TQ—16机

算法语言簡介

北京市計算中心編印

1975.5.

前　　言

一、TQ－16计算机在出厂时，配有一套编译系统。为方便上机单位，我们和科学院计算所、浙江省计算所协作，正在编制TQ－16机的另一编译系统。不久，北京市计算中心的TQ－16机将同时使用上述两套编译系统。为区别起见，我们把厂里提供的这套编译系统称为TQ－16机编译系统（I）。

二、这本资料介绍的是编译系统（I）所接受的算法语言。它应和我们已编印的《TQ－16机编译系统（I）使用说明》配套使用。

三、我们编写这本资料的依据是：

《怎样用算法语言编程序（X－2计算机、709计算机）》　　（上海人民出版社）

《集成电路中型通用电子数字计算机TQ－16编译系统使用手册》　（上海无线电十三厂）

《集成电路中型通用电子数字计算机TQ－16编译系统标准算法》　（上海无线电十三厂）

由于我们水平有限，又没有推广、使用编译系统（I）的经验，所以，这本资料会有许多错误和不足之处，希望同志们多提宝贵意见。

北京市计算中心

一九七五年五月

目　　录

§1　变量与表达式 …… (1)

§1.1　基本符号 …… (1)

§1.2　标识符 …… (2)

§1.3　常量　简单变量 …… (3)

§1.4　数组和下标变量 …… (5)

§1.5　函数和标准函数 …… (6)

§1.6　简单表达式 …… (11)

§1.7　条件表达式 …… (15)

§1.8　表达式语法小结 …… (17)

§2　基本语句 …… (18)

§2.1　赋值语句 …… (20)

§2.2　转语句　标号和开关 命名表达式 …… (21)

§2.3　条件语句 …… (24)

§2.4　循环语句 …… (25)

§2.5　空语句 …… (28)

§2.6　停语句 …… (29)

§3　说明与分程序 …… (30)

§3.1　复合语句与分程序 …… (30)

§3.2　简单变量说明 …… (31)

§ 3.3 数组说明 …… (32)
§ 3.4 开关说明 …… (36)
§ 3.5 过程说明和过程语句 …… (37)
§ 3.6 函数过程 …… (45)
§3. 7 标准过程和库过程 …… (50)
§ 3.8 量和它的作用域 …… (54)
§ 3.9 语句语法小结和程序中的註解 …… (59)

§ 4 关于代码交换的标准过程语句 …… (62)
§ 4.1 输入语句 …… (62)
§ 4.2 写鼓、读鼓语句 …… (64)
§ 4.3 写带、读带语句，付本语句 …… (66)
§ 4.4 输出语句 …… (68)

附录1：库过程一览表 …… (82)
附录2：TQ—16机的主要性能指标和指令简表 …… (97)

§一 变量与表达式

§1.1 基本符号

用算法语言编写的程序叫做源程序。源程序由算法语言的一系列基本符号组成。本算法语言的基本符号共有85个：

1。字母26个：A, B, C, D, E, F, G, H, I, J, K, L, M, N, O, P, Q, R, S, T, U, V, W, X, Y, Z；

2。数字10个：0, 1, 2, 3, 4, 5, 6, 7, 8, 9；

3。逻辑值（布尔值）2个：TRUE, FALSE；

4。算术运算符6个：+, −, *, /, ÷, ↑；

5。逻辑运算（布尔运算）符3个：∨, ∧, ¬；

6。关系运算符6个：<, ≤, =, ≥, >, ≠；

7。顺序运算符8个：GOTO, IF, THEN, ELSE, FOR, DO, STEP, PAUSE；

8。括号符7个：(,), [,], BEGIN, END, ●；

9。说明符7个：REAL, INTEGER, BOOLEAN, ARRAY, SWITCH, PROCEDURE, KU；

10。分隔符9个：., ', :, ;, ␣, #, STEP, UNTIL, COMMENT；

11。分类符1个：VALUE。

这些基本符号的意义和用途将在以后各节逐步介绍。这里只指出，

除了上述基本符号以外，源程序中不允许再使用任何别的符号，任何其它符号都一律视为非法符号。

註 1。字母无大写小写之分，例如A与a同。

2。由字母组合而成的基本符号叫做字定义符或组体定义符。上述字定义符均略写了前后两个 e，例如：BEGIN实际上应写作 eBEGINe。在以后各节介绍语言时，为了书写简单，我们同样也省略了 e，有时为了避免同标示符混淆，在这些符号下面加上了一横。但在源程序的正规书写特别是源程序的纸带穿孔时，这些 e 无论如何不能省略。

