DataGeneral

DIAGNOSTIC LISTING

LISTING

096-000442-00

PROGRAM

6038/6039 FLOPPY DISK
RELIABILITY

TAPE

095-000442-00

ABSTRACT

THIS IS A PROGRAM DESIGNED TO TEST THE 6038/6039 FLOPPY DISK. THE PROGRAM IS SET UP TO HANDLE A MAXIMUM OF EIGHT ADAPTERS FOR A TOTAL OF SIXTEEN DRIVES (2/ADAPTERS). THE PROGRAM ASSUMES ONE ADAPTER WITH DEVICE CODE 33.

```
0001 FPYRL MACRO REV 04.00              15:42:16 12/03/76
01
02
03
04
05
06
07              ;*****************************************************************
08              ;
09              ; NAME: FPYRL.SR                              PART NUMBER: 094-000040
10              ;
11              ;
12              ; DESCRIPTION: 6038/6039 FLOPPY DISK RELIABILITY
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 FPYRL
01                      ;  6038/6039 FLOPPY DISK RELIABILITY
02                      ;-------------------------------------
03                      ;-------------------------------------
04
05
06              ; 1.0   ABSTRACT:
07              ;---------------
08
09              ; THIS IS A PROGRAM DESIGNED TO TEST THE 6038/6039 FLOPPY DISK.
10              ; THE PROGRAM IS SET UP TO HANDLE A MAXIMUM OF EIGHT ADAPPTERS
11              ; FOR A TOTAL OF SIXTEEEN DRIVES(2/ADAPTER).
12              ;       THE PROGRAM ASSUMES ONE ADAPTER WITH DEVICE CODE 33.
13              ; IF THIS IS OK A CARRIAGE RETURN SHOULD BE TYPED.
14              ; IF NOT,THE DESIRED DEVICE CODE OR CODES SHOULD BE TYPED
15              ; SEPARATED BY A SPACE AND TERMINATED BY A CARRIAGE RETURN.
16              ;       THE PROGRAM CONSISTS OF EIGHT BASIC TESTS,AND COMMAND
17              ; STRING INTERPRETER.
18
19
20
21              ; 1.1 INPUTS:
22              ;-------------
23              ;       ALL INPUTS TO THE PROGRAM ARE DONE VIA THE TERMINAL
24              ; KEYBOARD.THE PROGRAM ASSUMES TURNKEY SYSTEM WITH NO SWITCHS.
25              ;       THE SWITCH REGISTER IS INTERNAL TO THE PROGRAM AND
26              ; SIMULATES THE SWITCH CONSOL.
27              ; THE SETTING AND CLEARING OF BITS WILL BE EXPLAINED LATER.
28
29              ; 2.0 HARDWARE REQUIREMENTS:
30              ;---------------------------
31              ;       HARDWARE REQUIRED TO RUN THIS PROGRAM IS:
32              ; SINGLE OR DUAL DRIVE FLOPPY(6038/6039).
33              ; A CPU
34              ; A TERMINAL(TTY OR CRT&KEYBOARD).
35              ;  HIGH SPEED PTR IS OPTIONAL DEPENDING ON SET UP.
36
37
38
39              ; 3.0   OPERATION:
40              ;-----------------
41
42              ;       THE PROGRAM HAS TWO STARTING ADDRESSES (2 OR 200).
43              ; WHEN STARTED,THE PROGRAM SIZES AND REPORTS MEMORY
44              ; LIMIT. IT THEN REQUESTS FLOPPY DEVICE CODES,AT THIS
45              ; TIME UP TO EIGHT(8) DEVICE CODES MAY BE ENTERED
46              ; SEPERATED BY A SPACE AND TERMINATED BY A CARRIAGE
47              ; RETURN.
48              ; THE PROGRAM NOW REQUESTS THE UNITS TO BE TESTED.
49              ; AFTER THESE HAVE BEEN SELECTED SWITCH SETTINGS
50              ; ARE REQUESTED. THIS WILL BE DISCUSSED LATER.
51              ; AFTER THE SWITCH SETTINGS HAVE BEEN SELECTED THE
52              ; PROGRAM REQUESTS THE LOADING OF SCRATCH DISKETTES.
53              ; WHEN THE DISKETTES HAVE BEEN LOADED THE OPERATOR
54              ; TYPES A CR AND THE PROGRAM CONTINUES.
55              ;       THE PROGRAM NOW REQUESTS THE TESTS TO BE RUN.
56              ; A CARRIAGE RETURN WILL SELECT TESTS 1 - 8 TO BE RUN
57              ; IN SEQUENCE.
58              ; TYPING A NUMERIC CHARATER  1-8 WILL SELECT AND
59              ; LOCK ON THAT TEST.
60              ; WHEN LOCKED ON A TEST UNITS DO NOT CHANGE.
```

```
 0003 FPYRL
01              ; A NUMERIC 9 WILL SELECT THE INTERCHANGABILITY
02              ; TEST,AND ANY ALPHA CHARACTER WILL
03              ; PUT YOU IN COMMAND STRING.
04              ; BELOW IS A LIST OF THE TESTS AND THE TYPE OF DATA:
05              ; TEST 1        ALL ZEROS
06              ; TEST 2        ALL ONES
07              ; TEST 3        ALTERNATE ONES AN ZEROS
08              ; TEST 4        ALTERNATE ZEROS AN ONES
09              ; TEST 5        ALL SIXS
10              ; TEST 6        SEVEN ZERO PATTERN
11              ; TEST 7        RANDOM DATA
12              ; TEST 8        RANDOMLY ORGANIZED ONES AND ZEROS
13              ; TEST 9        INTERCHANGABILITY
14
15              ;       THE PROGRAM IS COMPLETLY RE-ENTERANT. IF AT
16              ; ANY TIME WHILE RUNNING THE FIRST EIGHT TESTS YOU
17              ; WANT TO RESTART: TYPE A CNTL R(THIS RESTARTS WITH
18              ; THE CURRENT SWITCH SETTINGS) OR A CNTL D (THIS
19              ; RESTARTS WITH THE SWITCHS SET TO ZERO).
20              ; INTERCHANGE TEST RESTARTS AUTOMATICALLY WHEN
21              ; COMPLETED.
22
23
```

```
 0004 FPYRL
01              ; 4.0   LOGICAL DISK UNITS
02              ;-----------------------------------
03
04              ;       LOGICAL DISKS WILL BE OF THE FORMAT DDU DD=DEVICE CODE
05              ;   U=UNIT NO.
06
07
08
09              ; 5.0   INPUT CODES
10              ;-------------------
11
12              ;       SWITCH SETTINGS ARE DONE VIA THE TTY CONSOL.THIS IS BECA
13              ;   THE PROGRAM ASSUMES NO SWITCH CONSOL.
14              ;   THE PROGRAM HAS AN INTERNAL SWITCH REGISTER WHICH IS USED I
15              ;   THE SAME WAY AS THE SWITCH CONSOL WOULD BE USED.
16              ;   BITS ARE SET AND CLEARED BY TYPING CERTIAN CHARACTERS ON TH
17              ;   KEYBOARD. THESE CHARACTERS ARE: 0,1,2,3,4,5,6,7,8,9,A,B,C,D
18              ;   THE PRECEEDING CHARACTERS ARE USED TO TOGGLE BITS 0-15 IN T
19              ;   SWITCH REGISTER WHICH CORRESPOND TO SWITCHES 0-15 RESPECTIV
20              ;       "@" IS USED TO LOCK INTO THE INPUT ROUTINE SO THAT MORE
21              ;   ONE SWITCH SETTING CAN BE CHANGED AT ONE TIME.
22              ;       INDIVIDUAL SWITCHES(1-15) CAN BE CHANGED VIA THE INTERRU
23              ;   BY TYPING THE CORRESPONDING CHARACTER FOR THAT BIT(SWITCH).
24              ;       TYPING AN "M" WILL TYPE OUT THE CURRENT SWITCH SETTINGS.
25              ;   TWO OTHER FUNCTIONS ARE "CNTRL R"-WHICH WILL SAVE THE
26              ;   CURRENT SWITCH SETTINGS AND RESTART THE PROGRAM,AND "CNTRL
27              ;   WHICH CLEARS THE SWITCH REGISTERAND RESTARTS THE PROGRAM.
28              ;       BELOW IS A TABLE OF THE KEYS AND THEIR FUNCTION:
```

```
10005 FPYRL
01
02              ;       KEY             FUNCTION
03              ;-------------------------------------------------------------
04              ;       CNTRL D         CLEAR SWITCH REGISTER=RESTART PROGAM
05              ;       CNTRL R         SAVE SWITCH REGISTER=RESTART PROGRAM
06              ;       0               TOGGLE SWITCH 0=LOCK IN INPUT ROUTINE
07              ;       1               TOGGLE SWITCH 1=LOOP ON ERROR=0
08              ;       2               TOGGLE SWITCH 2=PRINT ON CONSOL=0
09              ;       3               TOGGLE SWITCH 3=PRINT ERRORS=0
10              ;       4               TOGGLE SWITCH 4=PRINT SUMMARY=0
11              ;       5               TOGGLE SWITCH 5=PRINT TO LINE PRINTER=1
12              ;       6                "       "    6
13              ;       7                "       "    7
14              ;       8                "       "    8
15              ;       9                "       "    9
16              ;       A                "       "    10
17              ;       B                "       "    11
18              ;       C                "       "    12
19              ;       D                "       "    13
20              ;       E                "       "    14
21              ;       F                "       "    15
22
23              ;    WHEN LOCKED IN INPUT,A "CR" WILL GET YOU OUT.
24
25
26
27
28              ;6.0  COMMAND STRING:
29              ;---------------------
30              ;       THE PROGRAM HAS A COMMAND STRING INTERPRETER.
31              ; IT IS ENTERED BY TYPING ANY ALPHA CHARATER WHEN TEST#
32              ; IS REQUESTED.
33              ; WHEN COMMAND STRING IS ENTERED IT REQUESTS UNIT TO
34              ; BE TESTED. A ZERO OR A ONE ARE THE ONLY RESPONSES
35              ; ALLOWED.
36              ; DATA TYPE IS NOW ASKED FOR:THE OPERATOR MAY SELECT ANY
37              ; ONE OF THE TYPES LISTED BY TYPING ITS CORRESPONING
38              ; NUMBER FOLLOWED BY A CARRIAGE RETURN.
39              ;       NOW THE COMMAND STRING MAY BE ENTERED.
40              ; THE COMMANDS ALLOWED ARE AS FOLLOWS:
41              ; WRITE READ SEEK RECAL RESTART LOOP LR.
42              ;       THE WRITE COMMAND IS RESTRICTED TO WRITING
43              ; OF ONE SECTOR OR ALL SECTORS. THIS IS BECAUSE OF
44              ; HARDWARE CONSTAINTS.
45              ;       THE RESTART COMMAND WILL TAKE YOU OUT OF
46              ; COMMAND STRING AND BACK TO THE START OF THE PROGRAM.
47              ;       THE FORMAT FOR COMMAND STRING IS:
48              ;
49              ; SEEK 56 WRITE 1 1 READ 1 1 RECAL LOOP
50              ;
51              ; *** NOTE ***
52              ; A SEEK COMMAND MUST BE FOLLOWED BY SOME OTHER COMMAND
53              ; SUCH AS READ WRITE ETC.A SEEK CAN NOT BE TERMINATED
54              ; WITH A CARRIAGE RETURN.
55
56              ; THE LOOP COMMAND ALLOWS LOOPING ON THE COMMAND STRING
57              ; OR THE PORTION OF THE STRING PRECEDING THE LOOP COMMAND.
58              ;EXITING FROM THE LOOP IS DONE BY HITTING A KEY ON THE
59              ; KEYBOARD.
60              ;  WHEN DATA IS REQUESTED THE RESPONSE IS A NUMBER
```

```
 0006 FPYRL
01              ; FROM 1-6. BELOW IS LISTED THE CORRESPONDING PATTERNS:
02              ;
03              ;       1= ALL ZEROS
04              ;       2= ALL ONES
05              ;       3= 125252
06              ;       4= 52525
07              ;       5= 66666
08              ;       6= 70707
09
10
11              ; 7.0   ERRORS
12              ;---------------
13              ;       WHEN ERRORS OCURR THEY ARE REPORTED IMMEDIATLY.
14              ;    THE PROGRAM THEN WILL GO BACK TO THE BEGINNING OF
15              ;    FAILING TEST AND RUN IT AGAIN. THIS WILL NOT OCURR,
16              ;    HOWEVER,IF SWITCH "1" IS SET.THE PROGRAM WILL PROCEED
17              ;    TO THE NEXT TEST.
18
```

```
                        .TITL FPYRL



          000401 .DUSR   NOP=401
          000033 .DUSR   FPY=33
          000402 .DUSR   SKIP=402
```



```



00000 005465             DIRT
00001 002441             INTR
00002 000200             JMP      .STRT
      000020             .LOC     20
00020 000000 AUTO:       0
00021 000000 AUTO1:      0
00022 000000 AUTO2:      0
      000045             .LOC     45
00045 005457 EGGS:       .EGGS
      000100             .LOC     100
00100 003772 ITYP?E:     TYP?E
00101 003554 .MESS:      MES?S
00102 003631 .CRLF:      CRL?F
00103 000246 .DTAB:      DTAB
00104 003641 .ZOCT:      ZOC?T
00105 004072 .TINO:      TIN?O
00106 004177 IINP?:      INP?J
00107 004413 .ENTR:      ENTER
00110 002023 .TSTC:      TSTC
00111 004431 .CYCL:      CYCLE
00112 002377 .RAN:       RAND
00113 002246 .DCHK:      DCHK
00114 005503 .LAST:      LAST
00115 005503 BUFF:       LAST
00116 000000 SWREG:      0
00117 001652 .STP:       HSTP
00120 001361 .DVR:       FDVR
00121 001723 .SCHK:      SCHK
00122 001663 .DARD:      RDDA
00123 003663 .PRTD:      PDE?C
00124 003645 .POCT:      POC?T
00125 002141 .RCAL:      RCAL
00126 001676 .VADD:      VADD
00127 003653 IPDC?S:     PDC?S
00130 002516 .DCHN:      DCHNG
00131 001465 .TCHK:      TCHK
00132 001331 .TESTX:     TESTX
00133 002161 .WADN:      WADN
00134 002202 .RDY:       RDY
00135 002651 .WRT:       WRT
00136 002132 .GEN:       GEN
00137 004076 .TIND:      TIN?D
00140 001274 .SEEK:      SEEK
00141 002122 .DGEN:      DGEN
00142 004562 .CMS:       CMDST
00143 002661 .PTAB:      PTAB
00144 002053 .GT43:      GT43
00145 005173 IINP:       INP
00146 005007 IHSS:       HSS
00147 005066 IGATM:      GATM
00150 001531 IEXEC:      EXEL
00151 004544 ISRH:       SRH
00152 001452 STAD:       STAK-1
00153 004634 IGCS:       GCS
00154 000000 TERM:       0
00155 000000 CSIF:       0
00156 000033 C33:        33
```

```
 0009 FPYRL
01 00157 000002 C2:     2
02 00160 012752 UBP:    UBUFF+UBUFF
03 00161 012776 CSBP:   CBUFF+CBUFF
04 00162 000000 LINCT:  0
05 00163 000000 NCHG:   0
06 00164 000000 TEST:   0
07 00165 177777 TCSW:   -1
08 00166 000000 TCON:   0
09 00167 000000 DCSW:   0
10 00170 000000 MPTR:   0
11 00171 000000 G4KSW:  0
12 00172 000000 RWS:    0
13 00173 000000 LKS:    0
14 00174 000000 NOCH:   0
15 00175 000000 PASS:   0
16       000200         .LOC    200
17 00200 002201 .STRT:  JMP     .+1
18 00201 000451         INIT
19 00202 000000 ESWI:   0
20 00203 000000 ECNT:   0
21 00204 000000 UNIT:   0
22 00205 000000 DSKA:   0
23 00206 000000 NADD:   0
24 00207 000000 SWRD:   0
25 00210 000000 TRK:    0
26 00211 000000 SEC:    0
27 00212 000000 SC:     0
28 00213 000000 CSIS:   0
29 00214 000000 AC3?:   0
30 00215 000000 NSEC:   0
31 00216 000377 P.377:  377
32 00217 000060 P.C60:  60
33 00220 000012 P.C12:  12
34 00221 000015 P.C15:  15
35 00222 000067 C67:    67
36 00223 000001 TSTN:   1
37 00224 000111 I:      "I
38 00225 000127 W:      "W
39 00226 000122 R:      "R
40 00227 000054 T43:    44.
41 00230 000033 DEVC:   33
42 00231 000115 LTRK:   77.
43 00232 000177 TMSK:   177
44 00233 000007 SMSK:   7
45 00234 000000 RBCK:   0
46 00235 000010 RPOP:   1012
47 00236 000040 SP:     40
48 00237 007777 K4:     7777
49 00240 177777 ENINT:  -1
50 00241 125252 DATA3:  125252
51 00242 052525 DATA4:  52525
52 00243 066666 DATA5:  66666
53 00244 070707 DATA6:  70707
54 00245 111627 KEY:    111627
55 00246 000010 DTAB:   .BLK    8.
```

```
!0010 FPYRL
01        000101 .DUSR    IMES?S=.MESS
02        000102 .DUSR    ICRL?F=.CRLF
03        000104 .DUSR    IZOC?T=.ZOCT
04        000105 .DUSR    ITIN?O=.TINO
05        000123 .DUSR    IPDE?C=.PRTD
06        000124 .DUSR    IPOC?T=.POCT
```

```
10011 FPYRL
01        000450          .LOC    450
02
03
04                ;THIS IS THE INITIALIZATION PORTION OF THE PROGRAM.
05
06 000450 005464          .EGGS+5
07 000451 022777 INIT:    LDA     0,0,-1          ;PICK UP SWITCH REGISTER INFO
08 000452 044116          STA     0,SWREG         ;;PUT IT IN THE SWITCH REGISTER
09 000453 006101          JSR     @.MESS          ;PRINT PROGRAM NAME
10 000454 003140          NAME                    ;"FLOPPY DISK RELIABILITY:REV.-"
11 000455 006102 SIZE:    JSR     @.CRLF
12 000456 020114          LDA     0,.LAST         ;GET LAST PROGRAM ADDRESS
13 000457 101400          INC     0,0             ;BUMP IT BY ONE
14 000460 040170          STA     0,MPTR          ;SET UP POINTER
15 000461 022170 NXTM:    LDA     0,@MPTR         ;READ MEMORY LOCATION
16 000462 105000          MOV     0,1             ;PUT DATA IN WORKING REGISTER
17 000463 124000          COM     1,1             ;COMPLEMENT THE DATA
18 000464 046170          STA     1,@MPTR         ;WRITE IT BACK
19 000465 032170          LDA     2,@MPTR         ;READ IT AGAIN
20 000466 132414          SUB#    1,2,SZR         ;WAS DATA READ SAME AS WRITTEN
21 000467 000406          JMP     EMEM            ;NO  SET UP END OF MEMORY
22 000470 042170          STA     0,@MPTR         ;YES  RESTORE ORIGINAL DATA
23 000471 010170          ISZ     MPTR            ;BUMP THE POINTER
24 000472 020170          LDA     0,MPTR
25 000473 101133          MOVZL#  0,0,SNC
26 000474 000765          JMP     NXTM            ;CHECK NEXT LOCATION
27 000475 014170 EMEM:    DSZ     MPTR            ;SET POINTER TO LAST LOCATION
28 000476 152000          ADC     2,2             ;GENERATE A MINUS 1
29 000477 050165          STA     2,TCSW          ;SET TIME CONSTANT SWITCH
30 000500 020237          LDA     0,K4            ;GET THE 4K CONSTANT
31 000501 024170          LDA     1,MPTR          ;GET THE MEMORY POINTER
32 000502 122432          SUBZ#   1,0,SZC         ;IS THERE MORE THAN 4K OF MEMORY
33 000503 000402          JMP     PMEM            ;NO      PRINT MEMORY SIZE
34 000504 050171          STA     2,G4KSW         ;YES     SET GREATER THAN 4K SWIT
35 000505 006101 PMEM:    JSR     @.MESS          ;PRINT MEMORY SIZE
36 000506 003166          MEMRY                   ;"HIGEST MEMORY ADDRESS= "
37 000507 024170          LDA     1,MPTR          ;GET THE ADDRESS
38 000510 006104          JSR     @.ZOCT          ;PRINT IT
39 000511 006102          JSR     @.CRLF          ;DO RETURN LINE FEED
40 000512 020103 RES?T:   LDA     0,.DTAB         ;GET DEVICE TABLE ADDRESS
41 000513 040021          STA     0,AUTO1         ;SET UP TO ACCEPT MULTIPLE DEVIC
42 000514 014021          DSZ     AUTO1
```

```
10012 FPYRL
01 000515 006101 BADI:    JSR     @.MESS          ;QUERY AS TO DEVICE CODES
02 000516 003202          DEVCD                   ;"FLOPPY DEVICE CODE=33, OK?"
