

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

"Made available under NASA sponsorship
in the interest of early and wide dis-
semination of Earth Resources Survey
Program information and without liability
for any use made thereof."

DRge
8.0-10202

JSC-14554

NASA CR
160598

"AS-BUILT" SPECIFICATION FOR CCIT7
PROCESSOR PROGRAM

Job Orders 73-783 and 71-475

(TIRF 78-0022)

(E80-10209) AS-BUILT SPECIFICATION FOR
CCIT7 PROCESSOR PROGRAM (Lockheed
Electronics Co.) 64 p HC A04/MF A01

N80-23788

CSCL 05B

G3/43

Unclass
00209

Prepared By

Lockheed Electronics Company, Inc.
Systems and Services Division
Houston, Texas

Contract NAS 9-15200

For

EARTH OBSERVATIONS DIVISION
SPACE AND LIFE SCIENCES DIRECTORATE



National Aeronautics and Space Administration

LYNDON B. JOHNSON SPACE CENTER

Houston, Texas

November 1978



LEC-12518

"AS-BUILT" SPECIFICATION FOR CCIT7
PROCESSOR PROGRAM

Job Orders 73-783 and 71-473
(TIRF 78-0022)

PREPARED BY

W. P. White

APPROVED BY

for James A. Wilkinson
P. L. Krumm, Acting Supervisor
Scientific Applications Software Section

E. M. Hsu
E. M. Hsu, Supervisor
Accuracy Assessment Section

Prepared By

Lockheed Electronics Company, Inc.

For

Earth Observations Division

Space and Life Sciences Directorate

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION
LYNDON B. JOHNSON SPACE CENTER
HOUSTON, TEXAS

November 1978

1. Report No. JSC-14554	2. Government Accession No.	3. Recipient's Catalog No.	
4. Title and Subtitle "As-Built" Specification for CCIT7 Processor Program		5. Report Date November 1978	6. Performing Organization Code
		7. Author(s) W. P. White Lockheed Electronics Company, Inc.	8. Performing Organization Report No. LEC-12518
9. Performing Organization Name and Address Lockheed Electronics Company, Inc. 1830 NASA Road 1 Houston, Texas 77058		10. Work Unit No.	11. Contract or Grant No. NAS 9-15200
		13. Type of Report and Period Covered Technical Report	
12. Sponsoring Agency Name and Address National Aeronautics and Space Administration Lyndon B. Johnson Space Center Houston, Texas 77058 Technical Monitor: J. D. Erickson		14. Sponsoring Agency Code	
		15. Supplementary Notes	
16. Abstract The program CCIT7 is a utility module of the Accuracy Assessment Software System of the Large Area Crop Inventory Experiment. This program accesses data originating on the Classification and Mensuration Subsystem/Crop Assessment Subsystem interface tapes (CCIT's) of the Earth Resources Interactive Processing System (version 7). The data items needed for subsequent Accuracy Assessment processing are written into three disk files. The data extracted consist of a summary of the classification stratified areal estimate, cluster-dot match, and analyst-labeled dots (types 1 and 2).			
17. Key Words (Suggested by Author(s)) Accuracy Assessment CCIT		18. Distribution Statement	
19. Security Classif. (of this report) Unclassified	20. Security Classif. (of this page) Unclassified	21. No. of Pages 65	22. Price*

*For sale by the National Technical Information Service, Springfield, Virginia 22161

JSC Form 1424 (Rev Nov 75)

NASA — JSC

PRECEDING PAGE BLANK NOT FILMED

CONTENTS

Section	Page
ABBREVIATIONS.	vii
1. SCOPE.	1-1
2. APPLICABLE DOCUMENTS	2-1
3. SYSTEM DESCRIPTION	3-1
3.1 <u>HARDWARE DESCRIPTION</u>	3-1
3.2 <u>MODULE DESCRIPTION</u>	3-1
3.3 <u>SOFTWARE DESCRIPTION</u>	3-6
3.3.1 MODULE CCIT7	3-7
3.3.2 SUBROUTINE READH	3-13
3.3.3 SUBROUTINE READRC.	3-17
3.3.4 SUBROUTINE HEADER.	3-20
3.3.5 SUBROUTINE BIASC7.	3-24
3.3.6 SUBROUTINE CLUST7.	3-28
3.3.7 SUBROUTINE RITE7	3-32
3.3.8 SUBROUTINE TURNON.	3-35
3.3.9 SUBROUTINE DOTS7	3-38
3.3.10 SUBROUTINE STCOD7.	3-44
3.3.11 SUBROUTINE PRINT7.	3-47
4. OPERATIONS	4-1
4.1 <u>OPERATORS GUIDE</u>	4-1
4.1.1 HARDWARE CONFIGURATION	4-1
4.1.2 PROGRAM EXECUTION.	4-1

Section	Page
4.2 <u>USERS GUIDE</u>	4-2
4.3 <u>MAINTENANCE DOCUMENTATION</u>	4-2
APPENDIX — FORMAT FOR .CLO FILE.	A-1

TABLES

Table	Page
1 TASK-BUILDER COMMAND FILE FOR CCIT7 PROCESSOR PROGRAM	3-6
2 BATCH RUN DECK SETUP.	4-3

FIGURES

Figure		Page
1	Data flow of the CCIT7 processor program.	3-2
2	Functional flow of the CCIT7 processor program.	3-3
3	Flow diagram for the CCIT7 processor program.	3-9
4	Listing for the CCIT7 processor program	3-11
5	Flow diagram for subroutine READH	3-15
6	Listing for subroutine READH.	3-16
7	Flow diagram for subroutine READRC.	3-18
8	Listing for subroutine READRC	3-19
9	Flow diagram for subroutine HEADER.	3-22
10	Listing for subroutine HEADER	3-23
11	Flow diagram for subroutine BIASC7.	3-25
12	Listing for subroutine BIASC7	3-26
13	Flow diagram for subroutine CLUST7.	3-30
14	Listing for subroutine CLUST7	3-31
15	Flow diagram for subroutine RITE7	3-33
16	Listing for subroutine RITE7.	3-34
17	Flow diagram for subroutine TURNON.	3-36
18	Listing for subroutine TURNON	3-37
19	Flow diagram for subroutine DOTS7	3-40
20	Listing for subroutine DOTS7.	3-42
21	Flow diagram for subroutine STCOD7.	3-45
22	Listing for subroutine STCOD7	3-46
23	Flow diagram for subroutine PRINT7.	3-48
24	Listing for subroutine PRINT7	3-49

ABBREVIATIONS

AA	Accuracy Assessment
CAMS	Classification and Mensuration Subsystem
CAS	Crop Assessment Subsystem
CCIT	CAMS/CAS interface tape
DEC	Digital Equipment Corporation
DPR	Data processing request
DTL	Data Techniques Laboratory
DTRM	Data terminal
EOD	Earth Observations Division
ERIPS	Earth Resources Interactive Processing System
LACIE	Large Area Crop Inventory Experiment
Pixel	Picture element
SAE	Stratified areal estimate
TIRF	Transmittal Information Request Form
UIC	User identification code

1. SCOPE

This document specifies the detailed design for a software module called CCIT7, which will manipulate and extract data from Accuracy Assessment (AA) data base files. These files are derived from Large Area Crop Inventory Experiment (LACIE), version 7, Classification and Mensuration Subsystem/Crop Assessment Subsystem (CAMS/CAS) interface tapes (CCIT's). The data extracted are output into three new data base files for direct input to AA analytical programs.

2. APPLICABLE DOCUMENTS

The following documents, of the exact issue shown, form parts of the specification to the extent specified herein.

- a. "As-Built" Design Specification for PDP 11/45 Accuracy Assessment System Using Disk Data File. JSC-13893 (LEC-11881), February 1978 (and references therein).
- b. Implementation of CCIT6A Processor Program. Transmittal Information Request Form (TIRF) 78-0022, May 11, 1978.
- c. CAM/CAS Interface Tape Interface Control Document. LACIE-C00708, revision A (JSC-09866), July 1976.
- d. Classification and Mensuration Subsystem (CAMS) Requirements. LACIE-C00200, volume II, revision D (JSC-11330), August 1977.
- e. "As-Built" Design Specification for CCIT6A Processor Program. JSC-14368 (LEC-12303), August 1978.

3. SYSTEM DESCRIPTION

The CCIT7 processor module accomplishes the data manipulations shown in figure 1. Basically, the CCIT data for a particular segment number, SSSS, and classification date, YYDDD, contained in file SSSSYDDDD.CCO are processed to obtain three output files required as input to existing or planned AA programs. The SSSSYDDDD.CLO file contains data needed for future programs. The SSSSYDDDD.AI1 and SSSSYDDDD.AI2 files are required for input to existing modules SPATL and MLTCRP.

3.1 HARDWARE DESCRIPTION

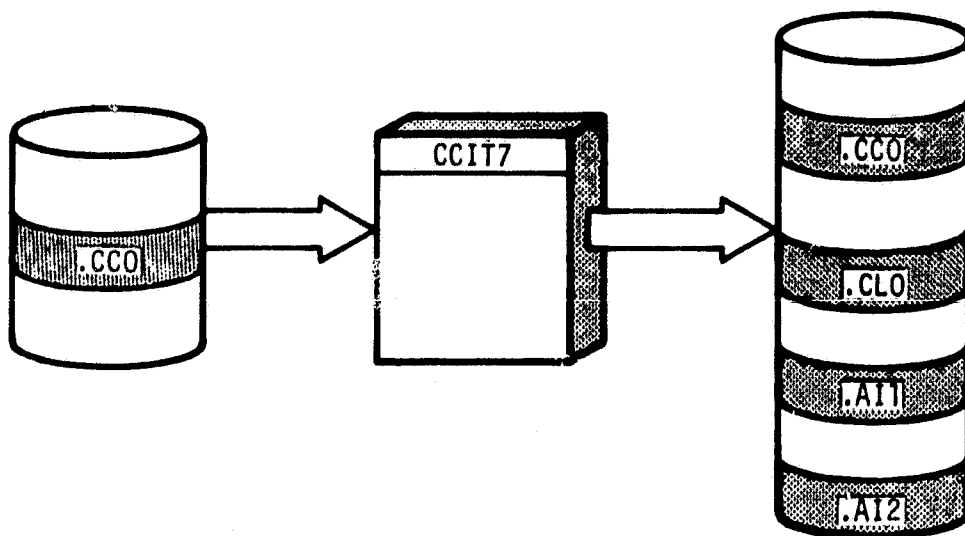
The PDP 11/45, with the following peripherals, is required.

- a. Card reader or user terminal
- b. Line printer
- c. Two disk units

3.2 MODULE DESCRIPTION

The CCIT7 module is implemented on the PDP 11/45 for time-sharing or background processing of CCIT data files into three output data files: an unformatted file of character data and two formatted files of analyst-labeled dots. See the functional flow diagram (fig. 2). The CCIT7 module represents a major revision of the CCIT6A module described in reference e (section 2).

The LACIE CCIT is a universal nonimaging tape containing extensive statistical and ancillary data for a series of Earth Resources Interactive Processing System (ERIPS) runs. Using the AA CCIT program, all data for a relevant segment are transferred to a Files-11 disk file named SSSSYDDDD.CCO,..., where SSSS is the segment number, YY is the year, and DDD is the day of the year. This .CCO file contains three 80-byte header records and a large number (>20) of 720-byte data records.



ORIGINAL PAGE IS
OF POOR QUALITY

Figure 1.— Data flow of the CCIT7 processor program.

~~3-2~~
4

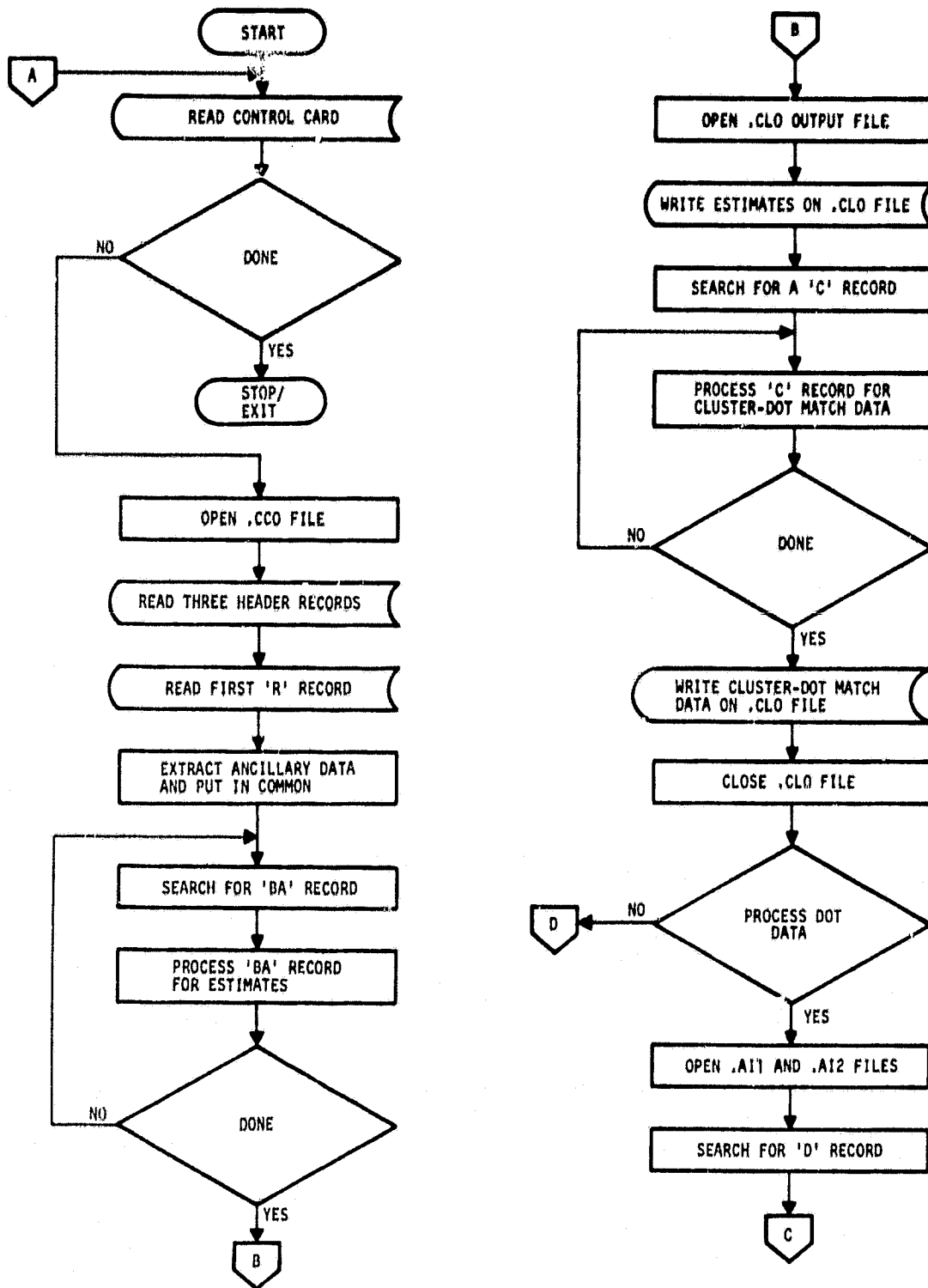


Figure 2.— Functional flow of the CCIT7 processor program.

