DataGeneral

DIAGNOSTIC LISTING

LISTING

096-000445-00

PROGRAM

6038/6039 FLOPPY DISK
DIAGNOSTIC

TAPE

095-000445-00

```
0001 FPYDI MACRO REV 04.00              15:23:31 12/03/76
01
02
03
04
05
06
07              /*****************************************************************
08              /
09              / NAME: FPYDIA.SR                              PART NUMBER: 094-000843
10              /
11              /
12              / DESCRIPTION: 6038/6039 FLOPPY DIAGNOSTIC
13              /
14              /
15              / REVISION HISTORY:
16              /
17              /       REV.            DATE
18              /
19              /       00              12/03/76
20              /
21              /
22              / COPYRIGHT (C) DATA GENERAL CORPORATION, 1976
23              / ALL RIGHTS RESERVED.
24              /*****************************************************************
```

```
0002 FPYDI
01
                .TITL FPYDIA
03                      /THIS IS THE FLOPPY DIAGNOSTIC
04
05
06              /FLOPPY CHECK OUT PROCEDURE
07              /
08              / 1.    IN ORDER TO PROPERLY CHECKOUT THE FLOPPY THESE STEPS
09              /       SHOULD BE FOLLOWED IN THE ORDER THAT THEY ARE LISTED
10              /       BEFORE RUNNING THE DIAGNOSTIC.
11              /
12              /       A) CHECK MSTCK (MASTER CLOCK) FOR 5MHZ CLOCK U36 PIN 3
13              /       250NS DURATION,50NS LOW,200NS HIGH
14              /
15              /       B) CHECK WRDATA- (WRITE DATA NOT) U18 PIN 1 FOR A HIGH
16              /
17              /       C) WITH DISKETTE IN FLOPPY AND DOOR CLOSED CHECK INDEX
18              /       U144 PIN 5
19              /
20              /       D) CHECK RDUR (RIGHT DEVICE UNIT READY) AND LDUR
21              /       (LEFT DEVICE UNIT READY) AT U36 PIN 8 AND 6 FOR A LOW.
22              /
23              /       E) CHECK RDCP (READ DATA CLOCK PULSE) AND RDDP (READ DAT
24              /       DATA PULSE) U91 PIN 3 AND 4
25              /
26              /       F) CHECK THAT ALL COMMAND REGISTER LINES ARE LOW U85 AND
27              /
28              /       G) CHECK RDEN (READ ENABLE) U61 PIN 6 FOR A LOW.
29              /
30              /       H) FOR NOVA COMPUTERS INSURE THAT DATA LINES ARE HIGH BU
31              /       AT +5 VOLTS.
32              /
33              /       I) LOAD THE DIAGNOSTIC
34              /
35              /2. HOW TO RUN THE DIAGNOSTIC
36              /
37              /       2.1) STARTING ADDRESS IS 200 RESTART ADDRESS IS 201.
38              /
39              /       2.2) AFTER MESSAGE ASKING FOR DEVICE # TYPE
40              /       DEVICE CODE,DRIVE #,# OF DRIVES TO BE TESTED
41              /
42              /       EXAMPLE: TYPE DEVICE # (CR)  33
43              /                 TYPE DRIVE # TO BE TESTED (CR) 0
44              /               TYPE # OF DRIVES TO BE TESTED (CR) 2
45              /
46              /              DEVICE CODE IS 33
47              /              START WITH DRIVE 0
48              /              TEST 2 DRIVES
49              /
50              /       2.3) IF FLOPPY IS RUNNING PROPERLY NO MESSAGES WILL BE
51              /       PRINTED EXCEPT **END OF PASS X
52              /
53              /       2.4) IF ERRORS ARE PRINTED THE TEST # WILL BE PRINTED AT
54              /       THE END OF THE ERROR MESSAGE.
55              /
56              /       2.5) IN ORDER TO MAKE IT EASIER TO INTERPERET THE ERROR
57              /       MESSAGES THERE IS DOCUMENTATION AT THE FRONT OF EACH TES
58              /       TO HELP IN THE UNDERSTANDING OF WHAT IS BEING DONE.
59              /
60              /3. HOW TO USE THE COMMAND STRING INTERPRETER
```

```
01              ;
02              ;       3.1) TYPE AN "I" ON THE TELETYPE WHILE THE PROGRAM IS RU
03              ;       MESSAGE WILL BE PRINTED OUT:
04              ;
05              ;       ENTER COMMAND STRING
06              ;       COMMAND,UNIT#,PATTERN,TRACK#,SECTOR#(CR)
07              ;
08              ;       TYPE COMMAND AS (READ,OR SEAK,OR WRITE,)
09              ;       TYPE UNIT # AS (0,OR 1,)
10              ;       TYPE PATTERN AS (ANY COMBINATION OF NUMBERS FROM 0-17777
11              ;       TYPE TRACK # AS (ANY COMBINATION OF NUMBERS FROM 0-114,)
12              ;       TYPE SECTOR # AS (ANY COMBINATION OF NUMBERS FROM 0-7(CR
13              ;
14              ;       IF YOU WANT TO RESTART THE PROGRAM TYPE ^R
15              ;               ***USE ONLY AFTER THE COMPLETE COMMAND STRING HA
16              ;                  ENTERED.
17              ;
18              ;4.HOW TO USE THE FUNCTION INTERPRETER
19              ;
20              ;       4.1) TYPE A ^F WHILE THE PROGRAM IS RUNNING
21              ;       MESSAGE WILL PRINT:
22              ;
23              ;       TYPE DESIRED FUNCTION (FUNC(S OR C OR)(CR)
24              ;
25              ;       TYPE FUNC=(DOA,DOB,DIA,DIB,DIC)
26              ;       IF YOU WANT A START PULSE WITH THE FUNCTION OR A CLEAR
27              ;       PULSE WITH THE FUNC TION TYPE A S OR C WITH THE FUNCTION
28              ;
29              ;       TO CONTINUE THE PROGRAM TYPE A ^R TO RESTART
30              ;
31
32
33              ;FILENAME = SWPAK
34
35
36              ;5.      SWITCH SETTINGS
37              ;
38              ;       LOCATION "SWREG" IS USED TO SELECT THE PROGRAM OPTIONS
39              ;       (NOT SYSTEM CONFIGURATION). WHILE RUNNING UNDER DTOS,
40              ;       THIS LOCATION WILL BE LOADED BY THE MONITOR,
41              ;       HOWEVER UNDER STAND ALONE AND PROGRAM LOAD MODES THIS
42              ;       LOCATION WILL BE SET ACCORDING TO THE ANSWERS SUPPLIED
43              ;       BY THE OPERATOR. IN ANY CASE THE OPTIONS CAN BE CHANGED
44              ;       OR VERIFIED BY USING ONE OF THE COMMANDS GIVEN IN SEC.
45              ;       5.2
46              ;
47              ;
48              ;5.1     SWITCH OPTIONS
49              ;       DIFFERENT BITS AND THEIR INTERPRETATION AT LOCATION
50              ;       "SWREG" IS AS FOLLOWS:
51              ;
52              ;       BIT     OCTAL   BINARY  INERPRETATION
53              ;               VALUE   VALUE
54              ;
55              ;       1               0       LOOP ON ERROR
56              ;               40000   1       SKIP LOOPING ON ERROR
57              ;
58              ;       2               0       PRINT TO CONSOLE
59              ;               20000   1       ABORT PRINT OUT TO CONSOLE
60              ;
```



```
01              ;       3               0       PRINT DETAILED ERROR ON THE
02              ;                               SELECTED DEVICE/DEVICES
03              ;               10000   1       ONLY X FAILURE REQUIRED
04              ;
05              ;       4               0       ALLOW END OF PASS PRINT OUT
06              ;               04000   1       SUPPRESS END OF PASS PRINT OUT
07              ;
08              ;       5               0       DO NOT PRINT ON THE LINE PRINTER
09              ;               02000   1       PRINT ON THE LINE PRINTER
10              ;
11              ;       6               0       DO NOT HALT ON ERROR
12              ;               01000   1       HALT ON ERROR
13              ;
14              ;       7               0       DO NOT PRINT PASSING OF
15              ;                               EACH TEST ON PRINTING DEVICE
16              ;               00400   1       PRINT PASSING OF EACH TEST
17              ;
18              ;5.2     SWITCH COMMANDS
19              ;       ONCE THE PROGRAM STARTS EXECUTING THE STATE OF ANY OF
20              ;       THE BITS CAN BE CHANGED BY HITTING KEYS 1 THROUGH 6, THE
21              ;       PROGRAM WILL CONTINUE RUNNING AFTER UPDATING THE OPTIONS
22              ;       EACH KEY WILL COMPLEMENT THE STATE OF THE BIT AFFILIAT-
23              ;       ED WITH IT, THUS BIT 4 CAN BE ALTERED BY HITTING KEY 4.
24              ;       SETTIN OF ANY BIT OF LOCATION "SWREG" WILL SET BIT 0.
25              ;       (DEFAULT MODE IS DEFINED AS ALL BITS OF SWREG SET TO 0)
26              ;       THE PROGRAM CAN BE LOCKED INTO SWITCH MODIFICATION MODE
27              ;       BY TYPING A 0, IN WHICH CASE MORE THAN ONE BITS CAN BE
28              ;       CHANGED BEFORE THE CONTROL IS ALLOWED TO RETURN TO THE
29              ;       MAIN PROGRAM.
30              ;
31              ;5.2.1   OTHER COMMANDS
32              ;
33              ;       "CR"    A "RETURN" CAN BE TYPED TO CONTINUE THE PROGRAM
34              ;               AFTER ITS LOCKED IN A SWITCH MODIFICATION MODE
35              ;
36              ;       ^D      THIS COMMAND GIVEN AT ANY TIME WILL RESET "SWREG
37              ;               TO DEFAULT MODE AND RESTART THE PROGRAM.
38              ;
39              ;       ^R      THIS COMMAND GIVEN AT ANY TIME WILL RESTART THE
40              ;               PROGRAM. SWITCHES ARE LEFT WITH THE VALUES THEY
41              ;               HAD BEFORE THE COMMAND WAS ISSUED.
42              ;
43              ;       ^O      THIS COMMAND GIVEN AT ANY TIME WILL CAUSE THE
44              ;               PROGRAM CONTROL TO GO TO ODT (SEE SEC. 6)
45              ;
46              ;       ^S      PRINT THE SUMMARY OF ERRORS ACCUMULATED SO FAR.
47              ;
48              ;       ^       THIS COMMAND GIVEN AT ANY TIME WILL PRINT THE
49              ;               CURRENT OPERATING MODES.
50              ;
51              ;
52              ;
53              ;6.      ERROR MESSAGES
54              ;
55              ;       ERRORS ARE FIRST REPORTED WITH A MESSAGE INDICATING
56              ;       WHAT THE PROBLEM IS, THEN THE ACCUMULATORS ARE PRINTED:
57              ;
58              ;       CRY     AC0     AC1     AC2     AC3     PC
59              ;
60              ;       REFER TO THE ERROR PC IN THE LISTING FOR A DISCRIPTION O
```

```
 0005 FPYDI
01                ;       IN THE ACCUMULATORS.
```

```
10006 FPYDI
01
02
03
04          000000 FPY=0
05
06          000000 .LOC 0
07 00000 010026 DIRT
08 00001 000000 0
09          000020 .LOC 20
10 00020 000000 BUFPT:  0               ;COMPARE BUFFER POINTER
11 00021 000000 CBUFP:  0                       ;COMPARE BUFFER POINTER
12          000045 .LOC 45
13 00045 000060 EGGS
14 00046 000000 0
15 00047 005630 ERR?
16 00050 000010 .BLK 8.
17 00060 000000 EGGS:   0
18 00061 000000         0
19 00062 000000         0
20 00063 000000         0
21 00064 000000         0
22 00065 000000         0
23 00066 000000 BACK:   0                       ;RETURN POINTER
24 00067 000000 RTNH:   0                       ;RETURN POINTER
25 00070 000000 RT?N:   0                       ;RETURN POINTER
26 00071 000000 RTN?N:  0                       ;RETURN POINTER
27 00072 000000 RTWCP:  0                       ;RETURN FOR WRITE/READ COMPARE
28 00073 005614 IITR:   ITR?R                   ;ERROR COUNTER POINTER
29 00074 003663 IERME:  ERMES                   ;ERROR MESSAGE POINTER
30 00075 003322 IRPRE:  RPRE                            ;POINTER TO READ PREAMBL
31 00076 003304 IWRSE:  WRSET                           ;POINTER TO SET WRITE RO
32 00077 003537 IDPI:   DRI                             ;POINTER TO DRIVE SETUP
33 00100 002742 ISEAK:  SEAK                            ;POINTER TO SEEK TRACK
34 00101 005423 ISET:   SETEL
35 00102 003106 IRITE:  DORITE
36 00103 003221 IREAD:  REED                            ;POINTER TO READ ROUTINE
37 00104 005630 IERR:   ERR?                            ;POINTER TO ERROR HANDLE
38 00105 003257 ITIME:  TIME                            ;WAIT FOR DONE ROUTINE
39 00106 003555 IERCK:  ERRCK                           ;CHECK ERROR BITS
40 00107 005765 IMES?S: MES?S                           ;TYPE A MESSAGE
41 00110 005765 IMESS:  MES?S                           ;TYPE A MESSAGE
42 00111 003545 IDCOA:  DOOA            ;POINTER TO DOAS INSTRUCTION
43 00112 002333 ICOMT:  COMTB           ;POINTER TO COMMAND TABLE
44 00113 002343 ICOMY:  COMY-1          ;POINTER TO COMMAND FUNCTION STO
45 00114 000000 SECEC:  0                       ;SECTOR ERROR DETECTED C
46 00115 000000 FPYRC:  0               ;FLOPPY NOT READY COUNT
```

```
0007 FPYDI
01
02 00116 000000 FLAGS:  0                               ;FLAGS STORAGE
03              ;////////////////////////////////////////////////////////////
04              ;/ 0 / 1 / 2 / 3/ 4 / 5 / 6 / 7 / 8 / 9 / 10/ 11/ 12/ 13/ 14/ 15
05              ;////////////////////////////////////////////////////////////
06              ; D   R ///////// NOT DEFINED YET ////////////////////////////
07              ; R   E ////////////////////////////////////////////////////////
08              ; I   A ////////////////////////////////////////////////////////
09              ; V   D ////////////////////////////////////////////////////////
10              ; E   / ////////////////////////////////////////////////////////
11              ; 0   W ////////////////////////////////////////////////////////
12              ; /   R ////////////////////////////////////////////////////////
13              ; 1   I ////////////////////////////////////////////////////////
14              ;     T ////////////////////////////////////////////////////////
15              ;     E ////////////////////////////////////////////////////////
16              ;////////////////////////////////////////////////////////////
17
18 00117 000000 FUNWD:  0                               ;FUNCTION WORD SETUP STD
19 00120 000007 SECTST: 7                       ;TEST SECTOR NUMBER
20 00121 177770 CN7:    -10                     ;CONSTANT -10
21 00122 000023 C23:    23                      ;CONTROL S CONSTANT
22 00123 000017 C17:    17                      ;CONTROL O CONSTANT
23 00124 000053 C42:    43.                     ;CONSTANT 42
24 00125 000000 TIMER:  0                       ;TIMEING STORAGE
25 00126 000000 RCOUNT: 0                       ;ROTATE COUNT
26 00127 003346 IWCP2:  WCPA2
27 00130 006303 ITIN?O: TIN?O
28 00131 003332 IWCPAT: WCPAT                   ;WRITE AND COMPARE PATTERNS POIN
29 00132 007377 IRBUF:  RBUF                    ;READ BUFFER POINTER
30 00133 006776 IWBUF:  BUF                     ;WRITE BUFFER POINTER
31 00134 007375 BUFED:  BUFND-1                 ;BUFFER END
32 00135 000000 RITFUN: 0                       ;RIGHT FUNCTION STORAGE
33 00136 000000 TSTNM:  0               ;TEST NUMBER STORAGE
34 00137 000000 DIFF:   0                       ;DIFFERENCE STORAGE
35 00140 000233 ILFUN:  233                     ;ILLEAGLE FUNCTION
36 00141 000000 SETTLE: 0                       ;SETTLE FUNCTION
37 00142 000020 REDFUN: 20                      ;READ FUNCTION
38 00143 000040 WITFUN: 40                      ;WRITE FUNCTION
39 00144 000010 RDPA:   10                      ;READ PREAMBLE FUNCTION
40 00145 000001 STEPOU: 1                       ;STEP HEAD OUT FUNCTION
41 00146 000002 STEPIN: 2                       ;STEP HEAD IN FUNCTION
42 00147 000115 C77:    77.                             ;MAX NUMBER OF TRACKS
43 00150 000132 C90:    90.                             ;TIMING CONSTANT
44 00151 052525 ALONZ:  052525          ;ALTERNATING ONE'S
45 00152 000000 BLANK:  0                               ;ALL ZEROS
46 00153 177777 ONEZ:   177777                          ;ALL ONE'S
47 00154 146314 PATRK:  146314                          ;TRACK TEST PATTERN
48 00155 000034 C1777:  34                              ;SECTOR MASK
49 00156 000177 C1776:  177                             ;TRACK MASK
50 00157 100000 BIT0:   100000                          ;BIT 0 REPRESENTATION
51 00160 040000 BIT1:   040000                          ;BIT 1 REPRESENTATION
52 00161 020000 BIT2:   020000                          ;BIT 2 REPRESENTATION
53 00162 010000 BIT3:   010000                          ;BIT 3 REPRESENTATION
54 00163 004000 BIT4:   004000                          ;BIT 4 REPRESENTATION
55 00164 002000 BIT5:   002000                          ;BIT 5 REPRESENTATION
56 00165 001000 BIT6:   001000                          ;BIT 6 REPRESENTATION
57 00166 000400 BIT7:   000400                          ;BIT 7 REPRESENTATION
58 00167 000200 BIT8:   000200                          ;BIT 8 REPRESENTATION
59 00170 000100 BIT9:   000100                          ;BIT 9 REPRESENTATION
60 00171 000040 BIT10:  000040                          ;BIT 10 REPRESENTATION
```

```
 0008 FPYDI
01 00172 000020 BIT11:  000020                          ;BIT 11 REPRESENTATION
02 00173 000010 BIT12:  000010                          ;BIT 12 REPRESENTATION
03 00174 000004 BIT13:  000004                          ;BIT 13 REPRESENTATION
04 00175 000002 BIT14:  000002                          ;BIT 14 REPRESENTATION
05 00176 000001 BIT15:  000001                          ;BIT 15 REPRESENTATION
06 00177 006776 IBUF:   BUF                             ;POINTER  TO BUFFER
07
08
09
```

```
10009 FPYDI
01
02        000200 .LOC 200
03 00200  002217          JMP     @ISTART         /GO TO START OF PROGRAM
04 00201  002237          JMP     @IBEGN          /RESTART PROGRAM
05 00202  000000 HELP:    0                       /TEST STARTING ADDRESS
06 00203  000000 PASS:    0                       /PASS CPUNT
07 00204  000000 AC3?:    0                       /STORAGE FOR AC3
08 00205  000000 SWREG:   0               /SWITCH REGISTER STORAGE
09 00206  003511 IINP?:   SERVS
10 00207  006042 ICRL?F:  CRL?F
11 00210  000064 IPDC?S:  PDC?S
12 00211  006056 IPOC?T:  POC?T
13 00212  006052 IZOC?T:  ZOC?T
14 00213  005466 IENT?R:  ENT?R
15 00214  005506 ICYC?E:  CYC?E
16 00215  005630 IERR?:   ERR?
17 00216  006203 ITYP?E:  TYP?E
18 00217  000500 ISTART:  START
19 00220  006410 IINP:    INP?J
20 00221  006553 IIODT:   ODT?J
21 00222  005367 ISUM:    SUMME
22 00223  000000 RET:     0
23 00224  000000 TIMIT:   0                       /TIMMER FOR HEAD LOADED
24 00225  000000 REET:    0                       /RETURN FOR SETTLE HEADS
25 00226  000000 R?TN:    0                       /RETURN POINTER FOR SEAK
26 00227  042104 KDD:     042104
27 00230  047523 KOS:     047523
28 00231  000006 CF:      6               /CONTROL F
29 00232  002350 IFOWL:   FOWL
30 00233  000111 CC:      "I
31 00234  000000 NUDEV:   0
32 00235  000000 COENT:   0
33 00236  000015 C15:     15
34 00237  000571 IBEGN:   BEGN
35 00240  000000 PASST:   0                       /DRIVE PASS COUNTER
36 00241  002444 IFUNX:   FUEX
37 00242  002454 IFNTB:   FUNTB
38 00243  060077 NOFUN:   NIO CPU
39 00244  002741 IBFPED:  BFPED
40 00245  002666 IBFPAT:  BFPAT
41 00246  002643 IDOPAT:  DOPAT
42 00247  002512 ITET:    TET
43 00250  002344 ICOM1:   COMY                    /POINTER FOR TRACK STORAGE
44 00251  002345 ICOM2:   COMY+1                  /POINTER TO SECTOR STORAGE
45 00252  002227 ICOMS:   COMST                   /POINTER TO COMMAND STRING ENTRY
46 00253  002320 IGOUD:   GOODF                   /POINTER TO GOOD RETURN FROM TUR
47 00254  002325 IGOOS:   GOOS                    /CORECT TURMINATOR CHECK FOR TRA
48 00255  002343 ICOME:   COMEX                   /POINTER TO COMMAND STRING EXICU
49 00256  003657 ISEQER:  SEQER                   /SEQUIENCE ERROR POINTER
50 00257  002000 ERTS1:   0                       /ERROR COUNTER FOR ERROR CHECK
51 00260  000000 COMRE:   0
52 00261  000000 UNITA:   0                       /DEVICE CODE STORAGE
53 00262  000000 SERCT:   0
54 00263  000000 WPAT:    0
55 00264  000000 ERBTC:   0                       /ERROR BIT SET COUNT
56 00265  000000 ILLFC:   0                       /ILLEAGLE FUNCTION COUNT
57 00266  000000 WRFLC:   0                       /WRITE FAULT DETECTED COUNT
58 00267  000000 DCLAT:   0                       /DATA CHANNEL LATENCY DETECTED C
59 00270  000000 CKWDC:   0                       /CHEK WORD ERROR DETECTED COUNT
60 00271  000602 IBGN1:   BEGN1
```

```
0010 FPYDI
01 00272  000000 DRI1:    0
02 00273  003412 ICPET:   CPET
03 00274  001102 ICWE2:   CWE2
04 00275  001110 ICWE6:   CWE6
05 00276  002500 PTET:    PTSTB
06 00277  002550 ITELP:   TETLP
07 00300  000000 LPWM:    0
08 00301  002044 IPLBG:   PLBGN
09 00302  000000 LDWRD:   0
10 00303  000112 CJ:      "J
11 00304  000015 TETMX:   15                      /LAST TEST #
12 00305  010126 CWE9:    ISZ     RCOUNT          /INCREMENT COUNTER
13 00306  002274          JMP     @ICWE2          /CONTINUE WAITING FOR DIB
14 00307  002275          JMP     @ICWE6          /DIB NOT RESPONDING
15        000500 .LOC 500
```

```
0011 FPYDI
01                      JSR     @ITET
02                      JSR     @ICYC?E         ;GO TO CYCLE ON TEST ROUTINE
03                      SUB     0,0             ;CLEAR AC0
04                      STA     0,ERTS1         ;CLEAR ERTS1
05              **%
06
07
08 00500 102400 START:  SUB     0,0             ;CLEAR AC0
09 00501 040272         STA     0,DRI1          ;CLEAR DRIVE 1 FLAG
10 00502 040234         STA     0,NUDEV         ;CLEAR # OF DEVICES
11 00503 006110         JSR     @IMESS          ;ASK OPERATOR FOR DEVICE CODE
12 00504 003761         NUMU
13 00505 006130         JSR     @ITIN?O         ;ACCEPT THE INPUT
14 00506 000772         JMP     START           ;TYPED A LETTER
15 00507 044261         STA     1,UNITA         ;SAVE THE DEVICE #
16 00510 006110         JSR     @IMESS          ;ASK FOR UNIT TO BE TESTED
17 00511 005150         WHDR
18 00512 006130         JSR     @ITIN?O         ;GO GET THE #
19 00513 000765         JMP     START           ;ALPHA TYPED
20 00514 044272         STA     1,DRI1          ;SAVE THE DRIVE #
21 00515 006110         JSR     @IMESS          ;ASK FOR # OF DRIVES TO BE TESTE
22 00516 005171         WMDR
23 00517 006130         JSR     @ITIN?O         ;ACCEPT THE NUMBER
24 00520 000760         JMP     START           ;ALPHA TYPED
25 00521 125225         MOVZR   1,1,SNR         ;WAS IT 1 OR 2
26 00522 000404         JMP     WRDK3           ;1
27 00523 044234         STA     1,NUDEV         ;IT WAS A 2
28 00524 125400         SUB     1,1             ;CLEAR AC0
29 00525 044272 WRDK4:  STA     1,DRI1          ;INSURE THAT WE START ON DRIVE 0
30 00526 006101 WRDK3:  JSR     @ISET           ;SETTLE THE DRIVE
31 00527 024133         LDA     1,IWBUF         ;GET WRITE BUFFER ADDRESS
32 00530 044177         STA     1,IBUF          ;FIX BUFFER POINTER
33 00531 006103         JSR     @IREAD          ;READ TRACK 0 SECTOR 1
34 00532 000000         0
35 00533 000001         1
36 00534 177777         -1
37 00535 026073         LDA     1,@IITR         ;SEE IF AN ERROR WAS ENCOUNTERED
38 00536 125004         MOV     1,1,SZR         ;ANY ERRORS
39 00537 000420         JMP     WRDK6           ;YES
40 00540 006106         JSR     @IERCK          ;SEE IF ANY ERRORS
41 00541 024133         LDA     1,IWBUF         ;GET THE WRITE BUFFER ADDRESS
42 00542 044177         STA     1,IBUF          ;SAVE IT
43 00543 026177         LDA     1,@IBUF         ;GET THE FIRST CHARACTERS
44 00544 030227         LDA     2,KDD           ;GET DD
45 00545 132415         SUB#    1,2,SNR         ;DO WE HAVE A MATCH
46 00546 000416         JMP     WRDK            ;YES
47 00547 010177         ISZ     IBUF            ;NO BUT LETS CHECK THE NEXT FOR
48 00550 026177         LDA     1,@IBUF         ;GET SECOND WORD
49 00551 030230         LDA     2,KOS           ;GET OS
50 00552 132415         SUB#    1,2,SNR         ;IS IT OS
51 00553 000411         JMP     WRDK            ;YES
52 00554 024157         LDA     1,BIT0          ;GET DRIVE 1 FLAG
53 00555 044116         STA     1,FLAGS         ;SET DRIVE ONE FLAG ANYWAY
54 00556 000413         JMP     BEGN            ;GO START THE TESTS
55 00557 006110 WRDK6:  JSR     @IMESS          ;TYPE MESSAGE
56 00560 004366         FLONG                   ;FLOPPY NO GOOD
57 00561 006130         JSR     @ITIN?O         ;WAIT TO PRECEED
58 00562 000775         JMP     WRDK6           ;NO ALPHA
59 00563 000406         JMP     BEGN            ;ACCEPT A (CR),(,),( ),(LF)
60 00564 006110 WRDK:   JSR     @IMESS          ;TELL HIM TO CHANGE THE DISK
```

```
0012 FPYDI
01 00565 004214         LSDK
02 00566 006130         JSR     @ITIN?O         ;ACCEPT A KEY
03 00567 000737         JMP     WRDK3           ;A KEY WAS HIT
04 00570 000736         JMP     WRDK3           ;MAKE SURE ITS A SCRATCH DISK
05                RES?T:
06 00571 102400 BEGN:   SUB     0,0             ;CLEAR AC0
07 00572 040300         STA     0,LPNM          ;0 THE LOOP #
08 00573 040257         STA     0,ERTS1         ;CLEAR ERROR COUNTER
09 00574 024272         LDA     1,DRI1          ;GET THE DRIVE INDICATOR
10 00575 125005         MOV     1,1,SNR         ;IF NOT 0 LEAVE DEVICE
11                                              ;SET UP FOR DRIVE 1
12 00576 040116         STA     0,FLAGS
13 00577 040203         STA     0,PASS
14 00600 040136         STA     0,TSTNM
15 00601 040235         STA     0,COENT         ;CLEAR COMMAND ENTRY
16 00602 000401 BEGN1:  JMP     TST1            ;GO DO THE TESTS
17
```

```
0013 FPYDI
01                ;*****************************************************************
02                ;TST#1  <CLEAR STATUS WORD AND TEST THAT NO ERROR BITS SET>
03                ;*****************************************************************
04
05 00603 010136 TST1:   ISZ     TSTNM           ;ENTERING A NEW TEST
06                ;       THIS TEST VERIFIES THAT THE FLOPPY IS READY.
07                ;       IF THE FLOPPY DOES NOT GO READY THE MESSAGE
08                ;       FLOPPY NOT READY IS PRINTED. IT ALSO VERIFIES
09                ;       THAT NO ERRORS ARE SET AFTER A START PULSE.
10                ;/////////////////////////////////////////////////////////////////
11
12 00604 024205          LDA     1,SWREG         ;GET THE SWITCH REGISTERS
13 00605 030166          LDA     2,BIT7          ;GET SWITCH 7
14 00606 133414          AND#    1,2,SZR         ;IS 7 SET
15 00607 000415          JMP     STAT6           ;YES SKIP TYPE OUT
16 00610 006110          JSR     @IMESS          ;PRINT HEADER
17 00611 004455          DKMES                   ;DISK #
18 00612 024261          LDA     1,UNITA         ;GET THE DEVICE CODE
19 00613 006212          JSR     @IZOC?T         ;PRINT VALUE
20 00614 006110          JSR     @IMESS          ;PRINT HEADER
21 00615 004466          DRMES                   ;DRIVE #
22 00616 126400          SUB     1,1             ;CLEAR AC0
23 00617 034116          LDA     3,FLAGS         ;GET THE FLAGS
24 00620 030157          LDA     2,BIT0          ;GET THE DRIVE BIT
25 00621 173414          AND#    3,2,SZR         ;IS IT DRIVE 0
26 00622 125400          INC     1,1             ;NO ITS DRIVE 1
27 00623 006212          JSR     @IZOC?T
28 00624 006276 STAT6:   JSR     @PTET           ;PRINT THE TEST #
29 00625 006213          JSR     @IENT?R         ;GO SET UP TEST
30 00626 000002          2
31 00627 102400          SUB     0,0             ;CLEAR AC0
32 00630 040126          STA     0,RCOUNT        ;SAVE 0 IN COUNTER LOC
33 00631 006111 STAT3:   JSR     @IDUOA          ;ISSUE FUNCTION
34 00632 060400          DIA     0,FPY           ;READ FLOPPY STATUS
35 00633 101103          MOVL    0,0,SNC         ;FLOPPY READY ?
36 00634 000410          JMP     STAT2           ;YES FLOP IS READY
37 00635 010126          ISZ     RCOUNT          ;INCREMENT COUNT
38 00636 000773          JMP     STAT3           ;GO BACK AND LOOK FOR READY
39 00637 006074          JSR     @IERME          ;REPORT FLOPPY NOT READY
40 00640 004267          FPYRD
41 00641 010115          ISZ     FPYRC           ;INCREMENT FLOPPY NOT READY COUN
42 00642 060077          NIO     CPU             ;NO OP
43 00643 000415          JMP     STAT5           ;GO TO CYCLE
44 00644 020141 STAT2:   LDA     0,SETTLE                ;GET READY TO SETTLE DOW
45 00645 006077          JSR     @IDRI           ;SET UP FOR CORRECT DRIVE
46 00646 006111          JSR     @IDUOA          ;ISSUE FUNCTION
47 00647 061100          DUAS    0,FPY                   ;TELL THE DRIVE TO SETTL
48 00650 006105          JSR     @ITIME                  ;WAIT FOR DONE
49 00651 006101          JSR     @ISET                   ;ISSUE SOMETHING BEFORE
50                                                       ;READING STATUS
51 00652 101415          INC#    0,0,SNR                 ;DID WE GET A RESPONCE
52 00653 000403          JMP     STAT1                   ;NO,REPORD DEVICE NOT RE
53 00654 006106          JSR     @IERCK                  ;CHECK FOR ERRORS
54 00655 000404          JMP     STAT4                   ;CONTINUE TESTING
55 00656 006074 STAT1:   JSR     @IERME                  ;REPORT DEVICE NOT RESPO
56 00657 004566          DDRER
57 00660 006104 STAT5:   JSR     @IERR                   ;PRINT THE AC'S
58                STAT4:  LOOPS
59 00661 006247          JSR     @ITET
60 00662 006214          JSR     @ICYC?E         ;GO TO CYCLE ON TEST ROUTINE
```

```
0014 FPYDI
01 00663 102400          SUB     0,0             ;CLEAR AC0
02 00664 040257          STA     0,ERTS1         ;CLEAR ERTS1
03
04
05 00665 000402          JMP     .+2
06 00666 000001 TN1:     1
07 00667 020136 ET1:     LDA     0,TSTNM
08 00670 024776          LDA     1,TN1
09 00671 122454          SUBO#   1,0,SZR         ;CHECK THE PROGRAM FLOW
10 00672 006256          JSR     @ISEQER
```

```
 0015 FPYDI
01                      ;*****************************************************************
02                      ;TST#2  <TEST HEAD HOME BIT IN STATUS WORD>
03                      ;*****************************************************************
04
05 00673 010136 TST2:   ISZ     TSTNM           ;ENTERING A NEW TEST
06                      ;       TEST THAT THE HEAD HOME BIT CAN BE DETETED
07                      ;       AND THAT IT IS ACTUALLY ON TRACK 00,
08                      ;       THE PROGRAM READS THE PREAMBLE TO VERIFIE THAT
09                      ;       IT IS ACTUALLY ON 00,IF READ PREAMBLE IS NOT
10                      ;       WORKING AN ERROR MESSAGE IS PRINTED
11                      ;       (HEAD IS NOT MOVING OR DISK UNFORMATED),
12                      ;       THIS SAYS THAT :
13                      ;               1) THE HEAD DID NOT MOVE PROPERLY
14                      ;               2) THE DISK IS UNFORMATED
15                      ;               3) THE DIC FUNCTION IS NOT WORKING
16                      ;               4) THE SAVE SIGNAL DID NOT SAVE THE HEADER ADDRE
17                      ;
18                      ;       POSSIBLE PROBLEMS ARE:
19                      ;       1) CHECK RP SIGNAL U85 PIN 7 (SHOULD GO HIGH
20                      ;       WHILE ALL OTHER SIGNALS REMAIN LOW)
21                      ;
22                      ;       2) VERIFY SAVE- AT U51 PIN 8 GOES LOW (STATIC HIGH)
23                      ;       ***SAVE- IS A VERY NARROW SIGNAL AND IS HARD TO SEE
24                      ;
25                      ;       3) CHECK SPDI AND SPCK AT U115 PIN 1 AND 8
26                      ;
27                      ;       4) CHECK DIC IS HAPPENING AT U138 PIN 9
28                      ;/////////////////////////////////////////////////////////////
29
30
31 00674 006276         JSR     @PTET           ;PRINT THE TEST #
32 00675 006213         JSR     @IENT?R                 ;GO SETUP TEST
33 00676 000100         100
34 00677 006100         JSR     @ISEAK                  ;GO SEEK A TRACK OTHER T
35 00700 000005         5                               ;WE WILL USE TRACK 5
36 00701 000000         0
37 00702 006101         JSR     @ISET                   ;DO SOMTHING TO BEABLE T
38 00703 006111         JSR     @IDOOA                  ;ISSUE FUNCTION
39 00704 060600         DIAC    0,FPY                   ;READ STATUS
40 00705 024160         LDA     1,BIT1                  ;GET BIT TO TEST HEAD HO
41 00706 107434         ANDZ#   0,1,SZR                 ;IS HEAD HOME?