3。为了避免字母O与数字0及字母X与乘号×混淆，有时写字母O作O，写乘号×为*。

§1。2 标识符

一个完整的程序一般都引进一系列变量、数组、过程、开关和标号等，为了识别它们，需要一一给它们取个名字，标识符就是用来给它们取名字的。

标识符是由字母开头的任意字母、数字串。原则上一个标识符所包含的字符个数不限，但规定只有前八个字符具有区分意义。

例如：

A，A1，MAX，B4C3，YMAXAIJK，YMAXAIJKE

中的每一个都可用作标识符。但后两个标识符被视为相同的，而2A

和 EX—1 不能取作标识符。

§1.3 常量 简单变量

1. 常量指常数和逻辑值（布尔值）。常数就是通常的数，在源程序中以习惯的十进制数出现，书写方式也与习惯一致，由数符、整数部分，小数部分构成，例如：

2，0.1，−3.4，+5，100000，0.00001，−123.56。此外，在源程序中也允许写成指数表示形式，例如上面各数也可以写成

$0.2_{10}+1$，$1_{10}-1$，$-0.34_{10}1$，$0.5_{10}+1$，$0.1_{10}+6$，$0.1_{10}-4$，$-0.123456_{10}+4$。

因此同一个数的表示形式不是唯一的，按规定，当数为正时，正号可以省略，在指数形式表示中$_{10}$后面的阶符为正时也可以省略。

逻辑值只有两个，即TRUE和FALSE，分别表示“真”和“假”。在源程序中，逻辑值就由英文字TRUE和FALSE表示。

常数用做算术运算，能出现在算术表达式中。逻辑值只能参加逻辑运算（布尔运算），即只能出现在布尔表达式中。

2. 变量包括简单变量和下标变量两种。简单变量（简变）用一标识符代表。例如设A，B是简变，则A+B就表示简变A的值加上简变B的值。简单变量不仅可以象常量那样用作运算对象，也可以作赋值对象，例如设C也是简变，则C:=A+B就表示将A+B的值

赋给简变C。

与常量不同，简变必须先说明（定义）后使用，而常量则无须特别说明，由它的表示形式自明。还有四个特定的简单变量——标准变量，其标识符是

H00000，H00001，H00002，H00003；

或

G00000，G00001，G00002，G00003。

标准变量无须说明即可使用，这是与一般的简变不同之处。注意，四个标准变量的每一个都对应于两个标识符，分别以H开头或以G开头。当且仅当在用电传命令向标准变量赋值时，H和G才有不同的含义，它们分别表示以2——10进制或8进制数形式赋值。在源程序中H0000I与G0000I无任何区别。

简单变量本身还分三种类型，实型、整型和布尔型。一个简变究竟属于哪一种类型由<类型说明>指定（见§3）。ALGOL系统规定不同类型的简变只能取不同类型的值：布尔型只能取布尔值，整型只能取整型值，实型可以取实型值。然而由于本机编译系统对实型数和整型数在机器上的处理是一样的，所以在应用时可以对这两种类型不加以区分。一个简单变量在<类型说明>中被说明是整型还是实型不是本质的，主要看其值是整数还是非整数，只要保证应该是整数时，相应的简变之值是整数就行了，而不管这个简变属于哪一种类型。当然在编写源程序中对要求取值为不同型的简变作不同的类型说明也有醒目的作用，比如它提醒编程序的人考虑一下在整型简变出现

时其值是否是整数!

