N O T I C E

THIS DOCUMENT HAS BEEN REPRODUCED FROM MICROFICHE. ALTHOUGH IT IS RECOGNIZED THAT CERTAIN PORTIONS ARE ILLEGIBLE, IT IS BEING RELEASED IN THE INTEREST OF MAKING AVAILABLE AS MUCH INFORMATION AS POSSIBLE

JSC 12656

PROGRAM DOCUMENTATION

8.0-10178.

NASA CR-
160647

FINAL DESIGN SPECIFICATION
FOR
DOT DATA BASE UPDATE DECK CONVERSION PROGRAM (DOTDEC)
CPD 722
Job Order 61-127

Prepared By

Lockheed Electronics Company, Inc.

Systems and Services Division

Houston, Texas

Contract NAS 9-15200

FOR

EARTH OBSERVATION DIVISION

SPACE AND LIFE SCIENCES DIRECTORATE

National Aeronautics and Space Administration

LYNDON B. JOHNSON SPACE CENTER
Houston, Texas

July 1977

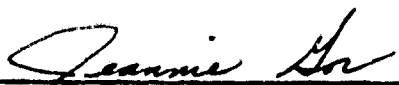LEC 10969
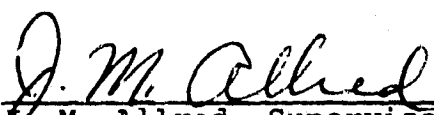
PROGRAM DOCUMENTATION

FINAL DESIGN SPECIFICATION
FOR
DOT DATA BASE UPDATE DECK CONVERSION PROGRAM (DOTDEC)
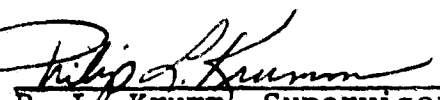
JOB ORDER 81-127

Prepared By

_____
Jeannie Gor

Approved By

_____   _____
J. M. Allred, Supervisor   P. L. Krumm, Supervisor
Physical Sciences Section   Software Development Section

_____
W. J. Reicks, Manager
Applied Mechanics Department

i

CONTENTS

1. SCOPE

This specification establishes the design of a computer program to convert the LACIE Procedure 1 Dot Data Base Update Deck to an EOD-LARSYS "field definition" deck.

The requirements specification for the program was provided by the Research, Test, and Evaluation (RT&E) Branch of the Earth Observations Division of NASA/JSC.

2. APPLICABLE DOCUMENTS

The following documents form a part of this specification to the extent specified herein.

- "Coding for Procedure 1 Data," provided by S. Yao/LEC

- User Procedures EOD-LARSYS, LEC-3984

- Task Agreement 77-2, Job Order 81-127

3. SYSTEM DESCRIPTION

3.1 HARDWARE DESCRIPTION

Not applicable.

3.2 SOFTWARE DESCRIPTION

The purpose of the program is to convert LACIE Procedure 1 Dot
Data Base Update deck(s) into EOD-LARSYS compatible "field
definition" decks. The program is coded in the IBM 360 Fortran
IV Language for operation under the Cambridge Monitor System (CMS)
on the IBM 360/67 at Purdue University.

3.2.1 SOFTWARE CONPOMENT NO. 1 (DOTDEC)

DOTDEC is an independent program. The function of DOTDEC is to
read cards punched in the format for updating the dot data base
file on the PDP 11/45 in JSC building 17, and to output (punch)
"field definition" cards compatible with input requirements of
the Univac 1108 EOD-LARSYS program. For example, the punched
card(s) output by DOTDEC may be used for dot input to the DOT
LABEL processor or the DOT DATA processor of EOD-LARSYS.

3.2.1.1 Linkages

DOTDEC does not require any subprograms, and is not referenced
by any other program.

3.2.1.2 Interfaces

The program is accessed via the LARS/Purdue terminal in JSC
building 17. The interface between the program and the user is
the LARS/Purdue IBM 360/67 Control Program (CP) and an associated
operating system, the Cambridge Monitor System (CMS).

The card reader/punch adjacent to the terminal in building 17
will be the program's primary input/output interface.