42 00707 000414         JMP     HHM1                    ;YES HEAD HOME BIT IN ER
43 00710 006100         JSR     @ISEAK                  ;NO,OK SEEK HEAD HOME
44 00711 000000         0                               ;TRACK 0
45 00712 000000         0
46 00713 006101         JSR     @ISET                   ;DO SOMTHING TO BE ABLE
47 00714 024160         LDA     1,BIT1                  ;GET BIT TO TEST HEAD HO
48 00715 107434         ANDZ#   0,1,SZR                 ;IS HEAD HOME NOW?
49 00716 000410         JMP     .+10                    ;GO TO NEXT TEST
50 00717 006074         JSR     @IERME                  ;HEAD HOME NOT BEING DET
51 00720 003676         HDHMC
52 00721 006104         JSR     @IERR                   ;GO REPORT ERROR
53 00722 000404         JMP     .+4                     ;GO TO NEXT TEST
54 00723 006074 HHM1:   JSR     @IERME                  ;HEAD HOME ALWAYS DETECT
55 00724 004627         HDHMS
56 00725 006104         JSR     @IERR                   ;GO REPORT ERROR
57                      LOOPS
58 00726 006247         JSR     @ITET
59 00727 006214         JSR     @ICYC?E         ;GO TO CYCLE ON TEST ROUTINE
60 00730 102400         SUB     0,0             ;CLEAR AC0
```

```
 0016 FPYDI
01 00731 040257         STA     0,ERTS1         ;CLEAR ERTS1
02
03
04
05 00732 000402         JMP     .+2
06 00733 000002 TN2:    2
07 00734 020136 ET2:    LDA     0,TSTNM
08 00735 024776         LDA     1,TN2
09 00736 122454         SUB0#   1,0,SZR         ;CHECK THE PROGRAM FLOW
10 00737 006256         JSR     @ISEQER
```

```
0017 FPYDI
01                ;*****************************************************************
02                ;TST#3  <TEST HEAD LOAD BIT>
03                ;*****************************************************************
04
05 00740 010136 TST3:   ISZ     TSTNM           ;ENTERING A NEW TEST
06                ;       THIS TEST VERIFIES THAT THE HEAD LOAD BIT WORKS
07                ;       AND THAT IT SETS AFTER SIX HEAD COMMANDS.
08                ;
09                ;       FIRST THE PROGRAM WAITS FOR THE HEAD LOAD BIT TO
10                ;       GO AWAY.IF THIS DOES NOT HAPPEN THE AN ERROR IS RE-
11                ;       PORTED .THEN A SETTLE COMMAND IS ISSUED ONE AT A TIME
12                ;       CHECKING THAT AFTER EVERY SETTLE TO SEE IF HEAD LOAD
13                ;       BIT IS SET.AFTER THE SIXTH HEAD LOAD THE PROGRAM WILL
14                ;       REPORT THAT THE HEAD IS NOT LOADED IF IT HASN!T SET.
15                ;       THE PROGRAM ALSO WILL REPORT AN ERROR IF IT SETS BEFORE
16                ;       SIX SETTLES HAVE BEEN ISSUED.
17                ;/////////////////////////////////////////////////////////////////
18
19
20 00741 006276          JSR     @PTET           ;PRINT THE TEST #
21 00742 006213          JSR     @IENT?R         ;SET UP THE TEST
22 00743 000100          100                     ;FOR 100 ITERATIONS
23 00744 102400          SUB     0,0             ;CLEAR AC0
24 00745 040126          STA     0,RCOUNT        ;STORE IN COUNTER
25 00746 006111 HDLD6:   JSR     @IDOOA          ;ISSUE FUNCTION
26 00747 060400          DIA     0,FPY           ;READ STATUS
27 00750 024161          LDA     1,BIT2          ;GET HEAD LOAD BIT
28 00751 107414          AND#    0,1,SZR         ;HEAD LOAD SET
29 00752 000402          JMP     HDLD4           ;YES
30 00753 000407          JMP     HDLD5           ;NO
31 00754 010126 HDLD4:   ISZ     RCOUNT          ;INCREMENT COUNTER
32 00755 000771          JMP     HDLD6           ;KEEP WAITING FOR HEAD LOAD
33                                               ;TO GO AWAY
34 00756 006074          JSR     @IERME          ;TELL OPERATOR HEAD LOAD ALWAYS
35 00757 004145          HDLDF
36 00760 006104          JSR     @IERR           ;DO REGULAR ERR MESS
37 00761 000436          JMP     HDLD2           ;GO TO LOOPS
38 00762 020175 HDLD5:   LDA     0,BIT14         ;GET A 4
39 00763 024174          LDA     1,BIT13         ;GET A 4
40 00764 123000          ADD     1,0             ;ADD TO MAKE 6
41 00765 100400          NEG     0,0             ;MAKE IT A NEGITIVE 6
42 00766 040126          STA     0,RCOUNT        ;SAVE AT COUNTER
43 00767 020141          LDA     0,SETTLE        ;GET A SETTLE FUNC
44 00770 006077          JSR     @IDRI           ;SET IT UP FOR CORRECT DRIVE
45 00771 006111 HDLD3:   JSR     @IDOOA          ;ISSUE FUNCTION
46 00772 061100          DOAS    0,FPY           ;SETTLE THE FLOPPY
47 00773 006105          JSR     @ITIME          ;WAIT FOR DONE
48 00774 010126          ISZ     RCOUNT          ;INCREMENT THE COUNTER
49 00775 000412          JMP     HDLD1           ;GO SEE IF HEAD LOAD
50                                               ;SET ANY WAY
51 00776 006111          JSR     @IDOOA          ;ISSUE FUNCTION
52 00777 060400          DIA     0,FPY           ;READ THE STATUS
53 01000 024161          LDA     1,BIT2          ;GET HEAD LOAD BIT
54 01001 107414          AND#    0,1,SZR         ;IS HEAD LOAD SET
55 01002 000415          JMP     HDLD2           ;YES GO TO CYCLE
56 01003 006074          JSR     @IERME          ;TELL OPERATOR THAT HEAD LOAD
57 01004 005107          HNLDS                   ;IS NOT SET AFTER 6 START PULSES
58 01005 006104          JSR     @IERR           ;GIVE REG ERROR MESS
59 01006 000411          JMP     HDLD2           ;GO TO CYCLE
60 01007 006111 HDLD1:   JSR     @IDOOA          ;ISSUE FUNCTION
```

```
0018 FPYDI
01 01010 060400          DIA     2,FPY           ;READ THE STATUS
02 01011 024161          LDA     1,BIT2          ;GET THE HEAD LOAD BIT
03 01012 147415          AND#    2,1,SNR         ;IS HEAD LOAD SET
04 01013 000756          JMP     HDLD3           ;NOT ISSUE ANOTHER SETTLE
05 01014 006074          JSR     @IERME          ;TELL OPERATOR THE HEAD LOAD
06 01015 005273          HLSTS                   ;SET BEFORE 6 START PULSES ISSUE
07 01016 006104          JSR     @IERR           ;REPORT REG ERROR MESS
08                HDLD2: LOOPS
09 01017 006247          JSR     @ITET
10 01020 006214          JSR     @ICYC?E         ;GO TO CYCLE ON TEST ROUTINE
11 01021 102400          SUB     0,0             ;CLEAR AC0
12 01022 040257          STA     0,ERTS1         ;CLEAR ERTS1
13
14 01023 000402          JMP     .+2
15 01024 000003 TN3:     3
16 01025 020136 ET3:     LDA     0,TSTNM
17 01026 024776          LDA     1,TN3
18 01027 122454          SUBO#   1,0,SZR         ;CHECK THE PROGRAM FLOW
19 01030 006256          JSR     @ISEQER
```

```
 0019 FPYDI
01                  /*****************************************************************
02                  /TST#4  <TEST ERROR SETS WITH CHECK WORD ERROR>
03                  /*****************************************************************
04
05 01031 010136 TST4:   ISZ     TSTNM           /ENTERING A NEW TEST
06                  /       THE TEST FIRST WRITES ALL 0'S ON TRACK 25
07                  /       SECTOR 0,THEN BIT15 IS SET IN THE FIRST WORD
08                  /       OF THE WRITE BUFFER,ANOTHER WRITE IS ISSUED
09                  /       TO THE SAME TRACK AND SECTOR AND IS ABORTED
10                  /       AFTER THE FLOPPYS BUFFER ADDRESS HAS INCREASED BY 7
11                  /       BY ISSUING A READ PREAMBLE TO THE FLOPPY.
12                  /       A READ IS THE ISSUED TO THE FLOPPY FOR THE
13                  /       SAME TRACK AND SECTOR , STATUS IS THEN CHECKED
14                  /       FOR A CHECK WORD ERROR AND ERROR,IF CHECK WORD
15                  /       ERROR DOES NOT OCCUR START LOOKING AT THE ERROR  ,
16                  /       TREE LOGIC.
17                  ////////////////////////////////////////////////////////////////
18
19
20 01032 006276          JSR     @PTET           /PRINT THE TEST #
21 01033 006213          JSR     @IENT?R                 /GO SET UP TEST
22 01034 000100          100
23
24 01035 024133          LDA     1,IWBUF         /GET WRITE BUFFER ADDRESS
25 01036 044235          STA     1,COENT         /SET THE COMMAND ENTRY FLAG
26 01037 044177          STA     1,IBUF          /SETUP THE BUFFER ADDRESS
27 01040 024152          LDA     1,BLANK         /GET ALL 0'S
28 01041 044263          STA     1,WPAT          /STORE AT WHAT PATTERN
29 01042 006127          JSR     @INCP2          /CLEAR THE TRACK AND SECTOR
30 01043 000025          25                      /TRACK 25
31 01044 000000          0                       /SECTOR 0
32 01045 024777          LDA     1,.-1           /CLEAR AC0
33 01046 044235          STA     1,COENT         /CLEAR COMMAND ENTRY SWITCH
34 01047 044126          STA     1,RCOUNT        /CLEAR RCOUNT FOR TIMING
35 01050 006101          JSR     @ISET           /MAKE SURE HEAD IS LOADED
36 01051 006075 CWE1:    JSR     @IRPRE          /GO TELL FLOPPY TO READ PREAMBLE
37 01052 006111          JSR     @IDOOA          /ISSUE FUNCTION
38 01053 066400          DIC     1,FPY           /GET THE SECTOR TRACK WORD FROM
39 01054 030155          LDA     2,C1777         /GET THE SECTOR MASK
40 01055 147400          AND     2,1             /MASK OUT THE TRACK BITS
41 01056 132404          SUB     1,2,SZR         /SEE IF ITS NEXT SECTOR
42 01057 000454          JMP     CWE7            /NO ,CONTINUE SEARCH
43 01060 020176          LDA     0,BIT15         /GET BIT 15
44 01061 042177          STA     0,@IBUF         /MESS UP FIST WORD IN BUFFER
45 01062 020177          LDA     0,IBUF          /YES ,LOAD AC0 WITH BUFFER ADDRE
46 01063 006111          JSR     @IDOOA          /ISSUE FUNCTION
47 01064 062000          DOB     0,FPY           /GIVE IT TO THE FLOPPY
48 01065 006111          JSR     @IDOOA          /READ BACK THE ADDRESS
49 01066 071400          DIB     2,FPY
50 01067 142414          SUB#    2,0,SZR         /IS IT THE SAME ADDRESS
51 01070 000451          JMP     CWE8            /NO
52 01071 020143          LDA     0,WITFUN        /LOAD AC0 WITH WRITE NEXT SECTOR
53 01072 006077          JSR     @IDRI           /SET IT UP FOR CORRECT DRIVE
54 01073 006111          JSR     @IDOOA          /ISSUE FUNCTION
55 01074 061100          DOAS    0,FPY           /TELL FLOPPY TO WRITE
56 01075 152400          SUB     2,2
57 01076 050126          STA     2,RCOUNT        /CLEAR COUNTER
58 01077 030177          LDA     2,IBUF          /GET BUFFER ADDRESS
59 01100 020120          LDA     0,SECTST        /GET A 7
60 01101 113000          ADD     0,2             /INCREASE BUFFER ADDRESS BY 7
```

```
 0020 FPYDI
01 01102 006111 CWE2:    JSR     @IDOOA          /ISSUE FUNCTION
02 01103 061400          DIB     0,FPY           /READ BUFFER ADDRESS FROM FLOPPY
03 01104 101015          MOV#    0,0,SNR         /IS ADDRESS 0
04 01105 000403          JMP     CWE6            /DIB IS NOT WORKING
05 01106 101414          INC#    0,0,SZR         /IS DIB WORKING
06 01107 000405          JMP     CWE5            /YES CONTINUE TESTING
07 01110 006074 CWE6:    JSR     @IERME          /NO,TELL OPERATOR
08 01111 004720          DIBNG
09 01112 006104          JSR     @IERR           /REPORT DIB NOT WORKING PROPERLY
10 01113 000441          JMP     CWE3+3          /SKKIP TEST COMPLETELY
11 01114 112414 CWE5:    SUB#    0,2,SZR         /HAS BUFFER ADDRESS INCREMENTED
12 01115 000305          JMP     CWE9            /NO KEEP WAITING FOR BUF+7 ADDRE
13 01116 006111          JSR     @IDOOA          /MAKE A MESS OF CHECKWORD
14 01117 060600          DIAC    0,FPY
15 01120 006103          JSR     @IREAD          /GO READ THAT SECTOR 25
16 01121 000025          25                      /THERESHOULD BE A CHECK WORD
17 01122 000000          0                       /ERROR THIS TIME
18 01123 177777          -1                      /ONLY ONE READ
19 01124 024173          LDA     1,BIT12         /GET CHECK WORD ERROR BIT
20 01125 107434          ANDZ#   0,1,SZR         /DID CHECK WORD ERROR SET
21 01126 000423          JMP     CWE3            /YES ,SEE IF ERROR IS SET
22 01127 101234          MOVZR#  0,0,SZR         /NO ,SEE IF ERROR SET ANY WAY
23 01130 000417          JMP     CWE4            /YES,ERROR SET SO CHECK WORD
24                                               /ERROR BIT BAD OR ANOTHER
25                                               /ERROR WAS SET,
26 01131 006104          JSR     @IERR           /GO REPORT ERROR
27 01132 000422          JMP     CWE3+3          /GO DO NEXT TEST
28 01133 010126 CWE7:    ISZ     RCOUNT          /INCREMENT TIMER
29 01134 000715          JMP     CWE1            /CONTINUE SEARCHING FOR SECTOR 0
30 01135 006074          JSR     @IERME          /REPORT CAN'T GET A COMPARE ON S
31                                               /ADDRESS
32 01136 004167          CNFSC
33 01137 006104          JSR     @IERR           /GO TO ERROR ROUTINE
34                                               /CANNOT GET A COMPARE ON SECTOR
35 01140 000414          JMP     CWE3+3          /GO TO CYCLE
36 01141 151415 CWE8:    INC#    2,2,SNR         /DID DIB RESPOND
37 01142 000746          JMP     CWE6            /NO
38 01143 006074          JSR     @IERME          /PRINT ERROR MESSAGE
39 01144 004474          DOBIB
40 01145 006104          JSR     @IERR           /PRINT AC'S
41                                               /AC0=GOOD DATA
42                                               /AC2=BAD DATA
43 01146 000406          JMP     CWE3+3          /CONTINUE
44 01147 006104 CWE4:    JSR     @IERR           /REPORT CHECK WORD ERROR
45                                               /DID NOT SET
46 01150 000404          JMP     CWE3+3          /CONTINUE TESTING
47 01151 101234 CWE3:    MOVZR#  0,0,SZR         /IS ERROR SET?
48 01152 000402          JMP     CWE3+3          /ERROR IS SET ALSO
49 01153 006104          JSR     @IERR           /CHECK WORD ERROR BIT IS
50                                               /SET BUT ERROR BIT IS CLEAR
51 01154 024173          LDA     1,BIT12         /GET THE CHECK WORD BIT
52 01155 030176          LDA     2,BIT15         /GET THE ERROR BIT
53 01156 147000          ADD     2,1             /ADD THEM TOGETHER
54 01157 124000          COM     1,1             /COMPLIMENT TO CLEAR
55 01160 123400          AND     1,0             /THEM FRO STATUS
56 01161 006106          JSR     @IERCK          /SEE IF ANY OTHER ERRORS SET
57                       LOOPS
58 01162 006247          JSR     @ITET
59 01163 006214          JSR     @ICYC?E         /GO TO CYCLE ON TEST ROUTINE
60 01164 102400          SUB     0,0             /CLEAR AC0
```

```
 0021 FPYDI
01 01165 040257          STA     0,ERTS1         /CLEAR ERTS1
```

```
 0022 FPYDI
01
02
03 01166 000402          JMP     .+2
04 01167 000004 TN4:     4
05 01170 020136 ET4:     LDA     0,TSTNM
06 01171 024776          LDA     1,TN4
07 01172 122454          SUBO#   1,0,SZR         /CHECK THE PROGRAM FLOW
08 01173 006256          JSR     @ISEQER
```

```
 0023 FPYDI
01                ;*****************************************************************
02                ;TST#5  <TEST ILLEAGLE FUNCTION AND ERROR BIT>
03                ;*****************************************************************
04
05 01174 010136 TST5:   ISZ     TSTNM           ;ENTERING A NEW TEST
06                ;       TESTS THAT ALL ILLEAGLE FUNCTIONS ARE DETECTED
07                ;       AND THAT DONE GETS SET AFTER THE DETECTION,
08                ;       ALSO VERIFIES THAT ILLEAGLE FUNCTION SETS ERROR,
09                ;
10                ;       IF AN ERROR OCCURS THE ILLEAGLE FUNCTION IS PRINTED
11                ;/////////////////////////////////////////////////////////////////
12
13
14 01175 006276          JSR     @PTET           ;PRINT THE TEST #
15 01176 006213          JSR     @IENT?R         ;GO SETUP TEST
16 01177 000100          100                     ;FOR 100 ITERATIONS
17 01200 006100          JSR     @ISEAK          ;SEAK A NEW TRACK
18 01201 000003          3                       ;TRACK 3
19 01202 000000          0
20 01203 020177          LDA     0,IBUF          ;GET BUFFER ADDRESS
21 01204 006111          JSR     @IDOOA          ;ISSUE FUNCTION
22 01205 062000          DOB     0,FPY           ;GIVE ADDRESS TO FLOPPY
23 01206 006101          JSR     @ISET           ;SETTLE THE DRIVE
24 01207 020140          LDA     0,ILFUN         ;GET THE ILLEAGLE FUNC
25 01210 006077          JSR     @IDRI           ;SET IT UP FOR CORRECT DRIVE
26 01211 006111          JSR     @IDOOA          ;ISSUE IT TO THE FLOPPY
27 01212 061100          DOAS    0,FPY           ;GIVE FLOPPY ILLEAGLE FUNCTION
28 01213 006105          JSR     @ITIME          ;WAIT FOR DONE
29                                               ;POSITION
30 01214 006111          JSR     @IDOOA          ;ISSUE READ STATUS
31 01215 060600          DIAC    0,FPY           ;READ STATUS AND CLEAR
32 01216 024171          LDA     1,BIT10         ;GET ILLEAGLE FUNC BIT
33 01217 107415          AND#    0,1,SNR         ;IS ILL FUN SET
34 01220 000405          JMP     ILE1            ;NO,SEE IF ERROR SET ANYWAY
35 01221 101234          MOVZR#  0,0,SZR         ;IS ERROR SET?
36 01222 000424          JMP     ILE6            ;YES, CONTINUE TESTING
37 01223 006104          JSR     @IERR           ;NO,REPORT ILL FUN BUT
38                                               ;NO ERROR BIT SET
39 01224 000422          JMP     ILE6            ;CONTINUE TESTING
40 01225 101234 ILE1:    MOVZR#  0,0,SZR         ;IS ERROR SET?
41 01226 000403          JMP     ILE2            ;YES,REPORT NO ILL FUN
42 01227 006104          JSR     @IERR           ;REPORT NO ILL FUN,
43                                               ;NO ERROR BIT SET
44 01230 000416          JMP     ILE6            ;CONTINUE TESTING
45 01231 006104 ILE2:    JSR     @IERR           ;REPORT ERROR BIT SET
46                                               ;ILL FUN NOT SET
47 01232 000414          JMP     ILE6            ;CONTINUE TESTING
48 01233 022073 ILE5:    LDA     0,@IITR         ;GET THE ERROR FLAG
49 01234 101005          MOV     0,0,SNR         ;WAS THER AN ERROR
50 01235 000417          JMP     ILE8            ;NO
51 01236 006074          JSR     @IERME          ;YES TELL OPERATOR
52 01237 005135          IFWAS
53 01240 024257          LDA     1,ERTS1         ;SEE IF WE ARE LOOPING
54 01241 125004          MOV     1,1,SZR         ;ARE WE LOOPING?
55 01242 000412          JMP     ILE8            ;YES
56 01243 024140          LDA     1,ILFUN         ;GET ILLEAGLE VALUE
57 01244 006212          JSR     @IZOC?T         ;PRINT THE VALUE
58 01245 000407          JMP     ILE8            ;CONTINUE TESTING
59 01246 124000 ILE6:    COM     1,1             ;COMPLIMENT THE ILLEAGLE FLAG
60 01247 123400          AND     1,0             ;CLEAR THE ILLEAGLE FLAG IF ITS
```

```
 0024 FPYDI
01 01250 101220          MOVZR   0,0             ;CLEAR ERROR
02 01251 101120          MOVZL   0,0             ;BIT
03 01252 006106          JSR     @IEHCK          ;SEE IF ANY OTHER ERRORS
04 01253 000760          JMP     ILE5            ;CONTINUE WITH TEST
05                ILE8:  LOOPS
06 01254 006247          JSR     @ITET
07 01255 006214          JSR     @ICYC?E         ;GO TO CYCLE ON TEST ROUTINE
08 01256 102400          SUB     0,0             ;CLEAR AC0
09 01257 040257          STA     0,ERTS1         ;CLEAR ERTS1
10
11 01260 000402          JMP     .+2
12 01261 000005 TN5:     5
13 01262 020136 ET5:     LDA     0,TSTNM
14 01263 024776          LDA     1,TN5
15 01264 122454          SUBO#   1,0,SZR         ;CHECK THE PROGRAM FLOW
16 01265 006256          JSR     @ISEQER
```

```
0025 FPYDI
01                      /*************************************************************
02                      /TST#6  <TEST BUFFER ADDRESS INCREMENTS PROPERLY>
03                      /*************************************************************
04
05 01266 01#136 TST6:   ISZ     TSTNM           /ENTERING A NEW TEST
06                      /       THIS TEST CHECKS THAT THE BUFFER ADDRESS INCREMENTS
07                      /       TO THE CORRECT ADDRESS AFTER A WRITE.
08                      /
09                      /       IF ANY ERRORS OCCUR GO TO THE BUFFER ADDRESS INC
10                      /       LOGIC.
11                      ////////////////////////////////////////////////////////////
12
13
14 01267 006276         JSR     @PTET           /PRINT THE TEST #
15 01270 006213         JSR     @IENT?R         /SETUP TEST
16 01271 000100         100
17 01272 03#133         LDA     2,IWBUF         /GET WRITE BUFFER ADDRESS
18 01273 05#177         STA     2,IBUF          /STORE IT AT BUFFER POINTER
19 01274 006076         JSR     @IWRSE          /SET THE WRITE SWITCH
20 01275 006102         JSR     @IRITE          /DO THE WRITE
21 01276 000100         100                     /TRACK 100
22 01277 000003         3                       /SECTOR 3
23 01300 006106         JSR     @IERCK          /WAS THERE ANY ERRORS
24 01301 030134         LDA     2,BUFED         /GET END OF BUFFER ADDRESS
25 01302 151400         INC     2,2             /INCREMNET BUFFER ADDRESS
26 01303 151400         INC     2,2             /CORRECT IT BECASE BUFED IS -1
27 01304 000111         JSR     @IDOOA          /GO READ THE BUFFER ADDRESS
28 01305 061400         DIB     0,FPY           /READ THE BUFFER ADDRESS
29 01306 112415         SUB#    0,2,SNR         /DO THEY EQUAL
30 01307 000405         JMP     BADD2           /YES ,TEST INCREMENT ON READ
31 01310 006074         JSR     @IERME          /REPORT BUFFER ADDRESS NOT INC
32                                              /REMENTING PROPERLY
33 01311 004534         BADDI
34 01312 006104         JSR     @IERR           /BUFFER ADDRESS DID NOT INC-
35                                              /REMENT PROPERLY OR THE DIB FUNC
36                                              /IS NOT WORKING
37 01313 000416         JMP     BADD1           /EXIT THE TEST
38 01314 006103 BADD2:  JSR     @IREAD          /GO DO A READ
39 01315 000000         0
40 01316 000000         0
41 01317 177777         -1
42 01320 000111         JSR     @IDOOA          /GET THE ADRESS
43 01321 071400         DIB     2,FPY           /DO A DIB
44 01322 034134         LDA     3,BUFED         /GET BUFFER END ADDRESS
45 01323 175400         INC     3,3             /INCREASE IT BY 1
46 01324 156415         SUB#    2,3,SNR         /DO THEY COMPARE
47 01325 000404         JMP     BADD1           /CONTINUE TESTS
48 01326 006074         JSR     @IERME          /PRINT ADDRESS ERROR
49 01327 004335         BADDR
50 01330 006104         JSR     @IERR           /PRINT THE AC'S
51                                              /AC0=STATUS
52                                              /AC2=BAD ADDRESS
53                                              /AC3=GOOD ADDRESS
54                      BADD1:  LOOPS
55 01331 006247         JSR     @ITET
56 01332 006214         JSR     @ICYC?E         /GO TO CYCLE ON TEST ROUTINE
57 01333 102400         SUB     0,0             /CLEAR AC0
58 01334 040257         STA     0,ERTS1         /CLEAR ERTS1
59
60 01335 000402         JMP     .+2
```

```
0026 FPYDI
01 01336 000006 TN6:    6
02 01337 020136 ET5:    LDA     0,TSTNM
03 01340 024776         LDA     1,TN6
04 01341 122454         SUBO#   1,0,SZR         /CHECK THE PROGRAM FLOW
05 01342 006256         JSR     @ISEQER
```

```
 0027 FPYDI
01                ;*****************************************************************
02                ;TST#7  <TEST SECTOR ERROR AND ERROR BIT>
03                ;*****************************************************************
04
05 01343 010136 TST7:   ISZ     TSTNM           ;ENTERING A NEW TEST
06                ;       THIS TEST DOES A READ NEXT SECTOR FOR SECTOR 0
07                ;       IF NO SECTOR ERROR OCCURS ANOTHER READ NEXT
08                ;       SECTOR FOR SECTOR 0 IS ISSUED,IF SECTOR ERROR
09                ;       DOES NOT SET ON THE SECOND TRY AN ERROR IS REPORTED
10                ;
11                ;       IF SECTOR ERROR WON'T SET CHECK:
12                ;               1) COMMAND REGISTER U100 PINS 5,7,10 FOR
13                ;                  ALL LOWS.
14                ;
15                ;               2) CHECK INPUTS FOR U98 PIN 3 AND 2 FOR
16                ;                  PROPER OPERATION.
17                ;       *** SECTOR ERROR FLIP FLOP WILL GIVE AN ERROR
18                ;           FOR EVERY SECTOR EXCEPT THE ONE BEING READ.
19                ;/////////////////////////////////////////////////////////////////
20
21
22 01344 006276          JSR     @PTET           ;PRINT THE TEST #
23 01345 006213          JSR     @IENT?R         ;SETUP TEST
24 01346 000100          100                     ;FOR 100 ITERATIONS
25 01347 024175          LDA     1,BIT14         ;GET A COUNT OF 2
26 01350 044262          STA     1,SERCT         ;SAVE IN TRY COUNTER
27 01351 006100          JSR     @ISEAK          ;SEAK A TRACK
28 01352 000040          40                      ;TRACK 40
29 01353 000000          0
30 01354 030177 MORE:    LDA     2,IBUF          ;GET BUFFER ADDRESS
31 01355 006111          JSR     @IDOOA          ;GIVE IT TO THE FLOPPY
32 01356 072000          DOB     2,FPY
33 01357 020142          LDA     0,REDFUN        ;GET READ FUNCTION
34 01360 006077          JSR     @IDRI           ;SET IT UP FOR CORRECT DRIVE
35 01361 006111          JSR     @IDOOA          ;TELL FLOPPY TO READ
36 01362 061100          DOAS    0,FPY
37 01363 006105          JSR     @ITIME          ;WAIT FOR DONE
38 01364 006111          JSR     @IDOOA          ;READ THE STATUS
39 01365 060600          DIAC    0,FPY           ;AND CLEAR IT
40 01366 024172          LDA     1,BIT11         ;GET SECTOR ERROR BIT
41 01367 123434          ANDZ#   1,0,SZR         ;IS SECTOR ERROR SET?
42 01370 000410          JMP     SECT1           ;YES SEE IF ERROR SET
43 01371 014262          DSZ     SERCT           ;DECREMENT TRY COUNTER
44 01372 000762          JMP     MORE            ;FIRST TRY DO IT AGAIN
45 01373 006074          JSR     @IERME          ;REPORT SECTOR ERROR WON'T SET
46 01374 004235          SECWS
47 01375 006104          JSR     @IERR           ;SECTOR ERROR IS NOT SETTING
48 01376 006106          JSR     @IERCK          ;SEE IF ANY OTHER ERROR SET
49 01377 000404          JMP     .+4             ;CONTINUE TESTING
50 01400 101214 SECT1:   MOVN#   0,0,SZR         ;IS ERROR SET?
51 01401 000402          JMP     .+2             ;YES CONTINUE TESTING
52 01402 006104          JSR     @IERR           ;SECTOR ERROR SET BUT
53                                               ;ERROR BIT IS CLEAR
54 01403 030176          LDA     2,BIT15         ;GET THE ERROR BIT
55 01404 147000          ADD     2,1             ;ADD IT TO THE SECTOR ERROR BIT
56 01405 124000          COM     1,1             ;COMPLIMENT TO CLEAR THE
57 01406 123400          AND     1,0             ;THE BITS
58 01407 006106          JSR     @IERCK          ;SEE IF ANY OTHER ERRORS
59                       LOOPS
60 0141# 006247          JSR     @ITET
```

```
 0028 FPYDI
01 01411 006214          JSR     @ICYC?E         ;GO TO CYCLE ON TEST ROUTINE
02 01412 102400          SUB     0,0             ;CLEAR AC0
03 01413 040257          STA     0,ERTS1         ;CLEAR ERTS1
04
05 01414 000402          JMP     .+2
06 01415 000007 TN7:     7
07 01416 020136 ET7:     LDA     0,TSTNM
08 01417 024776          LDA     1,TN7
09 0142# 122454          SUB0#   1,0,SZR         ;CHECK THE PROGRAM FLOW
10 01421 006256          JSR     @ISEQER
```

```
0029 FPYDI
01                ;*****************************************************************
02                ;TST#10 <TEST THE MASK FOR THE FLOPPY>
03                ;*****************************************************************
04
05 01422 010136 TST10:  ISZ     TSTNM           ;ENTERING A NEW TEST
06
07 01423 006276         JSR     @PTET           ;PRINT THE TEST #
08 01424 006213         JSR     @IENT?R         ;SETUP THE TEST
09 01425 000005         5
10 01426 006100         JSR     @ISEAK          ;SEEK TRACK 53
11 01427 000053         53
12 01430 000000         0
13 01431 020177         LDA     0,IBUF          ;GET A BUFFER ADDRESS
14 01432 006111         JSR     @IDOOA          ;GIVE IT TO THE FLOPPY
15 01433 062000         DOB     0,FPY
16 01434 024420         LDA     1,INSER         ;GET THE SERVICE ADDRESS
17 01435 044001         STA     1,1             ;STORE AT INTERUPT POINTER
18 01436 024153         LDA     1,ONEZ          ;GET ALL 1'S
19 01437 066077         MSKO    1               ;MASK ALL DEVICES
20 01440 060177         INTEN                   ;ENABLE INTERUPTS
21 01441 020144         LDA     0,RDPA          ;GET READ PREAMBLE
22 01442 006077         JSR     @IORI           ;SET IT UP FOR CORRECT DRIVE
23 01443 006111         JSR     @IDOOA          ;ISSUE IT TO FLOPPY
24 01444 061100         DOAS    0,FPY
25 01445 006105         JSR     @ITIME          ;WAIT FOR DONE
26 01446 060277         INTDS                   ;DISABLE INTERUPTS
27 01447 024104         LDA     1,IERR          ;GET ERROR ADDRESS
28 01450 044001         STA     1,1             ;IF INTERUPTED REPORT ERROR
29 01451 024152         LDA     1,BLANK         ;GET 0'S
30 01452 066077         MSKO    1               ;CLEAR THE MASKS
31 01453 000412         JMP     MAK1            ;CONTINUE TESTING
32 01454 001455 INSER:  MAK2
33 01455 063710 MAK2:   SKPDZ   TTI             ;SEE IF TTY INTERUPTED
34 01456 000404         JMP     MAK3            ;YES
35 01457 006074         JSR     @IERME          ;REPORT MASK DID NOT WORK
36 01460 005242         MKNG
37 01461 000403         JMP     MAK4            ;CONTINUE TESTING
38 01462 006074 MAK3:   JSR     @IERME          ;REPORT TTY INTERUPTED
39 01463 005214         MKTT
40 01464 006104 MAK4:   JSR     @IERR           ;REPORT ERROR
41              MAK1:   LOOPS
42 01465 006247         JSR     @ITET
43 01466 006214         JSR     @ICYC?E         ;GO TO CYCLE ON TEST ROUTINE
44 01467 102400         SUB     0,0             ;CLEAR AC0
45 01470 040257         STA     0,ERTS1         ;CLEAR ERTS1
46
47 01471 000402         JMP     .+2
48 01472 000010 TN10:   10
49 01473 020136 ET10:   LDA     0,TSTNM
50 01474 024776         LDA     1,TN10
51 01475 122454         SUBO#   1,0,SZR         ;CHECK THE PROGRAM FLOW
52 01476 006256         JSR     @ISEQER
```

```
0030 FPYDI
01                ;*****************************************************************
02                ;TST#11 <WRITE AND READ ALL PATTERNS TEST>
03                ;*****************************************************************
04
05 01477 010136 TST11:  ISZ     TSTNM           ;ENTERING A NEW TEST
06                ;       WRITES,READS AND COMPARES ALL TEST PATTERNS ON
07                ;       ALL TRACKS AND SECTORS.