3-3
5

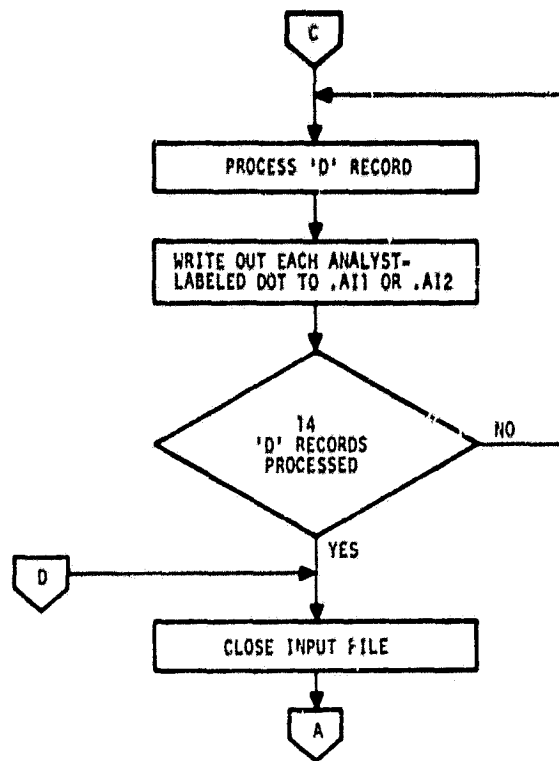


Figure 2.— Concluded.

3-4
6

The first step of the process is to read the name of the input .CCO file and open this file for reading. Then the three CCIT header records are read and ignored. The next record (720 bytes) is read, checked to verify that it is a recognition (R) record, and processed to extract the data processing request (DPR) number and acquisition dates used in the classification.

Next a 'BA' record is searched for and processed. The program extracts classification data on class picture element (pixel) populations, the ERIPS estimate, the stratified areal estimate (SAE), and the variance for each class. An output file SSSSYDDDD.CLO (where SSSS is the segment number and YYYY is the classification date derived from the DPR number) is opened. The first record of the .CLO file contains an integer word giving the number of crop classes for the classification. The estimate data are written as the second record. Sixteen bytes are required for each class; a LACIE version 7 CCIT can contain data on up to 26 classes. A line printer report is output after the disk files are written.

In the next step a 'C' record is searched for and processed. The total number of clusters (Q) and cluster-dot match data are extracted as Q-groups of 12 characters. (Generally, there are more than 15 clusters, so some of these data appear in additional 'C' records.) When all the cluster-dot data are assembled into a buffer, the number of clusters is written out as the third record of file SSSSYDDDD.CLO, and the match data are written as the second record of this file. A line printer report is then output, and the SSSSYDDDD.CLO file is closed.

Then the program tests to determine if analyst-labeled dot output files are required; this is the default condition. If this condition exists, the output files SSSSYDDDD.AI1 and SSSSYDDDD.AI2 are opened, and a search is made for the first dot record. There are 14 dot records listing all 209 dots. The program examines each dot to determine if it has been labeled by the analyst. If so, it is written out (line, sample, and label) to the proper file, depending on the dot type (1 or 2). The first dot record in

each output file also contains ancillary information on the segment (number and state code), classification date, acquisition dates, data terminal (DTRM) tape number, and type of label.

When all dots have been processed, the SSSSYDDD.AI1 and SSSSYDDD.AI2 output files and the input file are closed. No report is output detailing the .AI file dot data because these formatted files are easily examined using the Digital Equipment Corporation (DEC) PDP-11 utility program PIP.

3.3 SOFTWARE DESCRIPTION

The CCIT7 processor program consists of 11 user-supplied routines: CCIT7 (main program), READH, READRC, HEADER, BIASC7, CLUST7, RITE7, TURNON, DOTS7, STCOD7, and PRINT7. The program makes use of a card-image-formatted file, CCIT7.DAT, for program control and the line printer and user disk for output. The following sections provide a detailed description of each of the 11 routines. The recommended task-build command file (CCIT7.CMD), used to create the load module (CCIT7.TSK), is given in table 1.

TABLE 1.-- TASK-BUILDER COMMAND FILE FOR CCIT7 PROCESSOR PROGRAM

```
CCIT7,LP:/SH=CCIT7,READH,READRC,HEADER,BIASC7,CLUST7,  
RITE7,TURNON,DOTS7,STCOD7,PRINT7  
/  
FMTBUF=132  
UNITS=6  
ACTFIL=6  
ASG=SY:1  
ASG=SY:2  
ASG=SY:3  
ASG=SY:5  
ASG=LP:6  
PRI=50  
//
```

For simplicity, the definition of arrays carried in COMMON blocks, the definition of COMMON blocks, and the description of COMMON blocks are not repeated for each routine. Instead, each of these elements is described in the

routine of origin. Reference to the Interface subsections and to the compiler listings of each routine provides sufficient information to follow the data flow throughout the program.

3.3.1 MODULE CCIT7

3.3.1.1 Linkage

The CCIT7 program is the main program. It calls user subroutines READH, READRC, HEADER, BIASC7, CLUST7, DOTS7, and PRINT7. Subroutines BIASC7, DOTS7, and PRINT7 are called using multiple entry points.

3.3.1.2 Interface

Most communication with the user routines is handled via COMMON blocks. A single integer parameter is passed on call to READH, which indicates the number of CCIT header records to be read. A single integer parameter Q is passed to CLUST7 as a flag.

3.3.1.2.1 COMMON Block BUF

BUF contains a 720-byte array, A, which is used to hold one CCIT logical record for processing.

3.3.1.2.2 COMMON Block FNAME

FNAME contains a 24-byte array, FILNAM, and an integer variable, SKIP. FILNAM contains the input file name read from CCIT7.DAT. The value of SKIP determines whether the dot records are to be processed. If SKIP is nonzero, the dots are not processed.

3.3.1.2.3 COMMON Block B7

B7 contains the number of categories, NCAT, for the classification.

3.3.1.2.4 COMMON Block CLUSTR

CLUSTR contains a 60-byte by 12-byte array, CNAME, and an integer variable CNUM. CLUSTR provides an interface between subroutine CLUST7 and subroutine RITE7.

3.3.1.3 Input

The CCIT7 program receives all input CCIT data via subroutine READRC. Control data are provided directly by reading card images from file CCIT7.DAT.

3.3.1.4 Output

The CCIT7 program provides all output via subroutines PRINT7, RITE7, BIASC7, DOTS7, and CLUST7.

3.3.1.5 Storage

The CCIT7 program requires 1078 words of storage.

3.3.1.6 Description

The CCIT7 routine provides the control function for the program. Flow is controlled via tests on the first bytes (descriptive characters) of each logical record in the CCIT input file and by counts based upon the required number of records of a given type.

3.3.1.7 Flow Chart

The flow chart for CCIT7 is given in figure 3.

3.3.1.8 Listing

The listing for this subroutine is given in figure 4.

~~3.8~~
10

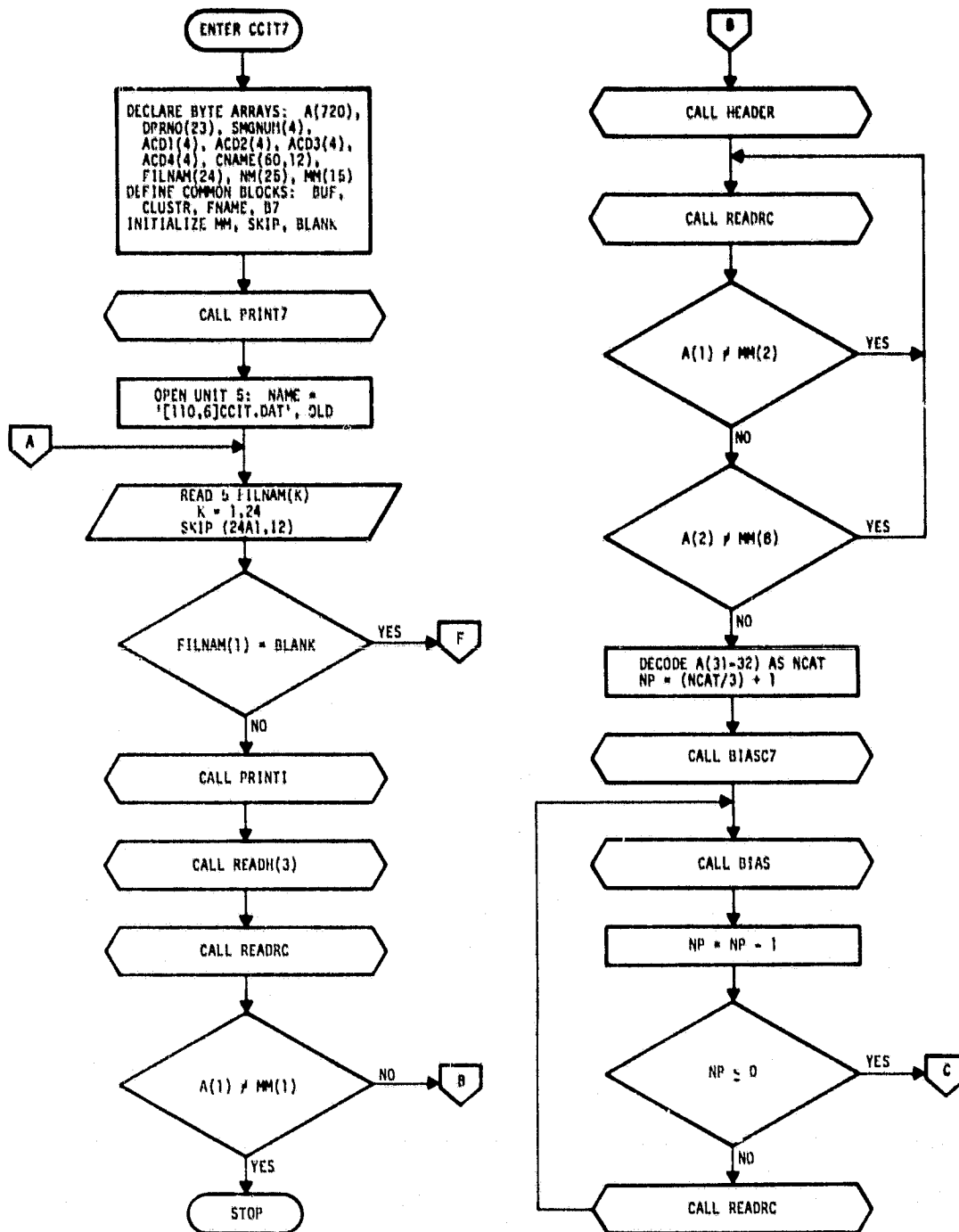


Figure 3.— Flow diagram for the CCIT7 processor program.

