# IDENTIFICATION

PRODUCT CODE: DEC-11-YPTC-D

PRODUCT NAME: TC11 DECTAPE FORMATTER

DATE CREATED: JANUARY 22, 1972

MAINTAINER: COMPUTER SPECIAL SYSTEMS

AUTHOR: ROBERT J. COLLINS

## 1.Ø ABSTRACT

The TC11 DECtape Formatter is a program for marking DECtapes on Unit Ø with a standard format of $578_{10}$ blocks of $256_{10}$ words each.

## 2.Ø REQUIREMENTS

A. Minimum configuration PDP-11

B. TC11 DECtape control and at least one transport.

C. DECtape(s)

## 3.Ø LOADING PROCEDURE

The normal procedure for loading Absolute Binary tapes should be followed.

## 4.Ø PRELIMINARY OPERATIONS

A. Mount a DECtape on Unit Ø, wrapping enough turns to insure tension.

B. Set the transport On-line with Write Enable on.

C. Set the TC11 WALL and WRTM switches (up).

## 5.Ø STARTING PROCEDURE

Load address $6ØØ_8$ and press START.

## 6.Ø OPERATING PROCEDURE

A. Once started the program writes all Timing and Mark Track information in one forward pass and then halts at address $1Ø5Ø_8$.

B. If the tape has run off the reel, remount it. Reset the WRTM switch (down) and press CONTINUE. The tape will back out for three blocks, then turn around and write the complement obverse of the last block number ($11\emptyset1_8$) in Writeall in all data slots until the End Zone is sensed. The tape then reverses and writes relevant information in all blocks down through $\emptyset$ in Writeall.

C. The tape then reverses and alternately searches for each block number in sequence then changes to Readall to check each block for correct data ($\emptyset$'s).

D. After all blocks have been verified forward, the tape is reversed again and alternately searches for each block number in reverse sequence then writes each block to $\emptyset$'s in Write Data mode. This insures parity conservation if the tape is to be read before being rewritten.

E. The program then halts at address $2316_8$. To format another tape, perform the operations in section 4.$\emptyset$ then press CONTINUE.

## 7.$\emptyset$ ERRORS

### 7.1 ERROR HALT

Address $3\emptyset12_8$ is the common error halt. Press CONTINUE to proceed from it.

## 7.2 ERROR SWITCHES

SR15(1) Inhibit error halt

SR14(1) Inhibit error printout

## 7.3 ERROR PRINTOUTS

Error printouts consist of:

