第三次全国电子计算机专业学术会议论文选集

中国电子学会电子計算机专业委員会編

（内部資料　注意保存）

國防工業出版社

第三次全国电子計算机
专业学术会議論文选集

中国电子学会电子計算机专业委員会編

國防工業出版社

1964

内 容 簡 介

本論文选集共收入論文72篇，内容分七部分，第一部分为体系設計与邏輯設計，第二部分为整机与邏輯綫路，第三部分为内存儲器，第四部分为外部設备，第五部分为电源与通風散热問題，第六部分为元件与綫路的研究，第七部分为工艺、测試与测試設备。这些論文总結了我国近年来的研究成果。

本論文选集以合訂本与分册（每部分为一分册）两种形式出版。

本論文选集可供从事数字电子計算机专业的研究、生产的工程技术人員及高等院校师生参考。

第三次全国电子計算机专业学术会議論文选集

中国电子学会电子計算机专业委員会編

国防工业出版社 出版

北京市书刊出版业营业許可証出字第 074 号

国防工业出版社印刷厂印装　　内部发行

787×1092 1/16　印張 44 3/8　1065 千字

1964 年12月第一版　1964 年12月第一次印刷　印数：0,001—1,500 册

統一书号：N15034·852　　定价：（科八-2）8.90元

前　言

一九六三年十月在西安市举行了第三次全国計算技术經驗交流会議。本論文选集收集了这次会議中的有关数字电子計算机方面的学术論文及工作經驗报告，共計72篇。在出版之前，論文报告的作者对內容进行了校閱，并作了某些修改。

本論文选集的內容表达了一九六一年以来国內計算技术在理論与实际方面的工作成果，包括数字电子計算机的各个方面，共分七个部分：体系設計与邏輯設計，整机与邏輯綫路，內存儲器，外部設备，电源与通風散热問題，元件与綫路的研究，工艺、測試与測試設备。

会議中收到的数字电子計算机方面的学术报告資料共123篇，除上述七部分內容外，还有綜述性报告与組織工作經驗介紹等，由于篇幅所限及其他种种原因，均未列入本論文选集。

本論文选集可供从事电子計算机研究試制人員、高等院校計算技术专业教学人員以及厂矿企业中应用电子計算机的研究技术人員参考。

在出版本論文选集时，得到了中国科学院計算技术研究所、国防工业出版社等单位及論文报告的作者的大力协助和支持，特此表示感謝。

編　者

目　录

第一部分

体系設計与邏輯設計

目　录

第一部分　体系設計与邏輯設計

一种内外一致的机器語言

（通用机新体系邏輯結构研究之一）

高庆獅等

摘　要

本文分两部分，第一部分是对語言本身的描述，描述是借用递归定义的方式进行，并加上适当的說明。第二部分是对本語言的实現的主要部分进行了簡略的討論。它并不企图代替具体机器的設計，而仅仅是对由本語言引起变化的有关部件，进行了原理性的討論，由于受时間和篇幅的限制，討論是很粗略的。

第一部分可供使用者参考，第二部分可供机器設計时参考。

一、引　言

通常，机器內部語言和外部語言是不一致的。內部語言，較多地照顧了工程因素，例如，解題能力强，設备省等等。外部語言，是照顧了使用的方便，例如，Algol，是为科学計算的使用者提供方便，而 Cobol，是为商业用戶提供方便。

人們自然会想到，通常的机器語言（例如 104 机的），作不大的修改，能否进一步为使用者提供些方便呢？反过来，通常的外部語言，例如 Algol，作了适当的修改，能否在增加設备不多的情况下，由机器直接实現呢？本文主要討論后者，而前者在“一种摆脫地址依賴性的机器語言”中討論。这些討論是初步的，而且是专題研究性质的，不是直接用到机器上去。可以想象，通常机器語言作为一点，Algol 作为另一点，之間联接一直綫，前者是靠近通常机器語言一端的一点，而后者是靠近 Algol 一端的另一点。对于一个具体机器而言，可以选擇的点（即方案）很多，必須根据具体条件来确定。例如，当元件很可靠，价格很低，有按內容取数的存儲器等等工程技术条件时，作为专为科学計算用的机器，选 Algol 作为机器語言也是允許的。

計算机的設計本身，实质上是在一定的使用要求及可能的技术条件下，进行方案选擇。显然，方案的好坏，不仅取决于选擇本身的正确性，而且取决于被选擇的方案集合的大小。所以，在本文中，例如表达式，我們討論了三种类型，而不只討論某一种。对一个具体机器而言，可任选一种，或任选二种，甚至全部三种都采用。到底选这十七种可能方案的那一种呢？有了这三种类型的綜合方案，选擇就較为方便了。