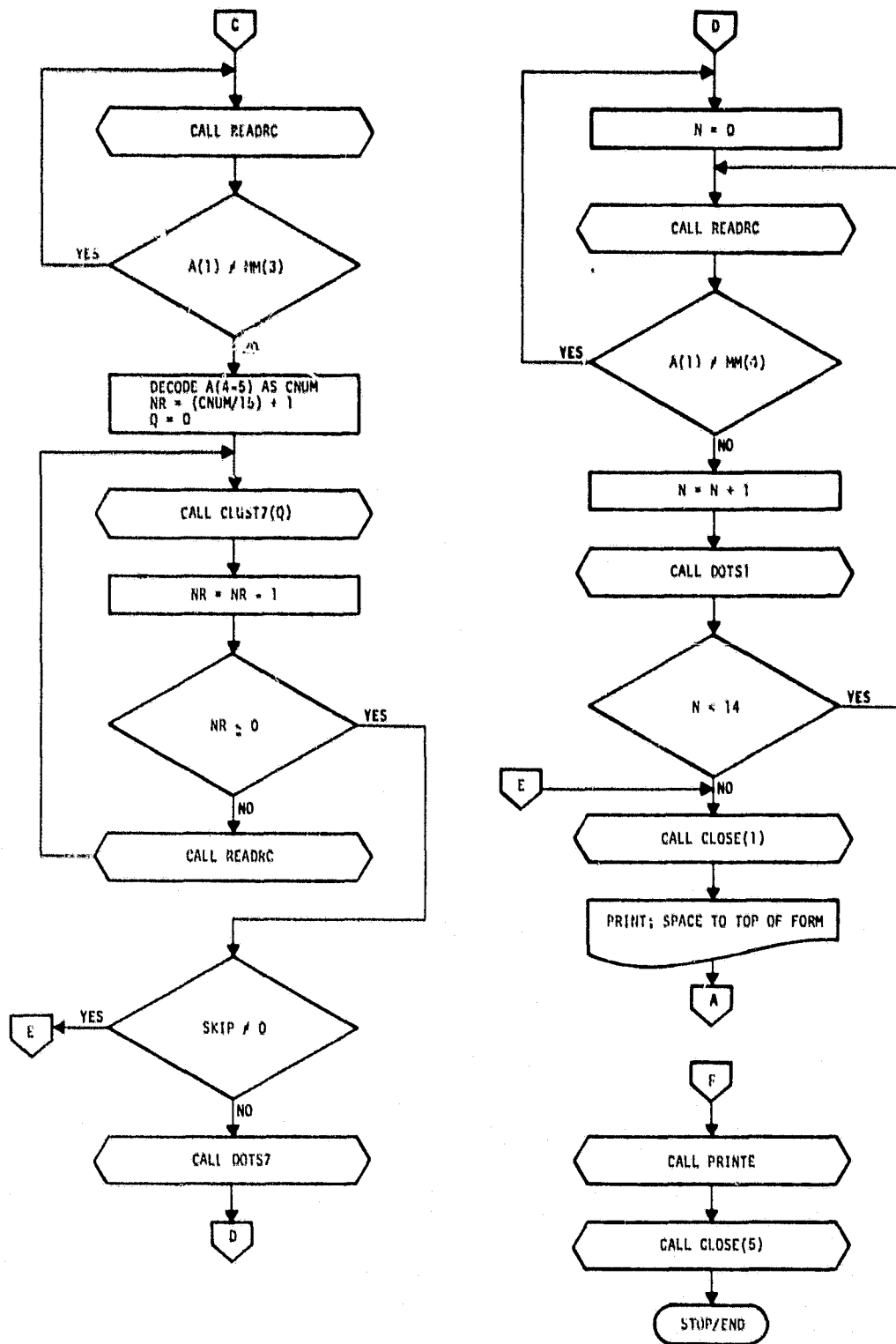


Figure 3.— Concluded.

3-10
12

```

0001 PROGRAM CCIT7
0002 IMPLICIT INTEGER(A-Z)
0003 SAVE A(20),DPNR(20),SMNR(4),ACR(4),ICD(4),WDR(4),
1 CNAME(60,12),FILNAM(24),NP(20),PMIS),BLANK
C.....
C MAIN OF CCIT7 PROCESSOR - READ A CARD IMAGE FROM EISB,CCIT7.DAT
C WHICH GIVES THE FILE NAME OF THE LACIES AS CCIT7 FILE ON DISC
C IN THE FORM I(31),SSSYDD,CGO
C A(20) IS THE BUFFER WHICH CONTAINS ONE CCIT7 DATA RECORD
C.....
0004 COMMON/BUFFA
0005 COMMON/B7/NCAT
0006 COMMON/CLUSTY/CRUM,CRUM
0007 COMMON/FNAME/FILNAM,SKIP
0008 DATA HZ/1MR,1MG,1HC,1HD,1D,1E,1H,1MG,70IN /
0009 DATA SKIP/0/
0010 DATA BLANK/1H /
0011 CALL PRINT7
C
0012 READ TRUY FILNAME AND SKIP FACTOR
OPEN UNIT=9,NAME='(110,6)CCIT7.CAT',TYPE='OLD',READONLY,
1 ACCESS='SEQUENTIAL',FORM='FATTEN')
0013 READ(9,10)(FILNAM(I),I=1,24),SKIP
0014 IF(FILNAM(1).EQ.BL.NK) GO TO 200
0015 CALL PRINT1
C
0016 READ 3 HEADER RECORDS OF 80 BYTES EACH
CALL READ(13)
C
0017 READ THE FIRST 'M' RECORD AND PROCESS FOR DPR NUMBER ACQUISITION
C
0018 DATES ETC
CALL READRC
IF(A(1).NE.NH(1)) STOP
0019 CALL HEADER
READ AND PROCESS BIAS CORRECTION SUMMARY RECORD
C
0020 AS FIRST RECORD OF FILE SSSSYDD,CLO
0021 CALL READRC
0022 IF(A(1).NE.NH(2)) GO TO 210
0023 IF(A(2).NE.NH(3)) GO TO 210
0024 DECODE(2,100,A(3))%CAT
NP = (NCAT/3) + 1
0025 CALL BIASCY
0026 CALL BIAS
0027 NP = NP + 1
0028 IF(NP.LE.0) GO TO 220
0029 CALL READRC
0030 GO TO 245
C
0031 READ FIRST CLUSTER RECORD
C WHEN DONE SUBROUTINE 'RITE7' IS CALLED WHICH WRITES THE NUMBER
C OF CLUSTERS AND THE CLUSTER-BY-MATCH DATA ROW AS YOU RECORDS
C OF FILE SSSSYDD,LD
0032 CALL READRC
0033 IF(A(1).NE.NH(3)) GO TO 220
0034 DECODE(2,100,A(3))CRUM
0035 NR = (CRUM/15) + 1
C = 1
0036 CALL CLUST7(10)
0037 NR = NR + 1
0038 IF(NR.LE.0) GO TO 225

```

ORIGINAL PAGE IS
 OF POOR QUALITY

3-11
 13

Figure 4.— Listing for the CCIT7 processor program.

```

FORTRAN IV-PLUS V02-51      12143106      37-JUL-78      PAGE 2
CG17.PTN /TRIPLCKCS/HQ

0039      CALL READRC
0040      GO TO 235
C
0041      IF (DAYS) GO TO 235
C
0042      INITIALIZE THE DAY OUTPUT FILES AND COUNTERS
CALL DBTS7
C
0043      READ DBTS RECORDS
N = 0
C
0044      CALL READRC
0045      IF (LINE) GO TO 230
0046      N = N + 1
C
0047      PREPARE A 'D' RECORD
CALL DBTS1
C
0048      IF (N) GO TO 240
CLOSE INPUT FILE
CALL CLOSE(1)
C
0049      250      CALL CLOSE(1)
0050      PRINT 102
0051      GO TO 260
C
0052      260      CALL PRINTY
0053      CALL CLOSE(1)
C
0054      STOP
0055      FORMAT(12)
0056      301      FORMAT(24A1,12)
0057      302      FORMAT(SM1)
0058      END

```

3-72
14

Figure 4.— Concluded.

3.3.2 SUBROUTINE READH

3.3.2.1 Linkage

Subroutine READH calls subroutine TURNON.

3.3.2.2 Interface

READH interfaces with TURNON via an integer parameter (passed on call) giving the logical unit number to be opened and via COMMON block NAME containing the name of the file to be opened. COMMON block FNAME (see section 3.3.1.2.2) interfaces CCIT7 with READH. COMMON block BUF provides no true interfacing function for this routine.

3.3.2.2.1 COMMON Block NAME

NAME contains a 25-byte array, NM, which contains the complete name of a file to be opened by subroutine TURNON. NAME also interfaces several subroutines with subroutine PRINT7. The last byte of array NM should contain the null (0) character.

3.3.2.3 Input

Header records from the CCIT input file are input.

3.3.2.4 Output

The only output is a read error message to the line printer.

3.3.2.5 Storage

READH requires 504 words of storage.

3.3.2.6 Description

Subroutine READH spaces past the three 80-byte CCIT header records, and the CCIT file name is written into the NM array. Subroutine TURNON opens the file on unit 1, the three records are read, and READH returns to CCIT7.

3.3.2.7 Flow Chart

The flow diagram for subroutine READH is given in figure 5.

3.3.2.8 Listing

The listing for this subroutine is given in figure 6.

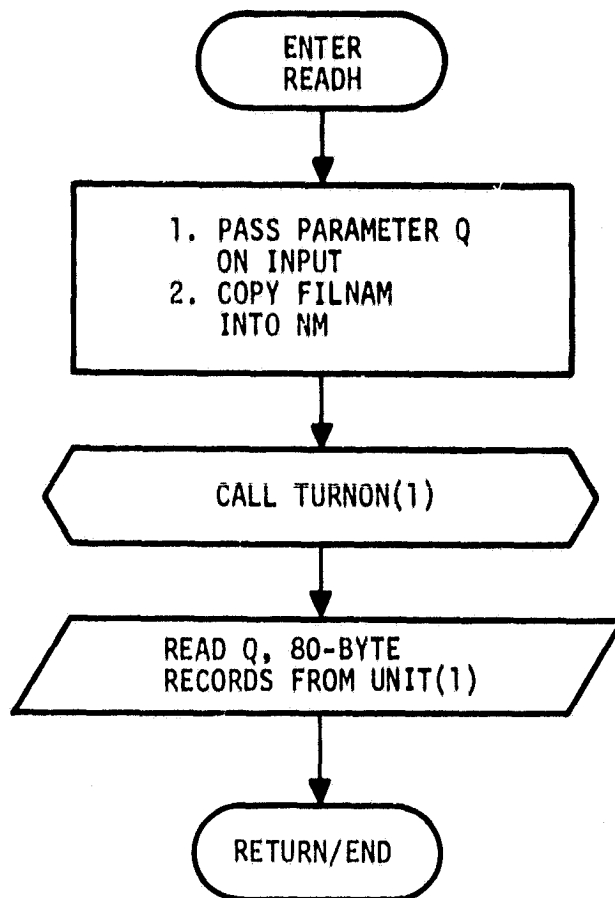


Figure 5.— Flow diagram for subroutine READH.

```

0001 SUBROUTINE READH10J
0002 IMPLICIT INTEGER(A-Z)
0003 READ UNIT=10,RECORG='F',FMT='(A10)',FILE='FILNAM124'.A(120)
0004 BYTE M(25),FILNAM(24).A(120)
0005 COMMON/MSW/PILVAP,SNIP
0006 C=LEN(M)/N
0007 DO 10 J=1,24
0008   M(J)=FILNAM(J)
0009   N(I25)=0
0010   C=LEN(M(I25))
0011   DO 1 J=1,C
0012     M(I25+J)=FILNAM(I25+J)
0013   RETURN
0014 10 PRINT*,M(J)
0015 200 FORMAT(' READ ERROR IN READH1 ITERATION ',I2, '/')
0016 STOP
0017 END

```

3-76
18

ORIGINAL PAGE IS
OF POOR QUALITY

Figure 6.- Listing for subroutine READH1.

3.3.3 SUBROUTINE READRC

3.3.3.1 Linkage

Subroutine READRC is called by CCIT7.

3.3.3.2 Interface

Subroutine READRC interfaces with CCIT7 via COMMON block BUF (see section 3.3.1.2.1).

3.3.3.3 Input

One data record read from the CCIT disk file is input.

3.3.3.4 Output

A read operation error message is output to the line printer.

3.3.3.5 Storage

This subroutine requires 441 words of storage.

3.3.3.6 Description

READRC reads one 720-byte logical data record from the CCIT input file into a buffer array, A.

3.3.3.7 Flow Chart

The flow diagram for subroutine READRC is given in figure 7.

3.3.3.8 Listing

The listing for this subroutine is given in figure 8.

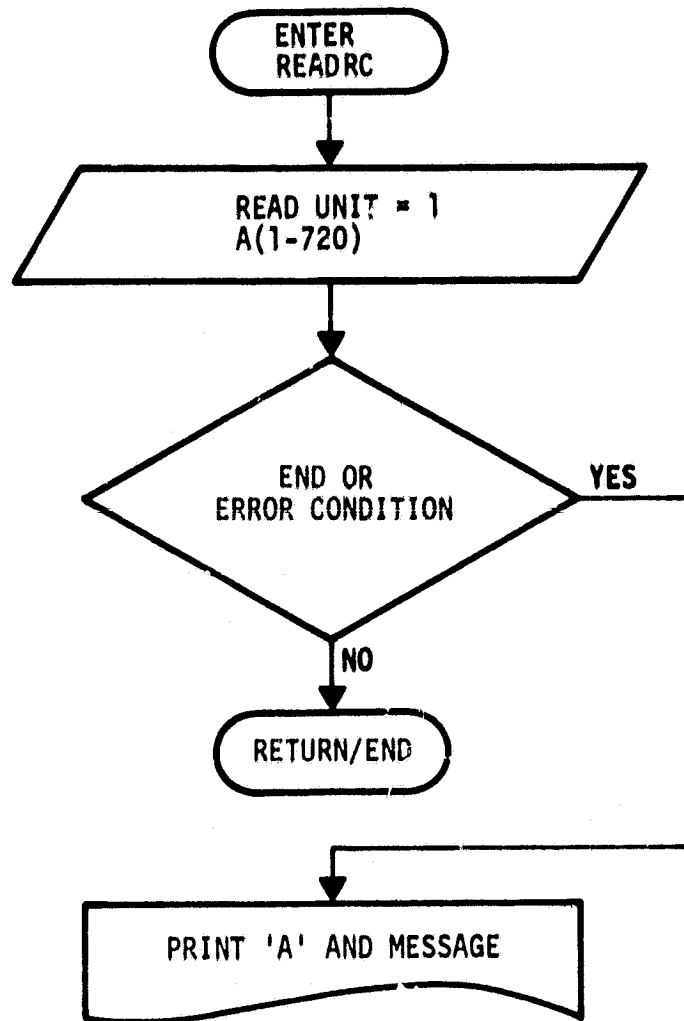


Figure 7.— Flow diagram for subroutine READRC.

```

FARTRAN IV-PLUS V02-51
READRC,FIN /INTBLRGR3/MR 07145149 05-PAY-78 PAGE 1
0001 SUBROUTINE READRC
0002 IMPLICIT INTEGER(A-Z)
0003 READ 5 CCIT DATA RECORD OF 720 BYTES INTO THE BUFFER A
0004 5 (A(720))
0005 COMMON/BUF77A
0006 READ(1,ERR=99,END=99)(A(K),K=1,720)
0007 RETURN
0008 90 PRINT 100,A
0009 100 FORTN10M '***ERROR IN READRC***, BUFFER CONTAINS***'
0010 05H ,6(120A1)
0011 END

```

3/19
21

Figure 8.— Listing for subroutine READRC.

