YOS—LD FORTRAN

說　明　书

重庆钢铁设计研究院

计　算　机　科

YOS-LD FORTRAN

说明书

目　　录

此为试读，需要完整PDF请访问：www.ertongbook.com

1．前　言

YOS-LD FORTRAN 在 YOS-LD监控管理下起作用。YOS-LD FORTRAN的源程序用纸带或卡片输入通过编译，变换成YOS-LD规定的目标程序。在执行时，用连接装配程序（LINKLD）或连接编辑程序（LINKEDT），首先作被编译的目的程序和库程序间的结合，然后变换成装配模块。YOS-LD FORTRAN可以用日常所使用的近似于数式的表现形式编制程序。由于这个道理，它不仅在解决科学技术计算等问题上发挥了很大的作用，而且具有便于编制实时处理程序的功能，所以在编制过程控制程序等方面也发挥了巨大的威力。

有关资料

YOS-LD FORTRAN 程序库说明书

FORTRAN 宏程序库

FORTRAN实时例行子程序程序库（FRTSL）

FORTRAN 科学用例行子程序库（FSSL）

2．FORTRAN 概要

2．1 面向科学技术计算和过程控制的 FORTRAN

YODIC计算机是可以用于过程控制和科学计算的各种系统的多用途计算机。YOS-LD FORTRAN备有满足这种多用途的功能。

在科学计算和控制程序中，要求FORTRAN在语言上、操作上有很多不同的功能。比如在科学计算中，简单而又便于程序化地计算各种问题，很有必要。FORTRAN 操作的容易性特别重要。另一方面，在控制程序中，与其在某种程度上长期利用同一程序，利用其操作性，不如发挥巧妙程序上的技术，在多重级中进行动作任务

间的数据传送和各个I／O的数据交换等，但各种数据处理程序必须一次存入，可以发挥程序编制上的技巧，以达到足够的功能。

YOS-LD FORTRAN 具有这两方面的功能，用户可以自由选择并利用其功能。

2．2 YOS-LD FORTRAN 的特点

YOS-LD FORTRAN以JIS 5000为基础，但在其中增添了各种扩充功能，特别是作为过程控制发挥了很大的作用。主要特点如下。

(1)可以使用位数据和压缩数据，可以记述复杂的数据构造。

(2)可以指定2进制数和16进制数，可以做各种逻辑运算。

(3)可以做屏蔽数据的位抽出运算

(4)程序可以具有多个入口。而且例行子程序的分支形返回也被确认。

(5)可以定义和输入输出直接存取文件。

(6)外部符号被确认。

(7)与标准过程输入输出数据包POPP密切相关，由于被扫描的过程输入输出数据使用特殊英文字母名称，所以能够很容易地存取。

(8)准备了调整用语句。

(9)可以利用直接插入(In line)汇编的功能。即在FORTRAN语句间写汇编语句。YAP-LD的宏功能可以表示FORTRAN以上的面向问题的表现形式和拐弯的有效表现形式。由于它和FORTRAN密切相关，所以成了比FORTRAN更强有力的程序语言。

(10)为了使用FORTRAN记述的通过编译得到的目的程序和用汇编语言(YAP-LD)记述的通过汇编所得到的目的程序的形式完

全相同，必须使用FORTRAN 书写的程序和用汇编书写的程序能够容易地结合。

(11)可以用FORTRAN语句记述程序的复盖构造，也容易编制比磁心中所分配的使用区域大的程序。

(12)可以用FORTRAN 语句记述系统例行子程序，即可以得到与程序在磁心上的存放位置无关的形式的程序。

(13)可以用汇编记述的系统例行子程序。进入例行子程序时可以直接把数据置入寄存器，从例行子程序返回时可以直接访问寄存器中的数据。

(14)使装入的目的程序最佳化，以便缩小存储空间，提高处理速度。

3 · FORTRAN 源程序的写法

3 · 1 语句，行和字符

FORTRAN源程序由一串语句和行所构成。

FORTRAN〃〃〃的每一个语句在编码表的第7列到72列之间书写。一个语句一行写不完时，第二行的第6列写上0和空白以外的字符，可以从第7列开始继续书写前一行未写完的内容。在这种情况，第1行叫做初始行，接下来的其它行叫做继续行。一个语句包括空白在内最多可以书写660个字符。

