26243

"Factibilidad de la elaboración de una metodología para la evaluación forestal en un area reducida de la Provincia de Misiones por procesamiento digital de información Landsat ".

Informe del experto

Guillermo Daniel Navas

1º Informe Final
2º Informe de Avance

0
H.1225
N 11
II

PROGRAMA CLISO 2

El programa realiza una clasificación no supervisada de un área utilizando el algoritmo de isodata . La versión implementada corresponde a la utilizada por el sistema LARSYS, según lo expuesto en el manual del usuario de dicho sistema . El método allí documentado es el siguiente :

Sean $x_1, x_2, \ldots x_n$ los N vectores de dimensión L a ser agrupados en M clases o clusters . Sean c_j , μ_j y σ_j^2 (todos vectores de dimension L) el centro , media y variancia, respectivamente, del cluster J-ésimo .

Entonces tenemos :

$$xj = \begin{bmatrix} x_1j \\ x_2j \\ \vdots \\ x_Lj \end{bmatrix} \qquad j = 1, 2, \ldots M$$

y :

$$cj = \begin{bmatrix} c_1j \\ c_2j \\ \vdots \\ c_Lj \end{bmatrix} \qquad \mu_j = \begin{bmatrix} \mu_1j \\ \mu_2j \\ \vdots \\ \mu_Lj \end{bmatrix} \qquad \sigma_j^2 = \begin{bmatrix} \sigma_1^2j \\ \sigma_2^2j \\ \vdots \\ \sigma_Lj^2 \end{bmatrix} \qquad j = 1, 2, \ldots M$$

Para comenzar el algoritmo, debe establecerse un conjunto inicial de centros de clusters . Esto se realiza calculando el vector de medias m y el vector de variancias s^2 para todo el conjunto de vectores de observaciones N . Un paralelepípedo rectangular, que usualmente contendrá un gran porcentaje de los vectores de observaciones, tiene los lados orientados en forma paralela a los ejes de coordenadas y dados por $m_i \pm s_i$ (s_i es la raíz cuadrada de la variancia s_i^2) .

Los centros iniciales de los clusters se eligen uniformemente espaciados a lo largo de la diagonal principal en este paralelepípedo .

De este modo, el centro inicial del cluster j-ésimo está dado por

$$c_{ij} = m_i + s_i \left[\frac{2(j-1)}{M-1} - 1 \right] \qquad \begin{array}{l} i = 1, 2, \ldots\ldots L \\ j = 1, 2, \ldots\ldots M \end{array}$$

Una vez establecidos los centros iniciales, se llega al modo de asignación. Este consiste en determinar la distancia euclídea de cada vector a cada centro de los clusters, y asignar cada vector al cluster con el centro más próximo. Si como resultado de esta asignación no cambia la asignación de clusters de un determinado porcentaje de los vectores, el algoritmo de clustering ha finalizado; de lo contrario se reemplazan los viejos centros de los clusters por las medias de los nuevos clusters y se vuelve al modo de asignación.

El algoritmo descripto se ha implementado con las siguientes modificaciones a pedido de la dirección técnica:

- Posibilidad de no considerar la información contenida en un canal determinado.
- Posibilidad de limitar el proceso a un número arbitrario de iteraciones.
- Generación de la solucion inicial partiendo de un paralelepípedo cuyos lados estan dados por $m_i \pm 2 s_i$.
- Grabación de los valores medios y las matrices de covariancias de cada cluster en un archivo scratch para permitir su posterior confirmación en el archivo de estadísticas del sistema.

Para la implementación se ha tomado :

L : Cantidad de canales (cuatro)

M : Cantidad de clusters (nueve)

Siendo :

CENTRO : Centros de los clusters por cluster por canal

SUMOBS : Suma de las observaciones por cluster por canal

CANOBS : Cantidad de observaciones por cluster

CLUSTER : Cluster al que está asignado cada pixel del área por fila por columna

LINEA : Una línea de las observaciones del área por canal por columna

COVAR : Covariancias por cluster por canal por canal

KV : Covariancias por canal por canal

MED : Medias por canal

RECSTA : Registro de estadísticas

ITMAX : Número máximo de iteraciones

MODCLU : Cantidad de pixels que sufrieron modificaciones en la iteración corriente del cluster al que están asignados

RESNBA : Cantidad de canales

RESNOB : Número de identificación de los canales por canal

ULTFIL : Cantidad de líneas del área

ULTCOL : Cantidad de columnas del área

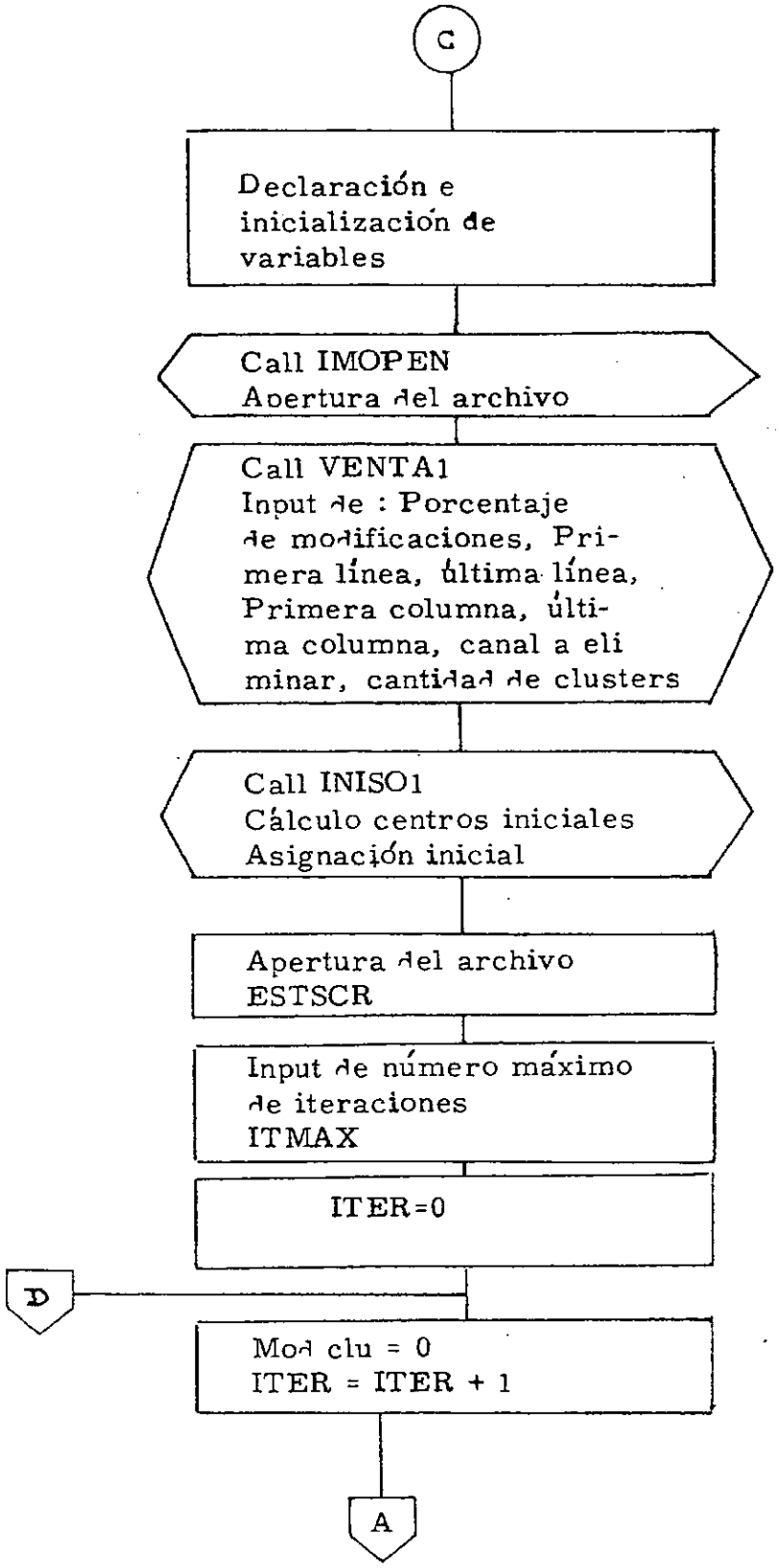
PORMOD : Porcentaje de modificación de pixels del cluster al que están asignados