3.3.4 SUBROUTINE HEADER

3.3.4.1 Linkage

Subroutine HEADER is called by CCIT7.

3.3.4.2 Interface

HEADER interfaces with CCIT7 via COMMON blocks BUF (see section 3.3.1.2.1) and DOTS and interfaces with PRINT7 (entry PRINTH) via COMMON block DOTS.

3.3.4.2.1 COMMON Block DOTS

DOTS contains a 23-byte array (DPRNO) that is used to hold the ERIPS DPR number, four 4-byte arrays (ACD1, ACD2, ACD3, and ACD4) that are used to store the acquisition dates used for the ERIPS run, and a 4-byte array (SMGNUM) that is used to store the LACIE segment number.

3.3.4.3 Input

There is no input to this subroutine.

3.3.4.4 Output

HEADER has no output.

3.3.4.5 Storage

This subroutine requires 445 words of storage.

3.3.4.6 Description

Subroutine HEADER selects byte data from the 'R' record of a CCIT (contained in buffer array A) and stores it into arrays in COMMON block DOTS. The data selected are the LACIE segment number [SMGNUM(1-4)], acquisition dates used for the ERIPS run [ACD1,ACD2,ACD3,ACD4], and ERIPS DPR number [DPRNO(1-23)]. The DPR number is printed in a message via a call to entry PRINTH of PRINT7.

3.3.4.7 Flow Chart

The flow diagram for subroutine HEADER is given in figure 9.

3.3.4.8 Listing

The listing for this subroutine is given in figure 10.

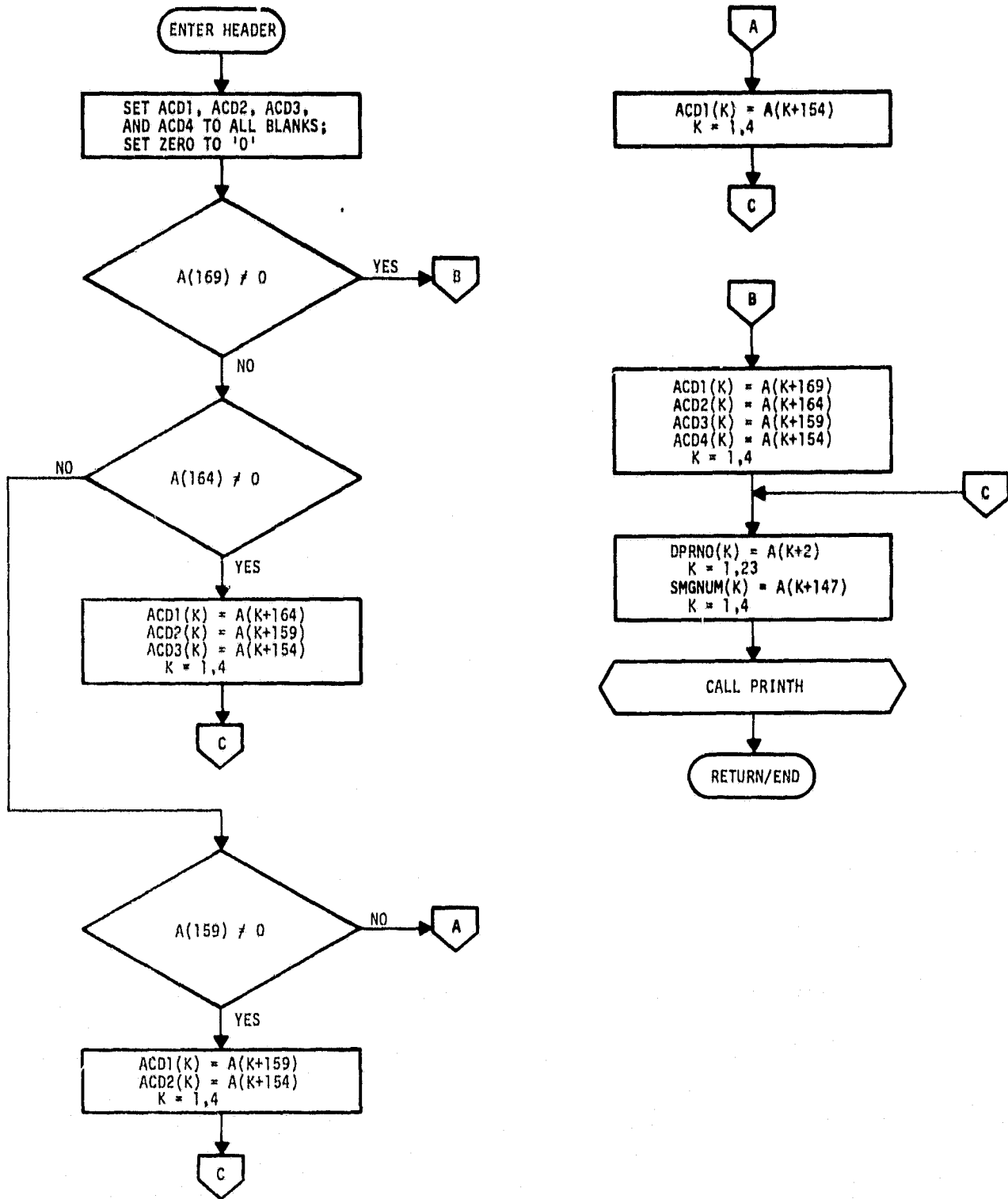


Figure 9.— Flow diagram for subroutine HEADER.

~~3-22~~
24


```

0001 C..... SUBROUTINE HEADER
C.....
C THIS SUBROUTINE OBTAINS THE DPR NUMBER AND ACQUISITION
C DATES FROM AN 'R' RECORD.
C THE ACQUISITION DATES ARE ORDERED BY LATEST DATE FIRST
C TO COMPARE TO OTHER AA SOFTWARE MODULE CONVENTIONS
C.....
0002 IPLICIT INTEGER(A-Z)
0003 BYTE A(720),ACD1(4),ACD2(4),ACD3(4),ACD4(4),
0004 COMMON/BUF/
0005 COMMON/DTS/DPRN,ACD1,ACD2,ACD3,ACD4,SHGNUM
0006 DATA ACD1/401H /ACD2/401H /ACD3/401H /ACD4/401H /
0007 DATA ZERP/180/
C CHECK TO SEE HOW MANY ACQUISITIONS WERE USED
C ASSUMED TO BE IN THE ORDER OF EARLIEST FIRST
C WITH ASCII ZERO FILL
0008 IF(A(100).NE.ZERP) GO TO 10
0009 IF(A(100).NE.ZERP) GO TO 20
0010 IF(A(150).NE.ZERP) GO TO 30
0011 DO 4 K=1,4
0012 ACD1(K) = A(K*194)
0013 GO TO 40
0014 DO 5 K=1,4
0015 ACD1(K) = A(K*199)
0016 ACD2(K) = A(K*194)
0017 GO TO 40
0018 DO 6 K=1,4
0019 ACD1(K) = A(K*104)
0020 ACD2(K) = A(K*199)
0021 ACD3(K) = A(K*194)
0022 GO TO 40
0023 DO 7 K=1,4
0024 ACD1(K) = A(K*109)
0025 ACD2(K) = A(K*104)
0026 ACD3(K) = A(K*199)
0027 ACD4(K) = A(K*194)
C WRITE THE DPR NUMBER
0028 DO 3 K=1,23
0029 DPRN(K) = A(K*2)
C WRITE SEGMENT NUMBER
0030 DO 1 K=1,4
0031 SHGNUM(K) = A(K*14)
C PRINT MESSAGE ON PRINTER
0032 CALL PRINTM
0033 RETURN
0034 END

```

3-23
35

Figure 10.— Listing for subroutine HEADER.

3.3.5 SUBROUTINE BIASC7

3.3.5.1 Linkage

BIASC7 is called once by the CCIT7 program. It calls subroutine TURNON once. An additional entry point, BIAS, is called one or more times by CCIT7.

3.3.5.2 Interface

BIASC interfaces with CCIT7 via COMMON blocks BUF (see section 3.3.1.2.1) and B7 (see section 3.3.1.2.3), and with TURNON via COMMON block NAME (see section 3.3.2.2.1).

3.3.5.3 Input

There is no input to this subroutine.

3.3.5.4 Output

BIASC7 writes two unformatted records onto unit 3. The data contained in this record are detailed in the appendix.

3.3.5.5 Storage

This subroutine requires 837 words of storage.

3.3.5.6 Description

BIASC7 codes the output file name as SSSSYDDDD.CLO, where SSSS is the segment number and YYDDD is the classification date. Unit 3 is opened for output via a call to subroutine TURNON. Then a record containing the integer NCAT (the number of categories or classes) is written on unit 3, and 12 times NCAT bytes of character data are output as the second record.

3.3.5.7 Flow Chart

The flow diagram for subroutine BIASC7 is given in figure 11.

3.3.5.8 Listing

The listing for this subroutine is given in figure 12.

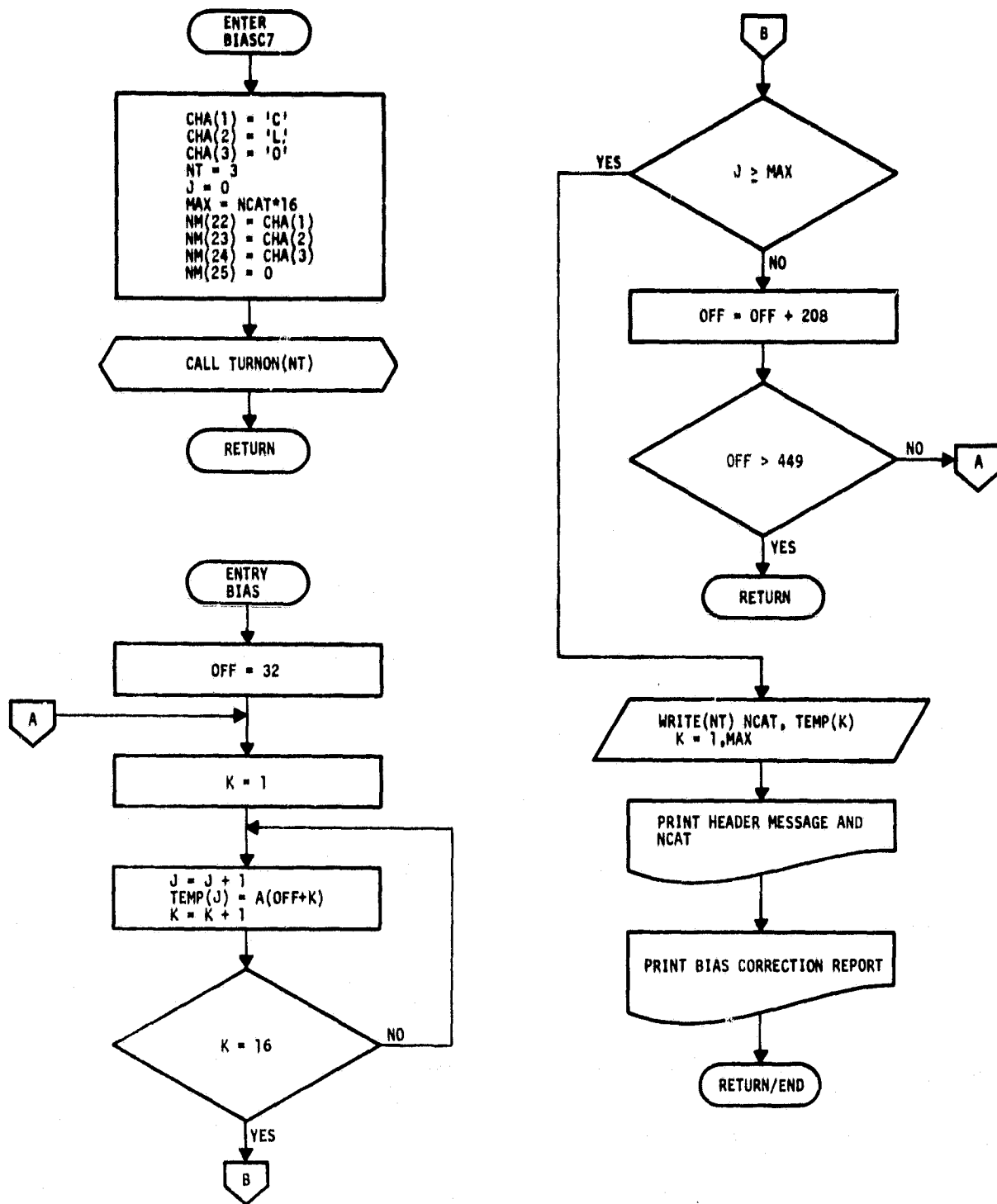


Figure 11.— Flow diagram for subroutine BIASC7.