03 000517 006105 NCOD:    JSR     @.TINO          ;GET OPERATOR INPUT
04 000520 000775          JMP     BADI            ;BAD INPUT TRY AGAIN
05 000521 030236          LDA     2,SP            ;GET SPACE CODE
06 000522 112414          SUB#    0,2,SZR         ;WAS TERMINATOR A SPACE
07 000523 000403          JMP     CKCK            ;NO      CHECK FOR RETURN
06 000524 040021          STA     1,@AUTO1        ;YES     SAVE DEVICE CODE
09 000525 000772          JMP     NCOD            ;GO GET NEXT CODE
10 000526 030220 CKCK:    LDA     2,P.C12         ;GET RETURN CODE
11 000527 112414          SUB#    0,2,SZR         ;WAS TERMINATOR A RETURN
12 000530 000765          JMP     BADI            ;NO      TRY AGAIN
13 000531 125015          MOV#    1,1,SNR         ;IS IT ZERO
14 000532 000763          JMP     BADI            ;YES     BAD INPUT
15 000533 046021          STA     1,@AUTO1        ;NO      SAVE DEVICE
16 000534 152000          ADC     2,2
17 000535 052021          STA     2,@AUTO1        ;TERMINATE THE TABLE
18 000536 050167          STA     2,DCSW          ;SET DEVICE CHANGER ENABLE
19 000537 102440 UREQ:    SUBO    0,0
20 000540 040213          STA     0,CSIS          ;RESET CSI SWITCH
21 000541 006101          JSR     @.MESS
22 000542 003357          TSTU
23 000543 006105          JSR     @.TINO
24 000544 000773          JMP     UREQ
25 000545 030220          LDA     2,P.C12
26 000546 112414          SUB#    0,2,SZR
27 000547 000770          JMP     UREQ
28 000550 125015          MOV#    1,1,SNR
29 000551 000411          JMP     BUN
30 000552 125222          MOVZR   1,1,SZC
31 000553 000404          JMP     UN0
32 000554 125223          MOVZR   1,1,SNC
33 000555 000762          JMP     UREQ
34 000556 125201          MOVR    1,1,SKP
35 000557 126400 UN0:     SUB     1,1
36 000560 044204          STA     1,UNIT
37 000561 126001          ADC     1,1,SKP
38 000562 044204 BUN:     STA     1,UNIT
39 000563 044174          STA     1,NOCM
40 000564 006101          JSR     @.MESS          ;REQUEST SWITCH SETTINGS
41 000565 003221          SETSW                   ;"INPUT SWITCH SETTINGS--HIT @ F
42 000566 102400          SUB     0,0             ;CLEAR AC0
43 000567 040213          STA     0,CSIS
44 000570 040163          STA     0,NCMG
45 000571 040166          STA     0,TCON
46 000572 040210          STA     0,TRK
47 000573 040175          STA     0,PASS          ;CLEAR PASS COUNTER
48 000574 040202          STA     0,ESWT          ;CLEAR ERROR SWITCH
49 000575 040203          STA     0,ECNT          ;CLEAR ERROR COUNTER
50 000576 101400          INC     0,0
51 000577 040223          STA     0,TSTN
52 000600 063610          SKPDN   TTI             ;DO WE HAVE INPUT
53 000601 000777          JMP     .-1             ;NO      WAIT
54 000602 006106          JSR     @IINPT          ;YES     GO GET IT
55 000603 006101          JSR     @.MESS          ;REQUEST SCRATCH DISKETTES
56 000604 003244          LOAD                    ;"LOAD SCRATCH DISKETTES: HIT A
57 000605 006105          JSR     @.TINO          ;WAIT FOR OPERATOR
58 000606 006102          JSR     @.CRLF
59 000607 006101 TSEL:    JSR     @.MESS          ;REQUEST TEST
60 000610 003440          TNUM                    ;TO BE RUN
```

```
#013 FPYNL
01 000611 006137          JSR     @.TIND          ;GET OPERATOR INPUT
02 000612 000407          JMP     .CMD            ;ALPHA! GO TO CSI
03 000613 125005          MOV     1,1,SNR         ;RUN ALL TESTS?
04 000614 000410          JMP     ALLT            ;YES! DO IT
05 000615 044223          STA     1,TSTN          ;NO! SAVE TEST NUMBER
06 000616 102000          ADC     0,0             ;GENERATE MINUS 1
07 000617 044163          STA     0,NCHG          ;SET NO TEST CHANGE SWITCH
08 000620 000405          JMP     ONET
09 000621 006102 .CMD:    JSR     @.CRLF
10 000622 024142          LDA     1,.CMS
11 000623 002405          JMP     SAVT
12 000624 125400 ALLT:    INC     1,1             ;BUMP AC1 TO 1
13 000625 034505 ONET:    LDA     3,TTAB          ;GET THE TEST TO BE RUN
14 000626 137000          ADD     1,3
15 000627 025400          LDA     1,0,3
16 000630 044164 SAVT:    STA     1,TEST          ;SET UP POINTER
17 000631 010175 REDO:    ISZ     PASS
18 000632 020103          LDA     0,.DTAB         ;GET DEVICE TABLE ADDRESS
19 000633 040022          STA     0,AUTO2         ;SET UP AUTO POINTER
20 000634 014022          DSZ     AUTO2           ;INITIALIZE POINTER
21 000635 020167 DCTM:    LDA     0,DCSW          ;GET DEVICE CHANGER SWITCH
22 000636 101005          MOV     0,0,SNR         ;IS IT SET
23 000637 000416          JMP     CONG            ;NO GO DO TESTS
24 000640 022022          LDA     0,@AUTO2        ;GET THE DEVICE CODE
25 000641 101152          MOVZL#  0,0,SZC         ;IS IT THE TERMINATOR
26 000642 000767          JMP     REDO            ;YES GO RESET POINTER
27 000643 006130          JSR     @.DCHN          ;GO CHANGE THE DEVICE NUMBER
28 000644 000230          DEVL
29 000645 000655          CONG
30 000646 004645          GCS.1
31 000647 054230          STA     3,DEVC          ;SAVE NEW DEVICE CODE
32 000650 006101          JSR     @.MESS
33 000651 003353          PASSN
34 000652 024175          LDA     1,PASS
35 000653 006123          JSR     @.PRTD
36 000654 006102          JSR     @.CRLF
37 000655 020165 CONG:    LDA     0,TCSW          ;GET TIME  CONSTANT SWITCH
38 000656 101015          MOV#    0,0,SNR         ;IS IT SET
39 000657 000450          JMP     GOOD            ;NO: GO TO TESTS
40 000660 010165          ISZ     TCSW            ;SET SWITCH TO ZERO
41 000661 020204          LDA     0,UNIT
42 000662 061033          DOA     0,FPY
43 000663 006125          JSR     @.RCAL          ;RECAL THE DRIVE
44 000664 102440          SUBO    0,0             ;CLEAR AC0
45 000665 040206          STA     0,NADD          ;SET UP FOR T0,S0
46 000666 006126          JSR     @.VADD          ;VERIFY ADDRESS
47 000667 024115          LDA     1,BUFF          ;GET BUFFER ADDRESS
48 000670 060033          DOB     1,FPY           ;SET UP FLOPPY
49 000671 024204          LDA     1,UNIT          ;GET UNIT NUMBER
50 000672 101700          INCS    0,0             ;SET UP FOR SEC. 1
51 000673 123000          ADD     1,0             ;ADD SEC. TO UNIT
52 000674 024235          LDA     1,RPOP          ;GET READ PREAMBLE
53 000675 125120          MOVZL   1,1             ;GENERATE A READ OPCODE
54
55 000676 123000          ADD     1,0             ;COMPLETE THE COMMAND
56 000677 061133          DOAS    0,FPY           ;SEND AND EXECUTE COMMAND
57 000700 063733 FDNC:    SKPDZ   FPY             ;FLOPPY DONE?
58 000701 000406          JMP     FDON            ;YEHS   GO COMPUTE CONSTANT
59 000702 010166          ISZ     TCON            ;NO! BUMP COUNTER
60 000703 000775          JMP     FDNC            ;GO CHECK AGAIN
```

```
#014 FPYNL
01 000704 006101          JSR     @.MESS          ;PRINT MESSAGE
02 000705 073444          NOFPY                   ;NO FLOPPY IN SYSTEM
03 000706 063077          HALT
04 000707 020166 FDON:    LDA     0,TCON          ;GET COUNT
05 000710 105220          MOVZR   0,1             ;DIVIDE BY 4
06 000711 125220          MOVZR   1,1
07 000712 107000          ADD     0,1             ;ADD COUNT
08 000713 044166          STA     1,TCON          ;SAVE TIMMING CONSTANT
09 000714 030115          LDA     2,BUFF          ;GET BUFFER ADDRESS
10 000715 021000          LDA     0,0,2           ;GET FIRST WORD IN BUFFER
11 000716 025001          LDA     1,1,2           ;GET SECOND WORD
12 000717 123000          ADD     1,0             ;ADD THEM
13 000720 024245          LDA     1,KEY           ;GET THE KEY WORD
14 000721 122434          SUBZ#   1,0,SZR         ;IS IT THE DOOS KEY
15 000722 000405          JMP     GOOD            ;NO: GO DO TESTS
16 000723 006101          JSR     @.MESS          ;YES! INFORM OPERATOR
17 000724 003512          DUMB
18 000725 006105          JSR     @.TINO          ;WAIT FOR RESPONSE
19 000726 000401          NOP
20 000727 020204 GOOD:    LDA     0,UNIT          ;GET UNIT NUMBER
21 000730 061033          DOA     0,FPY           ;SELECT DRIVE
22 000731 002164          JMP     @TEST           ;RUN TESTS
23
24 000732 000732 TTAB:    .
25 000733 000744          TEST1           ;ALL ZEROS
26 000734 000772          TEST2           ;ALL ONES
27 000735 001006          TEST3           ;125252
28 000736 001022          TEST4           ;52525
29 000737 001036          TEST5           ;66666
30 000740 001052          TEST6           ;70707
31 000741 001066          TEST7           ;RANDOM
32 000742 001102          TEST8           ;FLOATING ONES&ZEROS
33 000743 002563          INCH            ;INTERCHANGE TEST
```

```
10015 FPYRL
01                ;TESTS START HERE. TESTX ONLY EXECUTES IF MORE THAN 4K OF
02                ;MEMORY IS AVAILABLE,BECAUSE IT MOVES BUFFERS AROUND MEMORY.
03
04        020241 .DUSR            DGEN3=LDA       0,DATA3
05        020242 .DUSR            DGEN4=LDA       0,DATA4
06        020243 .DUSR            DGEN5=LDA       0,DATA5
07        020244 .DUSR            DGEN6=LDA       0,DATA6
08
09
10 00744 006101 TEST1:  JSR     @.MESS
11 00745 003333         DNUM
12 00746 020230         LDA     0,DEVC
13 00747 103120         ADDZL   0,0
14 00750 101120         MOVZL   0,0
15 00751 024204         LDA     1,UNIT
16 00752 125120         MOVZL   1,1
17 00753 125100         MOVL    1,1
18 00754 107000         ADD     0,1
19 00755 006104         JSR     @.ZOCT
20 00756 006102         JSR     @.CRLF
21 00757 006101         JSR     @.MESS
22 00760 003440         TNUM
23 00761 024223         LDA     1,TSTN
24 00762 006104         JSR     @.ZOCT
25 00763 006102         JSR     @.CRLF
26 00764 006107         JSR     @.ENTR          ;SET UP FOR LOOP
27 00765 060233         NIOC    FPY
28 00766 006125         JSR     @.RCAL
29 00767 006110         JSR     @.TSTC          ;RUN TEST
30 00770 102400         SUB     0,0             ;DATA PATTERN INSTRUCTION
31 00771 006111         JSR     @.CYCL
32
33 00772 006131 TEST2:  JSR     @.TCHK
34 00773 006132         JSR     @.TESTX
35 00774 006101         JSR     @.MESS
36 00775 003440         TNUM
37 00776 024223         LDA     1,TSTN
38 00777 006104         JSR     @.ZOCT
39 01000 006102         JSR     @.CRLF
40 01001 006107         JSR     @.ENTR          ;SET UP FOR LOOP
41 01002 006125         JSR     @.RCAL
42 01003 006110         JSR     @.TSTC          ;RUN    TEST
43 01004 102000         ADC     0,0             ;DATA PATTERN INSTUCTION
44 01005 006111         JSR     @.CYCL
45
46 01006 006131 TEST3:  JSR     @.TCHK
47 01007 006132         JSR     @.TESTX
48 01010 006101         JSR     @.MESS
49 01011 003440         TNUM
50 01012 024223         LDA     1,TSTN
51 01013 006104         JSR     @.ZOCT
52 01014 006102         JSR     @.CRLF
53 01015 006107         JSR     @.ENTR          ;SET UP FOR LOOP
54 01016 006125         JSR     @.RCAL
55 01017 006110         JSR     @.TSTC          ;RUN TEST
56 01020 020241         DGEN3                   ;DATA PATTERN INSTUCTION
57 01021 006111         JSR     @.CYCL
58
59 01022 006131 TEST4:  JSR     @.TCHK
60 01023 006132         JSR     @.TESTX
```

```
 0016 FPYRL
01 01024 006101         JSR     @.MESS
02 01025 003440         TNUM
03 01026 024223         LDA     1,TSTN
04 01027 006104         JSR     @.ZOCT
05 01030 006102         JSR     @.CRLF
06 01031 006107         JSR     @.ENTR          ;SET UP FOR LOOP
07 01032 006125         JSR     @.RCAL
08 01033 006110         JSR     @.TSTC          ;RUN TEST
09 01034 020242         DGEN4                   ;DATA PATTERN INSTRUCTION
10 01035 006111         JSR     @.CYCL
11
12 01036 006131 TEST5:  JSR     @.TCHK
13 01037 006132         JSR     @.TESTX
14 01040 006101         JSR     @.MESS
15 01041 003440         TNUM
16 01042 024223         LDA     1,TSTN
17 01043 006104         JSR     @.ZOCT
18 01044 006102         JSR     @.CRLF
19 01045 006107         JSR     @.ENTR          ;SET UP FOR LOOP
20 01046 006125         JSR     @.RCAL
21 01047 006110         JSR     @.TSTC          ;RUN TEST
22 01050 020243         DGEN5                   ;DATA PATTERN INSTUCTION
23 01051 006111         JSR     @.CYCL
24
25 01052 006131 TEST6:  JSR     @.TCHK
26 01053 006132         JSR     @.TESTX
27 01054 006101         JSR     @.MESS
28 01055 003440         TNUM
29 01056 024223         LDA     1,TSTN
30 01057 006104         JSR     @.ZOCT
31 01060 006102         JSR     @.CRLF
32 01061 006107         JSR     @.ENTR          ;SET UP FOR LOOP
33 01062 006125         JSR     @.RCAL
34 01063 006110         JSR     @.TSTC          ;RUN TEST
35 01064 020244         DGEN6                   ;DATA INSTRUCTION
36 01065 006111         JSR     @.CYCL
37
38 01066 006131 TEST7:  JSR     @.TCHK
39 01067 006132         JSR     @.TESTX
40 01070 006101         JSR     @.MESS
41 01071 003440         TNUM
42 01072 024223         LDA     1,TSTN
43 01073 006104         JSR     @.ZOCT
44 01074 006102         JSR     @.CRLF
45 01075 006107         JSR     @.ENTR          ;SET UP FOT LOOP
46 01076 006125         JSR     @.RCAL
47 01077 006110         JSR     @.TSTC          ;RUN TEST
48 01100 006112         JSR     @.RAN           ;DATA INSTRUCTION
49 01101 006111         JSR     @.CYCL
50
```

```
;W17 FPYRL
                ;THIS TEST WRITES A BINARY COUNT PATTERN ON THE DISK
                ;THEN DOES RANDOM READS AND WRITES OF RANDOM TRACKS
                ;AND SECTORS. IT LOOPS THROUGH 100 TIMES.

01162 006131 TESTB:  JSR     .TCHK
01163 006132         JSR     .TESTX
01164 006101         JSR     .MESS
01165 003440         TNUM
01166 024223         LDA     1,TSTN
01167 006123         JSR     .PRTD
01170 006102         JSR     .CRLF
01171 006107         JSR     .ENTR          ;SET UP FOR LOOP
01172 006125         JSR     .RCAL
01173 020115         LDA     0,BUFF         ;GET BUFFER ADDRESS
01174 044020         STA     0,AUTO         ;SET UP AUTO POINTER
01175 014020         DSZ     AUTO           ;AND INITIATE
01176 102440         SUBO    0,0            ;GENERATE FIRST WORD
01177 042020 BNFL:   STA     0,@AUTO        ;PUT DATA IN BUFFER
01200 101704         INCS    0,0,SZR        ;GENERATE A 2 BYTE
01201 101401         INC     0,0,SKP        ;BINARY COUNT PATTERN
01202 000402         JMP     BFUL           ;BUFFER FULL
01203 000774         JMP     BNFL           ;BUFFER NOT FULL,CONTINUE
01204 020235 BFUL:   LDA     0,RPOP         ;GET READ PREAMBLE OPCODE
01205 103120         ADDZL   0,0            ;COMPUTE A WRITE OPCODE
01206 040406         STA     0,DVRP         ;SAVE IT FOR DRIVER
01207 152000         ADC     2,2            ;GENERATE -1
01210 050172         STA     2,RWS          ;SET READ/WRITE SWITCH FOR WRITE
01211 020231         LDA     0,LTRK         ;GET NUMBER OF LAST TRACK
01212 040534         STA     0,CNTR         ;SET UP COUNTER
01213 006120 WRNT:   JSR     @.DVR          ;GO WRITE A TRACK
01214 000000 DVRP:   0                      ;OP CODE
01215 014531         DSZ     CNTR           ;DONE DISK?
01216 000403         JMP     NEWT           ;NO-MOVE TO NEXT TRACK
01217 006125         JSR     .RCAL          ;YES-HOME THE HEAD
01220 000406         JMP     GOTS           ;GO START RANDOM TEST
01221 010210 NEWT:   ISZ     TRK
01222 006144         JSR     .GT43
01223 006117         JSR     @.STP          ;GO STEP THE HEAD
01224 000002 STOP:   2                      ;HEAD MOVE OP CODE
01225 000766         JMP     WRNT           ;GO WRITE ANOTHER TRACK
01226 020236 GOTS:   LDA     0,SP           ;GET A CONSTANT
01227 101100         MOVL    0,0            ;MULTIPLY BY 2
01230 040516         STA     0,CNTR         ;SET UP COUNTER
01231 024233         LDA     1,SMSK         ;GET THE SECTOR MASK
01232 030232         LDA     2,TMSK         ;GET THE TRACK MASK
01233 151300         MOVS    2,2            ;MOVE TRACK TO LEFT BYTE
01234 127120         ADDZL   1,1
01235 147000         ADD     2,1            ;MAKE COMPOSITE MASK
01236 044511         STA     1,CMSK         ;SAVE IT TILL LATER
01237 006112 NEWN:   JSR     @.RAN          ;GET A RANDOM NUMBER
01240 024235         LDA     1,RPOP         ;GET READ PREAMBLE OPCODE
01241 101133         MOVZL#  0,0,SNC        ;DO WE READ OR WRITE
01242 152001         ADC     2,2,SKP        ;GENERATE -1 FOR WRITE
01243 152441         SUBO    2,2,SKP        ;GENERATE 0 FOR READ
01244 125120         MOVZL   1,1            ;GENERATE OPCODE FOR
01245 125120         MOVZL   1,1            ;READ OR WRITE
```

```
;W16 FPYRL
01166 050172         STA     2,RWS          ;SET READ/WRITE SWITCH
01167 044476         STA     1,OPCD         ;SAVE THE OPCODE
01170 024477         LDA     1,CMSK         ;GET THE COMPOSITE MASK
01171 123400         AND     1,0            ;PULL OUT TRACK AND SECTOR
01172 024233         LDA     1,SMSK         ;GET THE SECTOR MASK
01173 127120         ADDZL   1,1
01174 107400         AND     0,1            ;PULL OUT SECTOR
01175 131000         MOV     1,2
01176 125220         MOVZR   1,1
01177 125220         MOVZR   1,1
01200 125400         INC     1,1
01201 044462         STA     1,RSEC         ;SAVE SECTOR NO.
01202 101300         MOVS    0,0            ;MOVE TRACK TO RIGHT BYTE
01203 024232         LDA     1,TMSK         ;GET TRACK MASK
01204 107400         AND     0,1            ;PULL OUT TRACK NO.
01205 020464         LDA     0,MTRK
01206 122433         SUBZ#   1,0,SNC
01207 100420         SUB     0,1
01210 044210         STA     1,TRK          ;SAVE TRACK NO.
01211 125300         MOVS    1,1
01212 133000         ADD     1,2
01213 050206         STA     2,NADD
01214 020204         LDA     0,UNIT         ;GET UNIT NO.
01215 024446         LDA     1,RSEC         ;GET SECTOR NO.
01216 125300         MOVS    1,1            ;FORMAT THE SECTOR NO.
01217 123000         ADD     1,0            ;START COMMAND GENERATION
01220 024445         LDA     1,OPCD         ;GET THE OPCODE
01221 123000         ADD     1,0            ;COMPLETE THE COMMAND
01222 040442         STA     0,CMD          ;SAVE TTHE COMMAND WORD
01223 006140         JSR     @.SEEK
```

```
10019 FPYRL
01 01224 006126 DOIT:   JSR     @.VADD          ;GO VERIFY ADDRESS
02 01225 102520         SUBZL   0,0             ;COMPUTE A CONSTANT
03 01226 101300         MOVS    0,0             ;OF 400 OCTAL
04 01227 024172         LDA     1,RWS           ;GET THE READ/WRITE SWITCH
05 01230 030115         LDA     2,BUFF          ;GET THE BUFFER ADDRESS
06 01231 101005         MOV     0,0,SNR         ;IS IT A WRITE
07 01232 113000         ADD     0,2             ;SET UP READ BUFFER
08 01233 072033         DOB     2,FPY           ;SEND BUFFER ADDRESS TO FLOPPY
09 01234 020430         LDA     0,OCMD          ;GET THE COMMAND WORD
10 01235 006134         JSR     @.RDY           ;CHECK FLOPPY READY
11 01236 061133         DOAS    0,FPY           ;EXECUTE IT
12 01237 050234         STA     2,RBCK          ;SAVE BUFFER ADDRESS FOR DATA CH
13 01240 006133         JSR     @.WADN
14 01241 006121         JSR     @.SCHK          ;GO CHECK STATUS
15 01242 020172         LDA     0,RWS           ;GET READ/WRITE SWITCH
16 01243 101005         MOV     0,0,SNR         ;IS IIT A WRITE
17 01244 006113         JSR     @.DCHK          ;NO=GO CHECK DATA
18 01245 014421         DSZ     CNTR            ;DONE 100 TIMES
19 01246 000711         JMP     NEWN            ;NO=DO MORE
20 01247 006132         JSR     @.TESTX
21 01250 006131         JSR     @.TCHK
22 01251 020163         LDA     0,NCHG          ;GET NO CHANGE SWITCH
23 01252 101014         MOV#    0,0,SZR         ;IS IT SET
24 01253 006111         JSR     @.CYCL          ;GO TO CYCLE
25 01254 020174         LDA     0,NOCH
26 01255 101004         MOV     0,0,SZR
27 01256 000404         JMP     SURT
28 01257 020204         LDA     0,UNIT
29 01260 101132         MOVZL#  0,0,SZC
30 01261 002411         JMP     @TENT
31 01262 002411 SURT:   JMP     @DCENT
32
33 01263 000000 RSEC:   0
34 01264 000000 OCMD:   0
35 01265 000000 OPCD:   0
36 01266 000000 CNTR:   0
37 01267 000000 CMSK:   0
38 01270 000000 STEP:   0
39 01271 000114 MTRK:   76.