编码表的第1列记入字符C，可以作为程序内的注释行。如果第1列是C，则从第2列开始写入任意的注释。

注释行对程序的执行不带来任何影响，只是为了读写程序方便才使用的。

第1列为"/"的行称作扩展行，是为了插入用YAP汇编语

盲书写的程序而使用的，还可以书写第1列记入“+”号的调整语句。

为了表示程序的结束，程序的最后置E N D行。从编码表的第7列开始书写E N D。

用卡片穿孔源程序时，７０～８０列用于程序的识别和卡片的顺序确认，不作为语句的一部分。此栏记入的字符，在卡片的排列顺序中，是把其内部处理代码代换成原来数值时的值，但必须顺序递增。

可以用于FORTRAN的字符，是英文字母（Ａ～Ｚ），数字（０～９）和表３·１所示的特殊字符。把英文字母和数字的组合叫做字母数字。

	空白	(	左括号	,	引用符
=	等号	)	右〃〃	@	a 圈
+	C 加号	,	逗号	<	小于号
−	减号	.	点号	>	大〃〃
＊	星号	#	#〃	:	冒　号
／	斜线	&	and 号	〃	引　号

（注）纸带上出现的(TAB)和(NL)参阅１２·３章。

表３·１

为了阅读语句方便，在FORTRAN源程序的语句（７～７２列）中，空白可以自由便用。在字符常数或空行里面，空白作为标准的字符列，所便用的地方具有作为一个字符的意义，但在此以外的地方全部忽略不计。总之不作为隔开记号。

