[redacted]

514

HELIOS A
MICROMETERIOD IMPACT DATA
74-097A-12A

MAX-PLANCK-INSTITUT FÜR KERNPHYSIK
HEIDELBERG

HELIOS-A

MICROMETEOROID IMPACT DATA

74-097A-12A

This data set has been restored. There was originally one 9-track, 1600 BPI tape written in Binary. There is one restored tape. The DR tape is a 3480 cartridge and the DS tape is 9-track, 6250 BPI. The original tape was created on a DEC10 computer and the restored tape was created on an IBM 9021 computer. The DR and DS numbers along with the corresponding D number are as follows:

DR#	DS#	D#	FILES	TIME SPAN
DR005294	DS005294	D046013	1	12/19/74 - 01/02/80

REQ. AGENT	RAND NO.	ACQ. AGENT
DEW	V0106	HKH

HELIOS A
MICROMETEOROID IMPACT DATA
74-097A-12A

This data set consists of 1 magnetic tape. The tape is 1600 BPI, 9 track, binary with one file of data on it. The tape was created on a DEC-10 computer.

D#	C#	TIME SPAN
D-46103	C-21794	12/19/74 - 01/02/80

MAX-PLANCK-INSTITUT FÜR KERNPHYSIK

D-6900 HEIDELBERG 1
Postfach 103980
Saupfercheckweg 1 (beim Bierhelder Hof)
Telefon Sammel-Nr. (06221) 5161
Durchwahl (06221) 516
Telex 461666

James I. Vette
World Data Center A
Rockets and Satellites
Goddard Space Flight Center Code 601
Greenbelt, Maryland
20771 U.S.A.

July 30, 1981

Dear Mr. Vette,

In response to your letter of the 4th of June 1981 to Dr. Fechtig, I should like to send you a data-tape with the active times and the scientific results (micrometeoroid impact data) of our HELIOS A experiment E 10 together with the description.

Yours sincerely
W. Inka Rimpler
W. Inka Rimpler

DESCRIPTION OF THE DATA

OF THE

HELIOS A MICROMETEOROID EXPERIMENT

E 10

MAX-PLANCK-INSTITUT FÜR KERNPHYSIK

SAUPFERCHECKWEG 1

D-6900 HEIDELBERG

FEDERAL REPUBLIC OF GERMANY

JULY 1981

AUTHOR: W. INKA RIMPLER

Table of contents

1. General information to the data tape

2. Program that produced the data tape

3. Description of the Number System of the DEC-10

4. Description of the Tape Format

5. Description of the structure of the data on the tape

General Information to the data tape

The data are written on a 9 track tape with 1600 BPI, using a FORTRAN-program. The records are written on the tape in a binary DEC-10-style. The data ends with an EOF-mark.