二、語言的描述

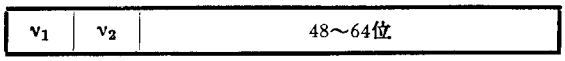
1. 机器的約定

內存儲器 2×4096 字；指令存儲器 4096 字，速度与內存儲器相同；变址存儲器64字，超高速存儲器 64 字，两者速度均为內存儲器的 8 倍；一級存儲器为 2^{17} 字，字长均为 50

位（～66 位）。

机器特点：一級存儲器的容量有穷，存儲方式是按地址取数，非按內容取数，采用 Atlas 的自动調度一級存儲器的工作方式，各大部件异步并行，有先行部件。

字的形式：（v_1 v_2 各一位）



在語句中，操作符和标志符組成的字，$v_1=0$；一、二、三型操作符和标志符組成的字，$v_2=0$；而被操作代碼（数或符号）組成的字，$v_1=1$。在数据中，所有的字 $v_1=1$。

2. 符号的約定

∷= 定义符。

{ 或者。

} 形式符号。

{"} 括号代之該行第一个 {"} 的括号內的內容。

<> 括号內非基本符号（即：非本語言系統的語言），而是元語言。

° 其下的符号可以不出現，或重复出現任意有限次。

! 其下的符号可以不出現，或出現一次。

°- -°及!- -!的- -为指出 ° 及 ! 的作用域，例如，

$$\overset{!\text{---}!}{a\ b\ c\ d\ e}\ ::=\begin{cases}de\\abcde\end{cases}$$

n 其下的符号重复出現 *n* 次，*n* 为任一正整数，例如 $\overset{5}{A}::=AAAAA$。

符号的解釋，由上而下，由外而內，最后为{"}。

3. 語言結构

（1）基本符号

(1.1) 十六进制代碼：以

n 代表0，1，2，3，4，5，6，7，8，9，*a*，*b*，*c*，*d*，*e*，*f* 之一。

(1.2) 标志符：以

A 代表 *A*，*B*，*C*，*D* 之一，但 A_0 中的 *A* 只代表 *A* 本身。

i 代表 *i*，*j* 之一。

x 代表 *x*，*y*，*z*，*w* 之一。

λ 代表 λ，μ 之一。

N 代表 *nn*。

Ж，ℱ，φ，Ψ 只代表自己。

(1.3) 形式符：

，是形式符，可以任意增删。不編碼，不穿孔。

Δ，否，为书写清晰所引进的符号。不編碼，不穿孔。

(1.4) 識别符：

注，；，“，”均为不編碼，不穿孔的符号，但不能任意增删。

(1.5) 操作符：包括运算符，控制符和輸入符。

运算符:

ᔕ, ⤙, ↟, ⇄, (, ‖, Ж, $\mathscr{F}$, ×, ÷, ×′, ÷′, +, −, +′, −′, ⏀, Sign, ⩽, ∠, =, ¬, ⧻, ∧, ∨, ⧺,), |, ➛, ↦。

控制符:

~, 如, $\overline{\text{如}}$, 返, ⤑, ㄐ, ⌙, ↪, ;, 停, 完,], [, 始, 終, 首, 末, 子, 了, ⊛, ⊕, ⋹, 型, ▽, 点, 灭, 訪, 开, 程, 調, 換, *, 讀, 写, 引, 傳, 穿, 印, 显, 入, 轉, 轉′, 則, $, 定。

輸入符:

§§, §, 空, 数。

(2) 名, 名′, 値, 变量, 行, 数据和注解

(2.1) 名

<名>∷= {
<控制字名>∷=A_{nN}
<語句名> ∷=<标号>∷=A_{nN}
<过程名> ∷= {Ж$_N$ / $\mathscr{F}_N$ / <标号>}
<开关名> ∷=A_{nN}
<数据名> ∷=A_n
}

N的規定:

如某名 A_{nN} 的 A_n 仅用一个，这时N可任意取一个。一般取: $N=00$，机器执行速度較快。

如某名 A_{nN} 的 A_n 用了二个或二个以上，这时$N\neq00$，一般取$N=01$, 02……。

作为过程名的Ж$_N$, $\mathscr{F}_N$ 中的$N\neq00$, Ж$_N$ 对应于过程語句，$\mathscr{F}_N$ 对应于函数指示符。

