DJS——8计算机

FORTRAN IV 编译系统设计说明

第一介册

编译总控

语检与中间文本生成

中国科学院高能物理研究所七室
FORTRAN IV编译系统设计组

1981年12月

目　录

编译总控

§1 结构说明

FORTRAN SYSTEM DIRECTOR 简称FSD，

其控制来自PPP——操作系统的专编语句QB，

它以FORTRAN Ⅳ语言书写的源程序作为输入，而以各源程序段生成的半目标模块作为输出，并以文件形式提供给联结编辑程序（见第四分册），从而结束自己的工作。多次扫描源程序，以源程序段为单位分三阶段进行的方式，其中经过语法检查、生成中间文本的编译第一阶段BY1，优化的编译第二阶段BY2，（见第二分册），生成半目标模块的编译第三阶段BY3（见第三分册）。各编译阶段的通讯联系是通过一张通讯表COM为依据。彼此接口简单，结构清楚，便于维护与改进。

各阶段的编译程序与工作单元采取复盖方式进入内存，从而节省了内存空间。输入直接取自源程序鼓文件YWG，编译阶段本处理源程序一级的修改，直接由操作语句实现，如果有修改，输入则直接取自腾正的源程序鼓文件GWG，输出是以鼓文件BWG提供。

编译阶段和自定义语句的联系，则直接取自约定的有关工作单元，详见框图，关于自定义作业语句的设计见第五分册。

§2 通讯表

COM00	0主段，1子程序段，2函数段，3快数据段	
01	0算态，1调态	
02	0不优化，1循环优化，2全局优化	
03	每次读YWG时的起始行号	
04	0不使用附加函数，1使用	
05	BY3用后的数存下界	
06	BY1用数存上界	
07	YWG文件的总页数	
10	从YWG读入缓冲区页数计数回	
11	读入YWG最后一页缓冲区末址	
12	读单词工作单元	
13		
14		
15		
16	当前编译的段名	
17		
20	W区头（优化常数区）	
21	H区头（临时工作单元）	
22	语法错工作单元	
23		
24	BY2用数存上界	
25	BY3用数存上界	
26	BY2用变址个数	
27		

接上页

COM30	DBD	调试字典头
31	PD	文本项头
32	SD	符号字典头
33	CD	常数字典头
34	FAD	数组字典头
35	CMD	公用字典头
36	PUD	共名字典头
37	EQD	等价字典头
40	LD	标号字典头
41	FLD	格式标号字典头
42	FD	格式字典头
43	DD	DATA字典头
44	SFD	语句函数字典头
45	NAD	地址参数字典头
46	DS	输入输出广指参数区长
47	DATA	区末址
50	V区末址（局部号区）	
51	Z区末址（假变址区）	
52	源程序输入结束特征	
53	数存可用末址	
54	（处理PAGE行工作单元）	
55		
56	记BWG文件起始行号工作单元	
57		