3.2.1.3 Inputs

DOTDEC requires as input the LACIE Procedure 1 Dot Data Base Update deck(s) in the following format with specified information:

Column	Description
1-4	Segment number
5-10	Blank
11-12	Line grid coordinate (from 01-11)
13	Blank
14-15	Column grid coordinate (from 01-19)
16*	Boundary pixel decision
17*	Ground truth label
18	Blank
19-23	Analyst information (not needed by DOTDEC)
24	Dot type ("S" or blank)
25-80	Additional information not utilized by DOTDEC

Only the following columns from the input cards are used by DOTDEC for output of the field definition deck(s):

- Columns 1-4 for the sample segment number

- Columns 11-12 for the line (= line grid coordinate *10)

- Columns 14-15 for the pixel (= column grid coordinate *10)

- Column 17 for a field name to be output on the field definition card

- Column 24 for dot type ("S" in column 24 represents type 1 and anything else will represent type 2)

*Column 17 is considered as the identification of the dot or the most likely identification, in the case of a boundary decision.

Column 16 is the comparison boundary identification.

Column 16	Column 17	
blank	W	"Wheat," single pixel
Ø	W	"Wheat" with "other" boundary confusion
B	W	"Wheat" with "barley" boundary confusion
R	A	"Oats" with "rye" boundary confusion
Ø	B	"Barley" with "other" boundary confusion
Ø	Ø	"Other" with "other" boundary confursion
blank	F	"flax"
blank	R	"Rye"
blank	Ø	"blank" if unidentified

3.2.1.4 Outputs

DOTDEC outputs (punch) field definition deck(s) in the following formats:

CC1	CC11
COMMENT	SAMPLE SEGMENT NO. XXXX
TYPE	1
W 1003	(1,1), (30,100)
W 1701	(1,1), (10,170)
TYPE	2
R 1507	(1,1), (70,150)
B 1511	(1,1), (110,150)
O 1709	(1,1), (90,170)

Each field definition card defines the sample, line coordinates of a dot. The first character of the field name is the identification of the dot read from column 17 of the input Dot Data Base Update Card. The next two digits in the field name are the line grid coordinate taken from columns 11-12 and the last two digits in the field name are the column grid coordinates taken from columns 14-15. The line and pixel points on the output field

definition card will be the result of line = line grid coordinate *10 and pixel = column grid coordinate *10. "COMMENT" cards with the sample segment numbers separates the "field definition" card decks for each of the sample segments and a "TYPE" card will precede the "field definitions" of the set of dots within a given type.

A full printer listing of the output field definition cards is also available. Also, the printer output will contain any error messages generated from the input of incorrectly formatted Dot Data Base update coordinates.

3.2.1.5 Storage Requirements

The program requires 43980 bytes (10995 (32-bit)words) of storage.

3.2.1.6 Description

DOTDEC reads up to 1000 input Dot Data Base update cards at a time and places the pertinent information into BUFF(1000,6). The information extracted and saved from the input cards is:

1. Sample segment number (card columns 1-4)

2. Line grid coordinate (card columns 11-12)

3. Pixel grid coordinate (card columns 14-15)

4. Boundary pixel ID (card columns 16)

5. Ground truth label (card column 17)

6. Dot type (card column 24)

DOTDEC then groups the input information by sample segment number. All dots within a given sample segment are ordered by ascending line number, and ascending pixels for a given line number. Following the ordering of the dots within a given sample segment, DOTDEC performs a grouping by dot type — Type 1 dots or Type 2 dots — depending on the presence of an "S" or

not an "S" respectively, in card column 24 of the input cards. After the grouping by sample segment, ordering by line and pixel, and sub-grouping by dot type, DOTDEC creates the punched card output in the following manner:

1. Each group of dots under a given sample segment number has an initial card punched containing the sample segment number. The format of this card is in the EOD-LARSYS "COMMENT" card format (see section 3.2.1.4).

2. Each sub-group of dots of a given type is preceded by a card punched in the format of an EOD-LARSYS (Procedure 1) "TYPE" card, containing the dot type "1" or "2" (see section 3.2.1.4).

