

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



National Aeronautics and
Space Administration

NASA CR

160694

JSC-16487

JUN 10 1980

Lyndon B. Johnson Space Center
Houston Texas 77058

"AS-BUILT" DESIGN SPECIFICATION
OF THE
DATA ORDER PROCESSOR, ORDHDT

Job Order 76-662
(NASA-CR-160694) AS-BUILT DESIGN
SPECIFICATION OF THE DATA ORDER PROCESSOR
ORDHDT (Lockheed Engineering and Management)
41 p HC A03/MF A01

N80-26065

CSCL 09B

Unclas

G3/61 23936

Prepared By
Lockheed Engineering and Management Services Company, Inc.
Systems and Services Division
Houston, Texas

Contract NAS 9-15800

March 1980

LEMSCO- 14596

1. Report No. JSC- 16487	2. Government Accession No.	3. Recipient's Catalog No.	
4. Title and Subtitle "As-Built" Design Specification of the Data Order Processor, ORDHDT		5. Report Date March 1980	6. Performing Organization Code SF6
		8. Performing Organization Report No. LEMSCO- 14596	10. Work Unit No.
7. Author(s) A. Rios	9. Performing Organization Name and Address Lockheed Engineering & Management Services Company, Inc. Systems and Services Division Houston, Texas 77058		11. Contract or Grant No. JAS 9-15800
12. Sponsoring Agency Name and Address National Aeronautics and Space Administration Lyndon B. Johnson Space Center Houston, Texas 77058 (J. M. Sulester, Tech. Monitor) <i>SF6</i>			13. Type of Report and Period Covered "As-Built" Design
14. Sponsoring Agency Code			
15. Supplementary Notes <i>James</i>			
16. Abstract The data order processor, ORDHDT, creates a computer compatible tape containing the AgRISTARS requirements for LANDSAT data to be ordered from Goddard Space Flight Center.			
17. Key Words (Suggested by Author(s))		18. Distribution Statement	
19. Security Classif. (of this report) Unclassified	20. Security Classif. (of this page) Unclassified	21. No. of Pages 42	22. Price*

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

JSC-16487

"AS-BUILT DESIGN SPECIFICATION
OF THE
DATA ORDER PROCESSOR, ORDHDT

Job Order 76-662

Prepared By
A. Rios


Approved By

LEMSCO

JSC



J. I. Morrow, Supervisor
Software Section



J. M. Sulister, Technical
Monitor Systems & Facilities Branch

Prepared By
Lockheed Engineering and Management Services Company, Inc.
For

Earth Observations Division
Space and Life Sciences Directorate

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION
LYNDON B. JOHNSON SPACE CENTER
HOUSTON, TEXAS 77058

PRECEDING PAGE BLANK NOT FILMED

March 1980

LEMSCO-14596

CONTENTS

Section	Page
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>PROCESSOR DESCRIPTION</u>	3-2
3.2.1 LIMS	3-5
3.2.2 PROGRAM ORDFIX	3-8
3.2.3 SORT	3-9
3.2.4 MODULE ORDHDT	3-10
3.2.4.1 Program ORDHDT	3-12
3.2.4.2 Subroutine ORDINT	3-15
3.2.4.3 Subroutine ORDWRT	3-16
3.2.4.4 ORDCOM.FTN	3-16
4. OPERATIONS	4-1
4.1 <u>OPERATOR'S GUIDE</u>	4-1
4.1.2 PROGRAM EXECUTION	4-1
Appendix	
A. DATA ORDER TAPE FORMAT (EBCDIC)	A-1
B. DATA ORDER REPORT	B-1
C. PROGRAM LISTINGS	C-1

FIGURES

Figure	Page
3-1 Data flow of data order processor.	3-3
3-2 Flow diagram of program ORDHDT	3-13
3-3 Flow diagram of subroutine ORDWRT.	3-17

1. SCOPE

This document provides the description and subroutine documentation of the data order processor, ORDHDT. As part of the LANDSAT IMAGERY VERIFICATION AND EXTRACTION SYSTEM (LIVES), ORDHDT creates a computer compatible tape containing the AgRISTARS requirements for LANDSAT data to be ordered from Goddard Space Flight Center.

2. APPLICABLE DOCUMENTS

1. "As-Built" Design Specification of the LANDSAT IMAGERY VERIFICATION AND EXTRACTION SYSTEM, JSC-14634 (LEC-12904), December 1979.
2. TIRF 79-0034, LANDSAT Data Order Processor, October 1979.
3. Action Document 76-662-08, HDT/LIVES, dated November 9, 1979.

3. SYSTEM DESCRIPTION

The LIVES master data base (LMDB) contains information on LANDSAT scenes needed by the Earth Observations Division at NASA's Johnson Space Center. This information is used to order scenes from the Image Processing Facility of Goddard Space Flight Center. The needed scenes are furnished on high density tapes.

The order for LANDSAT scene data is sent to GSFC via a card image tape. ORDHDT creates the card image data order tape.

The data order tape consists of two record types, a header record and a set of path/row span records. Each record is in an 80 character, card image, EBCDIC format. A separate tape is generated for each LANDSAT. The format of the data order tape is shown in Appendix A.

Geographic areas are ordered in each record by path number, the starting and ending row number of a sequential set of rows, and start and stop dates for which acquisitions are required. Several LMDB Areas of Interest may be covered by a single path/row span order. If the start and end dates of different Areas of Interest are overlapping or are "essentially contiguous" (the end of one Area of Interest is within 18 days of the start of another), the orders will be combined in one record. If the end date of a request in the LMDB is prior to the date the data order tape is created, that request will not be included.

3.1 HARDWARE DESCRIPTION

The hardware requirements include the following peripherals in addition to the PDP 11/45:

- a. One tape unit
- b. Operator's console
- c. One disk unit
- d. Line Printer

3.2 PROCESSOR DESCRIPTION

The data order processor, ORDHDT, performs a series of operations in creating the data order tape. A batch input stream (BIS) file is used to sequence these operations as follows:

- a. LIMS is used to extract the necessary data from the LMDB and create file ORDHDT.DT1.
- b. Program ORDFIX edits file ORDHDT.DT1 and creates file ORDHDT.SOR.
- c. The system sort processor is used to sort file ORDHDT.SOR and create file ORDHDT.RP1.
- d. Program ORDHDT processes the data in file ORDHDT.RP1 to create the data order tape and to print the data order report.

The data flow for the data order processor is shown in Figure 3-1.

Two batch input stream files are used by the data order processor:

ORDHDT2.BIS is used to order LANDSAT 2 data.

ORDHDT3.BIS is used to order LANDSAT 3 data.