此为试读，需要完整PDF请访问：www.ertongbook.com

下面3种写法，都表示相同的语句

```
GO␣TO␣36
GO    TO    36
␣␣GOTO␣036
```

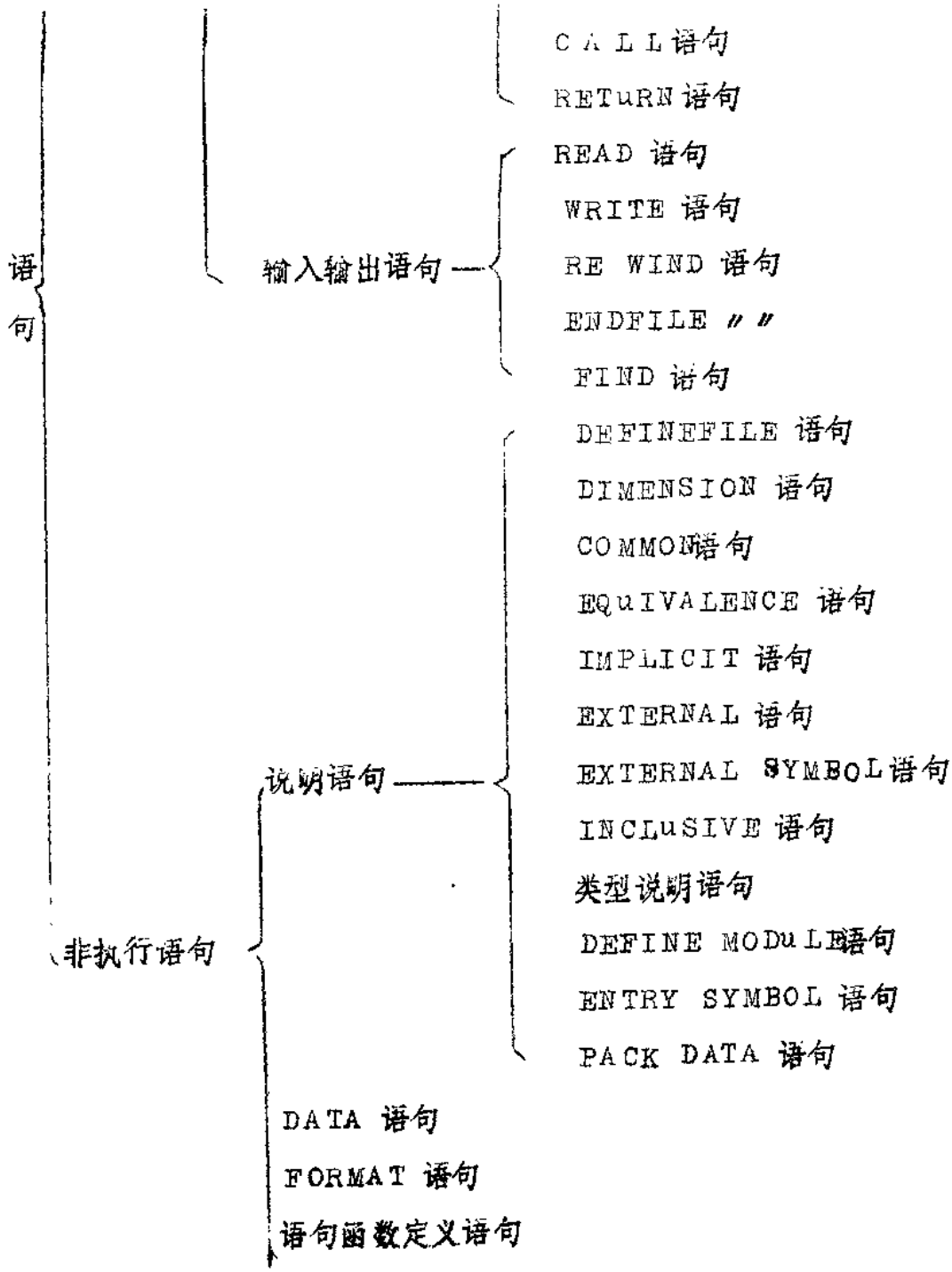
3．2 语句的构成

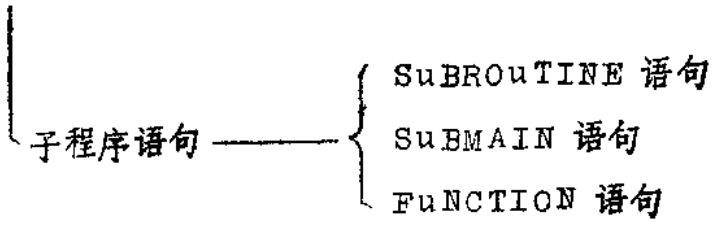
语句有表3．2中的各种语句。在语句中带有可以由其它语句访问的标号。把这种标号叫做语句标号，用1～99999的数字表示。在语句标号中，前面的0写不写是一样的。语句标号写在语句的首部（1列～5列之间）。

在同一程序内不允许带同一个语句标号。

在语句中有执行语句和非执行语句，执行语句是指定动作的语句，非执行语句是记述数据的特性及排列方式，编辑的信息，语句函数，程序单位的分类的语句，象表3．2这样分类。

- 执行语句
  - 赋值语句－算术（逻辑）赋值语句
  - 控制语句
    - ASSIGN语句
    - 无条件GO TO语句
    - 计算型GO TO语句
    - 赋值GO TO语句
    - 算术IF语句
    - 逻辑IF语句
    - DO语句
    - CONTINUE 语句
    - STOP语句
    - PAUSE语句


语句
非执行语句
输入输出语句
CALL语句
RETURN语句
READ 语句
WRITE 语句
RE WIND 语句
ENDFILE 〃〃
FIND 语句
说明语句
DEFINEFILE 语句
DIMENSION 语句
COMMON语句
EQUIVALENCE 语句
IMPLICIT 语句
EXTERNAL 语句
EXTERNAL SYMBOL语句
INCLUSIVE 语句
类型说明语句
DEFINE MODULE语句
ENTRY SYMBOL 语句
PACK DATA 语句
DATA 语句
FORMAT 语句
语句函数定义语句



（表3·2）

执行语句可以带有需要的语句标号，但非执行语句，除FORMAT语句外，不可以带语句标号。

FORMAT 语句必须带一定的语句标号。

3·3程序构成

把语句和END行以及扩展行的集合叫做程序。在此不管加进去的注释行。

3·3·1程序本体

在执行语句、DATA语句和语句函数定义语句之后带END行的部分或在它之前带有几个说明语句和FORMAT 语句的部分叫做程序本体。

程序本体至少必须有一个执行语句，但是FORMAT语句，DATA语句和语句函数定义语句不管有没有。语句函数定义语句必须在第一个执行语句之前，DATA语句必须在最后一个说明语句之后。FORMAT语句不管其位置在什么地方。

3·3·2程序单位

把由一个程序本体构成的程序称作主程序，把书写在SuBROuTINE 语句、FuNCTION语句或SuBMAIN语句后面的程序本体的单位称作子程序。主程序或子程序都叫做程序单位。在主程序的前面可以书写DEFINE MODuLE 语句。

3·3·3源程序的构成

源程序由一个以上的主程序，或一个以上的主程序和几个用SUBMAIN语句以外的语句开头的子程序和外部程序，或一个用SUBMATN语句开头的子程序所构成。源程序的单位在前面用来FILE␣××××××，在最后面用来FEND 表示。在此××××××是源模块名称，可以与来FILE 一同省略。

一旦编译源程序，则按程序单位制作目的程序，称此目的程序为目的模块。

3・4 编制程序的方法和编制程序的例子

YOS-LD FORTRAN 准备了专用编码表，编码表由下面的栏构成。各行的1～6列是书写语句标号的栏，6列是写继续行标志的栏，7～72列是记述语句的栏，73～80列是用来书写卡片识别顺序的栏。