3. Following the "TYPE" card, each dot of the given type has a card punched in the format of an EOD-LARSYS "field definition" card (see section 3.2.1.4). The dot ID in columns 1-6 of the card consists of the set of characters extracted from card column 17 (ground truth label), columns 11-12 (line grid coordinate, and columns 14-15 , (pixel grid coordinate). The "field definition" for the dot contains "(1,1)" for sample, line increment, and the sample, line coordinate of the dot in standard EOD-LARSYS "field definition" card format.

DOTDEC continues to read input cards and punch output cards, in sets of up to 1000 each, until an END-OF-FILE is detected.

DOTDEC also provides an optional printer listing of the input cards, the output cards, and diagnostic messages pertaining to any errors detected in the input cards.

3.2.1.7 Flow Chart

Not applicable.

3.2.1.8 Listing

See Appendix A.

4. OPERATION

DOTDEC is executed on the LARS/PURDUE IBM 360/67 computer, using the remote terminal facilities in JSC building 17. Program operation is described in terms of the terminal's operations necessary for execution from building 17.

4.1 USER DOCUMENTATION

Program DOTDEC is located on a permanent disc file which is referenced by an ID number and password provided by the Research, Test, and Evaluation Branch (RT&E). The program can be called up via the Hazeltine 2000 terminal or from the 2741 typewritter terminal by typing in 'DOTDEC.'

The following capabilities are provided to the user after the data deck is input via the card reader and DOTDEC has been executed on that data by typing in 'DOTDEC':

1. A listing of the input data cards

2. A listing of the punched output decks along with any error messages pertaining to the input data deck.

3. The punched cards output by DOTDEC via the card punch, formatted for input to the UNIVAC 1108 EOD-LARSYS program.

4.1.1 PROGRAM SET-UP AND EXECUTION

The input deck (LACIE Procedure 1 Dot Data Base Update deck) must be preceded by a LARS/PURDUE system 'ID' card. All inputs from the terminal are placed on an input buffer, so the 'ID' card identifies the correct inputs with the correct terminal user.

'ID' card format is:

CC1	CC10	
ID	JSC200	(RT&E account ID at LARS/PURDUE)

Basic order of activities for program execution are:

1. Terminal LOGIN and acquire temporary disc file for program execution

2. Transmit input card deck to Purdue

3. Execute DOTDEC

4. Initiate printing of output on line printer

5. Initiate punching of punched output

6. Retrieve printout and punched cards

7. Log out on terminal

8. Interpret the punched cards (on 026 keypunch machine)

Notes:

1. In the sequence of terminal commands below the user should note that the user typed inputs are only done after the system responds with a ('>') caret sign that indicates that the system is ready for the next instruction. Premature input of instructions without confirmation from the system that it is ready may produce surprising results.

   Messages shown in quotes (" ") indicate expected system response.

2. The [depress CR] instructions below simply mean to depress the carriage return key.

   ● On the Hazeltine 2000 terminal, carriage return is the 'CR' key.

   ● On the 2741 terminal, carriage return is the 'return' key.

4-2
10

3. To erase errors:

- On the Hazeltine 2000 terminal:

  '@' key is depressed to erase a typed-in character,
  '[' key is depressed to erase an entire line.

- On the 2741 terminal:

  '@' key is depressed to erase a typed-in character,
  '¢' ('cent' character) key is depressed to erase an
  entire line.

4. If operating on Hazeltine 2000 terminal, switch terminal
   connect box [green box adjacent to CRT screen] to 'LARS'.

4.1.1.1 Set-Up

<u>Terminal Activity</u>	<u>Description</u>
[DEPRESS 'CR']	
'RESTART'	System will either return with
>	'RESTART' and/or (>) sign. In
	either case log in with:
	'L JSC200'
>L JSC200	
[DEPRESS 'CR']	Terminal returns with
"ENTER PASWWORD="	"Enter password"
*******	Type password in over mask
[DEPRESS 'CR']	characters
"ENTER NAME=>"	Type in initials or your name
[DEPRESS 'CR']	
"YOUR OPERATORS ARE ....."	System returns with various
"READY AT ...."	messages
"CP"	When (>) sign returns next
	type in 'I CMS'
>I CMS	
[DEPRESS 'CR']	System returns
"CMS READY"	Type in - 'DISX SET S'