(2.2) 名′

<名′>∷= {
<控制字名′>∷= {A_n / $A_{n\Psi ON}$}
<語句名′> ∷=<标号′>∷= {A_n / $A_{n\Psi ON}$}
<过程名′> ∷= {Ж$_N$ / $\mathscr{F}_N$ / <标号′>}
<开关名′> ∷= {A_n / $A_{n\Psi ON}$}
<数据名′> ∷=A_n
}

名′是名在函数指示符、过程指示符一型括弧內参数（或在二型对应的部分）中及子程序的子后第二个; 之前出現的名的另一种形式。与名有确定的对应关系:

A_n 对应于 A_{n00} (数据名′对应于本身)。

$A_{n\Psi ON}$ 对应于 A_{nN}。

以上左右边的A, n, N相同。

(2.3) 值

$$<\text{值}>::=\begin{cases} <\text{数}> ::= \begin{cases} \eta_N \\ !\,\$ \begin{cases} +_q \\ -_q \end{cases} \end{cases}^{10}_{n}\ {}^{2}_{n} \\ <\text{邏輯值}>::= \begin{cases} 00 \\ 01 \end{cases} \\ <\text{符号值}>::= !\,\$ \begin{cases} 2 \\ 3 \end{cases}^{6} N \\ <\delta\text{一值}>::= \phi n^{5} \end{cases}$$

当数或符号值是出現在表达式中时，必須写上$；当数或符号值是在数据中出現时，不必写$。浮点数，前十个 n 是尾数，以 M 表示，后二个 n 是价碼以 n 表示。則数的大小为：

$$\pm \cdot M \times 2^{n-128}。$$

η_N 数的大小为 $\cdot N \times 2^8$。定点数，小数点在第一个 n 之前，尾数有十二个 n，以 M 表示，則数的大小为 $\pm \cdot M$。

每个单元，可存 6 个符号，每个符号 8 个二进位。

每个邏輯值 8 个二进位：00000000 或 00000001。

δ 一值即通常地址，有二十位，后十七位指出单元的地址，前三位指出单元中符号的位置。

(2.4) 变量

$$<\text{变量}>::= \begin{cases} A_n \begin{cases} !\,3\ \psi n \\ {}^{5}\ \phi n \end{cases} \circ\!\!-\!\!\circ\, i_n \\ !\ \nabla i_n \\ x_N \\ \lambda_N \\ A_{eN} \\ B_{eN} \end{cases}$$

$$<\delta\text{—变量}>::= \begin{cases} !\ \Delta i_n \\ A_n \end{cases}$$

当 ∇i_n 之前是操作符时，∇可以省去。

作为 δ —变量的 ∇i_n 或 A_n 均为 24 位，相当于作为变量的 ∇i_n 的后 24 位及作为变量的 A_n 本身。

作为变量的 $A_n \begin{cases} {}^{5}\ \psi n \\ {}^{3}\ \phi n \end{cases} \circ\!\!-\!\!\circ\, i_n$　　$A_{n i n i n} \circ\!\!-\!\!\circ$，$x_N$，$(N \fallingdotseq 00)\lambda_n$，$A_{eN}$，$B_{eN}$ 均为 50 位，当与 A_n 及<δ—变量>操作时，只进行后 24 位的操作。

x_N($N\fallingdotseq 0.0$)用法参見(5.7)节

A_n, ∇i_n, λ_n 是指向超高速存儲单元，这单元还可以被 A_{eN} 和 B_{eN} 所指向。其实际地址对应关系如下表：

超高速存儲分配

以下同行各列所指的单元相同。

B_{e00}			B_{e51}		i_1		B_{e00}	A_{e08}	A_{ec}	A_{e03}
B_{e01}			⋮		⋮		B_{e01}	A_{e07}	A_{eb}	A_{e02}
⋮			B_{e5f}		i_f		B_{e02}	A_{e06}	A_{e0a}	$A_{e\dot{0}1}$
$B_{e\mathrm{M}f}$			B_{e60}	A_0	B_0		B_{e03}	A_{e05}	A_{e09}	
B_{e20}	λ_0		B_{e61}	A_1	B_1		B_{e04}	A_{e04}	A_{e08}	
B_{e21}	λ_1		⋮	⋮	⋮		B_{e05}	A_{e03}	A_{e07}	
⋮	⋮		B_{e6e}	A_e	B_e		B_{e06}	A_{e02}	A_{e06}	
B_{e2f}	λ_f		B_{e6f}	x_{00}	y_{00}		B_{e07}	$A_{e\dot{0}1}$	A_{e05}	
B_{e30}	μ_0		B_{e70}	C_0	D_0		B_{e08}		A_{e04}	
B_{e31}	μ_1		B_{e71}	C_1	D_1		B_{e09}		A_{e03}	
⋮	⋮		⋮	⋮	⋮		B_{e0a}		A_{e02}	
B_{e3f}	μ_f		B_{e7d}	C_d	D_d		B_{e0b}		$A_{e\dot{0}1}$	
B_{e40}		i_0	B_{e7e}	Z_{00}	$\mathcal{F}_{00}$		B_{e0c}			
B_{e41}		i_1	B_{e7f}	W_{00}	Ж$_{00}$		B_{e0d}			
⋮		⋮					B_{e0e}			
B_{e4f}		i_f					⋮			
B_{e50}		j_0								