08                ;
09                ;       IF ERRORS OCCUR CHECK:
10                ;               1) POSSIBLE BAD MEDIA
11                ;               2) CHECK Y820 AT U60 PIN 13 TO
12                ;                  VERIFY IT OCCURS AT THE RIGHT TIME
13                ;               3) HEAD IS POSSIBLY OUT OF ALIGNMENT
14                ;
15                ;       IF WRITE FAULT ERRORS CHECK:
16                ;               1) U79 PIN 6 FOR 4MICRO SEC CELL CLOCK
17                ;               2) CHECK WRDATA AT U18 PIN 3 AND 5
18                ;               3) CHECK WRGATE AT U15 PIN 3 AND 5
19                ;               4) POSSIBLE DRIVE CONTROLLER LOGIC
20                ;                  PROBLEM.
21                ;/////////////////////////////////////////////////////////////////
22
23
24 01500 006276         JSR     @PTET           ;PRINT THE TEST #
25 01501 006213         JSR     @IENT?R         ;SETUP TEST
26 01502 000001         1                       ;1 ITERATIONS
27 01503 006131         JSR     @IWCPAT         ;DO WRITE READ AND COMPARE
28                      LOOPS
29 01504 006247         JSR     @ITET
30 01505 006214         JSR     @ICYC?E         ;GO TO CYCLE ON TEST ROUTINE
31 01506 102400         SUB     0,0             ;CLEAR AC0
32 01507 040257         STA     0,ERTS1         ;CLEAR ERTS1
```

```
0031 FPYDI
01
02
03 01510 000402          JMP     .+2
04 01511 0P0011 TN11:    11
05 01512 020136 ET11:    LDA     0,TSTNM
06 01513 024776          LDA     1,TN11
07 01514 122454          SUBO#   1,0,SZR         ;CHECK THE PROGRAM FLOW
08 01515 006256          JSR     @ISEQER
```

```
0032 FPYDI
01                ;*****************************************************************
02                ;TST#12 <TEST HEAD LOAD AFTER DRIVE CHANGE>
03                ;*****************************************************************
04
05 01516 010136 TST12:   ISZ     TSTNM          ;ENTERING A NEW TEST
06
07 01517 006276          JSR     @PTET          ;PRINT THE TEST #
08 01520 006213          JSR     @IENT?R        ;SETUP THE TEST
09 01521 000100          100                    ;FOR 100 ITERATION
10 01522 024172          LDA     1,BIT11        ;GET A 20
11 01523 030173          LDA     2,BIT12        ;GET A 4
12 01524 147000          ADD     2,1            ;MAKE A 24 OCTAL=20 DECIMAL
13 01525 030176          LDA     2,BIT15        ;GET A 1 TO MAKE 31 OCTAL
14 01526 147000          ADD     2,1            ;CREATE THE 31
15 01527 044126          STA     1,RCOUNT       ;SAVE THE COUNT
16 01530 024116          LDA     1,FLAGS        ;GET THE FLAGS
17 01531 030157          LDA     2,BIT0         ;GET THE DRIVE BIT
18 01532 020234          LDA     0,NUDEV        ;GET THE # OF DEVICES
19 01533 101005          MOV     0,0,SNR        ;DO WE HAVE 2 DRIVES
20 01534 000446          JMP     DCH4           ;NO SKIP THE TEST
21 01535 020141          LDA     0,SETTLE       ;GET A SETTLE FUNCTION
22 01536 147414          AND#    2,1,SZR        ;WHAT DRIVE ARE WE TO CHANGE TO
23 01537 000402          JMP     DCH1           ;SET IT FOR DRIVE 0
24 01540 143000          ADD     2,0            ;SET FOR DRIVE 1
25 01541 006111 DCH1:    JSR     @IDOOA         ;ISSUE THE SETTLE
26 01542 061100          DOAS    0,FPY
27 01543 006105          JSR     @ITIME         ;WAIT FOR DONE
28 01544 006111          JSR     @IDOOA         ;GET THE STATUS
29 01545 070400          DIA     2,FPY
30 01546 151132          MOVZL#  2,2,SZC        ;IS THE FLOPPY READY
31 01547 000412          JMP     DCH2           ;FLOPPY WENT NOT READY
32 01550 024161          LDA     1,BIT2         ;GET THE HEAD LOAD BIT
33 01551 133404          AND     1,2,SZR        ;IS THE HEAD LOADED
34 01552 000413          JMP     DCH3           ;YES SEE HOW LONG IT TOOK
35 01553 014126          DSZ     RCOUNT         ;NO DECREMENT THE COUNT
36 01554 000765          JMP     DCH1           ;CONTINUE SETTELING
37 01555 006074          JSR     @IERME         ;REPORT HEADS DID NOT LOAD AFTER
38 01556 004037          HDL20                  ;20 SETTLE TIMES
39 01557 006104          JSR     @IERR          ;PRINT THE AC'S
40 01560 000422          JMP     DCH4           ;CONTINUE TESTS
41 01561 006074 DCH2:    JSR     @IERME         ;REPORT THE FLOPPY WENT
42 01562 004267          FPYRD                  ;NOT READY AFTER A DRIVE
43                                              ;CHANGE
44 01563 006104          JSR     @IERR          ;PRINT THE AC'S
45 01564 000416          JMP     DCH4           ;CONTINUE TESTS
46 01565 030173 DCH3:    LDA     2,BIT12        ;GET A 18 OCTAL=8 DECIMAL
47 01566 024122          LDA     1,C23          ;GET A23 OCTAL
48 01567 146400          SUB     2,1            ;MAKE A 13 OCTAL
49 01570 030126          LDA     2,RCOUNT       ;GET THE REMANING COUNT
50 01571 132404          SUB     1,2,SZR        ;SUBTRACT THE 4
51 01572 000402          JMP     DCH5           ;SEE IF RESULT IS POSITIVE
52 01573 000407          JMP     DCH4           ;DRIVE CHANGE OK CONT TEST
53 01574 151133 DCH5:    MOVZL#  2,2,SNC        ;IS IT A NEG VALUE
54 01575 000402          JMP     DCH6           ;NO DRIVE CHANGE IN ERROR
55 01576 000404          JMP     DCH4           ;CONTINUE TESTING
56 01577 006074 DCH6:    JSR     @IERME         ;REPORT HEAD LOAD SET TO SOON
57 01600 004063          HLSDC                  ;ON A DRIVE CHANGE
58 01601 006104          JSR     @IERR          ;PRINT THE AC'S
59 01602 006101 DCH4:    JSR     @ISET          ;SETTLE THE DRIVE BEING TESTED
60                       LOOPS
```

```
 0033 FPYDI
01 01603 006247          JSR     #ITET
02 01604 006214          JSR     #ICYC?E          ;GO TO CYCLE ON TEST ROUTINE
03 01605 102400          SUB     0,0              ;CLEAR AC0
04 01606 040257          STA     0,ERTS1          ;CLEAR ERTS1
05
06 01607 000402          JMP     .+2
07 01610 000012 TN12:    12
08 01611 020136 ET12:    LDA     0,TSTNM
09 01612 024776          LDA     1,TN12
10 01613 122454          SUBO#   1,0,SZR          ;CHECK THE PROGRAM FLOW
11 01614 006256          JSR     #ISEQER
```

```
 0034 FPYDI
01                       ;*****************************************************************
02                       ;TST#13 <TEST THE HEAD SETTLE TIME>
03                       ;*****************************************************************
04
05 01615 010136 TST13:   ISZ     TSTNM            ;ENTERING A NEW TEST
06
07 01616 006276          JSR     #PTET            ;PRINT THE TEST #
08 01617 000213          JSR     #IENT?R          ;SET UP THE TEST
09 01620 000001          1                        ;FOR 1 ITERATION
10 01621 022073          LDA     0,#IITR          ;SEE IF WE HAD AN ERROR
11 01622 101004          MOV     0,0,SZR          ;ANY ERRORS?
12 01623 000403          JMP     HESE4            ;YES ,NOT 1ST TIME THROUGH
13 01624 040420          STA     0,HTRK           ;CLEAR THE TRACK
14 01625 040420          STA     0,HSEC           ;CLEAR SECTOR ADDRESS
15 01626 102400 HESE4:   SUB     0,0              ;MAKE SURE AC0=0
16 01627 040126          STA     0,RCOUNT         ;CLEAR THE COUNT
17 01630 000111 HESE2:   JSR     #IOOOA           ;GET THE STATUS
18 01631 006000          DIAC    0,FPY            ;READ THE STATUS
19 01632 024161          LDA     1,BIT2           ;GET THE HEAD LOAD BIT
20 01633 123415          AND#    1,0,SNR          ;IS THE HEAD LOADED
21 01634 000407          JMP     HESE1            ;NO GO READ THE SECTOR
22 01635 010126          ISZ     RCOUNT           ;INC THE COUNT
23 01636 000772          JMP     HESE2            ;CONT WAITING FOR HEADS
24 01637 006074          JSR     #IERME           ;PRINT THE HEADS WON'T
25 01640 004145          HDLDF                    ;UNLOAD
26 01641 006104          JSR     #IERR            ;PRINT THE AC'S
27 01642 000430          JMP     HESE3            ;CONTINUE TESTS
28 01643 006103 HESE1:   JSR     #IREAD           ;GO READ THE FLOPPY
29 01644 000000 HTRK:    0                        ;TRACK ADDRESS
30 01645 000000 HSEC:    0                        ;SECTOR ADDRESS
31 01646 177777          -1                       ;1 SECTOR
32 01647 006106          JSR     #IERCK           ;SEE IF ANY ERRORS
33 01650 006273          JSR     #ICPET           ;COMPARE THE INFO
34 01651 026073          LDA     1,#IITR          ;GET THE ERROR COUNT
35 01652 125004          MOV     1,1,SZR          ;WAS THERE ANY ERRORS
36 01653 000417          JMP     HESE3            ;YES DONT UPDATE JUST LOOP
37 01654 010771          ISZ     HSEC             ;INCREMENT SECTOR COUNT
38 01655 024770          LDA     1,HSEC           ;GET NEW SECTOR
39 01656 030120          LDA     2,SECTST         ;GET SECTOR TEST VALUE
40 01657 147414          AND#    2,1,SZR          ;DO WE HAVE A SECTOR OVERFLOW
41 01660 000746          JMP     HESE4            ;NO CONT
42 01661 152400          SUB     2,2              ;CLEAR THE AC
43 01662 050763          STA     2,HSEC           ;RESTORE TO SECTOR 0
44 01663 010761          ISZ     HTRK             ;CRETE NEW TRACK VALUE
45 01664 024760          LDA     1,HTRK           ;GET NEW TRACK
46 01665 030147          LDA     2,C77            ;GET MAX TRACK VALUE
47 01666 132414          SUB#    1,2,SZR          ;ARE WE DONE WITH ALL TRACKS
48 01667 000737          JMP     HESE4            ;NO CONT
49 01670 152400          SUB     2,2              ;YES CLEAR THE AC
50 01671 050753          STA     2,HTRK           ;CLEAR TRAK VALUE
51                 HESE3: LOOPS
52 01672 006247          JSR     #ITET
53 01673 006214          JSR     #ICYC?E          ;GO TO CYCLE ON TEST ROUTINE
54 01674 102400          SUB     0,0              ;CLEAR AC0
55 01675 040257          STA     0,ERTS1          ;CLEAR ERTS1
56
57 01676 000402          JMP     .+2
58 01677 000013 TN13:    13
59 01700 020136 ET13:    LDA     0,TSTNM
60 01701 024776          LDA     1,TN13
```

```
 0035 FPYDI
01 01702 122454          SUB0#   1,0,SZR         ;CHECK THE PROGRAM FLOW
02 01703 006256          JSR     @ISEQER
```

```
 0036 FPYDI
01                ;*****************************************************************
02                ;TST#14 <TEST BOTH FLOPPYS OF A DUAL DRIVE>
03                ;*****************************************************************
04
05 01704 010136 TST14:  ISZ     TSTNM           ;ENTERING A NEW TEST
06
07 01705 006213         JSR     @IENT?R         ;SETUP TEST
08 01706 000001         1
09 01707 024203         LDA     1,PASS          ;FIND OUT WHAT PASS
10 01710 125005         MOV     1,1,SNR         ;IS IT PASS 1
11 01711 000455         JMP     DU2             ;YES
12 01712 024234         LDA     1,NUDEV         ;FIND OUT HOW MANY DRIVES
13 01713 125005         MOV     1,1,SNR         ;ARE THERE MORE THAN 1
14 01714 000452         JMP     DU2             ;ONLY ONE DRIVE
15 01715 126400         SUB     1,1             ;CLEAR AC1
16 01716 044412         STA     1,DUTRK         ;CLEAR TRACK
17 01717 044412         STA     1,DUSEC         ;CLEAR SECTOR
18 01720 024132         LDA     1,IRBUF         ;GET READ BUFFER ADDRESS
19 01721 044177         STA     1,IBUF          ;SAVE THE ADDRESS
20 01722 126400         SUB     1,1             ;CLEAR AC1
21 01723 044116         STA     1,FLAGS         ;START WITH DRIVE 0
22 01724 006276         JSR     @PTET           ;PRINT THE TEST #
23 01725 006213         JSR     @IENT?R         ;SETUP THE TEST TO LOOP HERE
24 01726 000001         1
25 01727 006103 DU5:    JSR     @IREAD          ;GO READ
26 01730 000000 DUTRK:  0
27 01731 000000 DUSEC:  0
28 01732 177777         -1
29 01733 006106         JSR     @IERCK          ;SEE IF ANY ERRORS
30 01734 006273         JSR     @ICPET          ;GO COMPARE THE PATTREN
31 01735 026073         LDA     1,@IITR         ;WAS THERE AN ERROR
32 01736 125004         MOV     1,1,SZR         ;IN THE COMPARE
33 01737 000427         JMP     DU2             ;YES SO DON'T UPDATE SECTOR
34 01740 024116         LDA     1,FLAGS         ;GET THE FLAGS
35 01741 125102         MOVL    1,1,SZC         ;WAS IT DRIVE 1
36 01742 000404         JMP     DU3             ;YES
37 01743 125240         MOVOR   1,1             ;SET IT UP FOR DRIVE 1
38 01744 044116         STA     1,FLAGS         ;REPLACE THE FLAGS
39 01745 000762         JMP     DU5             ;DO THE SAME FOR DRIVE 1
40 01746 125220 DU3:    MOVZR   1,1             ;SET FOR DRIVE 0
41 01747 044116         STA     1,FLAGS         ;SAVE THE NEW FLAGS
42 01750 010761 DU4:    ISZ     DUSEC           ;UPDATE THE SECTOR COUNT
43 01751 024760         LDA     1,DUSEC         ;SEE IF WE NEED A NEW TRACK
44 01752 030120         LDA     2,SECTST                ;GET OVER FLOW BIT
45 01753 147414         AND#    2,1,SZR         ;HAVE WE OVER FLOWED
46 01754 000753         JMP     DU5             ;NO READ THE SECTOR
47 01755 126400         SUB     1,1             ;CLEAR  AC1
48 01756 044753         STA     1,DUSEC         ;MAKE IT SECTOR 0
49 01757 010751         ISZ     DUTRK           ;INCREMENT THE TRACK
50 01760 024750         LDA     1,DUTRK         ;FIND OUT IF ALL DONE
51 01761 030147         LDA     2,C77           ;GET HIGHEST TRACK
52 01762 132414         SUB#    1,2,SZR         ;ARE WE DONE
53 01763 000744         JMP     DU5             ;GO READ
54 01764 126400         SUB     1,1             ;CLEAR THE AC
55 01765 044743         STA     1,DUTRK         ;CLEAR TRACK VALUE
56                DU2:  LOOPS
57 01766 006247         JSR     @ITET
58 01767 006214         JSR     @ICYC?E         ;GO TO CYCLE ON TEST ROUTINE
59 01770 102400         SUB     0,0             ;CLEAR AC0
60 01771 040257         STA     0,ERTS1         ;CLEAR ERTS1
```

```
 0037 FPYDI
01
02 01772 000402          JMP     .+2
03 01773 000014 TN14:    14
04 01774 020136 ET14:    LDA     0,TSTNM
05 01775 024776          LDA     1,TN14
06 01776 122454          SUBO#   1,0,SZR         ;CHECK THE PROGRAM FLOW
07 01777 000256          JSR     @ISEQER
```

```
 0038 FPYDI
01                ;*****************************************************************
02                ;TST#15 <THIS IS THE PROGRAM LOAD TEST>
03                ;*****************************************************************
04
05 02000 010136 TST15:   ISZ     TSTNM           ;ENTERING A NEW TEST
06
07 02001 000276          JSR     @PTET           ;PRINT THE TEST #
08 02002 000213          JSR     @IENT?R         ;SETUP THE TEST
09 02003 000005          5                       ;FOR 5 ITERATIONS
10 02004 020170          LDA     0,BIT9          ;SETUP THE LOAD FOR 100 WORDS
11 02005 040126          STA     0,RCOUNT
12 02006 020301          LDA     0,IPLBG         ;GET POINTER TO PROGRAM LOAD PRO
13 02007 040021          STA     0,CBUFP         ;SAVE AT AUTO ONC LOC
14 02010 063710          SKPDZ   TTI             ;HAS A KEY BEEN HIT
15 02011 000206          JSR     @IINP?          ;YES,SERVICE IT
16 02012 062677          IORST                   ;REST THE SYSTEM
17 02013 061277          DOAC    0,CPU           ;REST MICRO NOVA
18 02014 000111          JSR     @IDOOA          ;ISUE A START PULSE
19 02015 060100          NIOS    FPY             ;TO THE FLOPPY
20 02016 000111 LPLP1:   JSR     @IDOOA          ;READ THE FIRST BYTE
21 02017 070500          DIAS    2,FPY
22 02020 000105          JSR     @ITIME          ;WAIT FOR FLOPPY DONE
23 02021 151300          MOVS    2,2             ;SWAP THE BYTE
24 02022 050302          STA     2,LOWRD         ;SAVE IT
25 02023 000111          JSR     @IDOOA          ;GET THE SECOND GYTE
26 02024 070500          DIAS    2,FPY
27 02025 034302          LDA     3,LOWRD         ;GET BYTE 1
28 02026 173000          ADD     3,2             ;MAKE THE WORD
29 02027 020021          LDA     1,@CBUFP        ;GET FIRST WORD FROM PROG
30 02030 146415          SUB#    2,1,SNR         ;DO THEY COMPARE
31 02031 000405          JMP     LPLP2           ;YES THEY COMPARE
32 02032 000074          JSR     @IERME          ;TELL OPERATOR PRO LOAD NG
33 02033 004252          PLNG
34 02034 000104          JSR     @IERR           ;AC1=GOOD  AC2=BAD
35 02035 000511          JMP     LPLP3           ;EXIT TEST
36 02036 014126 LPLP2:   DSZ     RCOUNT          ;DECREMENT THE COUNT
37 02037 000402          JMP     LPLP4           ;CONTINUE THE TEST
38 02040 000506          JMP     LPLP3           ;EXIT THE TEST
39 02041 152400 LPLP4:   SUB     2,2             ;CLEAR AC2
40 02042 050302          STA     2,LOWRD         ;CLEAR LOAD WORD
41 02043 000753          JMP     LPLP1           ;CONTINUE TEST
42
43                ;THE LOAD PROGRAM FOLLOWS
44 02044 000000 PLEGN:   0
45
46 02045 000007 7
47 02046 177700 177700
48 02047 000077 77
49 02050 177731 177731
50 02051 000131 131
51 02052 020014 20014
52 02053 024774 24774
53 02054 107400 107400
54 02055 044470 44470
55 02056 124000 124000
56 02057 010451 10451
57 02060 010451 10451
58 02061 010452 10452
59 02062 010453 10453
60 02063 125404 125404
```

```
 0039 FPYDI
01 02064 000773 773
02 02065 020763 20763
03 02066 030763 30763
04 02067 025000 25000
05 02070 155300 155300
06 02071 175700 175700
07 02072 045400 45400
08 02073 151400 151400
09 02074 101404 101404
10 02075 000772 772
11 02076 001733 1733
12 02077 061277 61277
13 02100 062677 62677
14 02101 004425 4425
15 02102 000404 404
16 02103 000776 776
17 02104 102520 102520
18 02105 004421 4421
19 02106 125113 125113
20 02107 000775 775
21 02110 102400 102400
22 02111 004416 4416
23 02112 127113 127113
24 02113 000776 776
25 02114 020430 20430
26 02115 004412 4412
27 02116 142034 142034
28 02117 000776 776
29 02120 101120 101120
30 02121 004406 4406
31 02122 000402 402
32 02123 000754 754
33 02124 020421 20421
34 02125 000377 377
35 02126 152221 152221
36 02127 152000 152000
37 02130 061077 61077
38 02131 063577 63577
39 02132 000777 777
40 02133 064377 64377
41 02134 147400 147400
42 02135 072577 72577
43 02136 151220 151220
44 02137 151220 151220
45 02140 125213 125213
46 02141 125122 125122
47 02142 001401 1401
48 02143 001400 1400
49 02144 000010 10
50 02145 000104 104
51
52                LPLP3:  LOOPS
53 02146 006247           JSR     @ITET
54 02147 006214           JSR     @ICYC?E         ;GO TO CYCLE ON TEST ROUTINE
55 02150 102400           SUB     0,0             ;CLEAR AC0
56 02151 040257           STA     0,ERTS1         ;CLEAR ERTS1
57
58 02152 000402           JMP     .+2
59 02153 000015 TN15:     15
60 02154 020136 ET15:     LDA     0,TSTNM
```

```
 0040 FPYDI
01 02155 024776           LDA     1,TN15
02 02156 122454           SUB0#   1,0,SZR         ;CHECK THE PROGRAM FLOW
03 02157 006256           JSR     @ISEQER
```

```
 0041 FPYDI
01                  /*****************************************************************
02                  /TST#16 <THIS IS THE END OF THE TESTS>
03                  /*****************************************************************
04
05 02160 010136 TST16:  ISZ     TSTNM           /ENTERING A NEW TEST
06
07 02161 102400 ENOFS:  SUB     0,0             /CLEAR AC0
08 02162 040300         STA     0,LPNM          /CLEAR LOOP #
09 02163 040257         STA     0,ERTS1         /CLEAR ERROR COUNTER
10 02164 040235         STA     0,COENT         /CLEAR COMMAND ENTRY FLAG
11 02165 040136         STA     0,TSTNM         /CLEAR TEST NUMBER
12 02166 030272         LDA     2,DRI1          /ARE WE ONLY TESTING
13 02167 151005         MOV     2,2,SNR         /DRIVE 1
14 02170 000404         JMP     ENOF1           /NO
15 02171 101240         MOVOR   0,0             /SET IT UP FOR DRIVE 1
16 02172 040116         STA     0,FLAGS         /CORRECT THE FLAGS
17 02173 102400         SUB     0,0             /LEAVE AC0=0
18 02174 030240 ENOF1:  LDA     2,PASST         /GET PASS NUMBER
19 02175 151400         INC     2,2             /INCREMENT IT
20 02176 050240         STA     2,PASST         /REPLACE WITH NEW VALUE
21 02177 024176         LDA     1,BIT15         /GET ODD PASS BIT
22 02200 133435         ANDZ#   1,2,SNR         /IS IT AN ODD PASS
23 02201 000407         JMP     ENOFT           /SETUP FOR DRIVE 0
24 02202 101240         MOVOR   0,0             /SET DRIVE 1 BIT
25 02203 040116         STA     0,FLAGS         /REPLACE NEW FLAGS
26 02204 034234         LDA     3,NUDEV         /GET THE NUMBER OF DRIVES
27 02205 175005         MOV     3,3,SNR         /ARE THERE 2 DRIVES
28 02206 000753         JMP     ENOFS           /NO GO TEST DRIVE ONE AGAIN
29 02207 002271         JMP     @IBGN1          /CONTINUE TESTING
30 02210 040240 ENOFT:  STA     0,PASST         /CLEAR PASS COUNTER
31 02211 010203         ISZ     PASS            /INCREMENT PROGRAM PASS COUNTER
32 02212 024272         LDA     1,DRI1          /GET DRIVE FLAG
33 02213 125004         MOV     1,1,SZR         /DO WE WANT TO DO ONLY
34                                              /DRIVE 1
35 02214 000402         JMP     ENOF2           /YES
36 02215 040116         STA     0,FLAGS         /REPLACE NEW FLAGS
37 02216 024205 ENOF2:  LDA     1,SWREG         /GET THE SWITCH REGISTER
38 02217 034163         LDA     2,BIT4          /GET SWITCH 4
39 02220 147414         ANDO#   2,1,SZR         /DON'T PRINT IF SWITCH 4 SET
40 02221 002271         JMP     @IBGN1          /CONTINUE TESTS
41 02222 006110         JSR     @IMESS          /TYPE END OF PASS MESSAGE
42 02223 005051         ENOFP
43 02224 024203         LDA     1,PASS          /GET PASS #
44 02225 000212         JSR     @IZOC?T         /PRINT IT IN OCTAL
45 02226 002271         JMP     @IBGN1          /CONTINUE TESTING
46
47
48                  /*****************************************************************
49                  /               THIS IS THE COMMAND STRING INTERPRETER
50                  /
51                  /       INPUT SHOULD BE TYPED AS
52                  /               COMMAND,UNIT#,PATTERN,TRACK#,SECTOR#(CR)
53                  /
54                  /EXAMPLE        SEEK,1,0,1,7
55                  /               SEEK TRACK 1 OF UNIT 1 ,THE SECTOR AND PATTERN A
56                  /               ARE IGNORED BUT MUST BE TYPED ANYWAY
57                  /
58                  /               READ,0,0,1,7
59                  /               READ TRACK 1 SECTOR 7 OF UNIT 0
60                  /               WITH A PATTERN OF ALL ZEROS
```

```
 0042 FPYDI
01                  /
02                  /               WRITE,1,177777,45,2
03                  /               WRITE TRACK 45 SECTOR 2 OF UNIT 1 WITH A
04                  /               PATTERN OF ALL ONES
05                  /       IN THE CASE OF A READ THE PATERN ENTERED WILL BE WRITTEN
06                  /IT IS READ THIS WAY THE PATERN SHOULD BE THERE
07                  /*****************************************************************
08
09 02227 006110 COMST:  JSR     @IMESS          /ASK FOR COMMAND STRING
10 02230 003775         ENCMS
11 02231 152400         SUB     2,2             /CLEAR AC2
12 02232 006130 NEXT:   JSR     @ITIN?O         /GET THE COMMAND
13 02233 000404         JMP     ASK             /GO FIND THE COMMAND
14 02234 151014         MOV#    2,2,SZR         /HAS HE GIVEN US A COMMAND
15 02235 000422         JMP     UNIT0           /YES SO GET UNIT #
16 02236 000771         JMP     COMST           /NO SO ASK AGAIN
17 02237 151014 ASK:    MOV#    2,2,SZR         /IS IT FIRST LETTER
18 02240 000772         JMP     NEXT            /NO DISREGARD THE REST
19 02241 151400         INC     2,2             /INCREMENT AC2
20 02242 034112         LDA     3,ICOMT         /GET TABLE ADDRESS
21 02243 050020         STA     3,BUFPT         /SAVE IT FOR LOADING
22 02244 036020 SERCH:  LDA     3,@BUFPT                /GET COMMAND FRO TABLE
23 02245 162415         SUB#    3,0,SNR         /HAVE WE FOUND A COMMAND
24 02246 000404         JMP     FIX             /YES SO LOAD IT IN COMMAND
25                                              /EXECUTER
26 02247 175014         MOV#    3,3,SZR         /ARE WE AT THE END OF THE TABLE
27 02250 000774         JMP     SERCH           /NO CONTINUE SERCH
28 02251 000756         JMP     COMST           /YES, INVALID COMMAND TYPED
29 02252 036020 FIX:    LDA     3,@BUFPT        /GET THE FUNCTION
30 02253 024113         LDA     1,ICOMY         /GET ADDRESS OF STORAGE
31 02254 044021         STA     1,CBUFP         /SAVE TO POINT
32 02255 056021         STA     3,@CBUFP        /STORE THE FUNCTION
33 02256 000754         JMP     NEXT            /GET NEXT CHARACTER
34 02257 006130 UNIT0:  JSR     @ITIN?O         /GET THE UNIT #
35 02260 000747         JMP     COMST           /ALPHA CHARC TYPED
36 02261 006253         JSR     @IGOOD          /SEE IF CORRECT TERMINATOR
37 02262 125004         MOV     1,1,SZR         /IS IT UNIT 0?
38 02263 000403         JMP     UNIT1           /SET IT UP FOR UNIT 1
39 02264 044116         STA     1,FLAGS         /CORRECT THE FLAGS
40 02265 000403         JMP     PAEX            /GO EXECEPT PATTERN
41 02266 024157 UNIT1:  LDA     1,BIT0          /GET UNIT 1 BIT
42 02267 044116         STA     1,FLAGS         /CORRECT THE FLAGS
43 02270 006130 PAEX:   JSR     @ITIN?O         /GET THE PATTERN
44 02271 002252         JMP     @ICOMS          /NO ALPHA ACCEPTED
45 02272 006253         JSR     @IGOOD          /SEE IF GOOD TERMINATOR
46 02273 044263         STA     1,WPAT          /STORE PATTERN AT WHAT PATTERN
47 02274 006130         JSR     @ITIN?O         /GET THE TRACK #
48 02275 002252         JMP     @ICOMS          /NO ALPHA ACCEPTED
49 02276 006254         JSR     @IGOOS          /SEE IF , OR START COMMAND
50 02277 000405         JMP     NOEX            /ACCEPT TRACK BUT NO EXECUTION Y
51 02300 034256         LDA     3,ICOM1         /GET POINTER TO TRACK STORAGE
52 02301 054020         STA     3,BUFPT         /STORE SO I CAN LOAD
53 02302 046020         STA     1,@BUFPT                /STORE TRACK ADDRESS
54 02303 000411         JMP     COMBG           /GO EXECUTE
55 02304 034256 NOEX:   LDA     3,ICOM1         /GET POINTER TO TRAK STORAGE
56 02305 054020         STA     3,BUFPT         /STORE IT AT POINTER
57 02306 046020         STA     1,@BUFPT        /STORE TRACK
58 02307 006130         JSR     @ITIN?O         /GET SECTOR ADDRESS
59 02310 002252         JMP     @ICOMS          /TYPED AN ALPHA CHARACTER
60 02311 034251         LDA     3,ICOM2         /GET POINTER TO SECTOR STORAGE
```

```
 0043 FPYDI
01 02312 054020          STA     3,@BUFPT        ;STORE AT POINTER
02 02313 046020          STA     1,@BUFPT        ;STORE SECTOR ADDRESS
03 02314 006255 COMBG:   JSR     @ICOME          ;GO DO THE COMMAND STRING
04 02315 063710          SKPDZ   TTI             ;ANYTHING TYPED ON TTY
05 02316 000206          JSR     @IINP?          ;YES GO SERVICE
06 02317 000775          JMP     COMBG           ;NO DO STRING AGAIN
07
08                ;*************************************************************
09                ;               THIS IS USED TO CHECK GOOD TERMINATOR
10                ;                       IN COMMAND STRING
11                ;*************************************************************
12
13 02320 054223 GOODF:   STA     3,RET           ;SAVE RETURN ADDRESS
14 02321 034236          LDA     3,C15           ;GET CARRAIGGE RETURN
15 02322 162415          SUB#    3,0,SNR         ;IS IT A CARRAIGE RETURN?
16 02323 002252          JMP     @ICOMS          ;YES,AND IT SHOULD NNOT BE
17 02324 002223          JMP     @RET            ;RETURN TO CALLER
18
19 02325 054223 GOOS:    STA     3,RET           ;SAVE RETURN ADDRESS
20 02326 034236          LDA     3,C15           ;GET ASCII CARRAIGE RETURN
21 02327 162414          SUB#    3,0,SZR         ;IS IT A (CR)
22 02330 002223          JMP     @RET            ;NO RETURN TO CALLER
23 02331 010223          ISZ     RET             ;RETURN TO CALLER TO EXECUTE COM
24 02332 002223          JMP     @RET            ;GO DO IT
25                ;*************************************************************
26                ;               THIS IS TTHE COMMAND TABLE
27                ;*************************************************************
28
29 02333 000000 COMTB:   0
30 02334 000123          "S
31 02335 006100          JSR     @ISEAK          ;SEEK FUNCTION
32 02336 000122          "R
33 02337 006127          JSR     @IWCP2          ;READ FUNCTION
34 02340 000127          "W
35 02341 006127          JSR     @IWCP2          ;WRITE FUNCTION
36 02342 000000          0                       ;END OF TABLE
37
38                ;*************************************************************
39                ;       THIS IS THE COMMAND EXECUTOR
40                ;*************************************************************
41
42 02343 054200 COMEX:   STA     3,COMRE         ;SAVE RETURN ADDRESS
43 02344 000000 COMY:    0                       ;HERE ISS WHERE THE
44 02345 000000          0                       ;THE COMMAND,TRACK
45 02346 000000          0                       ;AND SECTOR IS STORED
46 02347 002200          JMP     @COMRE          ;RETURN TO CALLER
```

```
10044 FPYDI
01
02
03
04                ;*************************************************************
05                ;       THIS IS THE FUNCTIION COMMAAND SRTING INTERPRETER
06                ;*************************************************************
07
06 02350 006110 FOWL:    JSR     @IMESS          ;ASK FOR THE FUNCTION TO
09                                                ;BE PREFORMED.