The added program shows how the tape was written.
As well we added a copy of the Hardware Reference Manual, describing the Number System and the Tape Format of the DEC-10. The organisation of the data and the scientific meaning of every word you will find in the added description.

```
00100   *
00200   *
00300   *       PROGRAM WRITES HELIOS A DATA FROM DISK ON TAPE
00400   *       MOUNT MTA:3/REELID:D7XX01/V:"1600 BPI,9 TRACK"/WRITE:YES
00500   *       EXE WDS.FOR,IUTIL.REL
00600   *
00700   *       AUTHOR:          W.I.RIMPLER
00800   *       DATE:            15-07-81
00900   *       MACHINE:         DEC-10
01000   *       LANGUAGE:        FORTRAN IV
01100   *
01200   *
01300   *
01400   *
01500   *
01600   *
01700   *
01800           REAL*8 MOUT(48),RR,RL,ZSI(66),C(250)
01900           DIMENSION XMZ(4), IREC(2219), ISDB(500), SDAT(47),
02000          &  CH(12)
02100            REAL*8 DLSSP,RNDLSS,FULTIM,RAZA,DELRH,DELRAS,WTIME,QSP
02200            INTEGER IA,IT,EA,ET,PF,PE,SEC,AD,AA,PC,SC,LC,DC,TP,WI,DI,O1,O2,EI
02300          FSI,MI,TI,BR,FM,ECT,SCD,SCH,SCM,SCMS,SPECTR(45),TIDE,GMD,GMH,GMM,
02400          FGMMS,O1O2C,EDSIC,IAMI,BCT,SSP,TSPC,DATEC,ESA,FOURY(45),ECTCK,
02500          FISPEIC,KENNUN,IENDBL
02600            EQUIVALENCE(DLSSP,ISDB(347)),(RNDLSS,ISDB(400))
02700            EQUIVALENCE(ISDB(323),FULTIM),(ISDB(327),BLOCKE)
02800          F,(ISDB(351),RAZA)
02900          F,(ISDB(332),DELRH)
03000          F,(ISDB(334),DELRAS)
03100          F,(ISDB(336),WTIME)
03200          F,(ISDB(3),FM)
03300          F,(ISDB(4),BR)
03400          F,(ISDB(7),IDM)
03500          F,(ISDB(9),GMD)
03600          F,(ISDB(10),GMH)
03700          F,(ISDB(11),GMM)
03800            EQUIVALENCE(ISDB(12),GMMS)
03900          F,(ISDB(13),SCD)
04000          F,(ISDB(14),SCH)
04100          F,(ISDB(15),SCM)
04200          F,(ISDB(16),SCMS)
04300          F,(ISDB(19),IA)
04400          F,(ISDB(20),IT)
04500          F,(ISDB(21),EA)
04600          F,(ISDB(22),ET)
04700          F,(ISDB(23),PF)
04800          F,(ISDB(24),PE)
04900            EQUIVALENCE (ISDB(25),ECT)
05000          F,(ISDB(26),TIDE)
05100          F,(ISDB(27),TI)
05200          F,(ISDB(28),SEC)
05300          F,(ISDB(29),AD)
05400          F,(ISDB(31),AA)
05500          F,(ISDB(32),PC)
05600          F,(ISDB(33),SC)
05700          F,(ISDB(34),LC)
05800          F,(ISDB(35),DC)
05900          F,(ISDB(36),ED)
06000            EQUIVALENCE(ISDB(37),SI)
```

```
06100          F,(ISDB(38),O1)
06200          F,(ISDB(39),O2)
06300          F,(ISDB(40),TP)
06400          F,(ISDB(41),WI)
06500          F,(ISDB(42),DI)
06600          F,(ISDB(43),MI)
06700          F,(ISDB(44),SPECTR)
06800          F,(ISDB(325),KENNUN)
06900          F,(ISDB(326),BCT)
07000          F,(ISDB(329),IENDBL)
07100           EQUIVALENCE(ISDB(338),ISPEIC)
07200          F,(ISDB(341),O1O2C)
07300          F,(ISDB(342),EDSIC)
07400          F,(ISDB(343),DATEC)
07500          F,(ISDB(344),IAMI)
07600          F,(ISDB(346),SSP)
07700          F,(ISDB(349),TSPC)
07800          F,(ISDB(350),ESA)
07900          F,(ISDB(353),FOURY)
08000          F,(ISDB(398),ECTCK)
08100             REAL*8 QCB,RVU,RV,RVO,RMU,RM,RMO
08200            EQUIVALENCE(ISDB(410),RV),(RM,ISDB(416))
08300          F,(ISDB(406),QCB),(ISDB(408),RVU),(ISDB(412),RVO),(ISDB(414),RMU)
08400          F,(ISDB(418),RMO),(ISDB(420),QSP)
08500     *
08600     *
08700             EQUIVALENCE(IREC(83),RR),(IREC(81),RL),(ZSI,IREC(585)),
08800          &  (IREC(1219),C)
08900     *
09000     *
09100     *
09200             OPEN(UNIT=1,DEVICE='DSK',ACCESS='SEQIN',MODE='BINARY',
09300          &  RECORDSIZE=96,FILE='MZHIG.OUT')
09400             OPEN(UNIT=2,DEVICE='DSK',ACCESS='SEQIN',MODE='BINARY',
09500          &  RECORDSIZE=2219,FILE='ALL.DAT')
09600             OPEN(UNIT=3,DEVICE='MTA',DENSITY='1600',MODE='BINARY',
09700          &  ACCESS='SEQOUT')
09800     *
09900     *
10000     *
10100             REWIND 3
10200     *
10300     *
10400     *        ACTIVE TIME ON TAPE
10500     *
10600     *
10700             DO 26 IK=1,4
10800      26     READ(1) MOUT
10900             DO 1 I=1,202
11000             DO 3 K=1,4
11100             XMZ(K)=0.
11200      3      CONTINUE
11300             DO 2 J=1,10
11400             READ(1,END=666,ERR=777) MOUT
11500             XMZ(1)=XMZ(1)+MOUT(17)
11600             XMZ(2)=XMZ(2)+MOUT(29)
11700             XMZ(3)=XMZ(3)+(MOUT(22)+MOUT(23)+MOUT(24))/3.
11800             XMZ(4)=XMZ(4)+(MOUT(30)+MOUT(31)+MOUT(32))/3.
11900      2      CONTINUE
12000             DO 4 K=1,4
```

```
12100           XMZ(K)=XMZ(K)/864000.
12200     4     CONTINUE
12300           WRITE(3,END=888,ERR=999) XMZ
12400     1     CONTINUE
12500           DO 5 K=1,4
 500            XMZ(K)=0.
12700     5     CONTINUE
12800           DO 6 K=1,4
12900           READ(1,END=666,ERR=777) MOUT
13000           XMZ(1)=XMZ(1)+MOUT(17)
13100           XMZ(2)=XMZ(2)+MOUT(29)
13200           XMZ(3)=XMZ(3)+(MOUT(22)+MOUT(23)+MOUT(24))/3.
13300           XMZ(4)=XMZ(4)+(MOUT(30)+MOUT(31)+MOUT(32))/3.
13400     6     CONTINUE
13500           DO 7 K=1,4
13600           XMZ(K)=XMZ(K)/4.
13700           XMZ(K)=XMZ(K)/86400.
13800     7     CONTINUE
13900           WRITE(3,END=888,ERR=999) XMZ
14000     *
14100     *
14200     *     PARTICLES ON TAPE
14300     *
14400           DO 8 I=1,235
14500           READ(2,END=666,ERR=777) IREC
14600           DO 9 J=1,500
14700     9     ISDB(J)=IREC(J+84)
14800           SDAT(1)=FLOAT(I)
14900           SDAT(2)=FLOAT(IREC(8))
15000           SDAT(3)=FLOAT(IREC(10))
 5100           SDAT(4)=FLOAT(IREC(12))
15200           SDAT(5)=FLOAT(IREC(14))
15300           SDAT(6)=FLOAT(IREC(16))
15400           SDAT(7)=FLOAT(DI)
15500           SDAT(8)=RVU
15600           SDAT(9)=RV
15700           SDAT(10)=RVO
15800           SDAT(11)=RMU
15900           SDAT(12)=RM
16000           SDAT(13)=0.
16100           IF(IA.EQ.16.OR.EA.EQ.16) SDAT(13)=1.
16200           SDAT(14)=RMO
16300           SDAT(15)=RR
16400           SDAT(16)=RL
16500           APEX=WINKEL(ISDB(28),ISDB(42))
16600           SDAT(17)=APEX
16700           SDAT(18)=FLOAT(IREC(32))
16800           SDAT(19)=QSP
16900           SDAT(20)=4.
17000           IF(IA.LE.12) SDAT(20)=3
17100           IF(IA.LE.7) SDAT(20)=2
17200           IF(IA.LE.4) SDAT(20)=1
17300           SDAT(21)=FLOAT(LC)
17400           XHYP1=ZSI(1)+ZSI(2)
17500           XHYP2=ZSI(23)+ZSI(24)
17600           XHYP3=ZSI(45)+ZSI(46)
17700           IHYP=3
17800           IF(XHYP1.GE.0.75) IHYP=0
17900           IF(IHYP.EQ.0) GOTO 40
18000           IF(XHYP2.GE.0.75) IHYP=1
```

```
18100            IF(IHYP.EQ.1) GOTO 40
18200            IF(XHYP3.GE.0.75) IHYP=2
18300     40     CONTINUE
18400            IF(ZSI(25).EQ.9999.OR.XHYP2.GE.0.99) ZSI(25)=0.
18500            IF(ZSI(26).EQ.9999.OR.XHYP2.GE.0.99) ZSI(26)=0.
18600            IF(ZSI(29).EQ.9999.OR.XHYP2.GE.0.99) ZSI(29)=1.
18700            IF(ZSI(30).EQ.9999.OR.XHYP2.GE.0.99) ZSI(30)=1.
18800            ZSI(35)=ZSI(35)-0.02
18900            SDAT(22)=ZSI(25)
19000            SDAT(23)=ZSI(26)
19100            SDAT(24)=ZSI(29)
19200            SDAT(25)=ZSI(30)
19300            SDAT(26)=ZSI(31)
19400            SDAT(27)=ZSI(32)
19500            SDAT(28)=ZSI(35)
19600            SDAT(29)=ZSI(36)
19700            SDAT(30)=ZSI(23)+ZSI(27)
19800            SDAT(31)=ZSI(23)+ZSI(24)
19900            SDAT(32)=FLOAT(IHYP)
20000            SDAT(33)=0.
20100            IF(IREC(103).EQ.16) SDAT(33)=1.
20200            XD=0.
20300            XLD=RM*RV**2.2
20400            IF(XLD.GT.1.15E-10.AND.IREC(32).EQ.1.AND.IREC(126)
20500           &  .EQ.1) XD=1.
20600            SDAT(34)=XD
20700            IF(IREC(2219).EQ.'U') XKENN=1.
20800            IF(IREC(2219).EQ.'C') XKENN=2.
20900            IF(IREC(2219).EQ.'E') XKENN=3.
21000            SDAT(35)=XKENN
21100            CMAX=0.
21200            DO 24 K=1,250
21300            IF(C(K).GE.CMAX)CMAX=C(K)
21400     24     CONTINUE
21500            CHSUM=0.
21600            DO 25 K=1,225
21700     25     CHSUM=CHSUM+C(K)
21800            IF(CHSUM.EQ.0.) CHSUM=1.
21900            CALL ISET(CH,12,0)
22000            DO 41 K=1,24
22100     41     CH(1)=CH(1)+C(K)/CHSUM
22200            DO 11 K=25,49
22300     11     CH(2)=CH(2)+C(K)/CHSUM
22400            DO 12 K=50,73
22500     12     CH(3)=CH(3)+C(K)/CHSUM
22600            DO 13 K=74,94
22700     13     CH(4)=CH(4)+C(K)/CHSUM
22800            DO 14 K=95,114
22900     14     CH(5)=CH(5)+C(K)/CHSUM
23000            DO 15 K=115,132
23100     15     CH(6)=CH(6)+C(K)/CHSUM
23200            DO 16 K=133,149
23300     16     CH(7)=CH(7)+C(K)/CHSUM
23400            DO 17 K=150,166
23500     17     CH(8)=CH(8)+C(K)/CHSUM
23600            DO 18 K=167,182
23700     18     CH(9)=CH(9)+C(K)/CHSUM
23800            DO 19 K=183,197
23900     19     CH(10)=CH(10)+C(K)/CHSUM
24000            DO 20 K=198,212
```

```
24200           DO 21 K=213,225
24300    21     CH(12)=CH(12)+C(K)/CHSUM
24400           DO 10 J=1,12
24500           SDAT(J+35)=CH(J)
24600     10    CONTINUE
24700           WRITE(3,END=888,ERR=999) SDAT
24800     8     CONTINUE
24900           END FILE 3
25000           STOP
25100     666   STOP' EOF BEI READ'
25200     777   STOP' PARITY-ERROR IN READ'
25300     999   STOP' PARITY-ERROR TAPE'
25400     888   STOP' EOF IN WRITE'
25500           END
```

```
28900           FUNCTION WINKEL(JSEC,JDI)
29000           INTEGER JSEC,JDI
29100           IF(JDI.EQ.0)WINKEL=(JSEC-88.)*360./128.
29200           IF(JDI.EQ.1)WINKEL=(JSEC-56.)*360./128.
29300           IF(WINKEL.LT.0.)WINKEL=WINKEL+360.
29400           RETURN
29500           END
```

Number System

A program can interpret a data word as a 36-digit, unsigned binary number, or the left and right halves of a word can be taken as separate 18-bit numbers. The PDP–10 repertory includes instructions that add or subtract one from both halves of a word, so the right half can be used for address modification when the word is addressed as an index register, while the left half is used to keep a control count.

The fixed-point arithmetic instructions use twos complement representations to do binary arithmetic. In a word used as a number, bit 0 (the leftmost bit) represents the sign, 0 for positive, 1 for negative. In a positive number the remaining thirty-five bits are the magnitude in ordinary binary notation. The negative of a number is obtained by taking its twos complement. If x is an n-digit binary number, its twos complement is $2^n - x$, and its ones complement is $(2^n - 1) - x$, or equivalently $(2^n - x) - 1$. Subtracting a number from $2^n - 1$ (i.e. from all 1s) is equivalent to performing the logical complement, i.e. changing all 0s to 1s and all 1s to 0s. Therefore, to form a twos complement one takes the logical complement (usually referred to merely as the complement) of the entire word including the sign, and adds 1 to the result. In a negative number the sign bit is 1, and the remaining bits are the twos complement of the magnitude.