The two files are shown here:

ORDHDT2.BIS

```
$JOB/NAME=ORDHDT2/LIMIT=99/ACCOUNT=333 33/MCR
$MESSAGE START OF ORDHDT BATCH JOB
$MCR PIP UNITS.SAT=UNITS.OR2/UP
$MCR LIM
$MCR PIP UNITS.SAT=UNITS.STD/UP
$RUN ORDFIX.TSK
$SORT/SIZ:23 ORDHDT.SOR/INP ORDHDT.RP1/OUT ORDHDT.SPC/SPE
$RUN ORDHDT.TSK
$MESSAGE END OF ORDHDT BATCH JOB
$EOJ
```

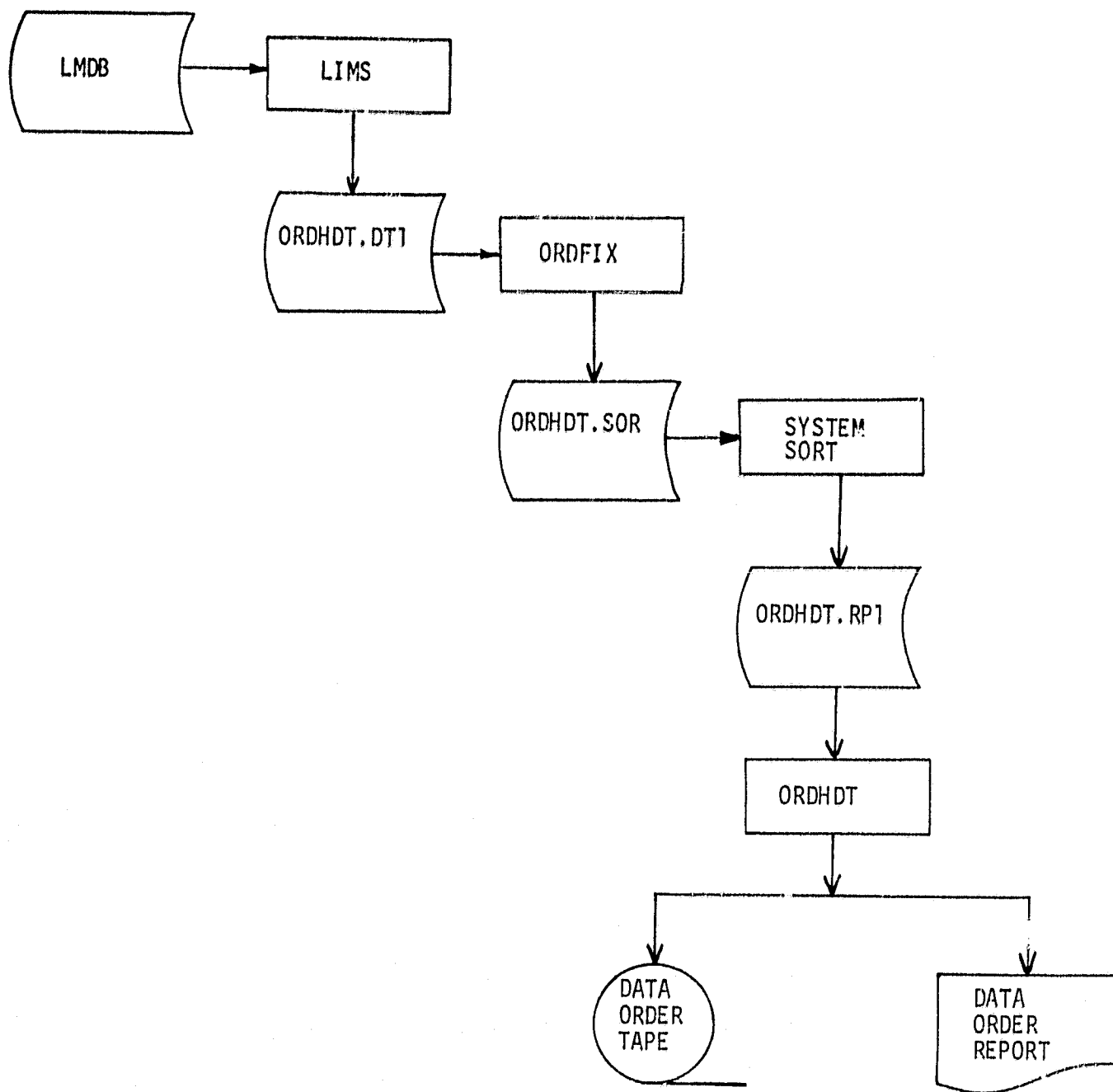


Figure 3-1 - Data flow of data order processor

ORDHDT3.BIS

\$JOB/NAME=ORDHDT3/LIMIT=99/ACCOUNT=333 33/MCR
\$MESSAGE START OF ORDHDT BATCH JOB
\$MCR PIP UNITS.SAT=UNITS.OR3/UP
\$MCR LIM
\$MCR PIP UNITS.SAT=UNITS.STD/UP
\$RUN ORDFIX.TSK
\$SORT/SIZ:23 ORDHDT.SOR/INP ORDHDT.RP1/OUT ORDHDT.SPC/SPE
\$RUN ORDHDT.TSK
\$MESSAGE END OFORDHDT BATCH JOB \$EOJ

3.2.1 LIMS

The LIMS function in the data order process is to extract the necessary data from the LMDB and create the file ORDHDT.DT1. The file is described here by character positions:

<u>Character Position</u>	<u>Content</u>
1-2	Blank
3-6	Area of Interest
7-12	Path/row
13	Blank
14-17	Start date
18	Blank
19-22	Stop date
23	LANDSAT No.

The last record in the file is an end-of-data record:

<u>Character Position</u>	<u>Content</u>
1	Blank
2	Asterisk(*)
3-9	Blank
10-12	999 Pseudo path number to sort last

LIMS uses one of two sets of UNITS and COMMAND files in performing its function. One set is for LANDSAT 2 data, the other for LANDSAT3. The names of these files are shown here:

LANDSAT 2

UNITS.OR2
ORDHDT.CM2

LANDSAT 3

UNITS.OR3
ORDHDT.CM3

Listings of these files follow:

UNITS.OR2

710111213
7DBO:ORDHDT.MSI
10LP:
11DBO:ORDHDT.DAT
12DBO:ORDHDT.DT1
13DBO:ORDHDT.CM2

ORDHDT.CM2

BEDBO:LMDB
ZZ1
SKRCTYPE1
SKPWRSRP
C01-2
DE 2
SKSWRSRP
C01-3
SN2,SATSEL.GT.99
SN4,SATSEL.GT.99
JT5,AA
RP5,CSO,BY:SATSEL,BY:USERID,BY:AOIID,' ',AOIID,PWRSRP,' ',ACQSRT,' 'ACQSTP,
'2'1
LAAA
JT6,BB
RP6,CSO,BY:SATSEL,BY:USERID,BY:AOIID,' ',AOIID,SWRSRP,' 'ACQSRT,' 'ACQSTP,
'2'1
LABB
HD1 * 999
EN

UNITS.OR3

710111213
7DBO:ORDHDT.MSI
10LP:
11DBO:ORDHDT.DAT
12DBO:ORDHDT.DT1
13DBO:ORDHDT.CM3

ORDHDT.CM3

BEDBO:LMDB

ZZ1

SKRCTYPE1

SKPWRSRP

C01-2

DE 2

SKSWRSRP

C01-3

SN2,SATSEL-(100*(SATSEL/100)),.GT.9

SN4,SATSEL-(100*(SATSEL/100)),.GT.9

JT5,AA

RP5,CSO,BY:SATSEL,BY:USERID,BY:AOIID,' 'AOIID,PWRSRP,' ',ACQSRT,' ',ACQSTP,
'3'

LAAA

JT6,BB

RP6,CSO,BY:SATEL,BY:USERID,BY:AOIID,' ',AOIID,SWRSPP,' ',ACQSRT,' ',ACQSTP,
'3'

LABB

HD1 * 999

EN

3.2.2 PROGRAM ORDFIX

- o Input

File ORDHDT.DT1

Current date

- o Output

File ORDHDT.SOR

- o Description

The purpose of ORDFIX is to arrange the input file by reversing the position of ROW/PATH and prefixing the year with a '7' or an '8' so the file can be sorted. Records with stop dates previous to current date are excluded.