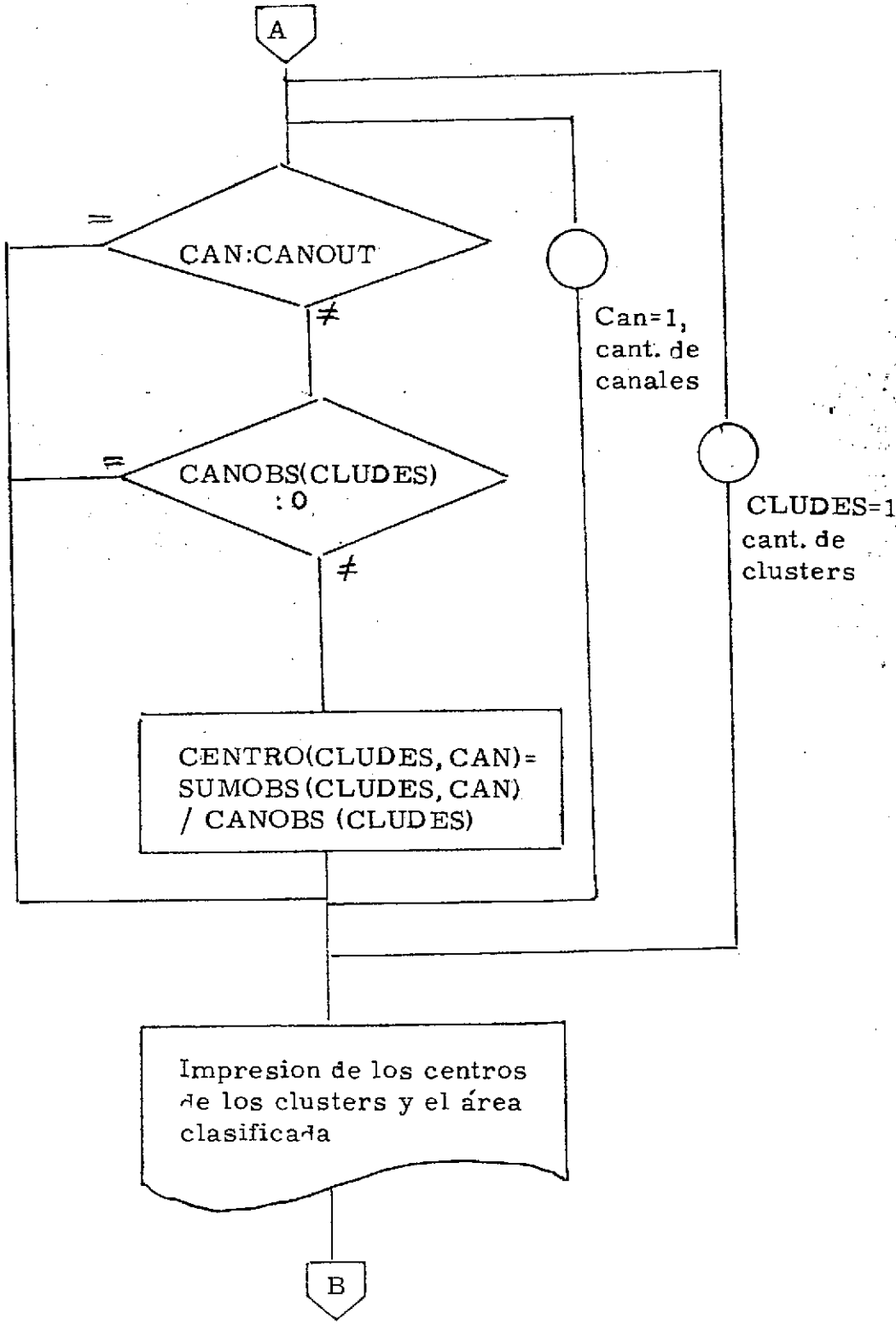
CANCLU : Cantidad de clusters

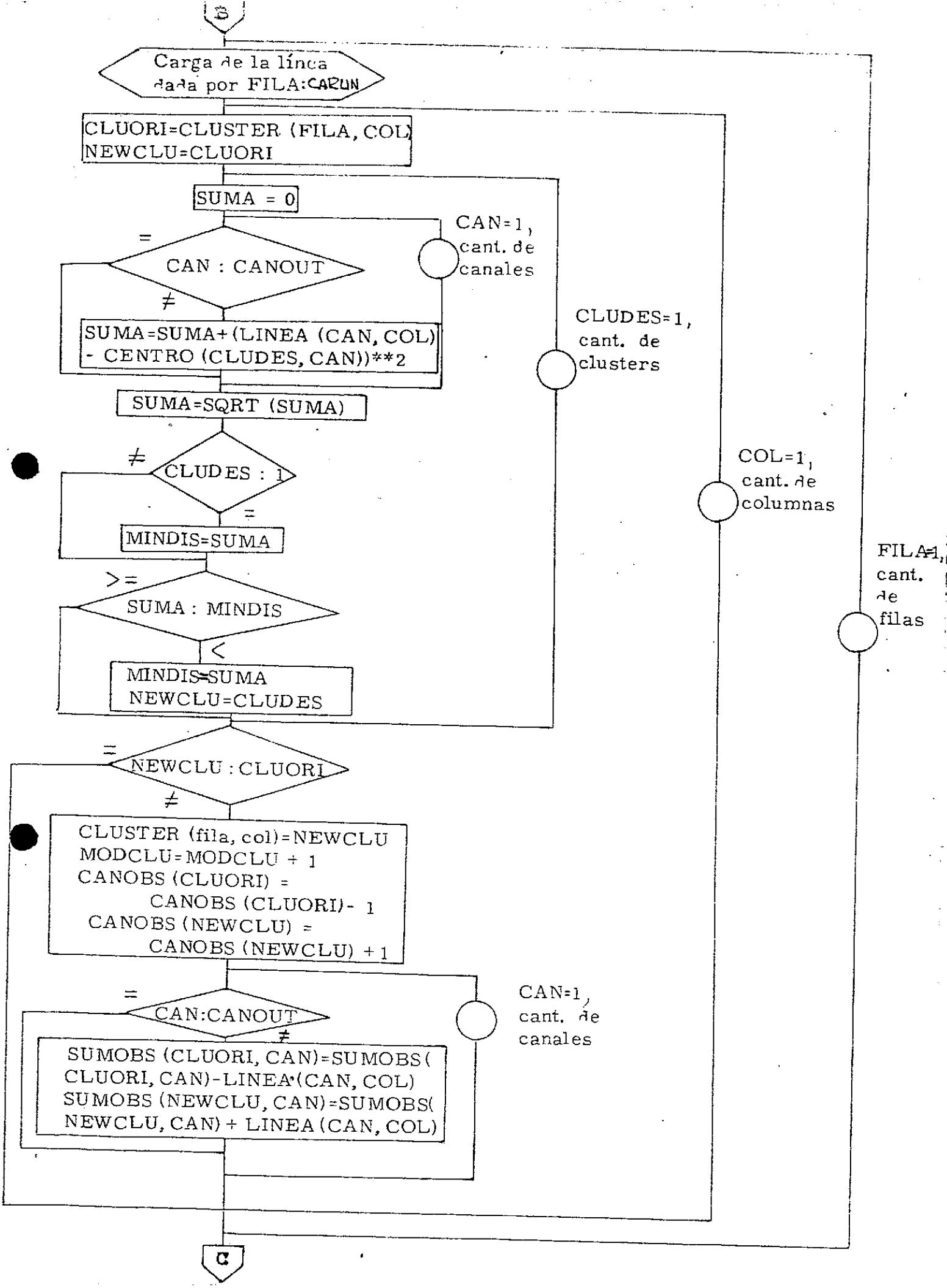
SUBRUTINAS DEL SISTEMA : PI(CNIE)

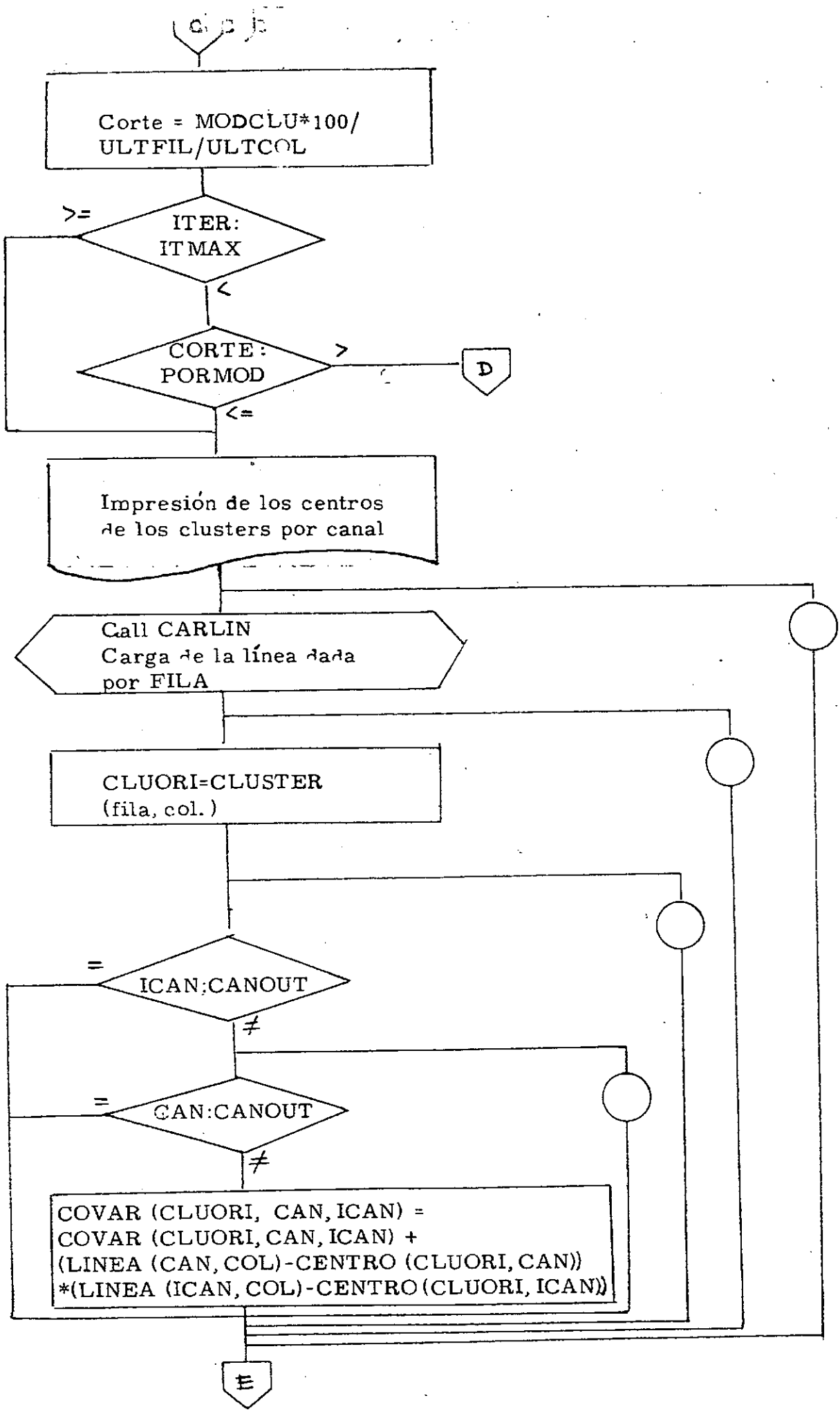
GRASTA - Graba un registro en el archivo de estadísticas abierto sobre el canal 7 (siete), tomando el contenido del vector RECSTA, y las variables IMAGE y KLASE del área common ESTADI.