$+153_{10} = +231_8 =$ 000 000 000 000 000 000 000 000 000 010 011 001

0 35

$-153_{10} = -231_8 =$ 111 111 111 111 111 111 111 111 111 101 100 111

0 35

A twos complement addition actually acts as though the words represented 36-bit unsigned numbers, i.e. the signs are treated just like magnitude bits. In the absence of a carry into the sign stage, adding two numbers with the same sign produces a plus sign in the result. The presence of a carry gives a positive answer when the summands have different signs. The result has a minus sign when there is a carry into the sign bit and the summands have the same sign, or the summands have different signs and there is no carry. Thus the program can interpret the numbers processed in

fixed point addition and subtraction as signed numbers with thirty-five magnitude bits or as unsigned 36-bit numbers. A computation on signed numbers produces a result that is correct as an unsigned 36-bit number even if overflow occurs, but the hardware interprets the result as a signed number to detect overflow. Adding two positive numbers whose sum is greater than or equal to 2^{35} gives a negative result, indicating overflow; but that result, which has a 1 in the sign bit, is the correct answer interpreted as a 36-bit unsigned number in positive form. Similarly adding two negatives gives a result which is always correct as an unsigned number in negative form.

Zero is represented by a word containing all 0s. Complementing this number produces all 1s, and adding 1 to that produces all 0s again. Hence there is only one zero representation and its sign is positive. Since the numbers are symmetrical in magnitude about a single zero representation, all even numbers both positive and negative end in 0, all odd numbers in 1 (a number all 1s represents –1). But since there are the same number of numbers with each sign and zero has a plus sign, there is one more negative number than there are strictly positive numbers (nonzero numbers with a plus sign). This is the most negative number and it cannot be produced by negating any positive number (its octal representation is 400000 000000 and its magnitude is one greater than the largest positive number).

If ones complements were used for negatives one could read a negative number by attaching significance to the 0s instead of the 1s. In twos complement notation each negative number is one greater than the complement of the positive number of the same magnitude, so one can read a negative number by attaching significance to the rightmost 1 and attaching significance to the 0s at the left of it (the negative number of largest magnitude has a 1 in only the sign position). In a negative integer, 1s may be discarded at the left, just as leading 0s may be dropped in a positive integer. In a negative fraction, 0s may be discarded at the right. So long as only 0s are discarded, the number remains in twos complement form because it still has a 1 that possesses significance; but if a portion including the rightmost 1 is discarded, the remaining part of the fraction is now a ones complement. Single precision multiplication produces a double length product, and the programmer must remember that discarding the low order part of a double length negative leaves the high order part in correct twos complement form only if the low order part is zero.

The computer does not keep track of a binary point — the programmer must adopt a point convention and shift the magnitude of the result to conform to the convention used. Two common conventions are to regard a number as an integer (binary point at the right) or as a proper fraction (binary point at the left); in these two cases the range of numbers represented by a single word is -2^{35} to $2^{35} - 1$ or -1 to $1 - 2^{-35}$. Since multiplication and division make use of double length numbers, there are special instructions for performing these operations with integral operands.

The format for double length fixed point numbers is just an extension of the single length format. The magnitude (or its twos complement) is the 70-bit string in bits 1–35 of the high and low order words. Bit 0 of the high order word is the sign, and bit 0 of the low order word is made equal to the

sign. The range for double length integers and proper fractions is thus -2^{70} to $2^{70} - 1$ and -1 to $1 - 2^{-70}$. The double precision instructions actually use quadruple length numbers for products and dividends. But numbers of any length are just a further extension of the basic format: thirty-five additional bits of the number in each lower order word, and bit 0 made equal to the sign. Remember that truncating a multiple length negative requires an adjustment for the twos complement unless the part discarded is zero. The convention for bit 0 of lower order words is inconsistent with that used for floating point format (see below). This does not affect the arithmetic instructions themselves, as they ignore bit 0 in all lower order words. However instructions that negate a doubleword use the floating point convention. This means that if such instructions are used for fixed point numbers, a problem could arise when comparing one double precision number with another.

Floating Point Numbers

The floating point instructions provide for conversion between fixed and floating forms and handle both single and double precision floating point numbers. The same format is used for a single precision number and the high order word of a double precision number. A floating point instruction interprets bit 0 as the sign, but interprets the rest of the word as an 8-bit exponent and a 27-bit fraction. For a positive number the sign is 0, as before. But the contents of bits 9–35 are now interpreted only as a binary fraction, and the contents of bits 1–8 are interpreted as an integral exponent in excess 128 (200_8) code. Exponents from –128 to +127 are therefore represented by the binary equivalents of 0 to 255 ($0–377_8$). Floating point zero and negatives are represented in exactly the same way as in fixed point: zero by a word containing all 0s, a negative by the twos complement. A negative number has a 1 for its sign and the twos complement of the fraction, but since every fraction must ordinarily contain a 1 unless the entire number is zero (see below), it has the ones complement of the exponent code in bits 1–8. Since the exponent is in excess 128 code, an actual exponent x is represented in a positive number by $x + 128$, in a negative number by $127 - x$. The programmer, however, need not be concerned with these representations as the hardware compensates automatically. For example, for the instruction that scales the exponent, the hardware interprets the integral scale factor in standard twos complement form but produces the correct ones complement result for the exponent.

$+153_{10} = +231_8 = +.462_8 \times 2^8$

$= \boxed{0\,|\,10\,001\,000\,|\,100\,110\,010\,000\,000\,000\,000\,000\,000}$

0 1 8 9 35

$-153_{10} = -231_8 = -.462_8 \times 2^8$

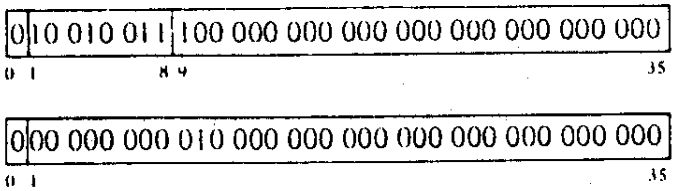
$= \boxed{1\,|\,01\,110\,111\,|\,011\,001\,110\,000\,000\,000\,000\,000\,000}$

0 1 8 9 35

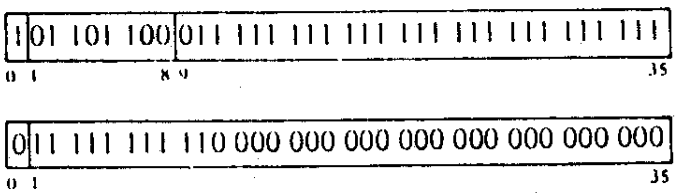
The floating point instructions assume that all nonzero operands are normalized, and they normalize a nonzero result. A floating point number is considered normalized if the magnitude of the fraction is greater than or equal to ½ and less than 1. The hardware may not give the correct result if the program supplies an operand that is not normalized or that has a zero fraction with a nonzero exponent.

Single precision floating point numbers have a fractional range in magnitude of ½ to $1 - 2^{-27}$. Increasing the length of a number to two words does not significantly change the range but rather increases the precision; in any format the magnitude range of the fraction is ½ to 1 decreased by the value of the least significant bit. In all formats the exponent range is –128 to +127.

The precaution about truncation given for fixed point multiplication applies to single precision floating point operations as they are done in extra length; but the programmer may request rounding, which automatically restores the high order part (the result) to twos complement form if it is negative. In double precision floating point instructions, all operands and results are double length, and all instructions calculate an extra length answer, which is rounded to double length with the appropriate adjustment for a twos complement negative. In double precision format the high order word is the same as a single precision number, and bits 1–35 of the low order word are simply an extension of the fraction, which is now sixty-two bits. Bit 0 of the low order word is made 0 in a result but is ignored in all operands; e.g. the number $2^{18} + 2^{-18}$ has this two-word representation in double precision format,



and its negative is



STANDARD MAGNETIC TAPE

A system for handling industry-standard magnetic tape consists of a TM10 control and up to eight tape transports; each unit in the system occupies a separate cabinet. DEC supplies several types of transports that differ in tape speed and tape handling characteristics. Each type is available in two versions, for recording information in seven tracks and nine tracks. Thus data transfer rates and timing depend on the transport, but each transport supplies information to the control such that transports of different speeds and recording formats can be operated by a single control. Transports currently available move tape at speeds of 45, 75 and 150 inches per second. Every transport accommodates two 10½-inch reels (one for supply, one for takeup) and can record information in three densities: 200, 556 and 800 bytes per inch (bpi). A full reel has 2400 feet of half-inch tape and at 800 bpi can store over 135 million bits of data in the 7-track format, or over 180 million bits in the 9-track format.

The program communicates with the tape control, which in turn governs all tape transports but communicates with only one transport at a time. Reading and writing (recording) can occur only when tape is moving forward (from supply reel to takeup reel), but the control can space the tape (*ie* move it to a new position) in either direction. Although only one transport can be reading, writing or spacing at a time, rewinding the entire tape onto the supply reel at high speed requires only initiation by the control. The rewinding transport then proceeds automatically while the control can operate another transport.