A. Address of error.

B. Address of subroutine call if error is in a subroutine.

C. Processor status at time of error.

D. An error message.

E. Optional relevant data (Good, Bad, etc.)

```
                        ;TOG-11  V,03
                        ;TC11 DECTAPE FORMATTER
                        ;ROBERT J. COLLINS
                        ;JUNE 27, 1970
                        ;MODIFIED MAY 12,1971
                        ;MODIFIED JANUARY 22,1972

                        ;FORMAT DECTAPES ON UNIT 0 IN STANDARD FORMAT OF
                        ;578(10) BLOCKS WITH 256(10) 18 BIT WORDS PER BLOCK.

000000                  R0=%0                           ;REGISTER DEFINITIONS
000001                  R1=%1
000002                  R2=%2
000003                  R3=%3
000004                  R4=%4
000005                  R5=%5
000006                  R6=%6
000007                  R7=%7
000006                  SP=R6
000007                  PC=R7
000000                  XX=HALT
177570                  SR=177570                       ;SWITCH REGISTER
177776                  PS=177776                       ;PROCESSOR STATUS
177776                  CC=PS
000240                  NOP=240
000000                  P0=0                            ;PI PRIORITY LEVELS
000040                  P1=40
000100                  P2=100
000140                  P3=140
000200                  P4=200
000240                  P5=240
000300                  P6=300
000340                  P7=340
000001                  B0=1                            ;BIT POSITION CONSTANTS
000002                  B1=2
000004                  B2=4
000010                  B3=10
000020                  B4=20
000040                  B5=40
000100                  B6=100
000200                  B7=200
```

```
000400                  B8=400
001000                  B9=1000
002000                  B10=2000
004000                  B11=4000
010000                  B12=10000
020000                  B13=20000
040000                  B14=40000
100000                  B15=100000
005726                  POP=5726                ;POP STACK ONCE, =TST(6)+
022626                  POPPOP=22626            ;POP STACK TWICE, =CMP (6)+,(6)+
000015                  CR=15                   ;CARRIAGE RETURN
000012                  LF=12                   ;LINE FEED
177560                  TKS=177560              ;KEYBOARD STATUS
177562                  TKB=177562              ;KEYBOARD BUFFER
177564                  TPS=177564              ;TELEPRINTER STATUS
177566                  TPB=177566              ;TELEPRINTER BUFFER
000060                  TKV=60                  ;KEYBOARD VECTOR
000064                  TPV=64                  ;TELEPRINTER VECTOR
177342                  TCCM=177342
177340                  TCST=177340
177350                  TCDT=177350
177346                  TCBA=177346
177344                  TCWC=177344
104000                  ERR=EMT
104400                  ER$=TRAP
000002                  CB=B1
000004                  ST=B2
000010                  DA=B3
000020                  BK=B4
000040                  VC=B5
```

```
                        .MACR   AND A,B
                        BIC     #-A-1,B         ;MASK A APPLIED TO B
                        .ENDM


                        .MACR   ROTR A
                        JSR     PC,RTR          ;ROTATE R0 A PLACES RIGHT
                        .WORD   A
                        .ENDM

                        .MACR   ROTL A
                        JSR     PC,RTL          ;ROTATE R0 A PLACES LEFT
                        .WORD   A
                        .ENDM

                        .MACR   LOOP    A,B
                        JSR     PC,LERCHKA      ;LOOP TO A ON  ERROR IF SR13=1
                        JMP     A
                B1      JSR     PC,LUPCHK       ;LOOP TO A IF SR12=1
                        JMP     A
                        .ENDM

                        .MACR   PNTM A
                        MOV     #A,R0           ;PRINT MESSAGE
                        JSR     PC,TYPOUT       ;POINTED TO BY A
                        .ENDM
```

```
                000000                  .=0

                000140                  .REPT   140
                                        .+2
                                        20367           ;SET TRACE BIT WITH A DUMMY COMPARE
                                        .ENDR

000000  000002                         .+2
000002  020367                         20367           ;SET TRACE BIT WITH A DUMMY COMPARE
000004  000006                         .+2
000006  020367                         20367           ;SET TRACE BIT WITH A DUMMY COMPARE
000010  000012                         .+2
000012  020367                         20367           ;SET TRACE BIT WITH A DUMMY COMPARE
000014  000016                         .+2
000016  020367                         20367           ;SET TRACE BIT WITH A DUMMY COMPARE
000020  000022                         .+2
000022  020367                         20367           ;SET TRACE BIT WITH A DUMMY COMPARE
000024  000026                         .+2
000026  020367                         20367           ;SET TRACE BIT WITH A DUMMY COMPARE
000030  000032                         .+2
000032  020367                         20367           ;SET TRACE BIT WITH A DUMMY COMPARE
000034  000036                         .+2
000036  020367                         20367           ;SET TRACE BIT WITH A DUMMY COMPARE
000040  000042                         .+2
000042  020367                         20367           ;SET TRACE BIT WITH A DUMMY COMPARE
000044  000046                         .+2
000046  020367                         20367           ;SET TRACE BIT WITH A DUMMY COMPARE
000050  000052                         .+2
000052  020367                         20367           ;SET TRACE BIT WITH A DUMMY COMPARE
000054  000056                         .+2
000056  020367                         20367           ;SET TRACE BIT WITH A DUMMY COMPARE
000060  000062                         .+2
000062  020367                         20367           ;SET TRACE BIT WITH A DUMMY COMPARE
000064  000066                         .+2
000066  020367                         20367           ;SET TRACE BIT WITH A DUMMY COMPARE
000070  000072                         .+2
000072  020367                         20367           ;SET TRACE BIT WITH A DUMMY COMPARE
000074  000076                         .+2
000076  020367                         20367           ;SET TRACE BIT WITH A DUMMY COMPARE
000100  000102                         .+2
000102  020367                         20367           ;SET TRACE BIT WITH A DUMMY COMPARE
000104  000106                         .+2
000106  020367                         20367           ;SET TRACE BIT WITH A DUMMY COMPARE
000110  000112                         .+2
000112  020367                         20367           ;SET TRACE BIT WITH A DUMMY COMPARE
000114  000116                         .+2
000116  020367                         20367           ;SET TRACE BIT WITH A DUMMY COMPARE
000120  000122                         .+2
000122  020367                         20367           ;SET TRACE BIT WITH A DUMMY COMPARE
000124  000126                         .+2
000126  020367                         20367           ;SET TRACE BIT WITH A DUMMY COMPARE
000130  000132                         .+2
000132  020367                         20367           ;SET TRACE BIT WITH A DUMMY COMPARE
000134  000136                         .+2
```

```
000136  020367          20367          ;SET TRACE BIT WITH A DUMMY COMPARE
000140  000142          ,.+2
000142  020367          20367          ;SET TRACE BIT WITH A DUMMY COMPARE
000144  000146          ,.+2
000146  020367          20367          ;SET TRACE BIT WITH A DUMMY COMPARE
000150  000152          ,.+2
000152  020367          20367          ;SET TRACE BIT WITH A DUMMY COMPARE
000154  000156          ,.+2
000156  020367          20367          ;SET TRACE BIT WITH A DUMMY COMPARE
000160  000162          ,.+2
000162  020367          20367          ;SET TRACE BIT WITH A DUMMY COMPARE
000164  000166          ,.+2
000166  020367          20367          ;SET TRACE BIT WITH A DUMMY COMPARE
000170  000172          ,.+2
000172  020367          20367          ;SET TRACE BIT WITH A DUMMY COMPARE
000174  000176          ,.+2
000176  020367          20367          ;SET TRACE BIT WITH A DUMMY COMPARE
000200  000202          ,.+2
000202  020367          20367          ;SET TRACE BIT WITH A DUMMY COMPARE
000204  000206          ,.+2
000206  020367          20367          ;SET TRACE BIT WITH A DUMMY COMPARE
000210  000212          ,.+2
000212  020367          20367          ;SET TRACE BIT WITH A DUMMY COMPARE
000214  000216          ,.+2
000216  020367          20367          ;SET TRACE BIT WITH A DUMMY COMPARE
000220  000222          ,.+2
000222  020367          20367          ;SET TRACE BIT WITH A DUMMY COMPARE
000224  000226          ,.+2
000226  020367          20367          ;SET TRACE BIT WITH A DUMMY COMPARE
000230  000232          ,.+2
000232  020367          20367          ;SET TRACE BIT WITH A DUMMY COMPARE
000234  000236          ,.+2
000236  020367          20367          ;SET TRACE BIT WITH A DUMMY COMPARE
000240  000242          ,.+2
000242  020367          20367          ;SET TRACE BIT WITH A DUMMY COMPARE
000244  000246          ,.+2
000246  020367          20367          ;SET TRACE BIT WITH A DUMMY COMPARE
000250  000252          ,.+2
000252  020367          20367          ;SET TRACE BIT WITH A DUMMY COMPARE
000254  000256          ,.+2
000256  020367          20367          ;SET TRACE BIT WITH A DUMMY COMPARE
000260  000262          ,.+2
000262  020367          20367          ;SET TRACE BIT WITH A DUMMY COMPARE
000264  000266          ,.+2
000266  020367          20367          ;SET TRACE BIT WITH A DUMMY COMPARE
000270  000272          ,.+2
000272  020367          20367          ;SET TRACE BIT WITH A DUMMY COMPARE
000274  000276          ,.+2
000276  020367          20367          ;SET TRACE BIT WITH A DUMMY COMPARE
000300  000302          ,.+2
000302  020367          20367          ;SET TRACE BIT WITH A DUMMY COMPARE
000304  000306          ,.+2
000306  020367          20367          ;SET TRACE BIT WITH A DUMMY COMPARE
000310  000312          ,.+2
000312  020367          20367          ;SET TRACE BIT WITH A DUMMY COMPARE
```

```
000314  000316                  ,+2
000316  020367                  20367           ;SET TRACE BIT WITH A DUMMY COMPARE
000320  000322                  ,+2
000322  020367                  20367           ;SET TRACE BIT WITH A DUMMY COMPARE
000324  000326                  ,+2
000326  020367                  20367           ;SET TRACE BIT WITH A DUMMY COMPARE
000330  000332                  ,+2
000332  020367                  20367           ;SET TRACE BIT WITH A DUMMY COMPARE
000334  000336                  ,+2
000336  020367                  20367           ;SET TRACE BIT WITH A DUMMY COMPARE
000340  000342                  ,+2
000342  020367                  20367           ;SET TRACE BIT WITH A DUMMY COMPARE
000344  000346                  ,+2
000346  020367                  20367           ;SET TRACE BIT WITH A DUMMY COMPARE
000350  000352                  ,+2
000352  020367                  20367           ;SET TRACE BIT WITH A DUMMY COMPARE
000354  000356                  ,+2
000356  020367                  20367           ;SET TRACE BIT WITH A DUMMY COMPARE
000360  000362                  ,+2
000362  020367                  20367           ;SET TRACE BIT WITH A DUMMY COMPARE
000364  000366                  ,+2
000366  020367                  20367           ;SET TRACE BIT WITH A DUMMY COMPARE
000370  000372                  ,+2
000372  020367                  20367           ;SET TRACE BIT WITH A DUMMY COMPARE
000374  000376                  ,+2
000376  020367                  20367           ;SET TRACE BIT WITH A DUMMY COMPARE
000400  000402                  ,+2
000402  020367                  20367           ;SET TRACE BIT WITH A DUMMY COMPARE
000404  000406                  ,+2
