UNIVERSITY OF CALIFORNIA

LICK OBSERVATORY TECHNICAL REPORTS

No. 10

IBM TAPE FOR LICK PDP8/I COMPUTERS

Lloyd B. Robinson

Santa Cruz, California
July 1974

IBM TAPE FOR LICK PDP-8/I COMPUTERS

CONTENTS

IBM Compatible Tape on PDP 8I

Introduction

The 9-level, IBM compatible tape transports connected to the Lick Observatory PDP 8 computers can be controlled using commands in the Lick FOCAL language. These commands allow reading or writing of single records of up to 576 IBM computer words (four 8-bit bytes per word, binary coded). The PDP 8 uses two 12-bit words per IBM 32-bit word, ignoring the most significant byte of each IBM word (+ or - signs are handled properly for integer data of absolute value less than $2^{22} - 1$.

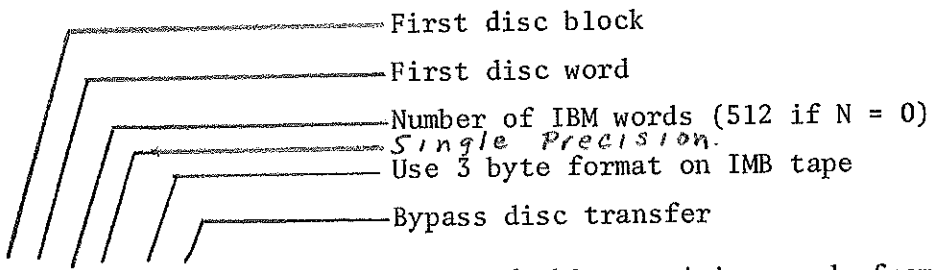
Data is written as "records" or "blocks", at a density of 800 8-bit bytes per inch, up to ~~2048~~ 2304 bytes per record. Records are separated by a "record gap" of 0.6" or longer of erased tape. "Tape marks" or "End of File" marks are followed by about 3.5" of erased tape. In order to locate a particular record on a tape, one must count records from the beginning of tape, or one may count "End of File" marks and then count records within a particular file.

When writing tape, a "read after write" error test is made. If an error is detected, the tape will backspace over the bad record, erase 5" of tape and try again on the next segment of tape. The IBM format allows gaps of any length greater than 0.6", so that large segments of bad tape can be safely skipped over by the writing program.

The instructions available are as follows:

X NAME(I) *	enable IBM tape routines.
SET D = FWRIT(B,W,N,P,Q)	copy N double precision words from DISC to IBM tape. Automatically rewrites any bad record. (D becomes equal to the number of bad segments found.)

*See X NAME( ) and X WHAT( ) instructions to determine I for a particular system tape.



SET D = FREAD(B,W,N,P,S3,Q) — Copy N double precision words from IBM tape to the disc. If an unreadable record is found, an attempt to read it is made, and D is set negative. Read 3 byte for S3, ignore disc for Q = 1.

The data Format assumes that N 12-bit lower precision words on the disc, starting at Block B, word W is followed immediately by N upper precision words. This is the format used for the image tube scanner. This also allows for easy treatment of single precision data. If P is non-zero, only N single precision words will be transferred from or to the disc. Each IBM word uses four 8-bit bytes. The PDP 8 data can use 3 bytes, with the unused byte being filled with 1 or 0 according to whether the data word is negative or positive. If S3 is non-zero, a 3 byte instead of a 4 byte format is read from tape.

SET D = FADV(N,J)	Advance N records, stop after passing a file mark if J = 0.
SET D = FBAK(N,J)	Space back N records. If J = 0, stop at first file mark and space forward over it.
X IBME(0)	Erase about 4 feet of tape.
SET D = FIBM(M)	D is set equal to the tape status, (masked by M unless M = 0).
X EOF(0)	Write an end of file mark.
X RWND(0)	Rewind up to beginning of tape ("BOT")
	The teletype will type "EOF" each time a file mark passes the read head.
X HUNT(0)	Move forward till double EOF is found and pass them.
*SET D =FIBMR(B,W,N,P,S3,Q)	uses successive disk words as Hi and Lo
*SET D =FIBMW(B,W,N,P,Q)	part of IBM tape words

*Special format for Automatic Measuring Engine

TYPICAL OPERATING PROGRAMS - IBM TAPE

A. Copy 200 words from the DISK to IBM tape; starting from word 3, block 8 on the disk.

```
1.1 X RWND (0); C - start at beginning of tape
1.2 X IBME (0); C - erase first 4 feet
1.3 X WRIT (8,3,200,1)
```

B. Copy Scan 13 from Dectape to IBM tape

```
1.1 X RWND (0)
1.2 X IBME (0)
1.3 Set SC = 13; X PUT (1,0,SC)
1.4 X WRIT (1,0,0,1); Put 512 word ID record on tape (single precision)
1.5 For J = 0,7; D 0 2

2.1 X MGET (SC*8+J)
2.2 X SAV (1); C - save on Disc as Record 1
2.3 X WRIT (8); C - store 512 channels on tape
                  C - record 1 start at block 8
```

C. Copy 150 blocks of microphotometer data from DEC tape to IBM tape

```
1.1 X MTAK (1,10,150)
1.2 X RWND (0)
1.3 X IBME (0)
1.4 F J = 1,3,148; X WRIT (J,0,387,1); C - 3 blocks = 387 words
1.5 X EOF (0)
```

(4)

```
CLICK FOCAL SCN73-C  NJJ@

01.01 C-IBM COPIER-PROG 47-L.B.R. (USES PROG.46-1 SPACE TOO)
01.02 COPIES 18 4096 WORD SCANS FROM DECTAPE TO IBM TAPE
01.03 C-WRITES 9 512 WORD RECORDS PER SCAN-
01.04 C-FIRST RECORD HOLDS 64 WORDS OF SCAN NO.THEN 64 WORDS I.D.
01.06 X NAME(0)
01.07 T !"DATA ON TAPE 7--IBM TAPE WRITE ENABLED",!!"FIRST I'LL SKIM TH
01.08 F K=1,144;X MTAK(K,K*8-1,1,7);C-SKIM OFF ID DATA
01.09 C
01.10 X NAME(98);C-IN PROG.46-1 SPACE
01.30 X RWND(0);C-START AT FRONT OF TAPE
01.40 X IBME(0);C-ERASE 4 FT. OF TAPE
01.45 C
01.50 X PUTN(180,0,0,1024);C-PRECLEAR ID BUFFER
01.60 F SC=0,17;DO 2;C-18 SCANS PER DECTAPE
01.65 C
01.70 X EOF(0);X EOF(0);C-ALLOW CONTINUED DATA WRITING LATER.
01.80 X RWND(0)
01.82 X MGET(1,0,7)
01.84 T !"ALL DONE"
01.99 Q

02.10 X PUTN(180,0,SC,64);DO 3;X WRIT(180);C-SAVE 512 WORD ID. RECORD
02.20 F K=0,7;X MGET(K+SC*8,0,7);X SAV(20);X WRIT(20*8)
02.90 R

03.01 C-PUT 64 WORD ID IN SECOND 64 WORDS OF BLOCK 180
03.10 F R=0,7;DO 4
03.90 R

04.01 C-GET AN 8 WORD SEGMENT OF ID. CODE
04.10 F N=0,7;S W(N)=FTAK(SC*8+R+1,N+121)
04.20 F N=0,7;X PUT(180,64+N+R*8,W(N))
04.90 R
*
```

ERROR DETECTION AND PREVENTION

Occasionally, dust or bad pieces of tape will cause writing errors. Most of these are automatically detected, erased and rewritten a few inches down the tape by the "X WRIT ( )" command. However, some errors will not be detected, or may be so marginal as to test okay and later produce read errors. The FOCAL "SET D = F WRIT ( )" command places the error count in D. If $D > 0$, it indicates that a bad section of tape has been erased with a (probably) good record following it. For maximum security, one should back over the record and read it, checking for any read error. This is because bad sections of tape are sometimes followed by marginally bad sections.