称运算器被乘数寄存器为后进先出器的頂(当它有內容时)。

打＊单元为后进先出器的端。右图中，右边三列分别为端＝B_{e07}, B_{e0b}, B_{e02}的三个例子。

(2.5) 行

<行>::=$\mathcal{S}$“<非“或”的任意加以約定及編碼的符号表中的符号的任意序列>”。

行是出現在一型过程語句括号內的参数(或二型，三型中对应的部分)。

(2.6) 数据

$$\text{<数据>::=数}\,\S\,A_{nN}\text{<}\left\{\begin{matrix}\left\{\begin{matrix}+_q\\-_q\end{matrix}\right\}^{10\ 2}_{n\ n}\\ {}^{6}_{2N}\end{matrix}\right\}\text{的任意序列>}$$

(2.7) 注解

<注解>::=注<非；的任意符号的任意序列>；

注解不輸入机器，也不穿孔，可以杆在程序的任何地方。

(3) 控制字和說明

$$\left\{\begin{matrix}\text{<循环控制字>::=}\left\{\begin{matrix}{}^{1\ \ \dot{5}}_{\ \ 0n}\\ A_{nN00}\\ 0'A_{nN}\end{matrix}\right\},\ \left\{\begin{matrix}{}^{5}_{1n}\\ 1'A_{nN}\end{matrix}\right\},\ \left\{\begin{matrix}\dot{1}\\ \ \end{matrix}\right\}\left\{\begin{matrix}{}^{5}_{2n}\\ 2'A_{nN}\end{matrix}\right\},\ \left\{\begin{matrix}\dot{1}\\ \ \end{matrix}\right\}\left\{\begin{matrix}{}^{5}_{1n}\\ 1'A_{nN}\end{matrix}\right\},\ \left\{\begin{matrix}\dot{1}\\ \ \end{matrix}\right\}\left\{\begin{matrix}{}^{5}_{3n}\\ 3'A_{nN}\end{matrix}\right\}\\ \text{<开关控制字>::=}0n, A_{nN}, 3n, A_{nN}\end{matrix}\right.$$

$$
\text{<控制字>::=}\begin{cases}\text{<外部控制字>::=}\begin{cases}\text{<讀写控制字>::=}0n,N,\begin{Bmatrix}{}^{*}A_{nN}00\\ {}_{5}\\ A_n\phi n\end{Bmatrix},3nNN,\\ \text{<引带控制字>::=}3n,N,N,\\ \text{<輸入輸出控制字>::=}0n,N,\begin{Bmatrix}{}^{*}A_{nN}00\\ {}_{5}\\ A_n\phi n\end{Bmatrix},3n\ NN,\\ \text{<傳送控制字>::=}0000\begin{Bmatrix}1{}^{*}A_{nN}00\\ {}_{5}\\ A_n\phi n\end{Bmatrix},0000\begin{Bmatrix}1\end{Bmatrix},3nNN,\end{cases}\end{cases}
$$

$$
\text{<說明>::=}\begin{cases}\text{<循环說明>::=§§<控制字名><循环控制字>,}\\ \text{<开关說明>::=§§<开关名><开关控制字>,}\\ \text{<輸入輸出說明>::=§§<控制字名><外部控制字>,}\\ \text{<过程說明>::=§<过程名><語句>返,}\end{cases}
$$

循环控制字是控制循环語句执行的参数，它由一个或几个字段組成，一个字段由一个或几个字节組成，一个字节由一个或几个参量組成，一个参量 24 位，前 4 位称为特征位，后 20 位是参数，参数結构形式与地址 20 位相对应。一个字节的最后一个参量的特征位为 1 （或 1′），其他参量的特征为 0 （或 0′），但一个字段的最后一个参量的特征位是 2（或 2′），而不是 1 （或 1′），一个控制字的最后一个参量的特征位是 3 （或 3′），而不是 2（或 2′）。