40 01272 000744 TENT:   TEST1
41 01273 000635 DCENT:  DCTM
```

```
10020 FPYRL
01 01274 054434 SEEK:   STA     3,SKR
02 01275 006134         JSR     @.RDY           ;CHECK FLOPPY READY
03 01276 006122         JSR     @.DARD          ;GO READ DISK ADDRESS
04 01277 024232         LDA     1,TMSK          ;GET THE TRACK MASK
05 01300 030210         LDA     2,TRK           ;GET THE DESIRED TRACK
06 01301 101300         MOVS    0,0             ;MOVE TRACK TO RIGHT BYTE
07 01302 123400         AND     1,0             ;PULL TRACK NO..
08 01303 142415         SUB#    2,0,SNR         ;IS THIS DESIRED TRACK
09 01304 002424         JMP     @SKR            ;YES=GO EXECUTE THE COMMAND
10 01305 142432         SUBZ#   2,0,SZC         ;NO=WHICH WAY TO STEP
11 01306 000406         JMP     STPO            ;STEP TOWARD ZERO
12 01307 112400         SUB     0,2             ;SET UP TO STEP AWAY
13 01310 050760         STA     2,STEP          ;SET UP STEP COUNTER
14 01311 102520         SUBZL   0,0             ;GENERATE STEP AWAY OPCODE
15 01312 101400         INC     0,0
16 01313 000404         JMP     DOST1           ;GO STEP HEAD
17 01314 142400 STPO:   SUB     2,0             ;COMPUTE DIFFERENCE
18 01315 040753         STA     0,STEP          ;SET UP COUNTER
19 01316 102520         SUBZL   0,0             ;GENERATE STEP TO
20 01317 040402 DOST1:  STA     0,STPC          ;SAVE OPCODE
21 01320 006117 DOST:   JSR     @.STP           ;GO MOVE THE HEAD
22 01321 000000 STPC:   0                       ;OPCOCDE
23 01322 014746         DSZ     STEP            ;DONE STEPPING?
24 01323 000775         JMP     DOST            ;NO=DO AGAIN
25 01324 020204         LDA     0,UNIT
26 01325 061133         DOAS    0,FPY
27 01326 006133         JSR     @.WADN
28 01327 002401         JMP     @SKR
29
30
31 01330 000000 SKR:    0
```

```
10021 FPYRL
01
02 01331 054425 TESTX:  STA     3,TURN
03 01332 020171         LDA     0,G4KSW         ;GET THE MORE THAN 4K INDICCATOR
04 01333 101004         MOV     0,0,SZR         ;IS IT SET
05 01334 000402         SKIP                    ;YES    MOVE THE BUFFERS
06 01335 002421 TXIT:   JMP     @TURN           ;NO     EXIT
07 01336 020114 MOVM:   LDA     0,.LAST         ;GET LAST PROGRAM ADDRESS
08 01337 040420         STA     0,FAD           ;SAVE IT AS LOW LIMIT
09 01340 020170         LDA     0,MPTR          ;GET LAST MEMORY ADDRESS
10 01341 126520         SUBZL   1,1             ;MOVE IT DOWN
11 01342 125300         MOVS    1,1             ;2000 OCTAL
12 01343 127120         ADDZL   1,1             ;LOCATIONS TO
13 01344 122400         SUB     1,0             ;PROTECT DTOS AREA
14 01345 040413         STA     0,HMAD          ;SAVE AS UPPER LIMIT
15 01346 006112 NNUM:   JSR     @.RAN           ;GET A RANDOM NUMBER
16 01347 024410         LDA     1,FAD           ;GET LOW LIMIT
17 01350 030410         LDA     2,HMAD          ;AND UPPER LIMIT
18 01351 106063         ADCZ#   0,1,SNC         ;IS NUMBER
19 01352 112433         SUBZ#   0,2,SNC         ;WITHIN LIMITS
20 01353 000773         JMP     NNUM            ;NO     GET A NEW NUMBER
21 01354 040115         STA     0,BUFF          ;YES    SAVE THE NEW BUFFEER
22 01355 000760         JMP      TXIT
23
24
25 01356 000000 TURN:   0
26 01357 000000 FAD:    0
27 01360 000000 HMAD:   0
```

```
10022 FPYRL
01              ;THIS ROUTINE GENERATES A COMMAND STACK FOR THE DESIRED
02              ;OPERATION AND EXECUTES IT FOR A COMPLETE TRACK.
03
04 01361 025400 FDVR:   LDA     1,0,3           ;PICK UP OP CODE FOR READ/WRITE
05 01362 175400         INC     3,3             ;BUMP THE RETURN POINTER
06 01363 054464         STA     3,DVRR          ;SAVE IT
07 01364 020204         LDA     0,UNIT          ;GET THE UNIT NUMBER
08 01365 123000         ADD     1,0             ;GENERATE PARTIAL COMMAND
09 01366 040462         STA     0,PCOM          ;SAVE TILL LATER
10 01367 020152         LDA     0,STAD          ;GET STACK ADDRESS
11 01370 040021         STA     0,AUTO1         ;INITIALIZE THE POINTER
12 01371 000401         NOP
13 01372 006122         JSR     @.DARD          ;WHERE ARE WE ON THE DISK
14 01373 040206         STA     0,NADD          ;SAVE DISK ADDRESS
15 01374 024233         LDA     1,SMSK          ;GET THE SECTOR MASK
16 01375 101200         MOVR    0,0             ;ALLIGN THE SECTOR
17 01376 101200         MOVR    0,0             ;IN THE LOWER 3 BITS
18 01377 123400         AND     1,0             ;AND PULL IT OUT
19 01400 101400         INC     0,0
20 01401 176440         SUBO    3,3             ;COMPUTE A
21 01402 175520         INCZL   3,3             ;CONSTANT OF 2
22 01403 054446         STA     3,GCNT          ;INITIALIZE GROUP COUNTER
23 01404 152520         SUBZL   2,2             ;INITIALIZE COMMAND COUNTER
24 01405 000407         JMP     FCOM            ;GO DO FIRST COMMAND
25 01406 151400 NXTC:   INC     2,2             ;BUMP THE COMMAND COUNTER
26 01407 024172         LDA     1,RWS           ;GET THE READ/WRITE SWITCH
27 01410 020442         LDA     0,LSEC          ;GET THE LAST SECTOR NUMBER
28 01411 125005         MOV     1,1,SNR         ;IS IT SET
29 01412 101401         INC     0,0,SKP         ;YES    BUMP SECTOR FOR READ
30 01413 163000         ADD     3,0             ;YES    BUMP SECTOR BY 2 FOR WRI
31 01414 040436 FCOM:   STA     0,LSEC          ;SAVE SECTOR NUMBER
32 01415 101000         MOVS    0,0             ;FORMAT SECTOR FOR COMMAND
33 01416 024432         LDA     1,PCOM          ;GET  THE PARTIIAL COMMAND
34 01417 123000         ADD     1,0             ;COMPLETE THE COMMAND
35 01420 042021         STA     0,@AUTO1        ;PUT IT IN THE STACK
36 01421 175100         MOVL    3,3             ;CHANGE CONSTANT FOR COUNT CHECK
37 01422 156415         SUB#    2,3,SNR         ;ARE WE DONE THIS GROUP
38 01423 000403         JMP     NXTG            ;YES    CHECK FOR DONE ALL
39 01424 175200         MOVR    3,3             ;RESTOORE OLD CONSTANT
40 01425 000761         JMP     NXTC            ;DO NEXT COMMAND
41 01426 014423 NXTG:   DSZ     GCNT            ;DECREMENT COUNTER-SKIP IF DONE
42 01427 000405         JMP     NDON            ;NOT DONE--SET UP FOR NEXT GROUP
43 01430 102000         ADC     0,0             ;DONE--GENERATE TERMINATOR
44 01431 042021         STA     0,@AUTO1        ;TERMINATE THE STACK
45 01432 006150         JSR     @IEXEC          ;EXECUTE THE COMMAND STACK
46 01433 002414         JMP     @DVRR           ;EXIT
```

```
10023 FPYRL
01 01434 024172 NDON:   LDA     1,RWS           ;GET READ/WRITE SWITCH
02 01435 125005         MOV     1,1,SNR         ;IS IT SET
03 01436 000406         JMP     NXTN            ;NO     SET FOR MORE READS
04 01437 020204         LDA     0,UNIT          ;GET UNIT NUMBER
05 01440 024235         LDA     1,NPOP          ;GET SECTOR BREAK
06 01441 123000         ADD     1,0             ;COMPUTE COMMAND
07 01442 042021         STA     0,@AUTO1                ;PUT IT IN THE STACK
08 01443 010407         ISZ     LSEC
09 01444 175200 NXTN:   MOVR    3,3
10 01445 152520         SUBZL   2,2             ;RESET COUNTER
11 01446 000741         JMP     NXTC+1          ;FINISH
12
13
14
15 01447 000000 DVRN:   0
16 01450 000000 PCOM:   0
17 01451 000000 GCNT:   0
18 01452 000000 LSEC:   0
19 01453 000012 STAK:   .BLK    10.
20
21
22 01465 054425 TCHK:   STA     3,TSR
23 01466 044421         STA     0,TS0
24 01467 044421         STA     1,TS1
25 01470 054421         STA     2,TS2
26 01471 020163         LDA     0,NCHG          ;GET NO CHANGE SWITCH
27 01472 101014         MOV#    0,0,SZR         ;IS IT SET
28 01473 000407         JMP     TEX             ;YES GET OUT
29 01474 020223         LDA     0,TSTN          ;GET THE TEST NUMBER
30 01475 024411         LDA     1,LTST          ;GET NUMBER OF LAST TEST
31 01476 106414         SUB#    0,1,SZR         ;ARE WE DONE ALL TESTS
32 01477 101401         INC     0,0,SKP         ;NO-BUMP TEST NUMBER
33 01500 004413         JSR     UNCK            ;YES-GO CHANGE UNIT NO.
34 01501 040223         STA     0,TSTN          ;SAVE THE TEST NO.
35 01502 030407 TEX:    LDA     2,TS2
36 01503 024405         LDA     1,TS1           ;RESTORE THE AC'S
37 01504 020403         LDA     0,TS0
38 01505 002405         JMP     @TSR            ;EXIT
39
40 01506 000010 LTST:   10
41 01507 000000 TS0:    0
42 01510 000000 TS1:    0
43 01511 000000 TS2:    0
44 01512 000000 TSR:    0
45
46 01513 054415 UNCK:   STA     3,USAV          ;SAVE RETURN POINTER
47 01514 102520         SUBZL   0,0
48 01515 040223         STA     0,TSTN          ;SET TEST NO. TO 1
49 01516 020174         LDA     0,NOCH
50 01517 101004         MOV     0,0,SZR
51 01520 000406         JMP     NRTN
52 01521 020204         LDA     0,UNIT          ;GET UNIT NO.
53 01522 101133         MOVZL#  0,0,SNC         ;IS IT UNIT ONE
54 01523 101241         MOVOR   0,0,SKP         ;NO-SET IT TO ONE
55 01524 102440         SUBO    0,0             ;YES-SET IT TO ZERO
56 01525 040204         STA     0,UNIT          ;SET UP UNIT NO.
57 01526 014402 NRTN:   ISZ     USAV            ;BUMP THE POINTER
58 01527 002401         JMP     @USAV           ;EXIT
59
60 01530 000000 USAV:   0
```

```
0024 FPYRL
01
02              ;THIS ROUTINE EXECUTES THE COMMAND STACK, CHECKS STATUS AFTER
03              ;EACH OPERATION, CHECKS DATA ON PREVIOUS READ AFTER START OF
04              ;NEXT READ.
05
06
07 01531 054517 EXEC:   STA     3,XRTN          ;SAVE THE RETURN ADDRESS
08 01532 102520         SUBZL   0,0
09 01533 101540         INCOL   0,0
10 01534 040515         STA     0,XCNT
11 01535 102440         SUBO    0,0
12 01536 040173         STA     0,LRS           ;CLEAR LAST READ SWITCH
13 01537 040202         STA     0,ESWT
14 01540 020152         LDA     0,STAD          ;GET THE STACK ADDRESS
15 01541 040021         STA     0,AUTO1         ;SET UP POINTER
16 01542 020115         LDA     0,BUFF          ;GET BUFFER ADDRESS
17 01543 024172         LDA     1,RWS           ;GET RWS
18 01544 125004         MOV     1,1,SZR         ;IS IT A READ OPERATION
19 01545 000411         JMP     SNDB            ;NO     SET UP FOR WRITE
20 01546 126000         ADC     1,1             ;SET FIRST
21 01547 044500         STA     1,FRSW          ;READ SWITCH
22 01550 125400         INC     1,1             ;COMPUTE 400 OCTAL
23 01551 125700         INCS    1,1             ;FOR READ BUFFER COMPUTATION
24 01552 123000         ADD     1,0             ;COMPUTE ADDRESS READ BUFFER 1
25 01553 040471         STA     0,RB1           ;SAVE IT
26 01554 123000         ADD     1,0             ;COMPUTE ADDRESS READ BUFFER 2
27 01555 040470         STA     0,RB2           ;SAVE IT
28 01556 062033 SNDB:   DOB     0,FPY           ;SEND BUFFER ADDRESS TO FLOPPY
29 01557 020021         LDA     1,@AUTO1        ;GET A COMMAND FROM STACK
30 01560 125415         INC#    1,1,SNR         ;IS THE TERMINATOR
31 01561 000450         JMP     XCMP            ;YES     GO TO COMPLETION
32 01562 040464         STA     0,COMB          ;NO      SAVE BUFFER ADDRESS
33 01563 000134         JSR     @.RDY           ;CHECK DISK READY
34 01564 065133         DOAS    1,FPY           ;SEND COMMAND-SRART OPERATION
35 01565 024172         LDA     1,RWS           ;GET THE READ/WRITE SWITCH
36 01566 125004         MOV     1,1,SZR         ;IS IT A READ
37 01567 000406         JMP     ONEX            ;NO      WAIT FOR COMPLETION
38 01570 010457         ISZ     FRSW            ;IS IT THE FIIRST READ
39 01571 000113         JSR     @.DCHK          ;NO      GO CHECK LAST BUFFER
40 01572 020202         LDA     0,ESWT
41 01573 101004         MOV     0,0,SZR
42 01574 000442         JMP     RERN
43 01575 000133 ONEX:   JSR     @.WADN
44
45 01576 000121         JSR     @.SCHK          ;GO CHECK STATUS
46 01577 020172         LDA     0,RWS           ;CHECK RWS
47 01600 101004         MOV     0,0,SZR         ;WAS IT A WRITE
48 01601 000410         JMP     WREX            ;YES CONTINUE WRITE
49 01602 020213         LDA     0,CSIS          ;GET CSI SWITCH
50 01603 101015         MOV#    0,0,SNR         ;IS IT SET
51 01604 000412         JMP     SNRD            ;NO      SET UP NEXT READ
52 01605 020215         LDA     0,NSEC          ;GET NUMBER OF SECTORS
53 01606 101235         MOVZR#  0,0,SNR         ;MORE THAN ONE SECTOR
54 01607 000416         JMP     XCMP1           ;NO GET OUT
55 01610 000406         JMP     SNRD            ;YES SET UP NEXT READ
56 01611 020202 WREX:   LDA     0,ESWT
57 01612 101014         MOV#    0,0,SZR
58 01613 000423         JMP     RERN
59 01614 020115         LDA     0,BUFF          ;GET BUFFER ADDRESS
60 01615 002741         JMP     SNDB            ;DO NEXT OPERATION
```

```
0025 FPYRL
01 01616 020427 SNRD:   LDA     0,RB2           ;GET READ BUFFER 2
02 01617 024427         LDA     1,COMB          ;GET LAST BUFFER USED
03 01620 044234         STA     1,RBCK          ;SAVE FOR DATA CHECKING
04 01621 122434         SUBZ#   1,1,SZR         ;WAS LAST BUFFER NO.2
05 01622 000734         JMP     SNDB            ;NO     USE IT
06 01623 020421         LDA     0,RB1           ;YES    USE BUFFER NO.1
07 01624 000732         JMP     SNDB            ;GO DO IT
08 01625 020420 XCMP1:  LDA     0,RB2           ;GET READ BUFFER TWO
09 01626 040234         STA     0,RBCK          ;SET UP FOR CHECKING
10 01627 126000         AUC     1,1
11 01630 044173         STA     1,LRS           ;SET LAST READ SWITCH
12 01631 024172 XCMP:   LDA     1,RWS           ;CHECK READ/WRITE SWITCH
13 01632 125004         MOV     1,1,SZR                 ;WAS IT A READ
14 01633 000402         JMP     EEXIT           ;NO     RETURN TO CONTROL
15 01634 000113         JSR     @.DCHK          ;CHECK LAST BUFFER
16 01635 002413 EEXIT:  JMP     @XRTN           ;EXIT
17
18
19 01636 006126 RERN:   JSR     @.VADD
20 01637 014412         DSZ     XCNT
21 01640 000675         JMP     EXEC+4
22 01641 006101         JSR     @.MESS
23 01642 003545         WRTER
24 01643 000772         JMP     EEXIT
25
26
27 01644 000000 RB1:    0
28 01645 000000 RB2:    0
29 01646 000000 COMB:   0
30 01647 000000 RWS:    0
31 01650 000000 XRTN:   0
32 01651 000000 XCNT:   0
```

```
10026 FPYRL
01
02 01652 021400 RSTP:   LDA     0,0,3           ;PICK UP STEP OPCODE
03 01653 175400         INC     3,3             ;BUMP RETURN ADDRESS
04 01654 054405         STA     3,STPR          ;SAVE FOR RETURN
05 01655 024204         LDA     1,UNIT          ;GET UNIT NUMBER
06 01656 123000         ADD     1,0             ;SET UP STEP COMMAND
07 01657 061133         DOAS    0,FPY           ;STEP THE HEAD
08 01660 006133         JSR     @.WADN
09 01661 002401         JMP     @STPR           ;RETURN
10
11 01662 000000 STPR:   0
12
13
14 01663 054412 RDDA:   STA     3,RDAR          ;SAVE RETURN ADDRESS
15 01664 006134         JSR     @.RDY
16 01665 024204         LDA     1,UNIT          ;GET UNIT NO.
17 01666 020235         LDA     0,RPOP          ;OP CODE FOR READ PREAMBLE
18 01667 107000         ADD     0,1             ;COMPUTE COMMAND
19 01670 065133         DOAS    1,FPY           ;SEND AND EXECUTE
20 01671 006133         JSR     @.WADN
21 01672 062433         DIC     0,FPY           ;BRING IN THE ADDRESS
22 01673 040205         STA     0,DSKA          ;SAVE IT
23 01674 002401         JMP     @RDAR           ;EXIT
24
25 01675 000000 RDAR:   0
26
27
28 01676 054420 VADD:   STA     3,VADR          ;SAVE RETURN ADDRESS
29 01677 040420         STA     0,VAD0
30 01700 044420         STA     1,VAD1
31 01701 020420         LDA     0,VAD10
32 01702 040420         STA     0,VADC
33 01703 004760 VRDA:   JSR     RDDA
34 01704 024206         LDA     1,NADD          ;GET  DESIRED ADDRESS
35 01705 106415         SUB#    0,1,SNR         ;OK
36 01706 000405         JMP     VEDA
37 01707 014413         DSZ     VADC
38 01710 000773         JMP     VRDA
39 01711 006101         JSR     @.MESS
40 01712 003426         ADDNF
41 01713 020404 VEDA:   LDA     0,VAD0
42 01714 024404         LDA     1,VAD1
43 01715 002401         JMP     @VADR           ;EXIT
44
45 01716 000000 VADR:   0
46 01717 000000 VAD0:   0
47 01720 000000 VAD1:   0
48 01721 000010 VAD10:  10
49 01722 000000 VADC:   0
```

```
0027 FPYRL
01
02                ;THIS ROUTINE READS STATUS,SAVES IT,AND DOES A BIT BY BIT
03                ;CHECK WHEN AN ERROR IS DETECTED.