Terminal Activity	Description
>DISX SET S [DEPRESS 'CR']	This statement asks for temporary disc space for your execution. The 'S' option asks for small disc which is sufficient for most purposes and is the easiest amount of space to obtain. If you feel you will need more space, you can use 'M' in place of the 'S' which asks for medium space or just leave no option which asks for large amount of space. The latter request is the most difficult to obtain. After making the request for temporary disc, wait for system to come back with:

"LINE END CHARACTER SET TO
"YOU ARE LINKED TO TEMP DISK ____
"P(_); _ FILES; _ REC IN USE, LEFT (OF __) __% full (__ CYL)"

	% full indicates how much of the disk is filled.
	If you feel there is not enough empty space left on the disc for your purposes type in:
>LIST F [DEPRESS 'CR']	Which will give a list of file names and their types that are on that disc. Then erase as many files from the disc as you want by typing:
>ERASE FFFF TTTT [DEPRESS 'CR']	Where FFFF is the file name and TTTT is it's type.

Terminal Activity	Description

"R; T=___"

After the terminal returns, continue the above procedure until you feel that there is enough storage disc space for your needs.

Next read in data deck.

4.1.1.2 Reading in the Input Data Deck on Card Reader

A deck of cards can be read when there is no other activity on the card reader terminal, reading, printing or punching. When there is no activity:

1. Remove blank cards from card READ/PUNCH hopper.

2. Clear any cards that may still be in station by depressing the 'NPRD' key.

3. Place an ID card in the front of the input deck (letters ID in columns 1-2 and JSC200 in columns 10-15).

4. Place deck in card hopper. (Cards should be face down with top edge of cards facing you.)

5. Place card weight on top of deck.

6. Turn the mode switch dial to 'TSM TRSP.'

7. Depress 'End-of-File' key. ('EOF' key)

8. Depress 'START' key, and keep it depressed until 'READY' light is on.

9. After all cards have been read, the card reader terminal will commence beeping.

10. Depress 'NPRO' key again to get out any cards that may be still in station.

11. Turn mode switch to 'OFF-LINE.'

12. Take card deck out of card bin.

4.1.1.3 Processing Data Deck

Terminal Activity	Description
[DEPRESS 'CR']	After returning to terminal wait a few seconds and depress 'CR'
	Wait for system to return following messages:
"**CARDS XFERED BY HOUSTON **" "CMS"	
>O READ DOTDEC DATA [DEPRESS 'CR']	Type in 'O READ DOTDEC DATA'
"R; >=_____" >DOTDEC	System returns Type in 'DOTDEC'
[DEPRESS 'CR'] "16.10.32 FILEDEF 5 DSK-P1 DOTDEC DATA"	System returns
"16.10.34 FILEDEF 6 DSK-P1 PRINT LISTING"	
"16.10.36 FILEDEF 7 DSK-P1 PUNCH OUTPUT"	
"16.10.37 LOAD DOTDEC (XEQ)"	
"EXECUTION BEGINS ..."	Execution of DOTDEC begins
"R; T=_____"	DOTDEC is completed.

4.1.1.4 Printing and Punching the Output

An offline printer copy of output is available and it is advisable to obtain since any errors that may have occurred will only show up on the offline listing.

To get an offline listing, wait until there is no other activity on the line printer or terminal.

Terminal Activity	Description

>O PRINT PRINT LISTING
[DEPRESS 'CR']

Type in 'O PRINT PRINT LISTING'

"R; T=____"

System returns

1. The LARS' terminal will commence an audible beeping; turn mode system knob on station to 'PRINT'.

2. Depress 'START' key on the printer control panel.

3. When the printing is completed, the beeping will commence again; turn mode system knob to "OFF LINE'.

4. Depress 'CARRIAGE STOP' on printer control panel; then 'CARRIAGE RESTORE' to eject paper from printer.

5. Tear off listing.

To obtain an off-line listing of the data deck follow the same procedure as above except type:

>O PRINT DOTDEC DATA
[DEPRESS 'CR']

'O PRINT DOTDEC DATA'

instead of >O PRINT PRINT LISTING.