Since IBM tape has no addressing ability, records are identified by their sequence on the tape or by their content. Dust or bad tape can sometimes cause the computer to lose the record count. Thus it is wise to include some identifying codes on at least some of the data records. Scanner data will usually be stored as one ID record, followed by 4 data records.

N.B. It is the responsibility of FOCAL and FORTRAN programmers to choose suitable error handling procedures.

It is the responsibility of the user or operator to be sure that damaged tapes are rejected and that the tape, transport, and especially the tape capstan and head are clean.

(6)

IBM 360 TAPE READING PROGRAM

The following JCL and FORTRAN statements will read 512-word records, written by the PDP-8. The LRECL and BLKSIZE are chosen to be the byte count + 4 and byte count + 8 respectively. The IBM 360 computer apparently will not accept single READ statements that implicitly or explicitly ask for more words than exist in a single record.

```
//TAPE JOB  ([redacted],ROBINSON,MSGLEVEL=1     READ DEC TAPES.
***MESSAGE     MOUNT LICK TAPE.
***MESSAGE     PLEASE RETURN TAPE WITH CARD DECK.  ************
//AB EXEC  WATFIV
//GO.FT08F001  DD  UNIT=2400,
//    LABEL=(,NL,,IN),
//    DISP=OLD,
//    VOL=SER=LICK,
//    DSNAME=LICK,
//    DCB=(RECFM=VS,LRECL=2052,BLKSIZE=2056)
//GO.SYSIN DD *
//
```

```
$JOB            NAME,KP=29,RUN=FREE
      DIMENSION I(2048)
      READ(8)   (I (K), K=1,512)
      READ(8)   (I (K), K=513,1024)
      READ(8)   (I (K), K=1025,1536)
      READ(8)   (I (K), K=1537,2048)
      WRITE(6,12)  (I(K), K=1,2048)
12    FORMAT (1X, 16I8)
      STOP
      END

$ENTRY
```

Read 4 records; 512 words each.

(7)

April 18, 1973
Jack Baldwin

" "
PRINTING ITS DATA LABELS ON THE IBM 360

The label information contained on Image Tube Scanner data tapes (both raw data tapes and SDRS reduced data tapes) is in a combination of integer format and packed ASCII format (two characters per word). See Appendix A of the SDRS Manual[1] for more details.

A FORTRAN FUNCTION named LETTER is available for converting between Tom's Packed ASCII and EBCDIC in A2 format. The deck may be obtained from Lloyd Robinson. The function is used as in the following statement:

I1 = LETTER(I2,ID)

I2 = single precision (4 byte) integer containing unconverted characters.

I1 = single precision integer to receive converted characters (I1 can be the same variable name as I2).

ID = single precision integer to indicate direction of conversion:

ID = 0 to translate ASCII to EBCDIC.

ID = 1 to translate EBCDIC to ASCII.

This function contains tables which occupy 250 words (1000 bytes) of 360 core. A listing of the function and a sample program are attached.

The integer data stored in the tape labels has for some variables a range 0 to +4095, while for other variables the range is -2047 to +2047. Variables with a range 0 → 4095 can be read off the magnetic tape as single precision FORTRAN integers and will be in the correct format. Variables which can be either positive or negative, on the other hand, require some further conversion. Negative numbers are stored in "two's complement" arithmetic in which the code for -1 is the same as for +4095, -2 = +4094, etc. The numbers read off the tape will appear to FORTRAN to be always positive, but variables of the type which can go negative are easily converted to their correct single precision FORTRAN integer values by the following algorithm:

INTGER = INPUT - 4096*(INPUT/2048)

This algorithm should <u>not</u> be applied to variables for which the full range 0 → 4095 of positive numbers is required (i.e., Grating Setting, Dwell Time, Slit Code, etc.).

---

[1] Baldwin, J.A. (1973) Scanner Data Reduction System, Lick Observatory Technical Reports No. 2.

```
      FUNCTION LETTER(C,D)
C  STANDARD FUNCTION FOR TRANSLATING BETWEEN TOM'S PACKED ASCII
C  AND EBCDIC PACKED IN A2 FORMAT.
C  D=0 FOR ASCII TO EBCDIC, D=1 FOR EBCDIC TO ASCII.
      INTEGER LETTER,A1,A2,A3,EBC2,EBC3,EBCTBL(64),C,ASCTBL(186)
      DATA EBCTBL/124,193,194,195,196,197,198,199,200,201,209,210,211,
     1 212,213,214,215,216,217,226,227,228,229,230,231,232,233,77,111,
     2 93,111,111,64,90,127,123,91,108,80,125,77,93,92,78,107,96,75,97,
     3 240,241,242,243,244,245,246,247,248,249,122,94,76,126,110,111/
      DATA ASCTBL/32,10*63,46,60,40,43,63,38,9*63,33,36,42,41,59,63,45,
     1 47,9*63,44,37,63,62,11*63,58,35,0,39,61,34,65*63,1,2,3,4,5,6,7,8,
     2 9,7*63,10,11,12,13,14,15,16,17,18,8*63,19,20,21,22,23,24,25,26,
     3 6*63,48,49,50,51,52,53,54,55,56,57/
      IF (D)100,100,200
C  ----ASCII TO EBCDIC----
C  UNPACK ASCII
  100 A1 = C - (C/4096)*4096
      A2=A1/64
      A3 = A1 - A2*64
C  CONVERT TO EBCDIC
      EBC2 = EBCTBL(A2+1)
      EBC3 = EBCTBL(A3+1)
C  PACK UP THE EBCDIC
      LETTER = EBC3*65536
      IF (EBC2-128)120,120,125
  120 LETTER = LETTER + EBC2*16777216
      RETURN
  125 LETTER = LETTER + (EBC2-128)*16777216 - 2147483647
      RETURN
C  ----EBCDIC TO ASCII----
C  UNPACK EBCDIC
  200 IF (C)230,210,210
  210 EBC2 = C/16777216
      EBC3 = (C-EBC2*16777216)/65536
      GO TO 250
  230 C = C - 2147483647
      EBC2 = C/16777216
      EBC3 = (C-EBC2*16777216)/65536
      EBC2 = EBC2 + 128
C  CONVERT TO ASCII
C  PACK UP ASCII
  250 IF (EBC2.LT.64.OR.EBC2.GT.249) EBC2=100
      LETTER = 64*ASCTBL(EBC2-63)
      IF (EBC3.LT.64.OR.EBC3.GT.249) EBC3=100
      LETTER = LETTER + ASCTBL(EBC3-63)
      RETURN
      END
```

(9)

FORTRAN IV G LEVEL 21 MAIN DATE = 73108 13/47

```
0001            INTEGER C(40)
0002            READ(5,100) C
0003            DO 20  I=1,40
0004         20 C(I)=LETTER(C(I),1)
0005            WRITE(6,300) C
0006            DO 40  I=1,40
0007         40 C(I)=LETTER(C(I),0)
0008            WRITE(6,200) C
0009            STOP
0010        100 FORMAT(40A2)
0011        200 FORMAT(1H ,40A2)
0012        300 FORMAT(1H1,2015,/,2015)
0013            END
```

IBM TAPE PROGRAM

The IBM handler program is somewhat different for AME data than for Image Scanner data. The scanner produces blocks of 512 lower precision words, then 512 upper precision words. When writing tape, these are rearranged in core to have each upper precision word followed by its own lower precision word. Data for the AME is already arranged in proper sequence, and no rearranging is necessary.