10 02351 003731          TYFH
11 02352 102400          SUB     0,0             ;CLEAR AC0
12 02353 040126          STA     0,RCOUNT        ;SAVE IN COUNTER
13 02354 020241          LDA     0,IFUNX         ;GET THE FUNCTION ROUTINE
                                                  ;STARTING ADDRESS
15 02355 101400          INC     0,0             ;INCREASE BY 1
16 02356 101400          INC     0,0             ;INCREASE IT BY 1
17 02357 040021          STA     0,CBUFP         ;SAVE THE POINTER
18 02360 020242          LDA     0,IFNTB         ;GET THE FUNCTION TABLE
19                                                ;ADDRESS
20 02361 040020          STA     0,BUFPT         ;SAVE THE TABLE POINTER
21 02362 006130          JSR     @ITIN?O         ;GET THE FIRST CHARACTER
22 02363 004421          JSR     FCSH            ;GO SEAARCH THE TABLE
23 02364 000764          JMP     FOWL            ;CAN'T ACCEPT A NUMBER
24 02365 010126          ISZ     RCOUNT          ;INC FOR 2ND CHAR
25 02366 006130          JSR     @ITIN?O         ;GET THE SECOND CHARACTER
26 02367 004415          J&R     FCSH            ;GO SEARCH THE TABLE
27 02370 000760          JMP     FOWL            ;CAN'T ACCEPT THE CHARACTER
28 02371 006130          JSR     @ITIN?O         ;GET THE 3RD CHARACTER
29 02372 004412          JSR     FCSH            ;GO SEARCH THE TABLE
30 02373 000755          JMP     FOWL            ;CAN'T ACCEPT THE CHARACTER
31 02374 026020          LDA     1,@BUFPT        ;GET THE FUNCTION
32 02375 046021          STA     1,@CBUFP                ;STORE IT IN THE ROUTINE
33 02376 006130          JSR     @ITIN?O         ;GET THE S,C, OR NO FUNCTION
34 02377 000421          JMP     FCSS            ;GO SEARCH FOR IT
35 02400 102400          SUB     0,0             ;CLEAR AC0
36 02401 024243          LDA     1,NOFUN
37 02402 046021          STA     1,@CBUFP        ;STORE THE NO-OP
38 02403 006241          JSR     @IFUNX          ;GO  DO THE FUNCTION
39
40 02404 054260 FCSH:    STA     3,COMRE         ;SAVE THE RETURN ADDRESS
41 02405 030126          LDA     2,RCOUNT        ;SEE IF ITS FIRST ITME THROUGH
42 02406 151015          MOV#    2,2,SNR         ;1ST ?
43 02407 000407          JMP     FCSH1           ;RETURN TO CALLER
44 02410 032020 FCSH2:   LDA     2,@BUFPT        ;SEARCH THE COMMAND TABLE
45 02411 112415          SUB#    0,2,SNR         ;DID WE FIND A MATCH
46 02412 000404          JMP     FCSH1           ;YES
47 02413 151004          MOV     2,2,SZR         ;IS IT THE END OF THE TABLE
48 02414 000774          JMP     FCSH2           ;NO KEEP SEARCHING
49 02415 002260          JMP     @COMRE          ;CARACTER IN ERROR
50 02416 010260 FCSH1:   ISZ     COMRE           ;GOOD FIND
51 02417 002260          JMP     @COMRE          ;RETURN TO CALLER
52
53 02420 054067 FCSS:    STA     3,RTNM          ;SAVE RETURN ADDRESS
54 02421 004763          JSR     FCSH            ;FIND THE CHARACTER
55 02422 000726          JMP     FOWL            ;DID NOT FIND
56 02423 026020          LDA     1,@BUFPT        ;GET THE FUNCTION
57 02424 014021          DSZ     CBUFP           ;DECRIMENT ROUTINE POINTER
58 02425 032021          LDA     2,@CBUFP        ;GET THE FUNCTION
59 02426 133000          ADD     1,2             ;ADD THE S,C
60 02427 014021          DSZ     CBUFP
```

```
 0045 FPYDI
01 02430 052221          STA     2,@CBUFP        ;RETURN THE CORRECTED FUNCTION
02 02431 026020          LDA     1,@BUFPT        ;GET THE NEXT INSTRUCTION
03 02432 046021          STA     1,@CBUFP        ;STORE IT IN THE ROUTINE
04 02433 006110          JSR     @IMESS          ;ASK WHAT DRIVE
05 02434 020150          WHDK
06 02435 006130          JSR     @ITIN?O         ;FIND OUT WHAT UNIT
07
08 02436 000712          JMP     FOWL            ;CAN'T ACCEPT THE CHARAC.
09 02437 125005          MOV     1,1,SNR         ;UNIT 0 OR 1
10 02440 000402          JMP     FCSS1           ;UNIT 0
11 02441 024157          LDA     1,BIT0          ;GET UNIT 1 BIT
12 02442 040116 FCSS1:   STA     0,FLAGS         ;CORRECT THE FLAGS
13 02443 002241          JMP     @IFUNX          ;GO DO THE ROUTINE
14
15
16                       ;*********************************************************
17                       ;       FUNCTION COMMAND EXECUTOR
18                       ;*********************************************************
19
20 02444 102400 FUEX:    SUB     0,0             ;CLEAR AC0
21 02445 006077          JSR     @IDKI           ;SET THE SETTLE FOR THE CORRECT
22 02446 006111          JSR     @IDOOA          ;ISSUE THE FUNCTION
23 02447 000000          0
24 02450 000000          0
25 02451 063710          SKPDZ   TTI             ;WAS A KEY HIT
26 02452 006206          JSR     @IINP?          ;YES
27 02453 000771          JMP     FUEX            ;DO THE SETTLE AGAIN
28
29                       ;*********************************************************
30                       ;       FUNCTION TABLE
31                       ;*********************************************************
32
33 02454 000000 FUNTB:   0
34 02455 000117          "O
35 02456 000101          "A
36 02457 061000          DOA     0,FPY
37 02460 000102          "B
38 02461 062000          DOB     0,FPY
39 02462 000111          "I
40 02463 000101          "A
41 02464 063400          DIA     0,FPY
42 02465 000102          "B
43 02466 061400          DIB     0,FPY
44 02467 000103          "C
45 02470 062400          DIC     0,FPY
46 02471 000123          "S
47 02472 000100          100
48 02473 060077          NIO     CPU
49 02474 000103          "C
50 02475 000200          200
51 02476 060077          NIO     CPU
52 02477 000000 FNTBE:   0
53                       ;*********************************************************
54                       ;       THIS PRINTS THE TEST # AT BEGINING OF TESTS
55                       ;*********************************************************
56
57 02500 054223 PTSTB:   STA     3,RET           ;SAVE RETURN ADDRESS
58 02501 024205          LDA     1,SWREG         ;GET THE SWITCH REGISTER
59 02502 030106          LDA     2,BIT7          ;GET SWITCH 7
60 02503 147414          AND#    2,1,SZR         ;DON'T PRINT IF SWITCH 7 SET
```

```
 0046 FPYDI
01 02504 002223          JMP     @RET            ;RETURN TO CALLER
02 02505 006110          JSR     @IMESS          ;PRINT TEST #
03 02506 005362          TETNM
04 02507 024136          LDA     1,TSTNM         ;GET THE TEST NUMBER
05 02510 006212          JSR     @IZOC?T         ;PRINT THE VALUE
06 02511 002223          JMP     @RET            ;RETURN TO CALLER
07
08                       ;*********************************************************
09                       ;       THIS INCREMENTS ERTS1 IF AN ERROR OCCURRED
10                       ;*********************************************************
11
12 02512 054225 TET:     STA     3,REET          ;SAVE RETURN ADDRESS
13 02513 063710          SKPDZ   TTI             ;WAS TTY KEY HIT
14 02514 006206          JSR     @IINP?          ;YES
15 02515 022073          LDA     0,@IITR         ;GET ERROR COUNTER
16 02516 101005          MOV     0,0,SNR         ;WAS THERE ANY ERRORS
17 02517 000417          JMP     TET2            ;NO
18 02520 020205          LDA     0,SWREG         ;GET THE SWITCH REGISTER
19 02521 024166          LDA     1,BIT7          ;GET SWITCH 7
20 02522 107414          AND#    0,1,SZR         ;DON'T PRINT IF SWITCH 7 SET
21 02523 000413          JMP     TET2            ;CONTINUE
22 02524 006074          JSR     @IERME          ;PRINT THE TEST NUMBER
23 02525 005362          TETNM
24 02526 020257          LDA     0,ERTS1         ;GET THE ERROR COUNTER
25 02527 101004          MOV     0,0,SZR         ;FIRST ENCOUNTER?
26 02530 000403          JMP     TET1            ;NO, DON'T PRINT
27 02531 024136          LDA     1,TSTNM         ;GET THE TEST NUMBER
28 02532 006212          JSR     @IZOC?T         ;PRINT IN OCTAL
29 02533 010257 TET1:    ISZ     ERTS1           ;INCREMENT THE ERROR COUNTER
30 02534 000402          JMP     TET2            ;RETURN TO CALLER
31 02535 060077          NIO     CPU             ;NO-OP
32 02536 020300 TET2:    LDA     0,LPNM          ;GET THE LOOP ON TEST VALUE
33 02537 101005          MOV     0,0,SNR         ;ARE WE LOOPING ON A TEST
34 02540 002225          JMP     @REET           ;NO RETURN TO CALLER
35 02541 034463          LDA     3,TSTBL         ;GET TEST TABLE ADDRESS
36 02542 117000          ADD     0,3             ;CREATE POINTER TO TEST
37                                               ;BEING LOOPED ON
38 02543 021400          LDA     0,0,3           ;GET TEST STARTING ADDRESS
39 02544 040225          STA     0,REET          ;FIX THE RETURN ADDRESS
40 02545 102400          SUB     0,0             ;CLEAR AC0
41 02546 040257          STA     0,ERTS1         ;CLEAR THE ERROR FLAG
42 02547 002225          JMP     @REET           ;RESTART THE TEST
43
44                       ;*********************************************************
45                       ;               HERE IS WHERE I LOOP ON SELECTED TESTS
46                       ;*********************************************************
47
48 02550 054225 TETLP:   STA     3,REET          ;SAVE RETURN ADDRESS
49 02551 006110 LP1:     JSR     @IMESS          ;ASK FOR TEST #
50 02552 005362          TETNM
51 02553 061304 LP5:     JSR     @ITIN?O         ;ACCEPT THE #
52 02554 006775          JMP     LP1             ;NO ALPHA
53 02555 044300          STA     1,LPNM          ;SAVE THE TEST #
54 02556 044136          STA     1,TSTNM         ;SAVE THE CORRECT TEST #
55 02557 034004          LDA     3,TETMX         ;GET THE MAX TEST #
56 02560 164404          SUB     3,1,SZR         ;SEE IF LEAGLE TEST #
57 02561 000426          JMP     LP5             ;SEE IF LESS THAN MAX #
58 02562 010203 LP7:     ISZ     PASS            ;INC THE PASS COUNTER
59 02563 030174          LDA     2,BIT13         ;FIND OUT IF TEST 14
60 02564 034173          LDA     3,BIT12         ;IS THE TEST #
```

```
 0047 FPYDI
01 02560 173000          ADD     3,2             ;CREATE A VALUE OF 14
02 02561 024300          LDA     1,LPNM          ;GET THE TEST #
03 02562 146415          SUB#    2,1,SNR         ;IS IT 14
04 02563 000422          JMP     LP3             ;YES GO FOR 2 DRIVES
05 02564 006110 LP2:     JSR     @IMESS          ;ASK WHAT DRIVE IS TO BE TESTED
06 02565 005150          WHDR
07 02566 006130          JSR     @ITIN?O         ;GET THE DRIVE # TO BE TESTED
08 02567 000775          JMP     LP2             ;NO ALPHA
09 02570 044272          STA     1,DRI1          ;SAVE AT DRIVE FLAG
10 02571 125004          MOV     1,1,SZR         ;IS IT DRIVE 0
11 02572 125240          MOVOR   1,1             ;NO SO SET FLAGS FOR 1
12 02573 044116          STA     1,FLAGS         ;SAVE THE DRIVE BIT
13 02574 034423 LP4:     LDA     3,TSTBL         ;GET THE TEST TABLE ADDRESS
14 02575 030300          LDA     2,LPNM          ;GET THE TEST NUMBER
15 02576 157000          ADD     2,3             ;CREATE POINTER TO STARTING
16 02577 021400          LDA     0,0,3           ;ADDRESS AND GRAB IT
17 02600 040225          STA     0,REET          ;SAVE THE ADDRESS
18 02601 002225          JMP     @REET           ;GO DO THE TEST
19 02602 125123 LP5:     MOVZL   1,1,SNC         ;IS IT MINUS
20 02603 000741          JMP     LP1             ;BAD TEST #
21 02604 000751          JMP     LP7             ;TEST # OK
22 02605 126400 LP3:     SUB     1,1             ;CLEAR AC1
23 02606 044116          STA     1,FLAGS         ;SET THE FLAGS TO DRIVE 0
24 02607 044234          STA     1,NUDEV         ;CLEAR # DEVICES
25 02610 044272          STA     1,DRI1          ;CLEAR DRIVE # TO 0
26 02611 010234          ISZ     NUDEV           ;FORCE #OF DEVICES TO SAY 2
27 02612 006131          JSR     @INCPAT         ;FORMAT THE DISKET WITH CORRECT
28                                               ;INFO
29 02613 024157          LDA     1,BIT0          ;GET DRIVE 1 FLAG
30 02614 044116          STA     1,FLAGS         ;SET UP FOR DRIVE 1
31 02615 006131          JSR     @INCPAT         ;NOW DO THE SAME FOR DRIVE 1
32 02616 000756          JMP     LP4             ;GO DO TEST 14
33
34 02617 002620 TSTBL:   TSTBQ
35 02620 000624 TSTBQ:   STAT6
36 02621 000624          STAT6
37 02622 000674          TST2+1
38 02623 000741          TST3+1
39 02624 001032          TST4+1
40 02625 001175          TST5+1
41 02626 001267          TST6+1
42 02627 001344          TST7+1
43 02630 001423          TST10+1
44 02631 001500          TST11+1
45 02632 001517          TST12+1
46 02633 001616          TST13+1
47 02634 001705          TST14+1
48 02635 002001          TST15+1
49
50
51                       ;*****************************************************************
52                       ;               LOAD THE BUFFER (WRITE) WITH ALL TEST PATTERNS
53                       ;*****************************************************************
54
55 02636 054223 DOPAT:   STA     3,RET           ;SAVE RETURN ADDRESS
56 02637 024245          LDA     1,IBFPAT        ;GET BUFFER PATTERNS ADDRESS
57 02640 044020          STA     1,BUFPT         ;SAVE IN AUTO INC LOC
58 02641 024244          LDA     1,IBFPED        ;GET THE BUFFER PATTERN END
59 02642 034133          LDA     3,IWBUF         ;GET THE WRITE BUFFER ADDRESS
60 02643 054177          STA     3,IBUF          ;MAKE IT THE WRITE BUFFER
```

```
 0048 FPYDI
01 02644 054021          STA     3,CBUFP         ;SAVE IN AN AUTO INC LOC
02 02645 032020 DOPA2:   LDA     2,@BUFPT        ;GET PATTERN
03 02646 052021          STA     2,@CBUFP        ;STORE IT IN BUF
04 02647 030020          LDA     2,BUFPT         ;GET CURRENT ADDRESS
05 02650 146414          SUB#    2,1,SZR         ;HAVE WE EXHAUSTED THE BUFFER
06 02651 000403          JMP     DOPA1           ;NO,SEE IF END OF WRITE BUFFER
07 02652 030245          LDA     2,IBFPAT        ;GET BUFFER PATTERN ADDRESS
08 02653 050020          STA     2,BUFPT         ;SAVE THE ADDRESS
09 02654 030021 DOPA1:   LDA     2,CBUFP         ;GET CURRET ADDRESS
10 02655 034134          LDA     3,BUFED         ;IS IT END OF BUFFER
11 02656 172414          SUB#    3,2,SZR         ;END OF BUFFER?
12 02657 000766          JMP     DOPA2           ;CONTINUE LOADING
13 02660 002223          JMP     @RET            ;RETURN TO CALLER
14
```

```
16049 FPYUI
01
02                ;*****************************************************************
03                ;       BUFFER PATTERN
04                ;*****************************************************************
05
06 02660 000000 BFPAT:  0
07 02661 000000         0
08 02662 177777         -1
09 02663 177776         -2
10 02664 177774         -4
11 02665 177770         -10
12 02666 177760         -20
13 02667 177740         -40
14 02670 177700         -100
15 02671 177600         -200
16 02672 177400         -400
17 02673 177000         -1000
18 02674 176000         -2000
19 02675 174000         -4000
20 02676 170000         -10000
21 02677 160000         -20000
22 02700 140000         -40000
23 02701 100000         -100000
24 02702 000001         1
25 02703 000002         2
26 02704 000004         4
27 02705 000010         10
28 02706 000020         20
29 02707 000040         40
30 02710 000100         100
31 02711 000200         200
32 02712 000400         400
33 02713 001000         1000
34 02714 002000         2000
35 02715 004000         4000
36 02716 010000         10000
37 02717 020000         20000
38 02720 040000         40000
39 02721 100000         100000
40 02722 052525         052525
41 02723 125252         125252
42 02724 111111         111111
43 02725 122222         122222
44 02726 133333         133333
45 02727 144444         144444
46 02730 155555         155555
47 02731 166666         166666
48 02732 177777         177777
49 02733 000000 BFPED:  0
```

```
16050 FPYUI
01
```

```
10051 FPYDI
01                ;THIS SECTION CONTAINS THE READ ,WRITE AND TIME SUBROUTINES
02                ;IT ALSO CONTINS ALL THE DIAGNOSTIC SUPPORT ROUTINES
03                ;*****************************************************************
04
05                ;?/?/?/?/?/?/?/?/?/?/?/?/?/?/?/?/?/?/?/?/?/?/?/?/?/?/?/?/?/?/?/?/?/?/?
06                ;                       SEEK SUBROUTINE
07                ;
08                ;  WHEN CALLED NEXT WORD AFTER JSR CONTINS THE TRAK TO BE SEEKED
09                ;?/?/?/?/?/?/?/?/?/?/?/?/?/?/?/?/?/?/?/?/?/?/?/?/?/?/?/?/?/?/?/?/?/?/?
10
11 02742 054226 SEAK:   STA     3,R?TN          ;SAVE AC3 IN RETURN POINTER
12 02743 054137         STA     3,DIFF          ;MAKE SURE DIFF IS NOT =1
13 02744 006075         JSR     @IRPRE          ;READ THE PREAMBLE
14 02745 006111         JSR     @IDOOA          ;ISSUE FUNCTION
15 02746 076400         DIC     3,FPY           ;GET TRACK SECTOR WORD FROM FLOP
16 02747 175300         MOVS    3,3             ;GET TRACK BITS IN LOWER BYTE
17 02750 024156         LDA     1,C1776         ;MASK OUT THE SECTOR BITS
18 02751 137420         ANDZ    1,3             ;AND KEEP THE TRACK BITS
19 02752 024157         LDA     1,BIT0          ;REPLACE THE DRIVE BIT FOR FUTUR
20 02753 032226         LDA     2,@R?TN         ;GET TRACK NUMBER
21 02754 050444         STA     2,SEAK5         ;SAVE THE TRACK ADDRESS
22 02755 156425         SUBZ    2,3,SNR         ;ARE WE ON THE RIGHT TRACK?
23 02756 000437         JMP     OVER            ;YES ,GO RETURN TO WHO CALLED YO
24 02757 020137         LDA     0,DIFF          ;GET DIFFERANCE STORAGE
25 02760 101404         INC     0,0,SZR         ;HAVE WE ALREADY FOUND OUR TRACK
26 02761 000405         JMP     SEAK9           ;NO ,CONTINUE LOOKING
27 02762 006074         JSR     @IERME          ;WE HAVE AN UNFORMATED DISK
28                                              ; OR HEADS ARE NOT
29                                              ;MOVING.....?
30 02763 005063         HEDOF                   ;TELL THE OPERATOR
31 02764 000435         JMP     SEAK4           ;WE HAVE AN ERROR SO SEAK
32                                              ;MECHANICALY
33 02765 000430         JMP     OVER            ;GO BACK AND LOOP
34 02766 020146 SEAK9:  LDA     0,STEPIN        ;GET STEPOUT FUNCTION
35 02767 054137 MOVMED: STA     3,DIFF          ;STORE AC3 IN DIFFERANCE STORAGE
36 02770 167435         ANDZ#   3,1,SNR         ;DETERMINE IF MOVE HEAD IN OR OU
37 02771 000421         JMP     MVHDIN          ;MOVE HEAD IN
38 02772 006077         JSR     @IDRI           ;SET IT UP FOR CORRECT DRIVE
39 02773 006111 AGAIN:  JSR     @IDOOA          ;ISSUE FUNCTION
40 02774 061100         DOAS    0,FPY           ;TELL THE FLOPPY TO MOVE HEAD
41 02775 006105         JSR     @ITIME          ;WAIT FOR DONE
42 02776 006111         JSR     @IDOOA          ;READ STATUS
43 02777 070400         DIA     2,FPY
44 03000 034160         LDA     3,BIT1          ;GET HEAD HOME BIT
45 03001 157415         AND#    2,3,SNR         ;IS THE HEAD HOME
46 03002 000402         JMP     SCRA1           ;KEEP SEAKING
47 03003 000403         JMP     SCRAP           ;GO SEE IF ITS TRACK 0
48                                              ;WE ARE SEAKING
49 03004 010137 SCRA1:  ISZ     DIFF            ;SEE IF WE KEEP MOVING THE HEAD
50 03005 000766         JMP     AGAIN           ;YES DO IT AGAIN
51 03006 102400 SCRAP:  SUB     0,0             ;PUT -1 IN AC0
52 03007 100000         COM     0,0             ;MAKE IT -1
53 03010 040137         STA     0,DIFF          ;STORE IT IN DIFFERANCE STORAGE
54 03011 000733         JMP     SEAK+2          ;GO MAKE SURE WE ARE THERE
55 03012 020145 MVHDIN: LDA     0,STEPOU        ;WE ARE GOING TO STEP IN
56 03013 174400         NEG     3,3             ;COMPLIMENT THE CONTENTS OF AC3
57 03014 000753         JMP     MOVMED          ;GO MOVE THE HEAD
58 03015 010226 OVER:   ISZ     R?TN            ;INCREMENT RETURN POINTER
59 03016 010226         ISZ     R?TN            ;INCREMENT RETURN POINTER
60                                              ;SHOULD NEVER SKIP BECAUSE OF 0
```

```
0052 FPYDI
01 03017 002226         JMP     @R?TN           ;RETURN TO THE CALLER
02 03020 000000 SEAK5:  0                       ;TRACK STORAGE FOR MECH SEEK
03 03021 063710 SEAK4:  SKPDZ   TTI
04 03022 006006         JSR     @IINP?
05 03023 024147         LDA     1,C77           ;GET THE MAX # OF TRACKS
06 03024 044126         STA     1,RCOUNT        ;SAVE SO WE DON'T HANG UP
07 03025 050773         STA     2,SEAK5         ;SAVE THE TRACK ADDRESS
08 03026 006111         JSR     @IDOOA          ;GET THE STATUS
09 03027 070400         DIA     2,FPY           ;GET THE STATUS
10 03030 034160         LDA     3,BIT1          ;GET THE HEAD HOME BIT
11 03031 157414         AND#    2,3,SZR         ;IS THE HEAD HOME
12 03032 000421         JMP     SEAK7           ;YES
13 03033 020145 SEAK2:  LDA     0,STEPOU        ;GET TTHE STEP TO 0 FUNC
14 03034 006077         JSR     @IDRI           ;SET THE STEP COMMAND TO THE
                                                ;CORRECT DRIVE
15
16 03035 006111         JSR     @IDOOA          ;ISSUE THE FUNC
17 03036 061100         DOAS    0,FPY           ;ISSUE STEP COMMAND
18 03037 006105         JSR     @ITIME          ;WAIT FOR DONE
19 03040 006111         JSR     @IDOOA          ;READ THE STATUS
20 03041 060400         DIA     0,FPY
21 03042 024160         LDA     1,BIT1          ;GET THE HEAD HOME BIT
22 03043 143414         AND#    2,0,SZR         ;IS THE HEAD HOME?
23 03044 000407         JMP     SEAK7           ;YES
24 03045 014126         DSZ     RCOUNT          ;DEC THE MAX TRACK COUNT
25 03046 000765         JMP     SEAK2           ;CONTINUE TEST
26 03047 006074         JSR     @IERME          ;REPORT HEAD HOME FAILURE
27 03050 003676         HDHMC
28 03051 006104         JSR     @IEHR           ;REPORT STANDARD ERROR
29 03052 000743         JMP     OVER            ;RETURN TO THE CALLER
30 03053 020146 SEAK7:  LDA     0,STEPIN        ;GET CORRECT FUNC TO POSS
                                                ;ITION THE HEAD
31
32 03054 006077         JSR     @IDRI           ;SET IT UP FOR CORRECT DRIVE
33 03055 030743         LDA     2,SEAK5         ;GET THE TRACK ADDRESS
34 03056 151005         MOV     2,2,SNR         ;IS IT TRACK 0?
35 03057 000406         JMP     SEAK8           ;YES
36 03060 006111 SEAK6:  JSR     @IDOOA          ;ISSUE THE FUNCTION
37 03061 061100         DOAS    0,FPY
38 03062 006105         JSR     @ITIME          ;WAIT FOR DONE
39 03063 014735         DSZ     SEAK5           ;DECREMENT THE TRACK COUNT
40 03064 000774         JMP     SEAK6           ;ISSUE HEAD STEP AGAIN
41 03065 024116 SEAK8:  LDA     1,FLAGS         ;GET THE FLAGS
42 03066 030176         LDA     2,BIT15         ;GET  FATAL ERROR FLAG
43 03067 147414         AND#    2,1,SZR         ;HAVE WE HAD A FATAL ERROR
44 03070 000410         JMP     SEAK5           ;YES SEE IF WE CAN LOOP ON IT
45 03071 147000         ADD     2,1             ;SET THE FATAL ERROR FLAG
46 03072 044116         STA     1,FLAGS         ;SAVE THE FLAGS
47 03073 006110         JSR     @IMESS          ;READ PREAMBLE FAILURE
48 03074 004765         RDPAF
49 03075 026226         LDA     1,@R?TN         ;GET THE TRAK ADDRESS
50 03076 006212         JSR     @IZOC?T         ;PRINT THE TRAK #
51 03077 000707         JMP     SCRAP           ;RETURN TO VERIFY THE TRACK
52 03100 024205 SEAK5:  LDA     1,SWREG         ;GET SWITCH REGISTER
53 03101 030160         LDA     2,BIT1          ;GET THE NO LOOP ON ERROR SWITCH
54 03102 147414         AND#    2,1,SZR         ;SEE IF WE LOOP ON ERRORS
55 03103 063077         HALT                    ;NO SO STOP ON FATAL ERROR
56 03104 014116         DSZ     FLAGS           ;CLEAR THE FATAL ERROR FLAG
57 03105 000637         JMP     SEAK+2          ;CONTINUE
```

```
#0053 FPYDI
01
02
03
04                ;*****************************************************************
05                ;  THIS IS THE READ WRITE SUBROUTINE TO DO A READ THE SUBROUTINE
06                ;  READ IS ENTERED IT WILL INTERN ENTER THIS ROUTINE AT DUDLEY W
07                ;  IT WILL SET IT UP TO DO A READ.................................
08                ;  WHEN CALLING A WRITE IT WILL ENTER THIS ROUTINE AT THE LABLE
09                ;  AND SET UP TO PREFORM A WRITE OPERATION........................
10                ;*****************************************************************
11
12 03100 054070 DURITE: STA     3,RT?N          ;STORE THE RETURN ADDRESS
13 03101 034121         LDA     3,CN7           ;GET A -7
14 03110 054126         STA     3,RCOUNT        ;SAVE IT IN ROTATE COUNT
15 03111 024160         LDA     1,BIT1          ;GET DRIVE BIT
16 03112 020116         LDA     0,FLAGS         ;GET THE FLAGS FOR THE WRITE
17                                              ;SWITCH
18 03113 123414         AND#    1,0,SZR         ;SEE IF READ OR WRITE
19 03114 000405         JMP     .+5             ;GO SETUP FOR READ
20 03115 020143         LDA     0,WITFUN        ;GET WRITE FUNCTION
21 03116 006077         JSR     @IDRI           ;SET IT UP FOR CORRECT DRIVE
22 03117 054135         STA     0,RITFUN        ;STORE IT AT RIGHT FUNCTION
23 03120 000404         JMP     .+4             ;SKIP OVER READ FUNCTION SET UP
24 03121 020142         LDA     0,REDFUN        ;GET READ FUNCTION
25 03122 006077         JSR     @IDRI           ;SET IT UP FOR CORRECT DRIVE
26 03123 054135         STA     0,RITFUN        ;STORE AT RIGHT FUNCTION STORAGE
27 03124 022070         LDA     0,@RT?N         ;GET TRACK ADDRESS
28 03125 010070         ISZ     RT?N            ;UPDATE POINTER
29 03126 054422         STA     0,TRACKS        ;STORE IT AT TRACKS WORD
30 03127 024124         LDA     1,C42           ;GET READY TO SEE IF WE NEED WRI
31 03130 100432         SUBZ#   0,1,SZC         ;IS IT 43 OR GREATER?
32 03131 000405         JMP     .+5             ;GO GET SECTOR ADDRESS
33 03132 030160         LDA     2,BIT1          ;GET WRITE SWITCH BIT
34 03133 024135         LDA     1,RITFUN        ;GET THE RIGHT FUNCTION
35 03134 133000         ADD     1,2             ;CREATE THE RIGHT FUNCTION WITH/
36                                              ;WRITE SWITCH
37 03135 054135         STA     2,RITFUN        ;SAVE THE RIGHT FUNCTION
38 03136 022070         LDA     0,@RT?N         ;GET THE SECTOR ADDRESS
39 03137 010070         ISZ     RT?N            ;UPDATE POINTER
40 03140 054444         STA     0,SECTOR        ;SAVE IT AT SECTOR ADDRESS WORD
41 03141 101120         MOVZL   0,0             ;NOW RATATE SECTOR ADDRESS UNTIL
42 03142 010126         ISZ     RCOUNT          ;ARE IN THE CORRECT POSITION
43 03143 000776         JMP     .-2             ;WE ARE NOT DONE,CONTINUE ROTATI
44 03144 030135         LDA     2,RITFUN        ;GET THE RIGHT FUNCTION
45 03145 113000         ADD     0,2             ;CREATE RIGHT FUNCTION WITH THE
46 03146 054117         STA     2,FUNWD         ;SAVE IT AT FUNCTION WORD STORAG
47 03147 006100         JSR     @ISEAK          ;NOW MAKE SURE WE ARE ON THE RIG
48 03150 000000 TRACKS: 0                       ;TRACK ADDRESS STORAGE
49 03151 000000         0
50 03152 006101         JSR     @ISET           ;MAKE SURE HEADS ARE LOADED
51 03153 006075         JSR     @IRPRE          ;TELL FLOPPY TO READ PREAMBLE
52 03154 006111         JSR     @IDOOA          ;ISSUE FUNCTION
53 03155 062400         DIC     0,FPY           ;GET TRACK/SECTOR WORD FROM FLOP
54 03156 034155         LDA     3,C1777         ;GET SECTOR MASK
55 03157 163000         AND     3,0             ;CLEAR THE TRACK BITS
56 03160 034424         LDA     2,SECTOR        ;GET SECTOR NUMBER
57 03161 014423         DSZ     SECTOR          ;SUBTRACT 1FROM ADDRESS
58 03162 060077         NIO     CPU             ;JUST INCASE SECTOR IS ONE
59                                              ;WE NEED THIS SO WE WON'T BOMB
60 03163 024421         LDA     1,SECTOR        ;GET CORRECTED SECTOR ADDRESS
```

```
#054 FPYDI
01 03164 054420         STA     2,SECTOR        ;REPLACE SECTOR VALUE
02 03165 125120         MOVZL   1,1             ;GET SECTOR IN RIGHT POSITION
03 03166 125120         MOVZL   1,1             ;ONE MORE TIME
04 03167 167400         AND     3,1             ;GET RID OF ANY GARBADGE
05 03170 122404         SUB     1,0,SZR         ;IF CORRECT SECTOR DO FUNCTION
06 03171 000422         JMP     SECW            ;KEEP SEARCHING FOR CORRECT SECT
07 03172 030177 REENTR: LDA     2,IBUF          ;GET BUFFER ADDRESS
08 03173 000111         JSR     @IDOOA          ;ISSUE FUNCTION
09 03174 072000         DOB     2,FPY           ;GIVE IT TO FLOPPY
10 03175 030117         LDA     2,FUNWD         ;GET THE FUNCTION WORD
11 03176 000111         JSR     @IDOOA          ;ISSUE FUNCTION
12 03177 071100         DOAS    2,FPY           ;TELL FLOPPY TO DO FUNCTION
13 03200 006105         JSR     @ITIME          ;WAIT FOR DONE
14 03201 000111         JSR     @IDOOA          ;ISSUE FUNCTION
15 03202 060600         DIAC    0,FPY           ;GET STATUS BEFORE WE RETURN
16 03203 002070         JMP     @RT?N           ;RETURN TO THE CALLER
17 03204 000000 SECTOR: 0                       ;SECTOR ADDRESS STORAGE
18 03205 024166 CONRW:  LDA     1,BIT7          ;GET BIT TO INCREASE SECTOR ADDR
19 03206 030117         LDA     2,FUNWD         ;GET THE FUNCTION WORD
20 03207 133000         ADD     1,2             ;INCREASE SECTOR ADDRESS BY 1
21 03210 054117         STA     2,FUNWD         ;REPLACE THE FUNCTION WORD
22 03211 056070 CONRTW: STA     3,@RT?N         ;STORE RETURN ADDRESS
23 03212 000760         JMP     REENTR          ;GO REENTER THE SUBROUTINE
24 03213 010125 SECW:   ISZ     RCOUNT          ;INCREMENT TIMER
25 03214 000737         JMP     TRACKS+3        ;KEEP LOOKING FOR SECTOR
26 03215 006074         JSR     @IERME          ;REPORT CAN'T GET SECTOR COMPARE
27 03216 041107         CNFSC
28 03217 006104         JSR     @IERR           ;REPORT SECTOR ADDRESS DOES NOT
                                                ;COMPARE WITH ANY SECTOR ADDRESS
                                                ;READ OF THE FLOPPY.