3-25
27

```

0001 SUBROUTINE BIASC7
0002 IMPLICIT INTEGER(I*)
C.....
C USES THE BIAS CORRECTION TABLE STRATIFIED AREA
C ESTIMATE DATA FROM A CCIT 'BA' RECORD
C FOR THE FIRST RECORD OF THE
C SSSSYDDJ.CLO OUTPUT FILE, LEAVES FILE OPEN FOR
C WRITING OF THE 'RE RECORD' NUMBER OF CLUSTERS,
C AND THE ALPHANUMERIC CLUSTER-CBY MATCH TABLE
C BY SUBROUTINE CUST7
C.....
0003 DATE = N(125), SGNUM(4), OPRA(23), ACD1(4), ACD2(4), ACD3(4),
0004 I = CCIT(1), CHA(3), TEM(12)
0005 CPMR(7) = 0
0006 C24-25/NAME/AM
0007 CPMR(7) = 0
0008 DATA CHAZING, 1ML, 1L0/
C.....
C THIS PART OPENS OUTPUT FILE AND SETS UP THE
C PRECESSING 'RE' ENTRY BIAS IS CALLED BNCE
C FOR EACH 'BA' RECORD PROCESSED
C.....
0010 NT = 3
0011 J = 0
0012 MAX = NCAT(15)
C CODE NAME FOR OUTPUT FILE IN FORM SSSSYDDJ.CLO
0013 BN(22) = CHA(1)
0014 N(23) = CHA(2)
0015 N(24) = CHA(3)
0016 N(25) = 0
C OPEN OUTPUT FILE = UNFORMATTED
0017 CALL TOR(NT, J)
0018 RETURN
C.....
C ENTER HERE ENCE FOR EACH 'BA' RECORD PROCESSED.
C ONLY ALL CLASSES ('CATGORTCS') HAVE BEEN WRITTEN
C INTO BUFFER TEMP. RATE TO DISK AND PRINT REPORT
C.....
0019 ENTRY BIAS
0020 OFF = 0
0021 2 0.1 4.1, 1.6
0022 J = J + 1
0023 TEMP(J) = A(CPMR)
0024 IF (J.GE. 20) GO TO 10
0025 OFF = OFF + 20
0026 IF (OFF.GE. 400) RETURN
0027 GO TO 2
C.....
C WRITE DISK RECORD AND PRINTER REPORT
C.....
0028 10 WRITE(NT) NCAT
0029 WRITE(NT) TEMP(1:20)
0030 WRITE(6, 200) CAT

```

ORIGINAL BASIS OF
 OF POOR QUALITY

3-26
 2.8

Figure 12.- Listing for subroutine BIASC7.

```

PERTKRN 110PLUS 702-51      10110125  29-JUL-78      PAGE 2
01ASC7.FTN /TRISLOCKS/WR

0031  -RITE(6,300)(TEMPIN),K01,PA1
0032  RETURN
0033  200  F0RNAT11H ,15X,71AS C0RRECTIEN '///.19X.F0R ',12,' CATAGORIES')
0034  300  F0RNAT11H ,261X,'CATAGORY LABEL ',11,' POPULATION ',211777.
      1 10X,'UNCORRECTED ',11,' ',1211,' CORRECTED ',111,' ',1211.
      2  VARIANCE ',211,' ',11,211777)
0035  END

```

~~3-27~~
29

Figure 12.- Concluded.

3.3.6 SUBROUTINE CLUST7

3.3.6.1 Linkage

Subroutine CLUST7 is called one to four times by CCIT7. CLUST7 calls RITE7 for data output and calls the PRINTC entry to PRINT7 for an output message.

3.3.6.2 Interface

CLUST7 interfaces with CCIT7 via COMMON block BUF (see section 3.3.1.2.1) and passed parameter Q, with RITE7 via COMMON block CLUSTR (see section 3.3.1.2.4), and with PRINT7 via passing parameter RCNUM on call.

3.3.6.3 Input

There is no input to this subroutine.

3.3.6.4 Output

Subroutine CLUST7 has no output.

3.3.6.5 Storage

This subroutine requires 836 words of storage.

3.3.6.6 Description

CLUST7 processes the CCIT 'C' records to provide the total number of clusters and the identity of the analyst-labeled (type 1) dot used to name each cluster.

On the first record processed, CLUST7 decodes bytes 4 and 5 to obtain the total number of clusters (CNUM) and bytes 6 and 7 to obtain the number of clusters contained on the record [RCNUM(1-15)]. Then for each cluster, the 12 bytes representing the cluster name (6 bytes) and dot name (6 bytes) used in labeling the cluster are copied into the array CNAME. PRINT7 is called via entry PRINTC to print a message containing the parameter RCNUM. When CNUM sets of data are written into CNAME, CLUST7 calls subroutine RITE7 for output.

3.3.6.7 Flow Chart

The flow diagram for subroutine CLUST7 is given in figure 13.

3.3.6.8 Listing

The listing for this subroutine is given figure 14.

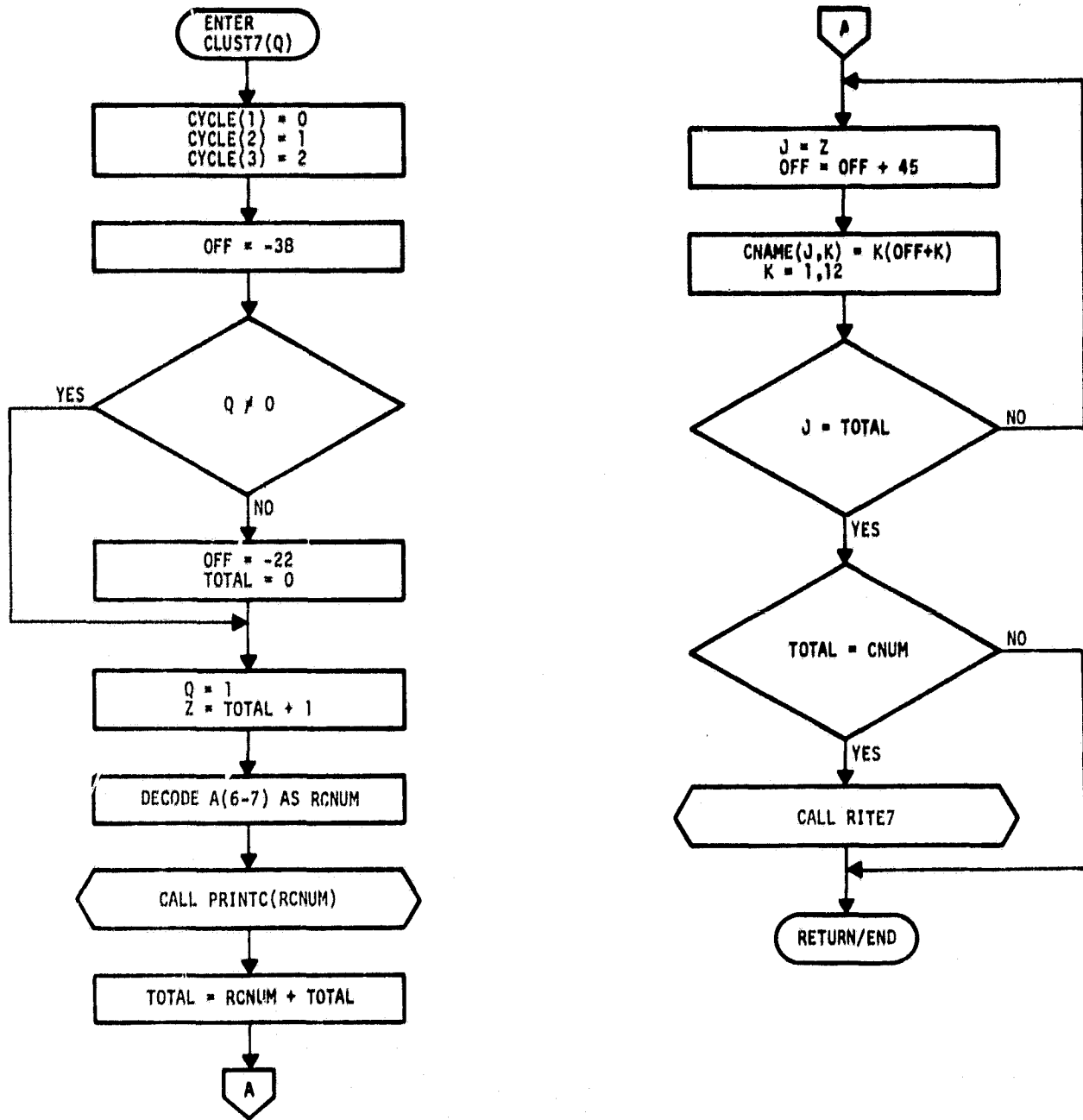


Figure 13.— Flow diagram for subroutine CLUST7.

FORTRAN IV-PLUS V02-51 14124148 06-JUL-78 PAGE 1
CLUST7.FTN /TRIPLOCKS/HR

```
0001 SUBROUTINE CLUST7(0)
0002 IMPLICIT INTEGER(I02)
C READS COT 'C' RECORDS AND CUTPLTS CLUSTER LABEL AND CLUSTER.
C DBY MATCH DATA TO DISC FILE NAMED SSSSYDDDD,CLO.
0003 BYTE A(720),CYCLE(3),CNAME(80:12)
C CNAME IS A STORAGE ARRAY TO PERPT WRITING THE DATA IN ONE
C RECORD TO DISC IN A FILE NAMED SSSSYDDDD,CLO
C CNUM IS THE TOTAL NUMBER OF CLUSTERS
C ZFF IS THE BYTE OFFSET WITHIN A 'C' RECORD
C RCNUM IS THE NO. OF CLUSTERS IN THE CURRENT RECORD
C ..
0004 CCMRN/BUFA
0005 CCMRN/CLUST/CNAME,CNUM
0006 DATA CYCLE/10,14,112/
0007 ZFF = 538
0008 IF(G.ME.0) GO TO 9
0009 ZFF = 22
0010 TOTAL = 0
0011 0 = 1
0012 2 = TOTAL + 1
0013 DECODE(2,100,ALC)RCNUM
0014 CALL PRINT(CNUM)
0015 TOTAL = RCNUM + TOTAL
0016 DO 1 J=0,TOTAL
0017 ZFF = ZFF + 42
0018 DO 1 K=1,12
0019 CNAME(K) = A(ZFF,K)
C CHECK IF DONE
C CALL RIVEN FOR DISK WRITE AND PRINTED REPORT
C
C
0020 IF(ZSTAL.EC.CNUM) CALL RITEZ
0021 RETURN
0022 100 PARMAT(12)
0023 END
```

3-31
33

ORIGINAL PAGE IS
OF POOR QUALITY

Figure 14.— Listing for subroutine CLUST7.

3.3.7 SUBROUTINE RITE7

3.3.7.1 Linkage

Subroutine RITE7 is called once by subroutine CLUST7.

3.3.7.2 Interface

Subroutine RITE7 interfaces with CLUST7 via COMMON block CLUSTR (see section 3.3.1.2.4).

3.3.7.3 Input

There is no input to RITE7.

3.3.7.4 Output

RITE7 writes two records onto a previously opened file (unit 3). This unit is opened in subroutine BIASC7 as an unformatted FORTRAN disk file. In addition, RITE7 writes a line printer report of the cluster-dot match data for AA evaluation.

3.3.7.5 Storage

This subroutine requires 484 words of storage.

3.3.7.6 Description

RITE7 writes two records onto unit 3. The first record is a single integer, CNUM. The second record consists of the array CNAME as CNUM 12-byte elements. The output file is closed via a call to the system routine CLOSE.

3.3.7.7 Flow Chart

The flow diagram for subroutine RITE7 is given in figure 15.

3.3.7.8 Listing

The listing for this subroutine is given in figure 16.

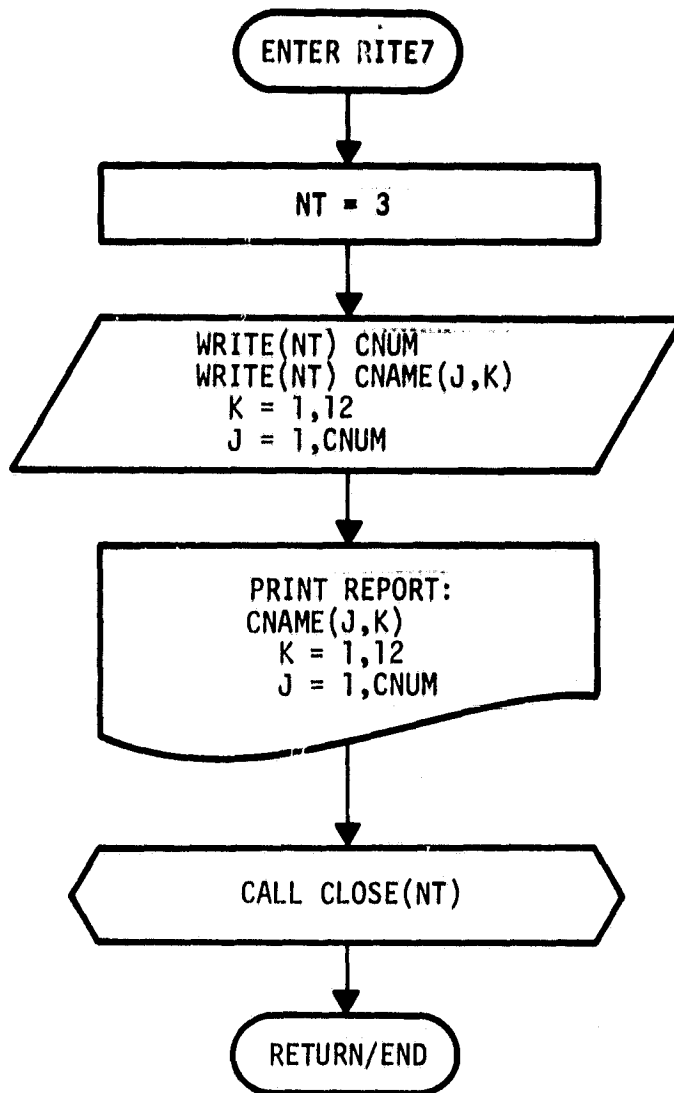


Figure 15.— Flow diagram for subroutine RITE7.

```

FBRTRAY IV-P1US 102351 14112164 30-JUN-78 PAGE 1
RITE7.FTN /TRILRCL5/13

0001 SUPPLY THE RITE7
0002 IMPLICIT INTEGER(=0)
C PRINTS AN UNFORMATTED FILE OF CLUSTER NAME AND CLUSTER * AT DBY MATCH
C DATA PATH RISC FOR AA PROCESSING...LACIETY VERSION
C ALSO PRINTS REPORT FOR AA INTERNAL USE
C OUTPUT WAS OPENED IN SUBROUTINE BIASC
0003 BYTE NAME(0,12)
0004 CNAME/CLUSTER/CNAME,CNUM
0005 I * 3
C PRINTS 14P RECORDS, FIRST, THE NUMBER OF CLUSTERS, THEN, THE ALPHANUMERIC
C DATA, 12 CHARACTERS, PER CLUSTER
0006 WRITE(N)CNUM
0007 WRITE(N)((CNAME(J),K=1,12),J=1,CNUM)
0008 WRITE(0,100)((CNAME(J,K),K=1,12),J=1,CNUM)
C CLOSE OUTPUT FILE
0009 CALL CLOSE(NY)
0010 RETURN
0011 100 FORMAT(24,15X,'CLUSTER NAME DBY NAME',//60I10X,6A1,9X,6A1,/)
0012 END

```

3-34
36

ORIGINAL PAGE IS
OF POOR QUALITY

Figure 16.— Listing for subroutine RITE7.