§3 框 图

说 明：自定义语句提供给编译的工作单元

CYWGM：源程序鼓文件YWG 末行号

CTT ：调态标志

CYH ：优化级别标志

CFH ：使用附加内部函数标志

CBYBQ：BY1工作区上界

CTSD ：调试字典头

CGYD：有无修改标志

BGWGS：有修改时表示腾正鼓文件GWG可用行号（末行号加1）

CDBWG：半目标鼓文件BWG定义否标志

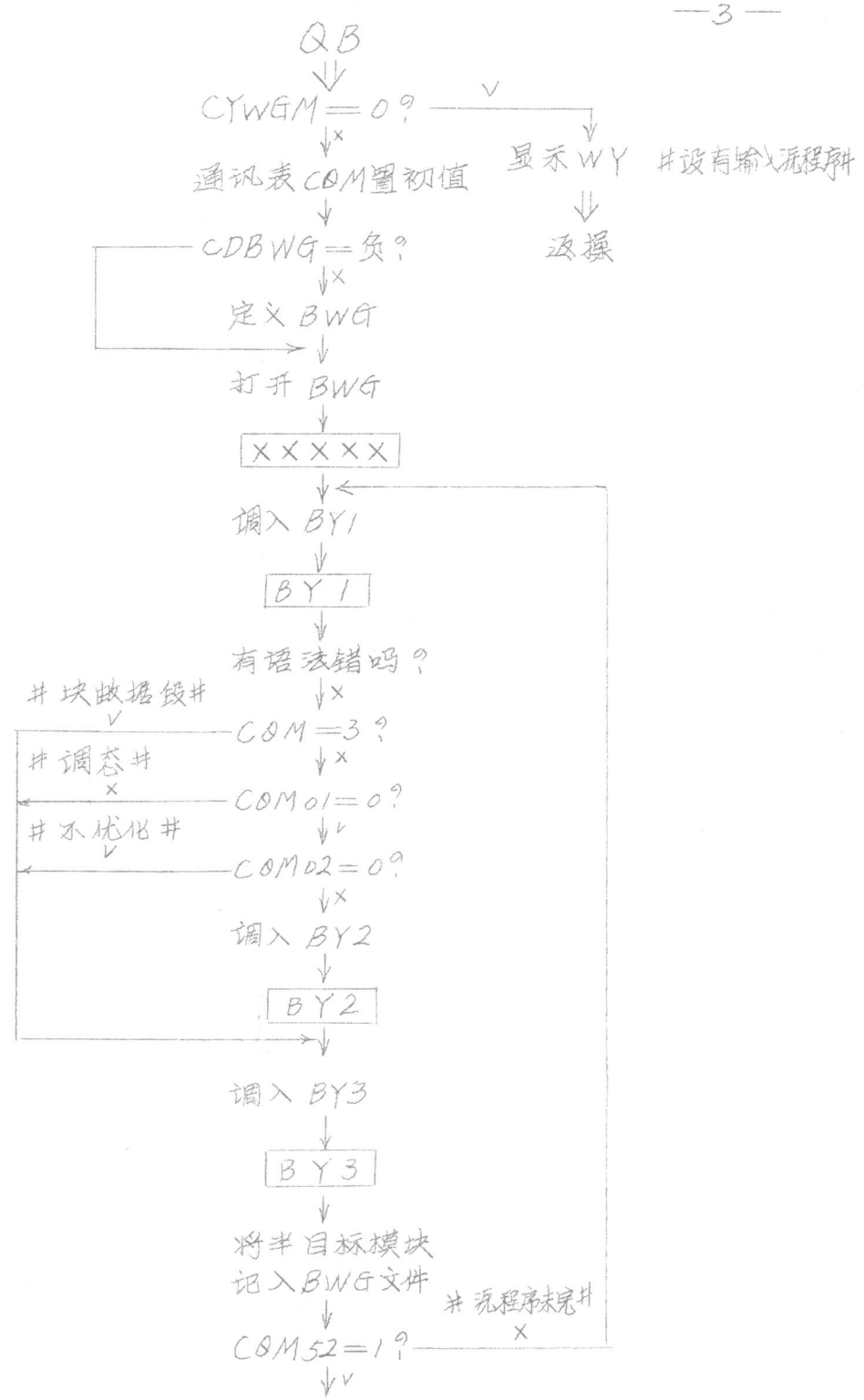
CBWGM：BWG文件末行数

XXXXX：是一个子程序，表示从YWG或GWG读入一页至缓冲区。

BY1 ：编译第一阶段总控

BY2 ：编译第二阶段总控

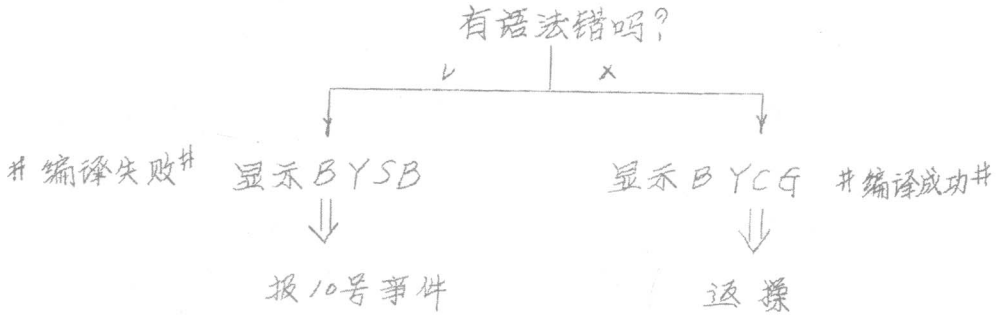
BY3 ：编译第三阶段总控



接下页

A

接上页



第 二 部 分

编译第一阶段结构说明

§1 编译第一阶段的主要工作要点

1、读单词并进行词法分析

2、语法检查

其中包括读单词中的词法（CT）检查，足状态表中的句法检查（JT），以及造字典仪息表和生成中间文本过程中的语法（YT）检查。如有错，则将段名，错误编号、行、序、宽行输出。

3、造字典仪息表

它们是符号字典（SD），数组字字典（FAD），公用字典（CMD），共名字典（PUD），等价字典（EQD），标号字典（LD），格式标号字典（FLD），常数字典(CD)，地址参数字典（NAD），数据初值字典（DD），格式字字典（FD）。

4、对段头句、语句函数定义句和所有可执行语句，可执行句前的定义性标号（包括编译生成的内部语句标号）都生成中间文本项，中间文本项有两类不可优化形式和优化形式。为了优化分块和调试定位需要于语句函数定义句前和可执行语句前还要生成行文本项。

5、字典和文本项的生成是在状态表的控制下借助于状态栈S和工作栈A由各编译子程序完成的，字典和文本项各自都是若干项组成的，各项之间彼此用链串起来，链头置于通讯表。

字典和文本项统称中间语言，经过编译第二阶段优化，编译第三阶段将它改造成可供装配代真的半目标模块形式。由编译总控 FSD 以文件形式交给联结编辑程序。

6、编译第一阶段于段结束行 END，进行段结束的处理工作，其中包括等价字典 EQD 的拼比工作，根据公用，共名字典进行分配相对地址，对共数组在分配头地址的同时计算好假头 $\overline{w_0}$ 填入数组字字典（FAD）的相应栏内。