29
30
31 03220 002070         JMP     @RT?N           ;RETURN TO CALLER
32
```

```
10055 FPYDI
01
02                ;*****************************************************************
03                ;   THIS IS THE READ SUBROUTINE THE THREE ADDRESS AFTER THE JSR
04                ;   RUCTION MUST CONTAIN TRACK ADDRESS ,SECTOR ADDRESS AND CONSE
05                ;   READ COUNT (-1 IF ONLY ONE READ TO BE DONE,.................
06                ;   AFTER THE INFORMATION IS GATHERED THIS ROUTINE WILL THEN TRA
07                ;   TO THE READ/WRITE SUBROUTINE AT LABLE DUDLEY WHERE THE FUNCT
08                ;   PREFORMED.....................................................
09                ;*****************************************************************
10
11 03221 054071 READ:   STA     3,RTN?N         ;SAVE RETURN ADDRESS
12 03222 022071         LDA     0,@RTN?N        ;GET TRACK ADDRESS
13 03223 010071         ISZ     RTN?N           ;UPDATE POINTER
14 03224 040415         STA     0,TRAKS         ;STORE AT THE TRAK STORAGE LOCAT
15 03225 022071         LDA     0,@RTN?N        ;GET THE SECTOR ADDRESS
16 03226 010071         ISZ     RTN?N           ;UPDATE POINTER
17 03227 040413         STA     0,SECS          ;STORE IT AT SECTORE STORAGE
18 03230 022071         LDA     0,@RTN?N        ;GET CONSECUTIVE READ COUNT
19 03231 010071         ISZ     RTN?N           ;UPDATE POINTER
20 03232 040424         STA     0,CONSEC        ;STORE IT AT THE CONSECUTIVE COU
21 03233 020116         LDA     0,FLAGS         ;GET THE FLAGS
22 03234 024160         LDA     1,BIT1          ;GET READ/WRITE BIT
23 03235 107415         AND#    0,1,SNR         ;IS READ SET
24 03236 123000         ADD     1,0             ;SET READ BIT
25 03237 040116         STA     0,FLAGS         ;RETURN NEW FLAGS
26 03240 004646         JSR     DURITE          ;GO DO THE READ
27 03241 000000 TRAKS:  0                       ;TRACK ADDRESS STORAGE
28 03242 000000 SECS:   0                       ;SECTORE ADDRESS STORAGE
29 03243 010413         ISZ     CONSEC          ;INCREMENT THE CONSECUTIVE READ
30 03244 000402         JMP     .+2             ;CONTINUE READING
31 03245 002071         JMP     @RTN?N          ;NO MORE TO READ ,RETURN TO CALL
32 03246 034117         LDA     3,FUN#D         ;GET READY TO FIND OUT IF OUR NE
33 03247 030163         LDA     2,BIT4          ;GET BIT 4 FOR TEST
34 03250 173435         AND2#   3,2,SNR         ;SEE IF OVERFLOW
35 03251 000405         JMP     .+5             ;OK DO THE READ
36 03252 156420         SUB2    2,3             ;CLEAR THE SECTOR ADDRESS
37 03253 054117         STA     3,FUN#D         ;REPLACE THE FUNCTION WORD
38 03254 004731         JSR     CONR#           ;GO REEENTER READ ROUTINE
39 03255 000766         JMP     SECS+1          ;SEE IF ANY MORE TO READ
40 03256 000000 CONSEC: 0                       ;CONSECUTIVE READ COUNT STORAGE
```

```
10056 FPYDI
01
02
03
04                ;*****************************************************************
05                ;       TIME ROUTINE   (WAIT FOR FLOPPY DONE)
06                ;*****************************************************************
07
08 03257 054223 TIME:   STA     3,RET           ;SAVE RETURN ADDRESS
09 03260 054204         STA     3,AC3?          ;SAVE CALLERS ADDRESS INCASE
10                                              ;OF ERROR
11 03261 175400         SUB     3,3             ;CLEAR AC3
12 03262 054125         STA     3,TIMER         ;CLEAR TIMER
13 03263 034261         LDA     3,UNITA         ;GET UNIT #
14 03264 024204         LDA     1,FLAT          ;GET SKIP FUNCTION
15 03265 167000         ADD     3,1             ;ADD THE DEVICE CODE
16 03266 044403         STA     1,FLAT1         ;STORE THE FUNCTION
17 03267 000402         JMP     FLAT1           ;GO DO IT
18 03270 063600 FLAT:   SKPDN   FPY             ;IS FLOPPY DONE
19 03271 063600 FLAT1:  SKPDN   FPY             ;IS FLOPPY DONE
20 03272 000402         JMP     .+2             ;NO GO INCREMENT THE TIMER
21 03273 002223         JMP     @RET            ;YES RETURN TO THE CALLER
22
23 03274 010125         ISZ     TIMER           ;INCREMENT THE TIME
24 03275 000774         JMP     FLAT1           ;CONTINUE WAITING
25 03276 006074         JSR     @IERME          ;REPORT FLOPPY WON'T GO DONE
26 03277 004612         FPYND                   ;FLOPPY DONE BIT WON'T SET
27 03300 006104         JSR     @IERR           ;REPORT ERROR
28 03301 010257         ISZ     ERTS1           ;DON'T LET IT PRINT ANY MORE
29 03302 002223         JMP     @RET            ;RETURN TO CALLER
30 03303 002223         JMP     @RET            ;RETURN TO CALLER
31
32
33                ;*****************************************************************
34                ;       THIS ROUTINE WILL SET THE WRITE /READ FLAG FOR A WRITE
35                ;*****************************************************************
36
37 03304 054223 WRSET:  STA     3,RET           ;SAVE RETURN ADDRESS
38 03305 020116         LDA     0,FLAGS         ;GET THE FLAGS
39 03306 024160         LDA     1,BIT1          ;GET READ /WRITE BIT
40 03307 124000         COM     1,1             ;COMPLIMENT THE BIT
41 03310 107400         AND     0,1             ;MAKE BIT 0 CLEAR
42 03311 044116         STA     1,FLAGS         ;REPLACE THE FLAGS
43 03312 006111         JSR     @IDOOA          ;READ THE STATUS
44 03313 060400         DIA     0,FPY
45 03314 024165         LDA     1,BIT6          ;GET THE WRITE PROTECT BIT
46 03315 123415         AND0#   1,0,SNR         ;ARE WE WRITE PROTECTED
47 03316 000403         JMP     WRSE1           ;NO
48 03317 006074         JSR     @IERME          ;PRINTT MESSAGE
49 03320 004111         CWPR                    ;DEVICE WRITE PROTECTED
50 03321 002223 WRSE1:  JMP     @RET            ;RETURN TO CALLER
51
52                ;*****************************************************************
53                ;           THIS IS THE READ PREAMBLE SUBROUTINE
54                ;*****************************************************************
55
56 03322 054067 RPRE:   STA     3,RTNM          ;SAVE RETURN ADDRESS
57 03323 006101         JSR     @ISET           ;SETTLE THE DRIVE
58 03324 020144         LDA     0,RDPA          ;GET READ PREAMBLE FUNCTION
59 03325 006077         JSR     @IDRI           ;SET IT UP FOR CORRECT DRIVE
60 03326 006111         JSR     @IDOOA          ;ISSUE FUNCTION
```

```
 0057 FPYDI
01 03327 061100          DOAS    0,FPY           ;TELL FLOPPY TO READ PREAMBLE
02 03330 006105          JSR     @ITIME          ;WAIT FOR DONE
03 03331 002007          JMP     @RTNH           ;RETURN TO CALLER
04
```

```
 0058 FPYDI
01
02                ;*****************************************************************
03                ;  THIS IS THE WRITE READ AND COMPARE PATTERNS SUBROUTINE
04                ;*****************************************************************
05
06 03332 054072 WCPAT:  STA     3,RTWCP         ;STORE RETURN ADDRESS
07 03333 054204         STA     3,AC3?          ;SAVE AC3 FOR ERROR PRINT OUT
08 03334 034257         LDA     3,ERTS1         ;GET ERROR SWITCH
09 03335 175004         MOV     3,3,SZR         ;ARE WE HERE ON AN ERROR LOOP
10 03336 000433         JMP     WCPAA           ;YES SO DON'T CLEAR SECTOR AND
11                                              ;TRACK ADDRESSES
12 03337 176400         SUB     3,3             ;CLEAR AC3
13 03340 054437         STA     3,TRACC         ;INITIALIZE TRACK AND SECTOR
14 03341 054437         STA     3,SECCC         ;POINTERS
15 03342 054443         STA     3,RTRK
16 03343 054443         STA     3,RSEC
17 03344 006246         JSR     @IDOPAT         ;LOAD THE NORMAL PATTERN
18 03345 000424         JMP     WCPAA           ;NORMAL ENTRY CONTINUE
19 03346 054072 WCPA2:  STA     3,RTWCP         ;COMMAND STRING ENTTRY
20 03347 026072         LDA     1,@RTWCP        ;GET POINTER TO INFO
21 03350 044427         STA     1,TRACC         ;STORE TRACK AT TRACK
22 03351 044434         STA     1,RTRK          ;AND STORE IT AT READ TRACK
23 03352 010072         ISZ     RTWCP           ;INCREMENT RETURN POINTER
24 03353 026072         LDA     1,@RTWCP        ;GET SECTOR ADDRESS
25 03354 044424         STA     1,SECCC         ;SAVE AT WRITE SECTOR
26 03355 044431         STA     1,RSEC          ;SAVE AT READ SECTOR
27 03356 054235         STA     3,CUENT         ;SET THE COMMAND STRING FLAG
28 03357 010072         ISZ     RTWCP           ;INCREMENT RETURN POINTER
29 03360 024263 WCPA1:  LDA     1,WPAT          ;GET THE WRITE PATTERN
30 03361 036133         LDA     2,IWBUF         ;GET BUFFER POINTER
31 03362 050177         STA     2,IBUF          ;SAVE AT BUFFER POINTER (INDEREC
32 03363 050202         STA     2,BUFPT         ;STORE AT AUTO INCREMENT LOC
33 03364 036134         LDA     2,BUFED         ;LOAD AC2 WITH BUFFER END ADDRES
34 03365 040020         STA     1,@BUFPT        ;LOAD BUFFER WITH PATTERN
35 03366 024020         LDA     0,BUFPT         ;SEE IF BUFFER FULL
36 03367 112414         SUB#    0,2,SZR         ;IS BUFFER FULL
37 03370 000775         JMP     .-3             ;NO KEEP LOADING
38 03371 063710 WCPAA:  SKPDZ   TTI             ;WAS A KEY HIT
39 03372 006206         JSR     @IINP?          ;YES
40 03373 036133         LDA     0,IWBUF         ;GET WRITE BUFFER ADDRESS
41 03374 040177         STA     0,IBUF          ;STORE IT AT BUFFER POINTER
42 03375 006076         JSR     @IWRSE          ;SET THE WRITE SWITCH
43 03376 006102         JSR     @IRITE          ;GO DO THE WRITE
44 03377 000000 TRACC:  0
45 03400 000000 SECCC:  0
46 03401 006106         JSR     @IERCK          ;SEE IF ANY ERRORS
47 03402 024132         LDA     1,IRBUF         ;GET READ BUFFER ADDRESS
48 03403 044177         STA     1,IBUF          ;STORE IT AT BUFFER POINTER
49 03404 006103         JSR     @IREAD          ;GO READ THE INFO
50 03405 000000 RTRK:   0
51 03406 000000 RSEC:   0
52 03407 177777         -1                      ;ONE READ ONLY
53 03410 006106         JSR     @IERCK          ;SEE IF ANY ERRORS
54 03411 000407         JMP     CPBUF           ;COMPARE THE READ
55 03412 054072 CPET:   STA     3,RTWCP         ;SAVE RETURN POINTER
56 03413 054235         STA     3,CUENT         ;SET THE ENTRY FLAG
57 03414 025774         LDA     1,-4,3          ;GET THE SECTOR ADDRESS
58 03415 044771         STA     1,RSEC          ;SAVE SECTOR ADDRESS
59 03416 025773         LDA     1,-5,3          ;GET THE TRACK ADDRESS
60 03417 044766         STA     1,RTRK          ;SAVE THE TRACK ADDRESS
```

```
 0059 FPYUI
01 03424 063710 CPBUF:  SKPDZ   TTI             ;DOES TTY NEED SERVICE
02 03421 006206         JSR     @IINP?          ;YES
03 03422 024132         LDA     1,IRBUF         ;GET READ BUFFER ADDRESS
04 03423 044020         STA     1,BUFPT         ;STORE IT AT BUFFER POINTER
05 03424 024133         LDA     1,IWBUF         ;GET WRITE BUFFER ADDRESS
06 03425 044021         STA     1,CBUFP         ;STORE AT COMPARE BUFFER POINTER
07 03426 024134         LDA     0,BUFED         ;GET BUFFER END
08 03427 026020 HERE:   LDA     1,@BUFPT        ;GET DATA
09 03430 032021         LDA     2,@CBUFP        ;GET COMPARE DATA
10 03431 146414         SUB#    2,1,SZR         ;EQUAL DATA?
11 03432 000435         JMP     CPERR           ;DATA DOES NOT COMPARE
12 03433 024021         LDA     1,CBUFP         ;SEE IF END OF BUFFER
13 03434 122414         SUB#    1,0,SZR         ;ARE WE THROUGH COMPARING?
14 03435 000772         JMP     HERE            ;NO,KEEP COMPARING
15 03436 026073 UPDAT:  LDA     1,@IITR         ;SEE IF ANY ERRORS
16 03437 125004         MOV     1,1,SZR         ;ANY ERRORS
17 03440 002072         JMP     @RTWCP          ;RETURN TO LOOP
18 03441 024235         LDA     1,COENT         ;WAS IT A COMMAND ENTRY
19 03442 125004         MOV     1,1,SZR         ;FIND OUT
20 03443 000423         JMP     WCPA3           ;YES SO RETUR TO SENDER CALLER
21 03444 010734         ISZ     SECCC           ;UPDATE SECTOR (WRITE)
22 03445 010741         ISZ     RSEC            ;UPDATE READ SECTOR
23 03446 024732         LDA     1,SECCC         ;GET NEW SECTOR NUMBER
24 03447 030120         LDA     2,SECTST        ;GET SECTOR TEST
25 03450 147414         AND#    2,1,SZR         ;HAVE WE DONE ALL SECTORS
26 03451 000720         JMP     WCPAA           ;NO KEEP GOING
27 03452 126400         SUB     1,1             ;CLEAR AC0
28 03453 044725         STA     1,SECCC         ;CLEAR RITE SECTOR
29 03454 044732         STA     1,RSEC          ;CLEAR READ SECTOR
30 03455 010722         ISZ     TRACC           ;INCREMENT TRACK
31 03456 024721         LDA     1,TRACC         ;GET NEW TRACK ADDRESS
32 03457 044726         STA     1,RTRK          ;GIVE IT TO READ ROUTINE
33 03460 030147         LDA     2,C77           ;GET LAST TRACK VALUE
34 03461 132414         SUB#    1,2,SZR         ;ARE WE THROUGH WITH THE TRACKS
35 03462 000767         JMP     WCPAA           ;NO,CONTINUE
36 03463 126400         SUB     1,1             ;CLEAR AC1
37 03464 044713         STA     1,TRACC         ;CLEAR WRITE TRACK
38 03465 044720         STA     1,RTRK          ;CLEAR READ TRACK
39 03466 002072 WCPA3:  JMP     @RTWCP          ;RETURN TO CALLER
40
41 03467 006074 CPERR:  JSR     @IERME          ;PRINT READ COMPARE ERROR
42 03470 063716         RDCER
43 03471 024257         LDA     1,ERTS1         ;WAS THER AN ERROR BEFORE
44 03472 125004         MOV     1,1,SZR         ;IF 0 NO
45 03473 000415         JMP     WCPA4           ;GO RETURN TO CALLER
46 03474 014020         DSZ     BUFPT           ;CORRECT POINTER TO CORRECT ADDR
47 03475 020020         LDA     0,BUFPT         ;GET BUFERR ERROR ADDRESSS
48 03476 024707         LDA     1,RTRK          ;GET TRACK ADDRESS
49 03477 030707         LDA     2,RSEC          ;GET SECTOR ADDRESS
50 03500 006104         JSR     @IERR           ;REPROT ERROR AC0=FAILING ADDRES
51                                              ;AC1=TRACK ADDRESS
52                                              ;AC2=SECTOR ADDRESS
53 03501 026020         LDA     1,@BUFPT        ;AC1=GOOD DATA
54 03502 014021         DSZ     CBUFP           ;CORRECT THE BAD DATA POINTER
55 03503 020021         LDA     0,CBUFP         ;AC0=BAD DATA ADDRESS
56 03504 152400         SUB     2,2             ;CLEAR AC2
57 03505 052073         STA     2,@IITR         ;CLEAR THE ERROR COUNTER
58 03506 032021         LDA     2,@CBUFP                ;AC2=BAD DATA
59 03507 006104         JSR     @IERR           ;REPORT THE ERROR
60
```

```
 0060 FPYUI
01 03510 002072 WCPA4:  JMP     @RTWCP          ;END THE TEST AND LOOP IF NECESS
02
03
04                      ;***************************************************************
05                      ;           TTY   SERVICE   ROUTINE
06                      ;***************************************************************
07
08 03511 054223 SERVS:  STA     3,RET           ;SAVE RETURN ADDRESS
09 03512 006220         JSR     @IINP           ;GO TO SEVICE THE TTY
10 03513 024122         LDA     1,C23           ;GET CONTROL S CONSTANT
11 03514 166435         SUBZ#   3,1,SNR         ;IS IT CONTROL S?
12 03515 000420         JMP     S1              ;YES GO SERVICE
13 03516 024123         LDA     1,C17           ;GET CONTROL O
14 03517 166435         SUBZ#   3,1,SNR         ;IS IT CONTROL O?
15 03520 000413         JMP     O1              ;YES GO SERVICE IT
16 03521 024233         LDA     1,CC            ;GET A "C"
17 03522 160415         SUB#    3,1,SNR         ;SEE IF ITS A I
18 03523 002252         JMP     @ICOMS          ;GO TO COMMAND STRING INTERPRETE
19 03524 024231         LDA     1,CF            ;GET CONTROL F
20 03525 166415         SUB#    3,1,SNR         ;IS IT CONTROL F
21 03526 002232         JMP     @IFUNL  ;GO TO FUNCTION INTERPRETER
22 03527 024303         LDA     1,CJ            ;GET A J
23 03530 166415         SUB#    3,1,SNR         ;LOOP ON A PARTICULAR TEST
24 03531 006277         JSR     @ITELP          ;YES
25 03532 002223         JMP     @RET            ;RETURN TO CALLER
26 03533 006221 O1:     JSR     @IIODT          ;GO TO ODT ROUTINE
27 03534 002223         JMP     @RET            ;RETURN TO CALLER
28 03535 006222 S1:     JSR     @ISUM           ;GO SUMMERIZE OR ERROR IF NOT PR
29 03536 002223         JMP     @RET            ;RETURN TO CALLER
30
31                      ;***************************************************************
32                      ;               CORRECT DRIVE SETUP ROUTINE
33                      ;***************************************************************
34
35 03537 054223 DRI:    STA     3,RET           ;SAVE RETURN ADDRESS
36 03540 030116         LDA     2,FLAGS         ;GET THE FLAGS TO FIND OUT WHAT
37 03541 024157         LDA     1,BIT0          ;GET DRIVE BIT POINTER
38 03542 133414         AND#    1,2,SZR         ;IS IT DRIVE 1?
39 03543 123000         ADD     1,0             ;MAKE FUNCTION FOR DRIVE ONE
40 03544 002223         JMP     @RET            ;RETURN TO CALLER
41
42                      ;***************************************************************
43                      ;                   DOAS HANDLER
44                      ;***************************************************************
45
46 03545 054066 DODA:   STA     3,BACK          ;SAVE RETURN ADDRESS
47 03546 024261         LDA     1,UNITA         ;GET UNIT #
48 03547 036066         LDA     3,@BACK         ;GET FUNCTION
49 03550 137000         ADD     1,3             ;CORRECT IT FOR DEVICE
50 03551 054402         STA     3,FUNS          ;SAVE IT
51 03552 010066         ISZ     BACK            ;CORRECT RETURN POINTER
52 03553 000000 FUNS:   0                       ;FUNCTION TO BE EXECUTED
53 03554 002066         JMP     @BACK           ;RETURN TO CALLER
```

```
0001 FPYDI
01
02
03                ;*****************************************************************
04                ;     THIS ROUTINE CHECKS ALL ERROR BITS IN THE STATUS REGISTER
05                ;*****************************************************************
06
07 03555 054007 ERRCK:  STA     3,RTNH          ;SAVE RETURN ADDRESS
08 03556 036073         LDA     3,@IITR         ;WAS THERE ANY ERRORS
09 03557 175004         MOV     3,3,SZR         ;NO
10 03560 000470         JMP     ERRC8           ;DONIT CHECK STATUS
11 03561 101212         MOVR#   0,0,SZC         ;IS ERROR SET
12 03562 000404         JMP     ERRNC           ;ERROR IS SET
13 03563 101112         MOVL#   0,0,SZC         ;IS FLOPPY READY?
14 03564 000465         JMP     FPYNRD          ;FLOPPY IS NOT READY
15 03565 000405         JMP     ERRC1           ;GO CHECK WRITE FAULT
16 03566 006074 ERRNC:  JSR     @IERME          ;REPORT ERROR BIT SET
17 03567 004665         SETER
18 03570 101264         ISZ     ERBTC           ;INCREMENT ERROR BIT COUNT
19 03571 060077         NIO     CPU             ;NO OPERATION
20 03572 024175 ERRC1:  LDA     1,BIT14         ;GET READY TO CHECK WRITE FAULT
21 03573 107415         AND#    0,1,SNR         ;IS WRITE FAULT SET?
22 03574 000405         JMP     ERRC2           ;NO ,GO CHECK DATA CHANNEL LATE
23 03575 006074         JSR     @IERME          ;REPORT WRITE FAULT SET
24 03576 004741         SETWF
25 03577 101266         ISZ     WRFLC           ;INCREMENT WRITE FAULT COUNT
26 03600 060077         NIO     CPU             ;NO OPERATION
27 03601 024174 ERRC2:  LDA     1,BIT13         ;GET READY TO TEST DATA CHANNEL
28 03602 107415         AND#    0,1,SNR         ;IS DATA CHANNEL SET?
29 03603 000405         JMP     ERRC3           ;NO ,GO CHECK CHECK WORD ERROR
30 03604 006074         JSR     @IERME          ;REPORT DATA CHANNEL LATE
31 03605 004702         SETDC
32 03606 101267         ISZ     DCLAT           ;INCREMENT DATA CHANNEL LATE COU
33 03607 060077         NIO     CPU             ;NO OPERATION
34 03610 024173 ERRC3:  LDA     1,BIT12         ;GET READY TO TEST CHECK WORD ER
35 03611 107415         AND#    0,1,SNR         ;CHECK WORD ERROR SET?
36 03612 000405         JMP     ERRC4           ;NO ,GO TEST SECTOR ERROR
37 03613 006074         JSR     @IERME          ;GO REPORT CHECK WORD ERROR
38 03614 004753         SETCW
39 03615 101270         ISZ     CKWDC           ;INC CHECK WORD COUNT
40 03616 060077         NIO     CPU             ;NO OPERATION
41 03617 024172 ERRC4:  LDA     1,BIT11         ;GET READY TO TEST SECTOR ERROR
42 03620 107415         AND#    0,1,SNR         ;IS SECTOR ERROR SET?
43 03621 000405         JMP     ERRC5           ;GO CHECK ILLEAGLE FUNCTION
44 03622 006074         JSR     @IERME          ;GO REPORT SECTOR ERROR SET
45 03623 005020         SETSE
46 03624 101114         ISZ     SECEC           ;INCREMENT SECTOR ERROR COUNT
47 03625 060077         NIO     CPU             ;NO OPERATION
48 03626 024171 ERRC5:  LDA     1,BIT10         ;GET READY TO TEST ILLEAGLE FUNC
49 03627 107415         AND#    0,1,SNR         ;IS ILL FUNC SET
50 03630 000405         JMP     ERRC6           ;NO SEE IF DEVICE IS WRITE PROTE
51 03631 006074         JSR     @IERME          ;REPORT ILLEAGLE FUNCTION SET
52 03632 005024         SETIL
53 03633 101265         ISZ     ILLFC           ;INCREMENT ILL FUNC COUNT
54 03634 060077         NIO     CPU             ;NO OPERATION
55 03635 024165 ERRC6:  LDA     1,BIT6          ;GET READY TO SEE IF WRITE PROTE
56 03636 107415         AND#    0,1,SNR         ;IS DEVICE WRITE PROTECTED?
57 03637 000406         JMP     ERRC7           ;NO GO RETURN TO CALLER
58 03640 036073         LDA     1,@IITR         ;GET ERROR COUNT
59 03641 125005         MOV     1,1,SNR         ;HAVE WE HAD ANY ERRORS?
60 03642 000403         JMP     ERRC7           ;NO DONT TELL HIM HE IS WRITE PR
```

```
0002 FPYDI
01 03643 006110         JSR     @IMESS          ;YES TELL HIM HE IS WRITE PROTEC
02 03644 005032         WKPRO
03 03645 024257 ERRC7:  LDA     1,ERTS1         ;GET ERROR COUNTER
04 03646 125004         MOV     1,1,SZR         ;ANY ERRORS
05 03647 006104         JSR     @IERR           ;PRINT ERROR MESSAGE
06 03650 002067 ERRC8:  JMP     @RTNH           ;RETURN TO THE CALLER
07 03651 006074 FPYNRD: JSR     @IERME          ;REPORT FLOPPY NOT READY
08 03652 004267         FPYRD
09 03653 006104         JSR     @IERR           ;FLOPPY IS NOT READY
10 03654 101115         ISZ     FPYRC           ;INC FLOPPY NOT READY COUNT
11 03655 060077         NIO     CPU             ;NO OPERATION
12 03656 000714         JMP     ERRC1           ;CONTINUE CHECKING ERRORS
13
14
15 03657 054007 SEQER:  STA     3,RTNH          ;SAVE RETURN ADDRESS
16 03660 006110         JSR     @IMESS          ;PROGRAMS ARE RUNNING OUT OF SEQ
17 03661 004643         SEAER
18 03662 002067         JMP     @RTNH           ;RETURN TO CALLER
19
20                ;*****************************************************************
21                ;          ERROR MESSAGE ROUTINE SO THEY DONIT KEEP PRINTIN
22                ;*****************************************************************
23
24 03663 054066 ERMES:  STA     3,BACK          ;SAVE RETURN POINTER
25 03664 054204         STA     3,AC3?          ;SAVE AC3
26 03665 024257         LDA     1,ERTS1         ;GET ERROR COUNTER
27 03666 125004         MOV     1,1,SZR         ;ANY ERRORS?