000406  020367                  20367           ;SET TRACE BIT WITH A DUMMY COMPARE
000410  000412                  ,+2
000412  020367                  20367           ;SET TRACE BIT WITH A DUMMY COMPARE
000414  000416                  ,+2
000416  020367                  20367           ;SET TRACE BIT WITH A DUMMY COMPARE
000420  000422                  ,+2
000422  020367                  20367           ;SET TRACE BIT WITH A DUMMY COMPARE
000424  000426                  ,+2
000426  020367                  20367           ;SET TRACE BIT WITH A DUMMY COMPARE
000430  000432                  ,+2
000432  020367                  20367           ;SET TRACE BIT WITH A DUMMY COMPARE
000434  000436                  ,+2
000436  020367                  20367           ;SET TRACE BIT WITH A DUMMY COMPARE
000440  000442                  ,+2
000442  020367                  20367           ;SET TRACE BIT WITH A DUMMY COMPARE
000444  000446                  ,+2
000446  020367                  20367           ;SET TRACE BIT WITH A DUMMY COMPARE
000450  000452                  ,+2
000452  020367                  20367           ;SET TRACE BIT WITH A DUMMY COMPARE
000454  000456                  ,+2
000456  020367                  20367           ;SET TRACE BIT WITH A DUMMY COMPARE
000460  000462                  ,+2
000462  020367                  20367           ;SET TRACE BIT WITH A DUMMY COMPARE
000464  000466                  ,+2
000466  020367                  20367           ;SET TRACE BIT WITH A DUMMY COMPARE
000470  000472                  ,+2
```

```
000472  020367                  20367           ;SET TRACE BIT WITH A DUMMY COMPARE
000474  000476                  ,+2
000476  020367                  20367           ;SET TRACE BIT WITH A DUMMY COMPARE
000500  000502                  ,+2
000502  020367                  20367           ;SET TRACE BIT WITH A DUMMY COMPARE
000504  000506                  ,+2
000506  020367                  20367           ;SET TRACE BIT WITH A DUMMY COMPARE
000510  000512                  ,+2
000512  020367                  20367           ;SET TRACE BIT WITH A DUMMY COMPARE
000514  000516                  ,+2
000516  020367                  20367           ;SET TRACE BIT WITH A DUMMY COMPARE
000520  000522                  ,+2
000522  020367                  20367           ;SET TRACE BIT WITH A DUMMY COMPARE
000524  000526                  ,+2
000526  020367                  20367           ;SET TRACE BIT WITH A DUMMY COMPARE
000530  000532                  ,+2
000532  020367                  20367           ;SET TRACE BIT WITH A DUMMY COMPARE
000534  000536                  ,+2
000536  020367                  20367           ;SET TRACE BIT WITH A DUMMY COMPARE
000540  000542                  ,+2
000542  020367                  20367           ;SET TRACE BIT WITH A DUMMY COMPARE
000544  000546                  ,+2
000546  020367                  20367           ;SET TRACE BIT WITH A DUMMY COMPARE
000550  000552                  ,+2
000552  020367                  20367           ;SET TRACE BIT WITH A DUMMY COMPARE
000554  000556                  ,+2
000556  020367                  20367           ;SET TRACE BIT WITH A DUMMY COMPARE
000560  000562                  ,+2
000562  020367                  20367           ;SET TRACE BIT WITH A DUMMY COMPARE
000564  000566                  ,+2
000566  020367                  20367           ;SET TRACE BIT WITH A DUMMY COMPARE
000570  000572                  ,+2
000572  020367                  20367           ;SET TRACE BIT WITH A DUMMY COMPARE
000574  000576                  ,+2
000576  020367                  20367           ;SET TRACE BIT WITH A DUMMY COMPARE

        000014                  ,=14

000014  003166                  TRCTRP
000016  000340                  P7

        000030                  ,=30

000030  002570                  ERROR
000032  000340                  P7
000034  002570                  ERROR
000036  000340                  P7
```

```
                000600                          .=600

000600  000005                 START:  RESET                   ;CLEAR THE WORLD
000602  012706  000600                 MOV     #START,SP       ;INITIALIZE THE STACK POINTER
000606  012767  000340  177162         MOV     #P7,PS          ;LOCKOUT PI BY RAISING CPU PRIORITY
000614  012700  101101                 MOV     #101101,R0      ;REVERSE END ZONE=55
000620  012704  170000                 MOV     #-4096.,R4      ;COUNTER FOR REZ
000624  012767  000013  176510         MOV     #13,TCCM        ;GO
000632  032767  100200  176502 LP1:    BIT     #B15+B7,TCCM    ;WAIT FOR ERROR OR READY
000640  001774                        BEQ     .-6             ;HAVE NEITHER
000642  100002                        BPL     .+6             ;HAVE READY
000644  004767  002404                JSR     PC,SERROR       ;STATUS ERROR
000650  010067  176474                MOV     R0,TCDT         ;LOAD MARK TRACK
000654  005204                        INC     R4              ;DONE ALL?
000656  001365                        BNE     LP1             ;LOOP IF NOT FINISHED
000660  012704  177471                MOV     #-199.,R4       ;COUNTER FOR IBS
```

```
000664  012700  010101            MOV     #10101,R0       ;INTERBLOCK SYNC=25
000670  032767  100200  176444    LP2:    BIT     #B15+B7,TCCM    ;WAIT FOR ERROR OR READY
000676  001774                    BEQ     .-6             ;HAVE NEITHER
000700  100002                    BPL     .+6             ;HAVE READY
000702  004767  002346            JSR     PC,SERROR       ;STATUS ERROR
000706  010067  176436            MOV     R0,TCDT         ;LOAD MARK CODE
000712  005204                    INC     R4              ;DONE ALL?
000714  001365                    BNE     LP2             ;IF NOT, KEEP LOADING
000716  012703  176676            MOV     #-578.,R3       ;COUNTER FOR NUMBER OF BLOCKS
000722  004767  000130    LP3:    JSR     PC,FRMBLK       ;WRITE CODES FOR ONE BLOCK
000726  005203                    INC     R3              ;DONE ALL?
000730  001374                    BNE     LP3             ;IF NOT, WRITE ANOTHER BLOCK
000732  012704  177471            MOV     #-199.,R4       ;COUNTER FOR IBS
000736  012700  010101            MOV     #10101,R0       ;INTERBLOCK SYNC=25
000742  032767  100200  176372    LP4:    BIT     #B15+B7,TCCM    ;WAIT FOR ERROR OR READY
000750  001774                    BEQ     .-6             ;HAVE NEITHER
000752  100002                    BPL     .+6             ;HAVE READY
000754  004767  002274            JSR     PC,SERROR       ;STATUS ERROR
000760  010067  176364            MOV     R0,TCDT         ;LOAD MARK CODE
000764  005204                    INC     R4              ;DONE ALL?
000766  001365                    BNE     LP4             ;IF NOT, KEEP WRITING IBS
000770  012704  170000            MOV     #-4096.,R4      ;COUNTER FOR FEZ
```

```
000774  012700  010010            MOV     #10010,R0       ;FORWARD END ZONE=22
001000  032767  100200  176334    LP5:    BIT     #B15+B7,TCCM    ;WAIT FOR ERROR OR READY
001006  001774                    BEQ     .-6             ;HAVE NEITHER
001010  100002                    BPL     .+6             ;HAVE READY
001012  004767  002236            JSR     PC,SERROR       ;STATUS ERROR
001016  010067  176326            MOV     R0,TCDT         ;LOAD MARK CODE
001022  005204                    INC     R4              ;DONE ALL?
001024  001365                    BNE     LP5             ;IF NOT KEEP WRITING FEZ
001026  032767  100200  176306    BIT     #B15+B7,TCCM    ;WAIT FOR ERROR OR READY
001034  001774                    BEQ     .-6             ;HAVE NEITHER
001036  100002                    BPL     .+6             ;HAVE READY
001040  004767  002210            JSR     PC,SERROR       ;STATUS ERROR
001044  005067  176272            CLR     TCCM            ;STOP TAPE, MARK & TIMING TRACKS ARE
001050  000000                HLT1:   XX                  ;WRITTEN, RESET WRTM SWITCH AND
001052  000167  000044            JMP     WRTBNO          ;SET WALL SWITCH THEN CONTINUE,
```

```
001056  012704  177366          FRMBLK: MOV     #-266.,R4       ;COUNTER FOR 1 BLOCK'S WORTH OF MKTRACK
001062  012700  005506                  MOV     #BLKMRK,R0      ;POINTER TO MARK CODE TABLE
001066  032767  100200  176246  LP6:    BIT     #B15+B7,TCCM    ;WAIT FOR ERROR OR READY
001074  001774                        BEQ     .-6             ;HAVE NEITHER
001076  100002                        BPL     .+6             ;HAVE READY
001100  004767  002150                JSR     PC,SERROR       ;STATUS ERROR
001104  011067  176240                MOV     (R0),TCDT       ;FETCH MARK CODE
001110  062700  000002                ADD     #2,R0           ;ADVANCE POINTER
001114  005204                      INC     R4              ;DONE ALL?
001116  001363                      BNE     LP6             ;IF NOT FETCH ANOTHER CODE
001120  000207                      RTS     PC              ;IF SO, EXIT

001122  004767  001232          WRTBNO: JSR     PC,COBLKG       ;GENERATE TABLE OF CMPOBV BLK NUMBERS
001126  012700  177775                  MOV     #-3,R0          ;THREE BLOCKS
001132  012767  004002  176202          MOV     #4002,TCCM      ;SEARCH REVERSE
001140  105267  176176          WLP1:   INCB    TCCM            ;GO
001144  032767  100200  176170          BIT     #B15+B7,TCCM    ;WAIT FOR ERROR OR READY
001152  001774                        BEQ     .-6             ;HAVE NEITHER
001154  100002                        BPL     .+6             ;HAVE READY
001156  004767  002072                JSR     PC,SERROR       ;STATUS ERROR
001162  005200                      INC     R0              ;FOUND THREE?
001164  001365                      BNE     WLP1            ;IF NOT KEEP SEARCHING
001166  012700  076677                MOV     #76677,R0       ;COMPLEMENT OBVERSE OF 1011 (LAST BLOCK#)
001172  012767  000017  176142          MOV     #17,TCCM        ;WRITEALL FORWARD
001200  032767  100200  176134          BIT     #B15+B7,TCCM    ;WAIT FOR ERROR OR READY
001206  001774                        BEQ     .-6             ;HAVE NEITHER
001210  100002                        BPL     .+6             ;HAVE READY
001212  004767  002036                JSR     PC,SERROR       ;STATUS ERROR
001216  152767  000003  176114  WLP2:   BISB    #3,TCST         ;SET XDA BITS FOR 0 REVERSE
001224  010067  176120                MOV     R0,TCDT         ;WRITE LAST BLOCK # FROM HERE TO END ZONE
001230  004767  001070                JSR     PC,FLAG1        ;WAIT TO SYNC UP FOR WRITING
001234  000770                      BR      WLP2            ;KEEP WRITING 1011
001236  012767  001101  001306          MOV     #1101,FBLK      ;INITIALIZE PARAMETERS FOR LAST BLOCK,
001244  012705  002560                MOV     #BYST,R5        ;XDA BIT TABLE AND
001250  012703  006532                MOV     #RBN,R3         ;TABLE OF REVERSE BLOCK NUMBERS
001254  012767  004017  176060          MOV     #4017,TCCM      ;WRITEALL REVERSE
001262  005367  001264          WALUP:  DEC     FBLK            ;DECREMENT BLOCK NUMBER
001266  022767  177776  001256          CMP     #-2,FBLK        ;LAST BLOCK WRITTEN?
001274  001531                        BEQ     CHECK           ;YES- CHECK DATA AND BLOCK NUMBERS

001276  012704  177777                  MOV     #-1,R4          ;-1 REVERSE =0 FWD
001302  012767  177374  003352          MOV     #-260.,TOG1     ;NUMBER OF WORDS
001310  032767  100200  176024  WLP3:   BIT     #B15+B7,TCCM    ;WAIT FOR ERROR OR READ
001316  001774                        BEQ     .-6             ;HAVE NEITHER
001320  100002                        BPL     .+6             ;HAVE READY
001322  004767  001726                JSR     PC,SERROR       ;STATUS ERROR
001326  152767  000003  176004          BISB    #3,TCST         ;1 KBV=0 FWD
001334  010467  176010                MOV     R4,TCDT         ;LOAD DATA WORD
001340  005267  003316                INC     TOG1            ;WRITE IT 260 TIMES
001344  001361                      BNE     WLP3            ;KEEP WRITING UNTIL FINISHED