- o Listing

The listing of ORDFIX is presented in Appendix C.

- o Restriction

Start or stop dates beyond 1988 will not be handled correctly.

3.2.3 SORT

- o Input

File ORDHDT.SOR

- o Output

File ORDHDT.RP1

- o Description

The system sort is used to sort the input file and to exclude records whose path number is zero.

The Specification File, ORDHDT.SPC, used in the sort is listed here;

```
HSORTR 16      X 23
O C 7 9EQC000
I
FNC 7 22
FDC 1 23
```

3.2.4 MODULE ORDHDT

This module constitutes the primary section of the data order processor. The module includes one main program, ORDHDT, two subroutines, ORDINT and ORDWRT, and one INCLUDE file, ORDCOM.FTN. Two utility routines are used: JULIAN - convert day, month year to Julian day; ATEB - convert ASCII to EBCDIC.

o Input

File ORDHDT.RP1

o Output

Data order tape (See Appendix A)

Data order report (See Appendix B)

o Common Block

One Common Block is used by the three routines

/ORD/

STORE - Data hold array to hold data to be output to tape.
REPT - Data hold array to hold data for print report
PATHRO - Path/row
SSDAY - Hold array for start/stop windows
SITE - Area of interest or site number.
OUTBUF - Tape output buffer.
START - Acquisition start date (YDDD).
STOP - Acquisition stop date (YDDD).
NREC - Current number of entries in array store.
TREC - Tape record counter.
PCOUNT - Current number of entries in array REPT.
DAYS - Array of cumulative days per year since 1 Jan 1979.
LAST - Pointer for array REPT to a corresponding row in array STORE.
LSAT - LANDSAT number.

o Hold arrays

Three hold arrays are used to hold input record data that may be combined with other record data: These arrays are described below:

STORE (N,1) Start row

STORE (N,2) Stop row

STORE (N,3) Start day

STORE (N,4) Stop day

SSDAY (N,1) Start day - 18 converted to number of days from 1 Jan 1979.

SSDAY (N,2) Stop day + 18, also converted.

REPT (M,1,1) Area of Interest or site number

REPT (M,2,1) Path number

REPT (M,3,1) Row number

REPT (M,4,1) Start day

REPT (M,5,1) Stop day

REPT (M,6,1) Corresponding row number in array STORE or 'N'.

o Module description

When the first record is read, the three hold arrays are initialized. As each succeeding record is read, an attempt is made to combine it with a stored record. If it cannot be combined, it is added to the hold arrays if possible. If the row number (PATH/ROW) of the Area of Interest is not the same or contiguous to one already in the hold arrays, the record in array STORE will be output to tape and the corresponding records in array REPT will be output to the printer. The hold arrays contain no gaps. Thus the record number output will be filled by pushing up each succeeding data row in each hold array.

If the hold arrays are emptied such as when the current record read indicates a change of PATH, the current record is used to reinitialize the hold arrays as if it were the first record.

When the end-of-data record is read, the hold arrays are emptied by outputting to tape and printer, an EOF mark is written on the tape and the program is terminated.

3.2.4.1 Program ORDHDT

This is the main program of Module ORDHDT.

o Input/Output

Communication with other routines is maintained through Common Block /ORD/ described earlier.

o Description

ORDHDT is the driver routine. The input data file is read and the data is stored in the hold arrays. An attempt is made to combine each input record with one of the entries stored in array STORE. If a new ROW is the same or contiguous to one stored, the new record will be combined if the START or STOP dates fall within a stored START minus 18 days or a stored STOP plus 18 days. Since the rows in the data file have been sorted in ascending order, any new row that is not the same or contiguous to a row in array STORE signals that that entry in array STORE cannot be combined with any future records and therefore should be output to tape.

o Flow

The flow diagram of program ORDHDT is presented in Figure 3-2.

o Listing

The listing of ORDHDT is presented in Appendix C.

o Restriction

Because of preset data in array DAYS:

Area of Interest START date must fall between 1 Jan 1979 and 31 Dec 1988.

Area of Interest STOP date must fall between current date and 31 Dec 1988.