Scanner Commands

READ WRIT

AME Commands

IBMR IBMW

All others are the same.

(11A)

IBM TAPE = Machine Language Control Codes

IOT

6701	START the function in command register
6702	STOP
6703	LOAD command register from accumulator
6704	READ status register to accumulator
6705	SKIP if wait flag is true
6706	SKIP if tape mark flag is true
6707	SKIP if encoder flag is true

STATUS WORD MEANING

Accumulator Bit	Command Word	Status Word
0	Record/the sign byte (all 1s for -ve)	End of tape
1	IBM format*	Tape ready
2	Erase	File protect
3	Forward	Rewinding
4	Reverse	Beginning of tape
5	Rewind	Write status
6	Read	Gap detected
7	Write	Read error
8	3 byte format	Tape transport on line
9	Stop when gap detected	
10	Write a tape mark	
11	Switch transport to 688 line	

*When writing mag tape to be read as "unformatted" data by the IBM 360 system, a special 8 byte code must precede each record on the tape. This code tells the 360 system how many bytes are in the record and what kind of record it is.

Writing these bytes created problems, since the PDP8 normally writes only 3 bytes of each 4 byte word, allowing the hardware to fill in the top byte with a plus or minus sign. In the programs listed here, the "3 byte" mode is used to write the first 12 bytes, which include the 8 bytes of preamble, plus the first data word. The hardware then switches to normal 4 byte mode, controlled by the "IBM Format", bit 1 in the command word. Coding to accomplish this sequence is located at location 6422, Page 18, and 6660, Page 22.

The special byte contents are as follows:

1,2 Record length(byte count +8)

3,4 Zero

5,6 Record length(byte count +4)

7,8 Zero - describes the record as a "single" record

```
L
/
/
/ICON-CONSTANTS FOR IBM TAPE
XLIST
/
IBSTAR=6701
IBSTOP=6702
IBSET=6703
IBSTAT=6704     /STATUS TO AC
SKWAIT=6705
SKFILE=6706     /SKIP TESTS
SKNCOD=6707
/
BUFR=14 /AUTO INDEX
COUNTR=ARG10
STASAV=ARG9
MERCNT=ARG10H
/
MVBAK1=6177
BAK1=6176
MVFOR1=6175
/
WAIT2=6377
GAPWT2=6376
BFSET2=6375
BOTES2=6374
/
WAIT3=6577
GAPWT3=6576
BFSET3=6575
MVBAK3=6574
MVFOR3=6573
/
XLIST
PAUSE
*
```

See Page (pointers loaded there):

	see Page
MVBAK1	15
BAK1	18
MVFOR1	15
WAIT2	12
GAPWT2	21
BFSET2	21
BOTES2	12
WAIT3	12
GAPWT3	21
BFSET3	21
MVBAK3	15
MVFOR3	15

Entries from FOCAL

KB1 +	AME	Scanner
66	IBMR,	READ
67	BAK	
70	ADV	
71	IBM?	
72	IBME	
73	IBMW,	WRIT
74	EOF	
75	RWND	
76	HUNT	
77	REVM	

CORE MAPS

Scanner-Microphotom.		A.M.E.	
6042-6173	IBM1	6042-6173	IBM1
6200-6363	IBM2	6200-6352	IBMB
6400-6566	IBM3	6400-6550	IBMC
6600-6715	IBM4	6600-6715	IBMD
6722-6775	REVM*		

* X REVM(B1, W1, N, B2, W2) reverses the order of N(≤ 9*129) words on DISC, copying from B1, W1 -- to B2, W2 --.

File 3 Tape 138
June 15/73
For Scanner Data.

(12)

```
.PALP
*OUT-S:IBM1
*
*IN-S:CON0,S:ICON,S:IBM1
*
*
*
*OPT-T

ARG1    0050

                /CON0
                XLIST
                PAUSE/
                /
                /ICON-CONSTANTS FOR IBM TAPE
                XLIST
                PAUSE/
                /
                /IBM1-WAIT,STATUS,DELAYS
                /X READ OR WRITE(B,W,N,I) N WORDS STARTING AT DISC BLOCK B
                /WORD W. DOUBLE PRECISION IF I IS ZERO.
                /S D=FIBM(M) GIVES STATUS,MASKED BY M,UNLESS M=0.
                /X IBME(0) ERASES 4 FEET OF TAPE.
                /EACH OPERATION STARTS BY TESTING WAIT,THEN 36 MSEC DELAY.
                /X HUNT(0)-LOOKS FOR DOUBLE END OF FILE.
                /
                *KB1+71
0231    6105    STATUS
0232    6121    ERASIM
                *KB1+75
0235    6140    REWIND
0236    6150    HUNT
                *FNKB1+75
0741    0364    364         /RWND
0742    2204    2204        /HUNT
                *FNKB1+71
0735    0435    435         /IBM
0736    0655    655         /IBME
                /
                *WAIT2
6377    6042    WAIT
                *WAIT3
6577    6042    WAIT
                *BOTES2
6374    6162    BOTEST
                /
                *6042
                /
6042    0000    WAIT,0   /TEST WAIT FLAG
6043    4267      JMS DELY36
6044    6704    LINTES,IBSTAT
6045    0320      AND P10
6046    7650      SNA CLA
6047    5260      JMP ONLINE
6050    4422      JMS I MESAGX
6051    1706    TEXT /OF
6052    0640    F
6053    1411    LI
6054    1605    NE
                /
```

```
6056  4276     JMS SECOND
6057  5244     JMP LINTES
               /
6060  6704   ONLINE,IBSTAT
6061  0317     AND P400
6062  7650     SNA CLA
6063  5260     JMP ONLINE     /REWINDING
6064  6705   TRY,SKWAIT
6065  5642     JMP I WAIT     /'WAIT' FLAG OFF.
6066  5264     JMP TRY
               /
6067  0000   DELY36,0         /36 MSEC DELAY
6070  3016     DCA 16         /COUNTER
6071  1671     TAD I .        /4.5 MICROSEC.DELAY
6072  2016     ISZ 16
6073  5271     JMP .-2
6074  7300     CLA CLL
6075  5667     JMP I DELY36
               /
6076  0000   SECOND,0         /2 SECOND DELAY
6077  1077     TAD M100
6100  3017     DCA 17
6101  4267     JMS DELY36
6102  2017     ISZ 17
6103  5301     JMP .-2
6104  5676     JMP I SECOND
               /
               /
6105  0000   STATUS,0
6106  6704     IBSTAT
6107  7040     CMA   /ALL BITS ARE COMPLEMENTED IN HARDWARE.
6110  3051     DCA ARG2
6111  1052     TAD ARG3
6112  7450     SNA
6113  5705     JMP I STATUS
6114  0051     AND ARG2
6115  3051     DCA ARG2       /MASK STATUS WITH ARG3
6116  5705     JMP I STATUS
               /
               /
6117  0400   P400,400
6120  0010   P10,10
               /
6121  0000   ERASIM,0
6122  4242     JMS WAIT
6123  6704     IBSTAT
6124  7006     RTL
6125  7700     SMA CLA
6126  5532     JMP I KILALL  /WRITE PROTECTED.
6127  1337     TAD ERASER
6130  6703     IBSET
6131  6701     IBSTAR
6132  7200     CLA
6133  4276     JMS SECOND
6134  6702     IBSTOP
6135  4776     JMS I BAK1    /MOVE BACK PAST GAP MARK.
6136  5721     JMP I ERASIM
               /
6137  6357   ERASER,6357     /MOVE FORWARD ERASING.
               /
```