开关控制字是控制部分的轉向語句和条件語句的轉向。它由一个或几个参量組成，最后一个参量的特征位为 3，其他为 0，每个参量包括一个語句名或过程名。

讀写控制字，依次指出了磁带的带号（n），区号（N），对应一級存儲器开始地址，傳送性质（n）傳送个数（NN）。其中内存儲器开始地址可以表現为名 A_{nN}，这是指該名对应的数据或程序等等的第一单元的实际地址。也可表現为变量 $A_{n\phi n}^{5}$，即該变量的实际地址，也就是名 A_n 或 A_{n00} 相对地址为 $\frac{5}{n}$ 的該单元实际地址。

引带控制字，依次指出該带的带号（n），区号（N），及性质（N）：向前引到 N 区，向前引 N 区，向后引到 N 区或向后引 N 区等等。

輸入輸出控制字，依次指出部件号（n），备用（N），一級存儲器地址开始地址，傳送性质（N）傳送个数（NN）。其中一級存儲器开始地址的表現形式与讀写控制字全同。

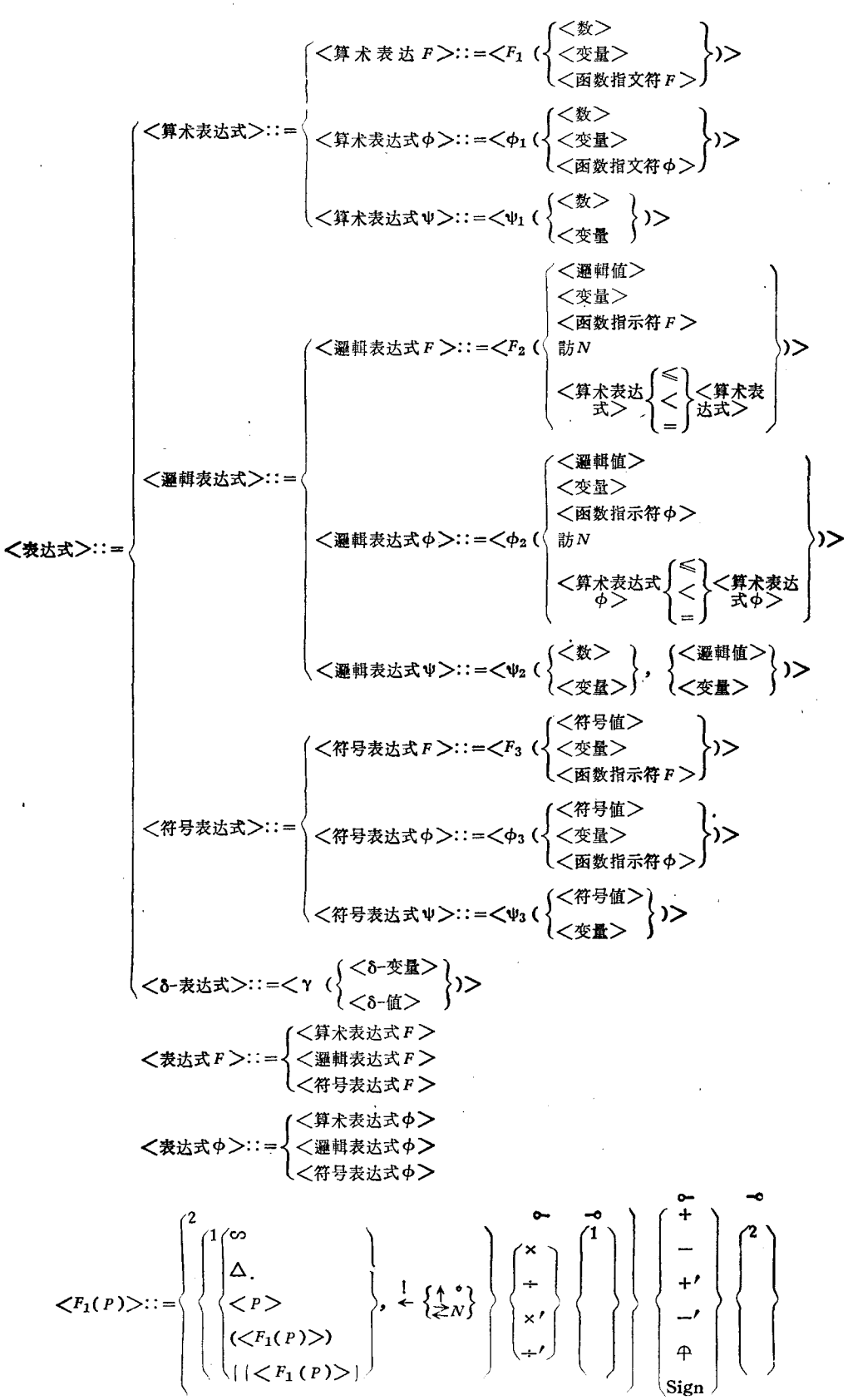
傳送控制字控制一級存儲器之間的傳送。由二个一級存儲器地址（前者为“从”后者为“到”）和傳送个数組成。