后面在介绍表达式和语句时，凡说到<简变>时一般是泛指三种类型简变中的任何一种。但由于在算术运算中，只能出现实型或整型变量，而在布尔运算中只能出现布尔型变量，所以在不致于模糊的地方我们不再特别指明简变的类型。

§1.4 数组与下标变量

可以把若干个简变合起来用一个数组表示，而数组中的任何一个分量用下标变量（下变）代表。因此下标变量总是和相应的数组相联系的。

例如设有20个量X1，X2，……，X20，如果我们定义一个数组，可以表成A〔1：20〕，其中A是我们给这个数组起的名字，称为数组标识符，而方括号中的1：20表示该数组的大小为20，下界为1，上界为20。这时，为了表示数组的第12个分量X12，我们可以使用下标变量A〔12〕。当然这个数组也可以定义作B〔0：19〕或C〔-1：18〕，可是要注意，此时为了表示该数组的第12个分量X12，我们应当用B〔11〕或C〔10〕。

刚才说的这些数组称为一维数组，还允许用二维、三维……数组。例如将上面这20个量定义为一个二维数组可以写成D〔1：5,1：4〕或E〔1：2，1：10〕，如果排列次序仍为X1，X2,……X20，则表示X11的下标变量分别是D〔3，3〕，E〔2,1〕。

数学中的矩阵相当于一个二维数组。例如设有M行N列矩阵

$$A=\begin{pmatrix} A_{11}, A_{12}, \cdots\cdots, A_{1n} \\ A_{21}, A_{22}, \cdots\cdots, A_{2n} \\ \cdots\cdots \\ A_{m1}, A_{m2}, \cdots\cdots, A_{mn} \end{pmatrix}$$

我们能用一个数组A〔1：M，1：N〕代表它的M*N个元素A11，……，AMN。这些元素在数组A中的排列次序规定为

A11，A12，……，A1N，A21，A22，…，AM1，……，Amn，

即一行一行的排下来，而不是一列一列的排。

N维数组的一般形式是：

A〔A1：B1，A2：B2，……，An：Bn〕，

其中A是数组标识符，Ai，Bi取整数值，且要求Ai≤Bi，i=1，……，n。这个数组共包含（B1－A1＋1）·（B2－A2＋1）·…·（Bn－An＋1）个分量。与此数组相应的下标变量有形式：

A〔L1，L2，……，Ln〕，

其中，下标L1，L2，……，Ln只能取整数值，且满足不等式

Ai≤Li≤Bi，i=1，2，……，n。

下标变量在源程序中的地位与简变很相似，即凡能用简变的地方都可用下变，只是内外存代码交换标准过程语句中的<可交换量>及

循环语句中的<循环参数>不允许是下变。

数组也象简变那样有实型、整型和布尔型之分，由<数组说明>指定（见§3．3），从而相应的下标变量也分为三种类型。上段关于简单变量类型的附註对数组、下标变量也适用。

§1．5 函数和标准函数

在数值计算中，经常要重复地计算某一个函数在不同自变量下的值。如果相应于每一自变量的函数计算都编写一段程序，这不仅很不经济，而且当用到该函数的地方很多时，由于计算机容量的限制使之实际上无法实现。为了解决这个问题，算法语言中引进了函数过程的概念，这时我们只须对一种函数编写一个<函数过程说明>，而在用到它的地方，只要写上该函数的名字及其自变量就行了（这称为函数命名符，以下为了方便，仍简称为函数）。

关于一般的函数将在§3．6专门介绍，本段只介绍标准函数的应用。所谓标准函数就是编译系统已经为用户准备好的那些最经常用到的函数，包括ＳＩＮＸ，√X等初等函数。用户如果用到某一标准函数，则无须编写函数过程说明就可以调用。为了区别自行定义的一般的函数，在所有标准函数名字（函数标识符）前面一律冠以＃号。调用的办法是先写上函数的名字，名字后面给出自变量并用圆括号括起来，例如ＳＩＮ0．5写作＃ＳＩＮ（0．5），$\sqrt{5.2}$ 写作＃SQRT（5．2）等等。作为自变量的可以是常量、变量（简变或下变），