(14)

```
6140  0000  REWIND,0
6141  4242    JMS WAIT
6142  1347    TAD CREWND
6143  6703    IBSET
6144  6701    IBSTAR
6145  7300    CLA CLL
6146  5740    JMP I REWIND
              /
6147  7677  CREWND,7677
              /
6150  0000  HUNT,0
6151  4775  SEARCH,JMS I MVFOR1      /MOVE PAST 1 RECORD
6152  6706    SKFILE
6153  5351    JMP SEARCH     /PASSED 4095 RECORDS-NO FILE
6154  4775    JMS I MVFOR1                    should count records
6155  6706    SKFILE
6156  5351    JMP SEARCH     /NOT A DOUBLE FILE END.
6157  5750    JMP I HUNT
              /
6160  7373  CMOVEF,7373      /STOP FOR A GAP
6161  0200  P200,200
              /
6162  0000  BOTEST,0         /TEST FOR BOT
6163  6704  BTEST,IBSTAT
6164  0361    AND P200
6165  7640    SZA CLA
6166  5762    JMP I BOTEST
6167  1360    TAD CMOVEF
6170  6703    IBSET
6171  6701    IBSTAR
6172  7300    CLA CLL
6173  5363    JMP BTEST      /MOVE OFF B.O.T.
```

```
.PALP
*OUT-S:IBM2
*
*IN-S:CON0,S:ICON,S:IBM2
*
*
*
*OPT-T

ARG1    0050

                /CON0
                XLIST
                PAUSE/
                /
                /ICON-CONSTANTS FOR IBM TAPE
                XLIST
                PAUSE/
                /
                /IBM2--READ IBM
                /SET D=XREAD(B,W,N,P,S3,Q).D -VE IF TAPE ERROR PERSISTS,BUT
                /READS ANYWAY! P FOR SINGLE PRECISION,S3 FOR 3BYTE,Q=NODISK
                /S D=FBAK(N,J).FADV(N,J).MOVES N RECORDS.;PAST EOF IF J NOT0
                /D IS BAD RECORD COUNT.
                /
                *MVBAK1
 6177   6266    MVBAKR
                *MVBAK3
 6574   6266    MVBAKR
                *MVFOR1
 6175   6344    MVFORC
                *MVFOR3
 6573   6344    MVFORC
                /
                *KB1+66
 0226   6200    IBREAD
 0227   6337    MOVBAK
 0230   6306    MOVFOR
                *FNKB1+66
 0732   2014    2014      /READ
 0733   3523    3523      /BAK
 0734   3466    3466      /ADV
                /
                *6200
 6200   0000    IBREAD,0
 6201   1263    IRNEXT,TAD M3
 6202   3264      DCA ERRCNT
 6203   5205      JMP TRY1
 6204   4266    TRYRED,JMS MVBAKR
 6205   4777    TRY1,JMS I WAIT2
 6206   4774      JMS I BOTES2   /MOVE OFF BOT
 6207   1361      TAD P2         /OFFSET FOR 1ST 24 BYTES IN 4 BYTE
 6210   4775      JMS I BFSET2
 6211   1056      TAD ARG7
 6212   7650      SNA CLA
 6213   1362      TAD P10        /4 BYTE MODE
 6214   1265      TAD CREAD      /STOP ON GAP
 6215   6703      IBSET
 6216   6701      IBSTAR
 6217   7300      CLA CLL
 6220   4776      JMS I GAPWT2   /WAIT FOR GAP
```

```
6221  1060     TAD STASAV
6222  7700     SMA CLA
6223  5226     JMP DATSAV     /NO READ ERROR
6224  2264     ISZ ERRCNT
6225  5204     JMP TRYRED
6226  6706   DATSAV,SKFILE
6227  5232     JMP NOFILE
6230  3060     DCA STASAV     /0 FOR FILE END
6231  5245     JMP REXIT
6232  2060   NOFILE,ISZ STASAV        /4096 FOR NORMAL READ
6233  1057     TAD ARG8
6234  7640     SZA CLA
6235  5245     JMP REXIT      /NO DISC XFER
6236  4775     JMS I BFSET2
6237  2014   LOWSAV,ISZ BUFR
6240  4251   NEXLOW,JMS PUTIT
6241  4775     JMS I BFSET2
6242  1055     TAD ARG6
6243  7650     SNA CLA
6244  4251     JMS PUTIT      /DOUBLE PRECISION-HIGH PART
6245  1060   REXIT,TAD STASAV
6246  3050     DCA ARG1
6247  3051     DCA ARG2
6250  5600     JMP I IBREAD
              /
6251  0000   PUTIT,0 /XFER DATA TO DISK
6252  1414   PUSH,TAD I BUFR
6253  3051     DCA ARG2
6254  4520     JMS I PUTWRX
6255  3052     DCA ARG3       /USE SUCCESIVE DISK WORDS
6256  3053     DCA ARG4
6257  2014     ISZ BUFR       /DOUBLE PRECISION
6260  2061     ISZ COUNTR
6261  5252     JMP PUSH
6262  5651     JMP I PUTIT
              /
6263  7775   M3,-3
6264  0000   ERRCNT,0
6265  7323   CREAD,7323       /READ FWRD,STOP ON GAP.7333 FOR 4 BYTE.
              /
6266  0000   MVBAKR,0         /MOVE BACK ONE RECORD.
6267  4777     JMS I WAIT2
6270  1305     TAD BMOVGS     /STOP ON GAP
6271  6703     IBSET
6272  6701     IBSTAR
6273  7300     CLA CLL
6274  4776     JMS I GAPWT2   /NOW READ REAL GAP
6275  6706     SKFILE
6276  5666     JMP I MVBAKR
6277  4777   BEXIT,JMS I WAIT2        /FILE MARK FOUND MOVING BACK
6300  1053     TAD ARG4
6301  7650     SNA CLA        /ARG4=1 TO BACK OVER FILE END.
6302  4344     JMS MVFORC     /MOVE UP PAST FILE MARK
6303  5666     JMP I MVBAKR
              /
6304  0040   P40,40
6305  7573   BMOVGS,7573      /STOP ON GAP
              /
6306  0000   MOVFOR,0         /MOVE ARG3 RECORDS
6307  1356     TAD FORWRD
```

```
6310  3355  SETMOV,DCA DIREC
6311  3017    DCA MERCNT
6312  1052    TAD ARG3
6313  7450    SNA
6314  5706    JMP I MOVFOR  /IGNORE ADV(0)
6315  7041    CIA
6316  3052    DCA ARG3
6317  4755  FMOVE,JMS I DIREC        /CAN BE MVFORC,MVBAKR
6320  6706    SKFILE
6321  7410    SKP
6322  5333    JMP FILEND    /FILE FOUND
6323  1060  GOON,TAD STASAV
6324  7710    SPA CLA
6325  2017    ISZ MERCNT    /COUNT BAD RECORDS
6326  2052    ISZ ARG3
6327  5317    JMP FMOVE
6330  1017  EXIT,TAD MERCNT
6331  3051    DCA ARG2
6332  5706    JMP I MOVFOR
6333  1053  FILEND,TAD ARG4
6334  7650    SNA CLA
6335  5330    JMP EXIT
6336  5323    JMP GOON
              /
6337  0000  MOVBAK,0
6340  1337    TAD MOVBAK
6341  3306    DCA MOVFOR
6342  1357    TAD BACKWD
6343  5310    JMP SETMOV
              /
6344  0000  MVFORC,0          /FORWARD 1 RECORD
6345  4777    JMS I WAIT2
6346  4774    JMS I BOTES2  /MOVE OFF BOT
6347  1363    TAD CMOVEF
6350  6703    IBSET
6351  6701    IBSTAR
6352  7300    CLA CLL
6353  4776    JMS I GAPWT2
6354  5744    JMP I MVFORC
              /
6355  0000  DIREC,0
6356  6344  FORWRD,MVFORC
6357  6266  BACKWD,MVBAKR
              /
6360  0040  PP40,40
6361  0002  P2,2
6362  0010  P10,10
6363  7373  CMOVEF,7373      /STOP FOR GAP
```