001346  032767  100200  175766          BIT     #B15+B7,TCCM    ;WAIT FOR ERROR OR READY
001354  001774                        BEQ     .-6             ;HAVE NEITHER
001356  100002                        BPL     .+6             ;HAVE READY
001360  004767  001670                JSR     PC,SERROR       ;STATUS ERROR
```

```
001364  112567  175750          MOVB    (R5)+,TCST      ;FETCH RELEVANT XDA BITS
001370  011367  175754          MOV     (R3),TCDT       ;FETCH RELEVANT CMP,OBV,BLK#
001374  062703  000002          ADD     #2,R3           ;ADVANCE TABLE POINTER
001400  020527  002570          CMP     R5,#BYST+10     ;SEE IF XDA BIT TABLE IS EXHAUSTED
001404  001002                  BNE     .+6             ;IF NOT, PROCEED
001406  012705  002560          MOV     #BYST,R5        ;IF SO, REINITIALIZE POINTER
001412  012704  002546          MOV     #WALBF,R4       ;SETUP POINTER FOR LAST FIVE WORDS
001416  012767  177776  003236  MOV     #-2,TOG1        ;COUNT OF TWO
001424  032767  100200  175710  WLP4:  BIT     #B15+B7,TCCM    ;WAIT FOR ERROR OR READY
001432  001774                  BEQ     .-6             ;HAVE NEITHER
001434  100002                  BPL     .+6             ;HAVE READY
001436  004767  001612          JSR     PC,SERROR       ;STATUS ERROR
001442  152767  000003  175670  BISB    #3,TCST         ;SET XDA 17 & 16
001450  012467  175674          MOV     (R4)+,TCDT      ;FETCH DATA WORD
001454  005267  003202          INC     TOG1            ;ADVANCE COUNT
001460  001361                  BNE     WLP4            ;KEEP WRITING UNTIL FINISHED
001462  032767  100200  175652  BIT     #B15+B7,TCCM    ;WAIT FOR ERROR OR READY
001470  001774                  BEQ     .-6             ;HAVE NEITHER
001472  100002                  BPL     .+6             ;HAVE READY
001474  004767  001554          JSR     PC,SERROR       ;STATUS ERROR
001500  142767  000003  175632  BICB    #3,TCST         ;CLEAR XDA 17 AND 16 FOR REVERSE BLK#
001506  012467  175636          MOV     (R4)+,TCDT      ;REVERSE BLOCK NUMBER
001512  012767  177776  003142  MOV     #-2,TOG1        ;COUNT OF TWO
001520  032767  100200  175614  WLP5:  BIT     #B15+B7,TCCM    ;WAIT FOR ERROR OR READY
001526  001774                  BEQ     .-6             ;HAVE NEITHER
001530  100002                  BPL     .+6             ;HAVE READY
001532  004767  001516          JSR     PC,SERROR       ;STATUS ERROR
001536  152767  000003  175574  BISB    #3,TCST         ;SET XDA 17 AND16
001544  012467  175600          MOV     (R4)+,TCDT      ;FETCH DATA WORD
001550  005267  003106          INC     TOG1            ;ADVANCE COUNT
001554  001361                  BNE     WLP5            ;KEEP WRITING UNTIL FINISHED
001556  000641                  BR      WALUP           ;FINISHED - DO NEXT BLOCK
```

```
PALX11  V003    19-APR-72       19:11   PAGE 9

001560  032767  100200  175554  CHECK:  BIT     #B15+B7,TCCM    ;WAIT FOR ERROR OR READY
001566  001774                          BEQ     .-6             ;HAVE NEITHER
001570  100002                          BPL     .+6             ;HAVE READY
001572  004767  001456                  JSR     PC,SERROR       ;STATUS ERROR
001576  112767  000003  175536          MOVB    #3,TCCM         ;REWIND INTO END ZONE

001604  004767  000514          CHKL1:  JSR     PC,FLAG1        ;SKIP NEXT INSTRUCTION IF EZ
001610  000765                          BR      CHECK+4         ;KEEP REWINDING
001612  005067  175524                  CLR     TCCM            ;STOP TAPE
001616  005067  003042                  CLR     TOG2            ;START WITH BLOCK#0
001622  112767  000003  175512  CHKL2:  MOVB    #3,TCCM         ;SEARCH FWD
001630  032767  100200  175504          BIT     #B15+B7,TCCM    ;WAIT FOR ERROR OR READY
001636  001774                          BEQ     .-6             ;HAVE NEITHER
001640  100002                          BPL     .+6             ;HAVE READY
001642  004767  001406                  JSR     PC,SERROR       ;STATUS ERROR
001646  026767  003012  175474          CMP     TOG2,TCDT       ;IS FWD BLOCK NUMBER CORRECT?
001654  001415                          BEQ     DATCHK                          ;YES = CHECK DATA
001656  016767  175466  002764          MOV     TCDT,BAD        ;NO = PUT INCORRECT BLK# IN BAD FOR DISPLAY
001664  005067  175452                  CLR     TCCM            ;STOP TAPE AND
001670  016767  002770  002750          MOV     TOG2,GOOD       ;GOOD DATA
001676  104002                          ERR+GB                  ;FWD BLK # WRITTEN INCORRECTLY
001700  003725                          EMES2
001702  012767  004003  175432          MOV     #4003,TCCM      ;TRY AGAIN
001710  012767  177377  002744  DATCHK: MOV     #-257,,TOG1     ;CHECK REV, CHKSUM AND DATA
001716  112767  000007  175410          MOVB    #7,TCCM         ;READ ALL
001724  032767  100200  175410  DCHKL1: BIT     #B15+B7,TCCM    ;WAIT FOR ERROR OR READY
001732  001774                          BEQ     .-6             ;HAVE NEITHER
001734  100002                          BPL     .+6             ;HAVE READY
001736  004767  001312                  JSR     PC,SERROR       ;STATUS ERROR
001742  005767  175402                  TST     TCDT            ;IT SHOULD=0
001746  001426                          BEQ     DATOK           ;IF SO PROCEED
001750  005067  175366                  CLR     TCCM            ;IF NOT STOP TAPE,
001754  005067  002666                  CLR     GOOD            ;GOOD=0
001760  016767  175364  002662          MOV     TCDT,BAD        ;GET BAD DATA
001766  016727  002672  000000          MOV     TOG2,#0         ;LOAD AND DEFINE BKN
        001772
                                        BKN=.-2
001774  012767  000401  002650          MOV     #257,,DATA      ;BASE
002002  066767  002634  002642          ADD     TOG1,DATA       ;GET WORD POSITION
002010  104032                          ERR+BK+GB+DA            ;DATA ERROR, "BAD" = WORD READ,
002012  003774                          EMES3                   ;"DATA" = WORD POSITION
```

```
002014  012767  004003  175320          MOV     #4003,TCCM      ;TO TRY AGAIN
002022  000670                          BR      CHKL1           ;RESTART CHECK
002024  005267  002632          DATOK:  INC     TOG1            ;CHECK ALL DATA IN BLOCK
002030  001335                          BNE     DCHKL1          ;KEEP CHECKING UNTIL DONE
002032  005267  002626                  INC     TOG2            ;SETUP FOR NEXT BLOCK
002036  022767  001102  002620          CMP     #1102,TOG2      ;HAVE ALL BLOCKS BEEN CHECKED?
002044  001266                          BNE     CHKL2           ;IF NOT KEEP CHECKING
002046  112767  000003  175266  RCHK:   MOVB    #3,TCCM         ;YES - FIND FWD EZ
002054  004767  000244                  JSR     PC,FLAG1        ;SKIP ON EZ
002060  000401                          BR      .+4             ;ERROR
002062  000402                          BR      .+6             ;OK
002064  104000                          ERR                     ;BLK# FOUND AFTER 1101
002066  004126                          EMES4
002070  012767  001101  002566          MOV     #1101,TOG2      ;FIRST REVERSE BLK=1101
002076  012767  004000  175236  RERCV:  MOV     #4000,TCCM      ;SET TAPE TO REVERSE
002104  012700  010736                  MOV     #DATBUF,R0      ;INITIALIZE DATA BUFFER POINTER
002110  005010                  ZLUP:   CLR     (R0)            ;ZERO DATA WORD
002112  062700  000002                  ADD     #2,R0           ;ADVANCE POINTER
002116  020027  011736                  CMP     R0,#DATBUF+512. ;IS BUFFER FILLED?
002122  001372                          BNE     ZLUP            ;IF NOT KEEP CLEARING
002124  112767  000003  175210  CHKL3:  MOVB    #3,TCCM         ;SEARCH REVERSE AND
002132  032767  100200  175202          BIT     #B15+B7,TCCM    ;WAIT FOR ERROR OR READY
002140  001774                          BEQ     .-6             ;HAVE NEITHER
002142  100002                          BPL     .+6             ;HAVE READY
002144  004767  001104                  JSR     PC,SERROR       ;STATUS ERROR

002150  026767  002510  175172          CMP     TOG2,TCDT       ;IS REVERSE BLOCK# CORRECT?
002156  001421                          BEQ     WRTDTA          ;YES - WRITE DATA
002160  016767  175164  002462          MOV     TCDT,BAD        ;NO - PUT INCORRECT BLK# IN BAD FOR DISPLAY
002166  005067  175150                  CLR     TCCM            ;STOP TAPE AND
002172  016767  002466  002440          MOV     TOG2,GOOD
002200  104002                          ERR+GB                  ;REVERSE BLK# WRITTEN INC.
002202  004200                          EMES5
002204  112767  000003  175130  RST1:   MOVB    #3,TCCM         ;TO TRY PASS AGAIN
002212  004767  000106                  JSR     PC,FLAG1        ;FIND FWD EZ AND
002216  000772                          BR      RST1            ;RESTART REVERSE
002220  000726                          BR      RERCV           ;CHECK PASS
002222  012767  177400  175114  WRTDTA: MOV     #-400,TCWC      ;WC FOR 1 BLOCK OF DATA
002230  012767  010736  175110          MOV     #DATBUF,TCBA    ;CA INITIALIZATION
002236  112767  000015  175076          MOVB    #15,TCCM        ;WRITE DATA TO MAKE PARITY OK
002244  032767  100200  175070          BIT     #B15+B7,TCCM    ;WAIT FOR ERROR OR READY
002252  001774                          BEQ     .-6             ;HAVE NEITHER
002254  100002                          BPL     .+6             ;HAVE READY
002256  000167  000306                  JMP     ERROR           ;HAVE ERROR
002262  005367  002376                  DEC     TOG2            ;DECREMENT BLOCK#
002266  100316                          BPL     CHKL3           ;IF BELOW 0 FINISH UP, OTHERWISE KEEP CHECKING
002270  112767  000003  175044  FINIS:  MOVB    #3,TCCM         ;FIND REVERSE EZ
002276  004767  000022                  JSR     PC,FLAG1        ;SKIP ON EZ
002302  000401                          BR      .+4             ;ERROR
002304  000402                          BR      .+6             ;OK
002306  104000                          ERR                     ;BLK# FOUND IN REVERSE AFTER 0
002310  004247                          EMES6
002312  005067  175024                  CLR     TCCM            ;STOP TAPE AND
002316  000000                          XX                      ;HALT.  PROGRAM IS FINISHED,  MOUNT NEW
002320  000167  176254                  JMP     START           ;TAPE, SET WRTM SWITCH AND CONTINUE
```

```
002324  032767  100200  175010  FLAG1:  BIT     #B15+B7,TCCM    ;TEST ERROR AND READY BITS
002332  001774                          BEQ     FLAG1           ;WAIT FOR EITHER FLAG
002334  100401                          BMI     .+4             ;IF ERROR - CHECK FURTHER
002336  000207                          RTS     PC              ;IF READY - EXIT
002340  005767  174774                  TST     TCST            ;IS ERROR END ZONE?
002344  100402                          BMI     .+6             ;YES - SKIP
002346  004767  000702                  JSR     PC,SERROR       ;STATUS ERROR
002352  062716  000002                  ADD     #2,(SP)         ;INCRENT EXIT POINTER TO SKIP ON RETURN
002356  000207                          RTS     PC              ;EXIT
002360  012700  001101          COBLKG: MOV     #1101,R0        ;COMPLEMENT OBVERSE BLOCKS 1101-0
002364  012701  006532                  MOV     #RBN,R1         ;BUFFER POINTER INITIALIZATION
002370  010002                  CBKL1:  MOV     R0,R2           ;FETCH WORD TO BE PROCESSED
002372  005102                          COM     R2              ;COMPLEMENT IT
002374  012767  177775  002260          MOV     #-3,TOG1        ;COUNT OF 3
002402  010203                          MOV     R2,R3           ;MOV WORD INTO WORK AREA
002404  006002                          ROR     R2              ;SWIFT NEXT 3 BITS INTO POSITION
002406  006002                          ROR     R2              ;FOR MANIPULATION
002410  006002                          ROR     R2              ;ON NEXT PASS
002412  042703  177776                  AND     1,R3
002416  010346                          BIC     #-1-1,R3        ;MASK 1 APPLIED TO R3