以上各外部控制字的傳送个数前的特征位均为 3。

过程說明即通常程序自带的閉子程序，入口在最前，出口在最后，如其名为 $\mathcal{K}_N$（$N\neq 00$）則通过过程語句来使用，如其名为 F_N（$N\neq 00$）則通过函数指示符来使用。如其名为 A_{nN}，則通过轉向語句来使用。

注：讀写带的傳送性质指傳送时进行处理的性质，例如十进制变为二进制，二进制变为十进制等等。

（4）表达式和函数指示符


<表达式>::=
<算术表达式>::=
<算术表达式F>::=<F1(<数> <变量> <函数指文符F>)>
<算术表达式φ>::=<φ1(<数> <变量> <函数指文符φ>)>
<算术表达式ψ>::=<ψ1(<数> <变量>)>
<邏輯表达式>::=
<邏輯表达式F>::=<F2(<邏輯值> <变量> <函数指示符F> 訪N <算术表达式> ≤ < = <算术表达式>)>
<邏輯表达式φ>::=<φ2(<邏輯值> <变量> <函数指示符φ> 訪N <算术表达式φ> ≤ < = <算术表达式φ>)>
<邏輯表达式ψ>::=<ψ2(<数> <变量>, <邏輯值> <变量>)>
<符号表达式>::=
<符号表达式F>::=<F3(<符号值> <变量> <函数指示符F>)>
<符号表达式φ>::=<φ3(<符号值> <变量> <函数指示符φ>)>
<符号表达式ψ>::=<ψ3(<符号值> <变量>)>
<δ-表达式>::=<γ(<δ-变量> <δ-值>)>
<表达式F>::=<算术表达式F> <邏輯表达式F> <符号表达式F>
<表达式φ>::=<算术表达式φ> <邏輯表达式φ> <符号表达式φ>

<F1(P)>::=
2
1
∞
Δ.
<P>
(<F1(P)>)
||<F1(P)>|
!
↑ °
≥N
× ÷ ×' ÷'
1
+ − +' −' ↑
Sign
2