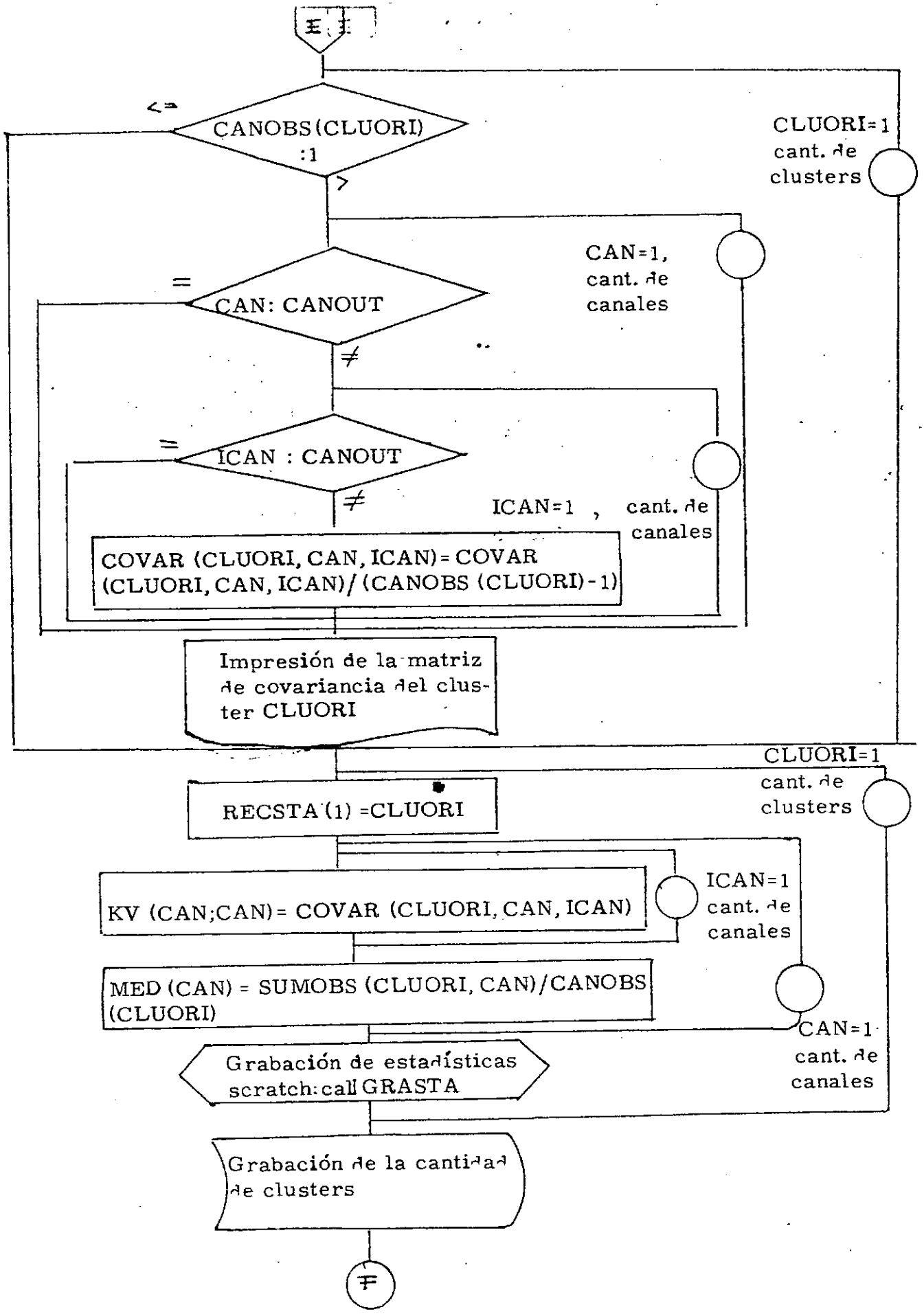
PROGRAMA CLISO2




A
CAN:CANOUT
=
≠
Can=1,
cant. de
canales
CANOBS(CLUDES)
:0
=
≠
CLUDES=1,
cant. de
clusters
CENTRO(CLUDES, CAN)=
SUMOBS(CLUDES, CAN)
/ CANOBS (CLUDES)
Impresion de los centros
de los clusters y el área
clasificada
B


B
Carga de la línea dada por FILA:CARUN
CLUORI=CLUSTER (FILA, COL)
NEWCLU=CLUORI
SUMA = 0
CAN : CANOUT
=
≠
CAN=1, cant. de canales
SUMA=SUMA+(LINEA (CAN, COL) - CENTRO (CLUDES, CAN))**2
CLUDES=1, cant. de clusters
SUMA=SQRT (SUMA)
CLUDES : 1
≠
=
MINDIS=SUMA
COL=1, cant. de columnas
SUMA : MINDIS
>=
<
MINDIS=SUMA
NEWCLU=CLUDES
FILA=1, cant. de filas
NEWCLU : CLUORI
=
≠
CLUSTER (fila, col)=NEWCLU
MODCLU=MODCLU + 1
CANOBS (CLUORI) = CANOBS (CLUORI)- 1
CANOBS (NEWCLU) = CANOBS (NEWCLU) + 1
CAN:CANOUT
=
≠
CAN=1, cant. de canales
SUMOBS (CLUORI, CAN)=SUMOBS(CLUORI, CAN)-LINEA(CAN, COL)
SUMOBS (NEWCLU, CAN)=SUMOBS(NEWCLU, CAN) + LINEA (CAN, COL)
C


Corte = MODCLU*100/
ULTFIL/ULTCOL
>=
ITER:
ITMAX
<
CORTE:
PORMOD
>
D
<=
Impresión de los centros
de los clusters por canal
Call CARLIN
Carga de la línea dada
por FILA
CLUORI=CLUSTER
(fila, col.)
=
ICAN:CANOUT
≠
=
CAN:CANOUT
≠
COVAR (CLUORI, CAN, ICAN) =
COVAR (CLUORI, CAN, ICAN) +
(LINEA (CAN, COL)-CENTRO (CLUORI, CAN))
*(LINEA (ICAN, COL)-CENTRO (CLUORI, ICAN))
E