04
05 01723 054474 SCHK:   STA     3,SRTN          ;SAVE RETURN ADDRESS
06 01724 060633 RDFS:   DIAC    0,FPY           ;READ THE STATUS
07 01725 044207         STA     0,SWRD          ;SAVE IT
08 01726 101233 ECHK:   MOVZRW  0,0,SNC         ;ANY ERRORS
09 01727 002470         JMP     @SRTN           ;NOPE GET OUT
10 01730 011203         ISZ     ECNT
11 01731 126000         ADC     1,1             ;YUP    SET THE
12 01732 044202         STA     1,ESWT          ;ERROR SWITCH
13 01733 006101         JSR     @.MESS          ;REPORT THAT AN ERROR EXISTS
14 01734 003024         STATE                   ;"STATUS ERROR"
15 01735 024223         LDA     1,TSTN          ;GET THE TEST NUMBER
16 01736 006123         JSR     @.PRTD          ;AND PRINT IT
17 01737 006101         JSR     @.MESS
18 01740 003417         TWKN
19 01741 024210         LDA     1,TRK
20 01742 006104         JSR     @.ZOCT
21 01743 006101         JSR     @.MESS
22 01744 003423         SECN
23 01745 024223         LDA     1,TSTN
24 01746 032453         LDA     2,@T8
25 01747 132415         SUBW    1,2,SNR
26 01750 002407         JMP     T8P
27 01751 014021         DSZ     AUTO1
28 01752 026021         LDA     1,@AUTO1
29 01753 030233         LDA     2,SMSK
30 01754 125300         MOVS    1,1
31 01755 147400         AND     2,1
32 01756 000402         JMP     .+2
33 01757 026443 T8P:    LDA     1,@T8S
34 01760 006104         JSR     @.ZOCT
35 01761 006102         JSR     @.CRLF          ;DO RETURN/LINE FEED
36 01762 024436         LDA     1,EMSK          ;GET ERROR MASK
37 01763 020207         LDA     0,SWRD
38 01764 107400         AND     0,1             ;PULL OUT ERROR BITS
39 01765 125224         MOVZR   1,1,SZR         ;ARE ANY BITS SET
40 01766 000404         JMP     WFLT            ;YES    CHECK THEM
41 01767 006101         JSR     @.MESS          ;REPORT NO DATA RECIEVED
42 01770 003000         NODAT
43 01771 000425         JMP     EEXT            ;EXIT
44 01772 125223 WFLT:   MOVZR   1,1,SNC         ;IS IT A WRITE FAULT
45 01773 000403         JMP     DCHL
46 01774 006101         JSR     @.MESS          ;REPORT WRITE FAULT
47 01775 003037         WRFLT
48 01776 125223 DCHL:   MOVZR   1,1,SNC         ;IS IT A DATA CHANNEL LATE
49 01777 000403         JMP     CHKW
50 02000 006101         JSR     @.MESS          ;REPORT DATA CHANNEL LATE
51 02001 003046         DCHLT
52 02002 125223 CHKW:   MOVZR   1,1,SNC         ;IS IT A CHECK WORD
53 02003 000403         JMP     SERR
54 02004 006101         JSR     @.MESS          ;REPORT CHECKWORD ERROR
55 02005 003076         CHKWD
56 02006 125223 SERR:   MOVZR   1,1,SNC         ;IS IT A SECTOR
57 02007 000403         JMP     ICMD
58 02010 006101         JSR     @.MESS          ;REPORT SECTOR ERROR
59 02011 003110         SECER
60 02012 125223 ICMD:   MOVZR   1,1,SNC         ;IS IT ILLEGAL CMD
```

```
 0028 FPYRL
01 02013 000403         JMP     EEXT
02 02014 006101         JSR     @.MESS          ;REPORT ILLEGAL COMMAND
03 02015 003124         ILCMD
04 02016 002401 EEXT:   JMP     @SRTN           ;OTHERWISE CONTINUE
05
06 02017 000000 SRTN:   0
07
08 02020 000077 EMSK:   77
09 02021 015006 T8:     LTST
10 02022 001203 T8S:    RSEC
```

```
10029 FPYRL
01
02
03                ;THIS ROUTINE CONTROLS TEST EXECTUTION.
04
05 02023 054462 TSTC:   STA     3,TSTR          ;SAVE  RETURN ADDRESS
06 02024 020231         LDA     0,LTRK          ;GET LAST TRK
07 02025 040461         STA     0,TCNT          ;SET UP COUNTER
08 02026 021400         LDA     0,0,3           ; GET DATA INSTRUTION
09 02027 040503         STA     0,GEN           ;SAVE IT
10 02030 004457 NEXC:   JSR     TEXC            ;START TEST EXECUTIIN
11 02031 000401         NOP
12 02032 010210         ISZ     TRK             ;BUMP THE TRK
13 02033 004420         JSR     GT43
14 02034 014452 GNEX:   DSZ     TCNT
15 02035 000404         JMP     NTRK            ;DO ANOTHER TRRACK
16 02036 006125         JSR     @.RCAL          ;DONE: RECALIBRATE THE DRIVE
17 02037 010446         ISZ     TSTR            ;BUMP RETURN POINTER
18 02040 002445         JMP     @TSTR           ;EXIT
19 02041 006117 NTRK:   JSR     @.STP           ;STEP THE HEAD
20 02042 000002 SAZ:    2                       ;AWAY FROM ZERO
21 02043 006122         JSR     @.DARD          ;GET DISK ADDRESS
22 02044 024252         LDA     1,TMSK          ;GET TRACK MASK
23 02045 101300         MOVS    0,0             ;PUT TRACK NO. IN LOW BYTE
24 02046 123400         AND     1,0             ;PULL OUT TRACK NO.
25 02047 024210         LDA     1,TRK           ;GET TRACK EXPECTED
26 02050 122414         SUB#    1,0,SZR         ;IS IT THE RIGHT TRACK
27 02051 000430         JMP     TERR            ;NO! REPORT ERROR
28 02052 000756         JMP     NEXC            ;CONTINUE
29
30
31
32 02053 054425 GT43:   STA     3,GT43R
33 02054 020210         LDA     0,TRK
34 02055 024227         LDA     1,T43           ;GET THE TRACK CONSTANT
35 02056 106032         ADCZ#   0,1,SZC         ;IS IT > 43
36 02057 000411         JMP     LCRC            ;NO     GO DO MORE
37 02060 020264         LDA     0,UNIT          ;GET UNIT NO.
38 02061 103132         ADDZL#  0,0,SZC         ;IS BIT SET
39 02062 000414         JMP     GRET            ;YES GET OUT
40 02063 101133         MOVZL#  0,0,SNC         ;CHECK FOR UNIT ONE
41 02064 102621         SUBZR   0,0,SKP         ;NOT UNIT ONE SET 0
42 02065 101241         MOVOR   0,0,SKP         ;UNIT ONE:SET BIT ONE
43 02066 101200         MOVR    0,,0            ;RESTORE UNIT NO.
44 02067 000407         JMP     GRET            ;GET OUT
45 02070 020264 LCRC:   LDA     0,UNIT          ;GET UNIT #
46 02071 103133         ADDZL#  0,0,SNC         ;IS BIT SET
47 02072 000404         JMP     GRET            ;NO GET OUT
48 02073 101132         MOVZL#  0,0,SZC         ;YES WHICH UNIT
49 02074 102621         SUBZR   0,0,SKP         ;1=CLEAR LC BIT
50 02075 102440         SUBO    0,0             ;0=CLEAR LC BIT
51 02076 040264 GRET:   STA     0,UNIT          ;SAVE UNIT NO.
52 02077 002401         JMP     @GT43R          ;GO FINISH DISK
53
54 02100 000000 GT43R:  0
55
56
57
58 02101 006101 TERR:   JSR     @.MESS          ;PRINT MESSAGE
59 02102 003470         TRKER                   ;"NO HEAD STEP"
60 02103 063077         HALT
```

```
 0030 FPYRL
01 02104 000735         JMP     NTRK            ;TRY STEPPING AGAIN
02
03
04 02105 000000 TSTR:   0
05 02106 000000 TCNT:   0
06
```

```
!0031 FPYRL
01
02
03 02107 054407 TEXC:   STA     3,TEXR          ;SAVE RETURN ADDRESS
04 02110 004412          JSR     DGEN            ;GO GENERATE THE DATA
05 02111 102000          ADC     0,0             ;SET R/W SWITCH FOR WRITE
06 02112 040172          STA     0,RWS           ;
07 02113 006120          JSR     @.DVR           ;GO DO IT
08 02114 000040 WRITE:   40
09 02115 010172          ISZ     RWS
10 02116 000000 TEXR:    0
11 02117 006120          JSR     @.DVR           ;DO THE READ
12 02120 000020 READ:    20
13 02121 002775          JMP     @TEXR           ;EXIT
14
15              ;THIS ROUTINE GENERATES THE DATA FOR WRITING.
16
17 02122 054416 DGEN:    STA     3,GENR          ;SAVE RETURN ADDRESS
18 02123 126440          SUBO    1,1
19 02124 125700          INCS    1,1             ;GENERATE CONSTANT
20 02125 124400          NEG     1,1             ;FOR BUFFER COUNTER
21 02126 044411          STA     1,DCNT          ;SET UP COUNTER
22 02127 020115          LDA     0,BUFF          ;GET ADDRESS OF BUFFER
23 02130 040020          STA     0,AUTO          ;SET UP POINTER
24 02131 014020          DSZ     AUTO
25 02132 000000 GEN:     0
26 02133 042020          STA     0,@AUTO         ;FILL THE BUFFER
27 02134 010403          ISZ     DCNT            ;BUMP THE COUNTEER
28 02135 000775          JMP     GEN             ;DO SOME MORE
29 02136 002402          JMP     @GENR           ;DONE-EXIT
30
31 02137 000000 DCNT:    0
32 02140 000000 GENR:    0
```

```
!0032 FPYRL
01
02
03              ;THIS ROUTINE RECALIBRATES THE FLOPPY, POSITIONS HEAD ON
04              ;TRACK 00.
05
06 02141 054417 RCAL:    STA     3,RRTN          ;SAVE  RETURN ADDRESS
07 02142 102440          SUBO    0,0
08 02143 044210          STA     0,TRK           ;SET TRACK = 0
09 02144 006134          JSR     @.RDY
10 02145 060633 HOME:    DIAC    0,FPY
11 02146 103123          ADDZL   0,0,SNC         ;IS HEAD HOME
12 02147 000406          JMP     STPZ            ;NO     SET UP TO MOVE HEAD
13 02150 020204          LDA     0,UNIT          ;GET THE UNIT NO.
14 02151 126620          SUBZR   1,1             ;GENERATE UNIT MASK
15 02152 123400          AND     1,0             ;MASK OUT LOW CURRENT
16 02153 040204          STA     0,UNIT          ;BIT
17 02154 002404          JMP     @RRTN           ;EXIT
18 02155 006117 STPZ:    JSR     @.STP           ;GO STEP HEAD
19 02156 000001 SIZ:     1
20 02157 000766          JMP     HOME            ;GO CHECK STATUS
21
22 02160 000000 RRTN:    0
23
24
25
26
27 02161 054415 WADN:    STA     3,WADNR
28 02162 040415          STA     0,DN0
29 02163 020166          LDA     0,TCON
30 02164 040414          STA     0,TIMR
31 02165 063733 DONE:    SKPDZ   FPY
32 02166 000406          JMP     WEXIT
33 02167 014411          DSZ     TIMR
34 02170 000775          JMP     DONE
35 02171 006101          JSR     @.MESS
36 02172 063402          TOUT
37 02173 002405          JMP     @TOER
38 02174 020403 WEXIT:   LDA     0,DN0
39 02175 002401          JMP     @WADNR
40
41 02176 000000 WADNR:   0
42 02177 000000 DN0:     0
43 02200 000000 TIMR:    0
44 02201 000632 TOER:    REDO+1
```

```
10033 FPYRL
01
02                ;THIS ROUTINE IS USED TO LOAD HEADS,OR AS A 10MS WAIT
03                ;AFTER A HEAD STEP OPERATION.
04
05 02202 054441 RDY:    STA     3,STLR          ;SAVE RETURN ADDRESS
06 02203 040437         STA     0,IVAD
07 02204 020441         LDA     0,SCON          ;GET SETTLE CONSTANT
08 02205 040437         STA     0,STLC          ;SET UP COUNTER
09 02206 060633         DIAC    0,FPY           ;GET STATUS WORD
10 02207 101122         MOVZL   0,0,SZC         ;READY?
11 02210 000424         JMP     NTR             ;NO GO WAIT
12 02211 103123         ADDZL   0,0,SNC         ;HEAD LOADED?
13 02212 000402         JMP     SETA            ;NO GO LOAD HEAD
14 02213 000411         JMP     REX             ;YES EXIT
15 02214 020204 SETA:   LDA     0,UNIT          ;GET THE UNIT NUMBER
16 02215 061133         DOAS    0,FPY           ;SEND TO FLOPPY
17 02216 006133         JSR     @.WADN
18 02217 060633         DIAC    0,FPY
19 02220 101122         MOVZL   0,0,SZC
20 02221 000413         JMP     NTR
21 02222 103123         ADDZL   0,0,SNC
22 02223 000403         JMP     SETN
23 02224 020416 REX:    LDA     0,IVAD
24 02225 002416         JMP     @STLR           ;EXIT
25
26 02226 014416 SETN:   DSZ     STLC            ;DECREMENT THE COUNTER
27 02227 000765         JMP     SETA            ;GO TRY AGAIN
28 02230 006101         JSR     @.MESS          ;PRINT ERROR MESSAGE
29 02231 003457         NOHLD                   ;"NO HEAD LOADED
30 02232 063077         HALT
31 02233 000761         JMP     SETA            ;CONTINUE
32
33 02234 006101 NTR:    JSR     @.MESS          ;PRINT MESSAGE
34 02235 003012         NRDY                    ;FLOPPY NOT READY
35 02236 060633 WAIT:   DIAC    0,FPY
36 02237 101132         MOVZL#  0,0,SZC         ;READY YET
37 02240 000776         JMP     WAIT            ;NO WAIT
38 02241 000753         JMP     SETA            ;YES GO CHECK HEAD LOAD
39
40
41
42 02242 000000 IVAD:   0
43 02243 000000 STLR:   0
44 02244 000000 STLC:   0
45 02245 000031 SCON:   25.
46
47
48                ;THIS ROUTTINE DOES DATA CHECKING AFTER A READ OPERATION.
49
50 02246 054523 DCHK:   STA     3,DRTN          ;SAVE RETURN ADDRESS
51 02247 020115         LDA     0,BUFF          ;GET ADDRESS OF WRITE BUFFER
52 02250 040516         STA     0,PNTR1         ;SET UP POINTER 1
53 02251 020254         LDA     0,RBCK          ;GET READ BUFFER TO BE CHECKED
54 02252 040515         STA     0,PNTR2         ;SET UP POINTER 2
55 02253 102440         SUBO    0,0
56 02254 040521         STA     0,DECT          ;CLEAR DATA ERROR COUNTER
57 02255 040521         STA     0,FERR          ;CLEAR FIRST ERROR SWITCH
58 02256 101700         INCS    0,0             ;GENERATE CONSTANT FOR COUNTER
59 02257 040511         STA     0,WRDC          ;SET UP WORD COUNTER
60 02260 022506 NXTW:   LDA     0,@PNTR1                ;GET WORD WRITTEN
```

```
 0034 FPYRL
01 02261 026506         LDA     1,@PNTR2                ;GET WORD READ
02 02262 122414         SUB#    1,0,SZR         ;DATA OK?
03 02263 000416         JMP     DERR            ;BAD DATA,REPORT
04 02264 126440 DNXT:   SUBO    1,1
05 02265 046502         STA     1,@PNTR2
06 02266 010500         ISZ     PNTR1           ;BUMP POINTER
07 02267 010502         ISZ     PNTR2           ;BUMP POINTER 2
08 02270 014500         DSZ     WRDC            ;DONE?
09 02271 000767         JMP     NXTW            ;DO NEXT WORD
10 02272 030213         LDA     2,CSIS          ;GET THE CSI SWITCH
11 02273 151014         MOV#    2,2,SZR         ;IS IT SET
12 02274 000404         JMP     DEXT            ;YES GET OUT
13 02275 030202         LDA     2,ESWT          ;CHECK ERROR SWITCH
14 02276 151004         MOV     2,2,SZR         ;IS IT SET
15 02277 006111         JSR     @.CYCL                  ;GO LOOP ON ERRORS
16 02300 002471 DEXT:   JMP     @DRTN           ;NO ERRORS-EXIT
17 02301 152000 DERR:   ADC     2,2
18 02302 050202         STA     2,ESWT          ;SET ERROR SWITCH
19 02303 010203         ISZ     ECNT
20 02304 040466         STA     0,GDAT          ;SAVE GOOD DATA
21 02305 044466         STA     1,BDAT          ;SAVE BAD DATA
22 02306 010467         ISZ     DECT            ;BUMP ERROR COUNTER
23 02307 020466         LDA     0,DECT          ;GET THE COUNTER
24 02310 126440         SUBO    1,1
25 02311 125540         INCOL   1,1             ;GENERATE A CONSTANT OF 3
26 02312 106432         SUBZ#   0,1,SZC         ;HAVE WE HAD THREE ERRORS
27 02313 000405         JMP     PERR            ;NO     PRINT ERROR
28 02314 030116         LDA     2,SWREG         ;READ THE SWITCH REGISTER
29 02315 151200         MOVR    2,2             ;CHECK FOR PRINT ALL ERRORS
30 02316 151223         MOVZR   2,2,SNC         ;IS SWITCH 14 SET
31 02317 000745         JMP     DNXT            ;GO CHECK NEXT WORD
32 02320 020456 PERR:   LDA     0,FERR
33 02321 101004         MOV     0,0,SZR
34 02322 000430         JMP     PERR1
35 02323 102000         ADC     0,0
36 02324 040452         STA     0,FERR
37 02325 006101         JSR     @.MESS
38 02326 003337         BUFA
39 02327 024115         LDA     1,BUFF
40 02330 006104         JSR     @.ZOCT
41 02331 006102         JSR     @.CRLF
42 02332 006101         JSR     @.MESS          ;REPORT DATA ERROR
43 02333 003265         DATER
44 02334 024223         LDA     1,TSTN          ;GET THE TEST NUMBER
45 02335 006123         JSR     @.PRTD          ;PRINT IT
46 02336 006102         JSR     @.CRLF          ;GO TO NEXT LINE
47 02337 006101         JSR     @.MESS          ;PRINT HEADER
48 02340 003277         HEADR
49 02341 024173         LDA     1,LRS           ;GET LAST READ SWITCH
50 02342 125015         MOV#    1,1,SNR         ;IS IT SET
51 02343 014021         DSZ     AUTO1
52 02344 014021         DSZ     AUTO1
53 02345 026021         LDA     1,@AUTO1
54 02346 030233         LDA     2,SMSK
55 02347 125300         MOVS    1,1
56 02350 147400         AND     2,1
57 02351 044423         STA     1,ESEC
58 02352 024420 PERR1:  LDA     1,GDAT          ;GET GOOD DATA WORD
59 02353 006104         JSR     @.ZOCT          ;PRINT IT
60 02354 024417         LDA     1,BDAT          ;GET BAD DATA WORD
```

```
 0035 FPYRL
01 002355 006104          JSR     @.ZOCT          /PRINT IT
02 002356 024411          LDA     1,PNTR2         /GET THE ADDRESS IN CORE
03 002357 006104          JSR     @.ZOCT          /PRINT IT
04 002360 024210          LDA     1,TRK           /GET THE TRACK
05 002361 006104          JSR     @.ZOCT          /PRINTT IT
06 002362 024412          LDA     1,ESEC
07 002363 006104          JSR     @.ZOCT          /PRINT IT
08 002364 006102          JSR     @.CRLF          /GO TO NEXT LINE
09 002365 006677          JMP     DNXT            /CONTINUE TILL DONE ALL
10
11 002366 000000 PNTR1:   0
12 002367 000000 PNTR2:   0
13 002370 000000 WRDC:    0
14 002371 000000 DKTN:    0
15 002372 000000 GDAT:    0
16 002373 000000 BDAT:    0
17 002374 000000 ESEC:    0
18 002375 000000 DECT:    0
19 002376 000000 FERR:    0
```

```
 0036 FPYRL
01
02
03                /THIS ROUTINE GENERATES RANDOM NUMBERS.
