二个下标从 1 到 3。

	列 1	列 2	列 3
行 1	82	4	7
行 2	12	13	14
行 3	91	1	31
行 4	24	16	10
行 5	2	8	2

假如指定第 2 行第 3 列的数，它是：

LIST(2,3)

因而，LIST(2,3) 具有值 14，而 LIST(4,1) 具有值 24。普通的数学符号用 LIST 表示数组 LIST 的任何元素。在 FORTRAN 中，它被写成 LIST(I,J)，其中 I 等于 1, 2, 3, 4 或 5，而 J 等于 1, 2 或 3。

2.7.1 说明一个数组的规模和类型。

一个数组的规模（元素的数目）是用数组下标的数目和各个标的最大值规定的。对于 FORTRAN 程序中的所有数组在使