Data transfers between tape and control are governed entirely by the control. Transfers between a TM10A control and memory are handled by the program over the IO bus, whereas the TM10B control is connected to a data channel for automatic transfer of data to and from memory, thus bypassing the central processor [§5.1].

TAPE FORMAT

The control writes characters containing seven or nine bits of information; one bit is written in each track. Every character is part of a data record or a file mark. A data record contains both data characters and error-checking characters. Every data character consists of a data byte and a lateral parity bit, which the control generates so that the number of 1s in the character is odd or even as specified by the program. The data bytes in a record taken together correspond to a block of words sent from memory to the control. To separate adjacent records the control automatically erases a segment of tape between them; this segment is called a "record gap". The control always stops the tape in a gap.

The recording technique used is NRZI (nonreturn to zero, inverting). In a given frame (*ie* character position) a change in the direction of magnetization in any track represents a 1 in the character bit corresponding to that track. Thus if the same bit is 0 in a string of characters, there is no change in the track corresponding to that bit; but for a string of 1s, the flux direction changes in every frame.

To facilitate tape processing, the program can group sets of data records into files. The end of a file is indicated by a 3-inch gap followed by a file mark. The file mark is a special record containing a single, special data character and its LPCC, which is equivalent. The control always terminates a function when it encounters a file mark; in particular, the control can space by files as well as by numbers of records.

Each tape has two physical markers to indicate its extremities. These markers are reflective strips that are sensed by photoelectric cells in the transport (one marker can be seen on the tape in the illustration at the end of §H4.2). The loadpoint marker is located about fifteen feet in from the beginning of the reel and denotes the logical beginning of the tape. Reverse functions stop automatically at this marker. A load point gap of at least three inches (twenty-five feet maximum) precedes the first record on the tape. The endpoint marker is about twenty-five feet from the physical end of the tape; the final fifteen feet of tape should be left for trailer, *ie* the program should not record more than ten feet beyond the endpoint (this is enough for a 4000-word record at low density). A status bit indicates when the tape is beyond the endpoint, but this condition stops the tape automatically only when it is spacing forward.

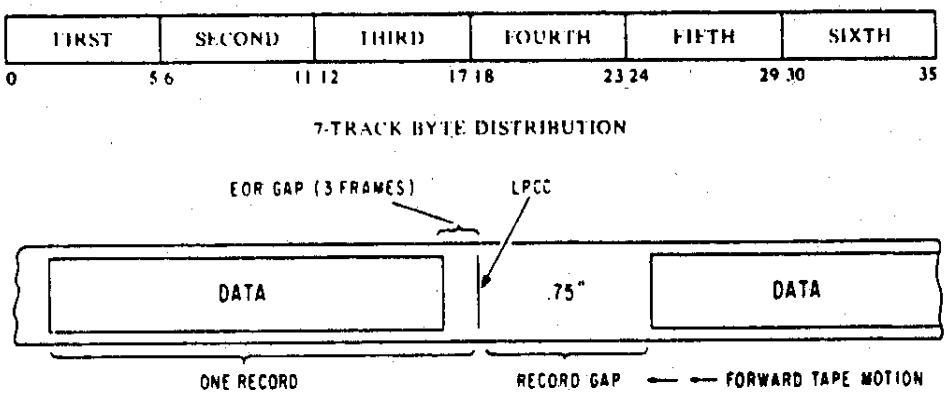
Tapes recorded on some IBM transports have a substandard loadpoint gap of only .5 inch and are thus not compatible with the TM10.

The markers are on the shiny side of the tape; the endpoint marker is against the edge nearer the transport, the loadpoint marker is against the opposite edge.

An annular groove is molded into the back of every reel, and the control cannot write on the tape unless the supply reel has a plastic (write enable) ring in this groove. By leaving the ring out, the operator can protect the data on the tape from accidental destruction (overwriting or erasure).

While the control is actually processing the data portion of a record, the data transfer rate is fixed. However, in a lengthy tape run, the effective (average) transfer rate depends on record length, which determines the percentage of tape taken up by gaps (at the highest density in 7-track format, each record gap could hold 100 additional words). The effective transfer rate is therefore a function of record length as well as tape speed and density.

Full words are transferred between memory and control even though the tape characters may contain 6-bit or 8-bit data bytes. To write, the control divides the words into data bytes, and when reading, the control reassembles the bytes into words. There are several ways in which this is done. For 7-track format, the program can select any density, and the control writes each word as six characters, each containing a 6-bit data byte. After the control writes the last data character for a record, it writes three blank frames (zero



7-TRACK RECORD FORMAT

characters) followed by a longitudinal parity check character (LPCC). The three blank frames constitute the end of record gap (EOR), which is used by the control to detect the end of record. The EOR is used in writing as well as reading since the tape encounters the write head first, and the control detects everything shortly after writing it. The LPCC (which may be zero) produces even parity in each of the tracks along the length of the record. The minimum record gap is .75 inch.

When the control reads or writes a data record, it checks that the (lateral) parity of every data character agrees with the parity specified by the program and checks that every track has even (longitudinal) parity.

For industry compatibility the program must select 800 bpi and odd parity.

The 9-track format is used for recording data compatible with systems based on 8-bit bytes. The program must select a density of 800 bpi, and the control writes bits 0–31 of each word in four characters, ignoring bits 32–35 altogether. The bits from left to right in each 8-bit byte are written in tracks 0–7. After writing the last data character, the control writes an EOR gap, a cyclic redundancy character (CRC), three more blank frames, and an LPCC. The control generates the CRC as described in §6 of USAS X3.22-1967, *USA Standard, Recorded Magnetic Tape for Information Interchange (800 CPI, NRZI)*. Taking the CRC bits as numbered in that document and the track scheme defined above, CRC bit 1 corresponds to the parity track and bits 2–9 correspond to tracks 0–7. The standard record gap is .5 inch minimum, .6 inch nominal, 25 feet maximum.

The difference in nominal gap length between 7-track and 9-track format is due entirely to a difference in head spacing.

When the control reads a record in 9-track format, it assembles data into 36-bit words. Each word is composed of four data bytes in bits 0–31 and the corresponding character parity error indicators in bits 32–35 (*eg* a 1 in bit 33

Characters are assembled into words in this manner by an IDPB loop or an ASCII or ASCIZ pseudoinstruction.

The "tracks" referred to here are simply a convenience for identifying the bits in the data bytes. The actual correspondence of character bits to physical tracks on tape is as follows:

BIT	2	0	4	P	5	6	7	1	3
TRACK	1	2	3	4	5	6	7	8	9

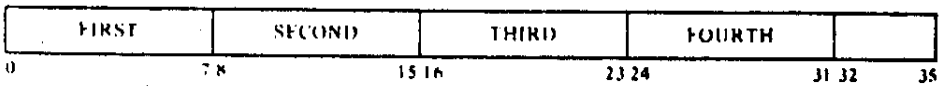
This scheme, which minimizes the effects of errors, is recommended in the standard referenced in the text.

A 1 in an error bit does not necessarily mean the corresponding byte is in error: the error could be in the parity track.

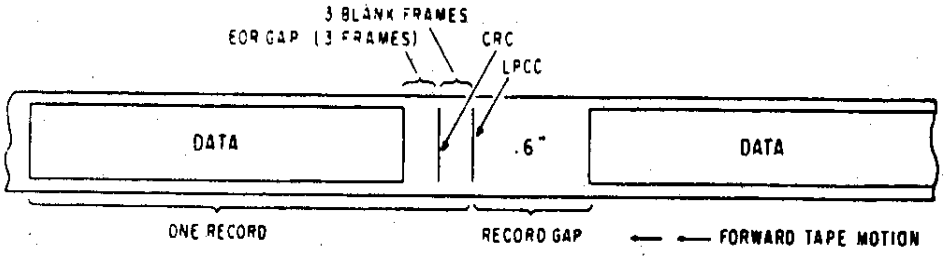
The program must use bit 32 and the parity being checked for in order to regenerate the bit actually read from the parity track of the CRC by the control, keeping in mind that the parity track does not contain an actual parity bit. The parity of the CRC will be odd if the number of data characters in the record is even, otherwise the CRC parity is even.

Errors discovered in a record in core dump format at 800 bpi can be corrected by rereading the record in 9-track format, if investigation of the CRC indicates the errors are confined to a single track. The program must then reconstruct the original words, four from each group of five 4-byte sets supplied by the control.

Two or more contiguous missing characters would be interpreted by the control as an EOR gap. This sets the Bad Tape flag and terminates the function.