(18)

```
.PALP
*OUT-S:IBM3
*
*IN-S:CON0,S:ICON,S:IBM3
*
*
*
*OPT-T

ARG1    0050

                /CON0
                XLIST
                PAUSE/
                /
                /ICON-CONSTANTS FOR IBM TAPE
                XLIST
                PAUSE/
                /
                /IBM3-WRITE TAPE
                /S D=FWRIT(B,W,N,P,Q).ERASED SEGMENT COUNT=D
                /B FIRST BLOCK,W FIRST WORD,Q-IGNORE DISC.P-SINGLE PRECISION
                /
                *BAK1
 6176   6555    BAK
                /
                *KB1+73
 0233   6400    BWRITE
 0234   6526    ENFILE
                *FNKB1+73
 0737   0634    634         /WRIT
 0740   0176    176         /EOF
                /
                *6400
 6400   0000    BWRITE,0
 6401   3017      DCA MERCNT
 6402   4336      JMS WRTEST      /WRITE PROTECT TEST.
 6403   4775    WRITGO,JMS I BFSET3
 6404   1056      TAD ARG7
 6405   7640      SZA CLA
 6406   5222      JMP NOWGO       /IGNORE DISC FOR Q!
 6407   2014      ISZ BUFR
 6410   4314      JMS GETIT       /XFER LOW ORDER FROM DISK
 6411   4775      JMS I BFSET3    /NOW ERASE HI ORDER
 6412   3414    CLER,DCA I BUFR
 6413   2014      ISZ BUFR        /DOUBLE WORD
 6414   2061      ISZ COUNTR
 6415   5212      JMP CLER
 6416   4775      JMS I BFSET3
 6417   1055      TAD ARG6
 6420   7650      SNA CLA
 6421   4314      JMS GETIT       /HI ORDER
 6422   1713    NOWGO,TAD I WRD7          /FIRST DATA WORD
 6423   7415      ASR
 6424   0013      13                /GET SIGN into 8 bits.
 6425   0310      AND P377
 6426   3712      DCA I WRD6
 6427   3711      DCA I WRD5      /PART OF IBM PREAMBLE-0 FOR SINGLE RECORD.
 6430   4777    REPEAT,JMS I WAIT3
 6431   1300      TAD CWRITE
 6432   6703      IBSET
```

```
6433  7200     CLA
6434  6701     IBSTAR
6435  6201     CDF
6436  1704  WTEST,TAD I IBMWC
6437  7640     SZA CLA
6440  5236     JMP WTEST      /WAIT TILL ALL DATA XFERRED.
6441  6211     CDF 10
6442  4776     JMS I GAPWT3   /WAIT TILL GAP PASSES READ HEAD.
6443  6702     IBSTOP
6444  1017     TAD MERCNT
6445  3051     DCA ARG2       /TELL FOCAL ABOUT ERASED SPOTS.
6446  1060     TAD STASAV
6447  7700     SMA CLA
6450  5600     JMP I BWRITE
6451  4777     JMS I WAIT3
6452  4775  WERROR,JMS I BFSET3      /SET COUNTR
6453  1302     TAD CMOVEB
6454  2017     ISZ MERCNT     /COUNT ERRORS
6455  6703     IBSET
6456  6701     IBSTAR
6457  7300  BAKING,CLA CLL
6460  1305     TAD M4         /BACK OVER A MEASURED SPACE
6461  4344     JMS SPACER
6462  2061     ISZ COUNTR
6463  5257     JMP BAKING
6464  1307     TAD M500
6465  4344     JMS SPACER     /ALLOW FOR PART OF GAP
6466  6702     IBSTOP
6467  4777  MOVOUT,JMS I WAIT3
6470  4775     JMS I BFSET3
6471  1301     TAD ERAS
6472  6703     IBSET          /ERASE CURRENT RECORD
6473  7300     CLA CLL
6474  6701     IBSTAR
6475  4344     JMS SPACER     /ERASE 1024*4 BYTE RECORD
6476  4775     JMS I BFSET3   /ERAS TOOK 1 WORD FROM CORE.
6477  5230     JMP REPEAT     /TRY TO REWRITE IT
              /
6500  1357  CWRITE,1357      /IBM UNFORMAT,NO STOP ON GAP!
6501  6357  ERAS,6357
6502  7577  CMOVEB,7577      /CAN NOT MOVE BACK WRITING!
6503  7351  CENDF1,7351
6504  7752  IBMWC,7752
6505  7774  M4,-4
6506  0040  P40,40
6507  7300  M500,-500
6510  0377  P377,377
6511  2576  WRD5,2576   }
6512  2577  WRD6,2577   }  Defined also on Page 22
6513  2600  WRD7,2600   }
              /
6514  0000  GETIT,0
6515  4541  GETNEX,JMS I GETWRX
6516  3052     DCA ARG3
6517  3053     DCA ARG4       /SUCCESIVE WORDS
6520  1051     TAD ARG2
6521  3414     DCA I BUFR
6522  2014     ISZ BUFR       /DOUBLE PRECISION
6523  2061     ISZ COUNTR
6524  5315     JMP GETNEX
```

```
6525  5714    JMP I GETIT
              /
6526  0000    ENFILE,0
6527  4777      JMS I WAIT3
6530  4336      JMS WRTEST      /TEST WRITE PROTECT OFF.
6531  1303      TAD CENDF1
6532  6703      IBSET
6533  6701      IBSTAR
6534  7300      CLA CLL
6535  5726      JMP I ENFILE    /END FILE STARTED.
              /
6536  0000    WRTEST,0
6537  6704      IBSTAT
6540  7006      RTL
6541  7700      SMA CLA
6542  5532      JMP I KILALL    /TRIED TO WRITE WITHOUT RING.
6543  5736      JMP I WRTEST
              /
6544  0000    SPACER,0
6545  3016      DCA 16
6546  6704    STEP,IBSTAT       /CLEAR ENCODER FF.
6547  6707      SKNCOD
6550  5347      JMP .-1
6551  2016      ISZ 16
6552  5346      JMP STEP
6553  7300      CLA CLL
6554  5744      JMP I SPACER
              /
6555  0000    BAK,0
6556  4774      JMS I MVBAK3    /FIND THE GAP MARK
6557  4777      JMS I WAIT3     /DELAY NEEDED AFTER NEGATING A MOTION.
6560  1302      TAD CMOVEB
6561  6703      IBSET
6562  6701      IBSTAR          /MOVE GAP PAST WRITE HEAD
6563  7300      CLA CLL
6564  4777      JMS I WAIT3
6565  6702      IBSTOP
6566  5755      JMP I BAK
```

```
 PALP
*OUT-S:IBM4
*
*IN-S:CON0,S:ICON,S:IBM4
*
*
*
*OPT-T

ARG1    0050

                /CON0
                XLIST
                PAUSE/
                /
                /ICON-CONSTANTS FOR IBM TAPE
                XLIST
                PAUSE/
                /
                /IBM4-WAIT FOR A GAP
                /SET BUFFER POINTERS
                /
                *GAPWT2
6376    6600    GAPWIT
                *GAPWT3
6576    6600    GAPWIT
                *BFSET2
6375    6632    BFSET
                *BFSET3
6575    6632    BFSET
                /
                *6600
6600    0000    GAPWIT,0
6601    6704    GWAIT,IBSTAT
6602    0230      AND PP200
6603    7650      SNA CLA
6604    5600      JMP I GAPWIT  /B.O.T.
6605    6704      IBSTAT
6606    0231      AND PP40
6607    7640      SZA CLA
6610    5201      JMP GWAIT     /WAIT FOR GAP
6611    6704      IBSTAT
6612    0037      AND P20
6613    7650      SNA CLA
6614    7144      CMA CLL RAL   /ERROR
6615    3060      DCA STASAV
6616    6704    GAPEND,IBSTAT
6617    0231      AND PP40
6620    7650      SNA CLA
6621    5216      JMP GAPEND    /WAIT TILL GAP PULSE DONE
6622    6706      SKFILE
6623    5600      JMP I GAPWIT
6624    4422      JMS I MESAGX
6625    0517    TEXT /EO
6626    0600    F/
6627    5600      JMP I GAPWIT
                /
6630    0200    PP200,200
6631    0040    PP40,40
                /
6632    0000    BFSET,0
```

```
6633  6201    CDF
6634  1302    TAD WRD0
6635  3713    DCA I IBMCA
6636  1307    TAD WRD6        /PREPARE PREAMBLE
6637  3014    DCA BUFR
6640  1054    TAD ARG5
6641  7450    SNA
6642  1314    TAD P1000
6643  3054    DCA ARG5
6644  1054    TAD ARG5
6645  7041    CIA
6646  3061    DCA COUNTR
6647  1061    TAD COUNTR
6650  1311    TAD M3
6651  7104    CLL RAL
6652  3712    DCA I IBMWC
6653  6211    CDF 10
6654  1054    TAD ARG5
6655  1315    TAD M1100
6656  7700    SMA CLA
6657  5532    JMP I KILALL    /TOO LONG
6660  7001    IAC             /DATA BYTES+4 TO BYTE 6
6661  1054    TAD ARG5
6662  7421    MQL
6663  7413    SHL
6664  0001    1                       12
6665  3705    DCA I WRD3      /FIRST ~~24~~ BYTES IN 3 BYTE MODE
6666  7501    MQA             /(WRITE ONLY)
6667  3706    DCA I WRD4
6670  1706    TAD I WRD4
6671  1310    TAD P4          /DATA BYTES +8 TO BYTE 2(AND 1)
6672  7421    MQL
6673  1705    TAD I WRD3
6674  7413    SHL
6675  0007    7
6676  3703    DCA I WRD1
6677  7501    MQA
6700  3704    DCA I WRD2
6701  5632    JMP I BFSET
                /
6702  2571    WRD0,2571
6703  2572    WRD1,2572
6704  2573    WRD2,2573
6705  2574    WRD3,2574
6706  2575    WRD4,2575
6707  2577    WRD6,2577
6710  0004    P4,4
6711  7775    M3,-3
6712  7752    IBMWC,7752
6713  7753    IBMCA,7753
6714  1000    P1000,1000
6715  6700    M1100,-1100
```