§2 字典项结构

1. 符号字典（SD）

A 6	CHAIN 18
B 6	P1 18
NAME	

每项两个全字，链式杂凑码查填，表头置于通讯表。关于链式杂凑法附后。

(1) A 字栏的 A_0、A_1、A_2 第一阶段作为工作单元用，意义如下

$A_0 = 1$ 参与等价

$A_1 = 1$ 相对地址 P_1 已分配

$A_2 = 1$ 该之素在无名块中

A_3、A_4 为种属 CAT2，下述。

A (2) B 字栏分类型种属两栏

TY	CATi

TY 0 1 2 3 4 5 6 7

类型 空 实(双)实 逻

其中 TY = 0，6，7 不用，双精度处理为实型，（上述为8进制表示）

CAT1，CAT2 合起来为符号名种属意义见下表

CAT1 / CAT2	0	1	2	3	4	5	6	7
0		V	A	PR	SF			
1		VF	AF	PRF				
2		VK	AK					
3		VB	AB					

V，A，PR，SF 分别表示变号、数组、过程、语函。

VF，AF，PRF 分别表示哑变号，哑数组，哑过程。

VK，AK 分别表示公用变号，公用数组。

VB，AB 分别表示共名变号，共名数组。

其余不用

(3) P1 栏、相对地址栏

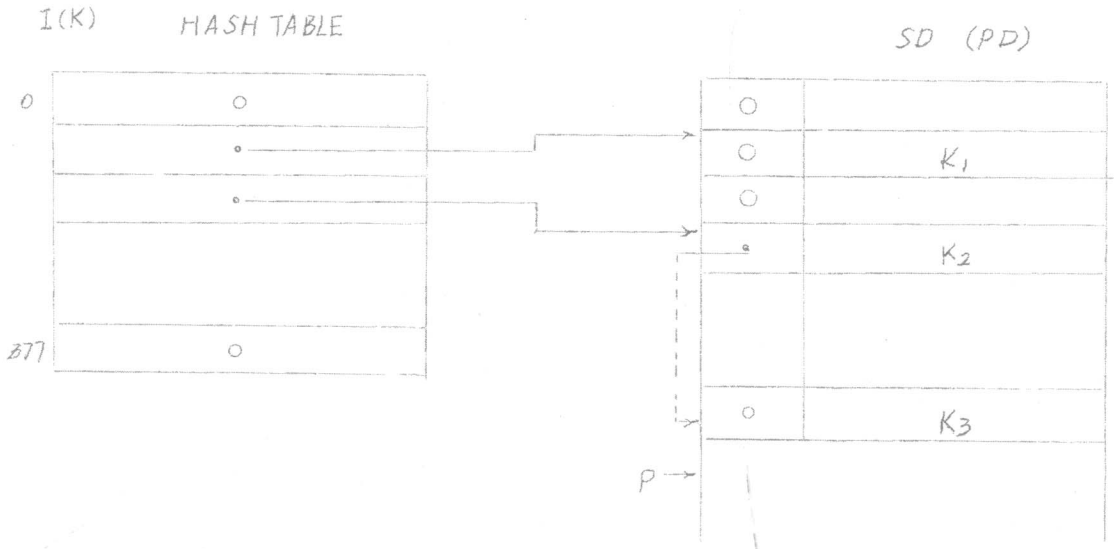
除过程（CAT1 = 3）和哑数组（CAT1 = 2，CAT2 = 1）不分配相对地址，其余都分配相对地址。对于数组 P1 栏是指向数组字，实数组的头地址是通过 P1 栏填入数组字的，对于函数定义段虽然种属也为 PR，但分配相对地址、为该段 V 区的第一个单元。语函定义句中的哑元不进 SD 字典，因为它是局部于句的，语函名却是局部于段、要分配相对地址的。公用块名也不进 SD 字典。

(4) NAME 栏

占一个全字长，鬼行码右齐存放，最多允许6个字符。

〈附注〉 关于链式杂凑法的查填方案

这种方法的思想是利用一张杂凑表（HASH TABLE）



右为杂凑表每项半字长，存放SD字典项地址的，该杂凑表的地址从0—377（八进制）共有256个单元杂凑函数I(K)的取值范围就是HASH TABLE表的地址，自变量K称为关键字，现在对我们来说是符号名。I(K)的选取是将符号名两个半字定点加在自然抽取中间八位二进制。

右为中间语言区，因为各字典项和文本项，都是各自以链进接，所以统一用一个指示字P，它指向自由项，每填完一项后加1。上图符号名 K_2 和 K_3 是同一杂凑码的情况，彼此以链链起来。

2. 数组字字典（FAD）

TY^3			CHAIN
I^2	l^1	$FLAG^2$	W_0
$\bar{W}_0$			
L^1	d_1		
L^1	d_2		
L^1	d_3		

TY 同 SD 字典

I 数组维数（1—3）

$I=\begin{cases}1 & \text{只有 } d_1 \text{ 栏}\\ 2 & \text{只有 } d_1, d_2 \text{ 栏}\\ 3 & d_1\ d_2\ d_3 \text{ 栏都有}\end{cases}$