3.3.8 SUBROUTINE TURNON

3.3.8.1 Linkage

Subroutine TURNON is called by subroutines READH, BIASC7, and DOTS7.

3.3.8.2 Interface

TURNON interfaces with its calling routines via COMMON block NAME (see section 3.3.2.2.1) and a passed parameter, NT.

3.3.8.3 Input

There is no input to this subroutine.

3.3.8.4 Output

TURNON has no output.

3.3.8.5 Storage

This subroutine requires 162 words of storage.

3.3.8.6 Description

TURNON opens a file with the file name contained in byte array NM. If NT equals 1, the input file is opened as UNIT equals 1. If NT equals 2 to 6, an unformatted file is opened as unit NT. If NT is greater than 6, a formatted file is opened as unit (NT - 6). Prior to opening the file, the routine prints a message containing the passed unit number parameter, NT, and the file name, NM.

3.3.8.7 Flow Chart

The flow diagram for subroutine TURNON is given in figure 17.

3.3.8.8 Listing

The listing for this subroutine is given in figure 18.

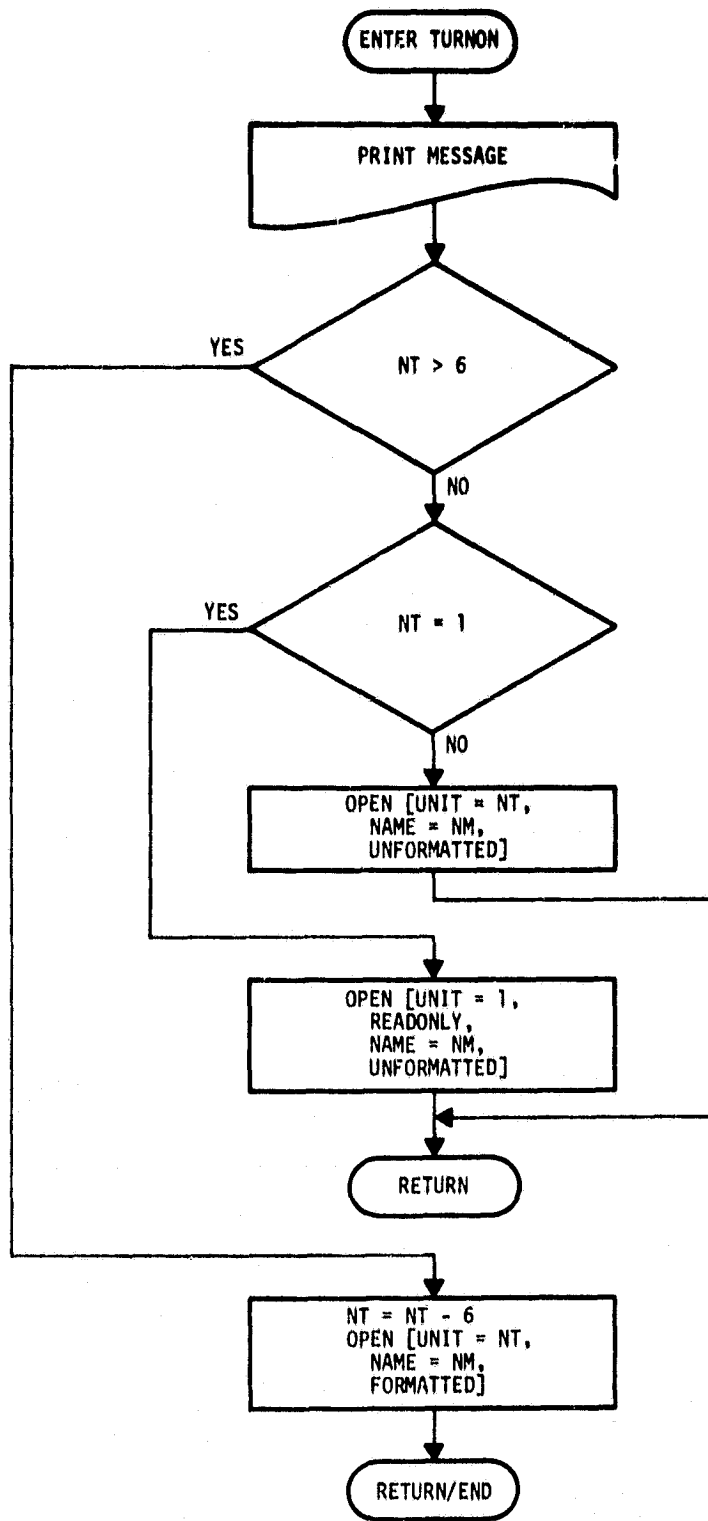


Figure 17.— Flow diagram for subroutine TURNON.

```

FORTRAN IV-PLUS V02-51          07148130    05-JAY-78          PAGE 1
TURNON:PTN 77R0C0007R0

0001 SUBROUTINE TURNON
0002 IMPLICIT INTEGER(A-Z)
C
C
C ROUTINE OPENS DIRECT ACCESS FILES FOR PROGRAM CC10A
C IF UNIT NUMBER PASSED INP) IS -1, OPEN A PERMITTED FILE
C ELSE OPEN AN UNFORMATTED FILE
C USE UNIT 1 FOR A FILE WHICH MUST BE YIELD
C
0003
0004
0005
0006 100  FORMAT(1H0,' OPENING UNIT ',11,' ON DISC TO ACCESS FILE '
0007 0.2931.77)
0008 IF(N1,0,6) GO TO 2
0009 IF(N1,0,1) OPEN(UNIT,NAMENH,TYPE,ACCESS,PERM)
0010 0 TYPE='B',ACCESS='SEQUENTIAL',PERM='UNFORMATTED')
0011 OPEN(UNIT,NAMENH,TYPE='LNK',ACCESS='SEQUENTIAL',
0012 PERM='UNFORMATTED')
0013 RETURN
0014
0015
0016
0017
0018
0019
0020
0021
0022
0023
0024
0025
0026
0027
0028
0029
0030
0031
0032
0033
0034
0035
0036
0037
0038
0039
0040
0041
0042
0043
0044
0045
0046
0047
0048
0049
0050
0051
0052
0053
0054
0055
0056
0057
0058
0059
0060
0061
0062
0063
0064
0065
0066
0067
0068
0069
0070
0071
0072
0073
0074
0075
0076
0077
0078
0079
0080
0081
0082
0083
0084
0085
0086
0087
0088
0089
0090
0091
0092
0093
0094
0095
0096
0097
0098
0099
0100
0101
0102
0103
0104
0105
0106
0107
0108
0109
0110
0111
0112
0113
0114
0115
0116
0117
0118
0119
0120
0121
0122
0123
0124
0125
0126
0127
0128
0129
0130
0131
0132
0133
0134
0135
0136
0137
0138
0139
0140
0141
0142
0143
0144
0145
0146
0147
0148
0149
0150
0151
0152
0153
0154
0155
0156
0157
0158
0159
0160
0161
0162
0163
0164
0165
0166
0167
0168
0169
0170
0171
0172
0173
0174
0175
0176
0177
0178
0179
0180
0181
0182
0183
0184
0185
0186
0187
0188
0189
0190
0191
0192
0193
0194
0195
0196
0197
0198
0199
0200
0201
0202
0203
0204
0205
0206
0207
0208
0209
0210
0211
0212
0213
0214
0215
0216
0217
0218
0219
0220
0221
0222
0223
0224
0225
0226
0227
0228
0229
0230
0231
0232
0233
0234
0235
0236
0237
0238
0239
0240
0241
0242
0243
0244
0245
0246
0247
0248
0249
0250
0251
0252
0253
0254
0255
0256
0257
0258
0259
0260
0261
0262
0263
0264
0265
0266
0267
0268
0269
0270
0271
0272
0273
0274
0275
0276
0277
0278
0279
0280
0281
0282
0283
0284
0285
0286
0287
0288
0289
0290
0291
0292
0293
0294
0295
0296
0297
0298
0299
0300
0301
0302
0303
0304
0305
0306
0307
0308
0309
0310
0311
0312
0313
0314
0315
0316
0317
0318
0319
0320
0321
0322
0323
0324
0325
0326
0327
0328
0329
0330
0331
0332
0333
0334
0335
0336
0337
0338
0339
0340
0341
0342
0343
0344
0345
0346
0347
0348
0349
0350
0351
0352
0353
0354
0355
0356
0357
0358
0359
0360
0361
0362
0363
0364
0365
0366
0367
0368
0369
0370
0371
0372
0373
0374
0375
0376
0377
0378
0379
0380
0381
0382
0383
0384
0385
0386
0387
0388
0389
0390
0391
0392
0393
0394
0395
0396
0397
0398
0399
0400
0401
0402
0403
0404
0405
0406
0407
0408
0409
0410
0411
0412
0413
0414
0415
0416
0417
0418
0419
0420
0421
0422
0423
0424
0425
0426
0427
0428
0429
0430
0431
0432
0433
0434
0435
0436
0437
0438
0439
0440
0441
0442
0443
0444
0445
0446
0447
0448
0449
0450
0451
0452
0453
0454
0455
0456
0457
0458
0459
0460
0461
0462
0463
0464
0465
0466
0467
0468
0469
0470
0471
0472
0473
0474
0475
0476
0477
0478
0479
0480
0481
0482
0483
0484
0485
0486
0487
0488
0489
0490
0491
0492
0493
0494
0495
0496
0497
0498
0499
0500
0501
0502
0503
0504
0505
0506
0507
0508
0509
0510
0511
0512
0513
0514
0515
0516
0517
0518
0519
0520
0521
0522
0523
0524
0525
0526
0527
0528
0529
0530
0531
0532
0533
0534
0535
0536
0537
0538
0539
0540
0541
0542
0543
0544
0545
0546
0547
0548
0549
0550
0551
0552
0553
0554
0555
0556
0557
0558
0559
0560
0561
0562
0563
0564
0565
0566
0567
0568
0569
0570
0571
0572
0573
0574
0575
0576
0577
0578
0579
0580
0581
0582
0583
0584
0585
0586
0587
0588
0589
0590
0591
0592
0593
0594
0595
0596
0597
0598
0599
0600
0601
0602
0603
0604
0605
0606
0607
0608
0609
0610
0611
0612
0613
0614
0615
0616
0617
0618
0619
0620
0621
0622
0623
0624
0625
0626
0627
0628
0629
0630
0631
0632
0633
0634
0635
0636
0637
0638
0639
0640
0641
0642
0643
0644
0645
0646
0647
0648
0649
0650
0651
0652
0653
0654
0655
0656
0657
0658
0659
0660
0661
0662
0663
0664
0665
0666
0667
0668
0669
0670
0671
0672
0673
0674
0675
0676
0677
0678
0679
0680
0681
0682
0683
0684
0685
0686
0687
0688
0689
0690
0691
0692
0693
0694
0695
0696
0697
0698
0699
0700
0701
0702
0703
0704
0705
0706
0707
0708
0709
0710
0711
0712
0713
0714
0715
0716
0717
0718
0719
0720
0721
0722
0723
0724
0725
0726
0727
0728
0729
0730
0731
0732
0733
0734
0735
0736
0737
0738
0739
0740
0741
0742
0743
0744
0745
0746
0747
0748
0749
0750
0751
0752
0753
0754
0755
0756
0757
0758
0759
0760
0761
0762
0763
0764
0765
0766
0767
0768
0769
0770
0771
0772
0773
0774
0775
0776
0777
0778
0779
0780
0781
0782
0783
0784
0785
0786
0787
0788
0789
0790
0791
0792
0793
0794
0795
0796
0797
0798
0799
0800
0801
0802
0803
0804
0805
0806
0807
0808
0809
0810
0811
0812
0813
0814
0815
0816
0817
0818
0819
0820
0821
0822
0823
0824
0825
0826
0827
0828
0829
0830
0831
0832
0833
0834
0835
0836
0837
0838
0839
0840
0841
0842
0843
0844
0845
0846
0847
0848
0849
0850
0851
0852
0853
0854
0855
0856
0857
0858
0859
0860
0861
0862
0863
0864
0865
0866
0867
0868
0869
0870
0871
0872
0873
0874
0875
0876
0877
0878
0879
0880
0881
0882
0883
0884
0885
0886
0887
0888
0889
0890
0891
0892
0893
0894
0895
0896
0897
0898
0899
0900
0901
0902
0903
0904
0905
0906
0907
0908
0909
0910
0911
0912
0913
0914
0915
0916
0917
0918
0919
0920
0921
0922
0923
0924
0925
0926
0927
0928
0929
0930
0931
0932
0933
0934
0935
0936
0937
0938
0939
0940
0941
0942
0943
0944
0945
0946
0947
0948
0949
0950
0951
0952
0953
0954
0955
0956
0957
0958
0959
0960
0961
0962
0963
0964
0965
0966
0967
0968
0969
0970
0971
0972
0973
0974
0975
0976
0977
0978
0979
0980
0981
0982
0983
0984
0985
0986
0987
0988
0989
0990
0991
0992
0993
0994
0995
0996
0997
0998
0999
1000

```

Figure 18.— Listing for subroutine TURNON.