也可以是表达式（§ 1。6）和函数本身。本编译系统一共提供了二十二个标准函数，详见表1。1。

依赖于函数所取的值的类型，相应地把函数也分作三类：实型、整型或布尔型的。标准函数的类型是已经确定了的，用户自行定义的一般函数，则必须在函数过程说明中指定其类型（见§ 3。6）。既然函数作为一个值，因此只能出现在算术表达式和布尔表达式中。例如可以写：

#SIN(X)+#SIN(Y)*#SQRT(X+Y)—Y

它实现 $SINX+SINY\cdot\sqrt{X+Y}-Y$ 的计算。象变量一样，以后在介绍表达式和语句时，一般不特别说明引用的函数是属于哪一类型，因为由表达式是算术型还是布尔型的可以明确这一要求。

表1。1

标准函数	数学含义
#ABS(E)	E的绝对值
#ENTIER(E)	不大于E的最大整数
#SIGN(E)	E的符号，即#SIGN(E)=$\begin{cases}1 & \text{若 } E>0\\ 0 & \text{若 } E=0\\ -1 & \text{若 } E<0\end{cases}$
#SIN(E)	E的正弦值，$\lvert E\rvert\leq\pi\cdot 2^{34}/4$
#COS(E)	E的余弦值，$\lvert E\rvert\leq\pi\cdot 2^{34}/4$
#ARCSIN(E)	E的反正弦值，$\lvert E\rvert\leq 1$

#ARCTG(E)	E的反正切值
#LN(E)	E的自然对数，$E>0$
#EXP(E)	E的指数函数，$E<63*ln2$
#SQRT(E)	E的平方根，$E\geq 0$
#CUBRT(E)	E的立方根
#RANUM(E)	产生一个〔0，1〕中均匀分布的伪随机数，这里E只有形式上的作用，可任取一值
#TG(E)	E的正切值，$\|E\|\leq\pi\cdot 2^{27}/4$
#AXP(E, F)	E的F次方值，其中要求： $E>0$，$-64ln2<F\ln E<63ln2$
#RANUMP(E)	它将#RANUM(E)的伪随机数置成初态，然后转一下#RANUM(E)取得伪机数，其中E只有形式上的作用，可任取一值
#WG(E)	把弧度值E转换成如下形式的度、分、秒值（称为度、分、秒的十进制数表示形式）： $\pm A_m A_{m-1}\cdots A_1\cdot B_1B_2\cdots B_n\ (A_m\neq 0)$ 它表示 $\pm A_mA_{m-1}\cdots A_1^{\circ}\ \pm B_1B_2'\pm B_3B_4\ B_5\cdots B_n''$
#GW(X)	把十进制数表示形式的度、分、秒值 $X=\pm A_mA_{m-1}\cdots A_1\cdot B_1B_2\cdots B_n$ 转换成弧度值
#BO(E, F)	$\#BO(E,F)=\begin{cases}\text{负数，若E值的第}\langle|F|\rangle\text{位上为}1\\ \text{正数}(>0)\text{，若E值的第}\langle|F|\rangle\text{位为}0\end{cases}$

标准函数	数　学　含　义
#E[F](E, F)	E值左移（当F≥0）或右移（当F<0）[F]位的结果
#A(E, F)	#A(E, F)=F值θ_1(1, 1,…, 1, F_i, 1, …, 1)
#V(E, F)	#V(E, F)=E值θ_2(0, 0,…, 0, F_i, 0,…, 0)
#F(E, F)	#F(E, F)=E值θ_3(0, 0,…, 0, F_i, 0,…, 0)

註：1。E、F均代表算术表达式，|F| 表示表达式F之绝对值的整数部分。第|F|位代码 $Fi=\begin{cases}0, \text{当 } F\geq 0,\\ 1, \text{当 } F<0。\end{cases}$

2。计算三角函数时，E的值以弧度计。

3。后五个标准函数是把E的值视为一个48位二进制代码，对它进行按位运算，θ_1，θ_2，θ_3 分别表示两个二进制代码的逻辑积，逻辑和和按位加，其运算法则见表1。2。