$l=\begin{cases}0 & \text{表示一个数组元素占一个全字长} \quad D=2\\ 1 & \text{表示一个数组元素占两个全字长（对于复型）} D=4\end{cases}$

FLAG 同 SD 字典 CAT2

W_0 为第一阶段结束分配的数组头

$\bar{W}_0=W_0-(1+d_1+d_1*d_2)*D$

d_1，d_2，d_3 相应三个维说明的指示字。

$L=\begin{cases}1 & d_i \text{ 为 SD 字典指示字}\\ 0 & d_i \text{ 为 CD 字典指示字}\end{cases}$

3、等价字典 EQD

每先填一项、顺序查填、头 EQD。放在通讯表。

每先项链先中第一个元素，以下先中各元素顺序进结，回

将最后一个元素和该片第一个元素连结起来即片中元素是循环连结。

片项和元素项的结构如下：

A_0'	A_1'	$CHAIN_1$
P_1		

$CHAIN_2$	
I^2	X_1
I_1	
J_1	
K_1	
offset	

A_0 = { 1 拼法后被链标志。 0 拼法后为独立等价片。

A_1 = { 1 相对地址已分配标志。 0 〃 尚未 〃 。

$CHAIN_1$ 链下一片项。

P_1 链片中第一元素项。

$CHAIN_2$ 链该片中下一元素项。

X_1 元素在SD字典指示字

I：下标个数

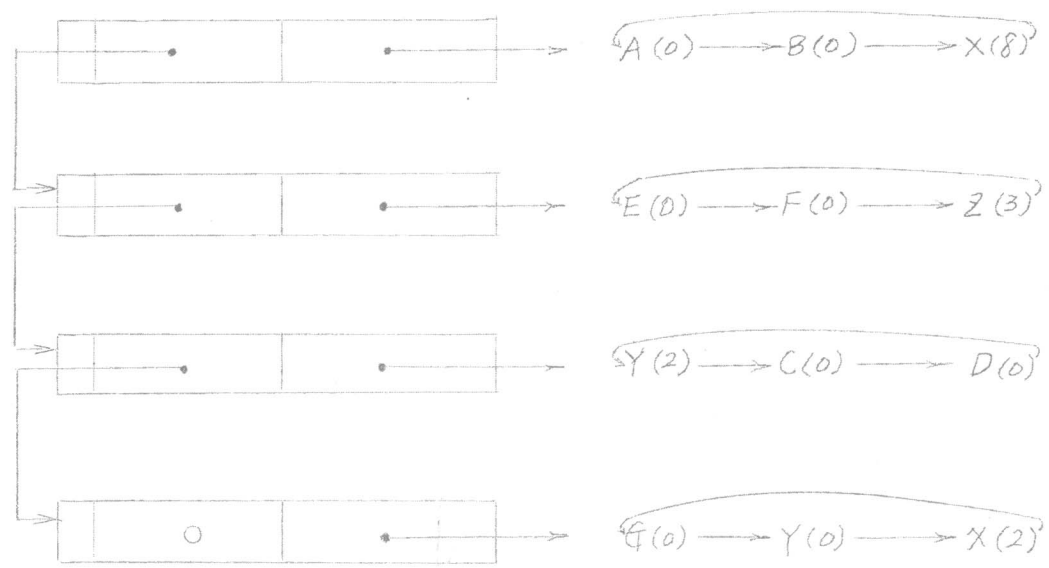
I_1，J_1，K_1 为三个下标值的指示字。

offset 位移栏 $offset = \sum (y_i - 1) * \bar{d}_i * L$ $\bar{d}_i$ 为部分体积

举例说明等价句的处理过程：

例 COMMON (9), G
EQUIVALENCE (A, B, X(2,2)), (E, F, Z(4)), (Y(3), C, D)
DOUBLE PRECISION X(3,3)
DIMENSION Y(5), Z(6)
EQUIVALENCE (G, Y(1), X(2,1))

(1) 去状态表建立等价字典（同时填好SD字典等价标志 A_0 栏并将元素项地址填于 P_1 栏、如为数组元素无素项地址填于数组字头 W_0 的位置上，填的过程中已填不再填）