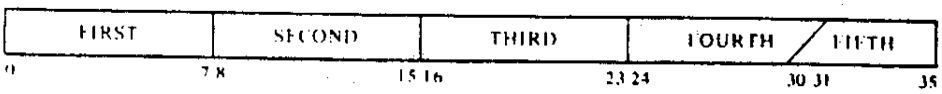
9-TRACK BYTE DISTRIBUTION



9-TRACK RECORD FORMAT

indicates an error in the character from which the byte in bits 8–15 was taken). If errors occur, the program can use the cyclic redundancy character to determine if the errors are confined to a single track, and if so, to correct them. After reading the data characters in a record, the control makes the CRC available in a word in which bits 0–7 contain information from the data tracks and bit 32 contains a parity error bit. The program can correct errors by using the procedure described in Appendix B of the standard. It is not necessary to reread the tape: the error pattern can be generated from the data in memory. If errors are confined to a single track, the program can correct the record by complementing the bit from the bad track in every byte whose error bit is 1.

To facilitate the use of 9-track tape for binary data applications, a core dump format is available in which the program can select any density, and the control uses the 9-track record format but writes full 36-bit words as five characters each. The first four bytes are taken from bits 0–31 of a word in the same manner as in 9-track format; the fifth data byte contains 0s in tracks



CORE DUMP BYTE DISTRIBUTION

0 and 1 and bits 30–35 of the word are contained in tracks 2–7. To reassemble the word during reading, the control ors the overlapping bits. The CRC is written in the usual fashion, but no error bits are supplied with the data bytes.

When writing in even parity in any recording format, the program must not supply a word containing a zero data byte, since this would result in a missing character (a blank frame), and no words beyond that point would be reassembled correctly. The control does not check for missing characters when reading, but such an event always terminates the function at the end of the current record.

Description of the data on the tape

The data consist of two parts:

1.) the active time of the experiment

2.) the scientific results (micrometeoroid impact data)

1.) We have four different active times, in fractions of the total available time, namely for

a) protected data, amplitude range 1

b) unprotected data, amplitude range 1

c) protected data, amplitude ranges 2,3,4

d) unprotected data, amplitude ranges 2,3,4

(for correlation with the data see word 20 and 21 of the scientific data)

The data start with day 349/1974 and end with day 181/1980.

We always took the average of 10 days.

So you have 203 records with respectively 4 words.

2.) For each of the 235 particles we have one record with 47 words.

We only used floating point numbers.

Meaning of the words:

word 1: number of particle

word 2: event-time year

word 3: event-time day of year

word 4: event-time hour

word 5: event-time minute

word 6: event-time second

word 7: 0. if the impact was on the Ecliptic-Sensor

1. if the impact was on the South-Sensor

word 8: lower limit velocity (1-σ)

word 9: most probable impact velocity

word 10: upper limit velocity (1-σ)

word 11: lower limit mass (1-σ)

word 12: most probable particle mass

word 13: 1. if the given mass is a lower limit (overflow data)

0. otherwise

word 14: upper limit mass (1-σ)

word 15: distance to sun (AU) at the time of the impact

word 16: true anomaly of HELIOS 1 at the time of the impact

word 17: sensor azimuth signifies the projected direction of the sensor axis to the ecliptic (90^0 is the direction to sun, 0^0 is perpendicular to the sun in direction of the apex of HELIOS 1)

word 18: 1. if true anomaly and sensor azimuth indicate an eccentric orbit ($e>0.$)

0. otherwise

word 19: total charge in the mass spectrum ($\sim 10^{-13}$ C)

word 20: Q_I-amplitude range

word 21: 1. for protected data

0. for unprotected data

word 22: meanvalue of the reciprocal semi-major axis

word 23: standard-derivation of the reciprocal semi-major axis

word 24: meanvalue of the eccentricity

word 25: standard-derivation of the eccentricity

word 26: meanvalue of the inclination

word 27: standard-derivation of the inclination

word 28: meanvalue of the perihel distance (AU)

word 29: standard-derivation of the perihel distance (AU)

word 30: probability for inbound (≥ 0.75) and outbound (≤ 0.25) trajectory at the moment of the impact. $0.25 <$ word 30 < 0.75 means that the particle was registered near his perihel or aphel. (c.f. K.D.Schmidt and E.Grün, 1979,1980)

word 31: probability of hyperbolic orbit ($e>1$). A hyperbolic orbit is given, if word 31 ≥ 0.75

word 32: shows, for which radiation pressure model (c.f. K.D.Schmidt and E.Grün,1979,1980) the particle has a hyperbolic orbit (word 31 ≥ 0.75)

0. if the particle has a hyperbolic orbit for the models 0,1 and 2

1. if the particle has a hyperbolic orbit for the models 1 and 2

2. if the particle has a hyperbolic orbit only for model 2

3. for no radiation pressure model is word 31 ≥ 0.75

word 33: 1. indicates an overflow of the positiv impact charge Q_I

0. otherwise

word 34: 1. density of eccentric particles (word 18) smaller than 0.1 g/cm^3 (c.f. N.Pailer and E.Grün, 1980 and Grün et al, 1980)

0. otherwise

word 35: chemical classification

1. if undefined spectrum

2. if chondritical spectrum

3. if ironrich spectrum

word 36-47: mass spectrum. The amount of ions is given for 12 massranges from 16 amu to 74 amu. The total amount of ions is normalized to 1 for the massrange 16 amu to 74 amu.

Summary of the organisation of the data on tape:

There are 438 records on tape.

The first 203 records with respectively 4 words (active time).

The last 235 records with respectively 47 words (scientific data).

Further details on the data can be obtained from

E.Grün

Max-Planck-Institut für Kernphysik

Saupfercheckweg 1

D-6900 Heidelberg

Federal Republic of Germany

References

1 "The Distribution of Orbital Elements of Interplanetary Dust in the Inner Solar System as Detected by the Helios Spaceprobe", K.D.Schmidt and E.Grün, Space Research XIX, ed. M.J.Rycroft, Pergamon Press, Oxford and New York, S. 439-442 (1979).

2 "Orbital Elements of Micrometeoroids Detected by the HELIOS 1 Space Probe in the Inner Solar System", K.D.Schmidt and E.Grün, Proceeding of IAU Symposium 90, eds. I.Halliday and B.A.McIntosh, in "Solid Particles in the Solar System", eds. I.Halliday and B.A.McIntosh, S. 321-324 (1980).

3 "The Penetration Limit of Thin Films", N.Pailer and E.Grün, Planet. Space Sci., Vol. 28, S.321-331 (1980).

4 "Orbital and Physical Characteristics of Micrometeoroids in the Inner Solar System as Observed by Helios 1", E.Grün, N.Pailer, H.Fechtig and J.Kissel, Planet. Space Sci., 28, S.333-349 (1980).

Helios-A
12/19/74- 1/02/80

DUMP OF TAPE HELIOS

INPUT TAPE HELIOS ON MT4
DATA INPUT H9 FL 1 15 STOP

FILE	1 RECORD	1 LENGTH	640BYTES						
(0)	00800000	05406418	7308 3F64	9F5E 0C40	7B16 C30C	405F 6F5C	0601 000C	0006 00C0	0000 0540	6A26 170D
(40)	3FC77CCC	06407C8C	2A0C 4068	A2DD 0D01	8000 0006	0080 0000	0540 6D30	D90F 3F7C	2528 0B40	7D6D 1D04
(80)	4068E69C	08018000	0006 0080	0000 0540	5FB8 510B	3F65 F1D4	0740 7566	0E01 405A	E695 0701	8000 0006
(120)	00800000	05000000	0000 0000	0000 0000	0000 0000	0000 0000	0001 8000	0006 0080	0000 0540	4B51 9F0C
(160)	3EEEEFEC	03404F43	1D01 3FF7	68C5 0101	8000 0006	0080 0000	0540 4A29	140E 3EDE	0FF3 0A40	5DC5 EF0D
(200)	40518013	07018000	0006 0080	0000 0540	461D 760B	3DFE 5ACF	0640 5A88	980D 404A	E2D3 0B01	8000 0006
(240)	00800000	053FF5AA	050D 3E6D	6E64 0140	656A 4803	4051 5137	0801 8000	0006 0080	0000 053F	D566 B000
(280)	3ED6AFEC	094071D3	CA0C 405A	A17C 0301	8000 0006	0080 0000	0540 610A	E001 405A	CCC7 0340	5A58 6A04
(320)	405F3220	01018000	0006 0080	0000 0540	5986 970F	4051 C8CF	0840 7025	260C 4063	E3CB 0801	8000 0006
(360)	00800000	05404603	2300 3FDC	F514 0440	6A17 B804	4062 387E	0101 8000	0006 0080	0000 053D	6F32 0600
(400)	3D45203C	0B3DE61D	280E 3DCA	932F 0B01	8000 0006	0080 0000	053D 6869	E502 5CEA	807E 033D	6910 D802
(440)	3D5C5B55	0B018000	0006 0080	0000 053E	541F 7305	3DDA 6696	0F3E 5CDC	B80C 3E4C	C8E0 0A01	8000 0006
(480)	00800000	053F7DB4	7003 3F41	F38D 063F	D3B6 E901	3FCA 8C6F	0F01 8000	0006 0080	0000 053F	D571 EF01
(520)	3F786AE1	0E4059E4	4504 4053	85A4 0601	8000 0006	0080 0000	053F ED4A	630C 3FCE	21BA 0C40	700A A502
(560)	40648EF9	02018000	0006 0080	0000 053F	B16F 3405	3FC6 1F22	0040 5284	FB00 4042	ED3C 0501	8000 0006
(600)	00800000	053FF2B6	BD0A 3FE8	C578 0640	4FB4 E500	4041 DF07	0801 8000	0006 0080	0000 0240	4BE5 980D