E
CANOBS(CLUORI) :1
<=
>
CLUORI=1 cant. de clusters
CAN=1, cant. de canales
=
CAN: CANOUT
≠
=
ICAN : CANOUT
≠
ICAN=1 , cant. de canales
COVAR (CLUORI, CAN, ICAN)= COVAR (CLUORI, CAN, ICAN)/ (CANOBS (CLUORI)-1)
Impresión de la matriz de covariancia del cluster CLUORI
CLUORI=1 cant. de clusters
RECSTA (1) =CLUORI
KV (CAN;CAN)= COVAR (CLUORI, CAN, ICAN)
ICAN=1 cant. de canales
MED (CAN) = SUMOBS (CLUORI, CAN)/CANOBS (CLUORI)
CAN=1 cant. de canales
Grabación de estadísticas scratch: call GRASTA
Grabación de la cantidad de clusters
F

```
      C
      C     ******************** CLISO2 ****************************
      C
0001          INTEGER*2 RECSTA(286),MED(11)
0002          REAL*4 MINDIS,KV(11,11)
0003          EQUIVALENCE (RECSTA(23),KV(1,1)),(RECSTA(265),MED(1))
0004          REAL*4 SUMOBS(32,4),COVAR(9,4,4)
0005          DATA COVAR/144*0./RECSTA/286*0/
0006          INTEGER*2 RESNLI,RESNCO,RESNBA,RESNOB(4),CANCLU
0007          COMMON RESNLI,RESNCO,RESNBA,RESNOB
0008          INTEGER*2 LINCOL(6),ERROR,FILA,COL,ULTFIL,ULTCOL,CLUORI,CLUDES
0009          DIMENSION VARIAN(4),CENTRO(32,4)
0010          INTEGER*2 CAN,LINEA(4,120),CANOBS(32),CANOUT
0011          LOGICAL*1 CLUSTER(50,120)
0012          COMMON /DATISO/ LINEA,CANOBS,CLUSTER,VARIAN,CENTRO,LINCOL
             *  ,CANCLU,SUMOBS,CANOUT
0013          INTEGER*4 KLASE(2),IMAGE(2)
0014          COMMON /ESTADI/ RECSTA,NUMREG,KLASE,IMAGE
      C
      C     ************************************************************
      C
0015          TYPE 123
0016    123   FORMAT('   COLOQUE CINTA PI EN UNIDAD 0 ')
0017            READ(5,*)NADA
0018          CALL IMOPEN
0019          CALL VENTA1(PORMOD,ERROR,CANOUT)
0020          CALL INISO1
0021          OPEN(UNIT=8,NAME='DK1:ESTSCR.DAT',TYPE='UNKNOWN',RECORDSIZE=286
             *   ,ACCESS='DIRECT')
0022          ULTFIL=LINCOL(5)
0023          ULTCOL=LINCOL(6)
0024          DO 101 I=1,4
0025    101   RECSTA(11+I)=RESNOB(I)
0026          RECSTA(11)=RESNBA
      C
      C     ************************************************************
      C
0027          TYPE 1513
0028    1513  FORMAT(' NUMERO MAXIMO DE ITERACIONES ?'$)
0029          READ(5,*)ITMAX
0030          ITER=0
0031    100   MODCLU=0
0032          ITER=ITER+1
0033          DO 2000 CLUDES=1,CANCLU
0034           DO 2010 CAN=1,RESNBA
0035             IF(CAN.EQ.CANOUT) GO TO 2010
0037             IF(CANOBS(CLUDES).NE.0) CENTRO(CLUDES,CAN)=
             *              SUMOBS(CLUDES,CAN)/CANOBS(CLUDES)
0039    2010   CONTINUE
0040    2000  CONTINUE
      C
      C     ******************************************
      C
0041    1111                WRITE(6,5555)
0042    5555                FORMAT('1')
0043                        DO 2222 J=1,RESNBA
0044                        WRITE(6,7777) (CENTRO(I,J),I=1,CANCLU)
0045    2222                WRITE(6,8888)
0046    7777                FORMAT(' ',12F10.5)
0047                        WRITE(6,8888)
```

```
0048                    DO 3333 I=1,ULTFIL
0049                    WRITE(6,8888) (CLUSTER(I,J),J=1,ULTCOL)
0050  3333              CONTINUE
0051  8888              FORMAT(' ',132I1)
      C
      C     ************************************
      C
0052          DO 1000 FILA=1,ULTFIL
0053           CALL CARLIN(FILA)
0054           DO 1100 COL=1,ULTCOL
0055             CLUORI=CLUSTER(FILA,COL)
0056          NEWCLU=1
0057             DO 1120 CLUDES=1,CANCLU
0058               SUMA=0
0059               DO 1121 CAN=1,RESNBA
0060              IF(CAN.EQ.CANOUT) GO TO 1121
0062              SUMA=SUMA+(LINEA(CAN,COL)-CENTRO(CLUDES,CAN))**2
0063  1121         CONTINUE
0064               SUMA=SQRT(SUMA)
0065               IF(CLUDES.EQ.1) MINDIS=SUMA
0067               IF(SUMA.GE.MINDIS) GO TO 1120
0069               MINDIS=SUMA
0070               NEWCLU=CLUDES
0071  1120       CONTINUE
0072             IF(NEWCLU.EQ.CLUORI) GO TO 1100
0074             CLUSTER(FILA,COL)=NEWCLU
0075             MODCLU=MODCLU+1
0076             CANOBS(CLUORI)=CANOBS(CLUORI)-1
0077             CANOBS(NEWCLU)=CANOBS(NEWCLU)+1
0078             DO 1130 CAN=1,RESNBA
0079               IF(CAN.EQ.CANOUT) GO TO 1130
0081               SUMOBS(CLUORI,CAN)=SUMOBS(CLUORI,CAN)-LINEA(CAN,COL)
0082               SUMOBS(NEWCLU,CAN)=SUMOBS(NEWCLU,CAN)+LINEA(CAN,COL)
0083  1130       CONTINUE
0084  1100    CONTINUE
0085  1000   CONTINUE
0086         CORTE=MODCLU*100./ULTFIL/ULTCOL
0087         IF(ITER.GE.ITMAX)GO TO 5341
0089         IF(CORTE.GT.PORMOD) GO TO 100
0091  5341   CONTINUE
0092         WRITE(6,900)
0093         WRITE(6,901)(CLUDES,CLUDES=1,CANCLU)
0094         DO 800 CAN=1,RESNBA
0095  800    WRITE(6,902)CAN,(CENTRO(CLUDES,CAN),CLUDES=1,CANCLU)
0096         DO 3000 FILA=1,ULTFIL
0097          CALL CARLIN(FILA)
0098          DO 3010 COL=1,ULTCOL
0099            CLUORI=CLUSTER(FILA,COL)
0100         DO 3020 ICAN=1,RESNBA
0101         IF(ICAN.EQ.CANOUT)GO TO 3020
0103            DO 3020 CAN=1,RESNBA
0104              IF(CAN.EQ.CANOUT) GO TO 3020
0106         COVAR(CLUORI,CAN,ICAN)=COVAR(CLUORI,CAN,ICAN)+
            $       (LINEA(CAN,COL)-CENTRO(CLUORI,CAN))*
            $       (LINEA(ICAN,COL)-CENTRO(CLUORI,ICAN))
0107  3020       CONTINUE
0108  3010    CONTINUE
0109  3000   CONTINUE
0110         DO 3100 CLUORI=1,CANCLU
0111          IF(CANOBS(CLUORI).LE.1) GO TO 3100
```

```
0113          DO 3300 CAN=1,RESNBA
0114            IF(CAN.EQ.CANOUT) GO TO 3300
0116            DO 3310 ICAN=1,RESNBA
0117              IF(ICAN.EQ.CANOUT) GO TO 3310
0119              COVAR(CLUORI,CAN,ICAN)=COVAR(CLUORI,CAN,ICAN)/
            $                 (CANOBS(CLUORI)-1)
0120  3310      CONTINUE
0121  3300    CONTINUE
0122          WRITE(6,750)CLUORI
0123          WRITE(6,755)(CAN,CAN=1,RESNBA)
0124          DO 3400 CAN=1,RESNBA
0125            WRITE(6,770)CAN,(COVAR(CLUORI,CAN,ICAN),ICAN=1,RESNBA)
0126  3400    CONTINUE
0127          WRITE(6,770)
0128  3100  CONTINUE
0129        DO 90 CLUORI=1,CANCLU
0130          RECSTA(1)=CLUORI
0131          DO 91 CAN=1,RESNBA
0132            DO 92 ICAN=1,RESNBA
0133              KV(CAN,ICAN)=COVAR(CLUORI,CAN,ICAN)
0134  92        CONTINUE
0135            MED(CAN)=SUMOBS(CLUORI,CAN)/CANOBS(CLUORI)
0136  91      CONTINUE
0137          CALL GRASTA(IEND)
0138  90    CONTINUE
0139  750   FORMAT(' MATRIZ DE COVARIANCIA DEL CLUSTER ',I2,/)
0140  755   FORMAT(' ',4X,4I7)
0141  770   FORMAT(' ',I4,4F7.2)
0142  900   FORMAT(' VALORES MEDIOS DE LOS CLUSTERS'/)
0143  901   FORMAT(' ',4X,32I4)
0144  902   FORMAT(' ',I5,32F4.0)
0145  9999  STOP
0146        END
```

```
FORTRAN IV       STORAGE MAP FOR PROGRAM UNIT .MAIN.

LOCAL VARIABLES, .PSECT $DATA, SIZE = 001276 (  351. WORDS)

NAME   TYPE  OFFSET     NAME   TYPE  OFFSET     NAME   TYPE  OFFSET
CAN    I*2   001204     CLUDES I*2   001202     CLUORI I*2   001200
COL    I*2   001172     CORTE  R*4   001234     ERROR  I*2   001166
FILA   I*2   001170     I      I*2   001214     ICAN   I*2   001240
IEND   I*2   001242     ITER   I*2   001220     ITMAX  I*2   001216
J      I*2   001224     MINDIS R*4   001162     MODCLU I*2   001222
NADA   I*2   001206     NEWCLU I*2   001226     PORMOD R*4   001210
SUMA   R*4   001230     ULTCOL I*2   001176     ULTFIL I*2   001174

COMMON BLOCK /      /, SIZE = 000016 (    7. WORDS)

NAME   TYPE  OFFSET     NAME   TYPE  OFFSET     NAME   TYPE  OFFSET
RESNLI I*2   000000     RESNCO I*2   000002     RESNBA I*2   000004
RESNOB I*2   000006

COMMON BLOCK /DATISO/, SIZE = 017620 ( 4040. WORDS)

NAME   TYPE  OFFSET     NAME   TYPE  OFFSET     NAME   TYPE  OFFSET
LINEA  I*2   000000     CANOBS I*2   001700     CLUSTE L*1   002000
VARIAN R*4   015560     CENTRO R*4   015600     LINCOL I*2   016600
CANCLU I*2   016614     SUMOBS R*4   016616     CANOUT I*2   017616

COMMON BLOCK /ESTADI/, SIZE = 001116 (  295. WORDS)

NAME   TYPE  OFFSET     NAME   TYPE  OFFSET     NAME   TYPE  OFFSET
RECSTA I*2   000000 EQV NUMREG I*2   001074     KLASE  I*4   001076
IMAGE  I*4   001106     KV     R*4   000054 EQV MED    I*2   001020 EQV

LOCAL AND COMMON ARRAYS:

NAME     TYPE    SECTION OFFSET  ------SIZE----- DIMENSIONS
CANOBS I*2       DATISO  001700  000100 (   32.) (32)
CENTRO R*4 VEC   DATISO  015600  001000 (  256.) (32,4)
CLUSTE L*1 VEC   DATISO  002000  013560 ( 3000.) (50,120)
COVAR  R*4 VEC   $DATA   000000  001100 (  288.) (9,4,4)
IMAGE  I*4       ESTADI  001106  000010 (    4.) (2)
KLASE  I*4       ESTADI  001076  000010 (    4.) (2)
KV     R*4 VEC   ESTADI  000054  000744 (  242.) (11,11)
LINCOL I*2       DATISO  016600  000014 (    6.) (6)
LINEA  I*2 VEC   DATISO  000000  001700 (  480.) (4,120)
MED    I*2       ESTADI  001020  000026 (   11.) (11)
RECSTA I*2       ESTADI  000000  001074 (  286.) (286)
RESNOB I*2       .$$$$.  000006  000010 (    4.) (4)
SUMOBS R*4 VEC   DATISO  016616  001000 (  256.) (32,4)
VARIAN R*4       DATISO  015560  000020 (    8.) (4)

SUBROUTINES, FUNCTIONS, STATEMENT AND PROCESSOR-DEFINED FUNCTIONS:

NAME   TYPE   NAME   TYPE   NAME   TYPE   NAME   TYPE   NAME   TYPE
CARLIN  R*4   GRASTA  R*4   IMOPEN  I*2   INISO1  I*2   SQRT    R*4
VENTA1  R*4
```

PROGRAMA CLISO 2

Output

1. Salida impresa

El programa imprime el contenido de la matriz CLUSTER, o sea el número del cluster al que queda asignado cada pixel del área clasificada .

Luego imprime los valores medios de los clusters por canal, y la matriz de covariancias de cada cluster por canal .

2. Grabación en disco

El programa graba en un archivo scratch en disco llamado ESTSCR un registro para cada cluster , que contiene los valores medios y la matriz de covariancia de ese cluster .

```
333323443343333332233223444432234322234344422245 4
443332223443333323233344333343235432223433432223 34
333442244433433332223333322332344432223334422223 4
334442454433333232222233333322343343222223433223 3
434532455445544222232122224454224444432122343333 33
444522443455555322322222233445322322443222122223 35
434442443345555323332232234443332222321123222233 3
434443543444455422233322244444333232222233322222 33
334433543334433343422332224444443331233333223333 33
443432344444444432221123332234444422223332222222 33
443444331232322344441124543323443321234531333322 22
344453344444555554442123453322233311235511233322 21
444453334444554444432223344332112321334532222223 33
544443345444444554422222333312332221233344333333 33
344433355555446664323321112123333223344333333333 33
334333335555556543322222333223333223433333333223 33
553444456666555544443444333323443124343334433234 32
443345556666665544443433343332443334343334333331
453334456666565534334433333333234443444443333433 42
543334566666444444323442322232333344444334333343 31
443344566666445544422233332233333444444433333343 31
443345665665445545443223432344443444434433333444 42
543345566654555545543322333445544434434542344445 55
444555666653445454554331123445533445444443244444 44
344555666665544443664333233444443444444444444344 444
455555556654444554553223233443343344434444444344 444
444555554445555654566321323444344344344345444444 44
444444444455565544555332223344343334533334444444 45
333333455566666554454333433445553334434523223433 55
433444555555556654444332333444555444433432333443 34
```

VALORES MEDIOS DE LOS CLUSTERS

	1	2	3	4	5	6
1	23.	24.	25.	25.	26.	26.
2	26.	27.	28.	28.	29.	28.
3	18.	21.	24.	26.	28.	34.
4	47.	57.	66.	74.	83.	96.