04
05 002377 044434 RAND:    STA     1,AC.1
06 002400 050434          STA     2,AC.2
07 002401 054434          STA     3,AC.3
08 002402 020434          LDA     0,RAN
09 002403 024435          LDA     1,ITRER
10 002404 125004          MOV     1,1,SZR
11 002405 001400          JMP     0,3
12 002406 105000          MOV     0,1
13 002407 030000-         LDA     2,-55
14 002410 147400          AND     2,1
15 002411 125120          MOVZL   1,1
16 002412 030425          LDA     2,K20
17 002413 050417          STA     2,BITCT
18 002414 152400          SUB     2,2
19 002415 176520          SUBZL   3,3
20 002416 137400          AND     1,3
21 002417 173000          ADD     3,2
22 002420 125220          MOVZR   1,1
23 002421 014411          DSZ     BITCT
24 002422 000773          JMP     .-5
25 002423 151200          MOVR    2,2
26 002424 101200          MOVR    0,0
27 002425 040411          STA     0,RAN
28 002426 024405          LDA     1,AC.1
29 002427 030405          LDA     2,AC.2
30 002430 034405          LDA     3,AC.3
31 002431 001400          JMP     0,3
32
33 002432 000000 BITCT:   0
34 002433 000000 AC.1:    0
35 002434 000000 AC.2:    0
36 002435 000000 AC.3:    0
37 002436 035726 RAN:     35726
38 002437 000020 K20:     20
39 002440 000000 ITRER:   0
```

```
10037 FPYRL
01 02441 044450 INTR:   STA     0,SV0           ;SAVE AC0
02 02442 044450         STA     1,SV1           ;SAVE AC1
03 02443 050450         STA     2,SV2           ;SAVE AC2
04 02444 050450         STA     3,SV3           ;SAVE AC3
05 02445 151200         MOVR    2,2             ;GET THE CARRY
06 02446 050447         STA     2,SAVC          ;SAVE IT
07 02447 071477         INTA    2               ;GET INTERRUPTING DEVICE CODE
08 02450 063733         SKPDZ   FPY             ;FLOPPY DONE SET
09 02451 000426         JMP     DSKC            ;GO CHECK DEVICE CODE
10 02452 063710         SKPDZ   TTI             ;KEYBOARD DONE SET
11 02453 000410         JMP     KBINP           ;GO GET INPUT
12 02454 063711         SKPDZ   TTO             ;CONSOL OUTPUT DONE SET
13 02455 000404         JMP     CLRTT           ;GO CLEAR IT
14 02456 006101 UKNWN:  JSR     @,MESS          ;PRINT MESSAGE
15 02457 003320         UNKWN                   ;'UNKNOWN INTERRUPT'
16 02460 000407         JMP     DISMIS          ;DISMIS THE INTERRUPT
17 02461 060211 CLRTT:  NIOC    TTO             ;CLEAR TTO DONE
18 02462 000405         JMP     DISMIS          ;DISMIS THE INTERRUPT
19 02463 060610 KBINP:  DIAC    0,TTI           ;GET THE CHARACTER
20 02464 024232         LDA     1,TMSK          ;GET THE MASK
21 02465 123400         AND     1,0             ;STRIP THE PARITY BIT
22 02466 006106         JSR     @IINP?          ;GO TO INPUT ROUTINE
23 02467 020422 DISMIS: LDA     0,SV0           ;RESTORE AC'S AND CARRY
24 02470 024422         LDA     1,SV1
25 02471 030422         LDA     2,SV2
26 02472 034423         LDA     3,SAVC
27 02473 175100         MOVL    3,3
28 02474 034420         LDA     3,SV3
29 02475 060177         INTEN                   ;TURN ON INTERRUPTS
30 02476 002000         JMP     @0              ;EXIT
31 02477 020103 DSKC:   LDA     0,.DTAB         ;GET ADDRESS OF DEVICE TABLE
32 02500 040021         STA     0,AUTO1         ;SET UP POINTER
33 02501 014021         DSZ     AUTO1           ;INITIALIZE POINTER
34 02502 022021 NDEV:   LDA     0,@AUTO1        ;GET A DEVICE CODE
35 02503 101414         INC#    0,0,SZR         ;IS IT THE TERMINATOR
36 02504 000402         SKIP
37 02505 000751         JMP     UKNWN           ;GO REPORT UNKNOWN INTERRUPT
38 02506 112414         SUB#    0,2,SZR         ;IS IT A DISK DEVICE CODE
39 02507 000773         JMP     NDEV            ;NOT THIS ONE,GO CHECK NEXT
40 02510 000757         JMP     DISMIS          ;DISK DEVICE CODE.DISMIS IT
41
42
43 02511 000000 SV0:    0
44 02512 000000 SV1:    0
45 02513 000000 SV2:    0
46 02514 000000 SV3:    0
47 02515 000000 SAVC:   0
```

```
10038 FPYRL
01 02516 171400 DCHNG:  INC     3,2
02 02517 151400         INC     2,2
03 02520 050441         STA     2,DCH,5
04 02521 024437         LDA     1,DCH,1
05 02522 037376         LDA     3,0-2,2
06 02523 137400         AND     1,3
07 02524 057376         STA     3,0-2,2
08 02525 123400         AND     1,0
09 02526 044434         STA     0,DCH,6
10 02527 031377         LDA     2,-1,2
11 02530 136414         SUB#    1,3,SZR
12 02531 122415         SUB#    1,0,SNR
13 02532 000424         JMP     DCH,2
14 02533 021000 DCH,4:  LDA     0,0,2
15 02534 103112         ADDL#   0,0,SZC
16 02535 101103         MOVL    0,0,SNC
17 02536 103113         ADDL#   0,0,SNC
18 02537 000412         JMP     DCH,3
19 02540 101200         MOVR    0,0
20 02541 162400         SUB     3,0
21 02542 123414         AND#    1,0,SZR
22 02543 000406         JMP     DCH,3
23 02544 034416         LDA     3,DCH,6
24 02545 163000         ADD     3,0
25 02546 041006         STA     0,0,2
26 02547 034412         LDA     3,DCH,5
27 02550 037776         LDA     3,0-2,3
28 02551 151400 DCH,3:  INC     2,2
29 02552 022407         LDA     0,@DCH,5
30 02553 142414         SUB#    2,0,SZR
31 02554 000757         JMP     DCH,4
32 02555 034405         LDA     3,DCH,6
33 02556 030403 DCH,22: LDA     2,DCH,5
34 02557 001001         JMP     1,2
35
36 02560 000077 DCH,1:  77
37 02561 000000 DCH,5:  0
38 02562 000000 DCH,6:  0
39                      .EJECT
```

```
 0039 FPYRL
01
02 02563 000107 INCH:   JSR     @.ENTR          ;SET UP FOR ERRORS
03 02564 000101         JSR     @.MESS          ;PRINT MESSAGE
04 02565 002671         SWAP                    ;INTERCHANGE TEST
05 02566 000105         JSR     @.TIND          ;GET OPERATOR RESPONSE
06 02567 030225         LDA     2,W             ;GET "W"CODE
07 02570 112414         SUB#    0,2,SZR         ;IS IT A "W"
08 02571 000403         JMP     RCHK            ;NO CHECK FOR R
09 02572 152000         ADC     2,2             ;YES! GENERATE A MINUS 1
10 02573 000405         JMP     PATS            ;GO SELECT PATTERN
11 02574 030226 RCHK:   LDA     2,R             ;GET "R""CODE
12 02575 112414         SUB#    0,2,SZR         ;IS IT R
13 02576 000765         JMP     INCH            ;NO TRY AGAIN
14 02577 152440         SUBO    2,2
15 02600 050172 PATS:   STA     2,RWS           ;SET READ/WRITE
16 02601 000101         JSR     @.MESS
17 02602 003503         NPASS
18 02603 000137         JSR     @.TIND
19 02604 000775         JMP     .-3
20 02605 044443         STA     1,ILOOP
21 02606 000101         JSR     @.MESS
22 02607 002733         PTRN                    ;PATTERN DESIRED
23 02610 000137         JSR     @.TIND          ;GET OPERATOR INPUT
24 02611 000767         JMP     PATS            ;BAD INPUT
25 02612 152520         SUBZL   2,2             ;GENERATE A 1
26 02613 132032         ADCZ#   1,2,SZC         ;IS NUMBER >1
27 02614 000764         JMP     PATS
28 02615 151540         INCOL   2,2             ;COMPUTE A 6
29 02616 151120         MOVZL   2,2
30 02617 132433         SUBZ#   1,2,SNC         ;IS NUMBER < 6
31 02620 000760         JMP     PATS            ;NO GET NEW NUMBER
32 02621 030440         LDA     2,PTAB          ;GET PATTERN TABLE
33 02622 133000         ADD     1,2
34 02623 021000         LDA     0,0,2
35 02624 042136         STA     0,@.GEN         ;PUT IT IN GENERATOR
36 02625 000141         JSR     @.DGEN          ;GO GENERATE DATA
37 02626 024235 CGEN:   LDA     1,RPOP          ;GET READ PREAMBLE OPCODE
38 02627 020172         LDA     0,RWS           ;GET READ WRITE SWITCH
39 02630 101014         MOV#    0,0,SZR         ;IS IT READ
40 02631 127121         ADDZL   1,1,SKP         ;NO GEN WRITE OPCODE
41 02632 125120         MOVZL   1,1             ;YES GEN. READ OPCODE
42 02633 044420         STA     1,WRON          ;SET UP FUNCTION
43 02634 020421         LDA     0,ICRT          ;GET RE-ENTRY INSTR.
44 02635 026421         LDA     1,@.NEXC        ;GET OLD INSTR.
45 02636 044421         STA     1,ICTP          ;SAVE OLD INSTR.
46 02637 042417         STA     0,@.NEXC        ;SET UP RE-ENTRY
47 02640 000125         JSR     @.RCAL          ;RECAL THE DRIVE
48 02641 000110         JSR     @.TSTC          ;GO EXECUTE COMMAND
49 02642 020415         LDA     0,ICTP                  ;GET OLD INSTR
50 02643 042413         STA     0,@.NEXC        ;RESTORE IT
51 02644 014404         DSZ     ILOOP
52 02645 000761         JMP     CGEN
53 02646 002401         JMP     @.IOUT
54
55 02647 000537 .IOUT:  UREQ
56 02650 000000 ILOOP:  0
57
58
59 02651 054407 WRT:    STA     3,WRTR          ;SAVVE RETURN ADDRESS
60 02652 000120         JSR     @.DVR           ;CALL DRIVER
```

```
 0040 FPYRL
01 02653 000000 WRON:   0
02 02654 002404         JMP     @WRTR           ;EXIT
03
04 02655 000135 ICRT:   JSR     @.WRT
05 02656 002030 .NEXC:  NEXC
06 02657 000000 ICTP:   0
07 02660 000000 WRTR:   0
08
09
10 02661 002661 PTAB:   .
11 02662 102440         SUBO    0,0             ;ALL ZEROS
12 02663 102000         ADC     0,0             ;ALL    ONES
13 02664 020241         OGEN3                   ;125252
14 02665 020242         OGEN4                   ;52525
15 02666 020243         OGEN5                   ;66666
16 02667 020244         OGEN6                   ;70707
17 02670 000000         0
```

```
10041 FPYRL
01                                      ;MESSAGE BLOCK
02
03
04
05
06 02671 047111 SWAP:   .TXT    /INTERRCHANGE TEST<15><12>
07 02702 053412 W" SELECTS WRITE ONLY<15><12>
08 02716 051042 "R" SELECTS READ ONLY <15><12>/
09 02733 042523 PTRN:   .TXT    /SELECT PATTERN <15><12>
10 02743 030412 1= ALL ZEROS<15><12>
11 02752 031012 2= ALL ONES<15><12>
12 02761 036463 3= 125252<15><12>
13 02766 032012 4= 52525<15><12>
14 02773 032412 5= 66666<15><12>
15 03006 033012 6= 70707<15><12>/
16 03006 040504 DATA:   .TXT    /DATA?<15><12>/
17 03012 046106 NRDY:   .TXT    /FLOPPY NOT READY<15><12>/
18 03024 052123 STATE:  .TXT    /STATUS ERROR: TEST# /
19 03037 051127 WRFLT:  .TXT    /WRITE FAULT<15><12>/
20 03046 040504 DCHLT:  .TXT    /DATA CHANNEL LATE<15><12>/
21 03060 047516 NODAT:  .TXT    /NO DATA RECEIVED ON READ<15><12>/
22 03076 044103 CHKWD:  .TXT    /CHECK WORD ERROR<15><12>/
23 03110 042523 SECER:  .TXT    /SECTOR ADDRESS ERROR<15><12>/
24 03124 046111 ILCMD:  .TXT    /ILLEGAL COMMAND SENT<15><12>/
25 03140 030066 NAME:   .TXT    /6038-6039 FLOPPY DISK RELIABILITY: REV. 0<15><1
26 03160 044510 MEMRY:  .TXT    /HIGHEST MEMORY ADDRESS=/
27 03202 047105 DEVCD:  .TXT    /ENTER FLOPPY DEVICE CODES <15><12>/
28 03221 047111 SETSW:  .TXT    /INPUT SWITCH SETTINGS: HIT 0 FIRST<15><12>/
29 03244 047514 LOAD:   .TXT    /LOAD SCRATCH DISKETTES:HIT A CR<15><12>/
30 03265 040504 DATER:  .TXT    /DATA ERROR: TEST# /
31 03277 047507 HEADR:  .TXT    /GOOD   BAD     ADDRESS TRACK   SECTOR <15><12>/
32 03320 047125 UNKWN:  .TXT    /UNKNOWN INTERRUPT <15><12>/
33 03333 044504 DNUM:   .TXT    /DISK# /
34 03337 052502 BUFA:   .TXT    /BUFFERS START AT LOC. /
35 03353 040520 PASSN:  .TXT    /PASS# /
36 03357 042524 TSTU:   .TXT    /TEST UNITS: CR=BOTH,1=UNIT0,2=UNIT1<12><15>/
37 03402 044524 TOUT:   .TXT    /TIMEOUT ERROR:NO DONE <12><15>/
38 03417 020040 TRKN:   .TXT    /   TRK:/
39 03423 042523 SECN:   .TXT    /SEC:/
40 03426 042101 ADDNF:  .TXT    /ADDRESS NOT FOUND<15><12>/
41 03440 042524 TNUM:   .TXT    /TEST# /
42 03444 047516 NOFPY:  .TXT    /NO FLOPPY IN SYSTEM<15><12>/
43 03457 047516 NOHLD:  .TXT    /NO HEAD LOADED<15><12>/
44 03470 051124 TRKER:  .TXT    /TRACK  ADDRESS ERROR<15><12>/
45 03503 020043 NPASS:  .TXT    /# OF PASSES /
46 03512 047531 DUMB:   .TXT    /YOU BLEW IT! LOAD SCRATCH DISKETTE UNIT 0:
47 03537 044510 HIT A KEY<15><12>/
48 03545 051127 WRTER:  .TXT    /WRITE ERROR<15><12>/
49       000000                 .NOLOC  0
50
```

```
10042 FPYRL
01
02
03
04
05
06              ;FILENAME=TTYIO
07
08              ;TELETYPE NON INTERRUPT PACKAGE
09              ;CARRY,AC0,AC1,AC2 SAVED
10
11              ;"MES?S" PRINTS ASCII MESSAGES AS SPECIFIED BY ASSEMBLER
12
13              ;"CRL?F" PRINTS A CARRIAGE RETURN
14
15              ;"POC?" PRINTS C(1) IN OCTAL
16              ;"ZOC?" PRINTS C(1) IN OCTAL, LEADING ZEROS SUPPRESSED
17              ;"PDE?C" PRINTS C(1) IN DECIMAL, LEADING ZEROS SUPPRESSED,
18              ;THE ABOVE THREE ARE FOLLOWED BY A TAB
19              ;"PDC?S" PRINTS C(1) IN DECIMAL, LEADING ZEROS SUPPRESSED,
20              ;FOLLOWED BY THE CHARACTER STORED AT CALLING LOCATION +1.
21              ;PROGRAM RETURNS TO CALLING LOCATION +2.
22
23              ;"TIN?O" ACCEPTS OCTAL, AND
24              ;"TIN?D" ACCEPTS DECIMAL SINGLE PRECISION SIGNED INTEGERS
25              ;INTO AC1 FROM THE TTI.  LEADING NULLS, TABS,
26              ;AND SPACES ARE IGNORED.  A 16 BIT UNSIGNED INTEGER IS
27              ;FORMED, THEN NEGATED IF A MINUS SIGN IS TYPED.
28              ;EXIT AT CALL+1 IF INPUT ERROR WITH AC0=BAD CHARACTER.
29              ; (NOT A LEGAL DIGIT OR TERMINATING CHARACTER)
30              ;EXIT AT CALL+2 UPON TERMINATING CHARACTER
31              ; WITH AC0=0, 0, 40, 12, 15, 55
32              ; FOR NULL, SPACE, LINE-FEED, CARRIAGE RETURN, COMMA
33              ;THE ABOVE WAIT FOR TTO DONE, THEN TTO IS CLEARED.
34              ;RUBOUT WILL DELETE THE LAST DIGIT TYPED IN 'TIN?O' AND
35              ;'TIN?D'
36
37              ;"CHC?T" PRINTS ASCII CHARACTER IN C(0)R; C(0)L MUST BE 0.
38              ;EXITS CALL +2 IF C(0)R=0; SIMULATES TAB
39
40              ;"TYP?E" PRINTS C(0)R TO THE TTO OR LPT OR BOTH AS PER THE
41              ;SWITCH SELECTION REGISTER 'SWREG'.
42              ;EXITS AT CALL+1.  REPLACE "TYP?E" WITH
43              ;INTERRUPT 'TYP?E' IF DESIRED.
44
45              ;"TPS?P" PRINTS A SPACE AND EXITS AT CALL+1 WITH AC0 = 40
46
47
```

```
10043 FPYRL
01                ;
02                ;MES?S ROUTINE
03                ;
04                ;THE CALLING SEQUENCE IS:
05                ;
06                ;       JSR     @!MES?S
07                ;       POINTER TO MESSAGE TO BE PRINTED
08                ;
09
10 03554 054551 MES?S:  STA     3,RTN?A         ;SAVE THE RTN ADDRESS
11 03555 004502         JSR     SAV?E           ;SAVE THE STATE OF MACHINE
12 03556 034547         LDA     3,RTN?A         ;
13 03557 010546         ISZ     RTN?A
14 03560 031400         LDA     2,0,3           ;C(2) POINTS TO MESSAGE
15 03561 024417         LDA     1,P377?         ;A 8 BIT MASK
16 03562 021000 MES?M:  LDA     0,0,2           ;C(2)=DATA WORD
17 03563 125112         MOVL#   1,1,SZC
18 03564 123701         ANDS    1,0,SKP
19 03565 123401         AND     1,0,SKP         ;C(0)=DATA CHARACTOR RIGHT
20 03566 151400         INC     2,2             ;INC TO NEXT WORD
21 03567 124000         COM     1,1             ;FLIP MASK
22 03570 004414         JSR     CHC?T           ;PRINT
23 03571 000771         JMP     MES?M           ;ANOTHER
24 03572 004402         JMP     .+2
25 03573 004411 PLS?T:  JSR     CHC?T
26 03574 004551 PEX?T:  JSR     RST?R           ;RESTORE THE STATE OF MACHINE
27 03575 002530         JMP     @RTN?A          ;EXIT
28
```

```
10044 FPYRL
01                ;
02                ;CHC?T ROUTINE
03                ;
04                ;THE CALLING SEQUENCE IS:
05                ;
06                ;       LDA     0,CHARACTER TO BE PRINTED (RIGHT BYTE)
07                ;       JSR     @!CHC?T
08                ;
09
10 03576 000000 PSP?:   0
11 03577 000P00 SPT?G:  0
12 03600 000377 P377?:  377
13 03601 000P11 PC1?1:  11
14 03602 000000 CHR?E:  0
15 03603 000000 CAC?0:  0                       ;AC0 SAVE IN CHC?T ROUTINE
16
17 03604 040777 CHC?T:  STA     0,CAC?0         ;SAVE AC0
18 03605 101315         MOVS#   0,0,SNR         ;RETURN +2 IF NULL
19 03606 001401         JMP     1,3
20 03607 175100         MOVL    3,3             ;FOR CARRY SAVE
21 03610 054772         STA     3,CHR?E         ;PRINT C(0) RIGHT
22 03611 034770         LDA     3,PC1?1         ;AC3 = 11
23 03612 116415         SUB#    0,3,SNR         ;SKIP IF A TAB IS NOT TO
                                                ;BE SIMULATED
24
25 03613 000403         JMP     CHA?3
26 03614 004556         JSR     TYP?E           ;PRINT IT
27 03615 000407         JMP     CHE?X           ;EXIT
28 03616 004551 CHA?3:  JSR     TPS?P           ;PRINT A SPACE
29 03617 020513         LDA     0,CHR?Z
30 03620 034410         LDA     3,PC7?          ;AC3 = 7
31 03621 163404         AND     3,0,SZR
32 03622 000774         JMP     CHA?3           ;SIMULATE A TAB WITH 1
                                                ;TO 7 SPACES
33
34 03623 040507         STA     0,CHR?Z
35 03624 020757 CHE?X:  LDA     0,CAC?0         ;RESTORE AC0
36 03625 034755         LDA     3,CHR?E         ;RESTORE CRY
37 03626 175200         MOVR    3,3             ;
38 03627 001400         JMP     0,3             ;EXIT
39
40 03630 000007 PC7?:   7
41
42
```

```
.0045 FPYRL
                        ;
                        ;CRL?F ROUTINE
                        ;
                        ;THE CALLING SEQUENCE IS:
                        ;
                        ;       JSR     @ICRL?F
                        ;
                        ;
03631 054474 CRL?F:     STA     3,RTN?A         ;SAVE RETURN
03632 004505            JSR     SAV?E           ;SAVE THE WORLD
03633 020405            LDA     0,K15?
03634 004536            JSR     TYP?E           ;PRINT CARRIAGE AND LF
03635 020402            LDA     0,K12?
03636 000735            JMP     PLS?T           ;GO TO RESTORE THE WORLD
03637 000012 K12?:      12
03640 000015 K15?:      15
```

```
.0046 FPYRL
                        ;
                        ;ZOC?T, POC?T, POC?S AND PDE?C ROUTINES.
                        ;
                        ;THE CALLING SEQUENCE IS:
                        ;
                        ;       LDA     1,OCTAL NUMBER TO BE PRINTED
                        ;               (LEADING ZEROES SUPPRESSED)
                        ;       JSR     @1ZOC?T
                        ;
                        ;THE CALLING SEQUENCE IS:
                        ;
                        ;       LDA     1,OCTAL NUMBER TO BE PRINTED
                        ;               (LEADING ZEROES NOT SUPPRESSED)
                        ;       JSR     @1POC?T
                        ;
                        ;THE CALLING SEQUENCE IS:
                        ;
                        ;       LDA     1,DECIMAL NUMBER TO BE PRINTED
                        ;               (LEADING ZEROES SUPPRESSED)
                        ;       JSR     @1PDE?C
                        ;
                        ;THE CALLING SEQUENCE IS:
                        ;
                        ;       LDA     1,DECIMAL NUMBER TO BE PRINTED
                        ;               (LEADING ZEROES SUPPRESSED)
                        ;       JSR     @1POC?S
                        ;       ALPHA   WHERE ALPHA IS THE CHARACTER PRINTED
                        ;               AFTER THE DECIMAL NUMBER


03641 054464 ZOC?T:     STA     3,RTN?A         ;SAVE THE RTN ADDRESS
03642 004475            JSR     SAV?E           ;SAVE THE WORLD
03643 102400            SUB     0,0             ;
03644 000404            JMP     ZPO?T           ;
03645 054460 POC?T:     STA     3,RTN?A         ;SAVE THE RTN ADDRESS
03646 004471            JSR     SAV?E           ;SAVE THE WORLD
03647 020464            LDA     0,PC670         ;
03650 152620 ZPO?T:     SUBZR   2,2             ;PRINT C(1) IN OCTAL
03651 034463            LDA     3,PC170         ;C(2)=100000, C(3)=10
03652 000416            JMP     PDC?1           ;
03653 175400 POC?S:     INC     3,3             ;UPDATE THE RTN ADDR PNTR
03654 054451            STA     3,RTN?A         ;
03655 004462            JSR     SAV?E           ;SAVE THE WORLD
03656 034447            LDA     3,RTN?A         ;
03657 021777            LDA     0,-1,3          ;READ THE CHARACTER TO BE
                                                ;PRINTED AFTER THE DECIMAL
                                                ;NUMBER
03660 040716            STA     0,PSP?          ;SAVE THE SPECIAL CHAR.
03661 102000            ADC     0,0             ;AC0= -1
03662 000404            JMP     PDC?2           ;
03663 054442 PDE?C:     STA     3,RTN?A         ;SAVE THE RTN ADDRESS
03664 004453            JSR     SAV?E           ;SAVE THE WORLD
03665 102400            SUB     0,0             ;
03666 034751 PDC?2:     LDA     3,K12?          ;C(3)=12
03667 030446            LDA     2,DEF?B         ;PRINT C(1) IN DECIMAL
03670 040747 PDC?1:     STA     0,SPT?G         ;ACTIVATE/DEACTIVATE THE TAG FOR
                                                ;SPECIAL CHARACTER
                                                ;BOTH ENTRIES PRINT NUMBER
03671 101415            INC#    0,0,SNR         ;SKIP IF AC0 IS NOT -1
03672 101400            INC     0,0
```

```
 0047 FPYRL
01 03673 044543          STA     0,ZSU?P         ;THEN TAB TO NEXT POSITION
02 03674 054442          STA     3,TMP?          ;SAVE AC3
03 03675 034541 DCO?T:   LDA     3,ZSU?P         ;ZEROS SUPPRESS STUF
04 03676 102001 DEC?T:   AUC     0,0,SKP         ;SKIP FIRST TIME HERE PER DIGIT
05 03677 146400          SUB     2,1             ;DIVIDE C(AC1)  BY C(AC2)
06 03700 101405          INC     0,0,SNR         ;
07 03701 151235          MOVZR#  2,2,SNR         ;FOR ZERO SUPPRESS
08 03702 034431          LDA     3,PC6?0         ;
09 03703 146453          SUB0#   2,1,SNC         ;SUBTRACT MORE?