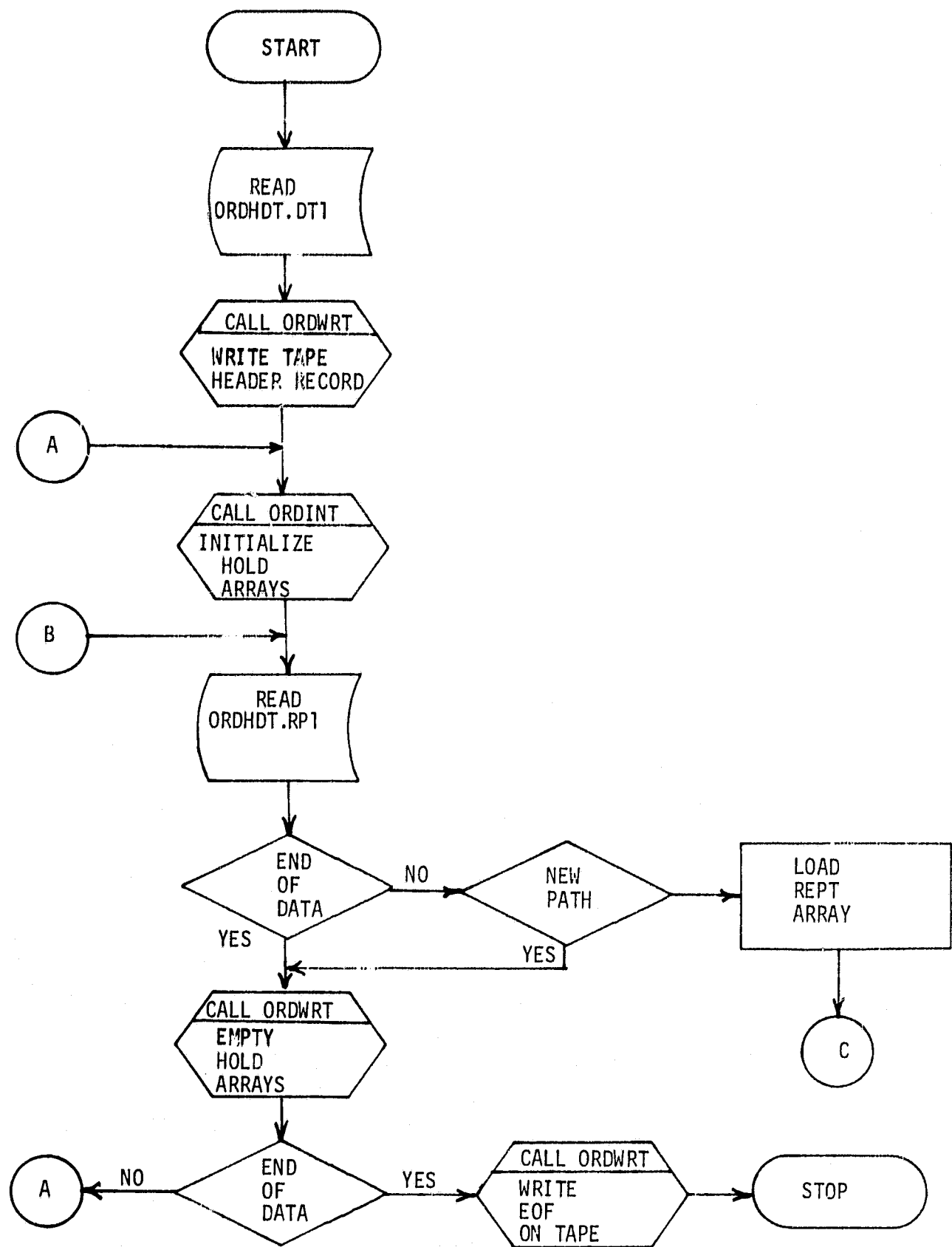


Figure 3-2 - Flow diagram of program ORDHDT

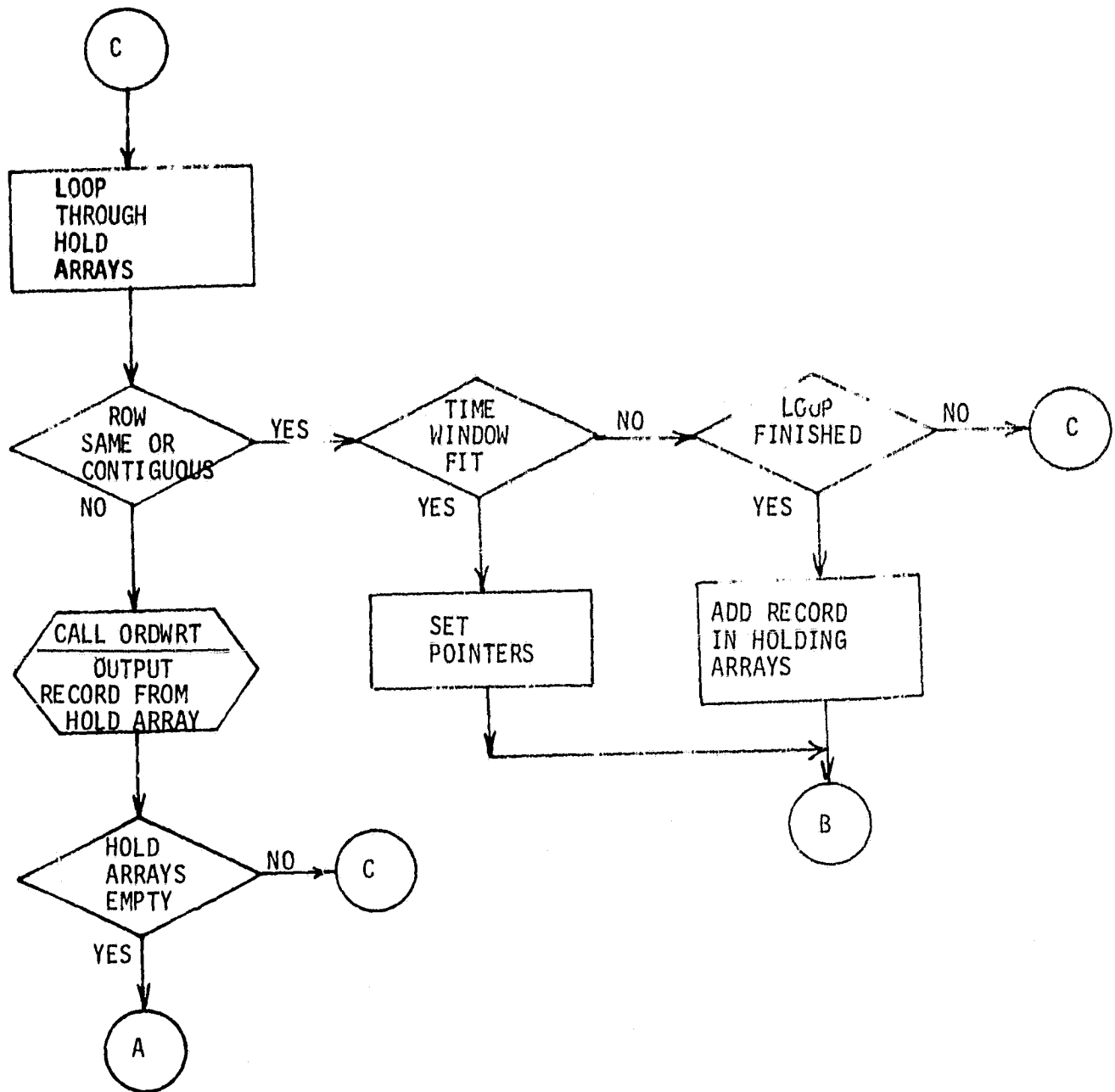


Figure 3-2 - Concluded

3.2.4.2 Subroutine ORDINT

The purpose of this subroutine is to make the first entry in hold arrays 'STORE', 'REPT', and 'SSDAY'. This occurs when the first record is read and each time the hold arrays are emptied by outputting the records in the hold arrays to tape.

Also, the path number is entered in the output buffer and converted from ASCII to EBCDIC.

o Listing

The listing of ORDINT is presented in Appendix C.

3.2.4.3 Subroutine ORDWRT

The purpose of this subroutine is to write the data order tape and print the data order report.

o Input/output

Communication between routines is handled through Common Block /ORD/ and through calling arguments: SUBROUTINE ORDWRT (IFLAG, INDEX)

IFLAG - If = 1, write header record.

 If = 2, write data record.

 If = 3, write EOF on tape.

INDEX - Pointer to row number in array STORE.

o Description

The first time this routine is called, it prints the heading for the data order report and the header record on the data order tape. On subsequent calls when IFLAG is set to 2, the row of data in array STORE selected by pointer INDEX is loaded in the output buffer and written on tape. The data in the report array REPT pertaining to that tape record is output to the printer. Any gaps left in the hold arrays by outputting data are filled by pushing up the stack in each array. If ORDWRT is called with IFLAG set to 3, an EOF mark is written on the output tape and the tape is rewound.

o Flow

The flow diagram of subroutine ORDWRT is presented in Figure 3-3.

o Listing

The listing of ORDWRT is presented in Appendix C.

3.2.4.4 ORDCOM.FTN

This file is used as an INCLUDE file and is listed in Appendix C.