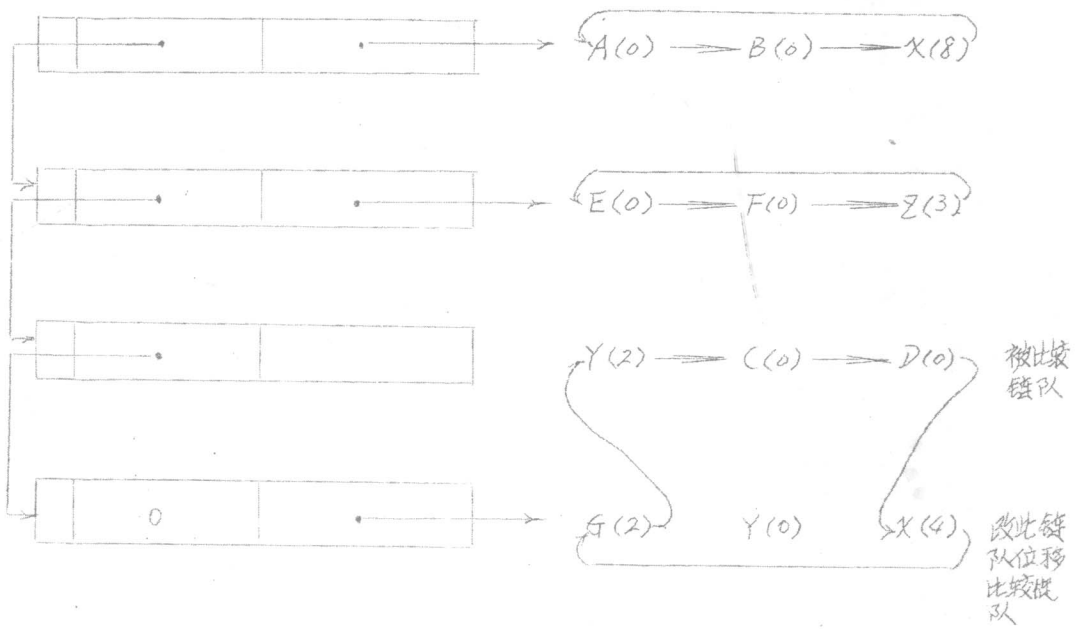


(2) 段结束扫描一遍等价字典（头EQD。在通讯表）这一框命名为 END_1

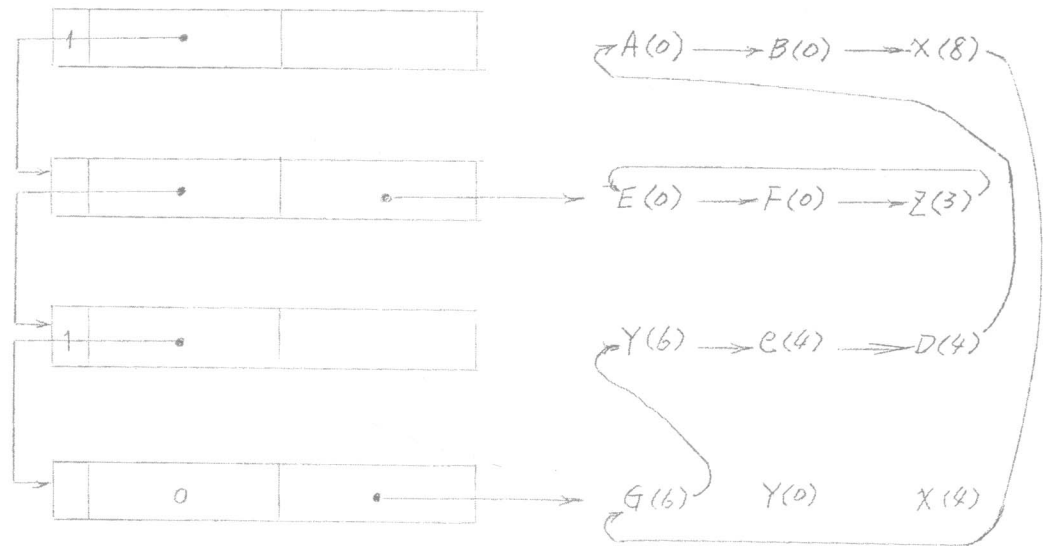
填的位移栏，上面(1)中已填好，每个元素后面括号中即表示位移。

(3) 段结束时二次扫描EQD（再从EQD开始、沿链进行）进行拼片工作。这一框命名为 END_2

一个先项一个先项进行，每个先项一个元素项一个元素项进行，顺序做指示字比较。每个先项的元素组成一个链队，第一个元素为队首，最后一个元素为队尾，队尾链向队首。在比较过程中如果没有发现相关元素时，被比较的元素的链地址等于该先队首地址时，对被比较链队的比较工作结束。一旦发现相关元素时，立刻修改比较链队位移，改旧链为新链，即将该队相关元素前一元素链向被比较链队的队首被比较链队的队尾链向此相关元素的下一个元素，同时将被比较链队的先项置疫链标志为1此时对比较链队的工作也告结束。上例首先发现第四链队（比较链队）第二元素和第三链队（被比较链队）第一元素相关，则改为如下形式



对被比较的链队工作结束时，再沿先链找一下先项比较过程中，如有疫链标志不做，继续找一下先项，直到没有疫链标志，再重复上述比较工作，上例再比较成如下形式。



当比较链队的元素的 链地址等于该链队队首地址时比较链队的工作结束。

疫链标志为0者即为各独立等价先现在为

E(0)，F(0)，Z(3)

G(6)，Y(6)，C(4)，D(4)，A(0)，B(0)，X(8)

(4) 拼先结束后转段结束时的单元分配工作：这一框命名为 END_3

i) 先分配公用、共名元素、扫描CDD字典分配单元，填好SD字典相应 P_1 栏、同时在SD字典 A_2 栏置好已分配标志。

对CDP字典是一块一块进行（有名块一个块名算一块、一级所有无名公用元素算一块、一个共名元素算一块），在每块进行分配时是一个元素进行，对于每一个元素首先根据在SD字典指示字、查其是否有参与等价标志，一旦发现参与等价、链时中断公用一共名分配由SD字典的 P_1 栏或相应此组字 W_0