28 03667 000405         JMP     NOPRT           ;YES SO DONIT PRINT
29 03670 020066         LDA     1,@BACK         ;GET MESSAGE ADDRESS
30 03671 044402         STA     1,MESSR         ;STORE IT FO PRINT OUT
31 03672 006110         JSR     @IMESS          ;PRINT THE MESSAGE
32 03673 000000 MESSR:  0                       ;MESSAGE ADDRESS STORAGE
33 03674 101066 NOPRT:  ISZ     BACK            ;UPDATE RETURN POINTER
34 03675 002066         JMP     @BACK           ;RETURN TO CALLER
```

```
10063 FPYDI
01
02                ;*****************************************************************
03                ;          ASCII MESSAGES TO BE PRINTED IN CASE OF ERRORS
04                ;*****************************************************************
05
06 03676 005215 HDHMC:  .TXTE /<15><12>HEAD HOME NOT BEING DETECTED/
07       142510
08       042101
09       044240
10       046717
11       120305
12       147516
13       120324
14       142502
15       047311
16       120107
17       142504
18       142724
19       152303
20       042305
21       000000
22 03716 005215 RDCER:  .TXTE /<15><12>READ COMPARE ERROR/
23       142722
24       042101
25       141640
26       046717
27       040520
28       142722
29       142640
30       151322
31       151317
32       000000
33 03731 005215 TYFM:   .TXTE /<15><12>TYPE DESIRED FUNCTION (FUNC(S OR C OR)(CR
34       054724
35       142520
36       042240
37       051705
38       151311
39       042305
40       143240
41       047125
42       152303
43       147711
44       120116
45       143050
46       047125
47       024303
48       120123
49       151317
50       141640
51       147640
52       124722
53       141450
54       124722
55       005215
56       000000
57 03761 005215 NUMU:   .TXTE /<15><12>TYPE DEVICE # (CR)   /
58       054724
59       142520
60       042240
```

```
0064 FPYDI
01       053305
02       141711
03       120305
04       120243
05       141450
06       124722
07       120240
08       000240
09 03775 005215 ENCMS:  .TXTE /<15><12>ENTER COMMAND STRING<15><12>COMMAND,UNIT#
10       047305
11       142724
12       120322
13       147703
14       046515
15       047101
16       120104
17       152123
18       144722
19       043516
20       005215
21       147703
22       046515
23       047101
24       120104
25       047125
26       152311
27       120243
28       040520
29       152324
30       151305
31       120116
32       151324
33       141501
34       121513
35       051654
36       141705
37       147724
38       121722
39       141450
40       124722
41       005215
42       000000
43 04037 005215 HDL20:  .TXTE /<15><12>HEAD LOAD NOT SET IN 20 SETTLE TIMES/
44       142510
45       042101
46       140240
47       040717
48       120104
49       147516
50       120324
51       142523
52       120324
53       047311
54       131240
55       120060
56       142523
57       152324
58       142714
59       152240
60       046711
```

```
 0065 FPYDI
01        051705
02        000000
03 04063  005215 HLSDC:  .TXTE /<15><12>HEAD LOAD SET TO EARLY AFTER DRIVE CHANGE
04        142510
05        042101
06        146240
07        040717
08        120104
09        142523
10        120324
11        147724
12        142640
13        151101
14        054714
15        040640
16        152306
17        151305
18        042240
19        144722
20        142526
21        141640
22        040510
23        043516
24        000305
25 04111  005215 CWPR:   .TXTE /<15><12>CAN'T WRITE, DEVICE WRITE PROTECTED/
26        040703
27        023516
28        120324
29        151327
30        152311
31        120305
32        042240
33        053305
34        141711
35        120305
36        151327
37        152311
38        120305
39        151120
40        152317
41        141705
42        142724
43        000104
44 04134  005215 HDWLD:  .TXTE /<15><12>HEAD WON'T LOAD/
45        142510
46        042101
47        153640
48        047317
49        152047
50        146240
51        040717
52        000104
53 04145  005215 HDLDF:  .TXTE /<15><12>HEAD LOAD BIT ALWAYS READS AS SET/
54        142510
55        042101
56        146240
57        040717
58        120104
59        144502
60        120324
```

```
 0066 FPYDI
01        146101
02        040727
03        051531
04        151240
05        040705
06        051504
07        040640
08        120123
09        142523
10        000324
11 04167  005215 CNFSC:  .TXTE /<15><12>CAN NOT GET A COMPARE ON SECTOR ADDRESS/
12        040703
13        120116
14        147516
15        120324
16        142507
17        120324
18        120101
19        147703
20        050115
21        151101
22        120305
23        047317
24        051640
25        141705
26        147724
27        120322
28        042101
29        151104
30        051705
31        000123
32 04214  005215 LSDK:   .TXTE /<15><12>LOAD SCRATCH DISKETTE TYPE (CR)/
33        147714
34        042101
35        051640
36        151303
37        152101
38        044303
39        042240
40        051711
41        142513
42        152324
43        120305
44        054724
45        142520
46        024240
47        151303
48        000251
49 04235  005215 SECWS:  .TXTE /<15><12>SECTOR ERROR WON'T SET/
50        142523
51        152303
52        151317
53        142640
54        151322
55        151317
56        153640
57        047317
58        152047
59        051640
60        152305
```

```
 0067 FPYDI
01        000000
02 04252  005215 PLNG:   .TXTE /<15><12>PROGRAM LOAD PROM ERROR/
03        151120
04        043717
05        040722
06        120115
07        147714
08        042101
09        050240
10        147722
11        120115
12        151305
13        147722
14        000322
15 04267  005215 FPYRD:  .TXTE /<15><12>FLOPPY IS NOT READY /
16        146306
17        050317
18        054520
19        144640
20        120123
21        147516
22        120324
23        142722
24        042101
25        120131
26        000000
27 04303  005215 DOBNG:  .TXTE /<15><12>DOB ISSUED BUT NOT READING CORRECT INFO W
28        147504
29        120102
30        051711
31        052523
32        042305
33        041240
34        152125
35        047240
36        152317
37        151240
38        040705
39        144504
40        043516
41        141640
42        151317
43        142722
44        152303
45        144640
46        143116
47        120317
48        144727
49        044324
50        042240
51        041311
52        000000
53 04335  005215 BADUR:  .TXTE /<15><12>BUFFER ADDRESS NOT INCREMENTING PROPERLY
54        052502
55        143306
56        151305
57        040640
58        042104
59        142722
60        051523
```

```
 0068 FPYDI
01        047240
02        152317
03        144640
04        141516
05        142722
06        142515
07        152116
08        047311
09        120107
10        151120
11        050317
12        151305
13        054714
14        024240
15        142722
16        042101
17        000251
18 04366  005215 FLONG:  .TXTE /<15><12>FLOPPY NOT WORKING,IF DOOS DISK IN DRIVE
19        146306
20        050317
21        054520
22        047240
23        152317
24        153640
25        151317
26        144513
27        043516
28        144654
29        120306
30        042104
31        051717
32        042240
33        051711
34        120113
35        047311
36        042240
37        144722
38        142526
39        030240
40        151240
41        050305
42        040714
43        142703
44 04420  005215 <15><12>WITH SCRATCH DISK AND TYPE (CR) WHEN READY TO CONTINUE/
45        144727
46        044324
47        051640
48        151303
49        152101
50        044303
51        042240
52        051711
53        120113
54        047101
55        120104
56        054724
57        142520
58        024240
59        151303
60        120251
```

```
 0069 FPYDI
01          044327
02          047305
03          151240
04          040705
05          054504
06          152240
07          120317
08          147703
09          152116
10          047311
11          142525
12          000000
13 04455 005215 DKMES:  .TXTE /<15><12>TESTING DISK # /
14          142724
15          152123
16          047311
17          120107
18          144504
19          045523
20          121640
21          000240
22 04466 120240 DRMES:  .TXTE /  DRIVE # /
23          151104
24          053311
25          120305
26          120243
27          000000
28 04474 005215 DOBIB:  .TXTE /<15><12>DOB IS NOT LOADING CORRECT ADDRESS OR DIB
29          147504
30          120102
31          051711
32          047240
33          152317
34          146240
35          040717
36          144504
37          043516
38          141640
39          151317
40          142722
41          152303
42          040640
43          042104
44          142722
45          051523
46          147640
47          124322
48          144504
49          120102
50          147516
51          120324
52          142722
53          042101
54          047311
55          120107
56          152311
57          041240
58          141501
59          000113
60 04534 005215 BADDI:  .TXTE /<15><12>BUFFER ADDRESS NOT INCREMENTING PROPERLY
```

```
 0070 FPYDI
01          052502
02          143306
03          151305
04          040640
05          042104
06          142722
07          051523
08          047240
09          152317
10          144640
11          141516
12          142722
13          142515
14          152116
15          047311
16          120107
17          151120
18          050317
19          151305
20          054714
21          042440
22          151327
23          152311
24          124705
25          000000
26 04566 005215 DURER:  .TXTE /<15><12> DEVICE IS NOT RESPONDING TO DEVICE #/
27          042240
28          053305
29          141711
30          120305
31          051711
32          047240
33          152317
34          151240
35          051705
36          147520
37          042116
38          047311
39          120107
40          147724
41          042240
42          053305
43          141711
44          120305
45          000243
46 04612 005215 FPYND:  .TXTE /<15><12>FLOPPY DONE NOT SETTING/
47          140306
48          050317
49          054520
50          042240
51          047317
52          120305
53          147516
54          120324
55          142523
56          152324
57          047311
58          000107
59 04627 005215 HDHMS:  .TXTE /<15><12>HEAD HOME ALWAYS SET/
60          142516
```

```
0071 FPYDI
01        042101
02        044240
03        046717
04        120305
05        146101
06        040727
07        051531
08        051640
09        152305
10        000000
11 04643  005215 SEAER:  .TXTE /<15><12>TESTS ARE RUNNING OUT OF SEQUINCE/
12        142724
13        152123
14        120123
15        151161
16        120305
17        052722
18        047116
19        047311
20        120107
21        052717
22        120324
23        143317
24        051640
25        150705
26        144525
27        141516
28        000305
29 04665  005215 SETER:  .TXTE /<15><12>ERROR IS SET IN STATUS /
30        151305
31        147722
32        120322
33        051711
34        051640
35        152305
36        144640
37        120116
38        152123
39        152101
40        051525
41        000240
42 04722  005215 SETDC:  .TXTE /<15><12>DATA CHANNEL LATE ERROR /
43        040504
44        040724
45        141640
46        040510
47        047116
48        146305
49        146240
50        152101
51        120305
52        151305
53        147722
54        120322
55        000000
56 04770  005215 DIBNG:  .TXTE /<15><12> DIB FUNCTION IS NOT WORKING <15><12>/
57        042240
58        041311
59        143240
60        047125
```

```
0072 FPYDI
01        152303
02        147711
03        120116
04        051711
05        047240
06        152317
07        153640
08        151317
09        144513
10        043516
11        100640
12        000012
13 04741  005215 SETWF:  .TXTE /<15><12>WRITE FAULT SET /
14        151327
15        152311
16        120305
17        040706
18        146125
19        120324
20        142523
21        120324
22        000000
23 04753  005215 SETCW:  .TXTE /<15><12>CHECK WORD ERROR /
24        044303
25        141705
26        120113
27        147727
28        042322
29        142640
30        151322
31        151317
32        000240
33 04765  005215 RDPAF:  .TXTE /<15><12>PREAMBLE READ ERROR  TRK # /
34        151120
35        040705
36        041115
37        142714
38        151240
39        040705
40        120104
41        151305
42        147722
43        120322
44        152240
45        043722
46        121640
47        000240
48 05004  005215 SETIL:  .TXTE /<15><12>ILLEGLE FUNCTION SET /
49        146311
50        142714
51        146107
52        120305
53        052706
54        141516
55        144724
56        047317
57        051640
58        152305
59        000240
60 05020  005215 SETSE:  .TXTE /<15><12>SECTOR ERROR SET /
```

```
 0073 FPYDI
01          142523
02          152303
03          151317
04          142640
05          151322
06          151317
07          051640
08          152305
09          000240
10 05032 005215 WRPRO:  .TXTE /<15><12>DEVICE IS WRITE PROTECTED /
11          142504
12          144526
13          142703
14          144640
15          120123
16          151327
17          152311
18          120305
19          151120
20          152317
21          141705
22          142724
23          120104
24          000000
25 05051 005215 ENDFP:  .TXTE /<15><12>**END OF PASS   /
26          125252
27          047305
28          120104
29          143317
30          050240
31          051501
32          120123
33          120240
34          000000
35 05063 005215 HEDOF:  .TXTE /<15><12>HEAD IS NOT MOVING OR DISK UNFORMATED/
36          142510
37          042101
38          144640
39          120123
40          147516
41          120324
42          147515
43          144526
44          043516
45          147640
46          120322
47          144524
48          045523
49          052640
50          143116
51          151317
52          040515
53          142724
54          000104
55 05107 005215 HNLDS:  .TXTE /<15><12>HEAD LOAD BIT NOT SETTING AFTER 6 STARTS/
56          142510
57          042101
58          146240
59          040717
60          120104
```

```
 0074 FPYDI
01          144502
02          120324
03          147516
04          120324
05          142523
06          152324
07          047311
08          120107
09          143101
10          142724
11          120322
12          120066
13          152123
14          151101
15          051724
16          000000
17 05135 005215 IFWAS:  .TXTE /<15><12>ILLEAGLE FUNC WAS /
18          146311
19          142714
20          043501
21          142714
22          143242
23          047125
24          120303
25          040727
26          120123
27          000000
28 05150 005215 WHDR:   .TXTE /<15><12>TYPE DRIVE # TO BE TESTED (CR) /
29          054724
30          142520
31          042240
32          144722
33          142526
34          121640
35          152242
36          120317
37          142502
38          152240
39          051705
40          142724
41          120104
42          141450
43          124722
44          000240
45 05171 005215 HMDR:   .TXTE /<15><12>TYPE # OF DRIVES TO BE TESTED (CR) /
46          054724
47          142520
48          121640
49          147640
50          120306
51          151104
52          053311
53          051705
54          152240
55          120317
56          142502
57          152240
58          051705
59          142724
60          120104
```

```
 0075 FPYDI
01        141450
02        124722
03        000240
04 05214  005215 MKTT1:   .TXTE /<15><12>ALL DEVICES MASKED OUT AND TTY INTERUPTED
05        146101
06        120314
07        142504
08        144526
09        142703
10        120123
11        040515
12        045523
13        042305
14        147640
15        152125
16        040640
17        042116
18        152240
19        054724
20        144640
21        152116
22        151305
23        050125
24        142724
25        000104
26 05242  005215 MKNG1:   .TXTE /<15><12>FLOPPY STILL INTERUPTED AFTER IT WAS MASK
27        146306
28        050317
29        054520
30        051640
31        144724
32        146314
33        144640
34        152116
35        151305
36        050125
37        142724
38        120104
39        143101
40        142724
41        120322
42        152311
43        153640
44        051501
45        046640
46        051501
47        142513
48        120104
49        052717
50        000324
51 05273  005215 HLSTS1:  .TXTE /<15><12>HEAD LOADED SET BEFORE 6 STARTS WHERE ISS
52        142510
53        042101
54        146240
55        040717
56        142504
57        120104
58        142523
59        120324
60        142502
```

```
 0076 FPYDI
01        147706
02        142722
03        033240
04        051640
05        040724
06        152322
07        120123
08        044327
09        151305
10        120305
11        051711
12        052523
13        042305
14        000000
15 05323  005215 SUMMD1:  .TXTE /<15><12><12>THIS IS A SUMMARY OF ERRORS DETECTED
16        152012
17        144510
18        120123
19        051711
20        040640
21        051640
22        046525
23        040515
24        054722
25        147640
26        120306
27        151305
28        147722
29        051722
30        042240
31        152305
32        141705
33        142724
34        120104
35        044327
36        146311
37        120305
38        052722
39        047116
40        047311
41        120107
42        146306
43        050317
44        054520
45        000000
46 05362  005215 TETNM1:  .TXTE /<15><12>TEST # /
47        142724
48        152123
49        121640
50        000240
51                ;*****************************************************************
52 05367  054200 SUMME1:  STA     3,COMRE         ;SAVE RETURN ADDRESS
53 05370  000110         JSR     @IMESS          ;PRINT MESSAGES
54 05371  005323         SUMMD
55 05372  000110         JSR     @IMESS
56 05373  004267         FPYRD
57 05374  024115         LDA     1,FPYRC
58 05375  000212         JSR     @IZOC?T
59 05376  000110         JSR     @IMESS
60 05377  004665         SETER
```

```
 0077 FPYDI
01 05400 024264          LDA     1,ERBTC
02 05401 006212          JSR     @IZOC?T
03 05402 006110          JSR     @IMESS
04 05403 004741          SETWF
05 05404 024266          LDA     1,WRFLC
06 05405 006212          JSR     @IZOC?T
07 05406 006110          JSR     @IMESS
08 05407 004702          SETDC
09 05410 024267          LDA     1,DCLAT
10 05411 006212          JSR     @IZOC?T
11 05412 006110          JSR     @IMESS
12 05413 004753          SETCW
13 05414 024270          LDA     1,CKWDC
14 05415 006212          JSR     @IZOC?T
15 05416 006110          JSR     @IMESS
16 05417 005004          SETIL
17 05420 024265          LDA     1,ILLFC
18 05421 006212          JSR     @IZOC?T
19 05422 002260          JMP     @COMRE
20                 ;*************************************************************
21
22 05423 054225 SETEL:   STA     3,REET  ;SAVE RETURN ADDRESS
23 05424 102400          SUB     0,0             ;CLEAR AC0
24 05425 040224          STA     0,TIMIT         ;CLEAR TIMMER
25 05426 020141          LDA     0,SETTLE        ;GET SETTLE COMMAND
26 05427 006077          JSR     @IDRI           ;SETUP FOR CORRECT DRIVE
27 05430 006111          JSR     @IDOOA          ;ISSUE FUNCTION
28 05431 061100          DOAS    0,FPY           ;ISSUE SETTLE TO FLOPPY
29 05432 040204          STA     0,AC3?          ;SAVE FUNCTION
30 05433 006105          JSR     @ITIME          ;WAIT FOR DONE
31 05434 006111          JSR     @IDOOA          ;ISSUE FUNCTION
32 05435 060600          DIAC    0,FPY           ;GET STATUS
33 05436 101132          MOVZL#  0,0,SZC         ;IS THE FLOPPY READY
34 05437 000405          JMP     SET2            ;NO WAIT FOR READY
35 05440 030161          LDA     2,BIT2          ;GET BIT TO TEST FOR HEAD LOADED
36 05441 113415          AND#    0,2,SNR         ;IS HEAD LOADED?
37 05442 000402          JMP     .+2             ;NO,INCREMENT TIMMER
38 05443 002225          JMP     @REET           ;RETURN TO CALLER
39 05444 010224 SET2:    ISZ     TIMIT           ;INCREMENT TIMMER
40 05445 000761          JMP     SETEL+3         ;KEEP SENDING SETTLE
41 05446 032073          LDA     2,@IITR         ;SEE IF ANY ERRORS
42 05447 151004          MOV     2,2,SZR         ;ANY ERRORS?
43 05450 000414          JMP     SET4            ;YES EXIT
44 05451 101132          MOVZL#  0,0,SZC         ;IS READY OK
45 05452 000410          JMP     SET3            ;NO
46 05453 030161          LDA     2,BIT2          ;GET THE HEAD LOAD BIT
47 05454 143414          AND#    2,0,SZR         ;IS THE HEAD LOADED
48 05455 002225          JMP     @REET           ;RETURN TO CALLER
49 05456 006074          JSR     @IERME          ;PRINT HEADS WON'T LOAD
50 05457 004134          MDWLD
51 05460 006106          JSR     @IERCK          ;SEE IF ANY ERRORS
52 05461 000403          JMP     SET4            ;PRINT AC'S AND RETURN
53 05462 006074 SET3:    JSR     @IERME          ;PRINT FLOPPY NOT READY
54 05463 004267          FPYRD
55 05464 006104 SET4:    JSR     @IERR           ;HEAD WON'T LOAD
56                                               ;AC3=FUNCTION
57                                               ;AC0=STATUS
58 05465 002225          JMP     @REET           ;RETURN TO CALLER
59
60
```

```
 0078 FPYDI
01                 ;FILENAME=      DIAGSUP
02
03                 ;THIS PACKAGE IS USED TO SETUP THE SUBTESTS FOR THE PROGRAMS
04                 ;THAT RUN ON NOVA,ECLIPSE AND MICRONOVA TYPE COMPUTERS.
05
06                 ;"ENT?R" ENTRY IS USED TO BEGIN A NEW TEST.
07                 ;"CYC?E" ENTRY IS USED TO END A TEST.
08                 ;"ERR?" ENTRY IS USED ON SEEING AN ERROR. WHEN THE FIRST ERROR
09                 ;OF A TEST IS SEEN A MESSAGE IS OUTPUT. THE PRECENT ERROR RATE
10                 ;CAN BE OBTAINED BY SETTING BIT 3 OF SWREG.
11                 ;
12                 ;NOTE:  THIS PACKAGE REQUIRES THE TTYIO AND SWPAK PAKAGES
13                 ;IN ORDER TO RUN, IT ALSO NEEDS THE FOLLOWING VARIABLES IN
14                 ;PAGE ZERO:
15                 ;
16                 ;AT LOC 201      HELP:   0       TEST STARTING ADDRESS
17                 ;AT LOC 203      PASS:   0       PROGRAM PASS COUNTER
18                 ;ALSO            AC3?:   0       STORAGE FOR AC3
19                 ;                SWREG:  0       NEEDED BY SWPAK
20                 ;                IINP?:  INP?J   POINTERS
21                 ;                         OR     TO
22                 ;                        INP?I
23                 ;                ICRL?F: CRL?F   TTYIO
24                 ;                IPOC?S: POC?S   PACKAGE
25                 ;                IPOC?T: POC?T   ROUTINES
26                 ;                IZOC?T: ZOC?T
27                 ;                IENT?R: ENT?R   POINTERS TO ROUTINES IN
28                 ;                ICYC?E: CYC?E   THIS PACKAGE
29                 ;                IERR?:  ERR?
30
31
32                 ;THE ENT?R ROUTINE STARTS EACH SUBTEST BY STORING ITS
33                 ;STARTING ADDRESS IN PAGE ZERO, SETTING ITS ITERATION
34                 ;COUNTER (ON THE FIRST PASS OF THE PROGRAM THE ITERATION
35                 ;COUNTER IS ALWAYS FORCED TO 1).  ENT?R THEN CLEARS
36                 ;THE ERROR COUNT AND STARTS THE TEST.
37                 ;
38                 ;THE CALL TO ENT?R MUST BE FOLLOWED BY A WORD CONTAINING
39                 ;THE ITERATION COUNT FOR THIS TEST.
40                 ;
41                 ;  EXAMPLE:
42                 ;
43                 ;                JSR     @IENT?R
44                 ;                10.
45                 ;
46                 ;THIS WILL START A TEST WITH ITS ITERATION COUNT SET TO 10.
47                 ;IENT?R IS A PAGE ZERO POINTER TO ENT?R
48
49 05460 175400 ENT?R:   INC     3,3             ;GET THE TEST STARTING ADDRESS
50 05467 054522          STA     3,LOO?R         ;STORE IN RETURN LOCATION
51 05470 054202          STA     3,HELP          ;STORE IN PAGE ZERO
52 05471 035777          LDA     3,-1,3          ;GET TEST ITERATION COUNT
53 05472 054520          STA     3,ITR?          ;SAVE IT
54 05473 054520          STA     3,ITR?T         ;SET TEST ITERATION COUNTER
55 05474 170460          SUBC    3,3
56 05475 054517          STA     3,ITR?R         ;CLEAR ERROR SWITCH
57 05476 054517          STA     3,ITR?C         ;CLEAR ERROR COUNTER
58 05477 034203          LDA     3,PASS          ;GET THE PROGRAM PASS COUNT
59 05500 175004          MOV     3,3,SZR         ;IS THIS THE FIRST PASS
60 05501 002510          JMP     @LOO?R          ;NO EXIT
```

```
 0079 FPYDI
01 05502 176520          SUBZL   3,3             ;YES FORCE ITERATION
02 05503 054507          STA     3,ITR?          ;COUNT TO 1
03 05504 054507          STA     3,ITR?T
04 05505 002504          JMP     @LOO?R          ;EXIT TO TEST
05                ;THE CYC?E ROUTINE IS USED TO END A SUBTEST IT HANDLES
06                ;THE ITERATION OF THE SUBTEST AND THE PRINTING OF THE
07                ;% ERROR RATE IF BIT 3 OF SWREG IS SET.
08                ;
09                ;THE CALL TO CYC?E IS:
10                ;
11                ;                       JSR     @ICYC?E
12                ;
13                ;WHERE ICYC?E IS A PAGE ZERO POINTER TO CYC?E
14
15 05506 054510 CYC?E:  STA     3,LOP?ET        ;STORE THE END OF TEST ADDRESS
16 05507 063710          SKPDZ   TTI             ;IF A TTY KEY HAS BEEN HIT
17 05510 006206          JSR     @IINP?          ;GOTO SERVICE ROUTINE
18 05511 014502          DSZ     ITR?T           ;ALL ITERATIONS DONE?
19 05512 002477          JMP     @LOO?R          ;NO, LOOP AGAIN
20 05513 034501          LDA     3,ITR?R
21 05514 175005          MOV     3,3,SNR         ;ANY ERRORS?
22 05515 002501          JMP     @LOP?ET         ;NO GO TO NEXT TEST
23 05516 034474          LDA     3,ITR?          ;YES RESET THE ITERATION COUNTER
24 05517 054474          STA     3,ITR?T
25 05520 034205          LDA     3,SWREG
26 05521 177100          ADDL    3,3
27 05522 177103          ADDL    3,3,SNC         ;IS BIT 3 SET?
28 05523 000424          JMP     CYC?2           ;NO EXIT
29 05524 040473          STA     0,AC0?          ;YES PRINT %ERROR RATE
30 05525 044473          STA     1,AC1?          ;SAVE THE WORLD
31 05526 050473          STA     2,AC2?
32 05527 102560          SUBCL   0,0
33 05530 040472          STA     0,CAR?
34 05531 006207          JSR     @ICRL?F         ;PRINT CARRAGE RETURN
35 05532 102400          SUB     0,0
36 05533 024462          LDA     1,ITR?C
37 05534 030471          LDA     2,K1070
38 05535 004442          JSR     MUL?
39 05536 030454          LDA     2,ITR?
40 05537 004424          JSR     DIV?D
41 05540 006210          JSR     @IPDC?S         ;PRINT FAILURE RATE
42 05541 000045          "%
43 05542 020460          LDA     0,CAR?          ;RESTORE THE WORLD
44 05543 101200          MOVR    0,0
45 05544 020453          LDA     0,AC0?
46 05545 024453          LDA     1,AC1?
47 05546 030453          LDA     2,AC2?
48 05547 176400 CYC?2:   SUB     3,3             ;RESET THE ERROR COUNTER
49 05550 054445          STA     3,ITR?C
50 05551 034443          LDA     3,ITR?R
51 05552 175004          MOV     3,3,SZR
52 05553 034205          LDA     3,SWREG
53 05554 177113          ADDL#   3,3,SNC         ;LOOP ON ERROR?
54 05555 002434          JMP     @LOO?R          ;YES
55 05556 002440          JMP     @LOP?ET         ;NO
56                ;AC1=REM  AC0=(AC0,AC1)/AC2
57 05557 102400 DIV?:    SUB     0,0
58 05560 054443 DIV?0:   STA     3,MDV?5
59 05561 142432          SUBZ#   2,0,SZC
60 05562 000413          JMP     MDV?3
```

```
 0080 FPYDI
01 05563 054440 DIV?D:   STA     3,MDV?5         ;DIVIDE
02 05564 034442          LDA     3,MDV?1
03 05565 125120          MOVZL   1,1
04 05566 101100 MDV?2:   MOVL    0,0
05 05567 142412          SUB#    2,0,SZC
06 05570 142400          SUB     2,0
07 05571 125100          MOVL    1,1
08 05572 175404          INC     3,3,SZR
09 05573 000773          JMP     MDV?2
10 05574 176441          SUBO    3,3,SKP
11 05575 176420 MDV?3:   SUBZ    3,3
12 05576 002425          JMP     @MDV?5
13                ;(AC0,AC1)=AC1*AC2
14 05577 102460 MUL?:    SUBC    0,0
15                ;(AC0,AC1)=AC1*AC2+AC0
16 05600 054423 MUL?A:   STA     3,MDV?5
17 05601 034425          LDA     3,MDV?1
18 05602 125203 MDV?4:   MOVR    1,1,SNC
19 05603 101201          MOVR    0,0,SKP
20 05604 143220          ADDZR   2,0
21 05605 175404          INC     3,3,SZR
22 05606 000774          JMP     MDV?4
23 05607 125260          MOVCR   1,1
24 05610 002413          JMP     @MDV?5
25 05611 000000 LOO?R:   0
26 05612 000000 ITR?:    0
27 05613 000000 ITR?T:   0
28 05614 000000 ITR?R:   0
29 05615 000000 ITR?C:   0
30 05616 000000 LOP?ET:  0
31 05617 000000 AC0?:    0
32 05620 000000 AC1?:    0
33 05621 000000 AC2?:    0
34 05622 000000 CAR?:    0
35 05623 000000 MDV?5:   0
36 05624 000000 .RT?N:   0
37 05625 000144 K1070:   100.
38 05626 177760 MDV?1:   -20
39 05627 001000 C1B?6:   001000
40
41                ;THE ERR? ROUTINE IS USED IN CASES WHEN A SUBTEST FAILS
42                ;THIS ROUTINE KEEPS A COUNT OF THE NUMBER OF TIMES
43                ;THE TEST FAILED AND FOR THE FIRST ERROR IN A TEST IT WILL
44                ;PRINT THE CONTENTS OF THE ACCUMULATORS, THE CARRY
45                ;AND THE PROGRAM COUNTER. THE PROGRAM THEN DOES ONE OF THREE
46                ;THINGS DEPENDING ON THE BIT SETINGS OF SWREG IT WILL
47                ;PRINT A MESSAGE SAYING IT HALTED AND STOP EXECUTION,
48                ;PRINT A MESSAGE SAYING IT IS LOOPING ON THE ERROR AND RESTART
49                ;THE TEST OR IT WILL CONTINUE TO THE NEXT TEST PRINTING NO
50                ;MESSAGE. IF BIT 6 OF SWREG IS SET THE ROUTINE WILL HALT ON
51                ;ANY ERROR ENCOUNTERED.
52                ;
53                ;THE CALL TO ERR? SHOULD BE EXECUTED ONLY IF THE SUBTEST
54                ;FAILS AND SHOULD BE PRECEEDED BY AN INSTRUCTION WHICH STORES
55                ;AC3 IN A PAGE ZERO LOCATION "AC3?" IF THIS IS NOT DONE THE
56                ;PRINTOUT FOR AC3 WILL BE ERRONEOUS.
57                ;
58                ;AN EXAMPLE OF THE CALL IS:
59                ;
60                ;       XXX                     ;INST SKPS IF TEST PASSED
```

```
 0081 FPYDI
01                       /       JMP     .+2             /TEST FAILED
02                       /       JMP     .+3             /TEST PASSED
03                       /       STA     3,AC3?          /STORE AC3 IN AC3?
04                       /       JSR     @IERR?
05                       /
06                       /WHERE IERR? IS A PAGE ZERO POINTER TO ERR?
07
08 05630 054774 ERR?1:   STA     3,.RT?N         /SAVE RETURN ADDRESS
09 05631 063710          SKPDZ   TTI             /IF A TTY KEY HAS BEEN HIT
10 05632 006206          JSR     @IINP?          /GOTO SERVICE ROUTINE
11 05633 010762          ISZ     ITR?C           /INCREMENT ERROR COUNT
12 05634 000401          JMP     .+1             /NO OP
13 05635 040762          STA     0,AC0?          /SAVE THE WORLD
14 05636 044762          STA     1,AC1?
15 05637 050762          STA     2,AC2?
16 05640 102560          SUBCL   0,0
17 05641 040761          STA     0,CAR?
18 05642 034752          LDA     3,ITR?R         /IS THIS THE FIRST ERROR
19 05643 175004          MOV     3,3,SZR         /YES SKIP
20 05644 000445          JMP     ERR?3           /NO RETURN
21 05645 006107          JSR     @IMES?S
22 05646 005720          MEA?DK
23 05647 024753          LDA     1,CAR?
24 05650 006212          JSR     @IZOC?T
25 05651 024746          LDA     1,AC0?          /PRINT THE AC'S, CARRY AND PC
26 05652 006211          JSR     @IPOC?T
27 05653 024745          LDA     1,AC1?
28 05654 006211          JSR     @IPOC?T
29 05655 024744          LDA     1,AC2?
30 05656 006211          JSR     @IPOC?T
31 05657 024204          LDA     1,AC3?
32 05660 006211          JSR     @IPOC?T
33 05661 024743          LDA     1,.RT?N
34 05662 044732          STA     1,ITR?R         /SET ERROR SWITCH
35 05663 006211          JSR     @IPOC?T
36 05664 034205          LDA     3,SWREG
37 05665 030742          LDA     2,C1B?6
38 05666 157400          AND     2,3
39 05667 024733          LDA     1,CAR?          /RESTORE THE WORLD
40 05670 125200          MOVR    1,1
41 05671 020726          LDA     0,AC0?
42 05672 024726          LDA     1,AC1?
43 05673 030726          LDA     2,AC2?
44 05674 175004          MOV     3,3,SZR         /HALT ON ERROR? NO SKIP
45 05675 000407          JMP     ERR?1           /YES GOTO HALT ROUTINE
46 05676 034205 ERR?2:   LDA     3,SWREG
47 05677 177132          ADDZL#  3,3,SZC         /LOOP ON ERROR?
48 05700 002724          JMP     @.RT?N          /NO RETURN
49 05701 006107          JSR     @IMES?S         /YES SAY SO AND RETURN
50 05702 005740          LOO?T
51 05703 002721          JMP     @.RT?N
52 05704 006107 ERR?1:   JSR     @IMES?S         /PRINT HALTED SND STOP
53 05705 005753          STO?P
54 05706 063077          HALT
55 05707 034205          LDA     3,SWREG         /RESTORE AC3 WITH SWREG
56 05710 000766          JMP     ERR?2           /CONTINUE FROM THE ERROR HALT
57 05711 034205 ERR?3:   LDA     3,SWREG         /HALT ON ERROR?
58 05712 030715          LDA     2,C1B?6
59 05713 157400          AND     2,3
60 05714 030705          LDA     2,AC2?
```

```
 0082 FPYDI
01 05715 175004          MOV     3,3,SZR
02 05716 063077          HALT                    /YES STOP EXECUTION
03 05717 002705          JMP     @.RT?N          /NO RETURN TO TEST
04 05720 005215 MEA?DK:  .TXTE |<15><12><15><12><15>
05       005215
06 05722 141615 CK?      AC0     AC1     AC2     AC3     PC <15><12><15>|
07       054722
08       040411
09       030303
10       040411
11       130703
12       040411
13       131303
14       040411
15       031703
16       050011
17       120303
18       005215
19       000215
20 05740 005215 LOO?T:   .TXTE |<15><12>LOOPING ON ERROR<15><12>|
21       147714
22       050317
23       047311
24       120107
25       047317
26       142640
27       151322
28       151317
29       005215
30       000000
31 05753 005215 STO?P:   .TXTE |<15><12>HALTED ON ERROR<15><12>|
32       040510
33       152314
34       042305
35       147640
36       120116
37       151305
38       147722
39       106722
40       000012
41
42
43                       /FILENAME=TTYIO
44
45                       /TELETYPE NON INTERRUPT PACKAGE
46                       /CARRY,AC0,AC1,AC2 SAVED
47
48                       /"MES?S" PRINTS ASCII MESSAGES AS SPECIFIED BY ASSEMBLER
49
50                       /"CRL?F" PRINTS A CARRIAGE RETURN
51
52                       /"POC?T" PRINTS C(1) IN OCTAL
53                       /"ZOC?T" PRINTS C(1) IN OCTAL, LEADING ZEROS SUPPRESSED
54                       /"POE?C" PRINTS C(1) IN DECIMAL, LEADING ZEROS SUPPRESSED,
55                       /THE ABOVE THREE ARE FOLLOWED BY A TAB
56                       /"POC?S" PRINTS C(1) IN DECIMAL, LEADING ZEROS SUPPRESSED,
57                       /FOLLOWED BY THE CHARACTER STORED AT CALLING LOCATION +1.
58                       /PROGRAM RETURNS TO CALLING LOCATION +2.
59
60                       /"TIN?O" ACCEPTS OCTAL, AND
```

```
0083 FPYDI
01                ;"TIN?D" ACCEPTS DECIMAL SINGLE PRECISION SIGNED INTEGERS
02                ;INTO AC1 FROM THE TTI.  LEADING NULLS, TABS,
03                ;AND SPACES ARE IGNORED.  A 16 BIT UNSIGNED INTEGER IS
04                ;FORMED, THEN NEGATED IF A MINUS SIGN IS TYPED.
05                ;EXIT AT CALL+1 IF INPUT ERROR WITH AC0=BAD CHARACTER.
06                ; (NOT A LEGAL DIGIT OR TERMINATING CHARACTER)
07                ;EXIT AT CALL+2 UPON TERMINATING CHARACTER
08                ; WITH AC0=0, 0, 40, 12, 15, 55
09                ; FOR NULL, SPACE, LINE-FEED, CARRIAGE RETURN, COMMA
10                ;THE ABOVE WAIT FOR TTO DONE, THEN TTO IS CLEARED.
11                ;RUBOUT WILL DELETE THE LAST DIGIT TYPED IN 'TIN?D' AND
12                ;'TIN?D'
13
14                ;"CHC?T" PRINTS ASCII CHARACTER IN C(0)R; C(0)L MUST BE 0.
15                ;EXITS CALL +2 IF C(0)R=0; SIMULATES TAB
16
17                ;"TYP?E" PRINTS C(0)R TO THE TTO OR LPT OR BOTH AS PER THE
18                ;SWITCH SELECTION REGISTER 'SWREG'.
19                ;EXITS AT CALL+1.  REPLACE "TYP?E" WITH
20                ;INTERRUPT 'TYP?E' IF DESIRED.
21
22                ;"TPS?P" PRINTS A SPACE AND EXITS AT CALL+1 WITH AC0 = 40
23
24
25                ;
26                ;MES?S ROUTINE
27                ;
28                ;THE CALLING SEQUENCE IS:
29                ;
30                ;       JSR     @!MES?S
31                ;       POINTER TO MESSAGE TO BE PRINTED
32                ;
33
34 05765 054551 MES?S:  STA     3,RTN?A         ;SAVE THE RTN ADDRESS
35 05766 004562         JSR     SAV?E           ;SAVE THE STATE OF MACHINE
36 05767 034547         LDA     3,RTN?A         ;
37 05770 010546         ISZ     RTN?A
38 05771 031400         LDA     2,0,3           ;C(2) POINTS TO MESSAGE
39 05772 024417         LDA     1,P37?7         ;A 8 BIT MASK
40 05773 021000 MES?M:  LDA     0,0,2           ;C(2)=DATA WORD
41 05774 125112         MOVL#   1,1,SZC
42 05775 123701         ANDS    1,0,SKP
43 05776 123401         AND     1,0,SKP         ;C(0)=DATA CHARACTOR RIGHT
44 05777 151400         INC     2,2             ;INC TO NEXT WORD
45 06000 124000         COM     1,1             ;FLIP MASK
46 06001 004414         JSR     CHC?T           ;PRINT
47 06002 000771         JMP     MES?M           ;ANOTHER
48 06003 000402         JMP     .+2
49 06004 004411 PLS?T:  JSR     CHC?T
50 06005 004551 PEX?T:  JSR     RST?R           ;RESTORE THE STATE OF MACHINE
51 06006 002530         JMP     @RTN?A          ;EXIT
52
53                ;
54                ;CHC?T ROUTINE
55                ;
56                ;THE CALLING SEQUENCE IS:
57                ;
58                ;       LDA     0,CHARACTER TO BE PRINTED (RIGHT BYTE)
59                ;       JSR     @!CHC?T
60                ;
```

```
0084 FPYDI
01
02 06007 000000 PSP?:   0
03 06010 000000 SPT?G:  0
04 06011 000377 P377?:  377
05 06012 000011 PC1?:   11
06 06013 000000 CHR?E:  0
07 06014 000000 CAC?0:  0                       ;AC0 SAVE IN CHC?T ROUTINE
08
09 06015 040777 CHC?T:  STA     0,CAC?0         ;SAVE AC0
10 06016 101315         MOVS#   0,0,SNR         ;RETURN +2 IF NULL
11 06017 001401         JMP     1,3
12 06020 175100         MOVL    3,3             ;FOR CARRY SAVE
13 06021 054772         STA     3,CHR?E         ;PRINT C(0) RIGHT
14 06022 034770         LDA     3,PC1?          ;AC3 = 11
15 06023 116415         SUB#    0,3,SNR         ;SKIP IF A TAB IS NOT TO
                                                ;BE SIMULATED
16
17 06024 000403         JMP     CHA?3
18 06025 004556         JSR     TYP?E           ;PRINT IT
19 06026 000407         JMP     CHE?X           ;EXIT
20 06027 004551 CHA?3:  JSR     TPS?P           ;PRINT A SPACE
21 06030 020513         LDA     0,CHR?Z
22 06031 034410         LDA     3,PC7?          ;AC3 = 7
23 06032 163404         AND     3,0,SZR
24 06033 000774         JMP     CHA?3           ;SIMULATE A TAB WITH 1
                                                ;TO 7 SPACES
25
26 06034 040507         STA     0,CHR?Z
27 06035 020757 CHE?X:  LDA     0,CAC?0         ;RESTORE AC0
28 06036 034755         LDA     3,CHR?E         ;RESTORE CRY
29 06037 175200         MOVR    3,3             ;
30 06040 001400         JMP     0,3             ;EXIT
31
32 06041 000007 PC7?:   7
33
34
35                ;
36                ;CRL?F ROUTINE
37                ;
38                ;THE CALLING SEQUENCE IS:
39                ;
40                ;       JSR     @!CRL?F
41                ;
42
43
44 06042 054474 CRL?F:  STA     3,RTN?A         ;SAVE RETURN
45 06043 004505         JSR     SAV?E           ;SAVE THE WORLD
46 06044 020405         LDA     0,K15?