10 03704 000773          JMP     .-5             ;YES,GO BACK
11 03705 054531          STA     3,ZSU?P         ;NO,SAVE ZERO SUPPRESS FLAG
12                                               ;C(0)=DIGIT
13 03706 163004          ADD     3,0,SZR         ;MAKE ASCII
14 03707 004675          JSR     CHC?T           ;PRINT
15 03710 034426          LDA     3,TMP?          ;RESTORE AC3
16 03711 102400          SUB     0,0             ;
17 03712 172423          SUBZ    3,2,SNC         ;DIVIDE C(AC2) BY C(AC3)
18
19 03713 000403          JMP     .+3             ;SKIP IF AC3 > AC2
20 03714 101400          INC     0,0             ;AC3 < AC2
21 03715 000775          JMP     .-3             ;SUBTRACT MORE
22 03716 111004          MOV     0,2,SZR         ;WAS IT LAST DIGIT?
23 03717 000756          JMP     DCO?T           ;NO,GET NEXT DIGIT
24 03720 034657          LDA     3,SPT?G         ;YES,CHECK THE SPECIAL CHAR FLAG
25 03721 020660          LDA     0,PC1?1         ;FOLLOW THE PRINTOUT WITH
26 03722 175405          INC     3,3,SNR         ;TAB IF NOT SPCL CHAR FLAG
27 03723 020653          LDA     0,PSP?          ;OTHERWISE FOLLOW WITH THE CHAR
28 03724 000647          JMP     PLS?T           ;TO EXIT
29
30 03725 000000 RTN?A:   0
31 03726 000000 PCR?Y:   0                       ;CRY SAVE LOCATION
32 03727 000000 PAC?0:   0                       ;AC0 SAVE LOCATION
33 03730 000000 PAC?1:   0                       ;AC1 SAVE LOCATION
34 03731 000000 PAC?2:   0                       ;AC2 SAVE LOCATION
35 03732 000000 CHR?Z:   0
36 03733 000060 PC6?0:   60
37 03734 000010 PC1?0:   10
38 03735 023420 DET?B:   10000.
39 03736 000000 TMP?:    0
```

```
10048 FPYRL
01
02                       ;
03                       ; SAV?E , SAVE THE WORLD ROUTINE
04                       ;
05                       ; THIS ROUTINE SAVES AC0,AC1,AC2 AND CRY
06                       ;
07
08 03737 040770 SAV?E:   STA     0,PAC?0         ;
09 03740 044770          STA     1,PAC?1         ;
10 03741 050770          STA     2,PAC?2         ;
11 03742 101100          MOVL    0,0             ;
12 03743 040763          STA     0,PCR?Y         ;
13 03744 001400          JMP     0,3             ;
14
15
16                       ;
17                       ; RST?R , RESTORE THE WORLD ROUTINE
18                       ;
19                       ; THIS ROUTINE RESTORES THE AC0,AC1,AC2 AND CRY
20                       ;
21
22 03745 020761 RST?R:   LDA     0,PCR?Y         ;
23 03746 101200          MOVR    0,0             ;
24 03747 020760          LDA     0,PAC?0         ;
25 03750 024760          LDA     1,PAC?1         ;
26 03751 030760          LDA     2,PAC?2         ;
27 03752 001400          JMP     0,3             ;
28
```

```
0049 FPYRL
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 03753 135005 RUB?:   MOV     1,3,SNR         ;CAN'T RUB-OUT IF AC1 = 0
17 03754 000467         JMP     TIN?R           ;RETURN WITH ILLEGAL CHARACTER
18 03755 126400         SUB     1,1
19 03756 156422         SUBZ    2,3,SZC         ;SKIP IF AC3 IS LESS THAN AC2
20 03757 125401         INC     1,1,SKP
21 03760 157001         ADD     2,3,SKP
22 03761 000775         JMP     .-3
23 03762 054501         STA     3,FST?D         ;"FST?D" IS NON -1
24 03763 020750         LDA     0,PC6?0         ;AC0 = 60
25 03764 163000         ADD     3,0
26 03765 004405         JSR     TYP?E           ;ECHO AND DELETE THE DIGIT
27 03766 000524         JMP     TIN?W
28
29 03767 040445 TPS?P:  STA     0,TAC?0         ;SAVE AC0
30 03770 020447         LDA     0,PC4?0
31 03771 101001         MOV     0,0,SKP         ;SKIP OVER AC0 SAVE
32 03772 040442 TYP?E:  STA     0,TAC?0         ;SAVE AC0
33 03773 175100         MOVL    3,3             ;SAVE CRY AND RTN ADDR
34 03774 054441         STA     3,TYP?R         ;TYPE THE RIGHT BYTE OF AC0
35 03775 034567         LDA     3,P177?         ;STRIP THE PARITY BIT
36 03776 163400         AND     3,0             ;
37 03777 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 04000 175404         INC     3,3,SZR         ;SKIP IF INT? IS -1
41 04001 034116         LDA     3,SWREG         ;READ THE SWITCHES
42 04002 177100         ADDL    3,3             ;SHIFT AC3 BY 2 PLACES
43 04003 175112         MOVL#   3,3,SZC         ;SKIP IF TYPEOUTS ARE NOT
44                                              ;SUPPRESSED
45 04004 000405         JMP     PLP?T
46 04005 061111 TTY?:   DOAS    0,TTO
47 04006 063511         SKPBZ   TTO
48 04007 000777         JMP     .-1
49 04010 060211         NIOC    TTO             ;CLEAR TTO DONE FLAG
50 04011 177100 PLP?T:  ADDL    3,3             ;SHIFT AC3 BY 2 PLACES
51 04012 177103         ADDL    3,3,SNC         ;SKIP IF THE OUTPUT IS
52                                              ;REQUIRED ON THE LPT
53 04013 000405         JMP     TPR?T
54 04014 061117         DOAS    0,LPT           ;OUTPUT THE CHARACTER TO LPT
55 04015 063517         SKPBZ   LPT             ;WAIT FOR LPT
56 04016 000777         JMP     .-1
57 04017 060217         NIOC    LPT             ;CLEAR THE DONE FLAG FOR LPT
58 04020 034544 TPR?T:  LDA     3,P177?         ;
59 04021 116043         ADCZ    0,3,SNC         ;SKIP IF IT WAS RUBOUT
60 04022 034415         LDA     3,PC4?0         ;AC3 = 40
```

```
0050 FPYRL
01 04023 162432         SUBZ#   3,0,SZC         ;SKIP FOR NON PRINTING CHR.
02 04024 010706         ISZ     CHR?Z
03 04025 034541         LDA     3,PC17?         ;AC3 = 15
04 04026 116445         SUBU    0,3,SNR         ;SKIP IF IT WAS NOT A "CR"
05 04027 054703         STA     3,CHR?Z         ;CLEAR THE HORZ POS
06 04030 020404         LDA     0,TAC?0         ;RESTORE AC0
07 04031 034404         LDA     3,TYP?R         ;RESTORE CRY AND RTN ADDR
08 04032 175200         MOVR    3,3             ;
09 04033 001400         JMP     0,3             ;RETURN
10 04034 000000 TAC?0:  0
11 04035 000000 TYP?R:  0
12
13 04036 000000 ZSU?P:  0
14 04037 000040 PC4?0:  40
15
16
```

```
10051 FPYRL
01                ;
02                ;TIN?O AND TIN?D ROUTINES.
03                ;
04                ;THE CALLING SEQUENCE IS:
05                ;
06                ;       JSR     @ITIN?O ;ACCEPT IN OCTAL
07                ;       ERROR RETURN    WITH BAD CHARACTER IN AC0
08                ;       NORMAL RETURN   WITH TERMINATING
09                ;                       CHARACTER IN AC0
10                ;                       0,40,12,15,55 FOR
11                ;                       NULL,SPACE,L/F,C/R AND
12                ;                       COMMA RESPECTIVELY.
13                ;
14                ;       NOTE:
15                ;            THE NUMBER IS ACCEPTED IN AC1 FROM TTI BY TIN?O AND
16                ;            BAD CHARACTER IS ANY CHARACTER THAT IS NOT A LEGAL
17                ;            DIGIT OR A TERMINATING CHARACTER
18                ;
19                ;THE CALLING SEQUENCE IS:
20                ;
21                ;       JSR     @ITIN?D ;ACCEPT IN DECIMAL
22                ;       ERROR RETURN    ;SAME FORMAT AS TIN?O
23                ;       NORMAL RETURN
24                ;
25
26
27 04040 020525 TIN?C:  LDA     0,PC1?2
28 04041 004731         JSR     TYP?E
29 04042 010663 TIN?X:  ISZ     RTN?A
30 04043 040420 TIN?R:  STA     0,FST?D         ;"FST?D" IS NON -1
31 04044 152000         ADC     2,2             ;AC2 = -1
32 04045 020771 TSI?:   LDA     0,ZSU?P
33 04046 175620         INCZR   3,3             ;AC3 IS 1 IF THE CHARACTER
34                                              ;TYPED WAS A + AND A 100000
35                                              ;IF IT WAS A -.
36 04047 054767         STA     3,ZSU?P
37 04050 101112         MOVL#   0,0,SZC         ;SKIP IF THE PREVIOUS SIGN
38                                              ;WAS A PLUS
39 04051 124400         NEG     1,1             ;TAKE TWO'S COMPLEMENT IF
40                                              ;THE PREVIOUS SIGN WAS "-"
41 04052 034656         LDA     3,PAC?1
42 04053 167000         ADD     3,1
43 04054 044654         STA     1,PAC?1         ;PAC?1 HAS THE INTERMEDIATE
44                                              ;RESULT
45 04055 126400         SUB     1,1
46 04056 151113         MOVL#   2,2,SNC         ;SKIP IF EXIT IS REQUIRED
47 04057 000433         JMP     TIN?#
48 04060 004665         JSR     RST?R           ;RESTORE THE WORLD
49 04061 022402         LDA     0,FST?D         ;RESTORE AC0
50 04062 002643         JMP     @RTN?A          ;RETURN
51
52 04063 000000 FST?D:  0
53
54 04064 054641 TOD?T:  STA     3,RTN?A         ;SAVE THE RTN ADDR
55 04065 004652         JSR     SAV?E           ;SAVE THE WORLD
56 04066 102000         ADC     0,0             ;AC1 = -1 (ENTRY FOR ODT)
57 04067 040774         STA     0,FST?D         ;LOOK FOR FIRST DIGIT
58 04070 101120         MOVZL   0,0             ;AC1 = -2
59 04071 000411         JMP     TIN?Z
60 04072 054633 TIN?O:  STA     3,RTN?A         ;OCTAL ENTRY,SAVE RTN ADDR
```

```
 0052 FPYRL
01 04073 004644         JSR     SAV?E           ;SAVE THE WORLD
02 04074 102120         ADCZL   0,0             ;OCTAL ENTRY SWITCH
03 04075 000404         JMP     TIN?O           ;
04 04076 054627 TIN?D:  STA     3,RTN?A         ;DECIMAL ENTRY,RTN SAVED
05 04077 004640         JSR     SAV?E           ;SAVE THE WORLD
06 04100 102440         SUBO    0,0             ;DECIMAL ENTRY SWITCH
07 04101 126400 TIN?O:  SUB     1,1
08 04102 030463 TIN?Z:  LDA     2,PC1?2
09 04103 113000         ADD     0,2             ;AC2 IS 10 FOR OCTAL AND 12
10                                              ;FOR DECIMAL NUMBERS
11 04104 102440         SUBO    0,0
12 04105 040731         STA     0,ZSU?P         ;SIGN AND LEADING SPACES FLAG
13 04106 034730 TIN?S:  LDA     3,ZSU?P
14 04107 175014         MOV#    3,3,SZR         ;SKIP FOR LEADING SPACES
15 04110 000732         JMP     TIN?X
16 04111 054617         STA     3,PAC?1
17 04112 063610 TIN?W:  SKPDN   TTI
18 04113 000777         JMP     .-1
19 04114 060610         DIAC    0,TTI
20 04115 004655         JSR     TYP?E
21 04116 034446         LDA     3,P1777
22 04117 163400         AND     3,0             ;STRIP THE PARITY BIT
23 04120 116415         SUB#    0,3,SNR         ;SKIP IF NOT A RUB-OUT
24 04121 000632         JMP     RUB?
25 04122 034715         LDA     3,PC47@
26 04123 116414         SUB#    0,3,SZR
27 04124 101015         MOV#    0,0,SNR
28 04125 000761         JMP     TIN?S           ;SPACE, OR NULL
29 04126 034442         LDA     3,TIN?2
30 04127 116405         SUB     0,3,SNR         ;COMMA
31 04130 000712         JMP     TIN?X
32 04131 175414         INC#    3,3,SZR         ;MINUS
33 04132 175235         MOVZR#  3,3,SNR         ;OR PLUS ?
34 04133 000712         JMP     TSI?            ;MODIFY THE SIGN
35 04134 034432 TIN?M:  LDA     3,PC175         ;AC3 = 15
36 04135 116415         SUB#    0,3,SNR         ;IS IT A CARRIAGE RETURN?
37 04136 000702         JMP     TIN?C           ;IF CR THEN GO TO TIN?C
38 04137 034426         LDA     3,PC1?2         ;AC3 = 12
39 04140 116404         SUB     0,3,SZR         ;SKIP FOR LINE FEED
40 04141 000403         JMP     TIN?N
41 04142 020424         LDA     0,PC175         ;AC0 = 15
42 04143 000676         JMP     TIN?C+1
43 04144 034423 TIN?N:  LDA     3,TIN?1
44 04145 117022         ADDZ    0,3,SZC         ;SKIP IF NOT A DIGIT
45 04146 156513         SUBL#   2,3,SNC         ;SKIP IF DIGIT
46 04147 000674         JMP     TIN?R
47 04150 010666         ISZ     ZSU?P           ;OUT OF LEADING SPACES
48 04151 102400         SUB     0,0             ;AC0 = 0
49 04152 010711         ISZ     FST?D           ;SKIP IF IT WAS FIRST DIGIT
50                                              ;FOR ODT
51 04153 121120         MOVZL   1,0
52 04154 105120         MOVZL   0,1
53 04155 125120         MOVZL   1,1             ;AC1 IS SHIFTED BY 3 PLACES
54 04156 167000         ADD     3,1             ;& OLD PAC?1'S + NEW DIGIT
55 04157 155220         MOVZR   2,3
56 04160 175232         MOVZR#  3,3,SZC         ;SKIP IF OCTAL MODE
57 04161 107000         ADD     0,1             ;ADD 2 OLD PAC?1'S
58 04162 000730         JMP     TIN?W
59
60 04163 000000 INT?:   0                       ;TYPE OUTSS CAN BE FORCED TO
```

```
 0053 FPYRL
01
02 04164 000177 P1727:  177
03 04165 000012 PC172:  12
04 04166 000015 PC175:  15
05 04167 177720 TIN21:  -60
06 04170 000054 TIN22:  54
07 04171 000100 C1020:  100
08
09
```

;TTY BY PLACING -1 IN THIS LOC.

```
 0054 FPYRL
01
02              ;FILENAME = SWPAK
03
04
05              ;1.     SWITCH SETTINGS
06              ;
07              ;       LOCATION "SWREG" IS USED TO SELECT THE PROGRAM OPTIONS
08              ;       (NOT SYSTEM CONFIGURATION). WHILE RUNNING UNDER DTOS,
09              ;       THIS LOCATION WILL BE LOADED BY THE MONITOR.
10              ;       HOWEVER UNDER STAND ALONE AND PROGRAM LOAD MODES THIS
11              ;       LOCATION WILL BE SET ACCORDING TO THE ANSWERS SUPPLIED
12              ;       BY THE OPERATOR. IN ANY CASE THE OPTIONS CAN BE CHANGED
13              ;       OR VERIFIED BY USING ONE OF THE COMMANDS GIVEN IN SEC.
14              ;       1.2
15              ;
16              ;
17              ;1.1    SWITCH OPTIONS
18              ;       DIFFERENT BITS AND THEIR INTERPRETATION AT LOCATION
19              ;       "SWREG" IS AS FOLLOWS:
20              ;
21              ;       BIT     OCTAL   BINARY  INERPRETATION
22              ;               VALUE   VALUE
23              ;
24              ;       1               0       LOOP ON ERROR
25              ;               40000   1       SKIP LOOPING ON ERROR
26              ;
27              ;       2               0       PRINT TO CONSOLE
28              ;               20000   1       ABORT PRINT OUT TO CONSOLE
29              ;
30              ;       3               0       PRINT DETAILED ERROR ON THE
31              ;                               SELECTED DEVICE/DEVICES
32              ;               10000   1       ONLY % FAILURE REQUIRED
33              ;
34              ;       4               0       ALLOW END OF PASS PRINT OUT
35              ;               04000   1       SUPPRESS END OF PASS PRINT OUT
36              ;
37              ;       5               0       DO NOT PRINT ON THE LINE PRINTER
38              ;               02000   1       PRINT ON THE LINE PRINTER
39              ;
40              ;       6               0       DO NOT HALT ON ERROR
41              ;               01000   1       HALT ON ERROR
42              ;
43              ;       7               0       DO NOT PRINT PASSING OF
44              ;                               EACH TEST ON PRINTING DEVICE
45              ;               00400   1       PRINT PASSING OF EACH TEST
46              ;
47              ;1.2    SWITCH COMMANDS
48              ;       ONCE THE PROGRAM STARTS EXECUTING THE STATE OF ANY OF
49              ;       THE BITS CAN BE CHANGED BY HITTING KEYS 1 THROUGH 6. THE
50              ;       PROGRAM WILL CONTINUE RUNNING AFTER UPDATING THE OPTIONS
51              ;       EACH KEY WILL COMPLEMENT THE STATE OF THE BIT AFFILIAT-
52              ;       ED WITH IT, THUS BIT 4 CAN BE ALTERED BY HITTING KEY 4.
53              ;       SETTIN OF ANY BIT OF LOCATION "SWREG" WILL SET BIT 0.
54              ;       (DEFAULT MODE IS DEFINED AS ALL BITS OF SWREG SET TO 0)
55              ;       THE PROGRAM CAN BE LOCKED INTO SWITCH MODIFICATION MODE
56              ;       BY TYPING A 0, IN WHICH CASE MORE THAN ONE BITS CAN BE
57              ;       CHANGED BEFORE THE CONTROL IS ALLOWED TO RETURN TO THE
58              ;       MAIN PROGRAM.
59              ;
60              ;1.2.1  OTHER COMMANDS
```

```
 0055 FPYRL
01                ;
02                ;       "CR"    A "RETURN" CAN BE TYPED TO CONTINUE THE PROGRAM
03                ;               AFTER ITS LOCKED IN A SWITCH MODIFICATION MODE
04                ;
05                ;       ^U      THIS COMMAND GIVEN AT ANY TIME WILL RESET "SWREG
06                ;               TO DEFAULT MODE AND RESTART THE PROGRAM.
07                ;
08                ;       ^R      THIS COMMAND GIVEN AT ANY TIME WILL RESTART THE
09                ;               PROGRAM, SWITCHES ARE LEFT WITH THE VALUES THEY
10                ;               HAD BEFORE THE COMMAND WAS ISSUED.
11                ;
12                ;       ^O      THIS COMMAND GIVEN AT ANY TIME WILL CAUSE THE
13                ;               PROGRAM CONTROL TO GO TO ODT (SEE SEC. 6)
14                ;
15                ;       ^S      PRINT THE SUMMARY OF ERRORS ACCUMULATED SO FAR.
16                ;
17                ;       M       THIS COMMAND GIVEN AT ANY TIME WILL PRINT THE
18                ;               CURRENT OPERATING MODES.
19                ;
20                ;
```

```
 0056 FPYRL
31
32
33
34                ;THIS PACKAGE IS USED TO CHANGE THE SETTINGS OF LOCATION
35                ;"SWREG" OF PAGE 0. THE PROGRAM CONTROL SHOULD ENTER "INP?I OR
36                ;INP?J" WITH AC3 HAVING THE RTN ADDR. IF THE TTI INPUT IS
37                ;ON INTERRUPT BASIS, ENTER AT "INP?I" OTHERWISE ENTER AT "INP?J"
38                ;THE INPUT IS ECHOED AFTER A "CR". IF THE COMMAND IS
39                ;NOT A LEGAL ONE THEN THE CONTROL IS RETURNED WITHOUT DOING
40                ;ANY THING, OTHERWISE ONE OF THE FOLLOWING COMMANDS IS
41                ;EXECUTED:
42                ;KEYS 1-9 AND A-F ARE USED TO COMPLEMENT THE CURRENT VALUE
43                ;OF BITS 1-15 OF "SWREG". IF ONE OF THESE KEYS IS HIT THE
44                ;CORRESPONDING BIT OF "SWREG" IS COMPLEMENTED AND THE CONTROL
45                ;IS RETURNED TO THE STATE PROGRAM HAD BEFORE HITTING THE KEY
46                ;TYPING OF A "0" WILL LOCK THE PROGRAM IN A SWITCH MODIFICATION
47                ;MODE IN WHICH CASE MORE THAN ONE BITS CAN BE CHANGED BEFORE
48                ;THE CONTROL IS ALLOWED TO RETURN TO THE MAIN PROGRAM, HITTING
49                ;THE "CR" KEY WILL UNLOCK THE PROGRAM FROM THIS MODE.
50                ;"^U" THIS COMMAND GIVEN AT ANY TIME WILL RESET THE "SWREG"
51                ;TO DEFAULT MODE (ALL ZEROS) AND RESTART THE PROGRAM AT ADD.
52                ;STORED IN LOCATION "INS?"
53                ;"^R" THIS COMMANND GIVEN AT ANY TIME  WILL RESTART THE PROG.