MATRIZ DE COVARIANCIA DEL CLUSTER 1

	1	2	3	4
1	2.96	4.38	-1.85	2.98
2	4.38	10.91	-4.42	7.69
3	-1.85	-4.42	4.31	2.96
4	2.98	7.69	2.96	20.96

MATRIZ DE COVARIANCIA DEL CLUSTER 2

	1	2	3	4
1	3.43	4.55	-3.27	0.59
2	4.55	10.67	-7.55	1.76
3	-3.27	-7.55	6.51	1.27
4	0.59	1.76	1.27	7.45

MATRIZ DE COVARIANCIA DEL CLUSTER 3

	1	2	3	4
1	3.40	4.84	-4.25	1.03
2	4.84	11.24	-9.74	2.36
3	-4.25	-9.74	9.53	0.06
4	1.03	2.36	0.06	6.47

MATRIZ DE COVARIANCIA DEL CLUSTER 4

	1	2	3	4
1	3.90	6.43	-5.96	1.06
2	6.43	15.93	-15.07	1.79
3	-5.96	-15.07	15.16	-0.18
4	1.06	1.79	-0.18	5.05

MATRIZ DE COVARIANCIA DEL CLUSTER 5

	1	2	3	4
1	4.14	5.94	-5.24	2.04
2	5.94	13.29	-12.18	2.93
3	-5.24	-12.18	12.55	0.34
4	2.04	2.93	0.34	10.16

MATRIZ DE COVARIANCIA DEL CLUSTER 6

	1	2	3	4
1	2.25	2.85	-4.02	-0.42
2	2.85	7.78	-11.32	-2.82
3	-4.02	-11.32	22.40	18.77
4	-0.42	-2.82	18.77	40.07

PROGRAMA EEDCLU

Este programa permite confirmar las estadísticas generadas por la última ejecución del programa CLISO2 en el archivo scratch ESTSCR . De este modo , eligiendo el número del cluster a confirmar, e indicando el nombre y número (permite almacenar varias zonas) de la clase, y el nombre de la imágen lo incorpora al archivo de estadísticas del sistema llamado ESTADI .

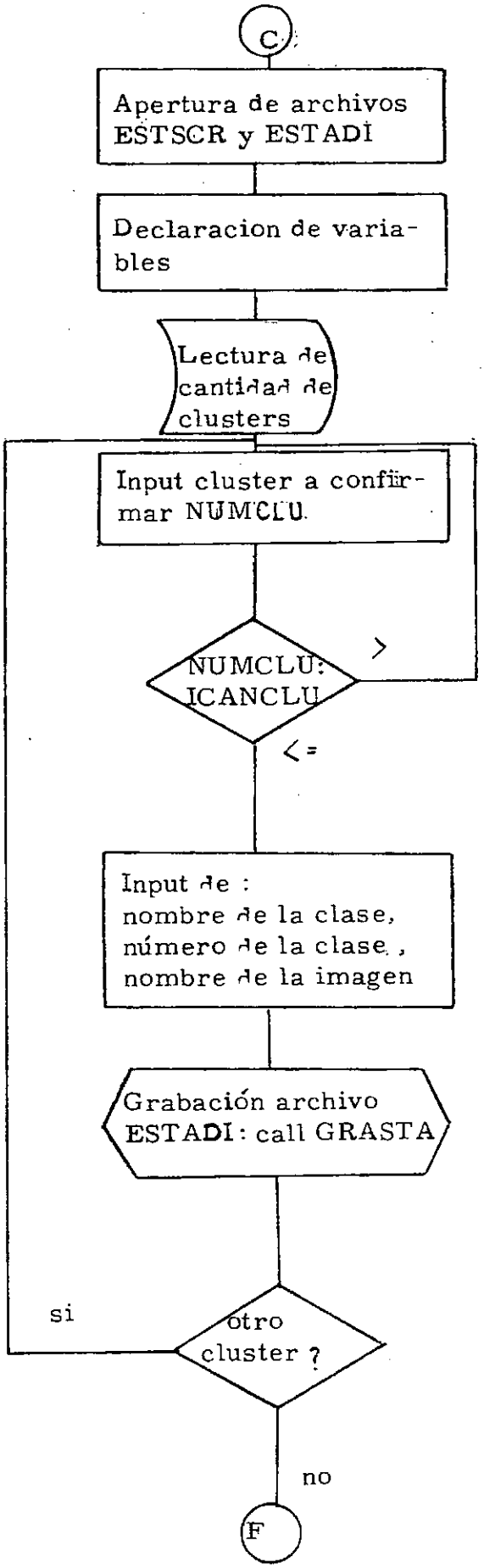
FORMATO DEL REGISTRO DEL ARCHIVO ESTADISTICO

(vector RECSTA)

1er byte	Ult byte	Tipo	Cant. elem.	CONTENIDO
1	20	I*2	10	No usado
21	22	I*2	1	Cantidad de canales
23	44	I*2	11	Identificación de los canales
45	529	R*4	11*11	Matriz de covariancia
530	572	I*2	11	Valores medios

Longitud del registro : 572 bytes

PROGRAMA : EEDCLU



```
      C
      C            PROGRAMA : EEDCLU
      C
0001        COMMON /ESTADI/ RECSTA,NUMREG,KLASE,IMAGE
0002        OPEN(UNIT=7,NAME='DK1:ESTSCR.DAT',TYPE='UNKNOWN',ACCESS='DIRECT',
           *RECORDSIZE=286)
0003        OPEN(UNIT=8,NAME='ESTADI.DAT',TYPE='UNKNOWN',ACCESS='DIRECT',
           *RECORDSIZE=286)
0004        INTEGER*2 RECSTA(286)
0005        INTEGER*2 DATO(1),ERROR
0006        REAL COVAR(11,11)
0007        INTEGER*2 MED(11)
0008        INTEGER*2 NBAND,BAND(11)
0009        INTEGER*4 KLASE(2),IMAGE(2)
0010        DOUBLE PRECISION DPKLAS,DPIMAG
0011        INTEGER*2 SINO,NO,BLANK
0012        EQUIVALENCE (RECSTA(23),COVAR(1,1)),(RECSTA(265),MED(1))
0013        EQUIVALENCE (RECSTA(11),NBAND),(RECSTA(12),BAND(1))
0014        EQUIVALENCE (KLASE(1),DPKLAS),(IMAGE(1),DPIMAG)
0015        DATA NO/'NO'/,BLANK/' '/
0016        READ(7'10)ICANCLU
      C
      C     *** ENTRADA DE DATOS ***
      C
0017  210   WRITE(5,200) ICANCLU
0018  200   FORMAT('/NUMERO DE CLUSTER A CONFIRMAR(1 A',I3,') ?')
0019        READ(5,201) NUMCLU
0020  201   FORMAT(I2)
0021        IF(NUMCLU.GT.ICANCLU) GO TO 210
0023        READ(7'NUMCLU) RECSTA
0024     60 WRITE(5,100)
0025    100 FORMAT(' NOMBRE DE LA CLASE ?'/' CLASS NAME ?')
0026        READ(5,101) DPKLAS
0027    101 FORMAT(A8)
0028      1 WRITE(5,102)
0029    102 FORMAT(' NUMERO DE LA CLASE ?'/' CLASS NUMBER ?')
0030        CALL NUMERO(DATO,1,ERROR)
0031        IF(ERROR.NE.0) GO TO 1
0033        NUMREG=DATO(1)
0034        WRITE(5,103)
0035    103 FORMAT(' NOMBRE DE LA IMAGEN ?'/' IMAGE NAME ?')
0036        READ(5,101) DPIMAG
      C
0037        WRITE(5,165) NUMREG,DPKLAS,DPIMAG
0038    165 FORMAT(' NUMERO = ',I6/
           *       ' CLASE  = ',A8/
           *       ' IMAGEN = ',A8)
0039        WRITE(5,166)
0040    166 FORMAT(' OK ?')
0041        READ(5,167) SINO
0042    167 FORMAT(A2)
0043        IF((SINO.EQ.BLANK).OR.(SINO.EQ.NO)) GO TO 210
      C
      C     *** FIN DE ENTRADA DE DATOS ***
      C
0045        CALL GRASTA(IEND)
0046        WRITE(5,220)
0047  220   FORMAT('/OTRO CLUSTER?')
0048        READ(5,167) SINO
0049        IF((SINO.EQ.NO).OR.(SINO.EQ.BLANK)) GO TO 999
```

```
0051          GO TO 210
0052   999    CONTINUE
0053          END
```

```
FORTRAN IV      STORAGE MAP FOR PROGRAM UNIT .MAIN.

LOCAL VARIABLES, .PSECT $DATA, SIZE = 000020 (    8. WORDS)

NAME   TYPE  OFFSET     NAME   TYPE  OFFSET     NAME   TYPE  OFFSET
BLANK  I*2   000004     ERROR  I*2   000006     ICANCL I*2   000012
IEND   I*2   000016     NO     I*2   000002     NUMCLU I*2   000014
SINO   I*2   000010


COMMON BLOCK /ESTADI/, SIZE = 001116 (  295. WORDS)

NAME   TYPE  OFFSET     NAME   TYPE  OFFSET     NAME   TYPE  OFFSET
RECSTA I*2   000000 EQV NUMREG I*2   001074     KLASE  I*4   001076 EQV
IMAGE  I*4   001106 EQV COVAR  R*4   000054 EQV MED    I*2   001020 EQV
NBAND  I*2   000024 EQV BAND   I*2   000026 EQV DPKLAS R*8   001076 EQV
DPIMAG R*8   001106 EQV

LOCAL AND COMMON ARRAYS:

NAME     TYPE    SECTION OFFSET  -------SIZE----- DIMENSIONS
BAND   I*2       ESTADI  000026  000026 (   11.) (11)
COVAR  R*4 VEC   ESTADI  000054  000744 (  242.) (11,11)
DATO   I*2       $DATA   000000  000002 (    1.) (1)
IMAGE  I*4       ESTADI  001106  000010 (    4.) (2)
KLASE  I*4       ESTADI  001076  000010 (    4.) (2)
MED    I*2       ESTADI  001020  000026 (   11.) (11)
RECSTA I*2       ESTADI  000000  001074 (  286.) (286)

SUBROUTINES, FUNCTIONS, STATEMENT AND PROCESSOR-DEFINED FUNCTIONS:

NAME   TYPE   NAME   TYPE   NAME   TYPE   NAME   TYPE   NAME   TYPE
GRASTA  R*4   NUMERO  I*2
```

PROGRAMA ALBA

INFORME DEFINITIVO

1. Descripción :

1.1 Este programa permite obtener una cinta en formato PI con 1 a 4 canales de salida. Estos son generados de acuerdo a una sucesión de operaciones algebraicas realizadas con los canales de entrada .