栏（如为数组元素）先去EQD找到相应的独立等价片，将其所有元素都分配在公用一共名区，置好相应独立片项A_1已分配标志改相应SD字典CAT为公用共名标志。检查是否图头，记录该块现行长度再回中断点继续进行分配，块结束时，将其该块起始点和长度填好。

注意有名块和无名块是分配在公用区、共名块分配在共名区，每区起始点为0，各块顺序分配。

ii）再扫描一遍EQD对$A_2=0$的独立等价片进行局部易分配、填好SD字典相应P_1栏和A_2栏（已分配标志）

iii）最后扫描一遍SD字典对于$A_2=0$项进行局部易分配，注意在SD字典中对于CAT1≥3的项除'SF'，'FU'外在独立SD字典项的同时A_2已置1，这些过程是不分配相对地址的第三阶段填链地址、第四阶段为过程入口地址。

有一点尚未说到的是、在进行各独立等价片分配的同时，还要检查片内各元素的相容性、因为排好的独立等价片内还可能出现下述情况。

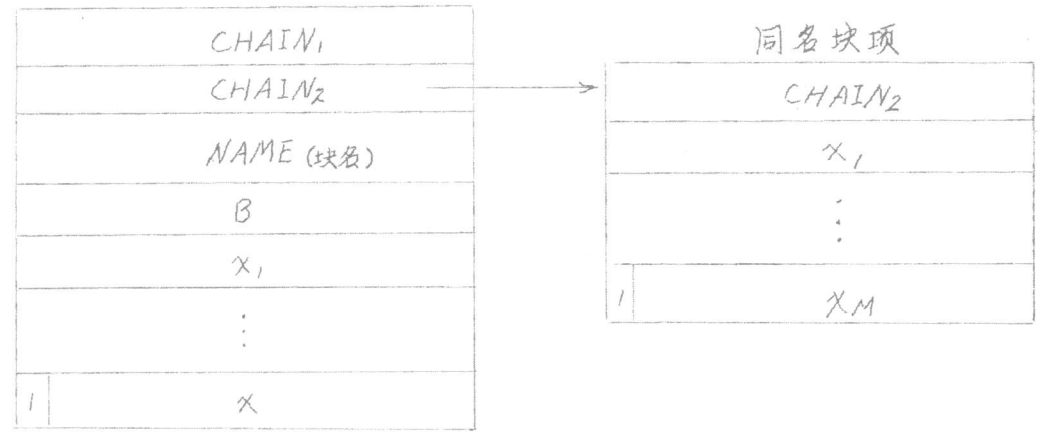
→ A(5) → B(6) → A(7) → C(0)

当分配A(7)时，A(5)已分配，此时判不相容错。

下述情况也可能出现但是允许的。

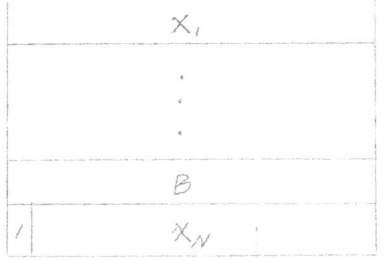
→ A(5) → B(6) → A(5) → C(0)

4、公用字典（CMD）



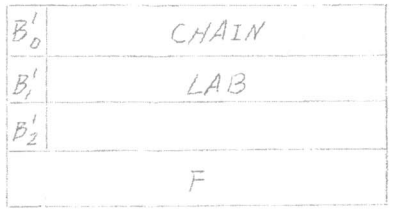
对于无名块、名字栏为空、按块名出现的顺序分配地址、每块的末地址置于B字栏中。

5. 共名字典（PUD）



按元素出现的顺序在共名区分配，其末地址置于B字栏中，第三阶段需根据元素指示字保留其名字。

6、标号字典（LD）



A

$B_0=\begin{cases}1 & \text{内部标号}\\ 0 & \text{一般。}\end{cases}$ $B_1=\begin{cases}1 & \text{定义}\\ 0 & \text{供用}\end{cases}$ $B_2=\begin{cases}1 & \text{不可供用}\\ 0 & \text{可供用}\end{cases}$