To obtain punched card output, again wait until the terminal is free of activity, then type:

>O PUNCH PUNCH OUTPUT
[DEPRESS 'CR']

'O PUNCH PUNCH OUTPUT'

1. Go over to terminal and place stack of blank cards into hopper face down with top edge of cards facing you .

2. Turn mode system knob to "PUNCH".

3. If beeping sound has not come on yet, wait until you hear it.

4. Depress 'START' button and keep it depressed until 'READY' light is on.

5. Punching is completed when beeping sound comes on again.

6. Remove extra blank cards from hopper.

7. Depress 'NPRO' button to get out any excess cards that might still be in system.

8. Turn mode system knob to 'OFF-Line'

9. Take punched cards out of bin.

4.1.1.5 Sign off LARS' system by typing in:

>CP LOGOUT
 [DEPRESS 'CR']

An example can be found on the following page.

Example:

```
L JSC200
ENTER PASSWORD:
██████◆
ENTER NAME:>GOR
YOUR OPERATORS THIS AFTERNOON ARE  DEANIE  AND  DOUG.
EXT POSSIBLE SHUTDOWN IS 0100-0800 WED.
READY AT 14.35.42 ON 06/21/77
CP
>I CMS
CMS (VER 3.2) READY!

>DISK SET S
LINE END CHARACTER SET TO \
YOU ARE LINKED TO TEMP DISK 21
P (192): 6 FILES! 176 REC IN USE, 120 LEFT (OF 296), 59% FULL (2 CYL)
! T=2.01/3.06 14.36.30

>
◆◆ CARDS  XFERED BY HOUSTON ◆◆
CMS
>O READ DOTDEC DATA
R! T=0.03/0.11 14.46.40

>DOTDEC
14.46.45 FILEDEF 5 DSK-P1 DOTDEC DATA
14.46.47 FILEDEF 6 DSK-P1 PRINT LISTING
14.46.48 FILEDEF 7 DSK-P1 PUNCH OUTPUT
14.46.50 LOAD DOTDEC (XEQ)
EXECUTION BEGINS...
R! T=0.64/0.98 14.46.54

>O PRINT PRINT LISTING
R! T=0.12/0.31 14.48.26

>O PUNCH PUNCH OUTPUT
R! T=0.06/0.16 14.50.16

>CP LOGOUT
CONNECT= 00:16:01  VIRTCPU= 000:02.88  TOTCPU= 000:05.03
OGOUT AT 14.51.39 ON 06/21/77
W~67 ONLINE
```

5. TEST PROCEDURE

5.1 DESCRIPTION OF TEST

The test run on this program was done with the data listed on the
following page. Data consisted of two invalid coordinates to
test error checking. Six different sample segments
were input at random. Also the coordinates were not input in
ascending line number and ascending pixel per line, to test the
program's ordering technique.

The printer listing of the punch output shows data grouped
according to sample segments and type and then ordered in
ascending line number and ascending pixel in each line.

... SEGMENT NUMBER 7654

TYPE
1,1,,250, 7...

SAMPLE SEGMENT NUMBER 7654

TYPE
?1,15,, 10, 50n
?,1,5,, 10, 60n
1,15,,170, 100n
1,15,,20, 110n

TYPE
1,15,, 30, 40n
1,15,, 50, 100n

SAMPLE SEGMENT NUMBER 1645

TYPE
?1,15,,150, 200

TYPE
?1,15,,100,1000

SAMPLE SEGMENT NUMBER 1856

TYPE
1,15,? 70, 200
?1,15,? 90,1000

TYPE
?1,15,? 40, 60n

SAMPLE SEGMENT NUMBER 1234

TYPE
1,15,, 65, 40n
,1,15,, 30, 70n

TYPE
?1,15,, 40, 20n

... SEGMENT NUMBER 1556

TYPE
1,15,, 90, 40n
?,1,15,, 20, 50n

TEST VERIFICATION

For: Dot Data Base Update Deck Conversion

This verification is being conducted to insure that the delivered program products satisfy the requirements as originally stated by the requesting organization.