54                ;AT ADDRESS STORED IN LOCATION "INS?"
55                ;"M" THIS COMMAND GIVEN AT ANY TIME WILL PRINT THE  CURRENT
56                ;OPERATING MODES.
57                ;BEFORE THE CONTROL IS RETURNED TO THE MAIN PROGRAM BIT 0 WILL
58                ;BE SET IF ANY OF THE OTHER BITS OF "SWREG" IS SET, OTHERWISE
59                ;IT WILL BE CLEARED
60                ;"OUT" ROUTINE USES ONE LOCATION IN PAGE 0 FOR BREAK POINT.
61                ;ADDRESS OF THIS LOCATION SHOULD BE PLACED AT "ODO?F"
62                ;
63                ;THE CALLING SEQUENCE IS:
64                ;
65                ;       JSR     @IINP? ,WHERE IINP? POINTS TO
66                ;               INP?I OR INP?J
67                ;               ENTER AT INP?I FOR INTERRUPT
68                ;               DRIVEN PROGRAMS ,OTHERWISE AT INP?J
69                ;
70
71 04172 000033 IN373: 33
72 04173 000136 IN1736: 136
73 04174 000104 IN1704: 104
74
75 04175 054574 INP?I:  STA     3,INP?R
76 04176 176121         ADCZL   3,3,SKP
77 04177 054572 INP?J:  STA     3,INP?R         ;SAVE THE RETURN ADDRESS
78 04200 054466         STA     3,INT?E
79 04201 004543         JSR     INS?V           ;SAVE THE STATUS
80 04202 040574         STA     0,INL?K         ;"INL?K" IS NOT =1
81 04203 060610 IN0?:   DIAC    0,TTI           ;READ THE INPUT
82 04204 034760         LDA     3,P1777         ;AC3 = 177
83 04205 163400         AND     3,0             ;GET RID OF THE PARITY BIT
84 04206 034760         LDA     3,PC175         ;AC1 = 15
85 04207 116415         SUB#    0,3,SNR         ;SKIP IF THE CHARACTER TYPED
86                                              ;WAS NOT "CR"
87 04210 000426         JMP     INR?
88 04211 040557         STA     0,INS?3         ;SAVE THE CHARACTER
89 04212 024757         LDA     1,C1070         ;AC1 = 100
90 04213 034752         LDA     3,PC172
```

```
0057 FPYKL
01 04214 116414          SUB#    0,3,SZR     ;SKIP IF IT IS A LINE FEED
02 04215 034755          LDA     3,IN373     ;AC3 = 33
03 04216 162453          SUBO#   3,0,SNC     ;DON'T SKIP IF AC0 IS 33
04                                           ;OR MORE
05 04217 000451          JMP     IN1?
06 04220 107000 IN5?:    ADD     0,1         ;AC1 = 100+ ASCII VALUE OF
07                                           ;CONTROL CHARACTER
08 04221 020752          LDA     0,IN1736    ;AC0 = 136
09 04222 006100          JSR     @ITYP?E     ;TYPE A
10 04223 121000          MOV     1,0
11 04224 006100          JSR     @ITYP?E
12 04225 034562          LDA     3,I127?     ;AC3 = 121
13 04226 162015          ADC#    3,0,SNR     ;SKIP IF IT IS NOT AR
14 04227 000405          JMP     IN6?
15 04230 034744          LDA     3,IN1704    ;AC3 = 104
16 04231 116404          SUB     0,3,SZR     ;SKIP IF IT WAS A AD
17 04232 000504          JMP     IN4?
18 04233 054116          STA     3,SWREG     ;LOAD "SWREG" WITH 0
19 04234 034541 IN6?:    LDA     3,INS?      ;AC3 = ADDRESS OF THE LOCATION
20                                           ;WHERE THE PROGRAM WILL START
21 04235 054534          STA     3,INP?R
22 04236 176400 INR?:    SUB     3,3
23 04237 165000          MOV     3,1
24 04240 006102          JSR     @ICRL?F
25 04241 054722          STA     3,INT?
26 04242 030116          LDA     2,SWREG
27 04243 176220          ADCZR   3,3         ;AC3 = 77777
28 04244 173404          AND     3,2,SZR     ;SKIP IF THE SWITCHES ARE SET
29                                           ;TO ALL ZERO'S
30 04245 172000          ADC     3,2
31 04246 050116          STA     2,SWREG
32 04247 034523          LDA     3,STA?T
33 04250 175223          MOVZR   3,3,SNC     ;SKIP IF DONE BIT ON TTO IS TO
34                                           ;BE LEFT SET
35 04251 060211          NIOC    TTO
36 04252 175220          MOVZR   3,3
37 04253 175204          MOVR    3,3,SZR     ;LOAD THE CARRY BIT AND SKIP
38                                           ;IF THE INTERRUPTS ARE NOT TO
39                                           ;BE ENABLED
40 04254 125004          MOV     1,1,SZR     ;SKIP IF THE INSTRUCTION BEING
41                                           ;EXECUTED IS A "P"
42 04255 176000          ADC     3,3         ;INTERRUPTS ARE TO BE LEFT DIS-
43                                           ;ABLED
44 04256 054514          STA     3,STA?T     ;"STA?T" IS 0 IF INTERUPTS ARE
45                                           ;TO BE ENABLED AND -1 OTHERWISE
46 04257 020506          LDA     0,INS?0     ;RESTORE THE ACCUMULATORS
47 04260 024506          LDA     1,INS?1
48 04261 030506          LDA     2,INS?2
49 04262 034506          LDA     3,INS?3
50 04263 010507          ISZ     STA?T
51 04264 060177          INTEN
52 04265 002504          JMP     @INP?R      ;START EXECUTING THE USER'S
53                                           ;PROGRAM
54
55
56 04266 000000 INT?E:   0
57 04267 003767 ITPS?P:  TPS?P
```

```
1?058 FPYKL
01 04270 006102 IN1?:    JSR     @ITYP?E     ;ECHO THE CHARACTER
02 04271 034517          LDA     3,IN67?     ;AC3 = 60
03 04272 152623          SUBZR   2,2         ;AC2 = 100000
04 04273 116405          SUB     0,3,SNR     ;SKIP IF THE DIGIT TYPED WAS
05                                           ;NOT 0
06 04274 000503          JMP     IN3?
07
08 04275 151221 IN2?:    MOVZR   2,2,SKP     ;SHIFT AC2 TO RIGHT
09 04276 125521          SUBZL   1,1         ;AC1 = 1
10 04277 175405          INC     3,3,SNR
11 04300 000501          JMP     IN3?+2
12 04301 147415          AND#    2,1,SNR     ;STAY IN LOOP UNTIL ALL BITS
13                                           ;OF SWREG ARE CHECKED
14 04302 000773          JMP     IN2?
15 04303 106400          SUB     0,1         ;WHEN THE CONTROL COMES HERE
16                                           ;FOR THE FIRST TIME AC1 = 100
17 04304 135000          MOV     1,3
18 04305 151225          MOVZR   2,2,SNR
19 04306 000430          JMP     IN4?
20 04307 024503          LDA     1,IN175     ;AC1 = 15
21 04310 167004          ADD     3,1,SZR     ;SKIP IF THE COMMAND WAS "M"
22 04311 000765          JMP     IN2?+1
23
24 04312 006102 INM?:    JSR     @ICRL?F     ;TYPE A "CR" AND "LF"
25 04313 152520          SUBZL   2,2         ;SET AC2= 1
26 04314 006127          JSR     @IPDC?S     ;PRINT THE CONTENTS OF AC1
27 04315 000040          40
28 04316 125400          INC     1,1
29 04317 034472          LDA     3,IN172     ;AC3 = 12
30 04320 166452          SUBO#   3,1,SZC     ;SKIP IF AC1 IS GREATER OR EQUAL
31                                           ;TO AC3
32 04321 006746          JSR     @ITPS?P     ;PRINT A SPACE
33 04322 151124          MOVZL   2,2,SZR     ;SKIP AFTER TYPING # 15
34 04323 000771          JMP     INM?+2
35 04324 006102          JSR     @ICRL?F
36 04325 030116          LDA     2,SWREG
37 04326 151140          MOVOL   2,2         ;AC2 HAD SWITCH SETTINGS
38 04327 126560          SUBCL   1,1         ;BRING THE CARRY BIT IN AC1
39 04330 006127          JSR     @IPDC?S     ;TYPE THE CONTENTS OF AC1
40 04331 000040          40
41 04332 006735          JSR     @ITPS?P     ;TYPE A SPACE
42 04333 151124          MOVZL   2,2,SZR     ;SKIP AFTER TYPING ALL THE 16
43                                           ;BITS
44 04334 000773          JMP     .-5
45 04335 006102          JSR     @ICRL?F
46 04336 010440 IN4?:    ISZ     INL?K       ;SKIP IF THE PROGRAM IS LOCKED
47                                           ;IN SWITCH INPUT MODE
48 04337 000677          JMP     INR?
49 04340 014436          DSZ     INL?K       ;NEVER SKIP
50 04341 063610          SKPDN   TTI         ;WAIT FOR OPERATOR INPUT
51 04342 000777          JMP     .-1
52 04343 000640          JMP     IN0?
53
54
55 04344 040421 INS?V:   STA     0,INS?0     ;SAVE THE ACC.
56 04345 044421          STA     1,INS?1
57 04346 050421          STA     2,INS?2
58 04347 102560          SUBCL   0,0         ;SAVE THE CARRY
59 04350 010716          ISZ     INT?E       ;SKIP IF DDT IS ENTERED THRU
60                                           ;INTERRUPT HANDLER
```

```
 0059 FPYKL
01 04351 063577          SKPBZ   CPU             ;SKIP IF INTERRUPTS ARE NOT
02                                               ;ENABLED
03 04352 101141          MOVOL   0,0,SKP
04 04353 101120          MOVZL   0,0
05 04354 060277          INTDS
06 04355 063611          SKPDN   TTO             ;SAVE THE TTO STATUS
07 04356 101121          MOVZL   0,0,SKP
08 04357 101140          MOVOL   0,0
09 04360 040412          STA     0,STA?T
10 04361 152000          ADC     2,2
11 04362 050601          STA     2,INT?          ;FORCE THE TYPE OUTS ON
12                                               ;THE TELETYPE
13 04363 002102          JMP     @ICHL?F
14
15 04364 005400 INS?A:   JSR     0,3
16 04365 000000 INS?0:   0
17 04366 000000 INS?1:   0
18 04367 000000 INS?2:   0
19 04370 000000 INS?3:   0
20 04371 000000 INP?R:   0
21 04372 000000 STA?T:   0
22 04373 177777 INB?A:   -1
23 04374 000000 INB?I:   0
24 04375 000512 INS?:    RES?T
25 04376 000000 INL?K:   0
26
27 04377 176000 IN3?:    ADC     3,3             ;AC3 = -1
28 04400 054776          STA     3,INL?K         ;LOCK IN SWITCH INPUT MODE
29 04401 024116          LDA     1,SWREG         ;READ THE CURRRENT VALUE OF
30                                               ;"SWREG"
31 04402 133414          ANDZ    1,2,SZR         ;TAKE XOR OF AC1 AND AC2
32 04403 146401          SUB     2,1,SKP
33 04404 147000          ADD     2,1
34 04405 044116          STA     1,SWREG         ;SAVE THE NEW VALUE OF "SWREG"
35 04406 000730          JMP     IN4?
36
37
38 04407 000121 I12?1:   121
39 04410 000060 IN6?0:   60
40 04411 000012 IN1?2:   12
41 04412 000015 IN1?5:   15
42
43
```

```
 0060 FPYKL
01
02
03                       ;FILENAME= DIAGSUP
04
05 04413 054415 ENTER:   STA     3,LOOPR         ;ADDRESS OF TOP OF LOOP
06 04414 034406          LDA     3,ITR           ;THIS ROUTINE INITIALIZES
07 04415 054406          STA     3,ITRCT         ;EACH TEST
08 04416 176400          SUB     3,3
09 04417 054202          STA     3,ESWT
10 04420 054203          STA     3,ECNT
11 04421 002407          JMP     @LOOPR
12
13 04422 000144 ITR:     144
14 04423 000000 ITRCT:   0
15 04424 000000 .RTRN:   0
16 04425 000000 SAV2:    0
17 04426 000000 SAV1:    0
18 04427 000000 SAV0:    0
19 04430 000000 LOOPR:   0
20
```

```
10061 FPYKL
01 04431 054773 CYCLE:  STA     3,.RTRN         ;END OF TEST ITERATION
02 04432 063710         SKPDZ   TTI             ;IF A TTY KEY HAS BEEN HIT
03 04433 006105         JSR     @IINP?          ;THEN SERVICE THE ROUTINE TO
04                                              ;SET THE SWITCHES
05 04434 020224         LDA     0,I             ;GET "I" CODE
06 04435 116415         SUB#    0,3,SNR         ;WAS AN "I" TYPED
07 04436 002451         JMP     @.INCM          ;YES! GO TO INTERCHANGE
08 04437 020163         LDA     0,NCHG          ;GET NO CHANGE SWITCH
09 04440 101014         MOV#    0,0,SZR         ;IS IT SET
10 04441 002767         JMP     @LOOPR          ;YES! REPEAT TEST
11 04442 014761         DSZ     ITRCT
12 04443 000434         JMP     CYCTS           ;NOT 100 TIMES ITERATED
13 04444 034756         LDA     3,ITR           ;RESET ITERATION CNTR
14 04445 054756         STA     3,ITRCT
15 04446 034202         LDA     3,ESWT
16 04447 175005         MOV     3,3,SNR
17 04450 002754         JMP     @.RTRN
18 04451 034116         LDA     3,SWREG         ;PLACE THE CONTENTS OF SWREG IN
19                                              ;AC3
20 04452 177100         ADDL    3,3             ;IF THE PERCENT FAILURE IS
21 04453 175100         MOVL    3,3
22 04454 177102         ADDL    3,3,SZC         ;REQUIRED THEN SKIP THE NEXT
23                                              ;INSTRUCTION
24 04455 000417         JMP     NOSUM
25 04456 040751         STA     0,SAV0
26 04457 044747         STA     1,SAV1
27 04460 050745         STA     2,SAV2
28 04461 006102         JSR     @ICRLF          ;PRINT CARRIAGE
29 04462 024203         LDA     1,ECNT
30 04463 030423         LDA     2,ER.1
31 04464 004444         JSR     MULT
32 04465 030735         LDA     2,ITR
33 04466 004426         JSR     DIVD
34 04467 006127         JSR     @IPDC?S         ;PRINT VALUE (EXAMPLE: 89%)
35 04470 000045 PCENT:  "%                      ;CHARACTOR
36 04471 020736         LDA     0,SAV0          ;RESTORE ACS
37 04472 024734         LDA     1,SAV1
38 04473 030732         LDA     2,SAV2
39 04474 176400 NOSUM:  SUB     3,3
40 04475 054202         STA     3,ESWT
41 04476 054203         STA     3,ECNT
42 04477 034202 CYCTS:  LDA     3,ESWT
43 04500 175005         MOV     3,3,SNR         ;SKIP IF THE PROGRAM HAS NOT
44 04501 002723         JMP     @.RTRN          ;SEEN AN ERROR
45 04502 034116         LDA     3,SWREG         ;IF THE PROGRAM CONTROL HAS COME
46                                              ;HERE DUE TO AN ERROR THEN CHECK
47                                              ;IF LOOP ON ERROR IS REQUIRED OR
48                                              ;NOT
49 04503 177133         ADDZL#  3,3,SNC         ;SWITCH 1, (0) LOOP ON THE TEST
50 04504 002724         JMP     @LOOPR          ;(1)=PROCEED TO THE NEXT TEST
51 04505 002717         JMP     @.RTRN
52 04506 000144 ER.1:   144
53 04507 002563 .INCM:  INCH
```

```
10062 FPYKL
01
02
03                      ;FILENAME=MULDIV
04
05                      ;AC1 REM AC0=(AC0,AC1)/AC2
06 04510 102400 DIVID:  SUB     0,0
07 04511 054431 DIVD0:  STA     3,MDV.5
08 04512 142432         SUBZ#   2,0,SZC
09 04513 000413         JMP     MDV.3
10 04514 054426 DIVD:   STA     3,MDV.5         ;DIVIDE
11 04515 034426         LDA     3,MDV.1
12 04516 125120         MOVZL   1,1
13 04517 101100 MDV.2:  MOVL    0,0
14 04520 142412         SUB#    2,0,SZC
15 04521 142400         SUB     2,0
16 04522 125100         MOVL    1,1
17 04523 175404         INC     3,3,SZR
18 04524 000773         JMP     MDV.2
19 04525 176441         SUBU    3,3,SKP
20 04526 176420 MDV.3:  SUBZ    3,3
21 04527 002413         JMP     @MDV.5
22
23                      ;(AC0,AC1)=AC1*AC2
24 04530 102460 MULT:   SUBC 0,0                ;MULTIPLY
25                      ;(AC0,AC1)=AC1*AC2+AC0
26 04531 054411 MULTA:  STA     3,MDV.5
27 04532 034411         LDA     3,MDV.1
28 04533 125203 MDV.4:  MOVR    1,1,SNC
29 04534 101261         MOVR    0,0,SKP
30 04535 143220         ADDZR   2,0
31 04536 175404         INC     3,3,SZR
32 04537 000774         JMP     MDV.4
33 04540 125260         MOVCR   1,1
34 04541 002401         JMP     @MDV.5
35 04542 000000 MDV.5:  0
36 04543 177760 MDV.1:  -20
```

```
10063 FPYRL
01      006101 MESSAGE=JSR @IMESTS      ;TYPE FOLLOWING MESSAGE IN ASCII
02      006145 INPUT=JSR @IINP          ;GET A LINE OF INPUT
03      006146 GETPAR=JSR @IHSS         ;GET HEAD-SECT-# SECT FROM INPUT LINE
04      006147 GETATM=JSR @IGATM        ;GET NAME OR # FROM INPUT LINE
05      006151 SEARCH=JSR @ISRH         ;SEARCH FOLLOWING TABLE FOR MATCH ON AC1
06      006102 PCRLF=JSR @.CRLF         ;DO CARRIAGE RETURN/LINE FEED ON TTY
```

```
10064 FPYRL
01                      ;SEARCH SUBROUTINE
02                      ;CALL SEARCH    AC1 = NAME (3 PACKED CHAR'S)
03                      ;RETURN - NOT FOUND
04                      ;RETURN - FOUND, AC2 = POINTER
05
06 04544 031400 SRH:    LDA 2,0,3       ;AC2 HAS TABLE ADDRESS
07 04545 021000         LDA 0,0,2       ;GET WORD
08 04546 101005         MOV 0,0,SNR     ;0 WORD ENDS TABLE
09 04547 001401         JMP 1,3         ;RETURN, NOT FOUND
10 04550 122415         SUB# 1,0,SNR    ;COMPARE
11 04551 001402         JMP 2,3         ;RETURN, FOUND - AC2 = POINTER
12 04552 151400         INC 2,2         ;INCREMENT POINTER
13 04553 000772         JMP SRH+1
14
```

```
10065 FPYRL
01
02                ;       ************************************************
03                ;       ******          COMMAND STRING INTERPRETER    *****
04                ;       ************************************************
05
06                        ;GET COMMAND STRING PARAMETERS
07                        ;UNIT SELECT
08 04554 000101 QUEST:    MESSAGE
09 04555 000334           MSG43               ; -?-
10 04556 000406           JMP CMDST+2
11 04557 006010           DIAC 1,TTI          ;GET TTI CHAR
12 04560 020232           LDA 0,TMSK
13 04561 107400           AND 0,1             ;MASK OUT PARITY
14 04562 102520 CMDST:    SUBZL 0,0           ;SET THE LSI FLAG
15 04563 040155           STA 0,CSIF
16 04564 102400           SUB 0,0             ;CLEAR (CR) FLAG
17 04565 040154           STA 0,TERM
18 04566 006210           NIOC TTI            ;CLEAR TTI FLAG
19 04567 006233           NIOC FPY            ;CLEAR DISK GARBAGE
20 04570 020156           LDA 0,C33           ;ESCAPE KEY?