1.2 El programa maneja hasta 4 canales de entrada y se les asigna (para la fórmula) los nombres A, B, C, D .

1.3 Cada uno de los canales de salida (llamados w, x, y y z) son generados por fórmulas del tipo :

$$\begin{bmatrix} w \\ x \\ y \\ z \end{bmatrix} = F1 \langle \text{canal de entrada} \rangle \langle \text{op} \rangle F2 \langle \text{canal de entrada} \rangle \left[\langle \text{op} \rangle F3 \langle \text{canal de entrada} \rangle \ldots\ldots F9 \langle \text{canal de entrada} \rangle \right] N$$

Donde :

F1... F9 = coeficientes del tipo : xx. x

$\langle$canal de entrada$\rangle$ = A, B, C, D

$\langle$op$\rangle$ = +, -, *, /.

N = fin de la fórmula

2. Operación

2.1 El programa debe recibir por parte del operador primera línea, última línea, primer pixel, último pixel de la cinta de entrada a procesar .

2.2 A continuación se le debe indicar la cantidad de canales de salida que se desean obtener .

2.3 La cinta de salida numera las líneas y los pixels a partir de 1 .

2.4 El programa informa al operador, por medio de un menú, las características de la operación .

2.5 Al cargar las fórmulas se deben tener en cuenta las siguientes reglas :

a) No se permiten blancos intermedios en la formula .

b) No se permite ningún caracter invalido (resultado imprevisible) .

c) Una fórmula siempre debe terminar con N, (de otra manera cancela la ejecución) .

d) Los coeficientes F1 a F9 deben estar presentes en la formula en forma correlativa .

e) La fórmula es procesada de izquierda a derecha (no existen prioridades entre los operadores) .

3. Subrutinas

ALBA usa las siguientes subrutinas

a) FORMUL

b) IMLINE

c) IMOPEN

d) OUTOPN

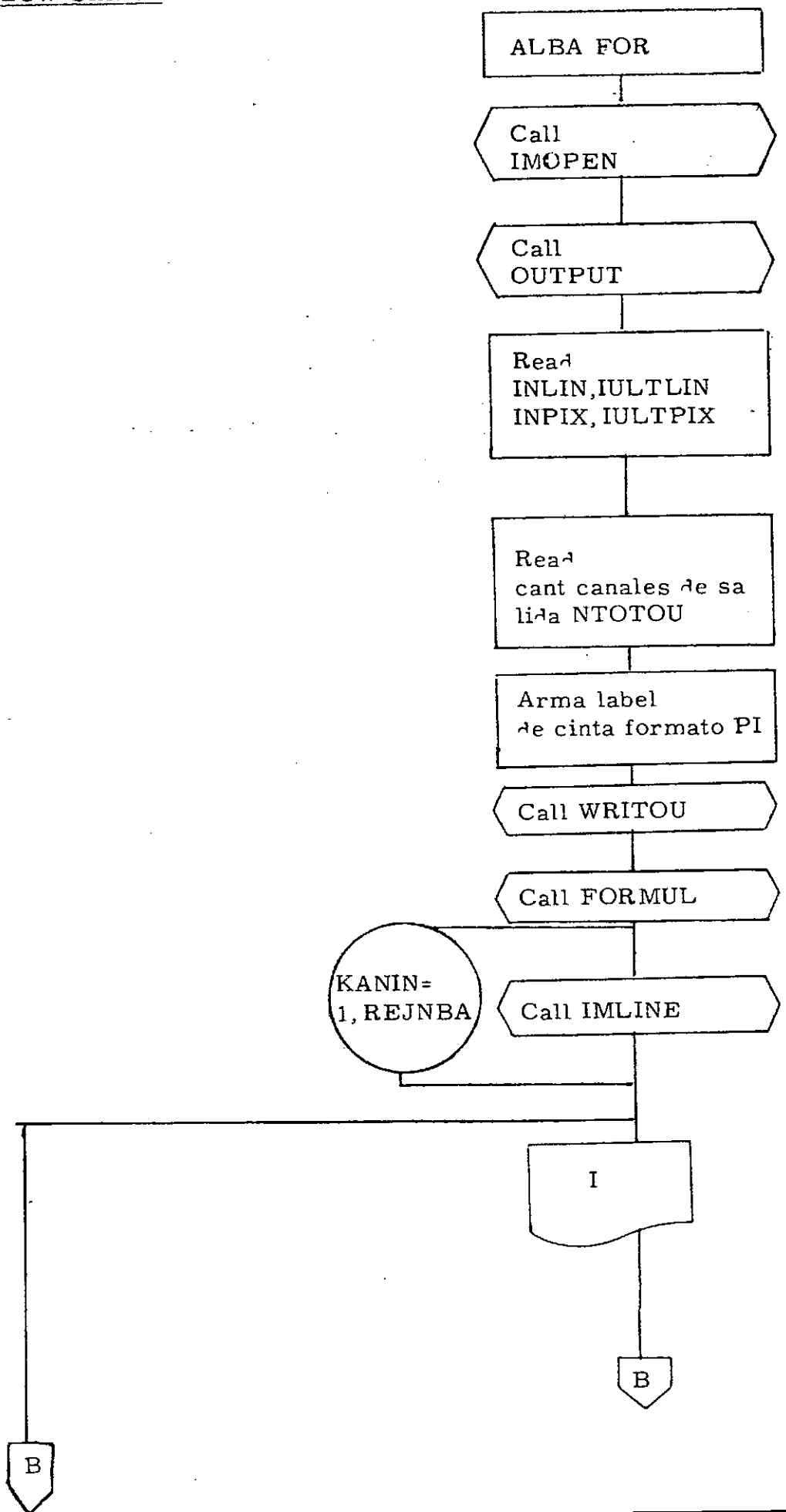
e) WRITOU

La subrutina FORMUL se documenta por separado .

Las demas subrutinas forman parte del paquete de subrutinas de INPUT/OUTPUT del sistema PI .

PROGRAMA ALBA

FLOW CHART



Apertura archivo de entrada

Apertura ar-- chivo de salida

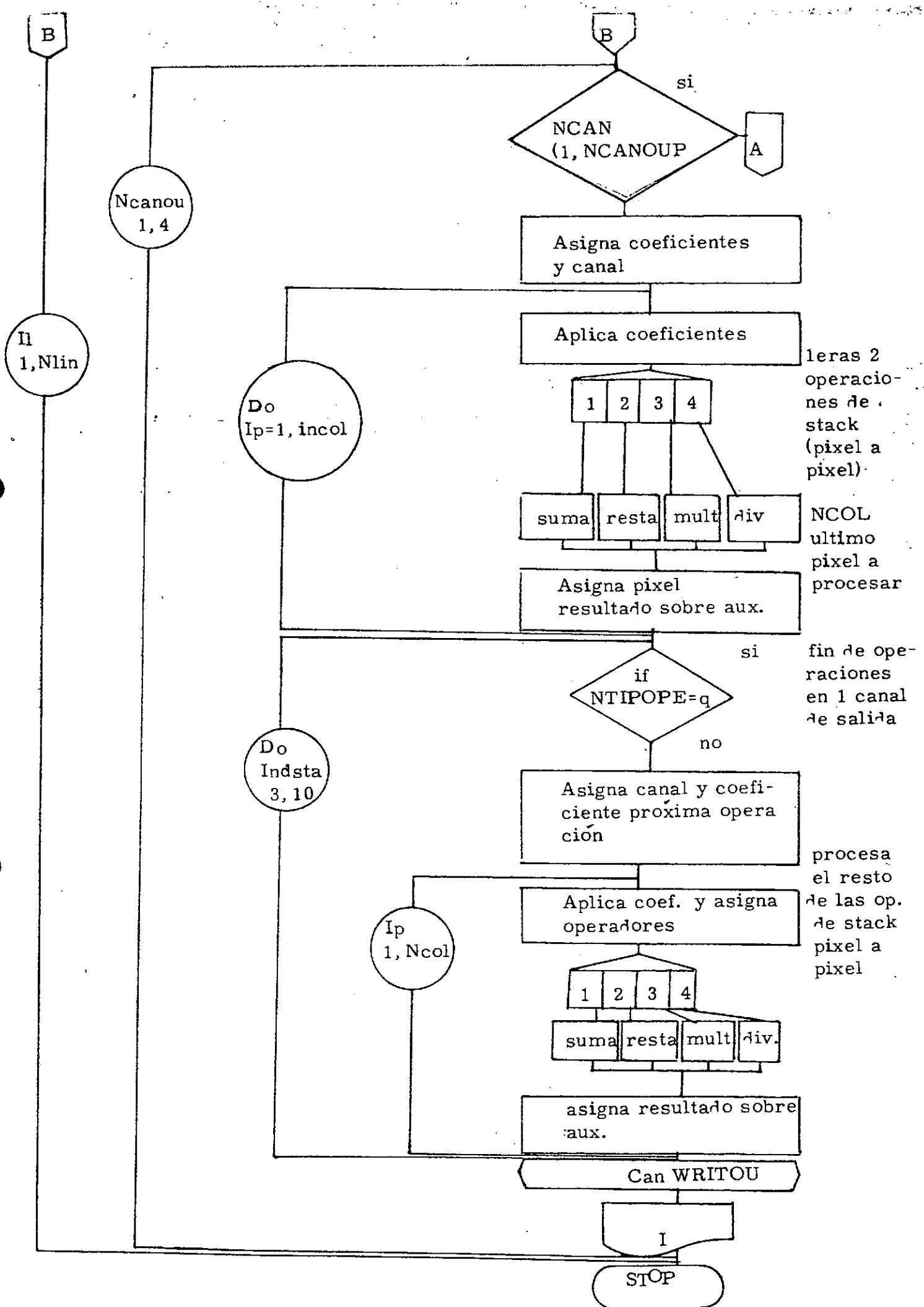
Lee primera linea, última linea, 1er pixel, último pixel