R. P. Huydon
_____
NASA Monitor

Thomas C. Minter
_____
Requestor

Jeannie Gor
_____
Developer

P.J. Austin Jr
_____
Cognizant System Manager

_____ 7-5-77
Quality Assurance

Jeannie Gor
_____
Test Conductor

Verification Date: ___7-5-77___

21

APPENDIX A

DOTDEC SOURCE PROGRAM LISTING

```
C     ...READ CARDS FROM PROCEDURE 1 OR DATA BASE
C     ...THE ORDERS FROM THE DEFAULT LARSYS PROCEDURE 1
C     ...
C     ...FIELDS WITH THE SAME SEGMENT NUMBERS ARE GROUPED AND SUCH
C     ...GROUPS ARE INDICATED BY THE PRESENCE OF A LARSYS CONTROL
C     ...CARD.  THE TYPE CARD IS PUNCHED AT THE START OF A GROUP
C     ...OF FIELDS WITH THAT SAME TYPE UNTIL A TYPE CHANGE IS MADE WITHIN
C     ...THAT SEGMENT.

      IMPLICIT INTEGER (A-N,P-Z)
      DATA TYP1/4040404040/, TYP2/6040404040/, NP/7404040404/
      DATA L/10/, LFORES/100/
      DATA END5/04/, END/62/
      DIMENSION SEG(1000,6) , TYP1(250,4), TYP2(250,4)
      DIMENSION ORDER(1000)
      DATA CHG1/'01','02','03','04','05','06','07','08',
     *     '10','11','12','13','14','15','16','17','18','19'/
      LOGICAL EOF
      EOF=.FALSE.
      L = 0
C 100 READ(1,1000,END=100) SEG,COUNT,LINE,PIXEL,PD,CLASS,ANINFO,TYPE,REST
1000  FORMAT(2X,I2,A1,A6,A1,A30)
C
C     CHECK FOR ERRORS IN INPUT CARD FORMATS
C
      IF(LINE.LT.1.OR.LINE.GT.111)GO TO 60
      IF(PIXEL.LT.1.OR.PIXEL.GT.191) GO TO 70
C*
      DO 1 C = 1, 1
      SEG(L,1) = SEGMENT
      SEG(L,2) = LINE
      SEG(L,3) = PIXEL
      SEG(L,4) = PD
      SEG(L,5) = CLASS
      SEG(L,6) = TYPE
      IF(SEG(L,1).1000) GO TO 10
      GO TO 120
C
60    WRITE(6,1100)
1100  FORMAT(5X,'ERROR - LINE COORDINATE IS OUT OF RANGE.')
      GO TO 60
70    WRITE(6,1200)
1200  FORMAT(5X,'ERROR - COLUMN COORDINATE IS OUT OF RANGE.')
      WRITE(6,1300)CARD IT ERROR IS -,A2,2X,I2,1X,I2,A1,A5,A1,A30)
      GO TO 10
C
100   EOF=.TRUE.
120   SEG(L,1)
      SEG = SEG(L,1)
      IF(SEG.GT.200) GO TO 195
      WRITE(6,1300)
C
      FORMAT(5X,'COORDS',6X,'SAMPLE SEGMENT NUMBER ',A6)
      WRITE(6,'SAMPLE SEGMENT NUMBER ',A6)
C
C     COLLECT COORDS WITH SAME SEGMENT NUMBERS.
C     ...WILL CONTAIN INDEXES TO BUFF ARRAY THAT HAVE SAME SEG.
C
      DO 125 J = 1, L
      IF(SEG(J).NE.SEG(J,1)) GO TO 125
      SEG(J) = J
      SEG(J,1)
125   CONTINUE
      IF(SEG(J,1) .GO TO 124
```

```
FILE DATDEC

0052        ORD = J
0053        ORDER(ORD) = SCR(SINC)
0054        GO TO 136

C*
C*         ORDER LINES AND PIXELS FROM LEFT-RIGHT AND TOP-BOTTOM
124        ORD = 0
123        DO 135 J = 1,SINC
           IF (SCR(J).EQ.-999) GO TO 133
           IJ = SCR(J)
           IK = BUFF(IJ,2)
           JTMP = BUFF(IJ,3)
           DO 126 JJ=1,SINC