21 04571 100405           SUB 0,1,SNR
22 04572 000442           JMP GCS             ;YES
23 04573 000102           PCRLF
24 04574 000101           MESSAGE
25 04575 000337           MSG17               ;"UNIT: "
26 04576 000145           INPUT               ;ACCEPT INPUT
27 04577 000160           UBP
28 04600 000412           JMP CMDC
29 04601 000147           GETATM
30 04602 125004           MOV 1,1,SZR         ;AC0=# AC1=NAME
31 04603 000751           JMP QUEST           ;NAMES NOT LEGAL
32 04604 024157           LDA 1,C2
33 04605 122432           SUBZ# 1,0,SZC
34 04606 000746           JMP QUEST           ;UNIT #>1
35 04607 101220           MOVZR   0,0
36 04610 101200           MOVR    0,0
37 04611 040204           STA 0,UNIT
38 04612 004406 CMDC:     JSR CMD,2           ;GET DATA ROUTINE
39 04613 000421           JMP GCS             ;COMMAND STRING ROUTINE
40
```

```
10066 FPYRL
01
02                        ;DATA SELECT
03
04 04614 000000 CMD2R:    0
05 04615 000101           MESSAGE
06 04616 000334           MSG43               ; -?-
07 04617 100402           JMP .+2
08 04620 054774 CMD.2:    STA 3,CMD2R
09 04621 000102           PCRLF
10 04622 000101           MESSAGE
11 04623 000006           DATA                ;"DATA: "
12 04624 000165           JSR     @,TINO
13 04625 000770           JMP     CMD2R+1
14 04626 030143           LDA     2,,PTAB
15 04627 133000           ADD     1,2
16 04630 021000           LDA     0,0,2
17 04631 042136           STA     0,@,GEN
18 04632 006141           JSR     @,DGEN
19 04633 002761           JMP @CMD2R          ;RETURN
```

```
10007 FPYRL
01
02
03
04                      ;GET COMMAND STRING
05
06 04634 000102 GCS:    PCRLF
07 04635 000161         MESSAGE
08 04636 005343         MSG19           ;"COMMAND STRING: "
09 04637 060233         NIOC FPY        ;CLEAR DISK
10 04640 000145         INPUT           ;ACCEPT INPUT
11 04641 000161         CSBP            ;CMD STR BYTE POINTER
12 04642 000401         JMP .+1         ;SAME AS BEFORE
13 04643 020161         LDA 0,CSBP      ;SET FOR RETURN TO BEGINNING
14 04644 042413         STA 0,@CSER+3   ;CSNL
15 04645 063710 GCS.1:  SKPDZ TTI
16 04646 000711         JMP CMDST-3     ;INTERRUPT CMD STRING
17 04647 000147         GETATM          ;LINCT=BYTE POINTER
18 04650 000151         SEARCH
19 04651 005315         DCT             ;NAME IN AC1
20 04652 000402         JMP CSER        ;NOT FOUND
21 04653 003010         JMP @DCT,1-DCT,2;DISPATCH
22
23 04654 000101 CSER:   MESSAGE         ;COMMAND STRING ERROR
24 04655 005334         MSG43           ; -?-
25 04656 000756         JMP GCS
26 04657 004743         CSNL
```

```
10008 FPYRL
01
02                      ;READ COMMAND
03
04 04660 101002 RE:     MOV 0,0,SZC
05 04661 000773         JMP CSER        ;CR TERMINATOR ILLEGAL
06 04662 102440         SUBO    0,0
07 04663 040172         STA     0,RWS           ;SET READ/WRITE FOR READ
08 04664 000146         GETPAR          ;GET R/W PARAMETERS
09 04665 004660 RE.4:   JSR     GADDR
10 04666 020152         LDA     0,STAD
11 04667 040020         STA     0,AUTO
12 04670 000134         JSR     @,RDY
13 04671 020204 RE.3:   LDA     0,UNIT          ;GET UNIT NUMBER
14 04672 024235         LDA     1,RPOP          ;GET BASE OPCODE
15 04673 030172         LDA     2,RWS
16 04674 151014         MOV#    2,2,SZR
17 04675 125120         MOVZL   1,1
18 04676 125120         MOVZL   1,1             ;GENERATE READ OPCODE
19 04677 123000         ADD     1,0             ;GENERATE PARTIAL COMMAND
20 04700 024211         LDA     1,SEC           ;GET SECTOR NUMBER
21 04701 125300         MOVS    1,1             ;ALLIGN IT
22 04702 123000         ADD     1,0             ;COMPLETE COMMAND WORD
23 04703 042020         STA     0,@AUTO         ;PUT IT IN STACK
24 04704 014212         DSZ     SC              ;DONE ALL SECTORS
25 04705 000407         JMP     RE.2            ;NO: DO MORE
26 04706 102000         ADC     0,0
27 04707 040213         STA     0,CSIS          ;SET CSI SWITCH
28 04710 042020         STA     0,@AUTO         ;TERMINATE STACK
29 04711 000126         JSR     @.VADD
30 04712 000150         JSR     @IEXEC          ;GO DO IT
31 04713 000403         JMP     RE.1
32 04714 010211 RE.2:   ISZ     SEC             ;BUMP SECTOR NUMBER
33 04715 000754         JMP     RE.3            ;SET UP NEXT COMMAND
34 04716 020154 RE.1:   LDA 0,TERM
35 04717 101004         MOV 0,0,SZR
36 04720 002402         JMP @.+2                ;ALL DONE
37 04721 000724         JMP GCS.1       ;GET ANOTHER COMMAND
38 04722 004562         CMDST
39
40                      ;WRITE COMMAND
41 04723 101002 WT:     MOV 0,0,SZC
42 04724 000730         JMP CSER        ;CR TERMINATOR ILLEGAL
43 04725 102000         ADC     0,0
44 04726 040172         STA     0,RWS
45 04727 101400         INC     0,0
46 04730 000146         GETPAR          ;GET R/W PARAMETERS
47 04731 020212         LDA     0,SC            ;GET SECTOR COUNT
48 04732 126520         SUBZL   1,1
49 04733 106435         SUBZ#   0,1,SNR         ;IS IT > 1
50 04734 000731         JMP RE.4
51 04735 000120         JSR     @.DVR           ;YES: GO DO FULL TRACK
52 04736 000040         40                      ;WRITE OPCODE
53 04737 000757         JMP     RE.1            ;EXIT
54
```

```
10069 FPYRL
01
02
03 04740 020162 CSL:    LDA 0,LINCT     ;SET UP FOR LOOP TO NEXT COMM WORD
04 04741 040402         STA 0,CSNL
05 04742 002402         JMP @.+2
06 04743 000000 CSNL:   0
07 04744 004645         GCS.1
08
09
10
11 04745 054423 GADDR:  STA     3,ARTN          ;SAVE RETURN ADDRESS
12 04746 006122         JSR     @.DARD          ;READ DISK ADDRESS
13 04747 024232         LDA     1,TMSK          ;GET TRACK MASK
14 04750 101300         MOVS    0,0
15 04751 123400         AND     1,0             ;MASK OFF SECTOR
16 04752 040210         STA     0,TRK           ;SAVE THE TRACK #
17 04753 006144         JSR     @.GT43          ;CHECK FOR > 43
18 04754 024210         LDA     0,TRK           ;GET TRACK #
19 04755 101300         MOVS    0,0             ;RE-ALLIGN IT
20 04756 024211         LDA     1,SEC           ;GET SECTOR
21 04757 034233         LDA     3,SMSK          ;GET SECTOR MASK
22 04760 152520         SUBZL   2,2
23 04761 125014         MOV#    1,1,SZR         ;IS SECTOR ZERO
24 04762 146401         SUB     2,1,SKP         ;SET TO SYNC ON SECTOR
25 04763 167000         ADD     3,1             ;YES: MAKE IT SEVEN
26 04764 127120         ADDZL   1,1             ;ALLIGN SECTOR
27 04765 123000         ADD     1,0             ;GENERATE ADDRESS
28 04766 040206         STA     0,NADD          ;SAVE IT
29 04767 002401         JMP     @ARTN
30
31 04770 000000 ARTN:   0
32
33
34 04771 101002 SK:     MOV     0,0,SZC
35 04772 000602         JMP     CSER
36 04773 102000         ADC     0,0
37 04774 006146         GETPAK
38 04775 006134         JSR     @.RDY
39 04776 006140         JSR     @.SEEK
40 04777 000717         JMP     RE.1
41
42 05000 006125 RCL:    JSR     @.RCAL
43 05001 000715         JMP     RE.1
44
45
46 05002 020741 LUP:    LDA     0,CSNL
47 05003 040162         STA     0,LINCT
48 05004 000641         JMP     GCS.1
49
50
51 05005 002401 RES:    JMP     @.+1
52 05006 000515         HADI
```

```
10070 FPYRL
01
02              ;       ***********************************************
03              ;       ****    GENERAL PURPOSE SUBROUTINES        *****
04              ;       ***********************************************
05
06
07              ;GET SECTOR=SECTOR COUNT
08              ;SET (C)=1 ON RETURN IF CR TERMINATOR
09
09 05007 054447 HSS:    STA 3,HSSRET
10 05010 101133         MOVZL#  0,0,SNC
11 05011 000413         JMP     NOSK
12 05012 006147         GETATM
13 05013 125002         MOV     1,1,SZC
14 05014 002153         JMP     @IGCS
15 05015 125004         MOV     1,1,SZR
16 05016 002153         JMP     @IGCS
17 05017 024231         LDA     1,LTRK
18 05020 106433         SUBZ#   0,1,SNC
19 05021 002153         JMP     @IGCS
20 05022 040210         STA     0,TRK
21 05023 002030         JMP     HSS.1
22 05024 006147 NOSK:   GETATM                  ;GET SECTOR #
23 05025 125002         MOV 1,1,SZC             ;CR ONLY?
24 05026 002153         JMP @IGCS
25 05027 125004         MOV 1,1,SZR             ;LETTER = RANDOM
26 05030 002153         JMP @IGCS
27 05031 024233         LDA 1,SMSK              ;CHECK SECTOR LIMIT
28 05032 106433         SUBZ# 0,1,SNC
29 05033 002153         JMP @IGCS               ;LIMIT EXCEEDED
30 05034 040211         STA 0,SEC               ;SECTOR #
31 05035 006147         GETATM                  ;GET # OF SECTORS
32 05036 034155         LDA 3,CSIP              ;IN COMMAND STRING?
33 05037 125005         MOV 1,1,SNR             ;LETTER ?
34 05040 000404         JMP .+4                 ;NO
35 05041 175004         MOV 3,3,SZR             ;ALLOW IF NOT CSI
36 05042 002153         JMP @IGCS               ;CSI
37 05043 000407         JMP .+7
38 05044 101005         MOV 0,0,SNR             ;NO SECTOR COUNT,
39 05045 002153         JMP @IGCS
40 05046 024235         LDA 1,RPOP              ;CHECK AGAINST MAX # OF SECTORS
41 05047 106433         SUBZ# 0,1,SNC
42 05050 002153         JMP @IGCS
43 05051 040212         STA 0,SC                ;# OF SECTORS
44 05052 040215         STA     0,NSEC
45 05053 102560 HSS.1:  SUBCL 0,0
46 05054 040154         STA 0,TERM
47 05055 002401         JMP @HSSRET
48 05056 000000 HSSRET: 0
```

```
10071 FPYRL
01
02                      ;LINE SCAN SUBROUTINE
03                      ;BYTE POINTER IS LINCT
04
05                      ;CALL   GETATM
06                      ;       RETURN + AC0= #
07                      ;                  AC1=NAME
08                      ;                  (C)=1 IF CR DELIMITER
09
10 05057 020510 GA.0:   LDA 0,FIND
11 05060 101005         MOV 0,0,SNR
12 05061 000413         JMP GA.1        ;NO NAME OR # YET
13 05062 020162         LDA 0,LINCT
14 05063 040502         STA 0,SLNCT     ;REMEMBER LINCT
15 05064 040502         STA 0,SRF       ;SET SEARCH FLAG
16 05065 000407         JMP GA.1
17
18 05066 054474 GATM:   STA 3,GARET
19 05067 102400         SUB 0,0
20 05070 040473         STA 0,ANAM
21 05071 040473         STA 0,ANUM
22 05072 040474         STA 0,SRF
23 05073 040474         STA 0,FIND
24 05074 030162 GA.1:   LDA 2,LINCT
25 05075 010162         ISZ LINCT
26 05076 151220         MOVZR 2,2
27 05077 021000         LDA 0,0,2
28 05100 024232         LDA 1,TMSK
29 05101 101002         MOV 0,0,SZC
30 05102 101300         MOVS 0,0
31 05103 123400         AND 1,0         ;(AC0)R = BYTE
32 05104 024236         LDA 1,SP
33 05105 122415         SUB# 1,0,SNR
34 05106 000751         JMP GA.0        ;SPACE DELIMITER
35 05107 024462         LDA 1,C54
36 05110 122415         SUB# 1,0,SNR
37 05111 000746         JMP GA.0        ;COMMA DELIMITER
38 05112 024221         LDA 1,P.C15
39 05113 122415         SUB# 1,0,SNR
40 05114 000442         JMP EXIT        ;CR DELIMITER
41 05115 024451         LDA 1,SRF
42 05116 125004         MOV 1,1,SZR
43 05117 000434         JMP EX.1        ;SEARCH FLAG ON
44 05120 030217         LDA 2,P,C60
45 05121 034222         LDA 3,C67
46 05122 162033         ADCZ# 3,0,SNC
47 05123 112032         ADCZ# 0,2,SZC
48 05124 000412         JMP ASSN        ;NOT #
49 05125 024233         LDA 1,SMSK
50 05126 123400         AND 1,0
51 05127 024435         LDA 1,ANUM
52 05130 127120         ADDZL 1,1       ;ASSEMBLE OCTAL #
53 05131 125120         MOVZL 1,1
54 05132 123000         ADD 1,0
55 05133 040431         STA 0,ANUM
56 05134 010433         ISZ FIND
57 05135 000737         JMP GA.1        ;GET MORE
```

```
10072 FPYRL
01
02 05136 024434 ASSN:   LDA 1,C37       ;ASSEMBLE NAME
03 05137 123400         AND 1,0
04 05140 024423         LDA 1,ANAM
05 05141 030427         LDA 2,C176K     ;3 LETTERS YET ?
06 05142 133404         AND 1,2,SZR
07 05143 000731         JMP GA.1        ;YES, IGNORE THE REST
08 05144 127120         ADDZL 1,1
09 05145 127120         ADDZL 1,1       ;5 LEFT
10 05146 125120         MOVZL 1,1
11 05147 123000         ADD 1,0
12 05150 040413         STA 0,ANAM
13 05151 010416         ISZ FIND
14 05152 000722         JMP GA.1        ;GET MORE
15
16 05153 020412 EX.1:   LDA 0,SLNCT
17 05154 040162         STA 0,LINCT
18 05155 101021         MOVZ 0,0,SKP    ;CLEAR CARRY
19 05156 101040 EXIT:   MOVO 0,0        ;SET CARRY
20 05157 020405         LDA 0,ANUM
21 05160 024403         LDA 1,ANAM
22 05161 002401         JMP @GARET
23
24 05162 000000 GARET:  0
25 05163 000000 ANAM:   0
26 05164 000000 ANUM:   0
27 05165 000000 SLNCT:  0
28 05166 000000 SRF:    0
29 05167 000000 FIND:   0
30 05170 176000 C176K:  176000
31 05171 000054 C54:    54
32 05172 000037 C37:    37
```

```
00073 FPYRL
01                      ;TELETYPE INPUT ROUTINE, NON INTERRUPT
02                      ;
03                      ;CALL   INPUT
04                      ;       ADDR OF BYTE POINTER
05                      ;       RETURN - CR ONLY
06                      ;       RETURN - NORMAL
07                      ;INPUT IS STORED R-L IN 7 BIT ASCII
08                      ;INPUT IS TERMINATED BY CR, (215) IS
09                      ;STORED.  LINE FEED ECHOS CR=LF
10                      ;NO DATA STORED, INPUT CONTINUES.
11
12 05173 054515 INP:    STA 3,INPRET
13 05174 024000         LDA 0,@0,3      ;GET BYTE POINTER
14 05175 040512         STA 0,BASE
15 05176 040513         STA 0,BPTR
16 05177 060214         NIOC TTI
17 05200 020513         LDA 0,M115.
18 05201 040511         STA 0,CHCNT
19 05202 063610 TTWAIT: SKPDN TTI       ;WAIT FOR INPUT
20 05203 000777         JMP .-1
21 05204 063611         DIAC 1,TTI      ;READ INPUT CHAR
22 05205 020232         LDA 0,TMSK
23 05206 107400         AND 0,1
24 05207 106415         SUB# 0,1,SNR
25 05210 000442         JMP RUB         ;RUB OUT
26 05211 020503         LDA 2,C12
27 05212 106414         SUB# 0,1,SZR
28 05213 000403         JMP .+3
29 05214 006102         PCRLF           ;LINE FEED TYPED
30 05215 000765         JMP TTWAIT
31 05216 063111         DOAS 1,TTO      ;ECHO CHAR
32 05217 030472         LDA 2,BPTR
33 05220 010471         ISZ BPTR
34 05221 034221         LDA 3,P.C15     ;CR CODE
35 05222 136405         SUB 1,3,SNR
36 05223 000456         JMP CRCOD       ;CR TYPED
37 05224 020232         LDA 0,TMSK      ;7 BIT MASK
38 05225 151223 INP.0:  MOVZR 2,2,SNC
39 05226 107401         AND 0,1,SKP     ;STORE INTO RH
40 05227 107701         ANDS 0,1,SKP
41 05230 000403         JMP .+3
42 05231 021000         LDA 0,0,2
43 05232 107000         ADD 0,1
44 05233 045000         STA 1,0,2       ;STORE BYTE
45 05234 010456         ISZ CHCNT
46 05235 000402         JMP .+2
47 05236 000435         JMP OVFL
48 05237 175004         MOV 3,3,SZR     ;AC3=0 IF CR TYPED
49 05240 000742         JMP TTWAIT
50 05241 010447         ISZ INPRET      ;ALL DONE
51 05242 006102 INP.1:  PCRLF
52 05243 020444         LDA 0,BASE
53 05244 040162         STA 0,LINCT
54 05245 010443         ISZ INPRET
55 05246 063511         SKPBZ TTO       ;WAIT FOR ECHO TO
56 05247 000777         JMP .-1         ;FINISH
57 05250 060211         NIOC TTO        ;CLEAR INTERRUPT
58 05251 002437         JMP @INPRET
```

```
00074 FPYRL
01
02 05252 020441 RUB:    LDA 0,M115.
03 05253 024437         LDA 1,CHCNT
04 05254 122405         SUB 1,0,SNR
05 05255 000725         JMP TTWAIT      ;NOTHING TO RUB
06 05256 014433         DSZ BPTR
07 05257 014433         DSZ CHCNT
08 05260 030431         LDA 2,BPTR
09 05261 151220         MOVZR 2,2
10 05262 021000         LDA 0,0,2       ;GET BYTE
11 05263 101003         MOV 0,0,SNC
12 05264 000405         JMP RUB1
13 05265 024216         LDA 1,P.377
14 05266 107400         AND 0,1         ;SAVE RH ONLY
15 05267 045000         STA 1,0,2
16 05270 101300         MOVS 0,0        ;ECHO LH (RUBBED OUT)
17 05271 061111 RUB1:   DOAS 0,TTO      ;ECHO RUBBED CHAR
18 05272 000710         JMP TTWAIT
19
20 05273 006102 OVFL:   PCRLF           ;LINE OVERFLOW
21 05274 006101         MESSAGE
22 05275 005354         MSG21           ;"INPUT OVERFLOW"
23 05276 014413         DSZ BPTR        ;BACKUP BYTE POINTER
24 05277 014413         DSZ CHCNT
25 05300 000702         JMP TTWAIT
26
27 05301 020406 CRCOD:  LDA 0,BASE
28 05302 112415         SUB# 0,2,SNR
29 05303 000737         JMP INP.1       ;YES
30 05304 020216         LDA 0,P.377
31 05305 176400         SUB 3,3
32 05306 000717         JMP INP.0
33
34 05307 000000 BASE:   0
35 05310 000000 INPRET: 0
36 05311 000000 BPTR:   0
37 05312 000000 CHCNT:  0
38 05313 177615 M115.:  -115.
39 05314 000012 C12:    12
```

```
10075 FPYRL
01                          ;DISK COMMAND TABLE
02
03 05315 044241 DCT:        044241          ;READ
04 05316 057111             057111          ;WRITE
05 05317 046245             046245          ;SEEK
06 05320 044243             044243          ;RECALIBRATE
07 05321 030757             030757          ;LOOP
08 05322 000622             000622          ;LR (LOOP RETURN)
09 05323 044263             044263          ;RESTART
10 05324 000000             0
11
12                          ;DISPATCHES
13
14 05325 004600 DCT,1:      RE
15 05326 004723             WT
16 05327 004771             SK
17 05330 005000             RCL
18 05331 005002             LUP
19 05332 004740             CSL
20 05333 005005             RES
```

```
10076 FPYRL
01                          ; =?=
02 05334 026640 MSG43:      .TXTE ! =?=!
03 05337 047125 MSG17:      .TXT    /UNIT: /
04 05343 047503 MSG19:      .TXT    /COMMAND STRING<15><12>/
05 05354 047111 MSG21:      .TXT    /INPUT OVERFLOW<15><12>/
06       000000             .NOLOC  0
```

```
10077 FPYRL
01
02 05365 000012 UBUFF:  .BLK 10.
03 05377 000060 CBUFF:  .BLK    50
04
05
06
07
08
```

```
10078 FPYRL
01 05457 000000 .EGGS:  0                ;DTOS SWITCH
02 05460 000000         0                ;DEVICE CODE
03 05461 000000         0                ;CAT SWITCH
04 05462 000000         0                ;NUMBER OF PASSES
05 05463 000000         0                ;RETURN ADDRESS
06 05464 000000         0                ;SWITCH REGISTER
07
08 05465 050306 DIRT:   .TXTE            ;FPY RELI  0/
09       120131
10       142722
11       144714
12       120240
13       000060
14 05473 000000         0
15 05474 000200         .STRT
16 05475 000000         0
17 05476 070707         70707
18 05477 070707         70707
19 05500 070707         70707
20 05501 070707         70707
21 05502 100033         100033
22 05503 000000 LAST:   0
23
24
25
26 05504 047503         .TXT    /COPYRIGHT (C) DGC, 1976
27       054520
28       044522
29       044107
30       020124
31       041450
32       020051
33       043504
34       020103
35       030440
36       033471
37 05517 040466 ALL RIGHTS RESERVED,/
38       046114
39       051040
40       043511
41       052110
42       020123
43       042522
44       042523
45       053122
46       042105
47       000056
48
49                      .END .STRT
50 00000-000055

**00000 TOTAL ERRORS, 0000 PASS 1 ERRORS
```

0089 FPYRL

	16/33	16/46	17/13	17/34	29/16	39/47	69/42
.RDY 000134	8/42	19/10	20/02	24/33	26/15	32/09	68/12
	69/38						
.RTRN 004424	60/15	61/01	61/17	61/44	61/51		
.SCHK 000121	8/31	19/14	24/45				
.SEEK 000140	8/46	18/30	69/39				
.STP 000117	8/29	17/38	20/21	29/19	32/18		
.STRT 000200	8/06	9/17	78/15	78/49			
.TCHK 000131	8/39	15/33	15/46	15/59	16/12	16/25	16/38
	17/05	19/21					
.TEST 000132	8/40	15/34	15/47	15/60	16/13	16/26	16/39
	17/06	19/20					
.TIND 000137	8/45	13/01	39/18	39/23			
.TINO 000105	8/19	10/04	12/03	12/23	12/57	14/18	39/05
	66/12						
.TSTC 000110	8/22	15/29	15/42	15/55	16/08	16/21	16/34
	16/47	39/48					
.VADD 000126	8/36	13/46	19/01	25/19	68/29		
.WADN 000133	8/41	19/13	20/27	24/43	26/08	26/20	33/17
.WRT 000135	8/43	40/04					
.ZOCT 000104	8/18	10/03	11/36	15/19	15/24	15/38	15/51
	16/04	16/17	16/30	16/43	27/20	27/34	34/40
	34/59	35/01	35/03	35/05	35/07		