3-21
39

ORIGINAL PAGE
OF POOR QUALITY

3.3.9 SUBROUTINE DOTS7

3.3.9.1 Linkage

Subroutine DOTS7 is called by CCIT7 once via the main entry and 14 times via entry DOTS1. DOTS7 calls subroutine TURNON twice and subroutines PRINT7 (via entry PRINTD) and STCOD7 once.

3.3.9.2 Interface

Subroutine DOTS7 interfaces with TURNON via COMMON block NAME (see section 3.3.2.2.1) and with CCIT7 via COMMON blocks BUF (see section 3.3.1.2.1) and DOTS (see section 3.3.4.2.1).

3.3.9.3 Input

There is no input to this subroutine.

3.3.9.4 Output

Subroutine DOTS7 writes formatted, card-image records onto two disk-based output files opened on the initial call to the routine.

3.3.9.5 Storage

This subroutine requires 976 words of storage.

3.3.9.6 Description

DOTS7 processes CCIT 'D' records into two formatted files of analyst-labeled dots. When called as DOTS7, the routine initializes the unit parameters, NT and MT, and the dot counters, KOUNT1 and KOUNT2. Then the elements of the array NM are set to name the file to receive the type 1 analyst-labeled dot data, and TURNON is called to open this file. NM(24) is redefined (1 → 2) to provide the name of the type 2 dot output file, and TURNON is called to open this file. Subroutine STCOD7 is called to obtain the two-byte parameter ST, the alphabetic state code for the segment. Control then returns to CCIT7.

When called as DOTS1, the routine processes one 720-byte 'D' record. For each analyst-labeled dot, one record is written. For type 1 dots, the data are written onto unit 2; for type 2 dots, the data are written onto unit 3. KOUNT1 is incremented for each type 1 dot, and KOUNT2 is incremented for each type 2 dot.

After processing all 209 dots (14 calls from CCIT7), a blank record is written into each output file. Then both output files are closed, and a message listing KOUNT1 and KOUNT2 is printed via a call to the PRINT7 subroutine entry PRINTD.

3.3.9.7 Flow Chart

The flow diagram for subroutine DOTS7 is given in figure 19.

3.3.9.8 Listing

The listing for this subroutine is given in figure 20.

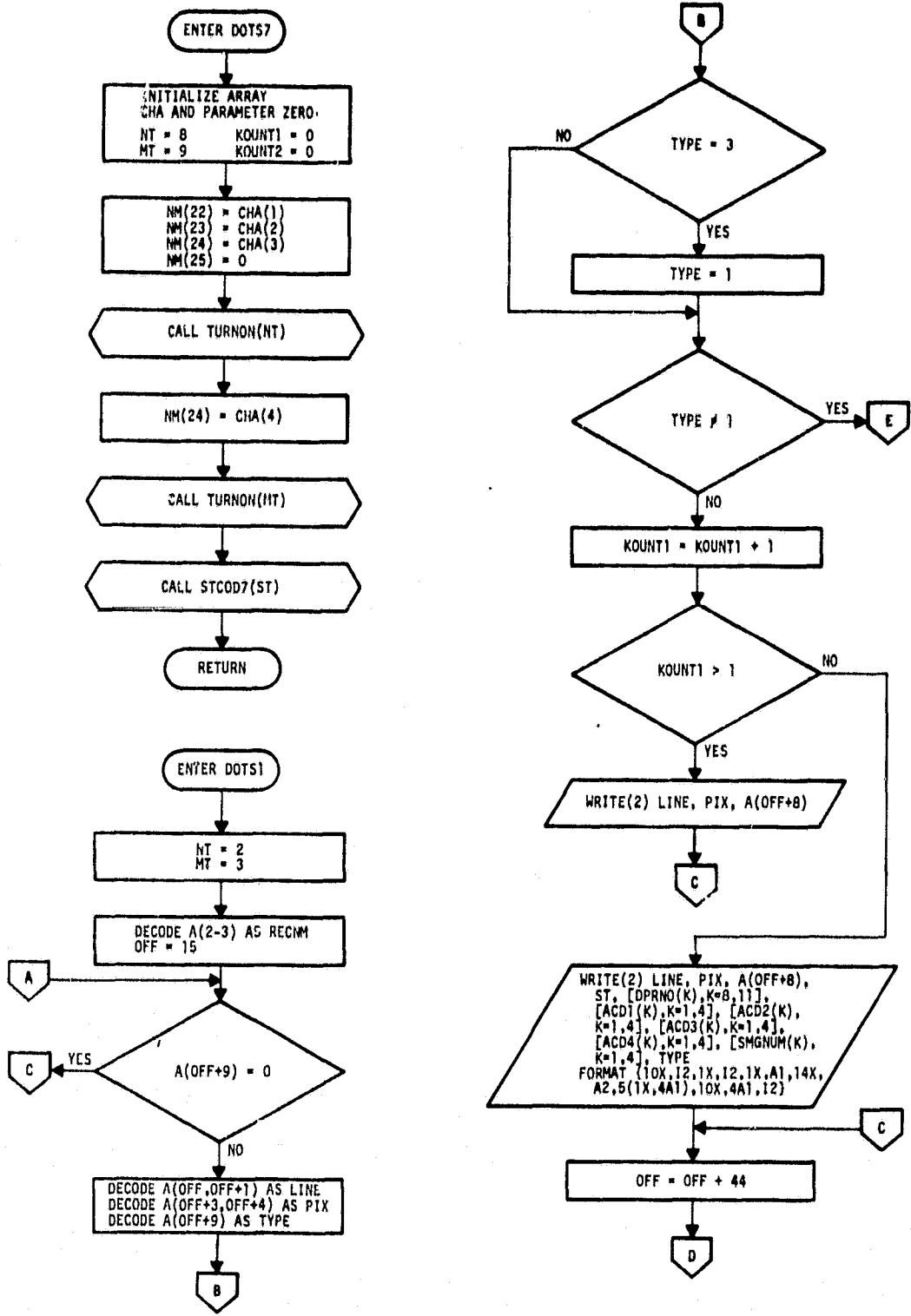


Figure 19.— Flow diagram for subroutine DOTS7.

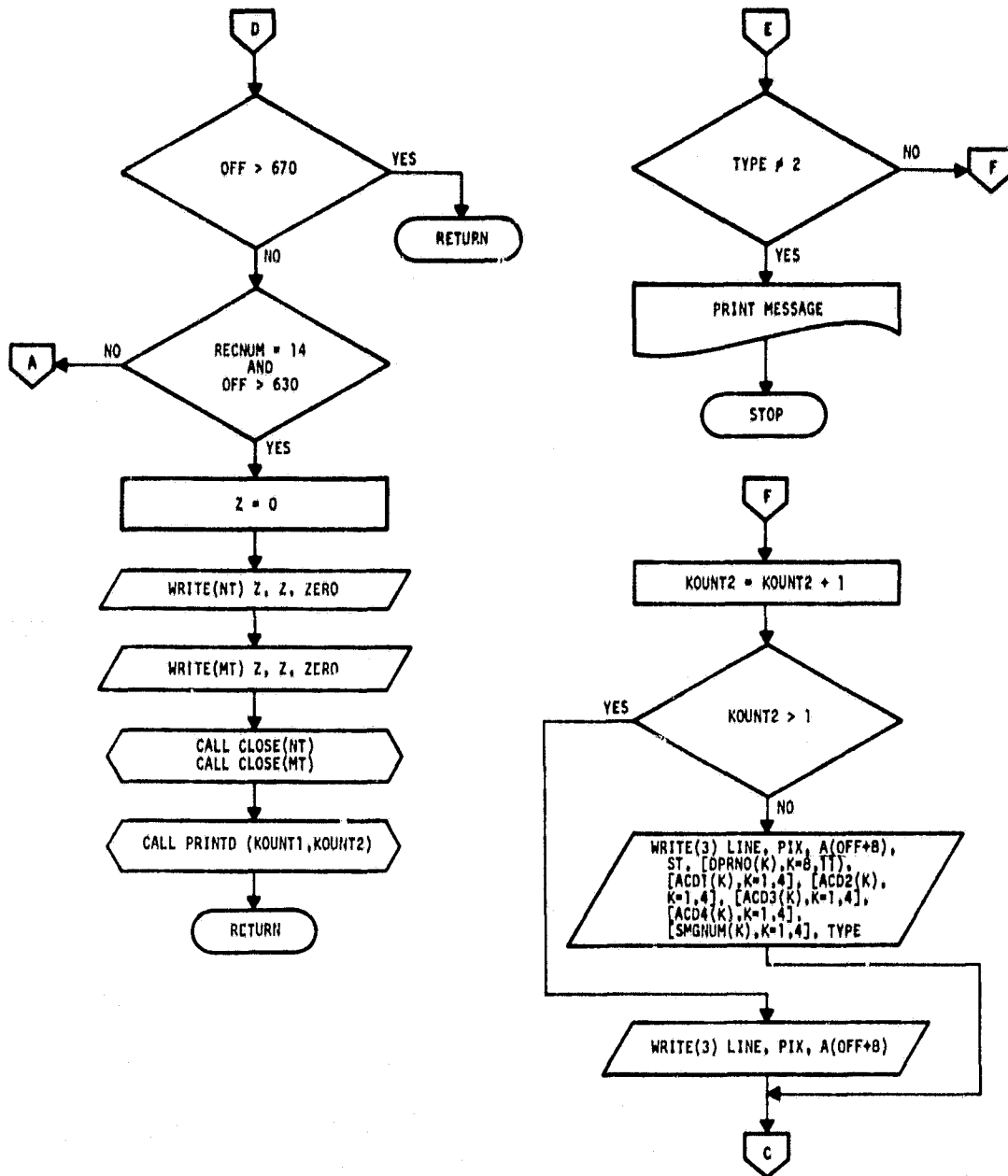


Figure 19.— Concluded.

~~3-41~~
43

```

0001 C     PRINT I, J, K
0002 C     .....
0003 C     .....
0004 C     .....
0005 C     .....
0006 C     .....
0007 C     .....
0008 C     .....
0009 C     .....
0010 C     .....
0011 C     .....
0012 C     .....
0013 C     .....
0014 C     .....
0015 C     .....
0016 C     .....
0017 C     .....
0018 C     .....
0019 C     .....
0020 C     .....
0021 C     .....
0022 C     .....
0023 C     .....
0024 C     .....
0025 C     .....
0026 C     .....
0027 C     .....
0028 C     .....
0029 C     .....

```

Figure 20.— Listing for subroutine DOTS7.

3-42
44

ORIGINAL PAGE IS
OF POOR QUALITY

```

EXPRT: 10-10-77 12:01:26 12-SEP-77
P2Y57.PF /TRFSL7C4S/43

C TYPE 1 D'Y S' RELACL AS TYPE 1
RECEI: 01101014(PPFV)TYPE
IFTYPE: P2.31YESS1
IFTYPE: P2.31YESS1
WRITE TYPE 1 PYS
WRITE TYPE 1 PYS
IF(A=1,C,100) 7:0
WRITE(1,40)LINE,P1,P2(PPF0)
1 (ACD1(K),K01,4),(ACD2(K),K01,4),(ACD3(K),K01,4)
2 (ACD4(K),K01,4),(ACD5(K),K01,4),TYPE
  2 12 12 701
DC36 WRITE(1,140)LINE,P1,P2(PPF0)
DC37 700
DC38 701 52 12 999
DC39 C WRITE OUT TYPE 2 LOTS
COUNT(KOUNT1) 1
DC40 IF(KOUNT2,51,4) 52 12 715
DC41 WRITE(1,140)LINE,P1,P2(PPF0)SY:(KOUNT2,51,4)
DC42 1 (ACD1(K),K01,4),(ACD2(K),K01,4),(ACD3(K),K01,4),(ACD4(K),K01,4)
  2 41:(SPONOR(K),K02,9)TYPE
  52 12 999
DC43 WRITE(1,140)LINE,P1,P2(PPF0)
DC44 710
DC45 999 EFF = OFF = 44
DC46 IF(PPF,67,670) RETURN
DC47 IF(RECM,50,54,AND,OFF,GT,630) GO TO 800
DC48 GO TO 200
C D0VE = WRITE BLANK RECORDS CLOSE BOTH OUTPUT FILES AND PRINT
  800
  2 0 0
DC49 WRITE(1,140) 2,2,ZERO
DC50 WRITE(1,140) 2,2,PER
DC51 CALL CLOSE(1)
DC52 CALL CLOSE(1)
DC53 CALL PRINT(KOUNT1,KOUNT2)
DC54 RETURN
C WRING TYPE D071 WRITE ERROR MESSAGE
DC55 PRINT 120:(10PPF0)
DC56 STOP
DC57
DC58 FORMATM * ERROR IN DETS...IMPROPER TYPE *CRS:??
DC59 100 FORMAT(I2)
DC60 110 FORMAT(I1)
DC61 140 FORMAT(10X,12,1X,12,1X,41,10X,42,5:11,4A1),10X,4A1,12)
DC62 END

```

3-43
45

Figure 20.- Concluded.

3.3.10 SUBROUTINE STCOD7

3.3.10.1 Linkage

STCOD7 is called once by subroutine DOTS7.

3.3.10.2 Interface

STCOD7 interfaces with DOTS7 via COMMON block FNAME (see section 3.3.1.2.2) and passed parameter ST.

3.3.10.3 Input

There is no input to this subroutine.

3.3.10.4 Output

Subroutine STCOD7 has no output.

3.3.10.5 Storage

This subroutine requires 1099 words of storage.

3.3.10.6 Description

STCOD7 locates the correct two-character alphabetic state code, ST, for a given segment number via table lookup. Note: The table given is only valid for AA LACIE U.S. Great Plains blind sites for the 1978 Transition Year.

3.3.10.7 Flow Chart

The flow diagram for subroutine STCOD7 is given in figure 21.

3.3.10.8 Listing

The listing for this subroutine is given in figure 22.