Graba label formato PI

Ver subrutina Formul adjun ta

Lee 1 línea de entrada en todos los cana- les e inicializa newline

Imprime 10 leros pixel en todos los canales p/control


B
B
si
NCAN
(1, NCANOUP
A
Ncanou
1, 4
Asigna coeficientes
y canal
Il
1, Nlin
Aplica coeficientes
1eras 2
operaciones de
stack
(pixel a
pixel)
Do
Ip=1, incol
1
2
3
4
suma
resta
mult
div
NCOL
ultimo
pixel a
procesar
Asigna pixel
resultado sobre aux.
si
fin de ope-
raciones
en 1 canal
de salida
if
NTIPOPE=q
no
Do
Indsta
3, 10
Asigna canal y coefi-
ciente próxima opera
ción
procesa
el resto
de las op.
de stack
pixel a
pixel
Aplica coef. y asigna
operadores
Ip
1, Ncol
1
2
3
4
suma
resta
mult
div.
asigna resultado sobre
aux.
Can WRITOU
I
STOP

```
      C*********************** ALBA.FOR ************************
      C
      C
0001        INTEGER*2 NOPE(10,4),NCAN(10,4)
0002        REAL*4 NOPE1,NOPE2,AUX(1024)
0003        REAL*4 NCOEF(10,4),NPIX
0004        COMMON /STACK/ NOPE,NCAN,NCOEF,NTOTOU
0005        DATA NOPE,NCAN,NCOEF/40*0,40*0,40*.0/
0006        DATA AUX/1024*.0/
0007        INTEGER*2 RESNLI,RESNCO,RESNBA,RESNOB(4)
0008        COMMON RESNLI,RESNCO,RESNBA,RESNOB
0009        INTEGER*2 KANIN,KANOU
0010        INTEGER*2 BUFIN1(1024,4),NEWLINE(1024)
0011        MAX=127
0012        MIN=0
0013        CALL IMOPEN
0014        CALL OUTOPN
0015        WRITE(5,1500)
0016  1500  FORMAT(' PRIMERA LINEA,ULTIMA,PRIMER PIXEL,ULTIMO:'$)
0017        READ(5,*)INLIN,IULTL,INPIX,IULTP
0018        NLIN=IULTL-INLIN+1
0019        NEWLINE(1)=NLIN
0020        NCOL=IULTP-INPIX+1
0021        NEWLINE(2)=NCOL
0022        WRITE(5,1010)
0023  1010  FORMAT(' CANT. DE CANALES CINTA DE SALIDA: '$)
0024        READ(5,*)NTOTOU
0025        NEWLINE(3)=NTOTOU
0026        DO 70 K=1,NTOTOU
0027  70    NEWLINE(K+3)=K
0028        CALL WRITOU(NEWLINE,1024)
0029        CALL FORMUL
0030        DO 20 IL = 1,NLIN
0031        LINPUT=IL+INLIN-1
0032        DO 2000 KANIN=1,RESNBA
0033        CALL IMLINE(NEWLINE,INPIX,INPIX,NCOL,LINPUT,KANIN)
0034        DO 444 I=1,1024
0035        BUFIN1(I,KANIN)=NEWLINE(I)
0036  444   NEWLINE(I)=0
0037  2000  CONTINUE
0038        WRITE(6,556) ((BUFIN1(J,K), J=1,10), K=1,RESNBA)
0039  556   FORMAT( 10I5)
0040        DO 700 NCANOU=1,4
0041        IF (NCAN(1,NCANOU).EQ.0) GO TO 700
0043        NTIPOPE=NOPE(2,NCANOU)
      C
      C************************** PRIM OPERACION ********************
      C
0044        NCAN1=NCAN(1,NCANOU)
0045        NCAN2=NCAN(2,NCANOU)
0046        COEF1=NCOEF(1,NCANOU)
0047        COEF2=NCOEF(2,NCANOU)
0048        DO 800 IP=1,NCOL
0049        IPI=IP+INPIX-1
0050        NOPE1=COEF1*BUFIN1(IPI,NCAN1)
0051        NOPE2=COEF2*BUFIN1(IPI,NCAN2)
0052        GO TO (111,222,333,445),NTIPOPE
0053  111   NPIX=NOPE1+NOPE2
0054        GO TO 666
0055  222   NPIX=NOPE1-NOPE2
```

```
0056          GO TO 666
0057   333    NPIX=NOPE1*NOPE2
0058          GO TO 666
0059   445    NPIX=NOPE1/NOPE2
0060   666    IF(NPIX.GT.MAX) NPIX=MAX
0062          IF(NPIX.LT.MIN) NPIX=MIN
0064          AUX(IP)=NPIX
0065   800    CONTINUE
       C
       C************************* RESTO OPERACIONES ***************
       C
0066          DO 900 INDSTA=3,10
0067          NTIPOPE=NOPE(INDSTA,NCANOU)
0068          IF (NTIPOPE.EQ.9)GO TO 910
0070          NCAN2=NCAN(INDSTA,NCANOU)
0071          COEF2=NCOEF(INDSTA,NCANOU)
0072          DO 810 IP=1,NCOL
0073          IPI=IP+INPIX-1
0074          NOPE2=COEF2*BUFIN1(IPI,NCAN2)
0075          NOPE1=AUX(IP)
0076          GO TO(100,200,300,400),NTIPOPE
0077   100    NPIX=NOPE1+NOPE2
0078          GO TO 600
0079   200    NPIX=NOPE1-NOPE2
0080          GO TO 600
0081   300    NPIX=NOPE1*NOPE2
0082          GO TO 600
0083   400    NPIX=NOPE1/NOPE2
0084   600    IF(NPIX.GT.MAX) NPIX=MAX
0086          IF(NPIX.LT.MIN) NPIX=MIN
0088          AUX(IP)=NPIX
0089   810    CONTINUE
0090   900    CONTINUE
0091   910    WRITE(6,*)
0092          DO 2222 I=1,1024
0093   2222   NEWLINE(I)=AUX(I)
0094          WRITE(6,556) (NEWLINE(I), I=1,10)
0095          WRITE(6,*)
0096          CALL WRITOU (NEWLINE,1024)
0097   700    CONTINUE
0098   20     CONTINUE
0099   500    STOP
0100          END
```

```
FORTRAN IV      STORAGE MAP FOR PROGRAM UNIT .MAIN.

LOCAL VARIABLES, .PSECT $DATA, SIZE = 034160 ( 7224. WORDS)

NAME   TYPE  OFFSET     NAME   TYPE  OFFSET     NAME   TYPE  OFFSET
COEF1  R*4   034106     COEF2  R*4   034112     I      I*2   034072
IL     I*2   034066     INDSTA I*2   034122     INLIN  I*2   034050
INPIX  I*2   034054     IP     I*2   034116     IPI    I*2   034120
IULTL  I*2   034052     IULTP  I*2   034056     J      I*2   034074
K      I*2   034064     KANIN  I*2   034040     KANOU  I*2   034042
LINPUT I*2   034070     MAX    I*2   034044     MIN    I*2   034046
NCANOU I*2   034076     NCAN1  I*2   034102     NCAN2  I*2   034104
NCOL   I*2   034062     NLIN   I*2   034060     NOPE1  R*4   034024
NOPE2  R*4   034030     NPIX   R*4   034034     NTIPOP I*2   034100

COMMON BLOCK /      /, SIZE = 000016 (    7. WORDS)

NAME   TYPE  OFFSET     NAME   TYPE  OFFSET     NAME   TYPE  OFFSET
RESNLI I*2   000000     RESNCO I*2   000002     RESNBA I*2   000004
RESNOB I*2   000006

COMMON BLOCK /STACK /, SIZE = 000502 (  161. WORDS)

NAME   TYPE  OFFSET     NAME   TYPE  OFFSET     NAME   TYPE  OFFSET
NOPE   I*2   000000     NCAN   I*2   000120     NCOEF  R*4   000240
NTOTOU I*2   000500

LOCAL AND COMMON ARRAYS:

NAME      TYPE    SECTION OFFSET  ------SIZE----- DIMENSIONS
AUX     R*4       $DATA   000000  010000 ( 2048.) (1024)
BUFIN1  I*2 VEC   $DATA   010000  020000 ( 4096.) (1024,4)
NCAN    I*2 VEC   STACK   000120  000120 (   40.) (10,4)
NCOEF   R*4 VEC   STACK   000240  000240 (   80.) (10,4)
NEWLIN  I*2       $DATA   030000  004000 ( 1024.) (1024)
NOPE    I*2 VEC   STACK   000000  000120 (   40.) (10,4)
RESNOB  I*2       .$$$$.  000006  000010 (    4.) (4)

SUBROUTINES, FUNCTIONS, STATEMENT AND PROCESSOR-DEFINED FUNCTIONS:

NAME   TYPE   NAME   TYPE   NAME   TYPE   NAME   TYPE   NAME   TYPE
FORMUL  R*4   IMLINE  I*2   IMOPEN  I*2   OUTOPN  R*4   WRITOU  R*4
```

SUBRUTINA FORMUL
INFORME DEFINITIVO

1. Funciones

1.1 Indicar por pantalla de video al operador la forma de cargar la/las fórmula/s con las cuales se van a procesar los canales de salida a partir de los canales de entrada de la imágen original .

1.2 Ingresar 1 fórmula de procesamiento para cada canal de salida .

1.3 Generar un stack de operaciones, que indican al programa principal (ALBA) como procesar cada una de las líneas de salida .

1.4 El stack está compuesto por 3 arreglos de la siguiente forma :

a) NCAN (10, 4) formato :

Contiene los canales de entrada a ser operados entre si y el indice del stack indica el orden de procesamiento . (Códigos válidos : 0, 1, 2, 3, 4) .

b) NOPE (10, 4) formato :

Contiene las operaciones a ser realizadas entre los canales de entrada para obtener el canal de salida . (códigos válidos : 0, 1, 2, 3, 4, 9)

0=no operar, 1=+, 2=-, 3=* . 4=1 , 9= fin operaciones .

c) NCOEF (10, 4) formato : real * 4

Contiene los coeficientes que deben afectar a cada canal de entrada antes de ver operador . Formato válido : xx. x

En todos los casos el segundo subindice (4) indica sobre cual de los canales de salida se debe operar .