FILE	1 RECORD	2 LENGTH	640BYTES						
(0)	01000000	0440461F	E70E 406A	FA82 0740	5E41 B508	0180 0000	0700 8000	0005 404F	8DC9 0B40	41BB 0301
(40)	406460A2	0A405DC2	1005 0180	0000 0600	8000 0005	405C 6043	0D40 4D1B	C1CC 406A	9B38 0640	6314 1108
(80)	01800000	06008000	0005 4058	4B50 0D40	516B 3A05	4064 3D28	0240 5900	1C06 0180	0000 0600	8000 0005
(120)	3EC5CA4A	0A3E6EAA	910D 3ED4	9D94 063E	CDD1 E70F	0180 0000	0600 8000	0005 3F5C	B224 0A3E	E425 1A09
(160)	3FE3BA24	073FD19A	1203 0180	0000 0600	8000 0005	3FF5 D203	083E 660E	D90B 4071	6EE8 0340	5520 E304
(200)	01800000	06008000	0005 3FED	90F4 053E	C08B A109	4079 86B0	0B40 5DD8	3B0E 0180	0000 0600	8000 0005
(240)	404046FF	043FCE9C	0602 406B	F295 0340	5EC8 C406	0180 0000	0600 8000	0005 3FFB	3900 043F	E11F 4107
(280)	4058D771	09405025	CB00 0180	0000 0600	8000 0005	4048 A037	003F E941	8506 406D	2296 0F40	5F33 9D03
(320)	01800000	06008000	0005 4048	2CB2 053F	ED82 1F00	4065 4D0B	0740 579E	7A0D 0180	0000 0600	8000 0005
(360)	404EE38F	033FF20A	330D 4079	E22E 0440	69D0 3A08	0180 0000	0600 8000	0005 4041	150A 023F	D9C7 010E
(400)	407663CA	02406F3B	D702 0180	0000 0600	8000 0005	3FF0 6644	073F C9E2	B107 406F	875E 0040	61AA 7F04
(440)	01800000	06008000	0005 3FF2	A9CD 033F	D5E1 5109	406E EB04	0F40 626A	220F 0180	0000 0600	8000 0005
(480)	3FFCFE5D	013FE2A5	AB04 4076	FE91 0A40	6509 0200	0180 0000	0600 8000	0005 3FFB	D3C1 083F	D3A3 D605
(520)	406BAC16	0C405873	B702 0180	0000 0600	8000 0005	3FC7 F270	0B3F 5210	6602 3FFC	A673 043F	ECA4 920F
(560)	01800000	06008000	0005 3FC8	2FC2 0D3F	56FE 2E03	4049 7567	0F40 40DA	6501 0180	0000 0600	8000 0005
(600)	40418F68	003FD3F3	0803 4076	7276 0E40	6765 2006	0180 0000	0600 8000	0003 3FDF	4D63 083F	7A55 CE05

FILE	1 RECORD	3 LENGTH	640BYTES						
(0)	01000000	034061EC	E807 405A	6C85 0A01	8000 0007	0080 0000	053F D6D3	E603 3FC1	C6DC 033F	E465 4105
(40)	3FDC7AAB	09018000	0006 0080	0000 053F	CEDE 0006	3FCB 5FCE	063F EE3A	B102 3FDD	87970001	8000 0006
(80)	00800000	053FE972	9605 3FD3	1685 0F40	5A5F 0B05	404B 9F4C	0201 8000	0006 0080	0000 053F	C4FD 5605
(120)	3E5996FA	0A4055EB	0503 3FF9	A6D3 0201	8000 0006	0080 0000	053F D206	B602 3DE0	40D8 0940	6F9B 1D0C
(160)	404FF7AD	0B018000	0006 0080	0000 0540	43A9 6F06	3FD3 20F1	0340 7853	B805 4065	0CEF 0501	8000 0006
(200)	00800000	054049AB	4003 3FF4	26C6 0A40	7E07 2D03	4073 0393	0901 8000	0006 0080	0000 053F	FFFD 6301
(240)	3FD8130A	0E407471	D701 4066	57F1 0601	8000 0006	0080 0000	0540 48E5	640A 3FED	EEE0 0E40	6C85 710F
(280)	40615D74	02018000	0006 0080	0000 053F	FAA7 6009	3FD2 34AD	0340 5F0D	4B0B 4057	5DA6 0301	8000 0006
(320)	00800000	053FF903	FE05 3FD0	0683 0240	6A96 2E0A	4064 5B7C	0401 8000	0006 0080	0000 053F	D8D1 2700
(360)	3F73454E	07406A93	B001 4061	A5FB 0801	8000 0006	0080 0000	053F D95A	C703 F6B	3985 0740	4B4C 4009
(400)	404649F8	0D018000	0006 0080	0000 053F	FC13 C706	3FDD 9A6B	0E40 670D	6C09 405F	752D 0301	8000 0006
(440)	00800000	053FED56	0404 3FC9	59D6 0F40	6882 460A	4061 C124	0301 8000	0006 0080	0000 0540	4623 5304
(480)	3FD14A1B	0B4065D7	6400 4060	06E1 0401	8000 0006	0080 0000	053F C58A	FF03 F65	587 60740	43C2 EC0F
(520)	3FFBB2E2	01018000	0006 0080	0000 053E	EFD2 EC0E	3EE7 475A	053E F911	9709 3EF1	2DBA 0C01	8000 0006
(560)	00800000	053ED5A8	FF03 3ECF	F50F 023E	DBCC 850E	3ED8 1D70	0B01 8000	0006 0080	0000 053F	E5B5 B804
(600)	3FC48EA7	0E4040AC	7D03 3FFA	4513 0201	8000 0006	0080 0000	0440 40B0	DC0B 3FC4	9040 0240	56C9 F90A