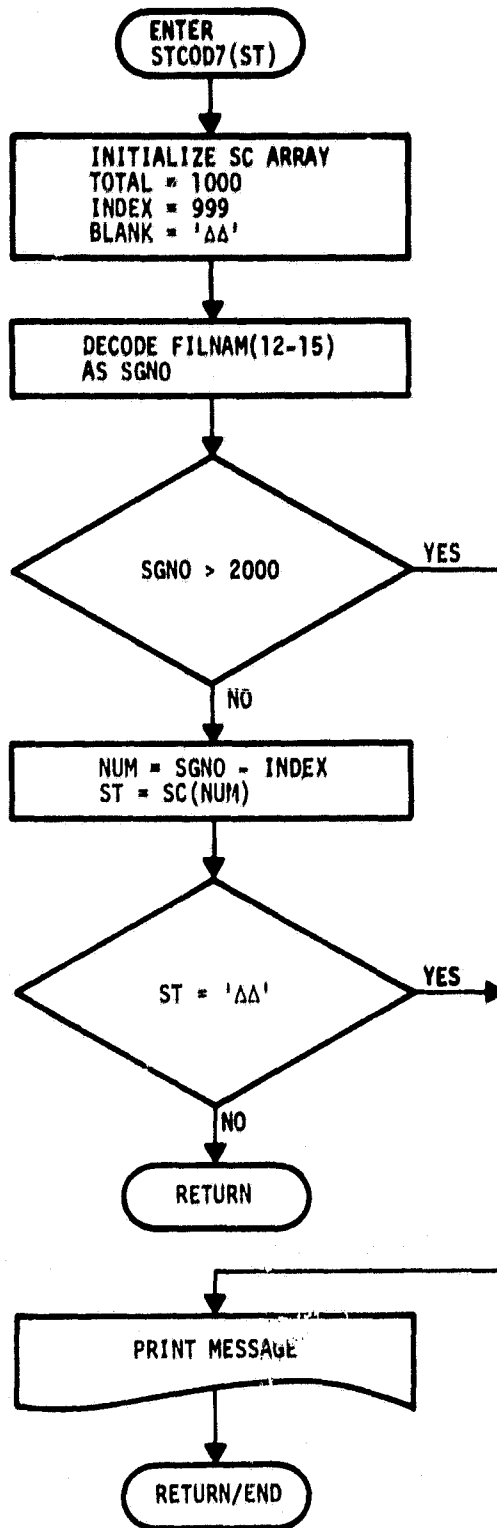


Figure 21.— Flow diagram for subroutine STCOD7.

3-45
47

3.3.11 SUBROUTINE PRINT7

3.3.11.1 Linkage

PRINT7 is called by CCIT7 via entries PRINT7 and PRINTE, by HEADER via entry PRINTH, by DOTS via entry PRINTD, and by CLUST7 via entry PRINTC. All other called routines are Image Processor system routines.

3.3.11.2 Interface

PRINT7 interfaces with HEADER via COMMON block DOTS (see section 3.3.4.2.1), with CCIT7 via COMMON block FNAME (see section 3.3.1.2.2), with DOTS7 via passed parameters K1 and K2, and with CLUST7 via passed parameter RCNUM.

3.3.11.3 Input

There is no input to this subroutine.

3.3.11.4 Output

PRINT7 prints messages on the line printer.

3.3.11.5 Storage

This subroutine requires 783 words of storage.

3.3.11.6 Description

PRINT7 provides most line printer message output for the CCIT7 processor. This output provides a processing record for AA status and tracking activity. The routine uses system routines TIME and DATE to obtain data for header and trailer line printer messages for each run.

3.3.11.7 Flow Chart

The flow diagram for subroutine PRINT7 is given in figure 23.

3.3.11.8 Listing

The program listing for this subroutine is given figure 24.

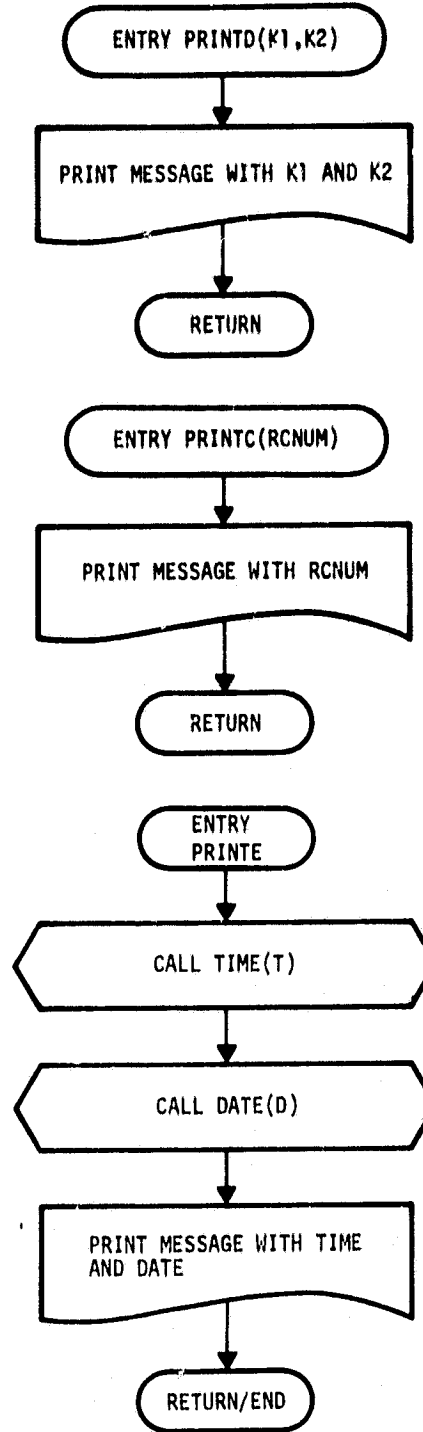
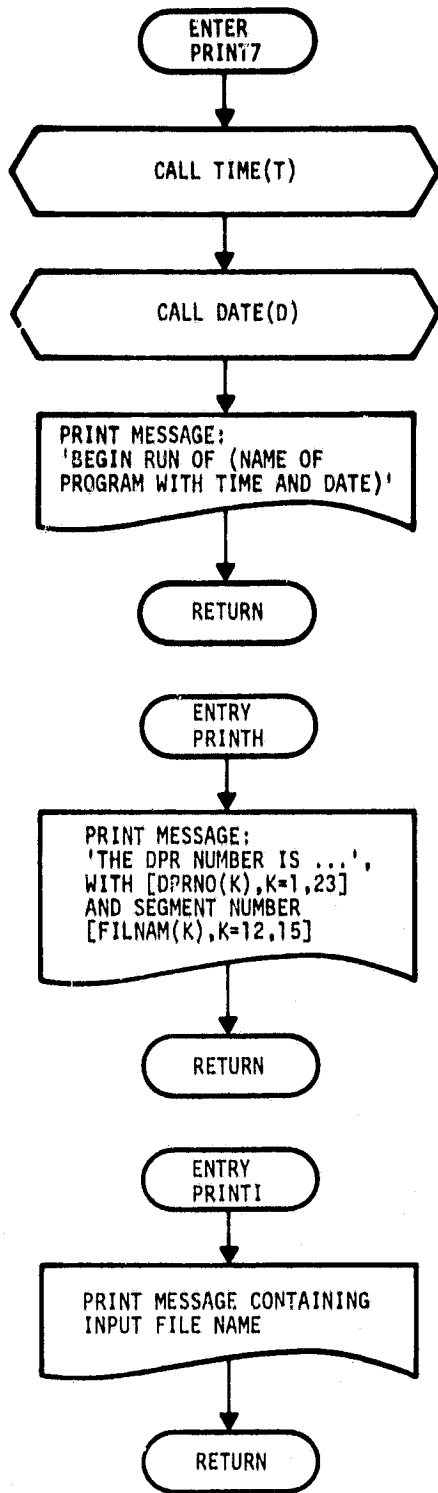


Figure 23.— Flow diagram for subroutine PRINT7.

```

0001 SUBROUTINE PRINT7
0002 IMPLICIT INTEGER(I,2)
0003 C MASTER SUBROUTINE - E PRINT HEADERS AND DATA OUTPUTS FOR CCIT
0004 C PROCESSOR PROGRAM - LACIES VERSION
0005 C BYTE A(120),FILNAM(24),DPRN(23),ACD1(4),ACD2(4),ACD3(4),
0006 C ACD4(4),SNGNUM(4),Y(8),D(6)
0007 C CMHNFNAME/FILNAM,SKIP
0008 C CMHNF/DPTS/DPRN,ACD1,ACD2,ACD3,ACD4,SNGNUM
0009 C CALL TIME(Y)
0010 C CALL DATE(D)
0011 PRINT 100,D,T
0012 FORMAT(1H,1,1) BEGIN RUN OF CCIT7 PROCESSOR PROGRAM!//,
0013 *10X,1H,1,9A1,1 AT '0A1)
0014 RETURN
0015 C CMHNF/FILNAM,K(1,23),(FILNAM(K),K(12,15)
0016 C PRINT 500,(DPRN(K),K(1,23))//,13H,10X,1EN FILE '1,20A3,0//)
0017 C FORMAT(1H,1, PROGRAM CCIT7',//,13H,10X,1EN FILE '1,20A3,0//)
0018 RETURN
0019 C ENTRY PRINTD(K1,K2)
0020 PRINT 700,K1,K2
0021 FORMAT(1H,10X,1 THERE HERE',13,1 TYPE 1, AND',
0022 *13,1 TYPE 2 DOTS ASSIGNED)
0023 RETURN
0024 C ENTRY PRINTC(RCNUM)
0025 PRINT 800,RCNUM
0026 FORMAT(1H,10X,1 PROCESSING',13,1 CLUSTERS FOR DOT WAYCNV)
0027 RETURN
0028 C ENTRY PRINTE
0029 CALL TIME(Y)
0030 CALL DATE(D)
0031 PRINT 200,D,T
0032 PRINT 200,D,T
0033 FORMAT(1H,1, END RUN OF CCIT7 PROCESSOR PROGRAM!//,10X,
0034 *18N,1,9A1,1 AT '1EA1)
0035 RETURN
0036 END

```

3-49
57

Figure 24.— Listing for subroutine PRINT7.

4. OPERATIONS

This section presents all the information necessary to obtain proper execution of the CCIT7 processor program.

4.1 OPERATORS GUIDE

This section explains the system hardware configuration and execution (run) setup for the CCIT7.

4.1.1 HARDWARE CONFIGURATION

The nominal configuration is the Earth Observations Division/Data Techniques Laboratory (EOD/DTL) PDP 11/45 processor with the RSX 11-D operating system. The system must have the input CCIT files resident on either the system disk or a user disk. The output files are written onto the same disk and under the same user identification code (UIC) as the resident input data. The input files are created using program AACCIT, described in JSC-13893. (See section 2 of this specification.)

4.1.2 PROGRAM EXECUTION

4.1.2.1 INTERACTIVE SETUP

- a. Edit file CCIT7.DAT for the proper file name and the value of parameter SKIP (24A1,I2). The file name takes the form:

DBX:[abc,d]SSSSYYDDD.wxy

where

X = Disk unit number

SSSSYYDDD = Input file name

wxy = Input file type; i.e., .CCO

[abc,d] = UIC for the input file

41
52

- b. Mount the proper disk pack on the drive.
- c. Type 'RUN CCIT7'.
- d. When message CCIT7-STOP appears on the monitor, collect a single-page report at the line printer, and check the listing to ensure that the ending message was printed and that the various steps were properly executed.

4.1.2.2 BATCH SETUP

- a. Prepare a batch run request detailing the disk configuration required.
- b. Set up a batch run deck as in table 2. The required steps follow:
 - Delete CCIT7.DAT.
 - Create CCIT7.DAT with card images, as given in section 4.1.2.1.
 - Run CCIT7.TSK.

4.2 USERS GUIDE

The CCIT7 program is designed to obtain a small fraction of the data from a CCIT disk file and to reformat these data into a form directly used by several AA software modules. This program will not execute correctly for CCIT's other than those created under LACIE version 7. The approximate dates of valid CCIT's for version 7 are 77305 through the date of issuance of this specification.

4.3 MAINTENANCE DOCUMENTATION

Not applicable.

TABLE 2.— BATCH RUN DECK SETUP

\$JOB/NAME=AA/MCR/LIMIT=99/ACCOUNT=1106

\$MCR PIP

CCIT7.DAT;*/DE

\$CREATE CCIT7.DAT

⋮

Card images for file name and SKIP parameter (24A1,I2)

⋮

Blank card

\$EOD

\$MCR REM RSXBAT

\$RUN CCIT7.TSK

\$EOJ

~~43~~
54

APPENDIX
FORMAT OF .CLO FILE

APPENDIX
FORMAT OF .CLO FILE

Record 1: The first record contains a single integer representing the number of classes detailed in the next record. Sixteen bytes of data follow for each class (minimum of 2 classes, maximum of 26 classes).

Record 2:

Byte number	Data description (ASCII)
1	Class 1 label (W, S, G, N, etc.)
2-6	Pixel population; PPPPP
7-9	Uncorrected proportion; M.MM (implied decimal point)
10-12	Corrected proportion; N.NN (implied decimal point)
13-16	Variance; .VVVV (implied decimal point)
17	Class 2 label (W, S, G, N, etc.)
18-22	Pixel population; PPPPP
23-25	Uncorrected proportion; M.MM (implied decimal point)
26-28	Corrected proportion; N.NN (implied decimal point)
29-32	Variance; .VVVV (implied decimal point)
⋮	⋮

Record 3: This record contains a single integer giving the number of clusters in the classification, CNUM.

Record 4: The fourth record contains 12 bytes of ASCII character data for each cluster; e.g., 12*CNUM bytes of data. The first 6 bytes of each group of 12 are the cluster label; e.g., NOCL17. The last six bytes of each group are the identity of the dot used to label the clusters; e.g., DOT103. Only type 1 dots are used to label clusters.

A2
51