NCAN, NOPE y NCOEF son procesados en conjunto para obtener una línea de salida . Por ejemplo :

La fórmula w = F1A + F1CN.En donde :

w= canal 1 de salida

4= canal 1 de entrada

C= canal 3 de entrada

N= fin de fórmula

y suponiendo que : coef F1= 1.2

coef F2= 2.5

generaría un stack de la siguiente forma

NCOEF(10,4)

índice stack	w	x	y	z
1	1.2	0	0	0
2	2.5	0	0	0
3	0	0	0	0
4	0	0	0	0
10	0	0	0	0

NCAN (10,4)

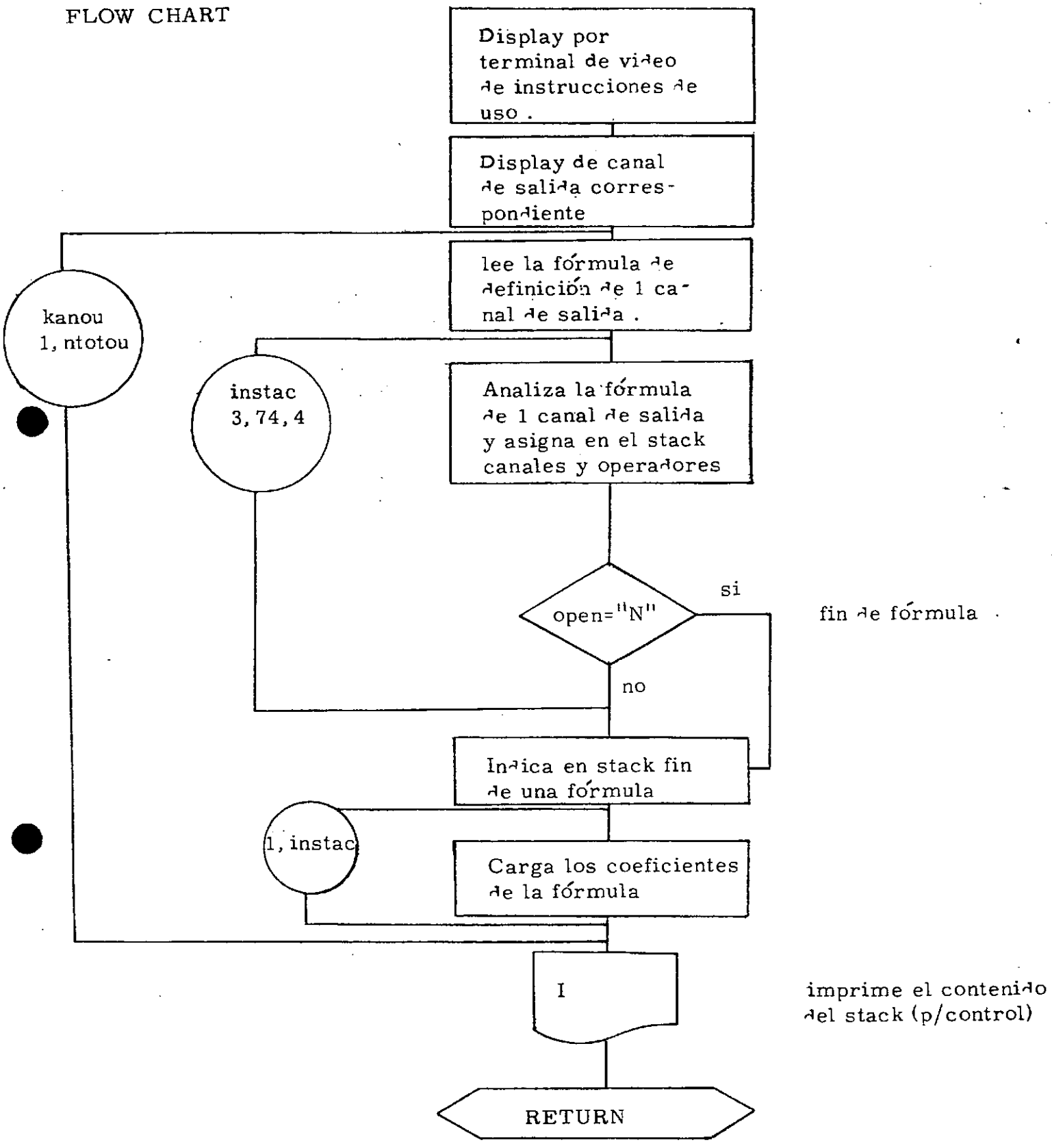
	w	x	y	z
1	1			
2	2			
3				
4				
10				

NOPE (10,4)

	w	x	y	z
1	0	0	0	0
2	1	0	0	0
3	9	0	0	0
4	0	0	0	0
10	0	0	0	0

SUBRUTINA FORMUL

FLOW CHART



```
      C****************** SUBRUTINA FORMUL.FOR ******************
      C
0001        SUBROUTINE FORMUL
0002        INTEGER*2 NOPE(10,4),NCAN(10,4),NTOTOU
0003        LOGICAL*1 STRING(74)
0004        INTEGER*2 NADA(9)
0005        DATA NADA/'F1','F2','F3','F4','F5','F6','F7','F8','F9'/
0006        REAL*4 NCOEF(10,4)
0007        COMMON/STACK/NOPE,NCAN,NCOEF,NTOTOU
0008        LOGICAL*1 CANAL,OPER,LETRA
0009        DO 111 I=1,5
0010  111   WRITE(5,*)
0011        TYPE 1050
0012  1050  FORMAT(' *** INSTRUCCIONES DE USO ***'//)
0013        TYPE 1051
0014  1051  FORMAT(' 1.LOS CANALES 1,2,3,Y 4 DE INPUT SE INDICAN EN LA
           1  FORMULA COMO: A,B,C Y D.'//)
0015        TYPE 1052
0016  1052  FORMAT(' 2.LOS OPERADORES VALIDOS SON: + , - , * , / , N .
           1  EL OPERADOR N INDICA FIN DE LA FORMULA.'//)
0017        TYPE 1053
0018  1053  FORMAT(' 3.LOS COEFICIENTES VALIDOS EN LA FORMULA SON:
           1  F1,F2,.....,F9 (EN ESE ORDEN).'//)
0019        TYPE 1054
0020  1054  FORMAT(' 4.EN LA FORMULA NO SE PERMITEN BLANCOS INTERMEDIOS.'//)
0021        TYPE 1055
0022  1055  FORMAT(' 5.EJEMPLO: W = F1A+F2B/F3AN '///)
0023        DO 150 KANOU=1,NTOTOU
0024        IF (KANOU.EQ.1) LETRA='W'
0026        IF (KANOU.EQ.2) LETRA='X'
0028        IF (KANOU.EQ.3) LETRA='Y'
0030        IF (KANOU.EQ.4) LETRA='Z'
0032        WRITE(5,1030) LETRA
0033  1030  FORMAT(' 'A,' = '$)
0034        READ(5,1040) STRING
0035  1040  FORMAT(74A)
0036        INSTAC=0
0037        DO 850 I=3,74,4
0038        INSTAC=INSTAC+1
0039        CANAL=STRING(I)
0040        OPER=STRING(I+1)
0041        IF(CANAL.EQ.'A') NCAIN=1
0043        IF(CANAL.EQ.'B') NCAIN=2
0045        IF(CANAL.EQ.'C') NCAIN=3
0047        IF(CANAL.EQ.'D') NCAIN=4
0049        IF(CANAL.EQ.'N') GO TO 860
0051        NCAN(INSTAC,KANOU)=NCAIN
0052        IF(OPER.EQ.'+') NOPER=1
0054        IF(OPER.EQ.'-') NOPER=2
0056        IF(OPER.EQ.'*') NOPER=3
0058        IF(OPER.EQ.'/') NOPER=4
0060        IF(OPER.EQ.'N') GO TO 860
0062        NOPE(INSTAC+1,KANOU)=NOPER
0063  850   CONTINUE
0064  860   NOPE(INSTAC+1,KANOU)=9
0065        WRITE(5,*)
0066        WRITE(5,1140)
0067  1140  FORMAT(' ** CARGA DE LOS COEFICIENTES F<N> **')
0068        DO 160 I=1,INSTAC
0069        WRITE(5,1070) NADA(I)
```

```
1070  FORMAT(' 'A2,': '$)
      READ(5,1080) NCOEF(I,KANOU)
1080  FORMAT(F4.1)
160   CONTINUE
      WRITE(5,*)
150   CONTINUE
      WRITE(6,1090) ((NOPE(J,K), K=1,4), J=1,10)
1090  FORMAT(4I4)
      WRITE(6,*)
      WRITE(6,1090) ((NCAN(J,K), K=1,4), J=1,10)
      WRITE(6,*)
      WRITE(6,1100) ((NCOEF(J,K), K=1,4), J=1,10)
1100  FORMAT(4F4.1)
      RETURN
      END
```

```
FORTRAN IV      STORAGE MAP FOR PROGRAM UNIT FORMUL

LOCAL VARIABLES, .PSECT $DATA, SIZE = 000204 (    66. WORDS)

NAME    TYPE  OFFSET      NAME    TYPE  OFFSET      NAME    TYPE  OFFSET
CANAL   L*1   000152      I       I*2   000156      INSTAC  I*2   000162
J       I*2   000170      K       I*2   000172      KANOU   I*2   000160
LETRA   L*1   000154      NCAIN   I*2   000164      NOPER   I*2   000166
OPER    L*1   000153


COMMON BLOCK /STACK /, SIZE = 000502 (   161. WORDS)

NAME    TYPE  OFFSET      NAME    TYPE  OFFSET      NAME    TYPE  OFFSET
NOPE    I*2   000000      NCAN    I*2   000120      NCOEF   R*4   000240
NTOTOU  I*2   000500

LOCAL AND COMMON ARRAYS:

NAME      TYPE    SECTION OFFSET  ------SIZE------ DIMENSIONS
NADA    I*2       $DATA   000112  000022 (     9.) (9)
NCAN    I*2 VEC   STACK   000120  000120 (    40.) (10,4)
NCOEF   R*4 VEC   STACK   000240  000240 (    80.) (10,4)
NOPE    I*2 VEC   STACK   000000  000120 (    40.) (10,4)
STRING  L*1       $DATA   000000  000112 (    37.) (74)
```

Especificaciones del equipo de computación

PDP 11/15 Modelo 20

24 K palabras de memoria (palabra de 2 bytes)

2 Unidades de cinta de 800 bpi

2 Unidades de discos removibles de 2.5 Mb cada uno

1 CRT modelo VT 100

1 Impresora de 130 caracteres por minuto

1 Monitor color RAMTEK de 256 líneas por 512 columnas

Sistema operativo RT11

Lenguaje Fortran IV Plus