LAB 标号 整常数存放

F 定义标号文本项指示字。

7、格式标号字典（FLD）

CHAIN	
B_1'	LAB
F	

$B_1=\begin{cases}1 & \text{定义}\\ 0 & \text{供用}\end{cases}$

LAB 格式标号 整常数存放

F 定义性出现为格式程序的入口地址

8、常数字典（CD）

TY[3]	CHAIN
常数	

TY 1 2 3 4 5 6

整 实 (双) 复 逻 文

除复型外两个全字长外，其余常数都占一个全字长

文字型常数最多8个字符，左齐宽行码存放

9、地址参数字典（NAD）

l'	N[7]	CHAIN[16]
A^2	B^6	X_1
		⋮
A^2	B^6	X_N

$l=\begin{cases}1 & \text{语证引用的实参地址表}\\ 0 & \text{外部过程参数表}\end{cases}$

外部过程参数表又分哑参地址表和实参地址表

对于哑参表 (1) A字栏为哑变号引用定义特征

$A_0=1$ 引用 $A_1=1$ 定义

非哑变号 即哑数组、哑过程无用。

(2) B字栏为10、因为对哑参情况 $X_1 \cdots X_n$ 都是指的SD指示字。

对于实参地址表 (1) A字栏为类型

$B=\begin{cases}10 & \text{SD 字典} & \text{实之为变量名、数组名、过程名}\\ 04 & \text{CD 字典} & \text{实之为常数}\\ 20 & h & \text{实之表达式}\\ 22 & h\text{-}add & \text{实之为数组元素}\end{cases}$

10、数据置初值字典（DD）

DIMENSION A(2)

例 DATA A, B(3), C/3*0.6, 2HAB/

	YFW·d	45	W d Y	
	YEW·d	44	W d Y	
	YDW·d	43	W d Y	（本编译处理为44）
	YGW·d	46	W d Y	
	YIW	50	W O Y	
	YBW	41	W O Y	
	YLW	53	W O Y	
	YAW	40	W O Y	
	nP	55	O O n	符号住为d拉的首址
内层	Y(	73	O O	
内层	)	74	O O O	
	nT	63	O n	
	nZ（字符）	63	O n	指定列格式
	W			
	11…1	62	O O Y	
	Y			
	nX	67	77 O n	
	Y	57		
	nH $h_1 \cdots h_n$	47	n	
			$h_1 h_2 \cdots$	
			$\cdots h_n$	
最外层		76	addr	

addr 为格式指令返回地址，如有内层括号、填最右边内层括号对应的左括号格式指令地址否则填最外层括号对应的左括号格式指令地址，而PPP根据名表是否处理完，决定其是否使用这一地址。

CHAIN	
A	
A	
1 B	4
2	
C	
0.6	
0.6	
0.6	
AB	

ADD

(1) 对于数组、变易直接填SD字典指示字，数组元素有一个相对于假头的位移

(2) 查复数打开存放，即在字典记有当复数，常的直接存放。

(3) ADD是常数开始存放的起始地址，为了保证从偶地址开始，如ADD二奇数，则空半字。从下半字开始。

II. 格式字字典（FD）

最外层 （

CHAIN
ADD

ADD. 为该格式句最末一条格式指令地址。

CHAIN 连下一格式句。

A

§3 可优文本项结构

1 标术运标先项

1	TY²	BI²	OP³	CHAIN 16
T'	d'	A 6		X_1
T'	d'	A 6		X_2
				⋮
T'	d'	A 6		X_N

TY: 0 1 2 3

复 查 实 (双)

BI: 第一阶段不用，第二阶段用

OP: 1 2 3

\+ × **

当 OP = + d = 1 表示那二元式的运标为减。其余无用

OP = × d = 1 表 〃 除。

T = 1 为原元素。

A = { 10 SD; 04 CD; 20 h; 22 h-add }

(指)

(底)	I	R	C
I	I	R	×
R	R	R	×
C	C	×	×

**

右

(左)	I	R	C
I	I	×	×
R	×	R	C
C	×	C	C

\+ − × /

A

2. 标术赋值句

1	TY²	OP³	CHAIN 16
	A 6		C: ·
	A 6		X_1
			⋮
1	A 6		X_N

TY 为表达式类型 TY 0、1、2、3

复 查 实 (双)

OP: 4

e前 A = { 10 C; 20 22 }

x_i 前 A = { 10 22 }

3. 基本外部函数、内部函数、附加内部函数

1	2	2	OP³	CHAIN
NUM				
	A 6			X_1
				⋮
1	A 6			X_N

OP 5 6 7

BIF INF AINF

NUM　在相应函数表中的序号（对于BEF 填入口地址）

$$A=\begin{cases}10\\04\\20\\22\end{cases}$$

尾标只有INF中求最大值、最小值才是有用的，其余无用，因为个数是固定的。

4　下标地址计标

基本上同标准运标先项结构，但加假头的二元式的A字栏为02，且加假头的先一定是下标地址计标的最后一先。

例如　A(10)的下标地址计标由两先组成。

T_1　＊　10

　　＊　D　　设D=2

T_2　＋　T_1

　　＋　$\overline{\omega_0}$.