Defined also on Page 19

— Max No. of IBM words.

(Note that core buffer used is 2571 -- 2577
and 2600 up to 2600+2200 = ~~4600~~ 4777
i.e. we use 2571 -- ~~4600~~ 4777.

(23)

File 3 Tape 15 B
June 21/23

use IBM1, IBMB
IBMC, IBMD
for double precision
access to disc.

```
.PALP
*OUT-S:IBMB
*
*IN-S:CON0,S:ICON,S:IBMB
*
*
*
*OPT-T

ARG1    0050

                /CON0
                XLIST
                PAUSE/
                /
                /ICON-CONSTANTS FOR IBM TAPE
                XLIST
                PAUSE/
                /
                /IBMB--READ IBM
                /SET D=IBMR(B,W,N,P,S3,Q.D -VEIF TAPE ERROB PERCISTS,BUD
                /READS ANYWAY! P FOR SINGLE PRECISION,S3 FOB 3BYTE,Q=NODISK
                /S D=FBAK(N,J)FADV(N,J).MOVE N RECORDS.;PASTEOF UNLESS J=0
                /D IS BAD RECORD COUNT.
                /
                *MVBAK1
 6177   6255    MVBAKR
                *MVBAK3
 6574   6255    MVBAKR
                *MVFOR1
 6175   6333    MVFORC
                *MVFOR3
 6573   6333    MVFORC
                /
                *KB1+66
 0226   6200    IBREAD
 0227   6326    MOVBAK
 0230   6275    MOVFOR
                *FNKB1+66
 0732   0672    672         /IBMR
 0733   3523    3523        /BAK
 0734   3466    3466        /ADV
                /
                *6200
 6200   0000    IBREAD,0
 6201   1252    IRNEXT,TAD M3
 6202   3253      DCA ERRCNT
 6203   5205      JMP TRY1
 6204   4255    TRYRED,JMS MVBAKR
 6205   4777    TRY1,JMC I WAIT2
 6206   4774      JMS I BOTES2    /MOVE OFF BOT
 6207   1350      TAD P2          /OFFSET FOR 1ST 24 BYTES IN 4 BYTE
 6210   4775      JMS I BFSET2
 6211   1056      TAD ARG7
 6212   7650      SNA CLA
 6213   1351      TAD P10         /4 BYTE MODE
 6214   1254      TAD CREAD       /STOP ON GAP
 6215   6703      IBSET
 6216   6701      IBSTAR
 6217   7300      CLA CLL
```

```
6221  1060    TAD STASAV
6222  7700    SMA CLA
6223  5226    JMP DATSAV      /NO READ ERROR
6224  2253    ISZ ERRCNT
6225  5204    JMP TRYRED
6226  6706  DATSAV, SKFILE
6227  5232    JMP NOFILE
6230  3060    DCA STASAV      /0 FOR FILE END
6231  5246    JMP REXIT
6232  2060  NOFILE, ISZ STASAV        /4096 FOR NORMAL READ
6233  1057    TAD ARG8
6234  7640    SZA CLA
6235  5246    JMP REXIT       /NO DISC XFER
6236  4775    JMS I BFSET2
6237  1414  PUSH, TAD I BUFR
6240  3051    DCA ARG2
6241  4520    JMS I PUTWRX
6242  3052    DCA ARG3        /USE SUCCESIVE DISK WORDS
6243  3053    DCA ARG4
6244  2061    ISZ COUNTR
6245  5237    JMP PUSH        /PUT DATA ON DISC
6246  1060  REXIT, TAD STASAV
6247  3050    DCA ARG1
6250  3051    DCA ARG2
6251  5600    JMP I IBREAD
              /
              /
6252  7775  M3, -3
6253  0000  ERRCNT0
6254  7323  CREAD, 7323        /READ FWRD, STOP ON GAP. 7333 FOR 4 BYTE.
              /
6255  0000  MVBAKR, 0          /MOVE BACK ONE RECORD.
6256  4777    JMS I WAIT2
6257  1274    TAD BMOVGS      /STOP ON GAP
6260  6703    IBSET
6261  6701    IBSTAR
6262  7300    CLA CLL
6263  4776    JMS I GAPWT2    /NOW READ REAL GAP
6264  6706    SKFILE
6265  5655    JMP I MVBAKR
6266  4777  BEXIT, JMS I WAIT2        /FILE MARK FOUND MOVING BACK
6267  1053    TAD ARG4
6270  7650    SNA CLA         /ARG4=1 TO BACK OVER FILEEND.
6271  4333    JMS MVFORC      /MOVE UP PAST FILE MARK
6272  5655    JMP I MVBAKR
              /
6273  0040  P40, 40
6274  7573  BMOVGS, 7573       /STOP ON GAP
              /
6275  0000  MOVFOR, 0          /MOVE ARG3 RECORDS
6276  1345    TAD FORWRD
6277  3344  SETMOV, DCA DIREC
6300  3017    DCA MERCNT
6301  1052    TAD ARG3
6302  7450    SNA
6303  5675    JMP I MOVFOR    /IGNORE ADV(0)
6304  7041    CIA
6305  3052    DCA ARG3
6306  4744  FMOVE, JMS I DIREC        /CAN BE MVFORCMVBAKR
```

```
6310  7410    SKP
6311  5322    JMP FILEND     /FILE FOUND
6312  1060  GOON,TAD STASAV
6313  7710    SPA CLA
6314  2017    ISZ MERCNT     /COUNT BAD RECORDS
6315  2052    ISZ ARG3
6316  5306    JMP FMOVE
6317  1017  EXIT,TAD MERCNT
6320  3051    DCA ARG2
6321  5675    JMP I MOVFOR
6322  1053  FILEND,TAD ARG4
6323  7650    SNA CLA
6324  5317    JMP EXIT
6325  5312    JMP GOON
              /
6326  0000  MOVBAK,0
6327  1326    TAD MOVBAK
6330  3275    DCA MOVFOR
6331  1346    TAD BACKWD
6332  5277    JMP SETMOV
              /
6333  0000  MVFORC0          /FORWARD 1 RECORD
6334  4777    JMS I WAIT2
6335  4774    JMS I BOTES2   /MOVE OFF BOT
6336  1352    TAD CMOVEF
6337  6703    IBSET
6340  6701    IBSTAR
6341  7300    CLA CLL
6342  4776    JMS I GAPWT2
6343  5733    JMP I MVFORC
              /
6344  0000  DIREC,0
6345  6333  FORWRD,MVFORC
6346  6255  BACKWD,MVBAKR
              /
6347  0040  PP40,40
6350  0002  P2,2
6351  0010  P10,10
6352  7373  CMOVEF,7373      /STOP FOR GAP
```

```
.PALP
*OUT-S:IBMC
*
*IN-S:CON0,S:ICON,S:IBMC
*
*
*
*OPT-T

ARG1    0050

                /CON0
                XLIST
                PAUSE/
                /
                /ICON-CONSTANTS FOR IBM TAPE
                XLIST
                PAUSE/
                /
                /IBMC-WRITE TAPE
                /S D=FIBMW(B,W,N,P,Q).ERASED SEGMEND COUNT=D
                /B FIRST BLOCK,W FIRST WORD,Q-IGNORE DISC.P-SINGLE PRECISION
                /
                *BAK1
6176    6537    BAK