47 06045 004536         JSR     TYP?E           ;PRINT CARRIAGE AND LF
48 06046 020402         LDA     0,K12?
49 06047 000735         JMP     PLS?T           ;GO TO RESTORE THE WORLD
50 06050 000012 K12?:   12
51 06051 000015 K15?:   15
52
53
54                ;
55                ;ZOC?T, POC?T, PDC?S AND PDE?C ROUTINES.
56                ;
57                ;THE CALLING SEQUENCE IS:
58                ;
59                ;       LDA     1,OCTAL NUMBER TO BE PRINTED
60                ;               (LEADING ZEROES SUPPRESSED)
```

```
0085 FPYDI
01                ;       JSR     @IZOC?T
02                ;
03                ;THE CALLING SEQUENCE IS:
04                ;
05                ;       LDA     1,OCTAL NUMBER TO BE PRINTED
06                ;               (LEADING ZEROES NOT SUPPRESSED)
07                ;       JSR     @IPOC?T
08                ;
09                ;THE CALLING SEQUENCE IS:
10                ;
11                ;       LDA     1,DECIMAL NUMBER TO BE PRINTED
12                ;               (LEADING ZEROES SUPPRESSED)
13                ;       JSR     @IPDE?C
14                ;
15                ;THE CALLING SEQUENCE IS:
16                ;
17                ;       LDA     1,DECIMAL NUMBER TO BE PRINTED
18                ;               (LEADING ZEROES SUPPRESSED)
19                ;       JSR     @IPDC?S
20                ;       ALPHA   WHERE ALPHA IS THE CHARACTER PRINTED
21                ;               AFTER THE DECIMAL NUMBER
22
23
24 06052 054464 ZOC?T:  STA     3,RTN?A         ;SAVE THE RTN ADDRESS
25 06053 004475         JSR     SAV?E           ;SAVE THE WORLD
26 06054 102400         SUB     0,0             ;
27 06055 000404         JMP     ZPO?T           ;
28 06056 054460 POC?T:  STA     3,RTN?A         ;SAVE THE RTN ADDRESS
29 06057 004471         JSR     SAV?E           ;SAVE THE WORLD
30 06060 020464         LDA     0,PC670         ;
31 06061 152620 ZPO?T:  SUBZR   2,2             ;PRINT C(1) IN OCTAL
32 06062 034463         LDA     3,PC170         ;C(2)=100000, C(3)=10
33 06063 000416         JMP     PDC?1           ;
34 06064 175400 PDC?S:  INC     3,3             ;UPDATE THE RTN ADDR PNTR
35 06065 054451         STA     3,RTN?A         ;
36 06066 004462         JSR     SAV?E           ;SAVE THE WORLD
37 06067 034447         LDA     3,RTN?A         ;
38 06070 021777         LDA     0,-1,3          ;READ THE CHARACTER TO BE
39                                              ;PRINTED AFTER THE DECIMAL
40                                              ;NUMBER
41 06071 040716         STA     0,PSP?          ;SAVE THE SPECIAL CHAR.
42 06072 102200         ADC     0,0             ;AC0= -1
43 06073 000404         JMP     PDC?2           ;
44 06074 054442 PDE?C:  STA     3,RTN?A         ;SAVE THE RTN ADDRESS
45 06075 004453         JSR     SAV?E           ;SAVE THE WORLD
46 06076 102400         SUB     0,0             ;
47 06077 034751 PDC?2:  LDA     3,K12?          ;C(3)=12
48 06100 030446         LDA     2,DET?B         ;PRINT C(1) IN DECIMAL
49 06101 040707 PDC?1:  STA     0,SPT?G         ;ACTIVATE/DEACTIVATE THE TAG FOR
50                                              ;SPECIAL CHARACTER
51                                              ;BOTH ENTRIES PRINT NUMBER
52 06102 101415         INC#    0,0,SNR         ;SKIP IF AC0 IS NOT -1
53 06103 101400         INC     0,0
54 06104 040543         STA     0,ZSU?P         ;THEN TAB TO NEXT POSITION
55 06105 054442         STA     3,TMP?          ;SAVE AC3
56 06106 034541 DCO?T:  LDA     3,ZSU?P         ;ZEROS SUPPRESS STUF
57 06107 102001 DEC?T:  ADC     0,0,SKP         ;SKIP FIRST TIME HERE PER DIGIT
58 06110 146400         SUB     2,1             ;DIVIDE C(AC1)  BY C(AC2)
59 06111 101405         INC     0,0,SNR         ;
60 06112 151235         MOVZR#  2,2,SNR         ;FOR ZERO SUPPRESS
```

```
0086 FPYDI
01 06113 034431         LDA     3,PC670         ;
02 06114 146453         SUBO#   2,1,SNC         ;SUBTRACT MORE?
03 06115 000773         JMP     .-5             ;YES,GO BACK
04 06116 054531         STA     3,ZSU?P         ;NO,SAVE ZERO SUPPRESS FLAG
05                                              ;C(0)=DIGIT
06 06117 163004         ADD     3,0,SZR         ;MAKE ASCII
07 06120 004675         JSR     CHC?T           ;PRINT
08 06121 034426         LDA     3,TMP?          ;RESTORE AC3
09 06122 102400         SUB     0,0             ;
10 06123 172423         SUBZ    3,2,SNC         ;DIVIDE C(AC2) BY C(AC3)
11
12 06124 000403         JMP     .+3             ;SKIP IF AC3 > AC2
13 06125 101400         INC     0,0             ;AC3 < AC2
14 06126 000775         JMP     .-3             ;SUBTRACT MORE
15 06127 111004         MOV     0,2,SZR         ;WAS IT LAST DIGIT?
16 06130 000756         JMP     DCO?T           ;NO,GET NEXT DIGIT
17 06131 034657         LDA     3,SPT?G         ;YES,CHECK THE SPECIAL CHAR FLAG
18 06132 020660         LDA     0,PC1?1         ;FOLLOW THE PRINTOUT WITH
19 06133 175405         INC     3,3,SNR         ;TAB IF NOT SPCL CHAR FLAG
20 06134 020653         LDA     0,PSP?          ;OTHERWISE FOLLOW WITH THE CHAR
21 06135 000647         JMP     PLS?T           ;TO EXIT
22
23 06136 000000 RTN?A:  0
24 06137 000000 PCR?Y:  0                       ;CRY SAVE LOCATION
25 06140 000000 PAC?0:  0                       ;AC0 SAVE LOCATION
26 06141 000000 PAC?1:  0                       ;AC1 SAVE LOCATION
27 06142 000000 PAC?2:  0                       ;AC2 SAVE LOCATION
28 06143 000000 CHR?Z:  0
29 06144 000060 PC6?0:  60
30 06145 000010 PC1?0:  10
31 06146 023420 DET?B:  10000.
32 06147 000000 TMP?:   0
33
34                ;
35                ; SAV?E , SAVE THE WORLD ROUTINE
36                ;
37                ; THIS ROUTINE SAVES AC0,AC1,AC2 AND CRY
38                ;
39
40 06150 040770 SAV?E:  STA     0,PAC?0         ;
41 06151 044770         STA     1,PAC?1         ;
42 06152 050770         STA     2,PAC?2         ;
43 06153 101100         MOVL    0,0             ;
44 06154 040763         STA     0,PCR?Y         ;
45 06155 001400         JMP     0,3             ;
46
47
48                ;
49                ; RST?R , RESTORE THE WORLD ROUTINE
50                ;
51                ; THIS ROUTINE RESTORES THE AC0,AC1,AC2 AND CRY
52                ;
53
54 06156 020761 RST?R:  LDA     0,PCR?Y         ;
55 06157 101200         MOVR    0,0             ;
56 06160 020760         LDA     0,PAC?0         ;
57 06161 024760         LDA     1,PAC?1         ;
58 06162 030760         LDA     2,PAC?2         ;
59 06163 001400         JMP     0,3             ;
60
```

```
 0087 FPYDI
01                       ;
02                       ;TYP?E AND TPS?P ROUTINES.
03                       ;
04                       ;THE CALLING SEQUENCE IS:
05                       ;
06                       ;        LDA     0,CHARACTER TO BE PRINTED (RIGHT BYTE)
07                       ;        JSR     @ITYP?E
08                       ;
09                       ;THE CALLING SEQUENCE IS:
10                       ;
11                       ;        JSR     @ITPS?P
12                       ;        NORMAL RETURN WITH AC0 = 40
13                       ;
14
15
16 06164 135005 RUB?:    MOV     1,3,SNR         ;CAN'T RUB-OUT IF AC1 = 0
17 06165 000467          JMP     TIN?R           ;RETURN WITH ILLEGAL CHARACTER
18 06166 126400          SUB     1,1
19 06167 156422          SUBZ    2,3,SZC         ;SKIP IF AC3 IS LESS THAN AC2
20 06170 125401          INC     1,1,SKP
21 06171 157001          ADD     2,3,SKP
22 06172 000775          JMP     .-3
23 06173 054501          STA     3,FST?D         ;"FST?D" IS NON -1
24 06174 020750          LDA     0,PC67?         ;AC0 = 60
25 06175 163000          ADD     3,0
26 06176 004405          JSR     TYP?E           ;ECHO AND DELETE THE DIGIT
27 06177 000524          JMP     TIN?W
28
29 06200 040445 TPS?P:   STA     0,TAC?0         ;SAVE AC0
30 06201 020447          LDA     0,PC47?
31 06202 101001          MOV     0,0,SKP         ;SKIP OVER AC0 SAVE
32 06203 040442 TYP?E:   STA     0,TAC?0         ;SAVE AC0
33 06204 175100          MOVL    3,3             ;SAVE CRY AND RTN ADDR
34 06205 054441          STA     3,TYP?R         ;TYPE THE RIGHT BYTE OF AC0
35 06206 034567          LDA     3,P177?         ;STRIP THE PARITY BIT
36 06207 163400          AND     3,0             ;
37 06210 034564          LDA     3,INT?          ;IF IT IS HERE DUE TO SWITCH
38                                               ;SETTING ROUTINE THEN THE TYPE
39                                               ;OUTS TO THE TTY WILL BE ENABLED
40 06211 175404          INC     3,3,SZR         ;SKIP IF INT? IS -1
41 06212 034205          LDA     3,SWREG         ;READ THE SWITCHES
42 06213 177100          ADDL    3,3             ;SHIFT AC3 BY 2 PLACES
43 06214 175112          MOVL#   3,3,SZC         ;SKIP IF TYPEOUTS ARE NOT
44                                               ;SUPPRESSED
45 06215 000405          JMP     PLP?T
46 06216 061111 TTY?:    DOAS    0,TTO
47 06217 063511          SKPBZ   TTO
48 06220 000777          JMP     .-1
49 06221 060211          NIOC    TTO             ;CLEAR TTO DONE FLAG
50 06222 177100 PLP?T:   ADDL    3,3             ;SHIFT AC3 BY 2 PLACES
51 06223 177103          ADDL    3,3,SNC         ;SKIP IF THE OUTPUT IS
52                                               ;REQUIRED ON THE LPT
53 06224 000405          JMP     TPR?T
54 06225 061117          DOAS    0,LPT           ;OUTPUT THE CHARACTER TO LPT
55 06226 063517          SKPBZ   LPT             ;WAIT FOR LPT
56 06227 000777          JMP     .-1
57 06230 060217          NIOC    LPT             ;CLEAR THE DONE FLAG FOR LPT
58 06231 034544 TPR?T:   LDA     3,P177?         ;
59 06232 116043          ADCO    0,3,SNC         ;SKIP IF IT WAS RUBOUT
60 06233 034415          LDA     3,PC47?         ;AC3 = 40
```

```
 0088 FPYDI
01 06234 162432          SUBZ#   3,0,SZC         ;SKIP FOR NON PRINTING CHR.
02 06235 010706          ISZ     CHR?Z
03 06236 034541          LDA     3,PC17?         ;AC3 = 15
04 06237 116445          SUBO    0,3,SNR         ;SKIP IF IT WAS NOT A "CR"
05 06240 054703          STA     3,CHR?Z         ;CLEAR THE HORZ POS
06 06241 020404          LDA     0,TAC?0         ;RESTORE AC0
07 06242 034404          LDA     3,TYP?R         ;RESTORE CRY AND RTN ADDR
08 06243 175200          MOVR    3,3             ;
09 06244 001400          JMP     0,3             ;RETURN
10 06245 000000 TAC?0:   0
11 06246 000000 TYP?R:   0
12
13 06247 000000 ZSU?P:   0
14 06250 000040 PC47?:   40
15
16
17                       ;
18                       ;TIN?O AND TIN?D ROUTINES.
19                       ;
20                       ;THE CALLING SEQUENCE IS:
21                       ;
22                       ;        JSR     @ITIN?O ;ACCEPT IN OCTAL
23                       ;        ERROR RETURN    WITH BAD CHARACTER IN AC0
24                       ;        NORMAL RETURN   WITH TERMINATING
25                       ;                        CHARACTER IN AC0
26                       ;                        0,40,12,15,55 FOR
27                       ;                        NULL,SPACE,L/F,C/R AND
28                       ;                        COMMA RESPECTIVELY.
29                       ;
30                       ;        NOTE:
31                       ;             THE NUMBER IS ACCEPTED IN AC1 FROM TTI BY TIN?O AND
32                       ;             BAD CHARACTER IS ANY CHARACTER THAT IS NOT A LEGAL
33                       ;             DIGIT OR A TERMINATING CHARACTER
34                       ;
35                       ;THE CALLING SEQUENCE IS:
36                       ;
37                       ;        JSR     @ITIN?D ;ACCEPT IN DECIMAL
38                       ;        ERROR RETURN    ;SAME FORMAT AS TIN?O
39                       ;        NORMAL RETURN
40                       ;
41
42
43 06251 020525 TIN?C:   LDA     0,PC17?
44 06252 004731          JSR     TYP?E
45 06253 010663 TIN?X:   ISZ     RTN?A
46 06254 040420 TIN?R:   STA     0,FST?D         ;"FST?D" IS NON -1
47 06255 152000          ADC     2,2             ;AC2 = -1
48 06256 020771 TSI?:    LDA     0,ZSU?P
49 06257 175620          INCZR   3,3             ;AC3 IS 1 IF THE CHARACTER
50                                               ;TYPED WAS A + AND A 100000
51                                               ;IF IT WAS A -.
52 06260 054767          STA     3,ZSU?P
53 06261 101112          MOVL#   0,0,SZC         ;SKIP IF THE PREVIOUS SIGN
54                                               ;WAS A PLUS
55 06262 124400          NEG     1,1             ;TAKE TWO'S COMPLEMENT IF
56                                               ;THE PREVIOUS SIGN WAS "-"
57 06263 034656          LDA     3,PAC?1
58 06264 167000          ADD     3,1
59 06265 044654          STA     1,PAC?1         ;PAC?1 HAS THE INTERMEDIATE
60                                               ;RESULT
```

```
0089 FPYDI
06266 126400          SUB     1,1
06267 151113          MOVL#   2,2,SNC         ;SKIP IF EXIT IS REQUIRED
06270 000433          JMP     TIN?W
06271 004665          JSR     RST?R           ;RESTORE THE WORLD
06272 020402          LDA     0,FST?D         ;RESTORE AC0
06273 002643          JMP     @RTN?A          ;RETURN

06274 000000 FST?D:   0

06275 054641 TOD?T:   STA     3,RTN?A         ;SAVE THE RTN ADDR
06276 004652          JSR     SAV?E           ;SAVE THE WORLD
06277 102000          ADC     0,0             ;AC1 = -1 (ENTRY FOR ODT)
06300 040774          STA     0,FST?D         ;LOOK FOR FIRST DIGIT
06301 101120          MOVZL   0,0             ;AC1 = -2
06302 000411          JMP     TIN?Z
06303 054633 TIN?O:   STA     3,RTN?A         ;OCTAL ENTRY,SAVE RTN ADDR
06304 004644          JSR     SAV?E           ;SAVE THE WORLD
06305 102120          ADCZL   0,0             ;OCTAL ENTRY SWITCH
06306 000404          JMP     TIN?Q
06307 054627 TIN?D:   STA     3,RTN?A         ;DECIMAL ENTRY,RTN SAVED
06310 004640          JSR     SAV?E           ;SAVE THE WORLD
06311 102440          SUBO    0,0             ;DECIMAL ENTRY SWITCH
06312 126400 TIN?Q:   SUB     1,1
06313 030463 TIN?Z:   LDA     2,PC1?2
06314 113000          ADD     0,2             ;AC2 IS 10 FOR OCTAL AND 12
                                              ;FOR DECIMAL NUMBERS
06315 102440          SUBO    0,0
06316 040731          STA     0,ZSU?P         ;SIGN AND LEADING SPACES FLAG
06317 034730 TIN?S:   LDA     3,ZSU?P
06320 175014          MOV#    3,3,SZR         ;SKIP FOR LEADING SPACES
06321 000732          JMP     TIN?X
06322 054617          STA     3,PAC?1
06323 063610 TIN?W:   SKPDN   TTI
06324 000777          JMP     .-1
06325 060610          DIAC    0,TTI
06326 004655          JSR     TYP?E
06327 034446          LDA     3,P1777
06330 163400          AND     3,0             ;STRIP THE PARITY BIT
06331 116415          SUB#    0,3,SNR         ;SKIP IF NOT A RUB-OUT
06332 000632          JMP     RUB?
06333 034715          LDA     3,PC470
06334 116414          SUB#    0,3,SZR
06335 101015          MOV#    0,0,SNR
06336 000761          JMP     TIN?S           ;SPACE, OR NULL
06337 034442          LDA     3,TIN?2
06340 116405          SUB     0,3,SNR         ;COMMA
06341 000712          JMP     TIN?X
06342 175414          INC#    3,3,SZR         ;MINUS
06343 175235          MOVZR#  3,3,SNR         ;OR PLUS ?
06344 000712          JMP     TSI?            ;MODIFY THE SIGN
06345 034432 TIN?M:   LDA     3,PC1?5         ;AC3 = 15
06346 116415          SUB#    0,3,SNR         ;IS IT A CARRIAGE RETURN?
06347 000702          JMP     TIN?C           ;IF CR THEN GO TO TIN?C
06350 034426          LDA     3,PC1?2         ;AC3 = 12
06351 116404          SUB     0,3,SZR         ;SKIP FOR LINE FEED
06352 000403          JMP     TIN?N
06353 020424          LDA     0,PC1?5         ;AC0 = 15
06354 000676          JMP     TIN?C+1
06355 034423 TIN?N:   LDA     3,TIN?1
06356 117022          ADDZ    0,3,SZC         ;SKIP IF NOT A DIGIT
```

```
0090 FPYDI
06357 156513          SUBL#   2,3,SNC         ;SKIP IF DIGIT
06360 000674          JMP     TIN?R
06361 010666          ISZ     ZSU?P           ;OUT OF LEADING SPACES
06362 102400          SUB     0,0             ;AC0 = 0
06363 010711          ISZ     FST?D           ;SKIP IF IT WAS FIRST DIGIT
                                              ;FOR ODT
06364 121120          MOVZL   1,0
06365 105120          MOVZL   0,1
06366 125120          MOVZL   1,1             ;AC1 IS SHIFTED BY 3 PLACES
06367 167000          ADD     3,1             ;& OLD PAC?1'S + NEW DIGIT
06370 155220          MOVZR   2,3
06371 175232          MOVZR#  3,3,SZC         ;SKIP IF OCTAL MODE
06372 107000          ADD     0,1             ;ADD 2 OLD PAC?1'S
06373 000730          JMP     TIN?W

06374 000000 IN?T:    0                       ;TYPE OUTSS CAN BE FORCED TO
                                              ;TTY BY PLACING -1 IN THIS LOC.
06375 000177 P1777:   177
06376 000012 PC1?2:   12
06377 000015 PC1?5:   15
06400 177720 TIN?1:   -60
06401 000054 TIN?2:   54
06402 000100 C10?0:   100
```

```
01
02
03              ;FILENAME = SWPAK
04
05
06              ;1.     SWITCH SETTINGS
07              ;
08              ;       LOCATION "SWREG" IS USED TO SELECT THE PROGRAM OPTIONS
09              ;       (NOT SYSTEM CONFIGURATION). WHILE RUNNING UNDER DTOS,
10              ;       THIS LOCATION WILL BE LOADED BY THE MONITOR.
11              ;       HOWEVER UNDER STAND ALONE AND PROGRAM LOAD MODES THIS
12              ;       LOCATION WILL BE SET ACCORDING TO THE ANSWERS SUPPLIED
13              ;       BY THE OPERATOR. IN ANY CASE THE OPTIONS CAN BE CHANGED
14              ;       OR VERIFIED BY USING ONE OF THE COMMANDS GIVEN IN SEC.
15              ;       1.2
16              ;
17              ;
18              ;1.1    SWITCH OPTIONS
19              ;       DIFFERENT BITS AND THEIR INTERPRETATION AT LOCATION
20              ;       "SWREG" IS AS FOLLOWS:
21              ;
22              ;       BIT     OCTAL   BINARY  INERPRETATION
23              ;               VALUE   VALUE
24              ;
25              ;       1               0       LOOP ON ERROR
26              ;               40000   1       SKIP LOOPING ON ERROR
27              ;
28              ;       2               0       PRINT TO CONSOLE
29              ;               20000   1       ABORT PRINT OUT TO CONSOLE
30              ;
31              ;       3               0       PRINT DETAILED ERROR ON THE
32              ;                               SELECTED DEVICE/DEVICES
33              ;               10000   1       ONLY 1 FAILURE REQUIRED
34              ;
35              ;       4               0       ALLOW END OF PASS PRINT OUT
36              ;               04000   1       SUPPRESS END OF PASS PRINT OUT
37              ;
38              ;       5               0       DO NOT PRINT ON THE LINE PRINTER
39              ;               02000   1       PRINT ON THE LINE PRINTER
40              ;
41              ;       6               0       DO NOT HALT ON ERROR
42              ;               01000   1       HALT ON ERROR
43              ;
44              ;       7               0       DO NOT PRINT PASSING OF
45              ;                               EACH TEST ON PRINTING DEVICE
46              ;               00400   1       PRINT PASSING OF EACH TEST
47              ;
48              ;1.2    SWITCH COMMANDS
49              ;       ONCE THE PROGRAM STARTS EXECUTING THE STATE OF ANY OF
50              ;       THE BITS CAN BE CHANGED BY HITTING KEYS 1 THROUGH 6. THE
51              ;       PROGRAM WILL CONTINUE RUNNING AFTER UPDATING THE OPTIONS
52              ;       EACH KEY WILL COMPLEMENT THE STATE OF THE BIT AFFILIAT-
53              ;       ED WITH IT, THUS BIT 4 CAN BE ALTERED BY HITTING KEY 4.
54              ;       SETTIN OF ANY BIT OF LOCATION "SWREG" WILL SET BIT 0,
55              ;       (DEFAULT MODE IS DEFINED AS ALL BITS OF SWREG SET TO 0)
56              ;       THE PROGRAM CAN BE LOCKED INTO SWITCH MODIFICATION MODE
57              ;       BY TYPING A 0, IN WHICH CASE MORE THAN ONE BITS CAN BE
58              ;       CHANGED BEFORE THE CONTROL IS ALLOWED TO RETURN TO THE
59              ;       MAIN PROGRAM.
60              ;
```



```
01              ;1.2.1  OTHER COMMANDS
02              ;
03              ;       "CR"    A "RETURN" CAN BE TYPED TO CONTINUE THE PROGRAM
04              ;               AFTER ITS LOCKED IN A SWITCH MODIFICATION MODE
05              ;
06              ;       ^D      THIS COMMAND GIVEN AT ANY TIME WILL RESET "SWREG
07              ;               TO DEFAULT MODE AND RESTART THE PROGRAM.
08              ;
09              ;       ^R      THIS COMMAND GIVEN AT ANY TIME WILL RESTART THE
10              ;               PROGRAM, SWITCHES ARE LEFT WITH THE VALUES THEY
11              ;               HAD BEFORE THE COMMAND WAS ISSUED.
12              ;
13              ;       ^O      THIS COMMAND GIVEN AT ANY TIME WILL CAUSE THE
14              ;               PROGRAM CONTROL TO GO TO ODT (SEE SEC. 6)
15              ;
16              ;       ^S      PRINT THE SUMMARY OF ERRORS ACCUMULATED SO FAR.
17              ;
18              ;       M       THIS COMMAND GIVEN AT ANY TIME WILL PRINT THE
19              ;               CURRENT OPERATING MODES.
20              ;
21              ;
```

```
0093 FPYDI
01
02
03
04
05                ;THIS PACKAGE IS USED TO CHANGE THE SETTINGS OF LOCATION
06                ;"SWREG" OF PAGE 0. THE PROGRAM CONTROL SHOULD ENTER "INP?I OR
07                ;INP?J" WITH AC3 HAVING THE RTN ADDR. IF THE TTI INPUT IS
08                ;ON INTERRUPT BASIS, ENTER AT "INP?I" OTHERWISE ENTER AT "INP?J"
09                ;THE INPUT IS ECHOED AFTER A "CR". IF THE COMMAND IS
10                ;NOT A LEGAL ONE THEN THE CONTROL IS RETURNED WITHOUT DOING
11                ;ANY THING, OTHERWISE ONE OF THE FOLLOWING COMMANDS IS
12                ;EXECUTED:
13                ;KEYS 1-9 AND A-F ARE USED TO COMPLEMENT THE CURRENT VALUE
14                ;OF BITS 1-15 OF "SWREG". IF ONE OF THESE KEYS IS HIT THE
15                ;CORRESPONDING BIT OF "SWREG" IS COMPLEMENTED AND THE CONTROL
16                ;IS RETURNED TO THE STATE PROGRAM HAD BEFORE HITTING THE KEY
17                ;TYPING OF A "0" WILL LOCK THE PROGRAM IN A SWITCH MODIFICATION
18                ;MODE IN WHICH CASE MORE THAN ONE BITS CAN BE CHANGED BEFORE
19                ;THE CONTROL IS ALLOWED TO RETURN TO THE MAIN PROGRAM. HITTING
20                ;THE "CR" KEY WILL UNLOCK THE PROGRAM FROM THIS MODE.
21                ;"AD" THIS COMMAND GIVEN AT ANY TIME WILL RESET THE "SWREG"
22                ;TO DEFAULT MODE (ALL ZEROS) AND RESTART THE PROGRAM AT ADD.
23                ;STORED IN LOCATION "INS?"
24                ;"AR" THIS COMMANND GIVEN AT ANY TIME  WILL RESTART THE PROG.
25                ;AT ADDRESS STORED IN LOCATION "INS?"
26                ;"M" THIS COMMAND GIVEN AT ANY TIME WILL PRINT THE  CURRENT
27                ;OPERATING MODES.
28                ;BEFORE THE CONTROL IS RETURNED TO THE MAIN PROGRAM BIT 0 WILL
29                ;BE SET IF ANY OF THE OTHER BITS OF "SWREG" IS SET, OTHERWISE
30                ;IT WILL BE CLEARED
31                ;"ODT" ROUTINE USES ONE LOCATION IN PAGE 0 FOR BREAK POINT.
32                ;ADDRESS OF THIS LOCATION SHOULD BE PLACED AT "ODO?F"
33                ;
34                ;THE CALLING SEQUENCE IS:
35                ;
36                ;       JSR     @IINP? ,WHERE IINP? POINTS TO
37                ;               INP?I OR INP?J
38                ;               ENTER AT INP?I FOR INTERRUPT
39                ;               DRIVEN PROGRAMS ,OTHERWISE AT INP?J
40                ;
41
42 06403 000033 IN3?:  33
43 06404 000136 IN136: 136
44 06405 000104 IN104: 104
45
46 06406 054574 INP?I:  STA     3,INP?R
47 06407 176121         ADCZL   3,3,SKP
48 06410 054572 INP?J:  STA     3,INP?R         ;SAVE THE RETURN ADDRESS
49 06411 054466         STA     3,INT?E
50 06412 004543         JSR     INS?V           ;SAVE THE STATUS
51 06413 040574         STA     0,INL?K         ;"INL?K" IS NOT =1
52 06414 060610 IN0?:   DIAC    0,TTI           ;READ THE INPUT
53 06415 034760         LDA     3,P177          ;AC3 = 177
54 06416 163400         AND     3,0             ;GET RID OF THE PARITY BIT
55 06417 034760         LDA     3,PC125         ;AC1 = 15
56 06420 116415         SUB#    0,3,SNR         ;SKIP IF THE CHARACTER TYPED
57                                              ;WAS NOT "CR"
58 06421 000426         JMP     INR?
59 06422 040557         STA     0,INS?3         ;SAVE THE CHARACTER
60 06423 024757         LDA     1,C1070         ;AC1 = 100
```

```
0094 FPYDI
01 06424 034752         LDA     3,PC172
02 06425 116414         SUB#    0,3,SZR         ;SKIP IF IT IS A LINE FEED
03 06426 034755         LDA     3,IN3?3         ;AC3 = 33
04 06427 162453         SUBO#   3,0,SNC         ;DON'T SKIP IF AC0 IS 33
05                                              ;OR MORE
06 06430 000451         JMP     IN1?
07 06431 107000 IN5?:   ADD     0,1             ;AC1 = 100+ ASCII VALUE OF
08                                              ;CONTROL CHARACTER
09 06432 024752         LDA     0,IN136                 ;AC0 = 136
10 06433 006216         JSR     @ITYP?E         ;TYPE A
11 06434 121000         MOV     1,0
12 06435 006216         JSR     @ITYP?E
13 06436 034562         LDA     3,I121          ;AC3 = 121
14 06437 162015         ADC#    3,0,SNR         ;SKIP IF IT IS NOT AR
15 06440 000405         JMP     IN6?
16 06441 034744         LDA     3,IN104                 ;AC3 = 104
17 06442 116404         SUB     0,3,SZR         ;SKIP IF IT WAS A AD
18 06443 000504         JMP     IN4?
19 06444 054205         STA     3,SWREG         ;LOAD "SWREG" WITH 0
20 06445 034541 IN6?:   LDA     3,INS?          ;AC3 = ADDRESS OF THE LOCATION
21                                              ;WHERE THE PROGRAM WILL START
22 06446 054534         STA     3,INP?R
23 06447 176400 INR?:   SUB     3,3
24 06450 165000         MOV     3,1
25 06451 006207         JSR     @ICHL?F
26 06452 054722         STA     3,INT?
27 06453 030205         LDA     2,SWREG
28 06454 176220         ADCZR   3,3             ;AC3 = 77777
29 06455 173404         AND     3,2,SZR         ;SKIP IF THE SWITCHES ARE SET
30                                              ;TO ALL ZERO'S
31 06456 172000         ADC     3,2
32 06457 050205         STA     2,SWREG
33 06460 034523         LDA     3,STA?T
34 06461 175223         MOVZR   3,3,SNC         ;SKIP IF DONE BIT ON TTO IS TO
35                                              ;BE LEFT SET
36 06462 060211         NIOC    TTO
37 06463 175220         MOVZR   3,3
38 06464 175204         MOVR    3,3,SZR         ;LOAD THE CARRY BIT AND SKIP
39                                              ;IF THE INTERRUPTS ARE NOT TO
40                                              ;BE ENABLED
41 06465 125004         MOV     1,1,SZR         ;SKIP IF THE INSTRUCTION BEING
42                                              ;EXECUTED IS A "P"
43 06466 176000         ADC     3,3             ;INTERRUPTS ARE TO BE LEFT DIS-
44                                              ;ABLED
45 06467 054514         STA     3,STA?T         ;"STA?T" IS 0 IF INTERUPTS ARE
46                                              ;TO BE ENABLED AND =1 OTHERWISE
47 06470 020506         LDA     0,INS?0         ;RESTORE THE ACCUMULATORS
48 06471 024506         LDA     1,INS?1
49 06472 030506         LDA     2,INS?2
50 06473 034506         LDA     3,INS?3
51 06474 010507         ISZ     STA?T
52 06475 060177         INTEN
53 06476 002504         JMP     @INP?R          ;START EXECUTING THE USER'S
54                                              ;PROGRAM
55
56
57 06477 000000 INT?E:  0
58 06500 006200 ITPS?P: TPS?P
```

```
10095 FPYDI
01
02 06501 006216 IN1?:   JSR     @ITYP?E         ;ECHO THE CHARACTER
03 06502 034517         LDA     3,IN6?0         ;AC3 = 60
04 06503 152620         SUBZR   2,2             ;AC2 = 180000
05 06504 116405         SUB     0,3,SNR         ;SKIP IF THE DIGIT TYPED WAS
06                                              ;NOT 0
07 06505 000503         JMP     IN3?
08
09 06506 151221 IN2?:   MOVZR   2,2,SKP         ;SHIFT AC2 TO RIGHT
10 06507 126520         SUBZL   1,1             ;AC1 = 1
11 06510 175405         INC     3,3,SNR
12 06511 000501         JMP     IN3?+2
13 06512 147415         AND#    2,1,SNR         ;STAY IN LOOP UNTIL ALL BITS
14                                              ;OF SWREG ARE CHECKED
15 06513 000773         JMP     IN2?