FILE	1 RECORD	4 LENGTH	640BYTES						
(0)	01000000	02404FE2	2407 0180	0000 0700	8000 0005	3FC1 3486	063F 4E26	FF05 3FDB	8574 0F3F	CF1B D506
(40)	01800000	06008000	0005 3FCD	7148 033E	D858 F307	4076 A6E9	0340 5756	DA0D 01F0	0000 0600	8000 0005
(80)	3FDCF29F	083DEC8A	9C08 4064	B13C 0C40	484A 710F	0180 0000	0600 8000	0005 3FD0	C9CF 013F	4141 B604
(120)	40525AC7	0B4047FB	460B 0180	0000 0600	8000 0005	3FC3 8003	023F 6C3A	6F05 3FED	6750 033F	F244 F00

```
(   160)  01800000  06008000  00054043  DF7E043F  EF4C2409  40708721  0C4061F1  6806010  00000600  80000005
(   200)  4051AABB  0D404647  2D074069  E8380740  5DF8360F  01800000  06008000  00054053  74BA0540  44ECFD0B
(   240)  406C2C55  05406124  49070180  00000600  80000005  404C14B0  0B40440A  150F407C  E6540D40  66F7D9C2
(   280)  01800000  06008000  00054048  7720073F  EECECC04  40750706  0F4063AD  F4050180  00000600  80000005
(   320)  404BFEAA  0E3FF420  F106406D  16F50D40  5DE0460F  01800000  06008000  00053FF8  E582003F  7B135F0C
(   360)  40504CE0  0B40485F  5C000180  00000600  80000005  3FEA7645  0B3F6015  A1034050  1D240740  54711404
(   400)  01800000  06008000  00054042  2061093F  C38E890E  405CE176  0F405069  5C060180  00000600  80000005
(   440)  3FE489CB  003F66A3  690C4058  BA060140  4B78C608  01800000  06008000  00053FE9  E6C003F  60E63D03
(   480)  40486BBD  09404326  F0010180  00000600  80000005  3FFE4720  0F3FC04B  7A074059  31700440  52E7E20F
(   520)  01800000  06008000  00054053  02B10D3F  D4DEE805  406CBD30  08406194  12040180  00000600  80000005
(   560)  405A0621  043FDBD9  61084068  F2460440  5BA77F04  01800000  06008000  00054052  7D480B3F  62D6E009
(   600)  406C5F14  0E4058AE  77DC0180  00000600  80000005  3FD90E92  0C3DDEF7  A5014066  17990B40  43F7400B

FILE      1  RECORD      5  LENGTH     640BYTES
(     0)  01000000  01018000  00070080  0000053F  CFBCF607  3DCA730E  01406A0D  0F023FF7  2351CE01  80000006
(    40)  00800000  053FF23A  40063DD8  27D20840  6FA39A02  4055DD83  0F018000  00060080  00000540  5354250B
(    80)  3FCEDD69  0D406D1E  F7004064  071C0C01  80000006  00800000  05405055  B9003FD2  E4E00E40  7505D30B
(   120)  406DCDB8  05018000  00060080  00000540  5BB13E0A  3FDFAC05  044072C8  100D406B  16BE0901  80000006
(   160)  00800000  0540645E  C5023FEA  16F20340  791EBD0E  406E54E7  0C018000  00060080  00000540  5C9D270A
(   200)  3FE08C12  08407625  1A0A4067  DA980001  80000006  00800000  053FFB25  550E3F7D  9ED70A40  77439C02
(   240)  4068F409  04018000  00060080  00000540  464E7602  3FC7B66E  05407901  3A0B4069  0EAD0501  80000006
(   280)  00800000  053E6CB4  85023DEE  E117023E  F437CA01  3EE0D8C9  04018000  00060080  00000540  40DF1B09
(   320)  3FC22B3F  0F40768B  0307406A  EA130A01  80000006  00800000  05404093  FF0E3FC1  DEA30E40  72B80900
(   360)  40679D70  0A018000  00060080  0000053F  EDAEBA02  3F6FE04D  03407483  44024065  62B40F01  80000006
(   400)  00800000  0540490C  330C3FCA  82030D40  7C248D09  4070A949  07018000  00060080  0000053F  EAC0950E
(   440)  3F6CE328  04406EAB  890B4060  F35D0C01  80000006  00800000  053F4753  4E093EC8  C07E038F  CD79C806
(   480)  3FC29277  0C018000  00060080  0000053F  654D940B  3EE75440  0A4054FF  FC0F404A  0B3A0801  80000006
(   520)  00800000  053FD224  1B093ED2  634D0E40  727B6E00  405C9EC4  0F018000  00060080  C000053F  53664402
(   560)  3D7FB211  0E405963  DE083FDB  AA4D0601  80000006  00800000  053F6BFF  87043D77  85D20D40  6AB5E70A
(   600)  3FFDCEE5  02018000  00060080  0000053E  FB6A3E0D  3CEB260B  0D406D84  3805404A  D4310401  80000006

FILE      1  RECORD      6  LENGTH     640BYTES
(     0)  00800000  053FF5AC  90063F76  756A0A40  692B340B  405D9BD4  03018000  00060080  0000053F  FFCDE007
(    40)  3FC1F9F4  0A405B75  95074051  9E3C0201  80000006  00800000  05404AB9  F5003FCC  661F0040  6B3F9304
(    80)  405FE7F6  0D018000  00060080  00000540  44757C00  3FC612BB  06406DE8  5E02405E  916A0301  80000006
(   120)  00800000  05404F86  D80D3FD1  76920E40  6D387109  40601E55  0E018000  00060080  00000540  51249104
(   160)  3FD46ADF  05407857  520A4067  E8C10601  80000006  00800000  053EE7EB  E4063E6A  4A2D053F  511F6C09
(   200)  3EFC45B1  09018000  00060080  00000500  00000000  00000000  00000000  00000000  00000001  80000006
(   240)  00800000  05000000  00000000  00000000  00000000  00000000  00018000  00060080  00000500  00000000
(   280)  00000000  00000000  00000000  00000001  80000006  00800000  053FE3AC  850F3F65  9E3D0940  45EE3609
(   320)  3FFBAC6F  06018000  00060080  00000540  4B399F06  3FCCB7A7  0640700F  9E0C4063  44F20801  80000006
(   360)  00800000  05404E44  29023FCF  C8850840  6EDC970B  40651900  05018000  00060080  0000053F  DC3FBE07
(   400)  3F5E1211  02404070  8F0F3FED  FF6C0901  80000006  00800000  053FE1C9  000D3F63  D5E6063F  F36FC207
(   440)  3FE39790  0E018000  00060080  0000053F  71BBFB06  3EC49E8B  0A405AF8  D70A4047  08FF0E01  80000006
(   480)  00800000  053F58B0  0A0C3DC0  717C0540  6D4EB104  3FF7FFBD  08018000  00060080  0000053E  CA735907
(   520)  3CECFE45  0D4072F0  D6013FE0  0D050001  80000006  00800000  053F4743  56053D52  2FD50D40  6CFEA70E
(   560)  405023EA  0D018000  00060080  0000053F  C6255400  3F45165E  054074E1  9B064066  FCA30301  80000006
(   600)  00800000  053FC106  14023F42  53000140  75A6C90F  40652021  05018000  00060080  0000023F  E7459501

FILE      1  RECORD      7  LENGTH     640BYTES
(     0)  01000000  043F6956  5509405C  C3CB0540  555CA70A  01800000  07008000  00053FF3  17F4003F  75653B07
(    40)  406DEB2C  00406272  6C0E0180  00000600  80000005  3FE28F97  003F6488  380C4055  A7A2C440  4C0C670A
(    80)  01800000  06008000  00050000  00000000  00000000  00000000  00000180  00000600  80000005
(   120)  00000000  00000000  00000000  00000000  00000000  01800000  06008000  00050000  00000000  00000000
(   160)  00000000  00000000  00000180  00000600  80000005  00000000  00000000  00000000  00000000  00000000
(   200)  01800000  06008000  00050000  00000000  00000000  00000000  00000000  00000180  00000600  80000005
(   240)  00000000  00000000  00000000  00000000  00000000  01800000  06008000  00053FCA  9BB1013F  4C19AF08
(   280)  40477053  00404150  A2090180  00000600  80000005  3FF80668  0C3F6A1B  0408405E  A0650340  57934E00
(   320)  01800000  06008000  00053F68  F8DC063E  EB125106  406F0CB8  0A4063F2  3A0A0180  00000600  80000005
(   360)  3FC5A3E8  0D3F4708  770B406D  81130940  61D2130C  01800000  06008000  00053F5A  0551013E  C44AA20C
(   400)  4078EF91  04405B0B  7C0F0180  00000600  80000005  3E4BD6AE  043CF090  F0013F66  5B080003F  4AA42108
(   440)  01800000  06008000  00053E46  91410130  4FA16007  40425914  013FC8FC  D6060180  00000600  80000005
(   480)  3F4B70F3  043DC5B6  93014061  60C10340  507B7108  01800000  06008000  00053FCA  4EA9C43F  4A2CBA02
(   520)  407C0D87  0A406FF3  1D0F0180  00000600  80000005  3FF555AB  0F3F77AE  6C0E4070  F1340740  6A91CF0F
(   560)  01800000  06008000  00054043  A8F20F3F  C4F16A0C  407CA9D4  0F407540  11070180  00000600  80000005
(   600)  40529ADA  043ED448  8E0E4079  41110F40  72F2AF05  01800000  06008000  00053045  6FCC0A3F  CFE60F05
```

FILE 1 RECORD 8 LENGTH 640BYTES

(0)	01000000	03406E4D	A90B4067	DDC60B01	80000007	008000000	05000000	00000000	00000000	00000000
(40)	00000000	00018000	00060080	00000500	00000000	00000000	00000000	00000000	00000001	80000006
(80)	00800000	05000000	00000000	00000000	00000000	00000000	00018000	00060080	00000500	00000000
(120)	00000000	00000000	00000000	00000001	80000006	008000000	05000000	00000000	00000000	00000000
(160)	00000000	00018000	00060080	00000500	00000000	00000000	00000000	00000000	00000001	80000006
(200)	00800000	05000000	00000000	00000000	00000000	00000000	00018000	00060080	00000500	00000000
(240)	00000000	00000000	00000000	00000001	80000006	008000000	05000000	00000000	00000000	00000000
(280)	00000000	00018000	00060080	0000053F	7733C608	3EF99617	03404127	9402 3FF8	C7C40101	80000006
(320)	00800000	053F51AD	C5013E54	ADC60040	7944200E	405DB1D9	06018000	00060080	0000053F	44EE3D06
(360)	3D76B550	0E40617A	E30F3FF9	4D460701	80000006	008000000	053E5EB3	130F3CD8	BCBE0340	42AEE906
(400)	3F7C2669	09018000	00060080	00000500	EFE22D03	3C76228B	0840421B	4F093FE7	68660301	80000006
(440)	00800000	053FCC25	BB0E3F4D	95460E40	6A0886 0B	40628436	06018000	00060080	00000540	4294AF01
(480)	3FC3E993	0D40738B	A6084068	A2410901	80000006	008000000	053FFE72	3806 3FC0	7CDC0040	72394003
(520)	406BC90A	06018000	00060080	0000053F	6B161F0F	3EED3A68	043FD837	EA043FD3	94F40F01	80000006
(560)	00800000	05000000	00000000	00000000	00000000	00000000	00018000	00060080	00000500	00000000
(600)	00000000	00000000	00000000	00000001	80000006	008000000	04000000	00000000	00000000	00000000