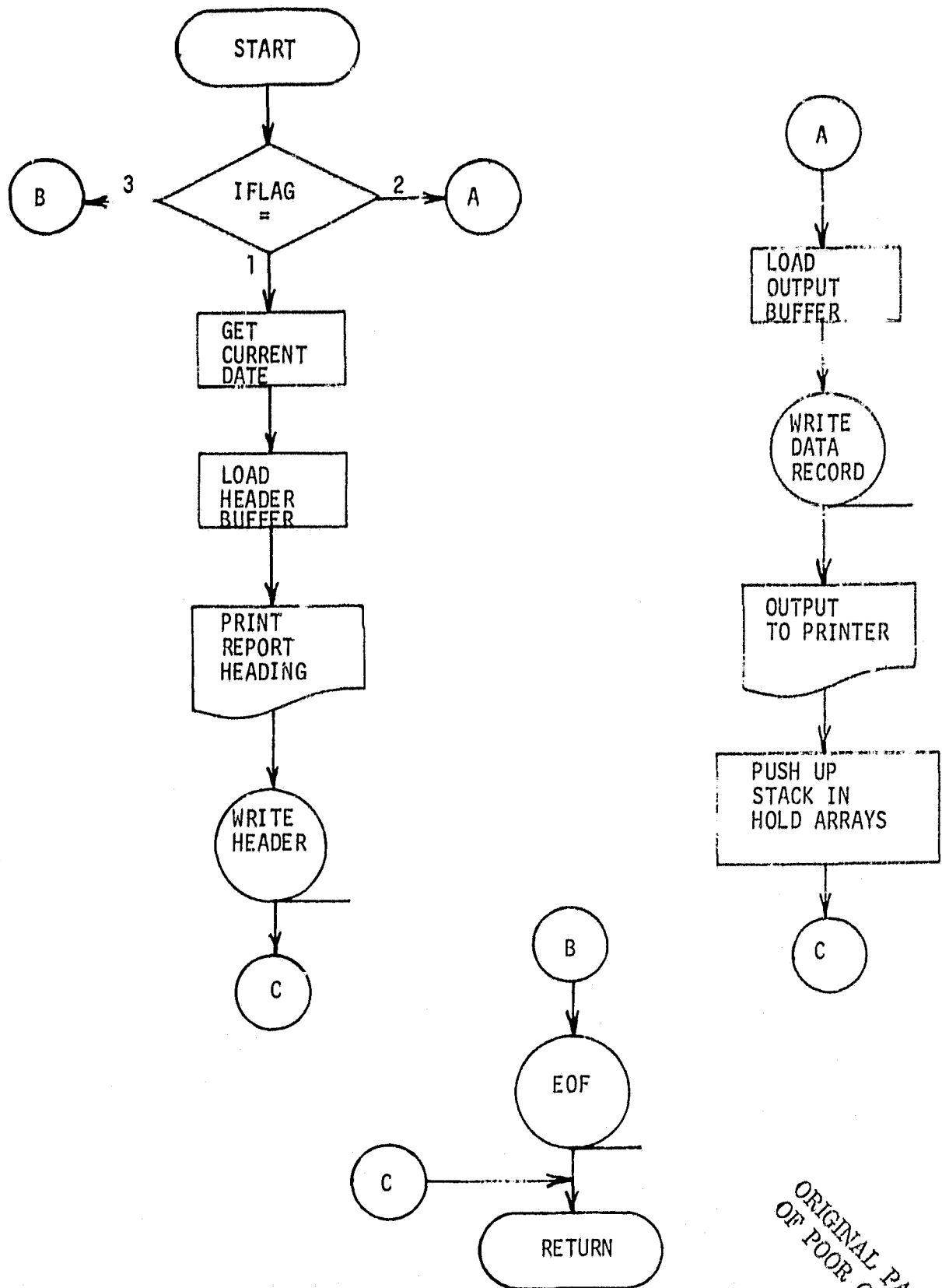


Figure 3-3 - Flow diagram of subroutine ORDWRT

ORIGINAL PAGE IS
OF POOR QUALITY

4. OPERATIONS

The section presents the information necessary to run the data order processor, ORDHDT.

4.1 OPERATOR'S GUIDE

This paragraph describes the system hardware configuration and the run setup for the data order processor.

4.1.1 HARDWARE CONFIGURATION

- a. PDP 11/45 Support processor.
- b. One tape drive.
- c. Line printer.
- d. Console.

4.1.2 PROGRAM EXECUTION

A request to run the data order processor must specify the LANDSAT number for which data is desired: LANDSAT 2 or LANDSAT 3. If both are requested, a separate tape must be created for each LANDSAT number. A LIVES Master Data Base (LMDB) must reside in the UIC from which the processor is to be run.

The following files must also reside in the user's UIC:

```
ORDHDT2.BIS
UNITS.OR2
ORDHDT.CM2
ORDFIX.TSK
ORDHDT.TSK
ORDHDT3.BIS
UNITS.OR3
ORDHDT.CM3
```

The following items should be performed:

1. Mount output tape on MTO, foreign.
2. Sign-on user's UIC (Normally [333,33]).

3. If the files listed above do not reside in user's UIC, enter:
MCR>PIP @[333,33]UPDORD (CR).
4. For LANDSAT 2 data, enter: MCR>BAT ORDHDT2.BIS\$.
5. For LANDSAT 3 data, enter: MCR>BAT ORDHDT3.BIS\$.
6. Check for console message: END OF ORDHDT BATCH JOB.
7. Label tape according to user instructions and save.
8. Three data files are created during each run and should be deleted:
ORDHDT.DT1, ORDHDT.SOR, ORDHDT.RP1.
9. After the data processor has been run, the following PIP commands will delete the proper files including those listed in item 8:

```
PIP   ORDHDT.*;*/DE
PIP   ORDHDT2.BIS;*/DE
PIP   ORDHDT3.BIS;*/DE
PIP   UNITS.OR2;*/DE
PIP   UNITS.OR3;*/DE
PIP   ORDFIX.TSK;*/DE
```

APPENDIX A

A.1 Data order tape format (EBCDIC)

Record 1 - ID Card Record

<u>Card Col. Number</u>	<u>Format</u>	<u>Description</u>
1 - 9	JSCFCPFSR	JSC identification
10	2/3	Spacecraft & ID 2 = LANDSAT 2, 3 = LANDSAT 3
11	b	Blank
12 - 19	ydddhhmm	Date of generation
20 - 80	b	Blank

Record 2-N. Path/Row Spans (length = 80 characters)

<u>Card Col. Number</u>	<u>Format</u>	<u>Description</u>
1 - 9	JSCFCPFSR	JSC identification
10 - 13	yddd	Date of generation
14	K	Special for JSC
15	0(zero)	OCC defined fixed value
16	.(period)	" " " "
17	B	" " " "
18	b	Blank
19 - 20	10	Value for JSC
21	b	Blank
22 - 24	001-251	WRS path number
25	b	Blank
26 - 28	001-248	WRS starting row number
29	b	Blank
30 - 32	001-248	WRS ending row number
33	b	Blank
35 - 36	20	OCC user priority code number
37	F	OCC defined fixed value

Record 2-N. Path/Row Spans (length = 80 characters)

<u>Card Col. Number</u>	<u>Format</u>	<u>Description</u>
38	A	A = weather category 3 (71-100%)
39	b	Blank
40	.	. = No RBV requested
41	D	D = MSS daylight requested
42	.	OCC defined fixed value
43-44	10	Minimum sun angle
45	.	MSS gain . = low
46	C	MSS mode C = compressed
47	b	Blank
48 - 51	yddd	On-time year and day of year
52	b	Blank
53 - 56	yddd	Off-time year and day of year
57 - 80	b	Blank

APPENDIX B

B.1 Data order report

AGRISTARS REQUEST FOR LANDSAT 2 DATA
 TAPE GENERATED ON 14-FEB 80 AT 13:34:00

RECORD NUMBER FROM TAPE	A OF I	PATH/ROW	START	STOP
RECORD 1				
	0322	015032	9060	0146
	0321	015032	9060	0146
	0320	015032	9060	0146
	0318	015033	9060	0146
	0317	015033	9060	0146
	0314	015033	9060	0146
	0313	015033	9060	0146
RECORD 2				
	0332	015035	9060	0146
RECORD N	0947	049028	9060	0146