002420  010203                  CBKL2:  MOV     R3,-(SP)        ;STACK IT
002422  006002                          MOV     R2,R3           ;MOVE BYTE(3) INTO WORK AREA
002424  006002                          ROR     R2              ;SHIFT
002426  006002                          ROR     R2              ;NEXT 3 BITS
                                        ROR     R2              ;INTO POSITION
002430  042703  177770                  AND     7,R3
002434  010346                          BIC     #-7-1,R3        ;MASK 7 APPLIED TO R3
002436  005267  002220                  MOV     R3,-(SP)        ;STACK BYTE
002442  001366                          INC     TOG1            ;DONE ALL?
002444  005003                          BNE     CBKL2           ;NO
002446  066603  000006                  CLR     R3              ;YES - CLEAR WORK AREA
002452  004567  000056                  ADD     6(SP),R3        ;ADD AND
                                        JSR     R5,RTHL         ;SWIFT BYTES IN REVERSE ORDER
```

```
002456  066603  000004          ADD     4(SP),R3        ;NEXT BYTE
002462  004567  000046          JSR     R5,RTHL         ;SHIFT
002466  066603  000002          ADD     2(SP),R3        ;NEXT BYTE
002472  004567  000036          JSR     R5,RTHL         ;SHIFT
002476  061603                  ADD     (SP),R3         ;LAST BYTE FROM STACK
002500  004567  000030          JSR     R5,RTHL         ;SHIFT
002504  004567  000024          JSR     R5,RTHL         ;SHIFT
002510  062703  000077          ADD     #77,R3          ;=00 FWD
002514  010311                  MOV     R3,(R1)         ;STORE CMP,ORV,BLK# IN TABLE
002516  062706  000010          ADD     #10,SP          ;ADVANCE STACK
002522  062701  000002          ADD     #2,R1           ;ADVANCE TABLE POINTER
002526  005300                  DEC     R0              ;DECREMENT BLK#
002530  100317                  BPL     CBKL1           ;LOOP UNTIL IT PASSES 0
002532  000207                  RTS     PC              ;THEN EXIT
002534  000241          RTHL:   CLC                     ;THREE BIT LEFT SHIFT, CLEAR CARRY
002536  006103                  ROL     R3              ;SHIFT 1
002540  006103                  ROL     R3              ;      2
002542  006103                  ROL     R3              ;      3
002544  000205                  RTS     R5              ;THEN EXIT
002546  177777          WALBF:  ,WORD   -1              ;-1 REVERSE =0 FORWARD
002550  177777                  ,WORD   -1
002552  001101          FBLK:   ,WORD   1101            ;BLOCK NUMBER GOES HERE
002554  177777                  ,WORD   -1
002556  177777                  ,WORD   -1
002560     003          BYST:   ,BYTE   3               ;TABLE OF XDA 17 & 16 FOR CMP,BBV,BLK#
002561     003                  ,BYTE   3
002562     000                  ,BYTE   0
002563     000                  ,BYTE   0
002564     001                  ,BYTE   1
002565     001                  ,BYTE   1
002566     002                  ,BYTE   2
002567     002                  ,BYTE   2
```

```
002570  032767  040000  174772  ERROR:  BIT     #B14,SR         ;TEST TTO DELETE SWITCH
002576  001107                          BNE     ERXIT2          ;EXIT NOW IF TTO DELETE SWITCH IS UP
002600  004567  001510                  JSR     R5,SAV05                ;SAVE AC0-5
002604  012700  003412                  MOV     #ERMS1,R0       ;MESSAGE POINTER TO R0
002610  004767  001770                  JSR     PC,TYPOUT       ;PRINT MESSAGE
002614  016600  000014                  MOV     14(SP),R0               ;FETCH SAVED PC
002620  162700  000002                  SUB     #2,R0           ;FIND ADDRESS OF ERROR CALL
002624  010005                          MOV     R0,R5           ;PUT POINTER IN R5
002626  004767  001564                  JSR     PC,PNTOCS       ;PRINT PC
002632  012700  003541                  MOV     #STPMS,R0       ;MESSAGE POINTER
002636  004767  001742                  JSR     PC,TYPOUT       ;PRINT "PS="
002642  016600  000016                  MOV     16(SP),R0       ;FETCH SAVED STATUS
002646  004767  001544                  JSR     PC,PNTOCS       ;AND PRINT IT
002652  032715  000400                  BIT     #B8,(R5)        ;TRAP?
002656  001402                          BEQ     GOGO            ;NO
002660  004767  000404                  JSR     PC,SUBER        ;SUBROUTINE ERROR
002664  016600  000014          GOGO:   MOV     14(SP),R0
002670  062766  000002  000014          ADD     #2,14(SP)
002676  011000                          MOV     (R0),R0
002700  004767  001700                  JSR     PC,TYPOUT
002704  032715  000004                  BIT     #ST,(R5)        ;STATUS?
002710  001402                          BEQ     .+6             ;NO
002712  004767  000204                  JSR     PC,STYPE        ;YES
002716  032715  000002                  BIT     #GB,(R5)        ;GOOD/BAD?
002722  001402                          BEQ     .+6             ;SKIP IF=0
002724  004767  000130                  JSR     PC,GBTYPE               ;GOOD/BAD
002730  032715  000010                  BIT     #DA,(R5)        ;DATA?
002734  001402                          BEQ     .+6             ;SKIP IF=0
002736  004767  000202                  JSR     PC,DATYPE               ;DATA
002742  032715  000020                  BIT     #BK,(R5)        ;BLK NO.?
002746  001402                          BEQ     .+6             ;NO
002750  004767  000256                  JSR     PC,BKTYPE       ;YES
002754  032715  000040                  BIT     #VC,(R5)        ;VECTOR?
002760  001402                          BEQ     .+6             ;NO
002762  004767  000222                  JSR     PC,VCTYPE       ;YES
002766  004567  001336          ERXIT:  JSR     R5,REST05       ;RESTORE AC0-5
002772  012700  000015                  MOV     #CR,R0          ;PRINT
002776  004767  001512                  JSR     PC,TTO          ;CR & LF
003002  032767  100000  174560  ERXIT1: BIT     #B15,SR         ;SR15(I)=DELETE ERROR HALT
003010  001001                          BNE     .+4             ;SKIP IF Z=0
003012  000000          EHALT1: XX                              ;ERROR HALT, CONTINUE
003014  000002                          RTI                     ;EXIT ERROR ROUTINE
003016  062716  000002          ERXIT2: ADD     #2,(SP)
003022  000767                          BR      ERXIT1
```

```
003024  032767  020000  174536  LERCHK: BIT     #B13,SR         ;SR13(I)=LOOP ON ERROR
003032  001002                  LERCHK: BNE     .+6             ;EXIT IF 1
003034  062716  000004                  ADD     #4,(SP)         ;ADVANCE RETURN VECTOR OVER LOOP JUMP
003040  000207                  LERXT:  RTS     PC              ;EXIT

003042  032767  010000  174520  LUPCHK: BIT     #B12,SR         ;SR12(I)=LOOP ALWAYS
003050  001002                  LUPCHK: BNE     .+6             ;EXIT IF 1
003052  062716  000004                  ADD     #4,(SP)         ;ADVANCE RETURN VECTOR OVER LOOP JUMP
003056  000207                          RTS     PC              ;EXIT

                                GBTYPE: PNTM    GBMS
003060  012700  003472                  MOV     #GBMS,R0                ;PRINT MESSAGE
003064  004767  001514                  JSR     PC,TYPOUT       ;POINTED TO BY GBMS
003070  016700  001552                  MOV     GOOD,R0         ;FETCH C(GOOD)
003074  004767  001306                  JSR     PC,PNTORJ       ;PRINT OCTAL NUMBER
                                        PNTM    SP2
003100  012700  003536                  MOV     #SP2,R0         ;PRINT MESSAGE
003104  004767  001474                  JSR     PC,TYPOUT       ;POINTED TO BY SP2
003110  016700  001534                  MOV     BAD,R0          ;FETCH C(BAD)
003114  004767  001266                  JSR     PC,PNTORJ       ;PRINT OCTAL NUMBER
003120  000207                          RTS     PC              ;EXIT


003122  012700  003513          STYPE:  MOV     #STMS,R0                ;POINTER TO "STATUS"
003126  004767  001452                  JSR     PC,TYPOUT       ;PRINT TEXT
003132  016700  001520                  MOV     STATUS,R0       ;FETCH C(RPCS)
003136  004767  001236                  JSR     PC,PNTOAL               ;PRINT OCTAL NUMBER
003142  000207                          RTS     PC              ;EXIT
003144  012700  003590          DATYPE: MOV     #DAMS,R0                ;POINTER TO "DATA"
003150  004767  001430                  JSR     PC,TYPOUT       ;PRINT TEST
003154  016700  001472                  MOV     DATA,R0         ;FETCH C(DATA)
003160  004767  001232                  JSR     PC,PNTOZS       ;PRINT OCTAL NUMBER
003164  000207                          RTS     PC              ;EXIT


003166  011627  000000          TRCTRP: MOV     (SP),#0         ;LOAD AND DEFINE BAD PI ADDRESS SLOT
        003170                          BADPAD=.-2
003172  162767  000006  177770          SUB     #6,BADPAD       ;ADJUST ADDRESS
003200  022626                          POPPOP                  ;ADJUST STACK
003202  104040                          ERR+VC                  ;PRINT ERROR
003204  003574                          TRCMES                  ;UNEXPECTED INTERRUPT
003206  000002                          RTI                     ;EXIT

                                VCTYPE: PNTM    VECMES
003210  012700  003562                  MOV     #VECMES,R0              ;PRINT MESSAGE
003214  004767  001364                  JSR     PC,TYPOUT       ;POINTED TO BY VECMES
003220  016700  177744                  MOV     BADPAD,R0       ;GET BAD VECTOR
003224  004767  001166                  JSR     PC,PNTOZS       ;PRINT ADDRESS
003230  000207                          RTS     PC              ;EXIT

                                BKTYPE: PNTM    BKMES
003232  012700  003657                  MOV     #BKMES,R0               ;PRINT MESSAGE
003236  004767  001342                  JSR     PC,TYPOUT       ;POINTED TO BY BKMES
```

```
003242  016700  176524              MOV     BKN,R0          ;GET BLK#
003246  004767  001144              JSR     PC,PNTOZS       ;PRINT IT
003252  000207                      RTS     PC              ;EXIT

003254  016767  174062  001374  SERROR: MOV     TCCM,STATUS
003262  104404                      ERS+ST                  ;STATUS ERROR
003264  003677                      EMES1
003266  000207                      RTS     PC

003270  012700  003436          SUBER:  MOV     #SUBRMS,R0
003274  004767  001304              JSR     PC,TYPOUT
003300  016600  000022              MOV     22(SP),R0
003304  162700  000006              SUB     #6,R0
003310  004767  001102              JSR     PC,PNTOZS
003314  000207                      RTS     PC
```

```
003316  017627  000000  000000  RTR:    MOV     @(SP),#0
        003322                          ROTOG=,-2
003324  062716  000002                  ADD     #2,(SP)
003330  005767  177766                  TST     ROTOG
003334  001001                          BNE     .+4
003336  000207                          RTS     PC
003340  000241                  RRLUP:  CLC
003342  006000                          ROR     R0
003344  005367  177752                  DEC     ROTOG
003350  001373                          BNE     RRLUP
003352  000207                          RTS     PC
003354  017667  000000  177740  RTL:    MOV     @(SP),ROTOG
003362  062716  000002                  ADD     #2,(SP)
003366  005767  177730                  TST     ROTOG
003372  001001                          BNE     .+4
003374  000207                          RTS     PC
003376  000241                  RLLUP:  CLC
003400  006100                          ROL     R0
003402  005367  177714                  DEC     ROTOG
003406  001373                          BNE     RLLUP
003410  000207                          RTS     PC
```

```
003412  046                     ERMS11: .ASCII  /&&ERROR AT ADDRESS @/
003413  046
003414  105
003415  122
003416  122
003417  117
003420  122
003421  040
003422  101
003423  124
003424  040
003425  101
003426  104
003427  104
003430  122
003431  105
003432  123
003433  123
003434  040
003435  100

003436  046                     SUBRMS: .ASCII  /&CALLED FROM SUBROUTINE AT @/
003437  103
003440  101