即

T_1

1	TY 2	2	＊ 3	CHAIN 16
	04			10
1	04			2

T_2

1	TY		＋ 3	CHAIN 16
	20			T_1
1	12			$\overline{\omega_0}$

A　因此在下标地址计标　A字栏可能有四种情况

$$A=\begin{cases}10 & SD\\04 & CD\\20 & k\\02 & FAD\end{cases}$$

（只在数组元素地址计标最后一先中加假头的二元式用到）

显然在其他地方使用T2的二元式的A字栏即为22。

§4　不可优文本项结构

一般形式

0	OP 7	CHAIN
（因各语句而不同）		

即　OP是七位编码，符号位是0，以区别于可优文本项。

1、段头句

OP：　MS　SUB　BLOCK

主段　无参子程序块数据段

0	OP 7	CHAIN 16
S		

S为段名在SD指示字。

OP：　SUBi　FUN

有参子程序　函数段

0	OP 7	CHAIN 16
S		
NAD		

S　同上

NAD　指向哑参地址表。

2、语函定义句

$F(X_1 \cdots X_N) = e$

l	CHAINI	
0	SF 7	CHAIN
N 8		F
L_1	TY_1 3	
L_2	TY_2 3	
⋮		
L_N	TY_N 3	
e 文本项		

CHAINI　专门链语函定义句　为第三阶段连理用

CHAIN　链中间文本项

N　　哑元个数（限制20个）

F　　语函名在SD指示字

TY_1、TY_2 ⋯ TY_N　语函哑元类型

e　　文本项中对哑元的操作按用该文本项中的假地址 L_1、L_2 ⋯ L_N

e　　文本项最末一项链为0

对于语函的引用　如优化直接在引用处以实元代替哑元进行计标右表达式，否则在第三阶段于V区分配于哑元地址，以便在语函引用处进行哑实结合之用。

$$l = \begin{cases} 1 & \text{右表达式只有一个数组元素} \\ 0 & \text{其他} \end{cases}$$

3、赋标号语句

0	AGN 7	CHAIN
K		
I		

K　标号在LD字典指示字

I　为左变号在SD字典指示字

4、转移语句

三种形式	GOTO	K	无条件转
	GOTO	$(K_1, K_2 \cdots K_N), I$	计标转
	GOTO	$I, (K_1, X_n \cdots K_N)$	赋值转

0	GOTO 7	CHAIN
K		

0	OP 7	CHAIN
N 8		I
K_1		
⋮		
K_N		

OP=GOKI　计标转

GOIK　赋值转

N　标号的个数

K　标号在LD字典指示字

I　左变号在SD字典指示字

5、 算术 IF 句

0	AIF 7	CHAIN
	A 6	e
K1		
K2		
K3		

$A=\begin{cases}10\\20\\30\end{cases}$

#e的类型可以是整实但不能为复型#

6、 逻辑 IF 句

IF (e) S1
S2

编译改造为

IF (e)
n_1 S_1
n_2 S_2

因此一个逻辑 IF 句文本项的顺序为

e 文本项
LIF 文本项
内部标号 n_1 文本项
S_1 文本项
n_2 （可能也是内部标号）文本项
S_2 文本项

0	LIF 7	CHAIN 16
	A 6	e
n_1		
n_2		

$A=\begin{cases}10\\20\\22\end{cases}$ #e为逻辑型#

n_1, n_2 为 LD 字块指示字

7、 外部过程调用（包括 子程序调用、函数段引用、语函引用）

OP: CALLi 有参子程序
CALL 无 〃 （此时没有 NAD 栏）
CFUN 函数段引用
CSF 语函引用

0	OP 7	CHAIN
S		
NAD		

NAD 实参地址表指示字

8、 打、暂打语句

OP STOP 打
PAVSE 暂打

0	OP	CHAIN 10
N3	n	

n 为最多五位八进制数

N＝0 本栏为空， N≠0 为 n 的八进制个数，右齐存放，广指参数区 DS 加 6 个半字长。

9、 返回语句链续语句

0	RETURN 7	CHAIN

链续语句不生成相应文本项

10、 Do 语句

例

```
Do  L  I  M1, M2, M3
---
L  CONTINUE
```

编译认为

```
    I = M1
INL ---
    ---
    ---
L
    I = I+M3
    IF (I.LE.M2) GOTO INL
```

为了优化方便在赋初值文本项和体开始的内部标号文本项之间加一个 Do 文本项它的形式是

0	Do^7	CHAIN
I		
0	M_1	
0	M_2	
0	M_3	
LIF		

I 为循变在 SD 指示字

M_1、M_2、M_3 为循参指示字 $\quad \ell = \begin{cases} 1 & SD \\ 0 & CD \end{cases}$

LIF 终结比较文本项指示字

因此文本项的顺序是

赋初值文本项（$I=M_1$）

Do 文本项

循环体开始处的内部标号文本项（INL）

循环体文本项

加增务文本项（$I=I+M_3$）

终值比较文本项（IF(I.LE.M2)GOTOINL）

11、 I/o 句

```
oP   READ   读
     WRITE  写
```

每一句生成

0	oP	CHAIN
V		
A	f	
$CHAIN_\ell$		

V 文件名（按 ppp 为 1-3 字母数字串）

$$f = \begin{cases} 0 & \text{无格式（固定格式）} \\ \neq 0 & \text{有格式（组编格式）此 } A = \begin{cases} 0 & \text{f 为格式标号在 FLD 指示字。} \\ 1 & \text{f 为数组名在 SD 指示字} \end{cases} \end{cases}$$

$CHAIN_\ell = 0$ 无名表

$\neq 0$ 有名表 链名表中第一元素项

元素项分三种

(1) 变量、数组形式

0	op^7	$CHAIN_\ell$ ^{16}
X		