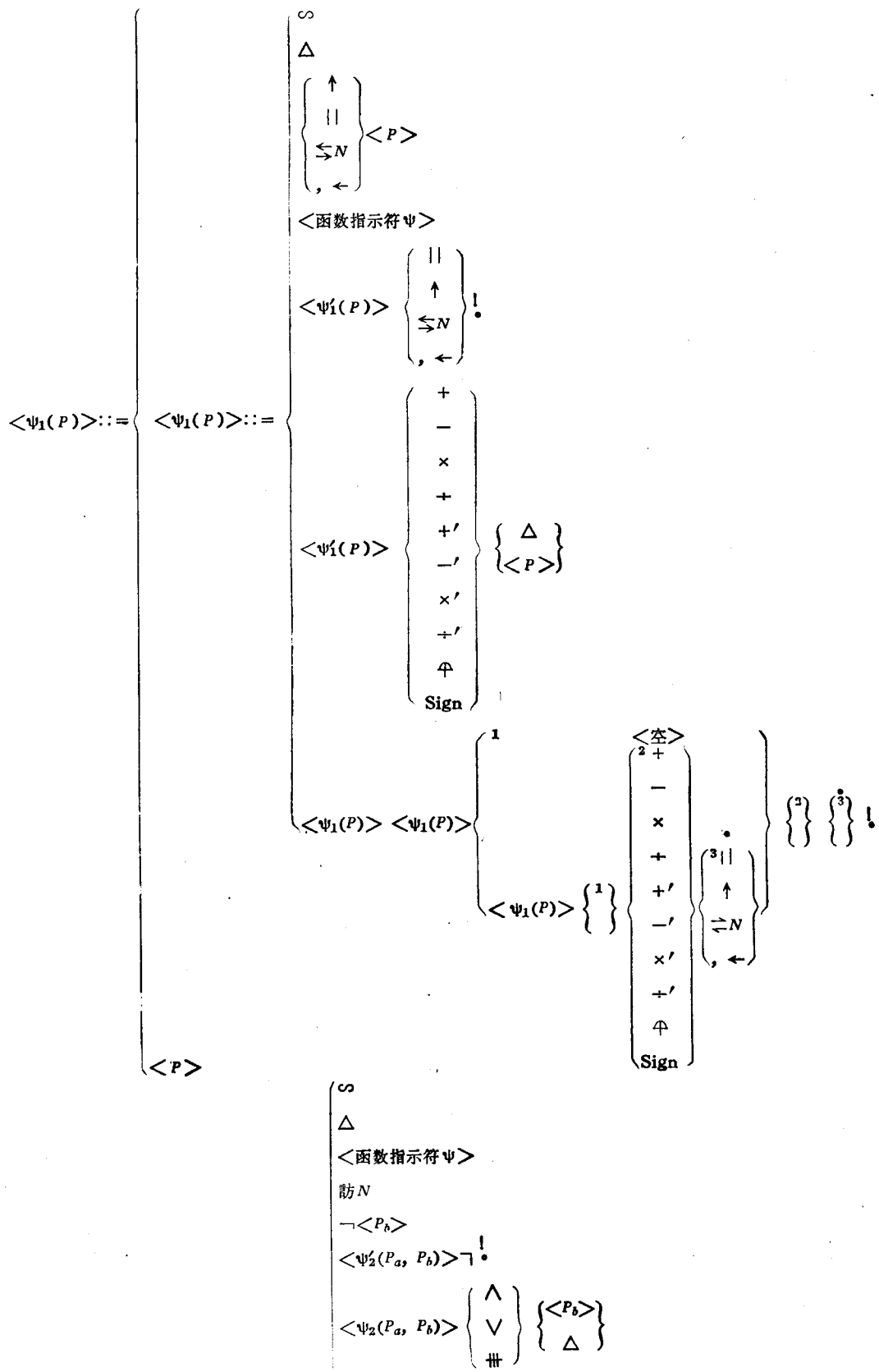
$$\langle F_2(P)\rangle ::= \left\{ \left\{ \left\{ \begin{array}{l} \text{!} \\ \neg \end{array} \right\}^{1} \left\{ \begin{array}{l} \infty \\ \Delta \\ \langle P\rangle \\ (\langle F_1(P)\rangle) \end{array} \right\}^{2} \right\} \left\{ \begin{array}{l} \wedge \\ \# \end{array} \right\}^{\circ\!\vdash} \left\{ \begin{array}{l} 1 \\ \end{array} \right\}^{\vdash\!\circ} \right\} \left\{ \begin{array}{l} \vee \\ \end{array} \right\}^{\circ\!\vdash} \left\{ \begin{array}{l} 2 \\ \end{array} \right\}^{\vdash\!\circ}$$

$$\langle F_3(P)\rangle ::= \left\{ \left\{ \left\{ \begin{array}{l} \text{!} \\ \neg \end{array} \right\}^{1} \left\{ \begin{array}{l} \infty \\ \Delta \\ \langle P\rangle \\ (\langle F_3(P)\rangle) \end{array} \right\}^{2} \right\} \left\{ \begin{array}{l} \wedge \\ \# \end{array} \right\}^{\circ\!\vdash} \left\{ \begin{array}{l} 1 \\ \end{array} \right\}^{\vdash\!\circ} \right\} \left\{ \begin{array}{l} \vee \\ \# \end{array} \right\}^{\circ\!\vdash} \left\{ \begin{array}{l} 2 \\ \end{array} \right\}^{\vdash\!\circ}$$

$$\langle \phi_1(P)\rangle ::= \left\{ \begin{array}{l} \infty \\ \Delta \\ \langle P\rangle \\ \langle \phi_1(P)\rangle \left\{ \begin{array}{l} \uparrow \\ \leftrightarrows N \\ , \leftarrow \\ \shortparallel \end{array} \right\} \\ \\ \\ \\ \\ \\ \langle \phi_1(P)\rangle\langle \phi_1(P)\rangle \left\{ \begin{array}{l} + \\ - \\ \times \\ \div \\ +' \\ -' \\ \times' \\ \div' \\ \dot{+} \\ \text{Sign} \end{array} \right\} \end{array} \right.$$

$$\langle \phi_2(P)\rangle ::= \left\{ \begin{array}{l} \infty \\ \Delta \\ \langle P\rangle \\ \langle \phi_2(P)\rangle\langle \phi_2(P)\rangle \left\{ \begin{array}{l} \wedge \\ \vee \\ \# \end{array} \right\} \\ \langle \phi_2(P)\rangle\neg \end{array} \right.$$

$$\langle \phi_3(P)\rangle ::= \left\{ \begin{array}{l} \infty \\ \Delta \\ \langle P\rangle \\ \\ \langle \phi_3(P)\ \langle \phi_3(P)\rangle \left\{ \begin{array}{l} \wedge \\ \vee \\ \# \\ \# \end{array} \right\} \\ \\ \langle \phi_3(P)\rangle\neg \end{array} \right.$$