END OF REPORT

Appendix C

```
C
C THE PURPOSE OF THIS PROGRAM IS TO FIX THE INPUT FILE FOR
C PROGRAM TORRDOT BY REVERSING THE POSITION OF ROW/PATH AND
C PREFIXING THE YEAR WITH A 171 OR AN 181 SO THAT THE FILE
C CAN BE SORTED. RECORDS WITH START DATES PREVIOUS TO CURRENT
C DATE ARE IGNORED.
C
0001 C IMPLICIT INTEGER(A-Z)
0002 C BYTE Y1(5), Y2(5), STAR, LSAT, ROW(5), PATH(5)
0003 C INTEGER*4 SITE, LDAY, NDAY, MM
C
0004 C OPEN(UNIT=1, NAME='TORRDOT.J11', TYPE='TEXT')
0005 C OPEN(UNIT=2, NAME='TORRDOT.S01', TYPE='TEXT')
C
0006 C CALL IDATE(MM, IDAY, IYR )
0007 C JDAY = JULIAN( MM, IDAY, IYR )
0008 C MM = 1000
0009 C IDAY = IYR * MM + JDAY
C
0010 C 20 CONTINUE
C
0011 C READ(1,40) STAR, SITE, ROW, PATH, Y1, Y2, LSAT
0012 C 40 FORMAT(1X,A1, A4, 6A1, 5A1, 5A1, A1)
C
0013 C IF( STAR .EQ. '1*' ) GO TO 100
C
0014 C Y1(1) = 181
0015 C Y2(1) = 181
0016 C IF( Y1(2) .EQ. 191 ) Y1(1) = 171
0017 C IF( Y2(2) .EQ. 191 ) GO TO 20
C
0018 C DECODE( 5, 60, Y2 ) NDAY
0019 C 60 FORMAT( 15 )
C
0020 C DO 80 I=2,4
0021 C IF(Y1(I) .EQ. 111) Y1(I) = 101
0022 C IF(Y2(I) .EQ. 111) Y2(I) = 101
0023 C 80 CONTINUE
C
0024 C IF( NDAY .LT. LDAY ) GO TO 20
C
0025 C 100 CONTINUE
C
0026 C WRITE(2,40) STAR, SITE, PATH, ROW, Y1, Y2, LSAT
C
0027 C IF(STAR .NE. '1*') GO TO 20
C
0028 C STOP
0029 C END
```



```
C  
C      MAIN PROGRAM = ORDHDT  
C  
C      THE PURPOSE OF THIS PROGRAM IS TO COMBINE THE DATA REQUESTS  
C      IN THE LIVES LMDB DATA BASE AND CREATE A CARD IMAGE TAPE  
C      TO ORDER THE PATH/ROW COMBINATIONS OF THE MDL.  
C  
0001      INCLUDE 'ORDHDT.M.FTN'  
0002 *      IMPLICIT INTEGER (A-Z)  
0003 *      PARAMETER (LEN1=50, LEN2=100)  
C  
0004 *      INTEGER*4  SDOFF(LEN,4), REPT(LEN2,4,2)  
C      *      *      SITE, APATH, AROW, START, STOP  
C  
0005 *      INTEGER*2  SDOAY(LEN,2),  DAYS(4)  
C  
0006 *      BYTE      PATHRO(8),  YR1(4), YR2(4),  OUTSP(80),  LSAT  
C  
0007 *      COMMON / ORD /  SDOFF, REPT,  PATHRO, SDOAY, SITE,  OUTSP  
C      *      *      START, STOP,  NREC,  TREL,  PCOUNT,  DAYS,  LAST,  LSAT  
C  
0008 *      EQUIVALENCE ( APATH,  PATHRO ( )  
C      *      *      *      AROW,  PATHRO(5) )  
C      *      *      *      YR1,  START ( )  
C      *      *      *      YR  STOP ( )  
C  
0009      INTEGER*4  PATH  
0010      BYTE  STAR  
0011      DATA PATHRO / 8, 1 /,  TREL / 0 /  
C  
0012      DATA DAYS / 165, 131, 1096, 1461, 1826, 2192, 2557,  
C      *      *      *      2922, 3287 /  
C  
0013      OPEN( UNIT=1, NAME='ORDHDT.RP1', TYPE='OLD' )  
C  
C      READ FIRST DATA RECORD  
C  
0014      READ(1,20) STAR, SITE, (PATHRO(I), I=2,7), START, STOP, LSAT  
0015 20 FORMAT(1X,A1,A1,5A1,1X,A4,1X,A4,A1)  
C  
0016      PATH= APATH  
0017      IFLAG = 1  
0018      CALL ORDWRT( IFLAG, 0 )  
C  
0019      50 CONTINUE  
C  
C      INITIALIZE HOLDING ARRAYS  
C  
0020      CALL ORDINT  
C  
0021      60 CONTINUE  
C  
0022      80 CONTINUE  
0023      READ(1,20, ENH=85) STAR, SITE, (PATHRO(I), I=2,7), START, STOP, LSAT  
C
```

```
0024      IF(STAR .EQ. 1#1) GO TO 85
0025      IF(PATH .EQ. APATH) GO TO 100
0026      PATH = APATH
0027      GO TO 90
0028      85 CONTINUE
0029      EOF = 1
0030      90 CONTINUE
C
C CLEAR ALL HOLDING ARRAYS
C
0031      IFLAG = 2
0032      INDEX = 1
0033      95 CONTINUE
0034      CALL ORDWRIT(IFLAG, INDEX)
0035      IF( NREC .GT. 0 ) GO TO 95
C
0036      IF( EOF ) 50, 50, 260
C
C
0037      100 CONTINUE
0038      PCOUNT = PCOUNT + 1
0039      REPT(PCOUNT,1,1) = SITE
0040      REPT(PCOUNT,2,1) = APATH
0041      REPT(PCOUNT,3,1) = ARROW
0042      REPT(PCOUNT,4,1) = START
0043      REPT(PCOUNT,5,1) = STOP
C
0044      110 CONTINUE
C
0045      DECODE( 3, 1#03, ARROW) IROW
0046      1003 FORMAT( I3 )
C
0047      120 CONTINUE
C
C LOOP THRU HOLDING ARRAYS TO SEE IF CURRENT RECORD CAN BE
C COMPINED WITH PREVIOUS RECORDS.
C
0048      NROW = NREC
0049      DO 240 I=1,NROW
0050      INDEX = I
0051      L = STORF(I,5) + 1
0052      IF( IROW .LE. L) GO TO 140
C
C CURRENT ROW NUMBER NOT SAME OR ADJACENT.
C TIME TO OUTPUT TAPE RECORD.
C
0053      CALL ORDWRIT( 2, INDEX )
0054      IF( NREC ) 50, 50, 120
C
0055      140 CONTINUE
0056      Y1 = YR1(1) : 4#
0057      Y2 = YR2(1) : 4#
0058      DECODE( 3, 1#03, YR1(2)) ISTART
0059      DECODE( 3, 1#03, YR2(2)) ISTOP
C
C CONVERT START AND STOP TIMES TO NUMBER OF DAYS FROM 1 JAN 79
```

ORIGINAL PAGE IS
OF POOR QUALITY

```

C
0060      IF(Y1 .EQ. 9) GO TO 160
0061      ISUR = Y1 + 1
0062      ISTART = ISTART + DAYS(ISUR)
0063 160 CONTINUE
0064      IF(Y2 .EQ. 9) GO TO 180
0065      ISUR = Y2 + 1
0066      ISTOP = ISTOP + DAYS(ISUR)
0067 180 CONTINUE
0068      IF(ISTART .GT. SSDAY(I,2)) GO TO 240
0069      IF(ISTOP .LT. SSDAY(I,1)) GO TO 240
0070      IF(SSDAY(I,1) .LE. (ISTART-18)) GO TO 200
0071      SSDAY(I,1) = ISTART - 18
0072      STORF(I,3) = ISTART
0073 200 CONTINUE
0074      IF(SSDAY(I,2) .GE. (ISTOP+18)) GO TO 220
0075      SSDAY(I,2) = ISTOP + 18
0076      STORF(I,4) = ISTOP
0077 220 CONTINUE
0078      IF(IROW .LE. STORF(I,2)) GO TO 230
0079      STORF(I,2) = IROW
0080 230 CONTINUE
0081      REPT(PCOUNT,4,1) = REPT(INDEX,6,1)
0082      GO TO 80
0083 240 CONTINUE