FILE 1 RECORD 9 LENGTH 640BYTES

(0)	01000000	02000000	00000180	00000700	80000005	00000000	00000000	00000000	00000000	00000000
(40)	01800000	06008000	00050000	00000000	00000000	00000000	00000000	00000180	00000600	80000005
(80)	00000000	00000000	00000000	00000000	00000000	01800000	06008000	00050000	00000000	00000000
(120)	00000000	00000000	00000180	00000600	80000005	00000000	00000000	00000000	00000000	00000000
(160)	01800000	06008000	00050000	00000000	00000000	00000000	00000000	00000180	00000600	80000005
(200)	3ED9A332	063E5B6E	230F3F6D	164B003F	6776BF01	01800000	06008000	00053FDF	21CB003F	6108DB06
(240)	40701FF3	00406BE4	B10C0180	00000600	80000005	3FD12927	0E3ED2F3	620D4073	87040340	5440FA06
(280)	01800000	06008000	00053EED	69050930	64643307	3FE9B6CA	053FCF73	E30F0180	00000600	80000005
(320)	3F40EB15	0A3D6360	4A0B4065	3E7F043F	F7595608	01800000	06008000	00053FD0	3CAF0730	D2B9E401
(360)	4071AA4A	0E40523C	F2030180	00000600	80000005	3FEE3E13	0E3F6AA0	210D4077	E72D0F40	6CA6660E
(400)	01800000	06008000	00053FFC	B4100930	7F34DD05	4072063C	00406A77	C0050180	00000600	80000005
(440)	40505671	0B3FD1F1	C50E406D	CF3C0C40	677B860D	01800000	06008000	00054040	870E0A3F	C14EE00D
(480)	40679220	05406107	5B050180	00000600	80000005	00000000	00000000	00000000	00000000	00000000
(520)	01800000	06008000	00050000	00000000	00000000	00000000	00000000	00000180	00000600	80000005
(560)	00000000	00000000	00000000	00000000	00000000	01800000	06008000	00050000	00000000	00000000
(600)	00000000	00000000	00000180	00000600	80000005	00000000	00000000	00000000	00000000	00000000

FILE 1 RECORD 10 LENGTH 640BYTES

(0)	01000000	01018000	00070080	00000500	00000000	00000000	00000000	00000000	00000001	80000006
(40)	00800000	05000000	00000000	00000000	00000000	00000000	00018000	00060080	00000500	00000000
(80)	00000000	00000000	00000000	00000001	80000006	008000000	05000000	00000000	00000000	00000000
(120)	00000000	00018000	00060080	00000500	00000000	00000000	00000000	00000000	00000001	80000006
(160)	00800000	053EC5EB	CD033E47	51CC023F	D3135604	3FC92D43	08018000	00060080	0000053F	616EC00D
(200)	3ED9045A	02407DB7	C4054060	1CCC0601	80000006	008000000	053F62EC	B8083EEF	7BB40B40	67EE0B04
(240)	3FF6AEAB	07018000	00060080	0000053F	63ABF103	3EF68C55	0A405AA6	720B3FD6	E5C00601	80000006
(280)	00800000	053F42BE	47023E5E	EE300B40	62764B04	40485F53	00018000	00060080	0000053F	CFB33A00
(320)	3F514B4A	09407C88	9D0B406C	53890601	80000006	008000000	053EF8F0	D3023F7B	5C0A0A40	5D74BA06
(360)	4056BDD8	0D018000	00060080	00030000	C0000000	43CA0000	00440840	00004278	00000043	42000000
(400)	43680000	00000000	000041C0	00000042	D0000000	43E40000	0026D3AE	6A0C294E	73840100	00000000
(440)	2BC98C4B	0D407D4F	DF0443D3	947A0E44	66A80000	00000000	00437EBA	A7014000	00000040	C0000000
(480)	406519A5	023FCF6A	560D3FDB	3DF8013F	C2224D0F	42C67B3D	0C427F35	EA0E4071	291209 3E	F0E23903
(520)	405F1315	03404B1D	6303A140	00000000	00000000	C0000000	0040C000	00004047	BEBA0F3F	5E25ED0A
(560)	3E4A1FFA	043E4156	26013EE6	32000 23E	49B50602	3D5D46F5	093D546D	5B0E3D4F	82290D3D	4930FD01
(600)	3D41121C	073CD078	9A060180	00030100	80000005	41400000	0043CA00	00004409	80000042	C3000000

FILE 1 RECORD 11 LENGTH 640BYTES

(0)	01000002	0C42F000	000041C0	00000040	C0000000	42C89F8F	0C42F7D4	13094362	DBC3052B	C4359A0E
(40)	2CF75DCF	01000000	00002E68	72150340	79F07809	43D7A33E	0644CF74	00000000	00000044	4B6F5507
(80)	41600000	0040C000	000040D9	A8230F40	424A960F	40594ED1	083F78C4	870B43D4	D2D70443	5734E507
(120)	3FE3B835	0A3FD41A	3B0B405A	93AE093F	5EC59C0B	41600000	00000000	00000000	00000041	40000000
(160)	3EFA0E58	0C3FE417	F6063E64	733D0A3D	63801B06	3DFB0BE1	083E6273	F4023DD1	6E5E073E	62CC2102
(200)	3EDEEF79	0F3E6CE9	460E3ECD	ED9A013E	7C4EB70F	01800003	02008000	03004160	C0000043	CA000000
(240)	44DB4000	0042CC00	00004268	00000042	C0000000	40C00000	00437249	240943E4	00000044	57800000
(280)	27482A05	02287E49	880F0000	00000029	EE805700	407950BF	0243DCDF	8307 44CF	74000000	00000000
(320)	44D1F480	0540C000	000040C0	00000040	72938A0A	3FFDB4E6	094057A8	430C3F7E	DC960C44	41639309
(360)	42F880C2	024060FB	FB003F6B	B17F0E40	62429C05	4077F020	0C000000	00000000	00000000	00000000

01000001101101000
211 541₈
353

01000011010010100
207 112₈
151

01000010110011010
211 546₈
350

```
(  160)  43620000  0042EC00  00000000  00000042  C2E0720A  42EE5[illegible]  02435B09  35C42DC3  4746042C  F5[illegible]703
(  200)  40C00000  00304063  3C0F3FD3  EDC60E44  7022C70C  43436000  00000000  0000466E  E1120341  40000000
(  240)  40C00000  004151B9  7A0A40D0  0E940C3F  F2D70409  3F7CE6AC  0142CA42  620F4279  74520E3F  C134[illegible]004
(  280)  3ED623FA  033FE508  ED093EE3  9BE70641  60000000  00000000  00000000  00004140  0000003B  FA779505
(  320)  3E4821B5  023FD0CB  A10E3E77  C308[illegible]0C3E  D9BF7202  3EF97A18  043EF13A  0E0C3EEC  2B6D0D3E  C2[illegible]CEC0B
(  360)  3E49BEE5  013E4BA1  47093DF4  70CA0401  [illegible]0000301  008[illegible]00003  00427000  000043CB  00000043  CA000000
(  400)  41D00000  00434C00  000042E4  00000040  C0000000  42E3FD[illegible]6  0743527D  F50543C4  0E500D27  D350070A
(  440)  2948E606  0B000000  00002A7F  92[illegible]B0B3F  CF3[illegible]510A  44C01C[illegible]E  0543C650  000040C0  00000043  D43FFB0D
(  480)  40C00000  0040C000  00004055  EA000F3F  F8389E0D  40721C42  003EDA08  C6004204  [illegible]EA30442  7[illegible]9[illegible]F50E
(  520)  3F7EE6C6  043E6833  D50B3D48  7D680840  7AE22E08  40C00000  00000000  00000000  00000040  C0000000
(  560)  3D5C2D7E  083E7F72  4A0C3EF9  5C75023E  4EF6D901  3ED816D4  053F4F8C  27023F70  F364073F  4D6D9E05
(  600)  3ECF113C  0F3E5868  2A0E3D76  C80A0B3D  D7673007  01800003  01008000  000342[illegible]  00000043  CE000000
```

EOJ STOP REQUESTED IN FILE 1

EOJ DUMP STOPPED AFTER FILE 1 # OF PERMANENT READ ERRORS 0

START TIME 11/12/81 09:05:58 STOP TIME 11/12/81 09:06:17

$$