<ψ1(P)>::=
<P>
<ψ1(P)>::=
∞
△
↑ || ≤N , ← <P>
<函数指示符ψ>
<ψ'1(P)> || ↑ ≤N , ← !
<ψ'1(P)> + − × ÷ +' −' ×' ÷' 个 Sign △ <P>
<ψ1(P)> <ψ1(P)> 1 <ψ1(P)> 1 <空> 2 + − × ÷ +' −' ×' ÷' 个 Sign 3 || ↑ ⇄N , ← 2 3 !
∞
△
<函数指示符ψ>
訪N
¬<Pb>
<ψ'2(Pa, Pb)>¬!
<ψ2(Pa, Pb)> ∧ ∨ 卌 <Pb> △

$$\langle\psi_2(P_a,\ P_b)\rangle::=\begin{cases}\langle\psi_2'(P_a,\ P_b)\rangle::=\begin{cases}\langle\psi_1(P_a)\rangle\begin{Bmatrix}\leqslant\\<\\=\end{Bmatrix}\begin{Bmatrix}\langle P_a\rangle\\\triangle\end{Bmatrix}\\\langle\psi_1(P_a)\rangle\langle\psi_1(P_a)\rangle\begin{Bmatrix}\leqslant\\<\\=\end{Bmatrix}!\\\langle\psi_2(P_a,\ P_b)\rangle\begin{Bmatrix}1\\\langle\psi_2(P_a,\ P_b)\rangle\begin{Bmatrix}1\end{Bmatrix}\end{Bmatrix}\end{cases}$$

$$\langle\psi_3(P)\rangle::=\begin{cases}\langle\psi_3'(P)\rangle::=\begin{cases}\infty\\\triangle\\\langle\text{函数指示符}\psi\rangle\\\neg\langle P\rangle\\\langle\psi_3'(P)\rangle\neg!\\\langle\psi_3(P)\rangle\begin{Bmatrix}\wedge\\\vee\\\dagger\\\ddagger\end{Bmatrix}\begin{Bmatrix}\langle P\rangle\\\triangle\end{Bmatrix}\\\langle\psi_3(P)\rangle\langle\psi_3(P)\rangle\begin{Bmatrix}1\\\langle\psi_3(P)\rangle\begin{Bmatrix}1\end{Bmatrix}\end{Bmatrix}\end{cases}\\\langle P\rangle\end{cases}$$

$$\langle\gamma(P)\rangle::=\begin{cases}\langle P\rangle\\\langle\gamma(P)\rangle\begin{Bmatrix}\sqcup\\\sqcup\end{Bmatrix}\langle P\rangle\end{cases}$$

$$\langle\text{函数指示符}\rangle::=\begin{cases}\langle\text{函数指示符}F\rangle::=\mathcal{F}_N\left(\begin{Bmatrix}1\langle\text{名}'\rangle\\\langle\text{表达式}\phi\rangle\end{Bmatrix}\begin{Bmatrix},\\\circ\end{Bmatrix}\begin{Bmatrix}\\1\end{Bmatrix}\right)\\\langle\text{函数指示符}\phi\rangle::=\begin{Bmatrix}\langle\text{名}'\rangle\\\langle\text{表达式}\phi\rangle\end{Bmatrix},\mathcal{F}_N\\\langle\text{函数指示符}\Psi\rangle::=\mathcal{F}_N\begin{Bmatrix}\langle\text{名}'\rangle\\\langle\text{变量}\rangle\\\langle\text{值}\rangle\end{Bmatrix}\end{cases}$$

(4.1) 三种类型

本系統語言有三种类型，除表达式，函数指示符和过程語句外，三种是一样的。对于表达式，函数指示符和过程語句，三种类型是：带括号表示（以F表示）；波兰表示（以ϕ表示）；順序表示（以Ψ表示）。第三种类型与通常的指令形式沒有本质区别，同样是順序