```

C NO FIT - ADD CURRENT RECORD TO HOLDING ARRAY

```

C
0084      NREC = NREC + 1
0085      LAST = LAST + 1
0086      STORF(NREC,1) = IROW
0087      STORF(NREC,2) = IRUW
0088      STORF(NREC,3) = ISTART
0089      STORF(NREC,4) = ISTOP
0090      REPT(PCOUNT,4,1) = LAST
0091      SSDAY(NREC,1) = ISTART - 18
0092      SSDAY(NREC,2) = ISTOP + 18
0093      GO TO 80

```

C 260 CONTINUE

```

C
0095      IFLAG = 3
0096      CALL ORDWRIT(7,IFLAG, INDEX)

```

```

C
0097      WRITE(6,280)
0098 280 FORMAT(////// 'END OF REPORT' )
0099      STOP
0100      END

```

```
0001 SUBROUTINE ORDINT
C
C
C THE PURPOSE OF THIS SUBROUTINE IS TO STORE THE INITIAL ENTRY
C INTO TEMPORARY ARRAYS 'STORE', 'REPT', AND 'SSDAY'.
C
0002 INCLUDE 'ORDINT.FT'
0003 * IMPLICIT INTEGER (A-Z)
0004 * PARAMETER (LA=50, LEN2=10)
C
0005 * INTEGER*4 STORE(LEN2,3), REPT(LEN2,6,2)
* * SITE, APATH, ANOW, START, STOP
C
0006 * INTEGER*2 SSDAY(LEN2,2), DAYS(9)
C
0007 * BYTE PATHR(8), YR1(4), YR2(4), LUTRUF(60), LAST
C
0008 * COMMON / ORD / STORE, REPT, PATHR, SSDAY, SITE, QUINUF
* * START, STOP, NPEC, TRFC, PCOUNT, DAYS, LAST, LAST
C
0009 * EQUIVALENCE ( APATH, PATHR(1) )
* * ( ANOW, PATHR(5) )
* * ( YR1, START )
* * ( YR2, STOP )
C
0010 NPEC = 1
0011 LAST = TRFC + 1
C
C LOAD FIRST DATA SET TO ARRAYS 'STORE' AND 'SSDAY'.
C
0012 DECODE(3, 1003, PATHR(5)) STORE(1,1)
0013 1003 FORMAT( I3)
0014 STORE(1,2) = STORE(1,1)
0015 STORE(1,3) = START
0016 STORE(1,4) = STOP
0017 DECODE( 3, 1003, YR1(2)) SSDAY(1,1)
0018 DECODE( 3, 1003, YR2(2)) SSDAY(1,2)
0019 SSDAY(1,1) = SSDAY(1,1) - 18
0020 SSDAY(1,2) = SSDAY(1,2) + 18
0021 Y1 = YR1(1) - 48
0022 Y2 = YR2(1) - 48
C
0023 IF(Y1 .EQ. 9) GO TO 40
0024 ISUB = Y1 + 1
0025 SSDAY(1,1) = SSDAY(1,1) + DAYS(ISUB)
0026 40 CONTINUE
0027 IF(Y2 .EQ. 9) GO TO 60
0028 ISUB = Y2 + 1
0029 SSDAY(1,2) = SSDAY(1,2) + DAYS(ISUB)
0030 60 CONTINUE
C
C STORE ENTRIES IN REPORT ARRAY.
C
0031 PCOUNT = 1
0032 REPT(PCOUNT,1,1) = SITE
```

ORIGINAL PAGE IS
OF POOR QUALITY

FORTRAN IV-PLUS
ORDINT.FT

V02-51
/TR;BLOCKS/AD

09:21:14

00-AR-60

PAGE 2

```
0033      REPT(PCOUNT,2,1) = APATH
0034      REPT(PCOUNT,3,1) = ARUN
0035      REPT(PCOUNT,4,1) = START
0036      REPT(PCOUNT,5,1) = STOP
0037      REPT(PCOUNT,6,1) = LAST

C
C  LOAD PATH NUMBER IN DATA OUTPUT BUFFER.
C
0038      OUTBUF(22) = PATHNO(2)
0039      OUTBUF(23) = PATHNO(3)
0040      OUTBUF(24) = PATHNO(4)
0041      CALL ATEP( OUTBUF(22), OUTBUF(23), 3 )

C
0042      RETURN
0043      END
```

Subroutine ORDINT

C-7

```

0001 SUBROUTINE ORDWRT( IFLAG, INPLY )
      C
      C THE PURPOSE OF THIS SUBROUTINE IS TO OUTPUT THE CARD IMAGE INFO
      C TO PLACE AN ORDER(REQUEST) FOR PATH/RW CALIBRATION OF A MDT.
      C THE DATA IN DYNAMIC ARRAYS (STORE1, REPT1, AND SDDAY) IS
      C REARRANGED WHEN A ROW OF DATA IS OUTPUT TO TAPE.
      C
0002 INCLUDE TORPDM.FTN
0003 * IMPLICIT INTEGER (A-Z)
0004 * PARAMETER LEN=50, LEN2=100
      C
0005 * INTEGER*4 STORE(LEN,4), REPT(LEN2,6,2)
      C * SITE, APATH, ARCW, START, STOP
      C
0006 * INTEGER*2 SDDAY(LEN,2), DAYS(4)
      C
0007 * BYTE PATHR(6), YR1(4), YR2(4), OUTBUF(80), LSAI
      C
0008 * COMMON / ORD / STORE, REPT, PATHR, SDDAY, SITE, OUTBUF
      C * START, STOP, INREQ, REFL, POINT, DAYS, LAST, LSAT
      C
0009 * EQUIVALENCE ( APATH, PATHR(1) )
      C *
      C **
      C *
      C **
      C *
      C
0010 INTEGER*4 HUF(20)
0011 INTEGER*2 IDEF(6), S(2)
0012 BYTE HEDHUF(60), DATEX(4), TIMEA(8)
      C
0013 EQUIVALENCE( OUTBUF, HUF )
      C
0014 DATA MT / INT /
0015 DATA IATCH / "1000 " /, IRWHD / "2400 /
0016 DATA IWLW / "0000 " /, IWFUF / "3000 /
      C
0017 DATA HEDHUF / J1, I1, I1, I1, I1, I1, I1, I1, I1, I1, I1, I1, I1, I1, I1, I1,
      C **
      C 11 * "40, 60 * "100 /
      C
0018 DATA OUTBUF / J1, I1, I1, I1, I1, I1, I1, I1, I1, I1, I1, I1, I1, I1, I1, I1,
      C **
      C 4 * I1, I1, I1, I1, I1, I1, I1, I1, I1, I1, I1, I1, I1, I1, I1, I1
      C **
      C 13 * I1, I1, I1, I1, I1, I1, I1, I1, I1, I1, I1, I1, I1, I1, I1, I1,
      C **
      C 101, I1, I1, I1, 34 * I1 /
      C
0019 DATA ZERO / 0 /
      C
0020 IF( IFLAG .GT. 1) GO TO 200
0021 CALL ASNLUN( 2, MT, ZERO )
0022 CALL WTQI( IATCH, 2, 1., S )
0023 CALL WTQI( IRWHD, 2, 1., S )
      C
      C WRITE TAPE HEADER RECORD.
      C
0024 HEDBUF(10) = LSAI
0025 CALL IDATE( MO, DAY, YR )