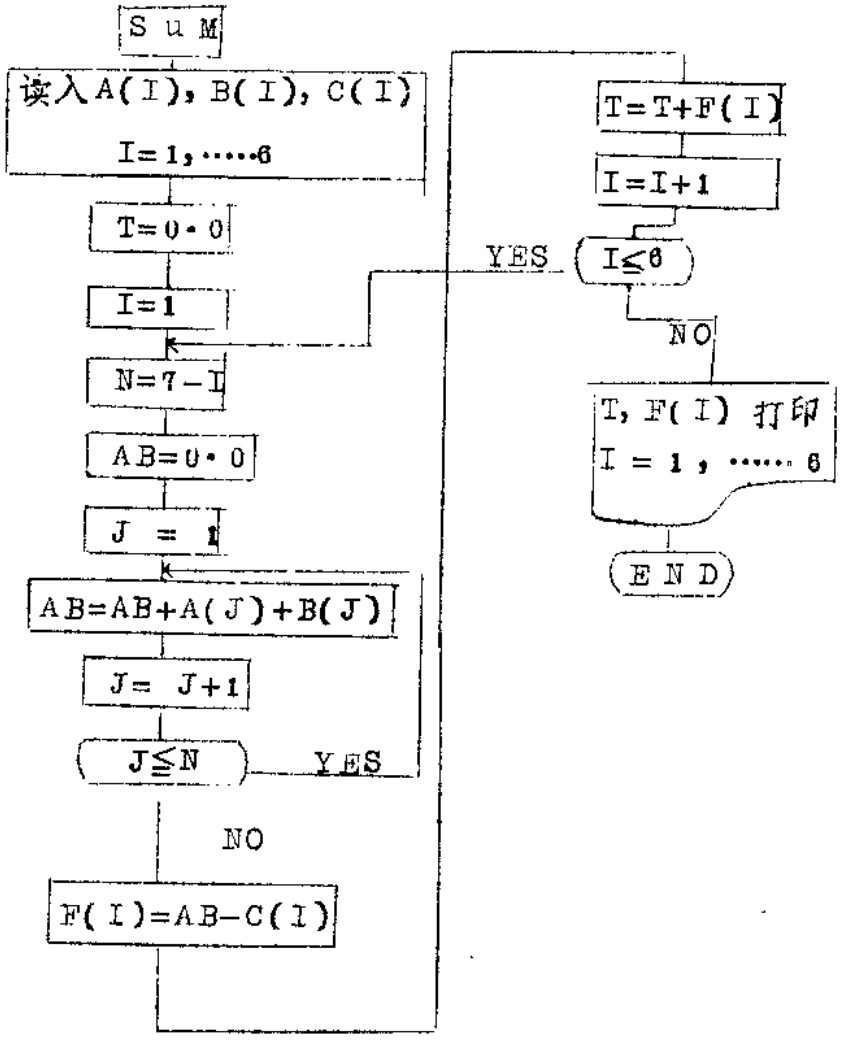
把写在编码表上的源程序在卡片或纸带上穿孔时的注意事项，参考12・3章。

在这里，表示了用YOS-LD FORTRAN记述的简单程序的例子。关于这个例子中的各种语句由下一章说明。

例1・读入数据A_i，B_i，C_i（$i=1, 2, \cdots\cdots 6$），进行

$$F_i=\sum_{j=1}^{i+j=7}(A_i+B_i)-C_i,\quad T=\sum_{i=1}^{6}F_i$$ 计算，输出

打印T，F_i（$i=1, 2, \cdots\cdots 6$）。


S u M
读入A(I), B(I), C(I)
I=1, ……6
T=0.0
I=1
N=7-I
AB=0.0
J = 1
AB=AB+A(J)+B(J)
J= J+1
J≦N
YES
NO
F(I)=AB-C(I)
T=T+F(I)
I=I+1
YES
I≦6
NO
T, F(I) 打印
I = 1, …… 6
E N D

```
      DEFINE    MODuLE    EXMPL 1
C   PROBLEM F(I)=SUM(A(J)+B(J))-C(I)
C             T=SuM(F(I))
      DIMENSION A(6), B(6), C(6) F(6)
      READ (1, 10) (A(I), I=1, 6)
      READ(1, 10) (B(I), I=1, 6)
      READ(1, 10) (C(I), I=1, 6)
   10 FORMAT (6F5.2)
      T=0.0
      DO  30  I=1, 6
      N=7-I
      AB=0.0
      DO  20  J=1, N
   20 AB=AB+A(J)+B(J)
      F(I)=AB-C(I)
   30 T=T+F(I)
      WRITE(7, 40) T, F
   40 FORMAT (1H1, 7 F8.2)
      STOP
      END
```


此为试读,需要完整PDF请访问: www.ertongbook.com