           IF (SCR(JJ).EQ.-999) GO TO 126
           IK = SCR(JJ)
           IF(IK.LE.BUFF(IK,2)) GO TO 127
           IF(IK.LT.BUFF(IK,2)) GO TO 126
           LTMP = BUFF(IK,2)
           JTMP = BUFF(IK,3)
127        JJ=JJ
           IF(JTMP.LT.BUFF(IK,3)) GO TO 126
           LTMP = BUFF(IK,3)
126        JJ = JJ
128        CONTINUE
130        CONTINUE

C*
C*         ORDER ARRAY WILL CONTAIN INDEXES OF BUFF ARRAY OF ALL ELEMENTS
C*         WITH SAME SEG. NUMBER AND WILL BE ORDERED LEFT-RIGHT,TOP-BOTTOM
           ORD = ORD + 1
           ORDER(ORD) = SCR(HIM)
           SCR(HIM) = -999
133        CONTINUE
135        CONTINUE
           DO 137 P=1,SINC
           IF(SCR(P).NE.-999) GO TO 123
137        CONTINUE
136        CONTINUE
           JK = 1

C*         ORDER SEGMENT INTO TYPE 1 AND 2
136        CONTINUE
           INC1 = 0
           INC2 = 0
           DO 160 K=OK,ORD
           KK = ORDER(K)
           IF(BUFF(KK,6).EQ.5) GO TO 140
           INC2 = INC2 + 1
           TYP2(INC2,1) = BUFF(KK,2)
           TYP2(INC2,2) = BUFF(KK,3)
           TYP2(INC2,3) = BUFF(KK,4)
           TYP2(INC2,4) = BUFF(KK,5)
           GO TO 160
140        INC1 = INC1 + 1
           TYP1(INC1,1) = BUFF(KK,2)
           TYP1(INC1,2) = BUFF(KK,3)
           TYP1(INC1,3) = BUFF(KK,4)
           TYP1(INC1,4) = BUFF(KK,5)
160        CONTINUE
           IF(K.EQ.OK)JK=1
           IF(INC1.LT.1) GO TO 150
           IF(INC1.GT.250.OR.INC2.GT.250) JK=K
           IF(INC1.GT.250.OR.INC2.GT.250) GO TO 165
150        CONTINUE
           IF(INC1.LT.1) GO TO 180
165        WRITE(6,1700)ORD
1700       FORMAT(/5X,'TYPE',6X,A1)
           PUNCH 1800,ORD
1800       FORMAT(' TYPE',16X,A1)
           DO 170 I = 1,INC1
           LTH = TYP1(I,1)=LNMULT
           PX=TYP1(I,2)=PXMULT
           X = TYP1(I,1)
           LTH = CHART(X)
           X = TYP1(I,2)
           PIX = CHART(X)
```

A-2
28

```
         WRITE(3,1000) TYP1(I,4),LIN,PIX,LP,RP,CLP,PX,LN,RP
1000     FORMAT( ... ) TYP1(I,4),LIN,PIX,LP,RP,CLP,PX,LN,RP
         FORMAT(PX,A1,IX,A2,A2,4X,A1,1.1.,A1,A2, I3 ,., I3   ;A1)
2000     FORMAT( A1,IX,A2,A2,4X,A1,1.1.,A1,A2, I3 ,., I3   ;A1)
176      CONTINUE
         I=(I+(2.LT.1) GO TO 195
         WRITE(6,1700) TY1
         CONTINUE
         DO 100 I = 1, L162
         LX = TYP2(I,3)=DEFAULT
         PX = TYP2(I,2)=PXDFLT
         X1 = TYP2(I,3)
         L1 = CHAR=T(X)
         PIX = CHAR=T(X)
         WRITE(6,2000) TYP2(I,4),LIN,PIX,LP,RP,CLP,PX,LN,RP
196      WRITE 2000,
155      CONTINUE
         WRITE(.SAG,1)=-999
         IF(60.GT-1) GO TO 128
260      GO TO 180
         IF(I.GT2.GGF) GO TO TE 5
         STOP
         END
```