003441  114
003442  114
003443  105
003444  104
003445  040
003446  106
003447  122
003450  117
003451  115
003452  040
003453  123
003454  125
003455  102
003456  122
003457  117
003460  125
003461  124
003462  111
003463  116
003464  105
003465  040
003466  101
003467  124
003470  040
003471  100

003472  046                     GBMS:   .ASCII  /&  GOOD     BAD&@/
003473  040
003474  040
003475  107
```

```
003496      117
003497      117
003500      104
003501      040
003502      040
003503      040
003504      040
003505      040
003506      102
003507      101
003510      104
003511      046
003512      100

003513      046                 STMS:   .ASCII  /&STATUS= @/
003514      123
003515      124
003516      101
003517      124
003520      125
003521      123
003522      075
003523      040
003524      100

003525      046                 PASMS:  .ASCII  /&&PASS# @/
003526      046
003527      120
003530      101
003531      123
003532      123
003533      043
003534      040
003535      100

003536      040                 SP2:    .ASCII  /  @/
003537      040
003540      100

003541      040                 STPMS:  .ASCII  /  PS= @/
003542      040
003543      120
003544      123
003545      075
003546      040
003547      100


003550      046                 DAMS:   .ASCII  /&DATA=   @/
003551      104
003552      101
003553      124
003554      101
003555      075
003556      040
```

```
003557      040
003560      040
003561      100

003562      046                 VECMES: .ASCII  /&VECTOR= @/
003563      126
003564      105
003565      103
003566      124
003567      117
003570      122
003571      075
003572      040
003573      100

003574      046                 TRCMES: .ASCII  /&UNEXPECTED INTERRUPT THROUGH THE FOLLOWING VECTOR@/
003575      125
003576      116
003577      105
003600      130
003601      120
003602      105
003603      103
003604      124
003605      105
003606      104
003607      040
003610      111
003611      116
003612      124
003613      105
003614      122
003615      122
003616      125
003617      120
003620      124
003621      040
003622      124
003623      110
003624      122
003625      117
003626      125
003627      107
003630      110
003631      040
003632      124
003633      110
003634      105
003635      040
003636      106
003637      117
003640      114
003641      114
003642      117
003643      127
```

```
003644      111
003645      116
003646      107
003647      040
003650      126
003651      105
003652      103
003653      124
003654      117
003655      122
003656      100

003657      046                 BKMES1:  .ASCII  /&BLOCK NUMBER= @/
003660      102
003661      114
003662      117
003663      103
003664      113
003665      040
003666      116
003667      125
003670      115
003671      102
003672      105
003673      122
003674      075
003675      040
003676      100

003677      046                 EMES11:  .ASCII  /&DECTAPE STATUS ERROR@/
003700      104
003701      105
003702      103
003703      124
003704      101
003705      120
003706      105
003707      040
003710      123
003711      124
003712      101
003713      124
003714      125
003715      123
003716      040
003717      105
003720      122
003721      122
003722      117
003723      122
003724      100

003725      046                 EMES21:  .ASCII  /&FORWARD BLOCK NUMBER WRITE-READ ERROR@/
003726      106
003727      117
```

```
003730      122
003731      127
003732      101
003733      122
003734      104
003735      040
003736      102
003737      114
003740      117
003741      103
003742      113
003743      040
003744      116
003745      125
003746      115
003747      102
003750      105
003751      122
003752      040
003753      127
003754      122
003755      111
003756      124
003757      105
003760      055
003761      122
003762      105
003763      101
003764      104
003765      040
003766      105
003767      122
003770      122
003771      117
003772      122
003773      100

003774      046                 EMES3I  .ASCII  /&DATA ERROR', "BAD" = WORD READ,"DATA" = WORD POSITION IN THE BLOCK/
003775      104
003776      101
003777      124
004000      101
004001      040
004002      105
004003      122
004004      122
004005      117
004006      122
004007      056
004010      040
004011      042
004012      102
004013      101
004014      104
004015      042
```

```
004016      040
004017      075
004020      040
004021      127
004022      117
004023      122
004024      104
004025      040
004026      122
004027      105
004030      101
004031      104
004032      056
004033      042
004034      104
004035      101
004036      124
004037      101
004040      042
004041      040
004042      075
004043      040
004044      127
004045      117
004046      122
004047      104
004050      040
004051      120
004052      117
004053      123
004054      111
004055      124
004056      111
004057      117
004060      116
004061      040
004062      111
004063      116
004064      040
004065      124
004066      110
004067      105
004070      040
004071      102
004072      114
004073      117
004074      103
004075      113

004076      046                     .ASCII  /&(0 = REVERSE CHECKSUM)@/
004077      050
004100      060
004101      040
004102      075
004103      040
```

```
004104      122
004105      105
004106      126
004107      105
004110      122
004111      123
004112      105
004113      040
004114      103
004115      110
004116      105
004117      103
004120      113
004121      123
004122      125
004123      115
004124      051
004125      100

004126      046                 EMES41  .ASCII  /&FORWARD BLOCK NUMBER FOUND AFTER 1101(8)0/
004127      106
004130      117
004131      122
004132      127
004133      101
004134      122
004135      104
004136      040
004137      102
004140      114
004141      117
004142      103
004143      113
004144      040
004145      116
004146      125
004147      115
004150      102
004151      105
004152      122
004153      040
004154      106
004155      117
004156      125
004157      116
004160      104
004161      040
004162      101
004163      106
004164      124
004165      105
004166      122
004167      040
004170      061
```

```
004171      061
004172      060
004173      061
004174      050
004175      070
004176      051
004177      100

004200      046                 EMES5:  .ASCII  /&REVERSE BLOCK NUMBER WRITE-READ ERROR@/
004201      122
004202      105
004203      126
004204      105
004205      122
004206      123
004207      105
004210      040
004211      102
004212      114
004213      117
004214      103
004215      113
004216      040
004217      116
004220      125
004221      115
004222      102
004223      105
004224      122
004225      040
004226      127
004227      122
004230      111
004231      124
004232      105
004233      055
004234      122
004235      105
004236      101
004237      104
004240      040
004241      105
004242      122
004243      122
004244      117
004245      122
004246      100

004247      046                 EMES6:  .ASCII  /&REVERSE BLOCK NUMBER FOUND BEFORE 0@/
004250      122
004251      105
004252      126
004253      105
004254      122
004255      123
```

```
004256      105
004257      040
004260      122
004261      114
004262      117
004263      103
004264      113
004265      040
004266      116
004267      125
004270      115
004271      102
004272      105
004273      122
004274      040
004275      106
004276      117
004277      125
004300      116
004301      104
004302      040
004303      102
004304      105
004305      106
004306      117
004307      122
004310      105
004311      040
004312      060
004313      100
```

```
                004314                          .EVEN

004314  010446                  SAV05:  MOV     R4,-(SP)        ;SAVE R0-R5 ON THE STACK
004316  010346                          MOV     R3,-(SP)        ;R5 WAS STACKED BY THE JSR
004320  010246                          MOV     R2,-(SP)        ;R4-R3 ARE STACKED ABOVE IT
004322  010146                          MOV     R1,-(SP)        ;WITH R0 ON TOP
004324  010046                          MOV     R0,-(SP)        ;R5 HOLDS THE RETURN PC, BUT AN
004326  000115                          JMP     (R5)            ;RTS WOULD POP THE STACK-SO JUMP OUT

004330  005726                  REST05: TST     (SP)+           ;MOVE SP OVER WORD SAVED BY JSR
004332  012600                          MOV     (SP)+,R0        ;R0-4
004334  012601                          MOV     (SP)+,R1        ;ARE POPPED
004336  012602                          MOV     (SP)+,R2        ;IN LIFO
004340  012603                          MOV     (SP)+,R3        ;SEQUENCE
004342  012604                          MOV     (SP)+,R4        ;R5 IS POPPED BY THE RTS AND
004344  000205                          RTS     R5              ;THE PC IS TAKEN FROM R5

004346  005067  000306          PNTDAL: CLR     SCON
004352  000407                          BR      DECGO
004354  012767  000020  000276  PNTDRJ: MOV     #20,SCON
004362  000403                          BR      DECGO
004364  012767  000060  000266  PNTDZS: MOV     #60,SCON
004372  012702  000012          DECGO:  MOV     #10.,R2
004376  000414                          BR      OCTGO+4
004400  005067  000254          PNTOAL: CLR     SCON
004404  000407                          BR      OCTGO
004406  012767  000020  000244  PNTORJ: MOV     #20,SCON
004414  000403                          BR      OCTGO
004416  012767  000060  000234  PNTOZS: MOV     #60,SCON
004424  012702  000010          OCTGO:  MOV     #8.,R2
004430  005027  000000                  CLR     #0
                004432                  ZSW=.-2
004434  004767  000114          DECREM: JSR     PC,IDIVR        ;DIVIDE (R0) BY (R2) WITH REMAINDER IN R1
004440  010146                          MOV     R1,-(SP)        ;STACK REMAINDER
004442  005267  177764                  INC     ZSW
004446  026727  177760  000006        CMP     ZSW,#6
004454  001402                          BEQ     POPTT           ;YES - POP AND PRINT
004456  004767  177752                  JSR     PC,DECREM       ;NO - DIVIDE NUMBER BY RADIX
004462  005367  177744          POPTT:  DEC     ZSW
004466  001002                          BNE     .+6
004470  005067  000164                  CLR     SCON
004474  012600                          MOV     (SP)+,R0        ;POP NUMBER FROM STACK
004476  001402                          BEQ     .+6
004500  005067  000154                  CLR     SCON
004504  166700  000150                  SUB     SCON,R0
004510  062700  000060                  ADD     #60,R0          ;MAKE ASCII
```

```
004514  032767  040000  173046  TTO:    BIT     #B14,SR         ;IF SR14=1,DELETE TYPEOUT
004522  001010                  BNE     TTOLF-2         ;EXIT
004524  010067  173036          MOV     R0,TPB          ;PRINT CONTENTS OF R0
004530  105767  173030          TTOLP:  TSTB    TPS             ;DONE YET?
004534  100375                  BPL     TTOLP           ;NO - KEEP LOOPING
004536  022700  000015          CMP     #CR,R0          ;WAS CHARACTER A CR!
004542  001401                  BEQ     TTOLF           ;YES - PRINT LINE FEED
004544  000207                  RTS     PC              ;RETURN TO POPTT OR MAIN PROGRAM
004546  012700  000012          TTOLF:  MOV     #LF,R0          ;PRINT LF
004552  000760                  BR      TTO             ;EXECUTE PRINT

004554  012703  000020          IDIVR:  MOV     #16.,R3
004560  005001                  CLR     R1
004562  006300                  SD:     ASL     R0
004564  006101                  ROL     R1
004566  020201                  CMP     R2,R1
004570  101002                  BHI     .+6
004572  160201                  SUB     R2,R1
004574  005200                  INC     R0
004576  005303                  DEC     R3
004600  001370                  BNE     SD
004602  000207                  RTS     PC

004604  010046          TYPOUT: MOV     R0,-(SP)        ;STACK ADDRESS POINTER FOR MESSAGE