```

ORIGINAL PAGE IS
OF POOR QUALITY

/TRIBLOCKS/NO

```
0026      LYR = IADDF( YR, 10 )
0027      ENCODE( 1, IAD1, MEMBUF(12)) LYR
0028      1001 FORMAT( 1 )
0029      JDAY = JUTIAN(MO, DAY, YR)
0030      ENCODE( 3, IAD3, MEMBUF(13)) JDAY
0031      1003 FORMAT( 13 )
C
0032      IF( MEMBUF(13) .EQ. 1 ) MEMBUF(15) = 101
0033      IF( MEMBUF(14) .EQ. 1 ) MEMBUF(16) = 101
C
0034      OUTBUF(10) = MEMBUF(12)
0035      OUTBUF(11) = MEMBUF(13)
0036      OUTBUF(12) = MEMBUF(14)
0037      OUTBUF(13) = MEMBUF(15)
C
0038      CALL DATE( DATEX )
0039      CALL TIME( TIMEX )
0040      MEMBUF(16) = TIMEX(1)
0041      MEMBUF(17) = TIMEX(2)
0042      MEMBUF(18) = TIMEX(4)
0043      MEMBUF(19) = TIMEX(5)
C
0044      *WRITE(6,1020) I SAT, DATEX, TIMEX
0045      1020 FORMAT(1H, //, EX, IAGHISTARS REQUEST FOR LANDSAT 1, A1, I DATE //
      ., 4X, I TAPE GENERATED ON 1, 9A1, I AT 1, 6A1 // //
      ., 1X, I RECORD NUMBER FROM TAPE, 7X, I OF 11, 7X, I PATH/ROW I
      ., 7X, I START, 6X, I STOP // )
C
0046      CALL ATER( MEMBUF, MEMBUF, 201 )
0047      CALL GETADR( IPRM, MEMBUF )
0048      IPRM(2) = 80
0049      CALL WTGTO( IMLB, 2, 1, 5, IPRM )
0050      IF( S(1) .EQ. 1 ) GO TO 60
0051      STAT = S(1) - 250
0052      WRITE(6,1040) STAT, IREC
0053      1040 FORMAT( // ** WRITE ERROR ON OUTPUT TAPE ** I /
      ., I STATUS = 1, 15, I TAPE RECORD NO. = 1, 15 // )
C
0054      STOP I ** ORDWRT ABORTED ** I
C
0055      60 CONTINUE
C
0056      CALL ATER( OUTBUF, OUTBUF, 80 )
0057      GO TO 500
C
0058      200 CONTINUE
C
0059      IF( IFLAG .GT. 2 ) GO TO 360
C
C
C
C
C
0060      ENCODE( 3, IAD3, OUTBUF(26)) STORE( INDEX, 1 )
0061      ENCODE( 3, IAD3, OUTBUF(30)) STORE( INDEX, 2 )
0062      BUF(13) = STORE( INDEX, 3 )
```

```

C
0063 IF(OUTBUF(26) .EQ. 1 1) OUTBUF(26) = 101
0064 IF(OUTBUF(27) .EQ. 1 1) OUTBUF(27) = 101
0065 IF(OUTBUF(30) .EQ. 1 1) OUTBUF(30) = 101
0066 IF(OUTBUF(31) .EQ. 1 1) OUTBUF(31) = 101

C
0067 DO 220 I=20,21
0068 OUTBUF(I) = OUTBUF(I+1)
0069 220 CONTINUE

C
0070 OUTBUF(29) = 1 1
0071 OUTBUF(52) = 1 1
0072 RUP(10) = STAGE(INDEX,4)
0073 CALL ATER(OUTBUF(26), OUTBUF(20), 7 1)
0074 CALL ATER(OUTBUF(30), OUTBUF(48), 9 1)

C
0075 TRFC = TRFC 1 1

C
0076 CALL GETADR( IPRT, OUTBUF 1)
0077 IPRT(2) = 20
0078 CALL STNDC( I=LN, 2, 1, 5, IPRT 1)
0079 IF( S(1) .EQ. 1 1) GO TO 240
0080 STAT = S(1) 1 250
0081 WRITE(6,1040) STAT, TRFC

C
0082 STOP 1 ** REPORT ABORTED ** 1

C
0083 240 CONTINUE

C
0084 NROW = NREC 1
0085 NREC = NREC 1
0086 WRITE(6,1050) TRFC
0087 1050 FORMAT(/ RECORDED 1,15)
0088 K = 0
0089 DO 290 I = 1,PCOUNT
0090 IF(REPT(I,6,1) .NE. TRFC) GO TO 260

C
C WRITE PRINT REPORT
C
0091 WRITE(6,1070) (REPT(I,J,1),J=1,5)
0092 1070 FORMAT(/32Y,A4,4X,2A4,8X,A4,6X,A4)
0093 GO TO 290
0094 260 CONTINUE

C
C SAVE DATA THAT WAS NOT OUTPUT TO TAPE
C
0095 K = K+1
0096 DO 280 J=1,6
0097 REPT(K,J,2) = REPT(I,J,1)
0098 280 CONTINUE

C
C RESTORE REPORT ARRAY
C
0099 290 CONTINUE
0100 IF( K .EQ. 0) GO TO 310

```

ORIGINAL PAGE IS
OF POOR QUALITY


```
0101      DO 300 I=1,K
0102      DO 300 J=1,6
0103      REPT(I,J,1) = REPT(I,J,2)
0104      300 CONTINUE
C
0105      310 CONTINUE
C
0106      PCOUNT = K
C
0107      IF(INDEX .GT. NRD) GO TO 500
C
C      RESTORE DATA IN HOLDING ARRAYS
C
0108      DO 340 I = INDEX, NRD
0109      DO 340 J = 1,4
0110      STORE(I,J) = STORE(I+1,J)
0111      IF( J .GT. 2 ) GO TO 340
0112      SSDAY(I,J) = SSPAY(I+1,J)
0113      340 CONTINUE
C
0114      GO TO 500
0115      360 CONTINUE
0116      CALL WTQIO( YVDF, 2, 1,, S)
0117      CALL WTQIO( YWDF, 2, 1,, S)
0118      CALL WTQIO( TRWD, 2, 1,, S)
C
0119      500 CONTINUE
0120      RETURN
0121      END
```

LP#(333,3)ORDCOM.FTN

```
1.      IMPLICIT INTEGER (A-Z)
2.      PARAMETER (LEN1=50, LEN2=100)
3.      C
4.      INTEGER*4 STORE(LEN1,4), REPT(LEN2,6,2)
5.      , SITE, APATH, AROW, START, STOP
6.      C
7.      INTEGER*2 SSDAY(LEN1,2), DAYS(9)
8.      C
9.      BYTE      PATHR0(6), YR1(4), YR2(4), OUTBUF(60), LAST
10.     C
11.     COMMON / ORD / STORE, REPT, PATHR0, SSDAY, SITE, OUTBUF
12.     , START, STOP, IREC, TREC, POINT, DAYS, LAST, LAST
13.     C
14.     EQUIVALENCE ( APATH, PATHR0 )
15.     ..          ( AROW, PATHR0(5) )
16.     ..          ( YR1, START )
17.     ..          ( YR2, STOP )
18.     C
```

ORIGINAL PAGE IS
OF POOR QUALITY

INCLUDE File ORDCOM.FTN

C-12