                /
                *KB1+73
0233    6400    BWRITE
0234    6510    ENFILE
                *FNKB1+73
0737    0677    677             /IBMW
0740    0176    176     /EOF
                /
                *6400
6400    0000    BWRITE,0
6401    3017      DCA MERCNT
6402    4320      JMS WRTEST    /WRITE PROTECT TEST.
6403    4775    WRITGO,JMS I BFSET3
6404    1056      TAD ARG7
6405    7640      SZA CLA
6406    5216      JMP NOWGO     /IGNORE DISC FOR Q!
6407    4541    GETNEX,JMS I GETWRX
6410    3052      DCA ARG3
6411    3053      DCA ARG4      /SUCCESIVE WORDS
6412    1051      TAD ARG2
6413    3414      DCA I BUFR
6414    2061      ISZ COUNTR
6415    5207      JMP GETNEX
6416    1707    NOWGO,TAD I WRD7        /FIRST DATA WORD
6417    7415      ASR
6420    0013      13            /GET SIGN
6421    0304      AND P377
6422    3706      DCA I WRD6
6423    3705      DCA I WRD5    /PART OF IBM PREAMBLE-0 FOR SINGLE RECORD.
6424    4777    REPEAT,JMS I WAIT3
6425    1274      TAD CWRITE
6426    6703      IBSET
6427    7200      CLA
6430    6701      IBSTAR
6431    6201      CDF
```

```
6433  7640      SZA CLA
6434  5232      JMP WTEST       /WAIT TILL ALL DATA XFERRED.
6435  6211      CDF 10
6436  4776      JMS I GAPWT3    /WAIT TILL GAP PASSES READ HEAD.
6437  6702      IBSTOP
6440  1017      TAD MERCNT
6441  3051      DCA ARG2        /TELL FOCAL ABOUT ERASED SPOTS.
6442  1060      TAD STASAV
6443  7700      SMA CLA
6444  5600      JMP I BWRITE
6445  4777      JMS I WAIT3
6446  4775    WERROR,JMS I BFSET3      /SET COUNTR
6447  1276      TAD CMOVEB
6450  2017      ISZ MERCNT      /COUNT ERRORS
6451  6703      IBSET
6452  6701      IBSTAR
6453  7300    BAKING,CLA CLL
6454  1301      TAD M4          /BACK OVER A MEASURED SPACE
6455  4326      JMS SPACER
6456  2061      ISZ COUNTR
6457  5253      JMP BAKING
6460  1303      TAD M500
6461  4326      JMS SPACER      /ALLOW FOR PART OF GAP
6462  6702      IBSTOP
6463  4777    MOVOUT,JMS I WAIT3
6464  4775      JMS I BFSET3
6465  1275      TAD ERAS
6466  6703      IBSET           /ERASE CURRENT RECORD
6467  7300      CLA CLL
6470  6701      IBSTAR
6471  4326      JMS SPACER      /ERASE 1024
                                        4 BYTE RECORD
6472  4775      JMS I BFSET3    /ERAS TOOK 1 WORD FROM CORE
6473  5224      JMP REPEAT      /TRY TO REWRITE IT
                /
6474  1357    CWRITE,1357       /IBM UNFORMAT,NO STOP ON GAP!
6475  6357    ERAS,6357
6476  7577    CMOVEB,7577       /CAN NOT MOVE BACK WRITING!
6477  7351    CENDF1,7351
6500  7752    IBMWC,7752
6501  7774    M4,-4
6502  0040    P40,40
6503  7300    M500,-500
6504  0377    P377,377
6505  2576    WRD5,2576
6506  2577    WRD6,2577
6507  2600    WRD7,2600
                /
                /
6510  0000    ENFILE,0
6511  4777      JMS I WAIT3
6512  4320      JMS WRTEST      /TEST WRITE PROTECT OFF.
6513  1277      TAD CENDF1
6514  6703      IBSET
6515  6701      IBSTAR
6516  7300      CLA CLL
6517  5710      JMP I ENFILE    /END FILE STARTED.
                /
6520  0000    WRTEST,0
                  IBSTAT
```

```
6523  7700     SMA CLA
6524  5532     JMP I KILALL  /TRIED TO WRITE WITHOUT RING.
6525  5720     JMP I WRTEST
               /
6526  0000   SPACER,0
6527  3016     DCA 16
6530  6704   STEP,IBSTAT     /CLEAR ENCODER FF.
6531  6707     SKNCOD
6532  5331     JMP .-1
6533  2016     ISZ 16
6534  5330     JMP STEP
6535  7300     CLA CLL
6536  5726     JMP I SPACER
               /
6537  0000   BAK,0
6540  4774     JMS I MVBAK3  /FIND THE GAP MARK
6541  4777     JMS I WAIT3   /DELAY NEEDED AFTER NEGATING A MOTION
6542  1276     TAD CMOVEB
6543  6703     IBSET
6544  6701     IBSTAR        /MOVE GAP PAST WRITE HEAD
6545  7300     CLA CLL
6546  4777     JMS I WAIT3
6547  6702     IBSTOP
6550  5737     JMP I BAK
```

```
.PALP
*OUT-S:IBMD
*
*IN-S:CON0,S:ICON,S:IBMD
*
*
*
*OPT-T

ARG1    0050

                /CON0
                XLIST
                PAUSE/
                /
                /ICON-CONSTANDS FOR IBM TAPE
                XLIST
                PAUSE/
                /
                /IBMD-WAIT FOR GAP
                /SET BUFFER POINTERS
                /
                *GAPWT2
6376    6600    GAPWIT
                *GAPWT3
6576    6600    GAPWIT
                *BFSET2
6375    6632    BFSET
                *BFSET3
6575    6632    BFSET
                /
                *6600
6600    0000    GAPWIT,0
6601    6704    GWAIT,IBSTAT
6602    0230      AND PP200
6603    7650      SNA CLA
6604    5600      JMP I GAPWIT  /B.O.T.
6605    6704      IBSTAT
6606    0231      AND PP40
6607    7640      SZA CLA
6610    5201      JMP GWAIT     /WAIT FOR GAP
6611    6704      IBSTAT
6612    0037      AND P20
6613    7650      SNA CLA
6614    7144      CMA CLL RAL   /ERROR
6615    3060      DCA STASAV
6616    6704    GAPEND,IBSTAT
6617    0231      AND PP40
6620    7650      SNA CLA
6621    5216      JMP GAPEND    /WAIT TILL GAP PULSE DONE
6622    6706      SKFILE
6623    5600      JMP I GAPWIT
6624    4422      JMS I MESAGX
6625    0517    TEXT /EO
6626    0600    F/
6627    5600      JMP I GAPWIT
               /
6630    0200    PP200,200
6631    0040    PP40,40
                /
6632    0000    BFSET,0
```