16 06514 106400         SUB     0,1             ;WHEN THE CONTROL COMES HERE
17                                              ;FOR THE FIRST TIME AC1 = 100
18 06515 135000         MOV     1,3
19 06516 151225         MOVZR   2,2,SNR
20 06517 000430         JMP     IN4?
21 06520 024503         LDA     1,IN1?5         ;AC1 = 15
22 06521 167004         ADD     3,1,SZR         ;SKIP IF THE COMMAND WAS "M"
23 06522 000765         JMP     IN2?+1
24
25 06523 006207 INM?:   JSR     @ICRL?F         ;TYPE A "CR" AND "LF"
26 06524 152520         SUBZL   2,2             ;SET AC2= 1
27 06525 006210         JSR     @IPDC?S         ;PRINT THE CONTENTS OF AC1
28 06526 000040         40
29 06527 125400         INC     1,1
30 06530 034472         LDA     3,IN1?2         ;AC3 = 12
31 06531 166452         SUBO#   3,1,SZC         ;SKIP IF AC1 IS GREATER OR EQUAL
32                                              ;TO AC3
33 06532 006746         JSR     @ITPS?P         ;PRINT A SPACE
34 06533 151124         MOVZL   2,2,SZR         ;SKIP AFTER TYPING # 15
35 06534 000771         JMP     INM?+2
36 06535 006207         JSR     @ICRL?F
37 06536 030205         LDA     2,SWREG
38 06537 151140         MOVOL   2,2             ;AC2 HAD SWITCH SETTINGS
39 06540 126560         SUBCL   1,1             ;BRING THE CARRY BIT IN AC1
40 06541 006210         JSR     @IPDC?S         ;TYPE THE CONTENTS OF AC1
41 06542 000040         40
42 06543 006735         JSR     @ITPS?P         ;TYPE A SPACE
43 06544 151124         MOVZL   2,2,SZR         ;SKIP AFTER TYPING ALL THE 16
44                                              ;BITS
45 06545 000773         JMP     .-5
46 06546 006207         JSR     @ICRL?F
47 06547 010440 IN4?:   ISZ     INL?K           ;SKIP IF THE PROGRAM IS LOCKED
48                                              ;IN SWITCH INPUT MODE
49 06550 000677         JMP     INR?
50 06551 014436         DSZ     INL?K           ;NEVER SKIP
51 06552 063610         SKPDN   TTI             ;WAIT FOR OPERATOR INPUT
52 06553 000777         JMP     .-1
53 06554 000640         JMP     IN0?
54
55
56 06555 040421 INS?V:  STA     0,INS?0         ;SAVE THE ACC.
57 06556 044421         STA     1,INS?1
58 06557 050421         STA     2,INS?2
59 06560 102560         SUBCL   0,0             ;SAVE THE CARRY
60 06561 010716         ISZ     INT?E           ;SKIP IF DDT IS ENTERED THRU
```

```
0096 FPYDI
01                                              ;INTERRUPT HANDLER
02 06562 063577         SKPBZ   CPU             ;SKIP IF INTERRUPTS ARE NOT
03                                              ;ENABLED
04 06563 101141         MOVOL   0,0,SKP
05 06564 101120         MOVZL   0,0
06 06565 060277         INTDS
07 06566 063611         SKPDN   TTO             ;SAVE THE TTO STATUS
08 06567 101121         MOVZL   0,0,SKP
09 06570 101140         MOVOL   0,0
10 06571 040412         STA     0,STA?T
11 06572 152000         ADC     2,2
12 06573 050501         STA     2,INT?          ;FORCE THE TYPE OUTS ON
13                                              ;THE TELETYPE
14 06574 002207         JMP     @ICRL?F
15
16 06575 005400 INS?A:  JSR     0,3
17 06576 000000 INS?0:  0
18 06577 000000 INS?1:  0
19 06600 000000 INS?2:  0
20 06601 000000 INS?3:  0
21 06602 000000 INP?R:  0
22 06603 000000 STA?T:  0
23 06604 177777 INB?A:  -1
24 06605 000000 INB?I:  0
25 06606 000571 INS?:   RES?T
26 06607 000000 INL?K:  0
27
28 06610 176000 IN3?:   ADC     3,3             ;AC3 = -1
29 06611 054776         STA     3,INL?K         ;LOCK IN SWITCH INPUT MODE
30 06612 024205         LDA     1,SWREG         ;READ THE CURRRENT VALUE OF
31                                              ;"SWREG"
32 06613 133414         AND#    1,2,SZR         ;TAKE XOR OF AC1 AND AC2
33 06614 146401         SUB     2,1,SKP
34 06615 147000         ADD     2,1
35 06616 044205         STA     1,SWREG         ;SAVE THE NEW VALUE OF "SWREG"
36 06617 000730         JMP     IN4?
37
38
39 06620 000121 I12?1:  121
40 06621 000060 IN6?0:  60
41 06622 000012 IN1?2:  12
42 06623 000015 IN1?5:  15
43
44
```

```
10097 FPYDI
01
02
03
04
05              ;FILENAME = ODTPK
06
07              ;2.     OCTAL DEBUG TOOL (ODT)
08              ;
09              ;       THE DIAGNOSTIC IS EQUIPED WITH A BULT IN ODT WHICH CAN
10              ;       BE ACCESSED BY HITTING CONTROL O (^O) AT ANY TIME DURING
11              ;       THE EXECUTION OF THE PROGRAM (AFTER SETTING THE PARA-
12              ;       METERS).
13              ;       ON ENTERING ODT THE ADDRESS OF THE LOCATION HAVING THE
14              ;       NEXT INSTRUCTION TO BE EXECUTED WILL BE TYPED-OUT.
15              ;
16              ;2.1    CONVENTIONS AND SYMBOLS
17              ;       THE FOLLOWING CONVENTIONS ARE USED BY THE ODT:
18              ;       ?       PRESSING ANY ILLEGAL KEY CAUSES THE ODT TO RES-
19              ;               POND WITH A "?".
20              ;       @       ODT IS READY AND AT YOUR SERVICE.
21              ;
22              ;2.2    COMMAND STRUCTURE
23              ;       AN ODT COMMAND THE FOLLOWING FORMAT:
24              ;               [ARGUMENT][COMMAND]
25              ;       AN ARGUMENT MAY BE ONE OF THE FOLLOWING:
26              ;       EXP     AN OCTAL EXPRESSION CONSISTING OF OCTAL NUMBERS
27              ;               SEPARATED BY PLUS (+) OR MINUS (-) SIGNS. LEAD-
28              ;               ING ZEROS NEED NOT BE TYPED.
29              ;       ADR     AN ADDRESS IS THE SAME AS AN EXPRESSION EXCEPT
30              ;               THAT BIT 0 IS NEGLECTED.
31              ;       A COMMAND IS A SINGLE TELETYPE CHARACTER
32              ;
33              ;2.3    ODT COMMANDS
34              ;       THE LOCATIONS THAT CAN BE EXAMINED AND MODIFIED BY THE
35              ;       USER ARE CALLED CELLS, THESE CELLS ARE OF TWO TYPE:
36              ;       INTERNAL CPU CELLS AND MEMORY LOCATIONS.
37              ;
38              ;2.3.1  OPENING INTERNAL CELLS
39              ;       THE COMMAND TO OPEN ONE OF THE INTERNAL REGISTERS IS OF
40              ;       THE FORM "NA" WHERE N IS ANY OCATL EXPRESSION BETWEEN
41              ;       0 AND 7
42              ;       0-3     FOR ACCUMULATORS 0-3
43              ;       4       FOR PC OF THE NEXT INSTRUCTION TO BE EXECUTED IN
44              ;               THE EVENT OF A "P" COMMAND.
45              ;       5       CPU AND TTO STATAS
46              ;               BIT     INTERPRETATION
47              ;               15      STATUS OF TTO DONE FLAG
48              ;               14      STATUS OF INTERRUPTS
49              ;               13      STATUS OF CARRY BIT
50              ;       6       ADDRESS OF THE LOCATION HAVING BREAK POINT (IF
51              ;               ANY)
52              ;       7       INSTRUCTION AT THE BREAK POINT LOCATION
53              ;
54              ;       OTHER COMMANDS TO OPEN CELLS ARE:
55              ;
56              ;       ADR/    OPEN THE CELL AND PRINT ITS CONTENTS
57              ;       ./      OPEN THE CELL CURRENTLY POINTED BY THE POINTER
58              ;               AND PRINT ITS CONTENTS.
59              ;       .+ADR/  ADD ADR TO THE POINTER, OPEN THE CELL AND PRINT
60              ;               ITS CONTENTS.

 0098 FPYDI
01              ;       .-ADR/  SUBTRACT ADR FROM THE POINTER, OPEN THE CELL AND
02              ;               PRINT ITS CONTENTS.
03              ;       "CR"    THE RETURN KEY IS USED TO CLOSE THE OPEN CELL
04              ;               WITH OR WITHOUT MODIFICATION.
05              ;       "LF"    LINE FEED IS USED TO CLOSE THE OPEN CELL WITH OR
06              ;               WITHOUT MODIFICATION AND TO OPEN THE SUCCEEDING
07              ;               CELL.
08              ;       /       CLOSE THE OPEN CELL WITHOUT MODIFICATION, AND
09              ;               OPEN THE CELL POINTED BY ITS CONTENTS.
10              ;       +ADR/   CLOSE THE OPEN CELL WITHOUT MODIFICATION, AND
11              ;               OPEN THE CELL POINTED BY ITS CONTENTS + ADR.
12              ;       -ADR/   CLOSE THE OPEN CELL WITHOUT MODIFICATION, AND
13              ;               OPEN THE CELL POINTED BY ITS CONTENTS - ADR.
14              ;
15              ;2.3.2  MODIFICATION OF A CELL
16              ;       ONCE A CELL HAS BEEN OPENED ITS CONTENTS CAN BE MODIFIED
17              ;       BY TYPING THE NEW VALUE THE CELL IS TO CONTAIN IN THE
18              ;       FORM OF AN OCTAL EXPRESSION FOLLOWED BY "CR" OR "LF".
19              ;       IF A + OR - IS TYPED AS THE FIRST CHARACTER OF THE EX-
20              ;       PRESSION THEN THE VALUE OF THE EXPRESSION IS ADDED TO OR
21              ;       SUBTRAC"ED FROM THE OLD CONTENTS OF THE CELL.
22              ;       ADDRESS ITSELF OR AN EXPREESION RELATIVE TO THE ADDRESS
23              ;       CAN BE DEPOSITED BY TYPING A "," OR ",+/-OCTAL EXPRESS-
24              ;       ION". A RUBOUT COMMAND GIVEN RIGHT AFTER OPENING A CELL
25              ;       ALLOWS THE MODIFICATION OF ITS CONTENTS AS IF THEY WERE
26              ;       TYPED IN JUST BEFORE THE COMMAND WAS ISSUED.
27              ;
28              ;2.3.3  OTHER ODT COMMANDS
29              ;       RUBOUT  THIS KEY IS USED TO DELETE ERRONEOUSLY TYPED
30              ;               DIGITS. EACH TIME THE KEY IS PRESSED RIGHT MOST
31              ;               DIGIT IS DELETED AND ECHOED ON THE TERMINAL. IF
32              ;               THE RUBOUT KEY IS PRESSED RIGHT AFTER OPENING A
33              ;               CELL THEN IT ALLOWS THE MODIFICATION OF THE CELL
34              ;               AS IF ITS CONTENTS WERE TYPED IN JUST BEFORE THE
35              ;               KEY WAS PRESSED.
36              ;       ADRB    INSERT A BREAK POINT AT LOCATION "ADR".
37              ;               ONLY ONE BREAK POINT CAN BE INSERTED AND ANY
38              ;               ENTRY TO ODT AFTER EXECUTING A BREAK POINT WILL
39              ;               CAUSE IT TO BE DELETED.
40              ;       D       DELETE THE BREAK POINT IF ANY.
41              ;       P       RESTART THE EXECUTION OF THE PROGRAM AT LOCATION
42              ;               POINTED BY 4A.
43              ;       ADRR    START EXECUTING THE PROGRAM AT "ADR" AFTER AN
44              ;               IO-RESET.
45              ;       K       KILL THE STRING TYPED SO FAR. THE ODT RESPONDS
46              ;               WITH A "?" AND THE OPEN CELL IS CLOSED WITHOUT
47              ;               MODIFICATION.
48              ;
49              ;
50              ;THE CALLING SEQUENCE IS:
51              ;
52              ;       JSR     @IODT?, WHERE IODT? POINTS TO ODT?I OR
53              ;               ODT?J
54              ;               ENTER AT ODT?I IF RUNNING ON
55              ;               INTERRUPT BASIS, OTHERWISE
56              ;               ENTER AT ODT?J
57              ;
```

```
10099 FPYDI

                        ;FILENAME= ODTPAK

06624 000210 ODA?C:  JSR     @IPDC?S        ;TYPE THE ACCUMULATOR NUMBER TO
                                            ;BE OPENED
06625 000101 N10?1:  101
06626 006652 ODA?:   JSR     @ITPS?P        ;TYPE A SPACE
06627 004746         JSR     INS?A          ;LOAD AC3 WITH THE ADDRESS OF
                                            ;INS?0
06630 171000         MOV     3,2
06631 133000         ADD     1,2
06632 034535         LDA     3,OD7?         ;AC3 = 7
06633 136453         SUBO#   1,3,SNC        ;SKIP IF ACC. NUMBER IS NOT OK
06634 000473         JMP     ODA?L

06635 116005 ODP?C:  ADC     0,3,SNR
06636 000612 ODR?T:  JMP     INR?+1         ;IF IT IS A "P" THEN PROCEED
06637 034762         LDA     3,OD670        ;AC3 = 60
06640 116005         ADC     0,3,SNR        ;SKIP IF AC0 IS NOT "/"
06641 000454         JMP     ODL?C
06642 145000         MOV     2,1
06643 175225         MOVZR   3,3,SNR        ;SKIP IF IT IS NOT A "."
06644 000434         JMP     ODW?T+4
06645 006107         JSR     @IMES?S        ;TYPE A "?"
06646 006774         TP??
06647 000425         JMP     ODW?T

06650 054731 ODT?I:  STA     3,INS?3        ;NORMALLY THE PROGRAM SHOULD
                                            ;ENTER ODT AT THIS LOCATION
                                            ;THROUGH AN INTERRUPT HANDLER
06651 034000         LDA     3,0            ;AC3 = RETURN ADDRESS
06652 000402         JMP     .+2
06653 054726 ODT?J:  STA     3,INS?3        ;ODT SHOULD BE ENTERED AT THIS
                                            ;POINT THRU A JSR
06654 054726         STA     3,INP?R        ;"INP?R" HAS THE RETURN ADD.
06655 004703         JSR     INS?V+3        ;SAVE  THE STATUS
06656 152400         SUB     2,2
06657 024723         LDA     1,INP?R
06660 044722 OUT?1:  STA     1,INP?R
06661 006212         JSR     @IZOC?T        ;TYPE THE HALT LOCATION+1
06662 151404         INC     2,2,SZR        ;SKIP IF THE BREAK POINT HAS
                                            ;TO BE DELETED
06663 000410         JMP     ODW?T-1
06664 020721 ODD?B:  LDA     0,INB?I        ;LOAD AC0 IN CASE THE BREAK
                                            ;POINT IS TO BE REMOVED
06665 034717         LDA     3,INB?A
06666 126001         ADC     1,1,SKP
06667 055000 ODB?P:  STA     3,0,2
06670 175414         INC#    3,3,SZR        ;SKIP IF THERE IS NO BREAK
                                            ;POINT
06671 042713         STA     0,@INB?A
06672 044712         STA     1,INB?A        ;ENABLE BREAK POINTS
06673 060210         NIOC    TTI            ;CLEAR THE TTY INPUT DONE SIGNAL
06674 006207 ODW?T:  JSR     @ICRL?F
06675 054474         STA     3,ODL?T        ;MAKE THE TAG FOR OPEN LOCATION
                                            ;NON -1 (NO LOCATION IS OPEN)
06676 006107         JSR     @IMES?S
06677 006775         ODT?P
06700 006472         JSR     @ITOD?T        ;ACCEPT OCTAL NUMBERS SEPERATED
                                            ;BY + OR - SIGNS

 0100 FPYDI
06701 060077         NIO     CPU            ;NO-OP
06702 034720         LDA     3,OD1?2
06703 116414         SUB#    0,3,SZR        ;CHECK FOR A LINE FEED OR RETURN
06704 034717         LDA     3,OD175
06705 116414         SUB#    0,3,SZR
06706 000426         JMP     ODD?R          ;DECODE REST OF THE COMMANDS
06707 010462         ISZ     ODL?T          ;SKIP IF A LOCATION IS OPEN
06710 000402         JMP     .+2
06711 045000         STA     1,0,2          ;RESTORE THE OPEN LOCATION
06712 101233         MOVZR#  0,0,SNC        ;SKIP IF IT WAS NOT A "CR"
06713 000761         JMP     ODW?T
06714 145400         INC     2,1            ;IF IT IS A LINE FEED THEN ADD
                                            ;1 TO AC2
06715 152220 ODL?C:  ADCZR   2,2            ;AC2 = 77777
06716 147400         AND     2,1            ;NEGLECT ADDRESS BIT 0
06717 020450         LDA     0,OD7?         ;AC0 = 7
06720 004655         JSR     INS?A          ;LOAD AC3 WITH THE ADDRESS
                                            ;OF INS?0
06721 166422         SUBZ    3,1,SZC        ;SKIP IF THE ADDRESS IN AC1 IS
                                            ;LESS THAN THE ADDRESS OF INS?0
06722 122433         SUBZ#   1,0,SNC        ;SKIP IF AC1 IS 7 OR LESS
06723 167001         ADD     3,1,SKP
06724 000700         JMP     ODA?C          ;OTHERWISE IT MUST BE AN ACC.
06725 006212         JSR     @IZOC?T        ;TYPE AC1
06726 131000         MOV     1,2
06727 126000 ODA?L:  ADC     1,1
06730 044441         STA     1,ODL?T        ;OPEN A LOCATION
06731 025000         LDA     1,0,2
06732 006211         JSR     @IPOC?T        ;TYPE THE OPEN LOCATION
06733 000745         JMP     ODW?T+4

06734 034671 ODD?R:  LDA     3,N10?1        ;AC3=101
06735 116405         SUB     0,3,SNR        ;IF AN "A" HAS BEEN TYPED
06736 000670         JMP     ODA?           ;THEN GO TO SERVICE ACCUMULATOR
                                            ;ROUTINE
06737 175655         INCOR#  3,3,SNR        ;SKIP IF IT WAS NOT A "B"
06740 034644         LDA     3,INB?A        ;MAKE SURE THAT THIS IS THE
06741 175645         INCOR   3,3,SNR        ;FIRST BREAK POINT
06742 125112         MOVL#   1,1,SZC        ;SKIP IF BIT 0 OF THE ADDRESS
                                            ;WHERE THE BREAK POINT WILL
                                            ;BE PLACED IS 0

06743 000414         JMP     ODD?C          ;DECODE OTHER COMMANDS
06744 044640         STA     1,INB?A        ;SAVE THE BREAK POINT ADD.
06745 036637         LDA     3,@INB?A
06746 054637         STA     3,INB?I        ;SAVE THE BREAK POINT INS?
06747 030421         LDA     2,ODD?F        ;BRING THE ADDRESS OF LOC.
                                            ;IN PAGE 0 TO BE USED BY THE
                                            ;BREAK POINT INSTRUCTION

06750 020423         LDA     0,ODI?N
06751 143000         ADD     2,0            ;AC0 = BREAK POINT INST.
06752 004715         JSR     ODB?P
06753 054626 ODB?E:  STA     3,INS?3        ;THIS IS THE BREAK POINT ENTTRY
06754 004601         JSR     INS?V          ;SAVE EVERY THING
06755 024627         LDA     1,INB?A
06756 000702         JMP     ODT?1

06757 175665 ODD?C:  INCCR   3,3,SNR        ;SKIP IF THE COMMAND IS NOT
                                            ;"D" TO DELETE BREAK POINT
06760 000704         JMP     ODD?B
06761 034637         LDA     3,I12?1        ;AC3=121
```

```
 0101 FPYDI
01 06762 162014          ADC#    3,0,SZR         ;IF IT WAS A "R" THEN START THE
02                                               ;USERS PROGRAM
03 06763 000652          JMP     ODP?C
04 06764 062677          IORST                   ;IORST
05 06765 044615          STA     1,INP?R
06 06766 000650          JMP     ODR?T
07
08        006622 OD172=  IN172
09        006623 OD175=  IN175
10        006621 OD670=  IN670
11 06767 000007 OD77:    7
12 06770 000377 ODO?F:   377                     ;ADDRESS IN PAGE 0 TO BE USED
13                                               ;BY THE BREAK POINT
14 06771 000000 ODL?T:   0
15 06772 006275 ITOD?T:  TOD?T
16 06773 002000 ODI?N:   JMP     @0
17 06774 000077 TP??:    77
18 06775 000100 ODT?P:   "@
19
```

```
 0102 FPYDI
01 06776 000400 BUF:     .BLK 256.               ;WRITE BUFFER
02 07376 000000 BUFND:   0                       ;BUFFER END
03 07377 000400 RBUF:    .BLK 256.               ;READ BUFFER
04 07777 000000 RBUFD:   0                       ;READ BUFFER END
05                       ;       THIS PROGRAM IS TO BE RUN STANDALONE MODE ONLY
06
07
08 10000 047503          .TXT    /COPYRIGHT (C) DGC, 1976
09       054520
10       044522
11       044107
12       020124
13       041450
14       020051
15       043504
16       026103
17       030440
18       033471
19 10013 040466 ALL RIGHTS RESERVED./
20       046114
21       051040
22       043511
23       052110
24       020123
25       042522
26       042523
27       053122
28       042105
29       000056
30
31 10026 050306 DIRT:    .TXTE /FPY DIAG 00/
32       120131
33       144504
34       043501
35       030240
36       000060
37 10034 000000          0
38 10035 000200          200
39 10036 002002          002002
40 10037 070707          70707
41 10040 070707          70707
42 10041 070707          70707
43 10042 070707          70707
44 10043 000000          0
45                .END

**00000 TOTAL ERRORS, 00000 PASS 1 ERRORS
```

DKMES	004455		13/17	69/13					
DUBIB	004474		20/39	69/28					
DUBNG	004303		67/27						
DUDA	003545		6/42	60/46					
DUPA1	002661		48/06	48/09					
DUPA2	002652		48/02	48/12					
DUPAT	002643		9/41	47/55					
DURIT	003106		6/35	53/12	55/26				
DXI	003537		6/32	60/35					
DXI1	000272		10/01	11/09	11/20	11/29	12/09	41/12	41/32
			47/09	47/25					
DXMES	004466		13/21	69/22					
DU2	001766		36/11	36/14	36/33	36/56			
DU3	001746		36/36	36/40					
DU4	001750		36/42						
DU5	001727		36/25	36/39	36/46	36/53			
DUSEC	001731		36/17	36/27	36/42	36/43	36/48		
DUTRK	001730		36/16	36/26	36/49	36/50	36/55		
EGGS	000060		6/13	6/17					
ENCMS	003775		42/10	64/09					
ENDTS	000125	MC	6/03						
ENOF1	002174		41/14	41/15					
ENOF2	002216		41/35	41/37					
ENOFP	005051		41/42	73/25					
ENOFS	002161		41/07	41/28					
ENOFT	002210		41/23	41/30					
ENT?R	005466		9/14	78/49					
ERBTC	000264		9/55	61/18	77/01				
ERMES	003663		6/29	62/24					
ERRC1	003572		61/15	61/20	62/12				
ERRC2	003601		61/22	61/27					
ERRC3	003614		61/29	61/34					
ERRC4	003617		61/36	61/41					
ERRC5	003626		61/43	61/48					
ERRC6	003635		61/50	61/55					
ERRC7	003645		61/57	61/60	62/03				
ERRC8	003650		61/10	62/06					
ERRCK	003555		6/39	61/07					
ERRNC	003566		61/12	61/16					
ERRNU	000002		6/03						
ERROR	000157	MC	6/03	12/18	13/06	19/06	23/06	27/06	
ERR?	003630		6/15	6/37	9/16	81/08			
ERR?1	005704		81/45	81/52					
ERR?2	005676		81/46	81/56					
ERR?3	005711		81/20	81/57					
ERTS1	000257		9/50	12/08	14/02	16/01	18/12	21/01	23/53
			24/09	25/58	28/03	29/45	30/32	33/04	34/55
			36/60	39/56	41/09	46/24	46/29	46/41	56/28
			58/08	59/43	62/03	62/26			
E11	000667		14/07						
ET10	001473		29/49						
ET11	001512		31/05						
ET12	001611		33/08						
ET13	001700		34/59						
ET14	001774		37/04						
ET15	002154		39/60						
ET2	000734		16/07						
ET3	001025		18/16						

ET4	001170		22/05						
ET5	001262		24/13						
ET6	001337		26/02						
ET7	001416		28/07						
ETEST	000120	MC	6/03						
FCSH	002404		44/22	44/25	44/29	44/40	44/54		
FCSH1	002416		44/43	44/46	44/50				
FCSH2	002410		44/44	44/48					
FCSS	002420		44/34	44/53					
FCSS1	002442		45/10	45/12					
FIX	002252		42/24	42/29					
FLAGS	000116		7/02	11/53	12/12	13/23	32/16	36/21	36/34
			36/38	36/41	41/16	41/25	41/36	42/39	42/42
			45/12	47/12	47/23	47/30	52/41	52/46	52/56
			53/16	55/21	55/25	56/38	56/42	60/36	
FLAT	003270		56/14	56/18					
FLAT1	003271		56/16	56/17	56/19	56/24			
FLONG	004366		11/56	68/18					
FNTBE	002477		45/52						
FOWL	002350		9/29	44/08	44/23	44/27	44/30	44/55	45/08
FPY	000000		6/04	13/34	13/47	15/39	17/26	17/46	17/52
			18/01	19/30	19/47	19/49	19/55	20/02	20/14
			23/22	23/27	23/31	25/28	25/43	27/32	27/36
			27/39	29/15	29/24	32/26	32/29	34/18	38/19
			38/21	38/26	45/36	45/38	45/41	45/43	45/45
			51/15	51/40	51/43	52/09	52/17	52/20	52/37
			53/53	54/09	54/12	54/15	56/18	56/19	56/44
			57/01	77/26	77/32				
FPYND	004612		56/26	70/46					
FPYNR	003651		61/14	62/07					
FPYRC	000115		6/46	13/41	62/10	76/57			
FPYRD	004267		13/40	32/42	62/08	67/15	76/56	77/54	
FST?D	006274		87/23	88/46	89/05	89/08	89/13	90/05	
FUEX	002444		9/36	45/29	45/27				
FUNS	003553		60/50	60/52					
FUNTB	002454		9/37	45/33					
FUNWD	000117		7/18	53/46	54/10	54/19	54/21	55/32	55/37
GUODF	002320		9/46	43/13					
GUOS	002325		9/47	43/15					
HDMMC	003676		15/51	52/27	63/06				
HDMMS	004627		15/55	70/56					
HDL20	004037		32/38	64/43					
HDLD1	001207		17/49	17/60					
HDLD2	001017		17/37	17/55	17/59	18/08			
HDLD3	000771		17/45	18/04					
HDLD4	000754		17/29	17/31					
HDLD5	000762		17/30	17/38					
HDLD6	000746		17/25	17/32					
HDLDF	004145		17/35	34/25	65/53				
HDWLD	004134		65/44	77/50					
HEA?D	005720		81/22	82/04					
HEDUF	005063		51/30	73/35					
HELP	000202		9/05	78/51					
HERE	003427		59/08	59/14					
HESE1	001643		34/21	34/28					
HESE2	001630		34/17	34/23					
HESE3	001672		34/27	34/36	34/51				
HESE4	001626		34/12	34/15	34/41	34/48			

HHM1 000723	15/42	15/54					
HLSDC 004063	32/57	65/03					
HLSTS 005273	18/06	75/51					
HMDR 005171	11/22	74/45					
HNLOS 005107	17/57	73/55					
HSEC 001645	34/14	34/30	34/37	34/38	34/43		
HTRK 001644	34/13	34/29	34/44	34/45	34/50		
I1271 006620	94/13	96/39	100/60				
IBEGN 000237	9/04	9/34					
IBFPA 000245	9/40	47/56	48/07				
IBFPE 000244	9/39	47/58					
IBGN1 000271	9/60	41/29	41/40	41/45			
IBUF 000177	8/06	11/32	11/42	11/43	11/47	11/48	19/26
	19/44	19/45	19/58	23/20	25/18	27/30	29/13
	36/19	47/60	54/07	58/31	58/41	58/48	
ICOM1 000250	9/43	42/51	42/55				
ICOM2 000251	9/44	42/60					
ICOME 000255	9/48	43/03					
ICOMS 000252	9/45	42/44	42/48	42/59	43/16	60/18	
ICOMT 000112	6/43	42/20					
ICOMY 000113	6/44	42/30					
ICPET 000273	10/02	34/33	36/30				
ICRL? 000207	9/10	79/34	94/25	95/25	95/36	95/46	96/14
	99/54						
ICWE2 000274	10/03	10/13					
ICWE6 000275	10/04	10/14					
ICYC? 000214	9/15	13/60	15/59	18/10	20/59	24/07	25/56
	28/01	29/43	30/30	33/02	34/53	36/58	39/54
IDOUA 000111	6/42	13/33	13/46	15/38	17/25	17/45	17/51
	17/60	19/37	19/46	19/48	19/54	20/01	20/13
	23/21	23/26	23/30	25/27	25/42	27/31	27/35
	27/38	29/14	29/23	32/25	32/28	34/17	38/18
	38/20	38/25	45/22	51/14	51/39	51/42	52/08
	52/16	52/19	52/36	53/52	54/08	54/11	54/14
	56/43	56/60	77/27	77/31			
IDOPA 000246	9/41	58/17					
IDRI 000077	6/32	13/45	17/44	19/53	23/25	27/34	29/22
	45/21	51/38	52/14	52/32	53/21	53/25	56/59
	77/26						
IENT? 000213	9/14	13/29	15/32	17/21	19/21	23/15	25/15
	27/23	29/08	30/25	32/08	34/08	36/07	36/23
	38/08						
IERCK 000106	6/39	11/40	13/53	20/56	24/03	25/23	27/48
	27/58	34/32	36/29	58/46	58/53	77/51	
IERME 000074	6/29	13/39	13/55	15/50	15/54	17/34	17/56
	18/05	20/07	20/30	20/38	23/51	25/31	25/48
	27/45	29/35	29/38	32/37	32/41	32/56	34/24
	38/32	46/22	51/27	52/26	54/26	56/25	56/48
	59/41	61/16	61/23	61/30	61/37	61/44	61/51
	62/07	77/49	77/53				
IERR 000104	6/37	13/57	15/52	15/56	17/36	17/58	18/07
	20/09	20/26	20/33	20/40	20/44	20/49	23/37
	23/42	23/45	25/34	25/50	27/47	27/52	29/27
	29/40	32/39	32/44	32/58	34/26	38/34	52/28
	54/28	56/27	59/50	59/59	62/05	62/09	77/55
IERR? 000215	9/16						
IFNTB 000242	9/37	44/18					
IFOWL 000232	9/29	60/21					

IFUNX 000241	9/36	44/13	44/38	45/13			
IFWAS 005135	23/52	74/17					
IGOOD 000253	9/46	42/36	42/45				
IGOOS 000254	9/47	42/49					
IINP 000220	9/19	60/09					
IINP? 000206	9/09	38/15	43/05	45/26	46/14	52/04	58/39
	59/02	79/17	81/10				
IIODT 000221	9/20	60/26					
IITR 000073	6/28	11/37	23/48	34/10	34/34	36/31	46/15
	59/15	59/57	61/08	61/58	77/41		
ILE1 001225	23/34	23/40					
ILE2 001231	23/41	23/45					
ILE5 001233	23/48	24/04					
ILE6 001246	23/36	23/39	23/44	23/47	23/59		
ILE8 001254	23/50	23/55	23/58	24/05			
ILFUN 000140	7/35	23/24	23/56				
ILLFC 000265	9/56	61/53	77/17				
IMESS 000110	6/41	11/11	11/16	11/21	11/55	11/60	13/16
	13/20	41/41	42/09	44/08	45/04	46/02	46/49
	47/05	52/47	62/01	62/16	62/31	76/53	76/55
	76/59	77/03	77/07	77/11	77/15		
IMES? 000107	6/40	81/21	81/49	81/52	99/24	99/57	
IN0? 006414	93/52	95/53					
IN1? 006501	94/06	95/02					
IN170 006405	93/44	94/16					
IN172 006622	95/30	96/41	101/08				
IN173 006404	93/43	94/09					
IN175 006623	95/21	96/42	101/09				
IN2? 006506	95/09	95/15	95/23				
IN3? 006610	95/07	95/12	96/28				
IN373 006403	93/42	94/03					
IN4? 006547	94/18	95/20	95/47	96/36			
IN5? 006431	94/07						
IN6? 006445	94/15	94/20					
IN670 006621	95/03	96/40	101/10				
IN67A 006604	96/23	99/46	99/51	99/52	100/37	100/43	100/44
	100/54						
IN67I 006605	96/24	99/44	100/45				
INL?K 006607	93/51	95/47	95/50	96/26	96/29		
INM? 006523	95/25	95/35					
INP?I 006406	93/46						
INP?J 006410	9/19	93/48					
INP?R 006602	93/46	93/48	94/22	94/53	96/21	99/35	99/38
	99/39	101/05					
INR? 006447	93/58	94/23	95/49	99/17			
INSER 001454	29/16	29/32					
INS? 006606	94/20	96/25					
INS70 006576	94/47	95/56	96/17				
INS71 006577	94/48	95/57	96/18				
INS72 006600	94/49	95/58	96/19				
INS73 006601	93/59	94/50	96/20	99/28	99/33	100/52	
INS7A 006575	96/16	99/08	100/17				
INS7V 006555	93/50	95/56	99/36	100/53			
INT? 006374	87/37	90/16	94/26	96/12			
INT?E 006477	93/49	94/57	95/60				
IPOC? 000210	9/11	79/41	95/27	95/40	99/84		
IPLBG 000301	10/08	38/12					
IPOC? 000211	9/12	81/26	81/28	81/30	81/32	81/35	100/29

0115 FPYDI

41/06