004606  117600  000000  TPOFCH: MOVB    @(SP),R0        ;FETCH ASCII BYTE
004612  022700  000100          CMP     #100,R0         ;IS IT @(TERMINATOR)?
004616  001411                  BEQ     TPOUTX          ;YES - EXIT
004620  022700  000046          CMP     #46,R0          ;IS IT CRLF FLAG?
004624  001002                  BNE     .+6             ;NO
004626  012700  000015          MOV     #CR,R0          ;YES- CHANGE DATA TO CRLF
004632  004767  177656          JSR     PC,TTO          ; PRINT
004636  005216                  INC     (SP)            ;MOVE POINTER TO NEXT BYTE
004640  000762                  BR      TPOFCH          ;FETCH NEXT CHARACTER
004642  005726          TPOUTX: POP                     ;POP STACK TO REACH RETURN VECTOR
004644  000207                  RTS     PC              ;EXIT

                                ;VARIABLE SLOTS

004646  000000          GOOD:   XX
004650  000000          BAD:    XX
004652  000000          DATA:   XX
004654  000000          DIVID:  XX
004656  000000          STATUS: XX
004660  000000          SCON:   XX
004662  000000          TOG1:   XX
004664  000000          TOG2:   XX
004666  000000          TOG3:   XX
004670  000000          TOG4:   XX
004672  000000          WORK:   XX
004674  000000          BTOG:   XX
004676  000000          TEMP1:  XX
```

```
004700  000000          PASCNT: XX
004702  000030          MASK:   XX
004704  000000          LERR:   XX
004706  000000          DEBUG:  XX
        005176                  .=DEBUG+200

        005506                  .=.+400
```

```
005506  010101                  BLKMRK: .WORD   010101          ;INTERBLOCK SYNC=25
005510  010110                          .WORD   010110          ;FORWARD BLOCK MARK=26
005512  011010                          .WORD   011010          ;REVERSE GUARD=32
005514  001000                          .WORD   001000          ;LOCK=10
005516  001000                          .WORD   001000          ;REVERSE PARITY=10
005520  001000                          .WORD   001000          ;REVERSE FINAL=10
005522  001000                          .WORD   001000          ;REVERSE PRE-FINAL=10
        000374                          .REPT   374
                                        .WORD   111000          ;MARK DATA=70
                                        .ENDR

005524  111000                          .WORD   111000          ;MARK DATA=70
005526  111000                          .WORD   111000          ;MARK DATA=70
005530  111000                          .WORD   111000          ;MARK DATA=70
005532  111000                          .WORD   111000          ;MARK DATA=70
005534  111000                          .WORD   111000          ;MARK DATA=70
005536  111000                          .WORD   111000          ;MARK DATA=70
005540  111000                          .WORD   111000          ;MARK DATA=70
005542  111000                          .WORD   111000          ;MARK DATA=70
005544  111000                          .WORD   111000          ;MARK DATA=70
005546  111000                          .WORD   111000          ;MARK DATA=70
005550  111000                          .WORD   111000          ;MARK DATA=70
005552  111000                          .WORD   111000          ;MARK DATA=70
005554  111000                          .WORD   111000          ;MARK DATA=70
005556  111000                          .WORD   111000          ;MARK DATA=70
005560  111000                          .WORD   111000          ;MARK DATA=70
005562  111000                          .WORD   111000          ;MARK DATA=70
005564  111000                          .WORD   111000          ;MARK DATA=70
005566  111000                          .WORD   111000          ;MARK DATA=70
005570  111000                          .WORD   111000          ;MARK DATA=70
005572  111000                          .WORD   111000          ;MARK DATA=70
005574  111000                          .WORD   111000          ;MARK DATA=70
005576  111000                          .WORD   111000          ;MARK DATA=70
005600  111000                          .WORD   111000          ;MARK DATA=70
005602  111000                          .WORD   111000          ;MARK DATA=70
005604  111000                          .WORD   111000          ;MARK DATA=70
005606  111000                          .WORD   111000          ;MARK DATA=70
005610  111000                          .WORD   111000          ;MARK DATA=70
005612  111000                          .WORD   111000          ;MARK DATA=70
005614  111000                          .WORD   111000          ;MARK DATA=70
005616  111000                          .WORD   111000          ;MARK DATA=70
005620  111000                          .WORD   111000          ;MARK DATA=70
005622  111000                          .WORD   111000          ;MARK DATA=70
005624  111000                          .WORD   111000          ;MARK DATA=70
005626  111000                          .WORD   111000          ;MARK DATA=70
005630  111000                          .WORD   111000          ;MARK DATA=70
005632  111000                          .WORD   111000          ;MARK DATA=70
005634  111000                          .WORD   111000          ;MARK DATA=70
005636  111000                          .WORD   111000          ;MARK DATA=70
005640  111000                          .WORD   111000          ;MARK DATA=70
005642  111000                          .WORD   111000          ;MARK DATA=70
005644  111000                          .WORD   111000          ;MARK DATA=70
005646  111000                          .WORD   111000          ;MARK DATA=70
005650  111000                          .WORD   111000          ;MARK DATA=70
005652  111000                          .WORD   111000          ;MARK DATA=70
```

```
005654  111000                  .WORD   111000          ;MARK DATA=70
005656  111000                  .WORD   111000          ;MARK DATA=70
005660  111000                  .WORD   111000          ;MARK DATA=70
005662  111000                  .WORD   111000          ;MARK DATA=70
005664  111000                  .WORD   111000          ;MARK DATA=70
005666  111000                  .WORD   111000          ;MARK DATA=70
005670  111000                  .WORD   111000          ;MARK DATA=70
005672  111000                  .WORD   111000          ;MARK DATA=70
005674  111000                  .WORD   111000          ;MARK DATA=70
005676  111000                  .WORD   111000          ;MARK DATA=70
005700  111000                  .WORD   111000          ;MARK DATA=70
005702  111000                  .WORD   111000          ;MARK DATA=70
005704  111000                  .WORD   111000          ;MARK DATA=70
005706  111000                  .WORD   111000          ;MARK DATA=70
005710  111000                  .WORD   111000          ;MARK DATA=70
005712  111000                  .WORD   111000          ;MARK DATA=70
005714  111000                  .WORD   111000          ;MARK DATA=70
005716  111000                  .WORD   111000          ;MARK DATA=70
005720  111000                  .WORD   111000          ;MARK DATA=70
005722  111000                  .WORD   111000          ;MARK DATA=70
005724  111000                  .WORD   111000          ;MARK DATA=70
005726  111000                  .WORD   111000          ;MARK DATA=70
005730  111000                  .WORD   111000          ;MARK DATA=70
005732  111000                  .WORD   111000          ;MARK DATA=70
005734  111000                  .WORD   111000          ;MARK DATA=70
005736  111000                  .WORD   111000          ;MARK DATA=70
005740  111000                  .WORD   111000          ;MARK DATA=70
005742  111000                  .WORD   111000          ;MARK DATA=70
005744  111000                  .WORD   111000          ;MARK DATA=70
005746  111000                  .WORD   111000          ;MARK DATA=70
005750  111000                  .WORD   111000          ;MARK DATA=70
005752  111000                  .WORD   111000          ;MARK DATA=70
005754  111000                  .WORD   111000          ;MARK DATA=70
005756  111000                  .WORD   111000          ;MARK DATA=70
005760  111000                  .WORD   111000          ;MARK DATA=70
005762  111000                  .WORD   111000          ;MARK DATA=70
005764  111000                  .WORD   111000          ;MARK DATA=70
005766  111000                  .WORD   111000          ;MARK DATA=70
005770  111000                  .WORD   111000          ;MARK DATA=70
005772  111000                  .WORD   111000          ;MARK DATA=70
005774  111000                  .WORD   111000          ;MARK DATA=70
005776  111000                  .WORD   111000          ;MARK DATA=70
006000  111000                  .WORD   111000          ;MARK DATA=70
006002  111000                  .WORD   111000          ;MARK DATA=70
006004  111000                  .WORD   111000          ;MARK DATA=70
006006  111000                  .WORD   111000          ;MARK DATA=70
006010  111000                  .WORD   111000          ;MARK DATA=70
006012  111000                  .WORD   111000          ;MARK DATA=70
006014  111000                  .WORD   111000          ;MARK DATA=70
006016  111000                  .WORD   111000          ;MARK DATA=70
006020  111000                  .WORD   111000          ;MARK DATA=70
006022  111000                  .WORD   111000          ;MARK DATA=70
006024  111000                  .WORD   111000          ;MARK DATA=70
006026  111000                  .WORD   111000          ;MARK DATA=70
006030  111000                  .WORD   111000          ;MARK DATA=70
```

```
006032  111000                  .WORD   111000          ;MARK DATA=70
006034  111000                  .WORD   111000          ;MARK DATA=70
006036  111000                  .WORD   111000          ;MARK DATA=70
006040  111000                  .WORD   111000          ;MARK DATA=70
006042  111000                  .WORD   111000          ;MARK DATA=70
006044  111000                  .WORD   111000          ;MARK DATA=70
006046  111000                  .WORD   111000          ;MARK DATA=70
006050  111000                  .WORD   111000          ;MARK DATA=70
006052  111000                  .WORD   111000          ;MARK DATA=70
006054  111000                  .WORD   111000          ;MARK DATA=70
006056  111000                  .WORD   111000          ;MARK DATA=70
006060  111000                  .WORD   111000          ;MARK DATA=70
006062  111000                  .WORD   111000          ;MARK DATA=70
006064  111000                  .WORD   111000          ;MARK DATA=70
006066  111000                  .WORD   111000          ;MARK DATA=70
006070  111000                  .WORD   111000          ;MARK DATA=70
006072  111000                  .WORD   111000          ;MARK DATA=70
006074  111000                  .WORD   111000          ;MARK DATA=70
006076  111000                  .WORD   111000          ;MARK DATA=70
006100  111000                  .WORD   111000          ;MARK DATA=70
006102  111000                  .WORD   111000          ;MARK DATA=70
006104  111000                  .WORD   111000          ;MARK DATA=70
006106  111000                  .WORD   111000          ;MARK DATA=70
006110  111000                  .WORD   111000          ;MARK DATA=70
006112  111000                  .WORD   111000          ;MARK DATA=70
006114  111000                  .WORD   111000          ;MARK DATA=70
006116  111000                  .WORD   111000          ;MARK DATA=70
006120  111000                  .WORD   111000          ;MARK DATA=70
006122  111000                  .WORD   111000          ;MARK DATA=70
006124  111000                  .WORD   111000          ;MARK DATA=70
006126  111000                  .WORD   111000          ;MARK DATA=70
006130  111000                  .WORD   111000          ;MARK DATA=70
006132  111000                  .WORD   111000          ;MARK DATA=70
006134  111000                  .WORD   111000          ;MARK DATA=70
006136  111000                  .WORD   111000          ;MARK DATA=70
006140  111000                  .WORD   111000          ;MARK DATA=70