(30)

```
6633  6201      CDF  — (shouldn't we have wrdØ-1 ?)
6634  1302      TAD WRDØ
6635  3713      DCA I IBMCA
6636  1307      TAD WRD6        /PREPARE PREAMBLE
6637  3014      DCA BUFR
664Ø  1Ø54      TAD ARG5
6641  745Ø      SNA
6642  1314      TAD P1ØØ4       /516 WORDS NORMALLY
6643  3Ø54      DCA ARG5
6644  1Ø54      TAD ARG5
6645  71Ø4      CLL RAL         /2 PDP WORDS PER WORD
6646  7Ø41      CIA
6647  3Ø61      DCA COUNTR
665Ø  1Ø61      TAD COUNTR
6651  1311      TAD M6          /6 WORDS OF PREAMBLE TO WRITE
6652  3712      DCA I IBMWC
6653  6211      CDF 1Ø
6654  1Ø54      TAD ARG5
6655  1315      TAD M11ØØ
6656  77ØØ      SMA CLA
6657  5532      JMP I KILALL    /TOO LONG
666Ø  7ØØ1      IAC             /DATA BYTES+4 TO BYTE 6
6661  1Ø54      TAD ARG5
6662  7421      MQL
6663  7413      SHL
6664  ØØØ1      1
6665  37Ø5      DCA I WRD3      /FIRST 24 BYTES IN 3 BYTE MODE
6666  75Ø1      MQA             /(WRITE ONLY)
6667  37Ø6      DCA I WRD4
667Ø  17Ø6      TAD I WRD4
6671  131Ø      TAD P4          /DATA BYTES +8 TO BYTE 2(AND 1)
6672  7421      MQL
6673  17Ø5      TAD I WRD3
6674  7413      SHL
6675  ØØØ7      7
6676  37Ø3      DCA I WRD1
6677  75Ø1      MQA
67ØØ  37Ø4      DCA I WRD2
67Ø1  5632      JMP I BFSET
                /
67Ø2  2571      WRDØ,2571
67Ø3  2572      WRD1,2572
67Ø4  2573      WRD2,2573
67Ø5  2574      WRD3,2574
67Ø6  2575      WRD4,2575
67Ø7  2577      WRD6,2577
671Ø  ØØØ4      P4,4
6711  7772      M6,-6
6712  7752      IBMWC,7752
6713  7753      IBMCA,7753
6714  1ØØ4      P1ØØ4,1ØØ4
6715  67ØØ      M11ØØ,-11ØØ
```

File 3 Type 13B
July 13/73
(on my MICA Tape)
(with IBM 1,2,3,4)

(31)

```
.PALP
*OUT-S:REVM
*
*IN-S:CONG,S:REVM
*
*
*OPT-T

ARG1    0350

                /CONG
                XLIST
                PAUSE/
                /
                /REVM
                /X REVM(B1,W1,N,B2,W2)MOVES N WORDS AT B1,W1 TO
                /B2,W2 BUT REVERSES THE ORDER.N UP TO 9*129 WORDS
                /B1 MAY =B2
                /
                SAVE=ARG10
                COUNT=ARG9
                WORD=ARG8
                /
                *FNKS1+77
0743    2275    2275    /REVM
                *KB1+77
0237    6722    REVERS
                /
                *6722   /PART OF IBM PACKAGE
6722    0000    REVERS,0
6723    1054      TAD ARG5
6724    1375      TAD M2212
6725    7700      SMA CLA
6726    5532      JMP I KILALL  /TOO MANY WORDS
6727    1054      TAD ARG5
6730    3061      DCA SAVE
6731    1061      TAD SAVE
6732    7041      CIA
6733    3060      DCA COUNT
6734    1374      TAD P2567
6735    3057      DCA WORD      /FIRST OF BUFFER
6736    4541    NEXTIN, JMS I GETWRX
6737    3052      DCA ARG3
6740    3053      DCA ARG4      /FOR SEQUENTIAL INPUT
6741    1051      TAD ARG2
6742    3457      DCA I WORD
6743    2057      ISZ WORD
6744    2060      ISZ COUNT
6745    5336      JMP NEXTIN
6746    1061    OUT, TAD SAVE
6747    7041      CIA
6750    3060      DCA COUNT
6751    7240      CLA CMA
6752    1374      TAD P2567
6753    1061      TAD SAVE
6754    3057      DCA WORD      /LAST WORD IN BUFFER
6755    1055      TAD ARG6
6756    3052      DCA ARG3
6757    1056      TAD ARG7
6760    3053      DCA ARG4      /SETTING OUTPUT ADDRESS FOR DISC
```

```
6761  1457  NEXOUT,TAD I WORD
6762  3051    DCA ARG2
6763  4520    JMS I PUTWRX
6764  3052    DCA ARG3        /SEQUENTIAL
6765  3053    DCA ARG4
6766  7240    CLA CMA
6767  1057    TAD WORD
6770  3057    DCA WORD        /MOVE BACK ONE WORD
6771  2060    ISZ COUNT
6772  5361    JMP NEXOUT
6773  5722    JMP I REVERS
              /
6774  2567  P2567,2567
6775  5566  M2212,-2212
```