表1。2

X	Y	Xθ_1 Y	Xθ_2 Y	Xθ_3 Y
1	1	1	1	0
1	0	0	1	1
0	1	0	1	1
0	0	0	0	0

§1.6 简单表达式

1。表达式描述某一运算规则，执行表达式的结果得到一个值，依赖于值的类型，表达式分为算术表达式和布尔表达式两种，前者取的值是数值，后者取的值是布尔值。每种表达式又有简单型和条件型之分。

还有一种表达式是命名表达式，它取的值既非数值也非布尔值，而是一个标号，它是与转语句有关的一种特殊的表达式，我们在§2.2里介绍。本段先介绍简单算术表达式和简单布尔表达式，下一段再介绍条件算术表达式和条件布尔表达式。

2。简单算术表达式

简单算术表达式描述某一算术运算。参加运算的可以是常数、数值型的变量和函数等，这些统称为初等量。可以施行的算术运算包括加法、减法、乘法、除法、整除和乘方，其运算符分别是

+、－、*、／、÷、↑

用若干个算术运算符将若干个初等量（运算对象）连起来就得到一个简单算术表达式。运算符一般出现在两个初等量之间，指明这两个量的运算方式，例如A＋B是表示A与B相加。但加号“＋”和减号“－”也允许出现在表达式的最左边，表示“正”和“负”的意思，这与代数公式的意义是一致的。例如－A＋B等同于B－A。运算符＋，－，*，／表示通常的四则运算。整除÷则表示两量相除后的商只取其整数部分，如5÷2＝2≠2。5，－7÷2＝－3。乘方运

算A ↑ B表示的是A^B，本编译系统限制B只能取整数值，否则为错。例如$A^{0.3}$不能表成A ↑ 0．3。为了计算它，可以调用标准函数#AXP，写成#AXP(A，0．3)，见§1．5。

一个表达式中可能出现若干个运算符，到底先做哪个运算呢？算法语言中规定先↑，后*或／或÷，最后再+或−。

按上述方式构成一个表达式后，如果用圆括号括起来，则又可以视为一个初等量，从而可作为一运算对象出现在表达式中。这样，一个比较复杂的表达式中除了有若干运算符外还可能有若干圆括号，执行这样的表达式时要先作里层括号内的运算，再依次向外进行。于是简单算术表达式的构成方式和运算次序与人们习惯的计算方式就几乎完全一致了。例如表达式

−A+5*(3．2−#SIN(B)／(C+1))−0．2*C↑3

所实现的算术运算是

$$-A+5\left(3.2-\frac{SINB}{C+1}\right)-0.2C^3。$$

书写表达式时要注意的是乘号*绝不能省，也不能用方括弧，无论有多少层括弧，一律用圆括号，否则为语法错。

最后指出，常数、变量、函数本身也是简单算术表达式的特殊情形，如5，−2，A，−#SIN(B)都是表达式。此外，本编译系统规定，一般的自行定义的函数只允许出现在包含它的“最大的”表达式的最左端，比方允许MAX(A)+B，而不允许B+MAX

（A），这里MAX（A）是函数命名符，但标准函数不在此列。

3。简单布尔表达式

布尔表达式是描述逻辑运算的。作为运算对象（布尔初等量）可以是布尔值、布尔变量和布尔函数。布尔变量和布尔函数只能取布尔值，因此布尔运算实质上就是对两个布尔值——TRUE（真）和FALSE（假）施行运算，其运算结果也只能取两个布尔值中的一个。运算种类包括逻辑乘、逻辑和和逻辑非三种，其运算符分别以

$$\wedge, \quad \vee, \quad \neg$$

表示，运算规则在表1。3给出。

表1。3

X	Y	X∧Y	X∨Y	¬X
TRUE	TRUE	T	T	F
T	FALSE	F	T	F
F	T	F	T	T
F	F	F	F	T

註：若将TRUE和FALSE分别以二进制码1和0代表，则对照表1。2，容易看出∧，∨与θ_1，θ_2是相当的。运算¬表示否定的意义，非真即假，非假即真。

能作为布尔运算运算对象的除布尔值、布尔变量和布尔函数以外，还有<关系式>。关系式是对两个简单算术表达式进行比较的式子，比较的结果产生一个布尔值，比较运算的种类有小于、小于或等于、等于、