006142  111000                  .WORD   111000          ;MARK DATA=70
006144  111000                  .WORD   111000          ;MARK DATA=70
006146  111000                  .WORD   111000          ;MARK DATA=70
006150  111000                  .WORD   111000          ;MARK DATA=70
006152  111000                  .WORD   111000          ;MARK DATA=70
006154  111000                  .WORD   111000          ;MARK DATA=70
006156  111000                  .WORD   111000          ;MARK DATA=70
006160  111000                  .WORD   111000          ;MARK DATA=70
006162  111000                  .WORD   111000          ;MARK DATA=70
006164  111000                  .WORD   111000          ;MARK DATA=70
006166  111000                  .WORD   111000          ;MARK DATA=70
006170  111000                  .WORD   111000          ;MARK DATA=70
006172  111000                  .WORD   111000          ;MARK DATA=70
006174  111000                  .WORD   111000          ;MARK DATA=70
006176  111000                  .WORD   111000          ;MARK DATA=70
006200  111000                  .WORD   111000          ;MARK DATA=70
006202  111000                  .WORD   111000          ;MARK DATA=70
006204  111000                  .WORD   111000          ;MARK DATA=70
006206  111000                  .WORD   111000          ;MARK DATA=70
```

```
006210  111000                  .WORD   111000          ;MARK DATA=70
006212  111000                  .WORD   111000          ;MARK DATA=70
006214  111000                  .WORD   111000          ;MARK DATA=70
006216  111000                  .WORD   111000          ;MARK DATA=70
006220  111000                  .WORD   111000          ;MARK DATA=70
006222  111000                  .WORD   111000          ;MARK DATA=70
006224  111000                  .WORD   111000          ;MARK DATA=70
006226  111000                  .WORD   111000          ;MARK DATA=70
006230  111000                  .WORD   111000          ;MARK DATA=70
006232  111000                  .WORD   111000          ;MARK DATA=70
006234  111000                  .WORD   111000          ;MARK DATA=70
006236  111000                  .WORD   111000          ;MARK DATA=70
006240  111000                  .WORD   111000          ;MARK DATA=70
006242  111000                  .WORD   111000          ;MARK DATA=70
006244  111000                  .WORD   111000          ;MARK DATA=70
006246  111000                  .WORD   111000          ;MARK DATA=70
006250  111000                  .WORD   111000          ;MARK DATA=70
006252  111000                  .WORD   111000          ;MARK DATA=70
006254  111000                  .WORD   111000          ;MARK DATA=70
006256  111000                  .WORD   111000          ;MARK DATA=70
006260  111000                  .WORD   111000          ;MARK DATA=70
006262  111000                  .WORD   111000          ;MARK DATA=70
006264  111000                  .WORD   111000          ;MARK DATA=70
006266  111000                  .WORD   111000          ;MARK DATA=70
006270  111000                  .WORD   111000          ;MARK DATA=70
006272  111000                  .WORD   111000          ;MARK DATA=70
006274  111000                  .WORD   111000          ;MARK DATA=70
006276  111000                  .WORD   111000          ;MARK DATA=70
006300  111000                  .WORD   111000          ;MARK DATA=70
006302  111000                  .WORD   111000          ;MARK DATA=70
006304  111000                  .WORD   111000          ;MARK DATA=70
006306  111000                  .WORD   111000          ;MARK DATA=70
006310  111000                  .WORD   111000          ;MARK DATA=70
006312  111000                  .WORD   111000          ;MARK DATA=70
006314  111000                  .WORD   111000          ;MARK DATA=70
006316  111000                  .WORD   111000          ;MARK DATA=70
006320  111000                  .WORD   111000          ;MARK DATA=70
006322  111000                  .WORD   111000          ;MARK DATA=70
006324  111000                  .WORD   111000          ;MARK DATA=70
006326  111000                  .WORD   111000          ;MARK DATA=70
006330  111000                  .WORD   111000          ;MARK DATA=70
006332  111000                  .WORD   111000          ;MARK DATA=70
006334  111000                  .WORD   111000          ;MARK DATA=70
006336  111000                  .WORD   111000          ;MARK DATA=70
006340  111000                  .WORD   111000          ;MARK DATA=70
006342  111000                  .WORD   111000          ;MARK DATA=70
006344  111000                  .WORD   111000          ;MARK DATA=70
006346  111000                  .WORD   111000          ;MARK DATA=70
006350  111000                  .WORD   111000          ;MARK DATA=70
006352  111000                  .WORD   111000          ;MARK DATA=70
006354  111000                  .WORD   111000          ;MARK DATA=70
006356  111000                  .WORD   111000          ;MARK DATA=70
006360  111000                  .WORD   111000          ;MARK DATA=70
006362  111000                  .WORD   111000          ;MARK DATA=70
006364  111000                  .WORD   111000          ;MARK DATA=70
```

```
006366  111000          .WORD   111000          ;MARK DATA=70
006370  111000          .WORD   111000          ;MARK DATA=70
006372  111000          .WORD   111000          ;MARK DATA=70
006374  111000          .WORD   111000          ;MARK DATA=70
006376  111000          .WORD   111000          ;MARK DATA=70
006400  111000          .WORD   111000          ;MARK DATA=70
006402  111000          .WORD   111000          ;MARK DATA=70
006404  111000          .WORD   111000          ;MARK DATA=70
006406  111000          .WORD   111000          ;MARK DATA=70
006410  111000          .WORD   111000          ;MARK DATA=70
006412  111000          .WORD   111000          ;MARK DATA=70
006414  111000          .WORD   111000          ;MARK DATA=70
006416  111000          .WORD   111000          ;MARK DATA=70
006420  111000          .WORD   111000          ;MARK DATA=70
006422  111000          .WORD   111000          ;MARK DATA=70
006424  111000          .WORD   111000          ;MARK DATA=70
006426  111000          .WORD   111000          ;MARK DATA=70
006430  111000          .WORD   111000          ;MARK DATA=70
006432  111000          .WORD   111000          ;MARK DATA=70
006434  111000          .WORD   111000          ;MARK DATA=70
006436  111000          .WORD   111000          ;MARK DATA=70
006440  111000          .WORD   111000          ;MARK DATA=70
006442  111000          .WORD   111000          ;MARK DATA=70
006444  111000          .WORD   111000          ;MARK DATA=70
006446  111000          .WORD   111000          ;MARK DATA=70
006450  111000          .WORD   111000          ;MARK DATA=70
006452  111000          .WORD   111000          ;MARK DATA=70
006454  111000          .WORD   111000          ;MARK DATA=70
006456  111000          .WORD   111000          ;MARK DATA=70
006460  111000          .WORD   111000          ;MARK DATA=70
006462  111000          .WORD   111000          ;MARK DATA=70
006464  111000          .WORD   111000          ;MARK DATA=70
006466  111000          .WORD   111000          ;MARK DATA=70
006470  111000          .WORD   111000          ;MARK DATA=70
006472  111000          .WORD   111000          ;MARK DATA=70
006474  111000          .WORD   111000          ;MARK DATA=70
006476  111000          .WORD   111000          ;MARK DATA=70
006500  111000          .WORD   111000          ;MARK DATA=70
006502  111000          .WORD   111000          ;MARK DATA=70
006504  111000          .WORD   111000          ;MARK DATA=70
006506  111000          .WORD   111000          ;MARK DATA=70
006510  111000          .WORD   111000          ;MARK DATA=70
006512  111000          .WORD   111000          ;MARK DATA=70
006514  111011          .WORD   111011          ;PRE-FINAL=73
006516  111011          .WORD   111011          ;FINAL=73
006520  111011          .WORD   111011          ;PARITY=73
006522  111011          .WORD   111011          ;REVERSE LOCK=73
006524  101001          .WORD   101001          ;GUARD=51
006526  100101          .WORD   100101          ;REVERSE BLOCK MARK=45
006530  010101          .WORD   010101          ;INTERBLOCK SYNC=25

        010736  RBN:    .=.+2204

        011736  DATBUF:  .=.+1000
        000001          .END                    ;THAT'S ALL FOLKS!
```

```
B0      000001          EMES6   004247          POPTT   004     TPS     177564
B1      000002          ERMS1   003412          PS      177776  TPV     000064
B10     002000          ERR     104000          R0      000000R TRCMES  003574
B11     004000          ERROR   002570          R1      000001R TRCTRP  003166
B12     010000          ERS     104400          R2      000002R TTO     004514
B13     020000          ERXIT   002766          R3      000003R TTOLF   004546
B14     040000          ERXIT1  003002          R4      000004R TTOLP   004530
B15     100000          ERXIT2  003016          R5      000005R TYPOUT  004604
B2      000004          FBLK    002552          R6      000006R VC      000040
B3      000010          FINIS   002270          R7      000007R VCTYPE  003210
B4      000020          FLAG1   002324          RBN     006532  VECMES  003562
B5      000040          FRMBLK  001056          RCHK    002046  WALBF   002546
B6      000100          GB      000002          RERCV   002070  WALUP   001262
B7      000200          GBMS    003472          REST05  004330  WLP1    001140
B8      000400          GBTYPE  003060          RLLUP   003376  WLP2    001216
B9      001000          GOGO    002664          ROTOG   003322  WLP3    001310
BAD     004650          GOOD    004646          RRLUP   003340  WLP4    001424
BADPAD  003170          HLT1    001050          RST1    002204  WLP5    001520
BK      000020          IDIVR   004554          RTHL    002534  WORK    004672
BKMES   003657          LERCHK  003024          RTL     003354  WRTBNO  001122
BKN     001772          LERR    004704          RTR     003316  WRTDTA  002222
BKTYPE  003232          LERXT   003040          SAV05   004314  XX      000000
BLKMRK  005506          LF      000012          SCON    004660  ZLUP    002110
BTOG    004674          LP1     000632          SD      004562  ZSW     004432
BYST    002560          LP2     000670          SERROR  003254
CBKL1   002370          LP3     000722          SP      000006R
CBKL2   002420          LP4     000742          SP2     003536
CC      177776          LP5     001000          SR      177570
CHECK   001560          LP6     001066          ST      000004
CHKL1   001604          LUPCHK  003042          START   000600
CHKL2   001622          MASK    004702          STATUS  004656
CHKL3   002124          NOP     000240          STMS    003513
COBLKG  002360          OCTGO   004424          STPMS   003541
CR      000015          P0      000000          STYPE   003122
DA      000010          P1      000040          SUBER   003270
DAMS    003550          P2      000100          SUBRMS  003436
DATA    004652          P3      000140          TCBA    177346
DATBUF  010736          P4      000200          TCCM    177342
DATCHK  001710          P5      000240          TCDT    177350
DATOK   002024          P6      000300          TCST    177340
DATYPE  003144          P7      000340          TCWC    177344
DCHKL1  001724          PASCNT  004700          TEMP1   004676
DEBUG   004706          PASMS   003525          TKB     177562
DECGO   004372          PC      000007R         TKS     177560
DECREM  004434          PNTDAL  004346          TKV     000060
DIVID   004654          PNTDRJ  004354          TOG1    004662
EHALT1  003012          PNTDZS  004364          TOG2    004664
EMES1   003677          PNTOAL  004400          TOG3    004666
EMES2   003725          PNTORJ  004406          TOG4    004670
EMES3   003774          PNTOZS  004416          TPB     177566
EMES4   004126          POP     005726          TPOFCH  004606
EMES5   004200          POPPOP  222626          TPOUTX  004642
```

ERRORS DETECTED: 0

RUN-TIME:  11 SECONDS

5K CORE USED