```
0001  .MAIN
01
02
03              ;                                   RCSL:   44-RT1667
04              ;                                   AUTHOR: H.O.HANSEN
05              ;                                   EDITED: 78.03.15
06                                                  ID:60
07
08
09
10
11                  1. FDC    DEV 61
12
13                  2. FDC    DEV 64
14
15
16
17
18
19
20
21
22
23
24
25              ;                       PROGRAM
26              ;                       *******
27
28
29              ;                   FDC 703  TESTPROGRAM
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48              ;       KEYWORDS:   RC3600 FLOPPY DISC CONTROLLER, STAND ALONE
49              ;                   TESTPROGRAM.
50              ;
51              ;       ABSTRACT:   THE PROGRAM IS A DIAGNOSTICTEST TO CHECK THE
52              ;                   CONTROLLER FDC703, AND THE MAIN-FUNCTIONS OF
53              ;                   THE DISC-DRIVE.
54
55
56
57
58
59              ;       ASCII TAPE:                 RCSL: 44-RT1668  (MAG.TAPE)
60              ;       S-ABS.BINARY TAPE:          RCSL: 44-RT1669  (WITH SELFSTART
61              ;                                                     AUTOLOAD HEADER)
62              ;       BINARY CARDS:               RCSL: 44-RT1670
63              ;       PREWR. TEST DISCETTE:       RCSL: 44-RT1570
```

```
| 0002  .MAIN
01
02                ;            ****************************
03                ;            *                          *
04                ;            *  REVISION HISTORY:       *
05                ;            *                          *
06                ;            *  REV.    DATE            *
07                ;            *                          *
08                ;            *  00      77.09.07        *
09                ;            *  01      77.10.14        *
10                ;            *  02      78.03.15        *
11                ;            *                          *
12                ;            ****************************
13                ;
14                ;
15                ;;0. INDEX
16                ;
17                ;       1. ABSTRACT                        PAGE 2
18                ;       2. MACHINE REQUIREMENTS            PAGE 2
19                ;       3. SWITCH SETTINGS AND INIMESS     PAGE 3
20                ;       4. OPERATING PROCEDURE             PAGE 6
21                ;       5. PROGRAM DESCRIPTION             PAGE 7
22                ;       6. STRAPPINGS ETC.                 PAGE 10
23                ;       7. STATUS BIT TABLE                PAGE 10
24                ;       8. SOFTWARE SURVEY                 PAGE 11
25                ;       9. BELONGING TESTEQUIPMENT         PAGE 11
26                ;       10. MESSAGES FROM THIS TEST        PAGE 11
27                ;
28                ;
29                ;
30                ;
31                ;1. ABSTRACT
32                ;
33                ;       THE TESTPROGRAM WILL, AFTER START IN ONE OF THE DIFFE-
34                ;       RENT STARTADDRESSES, CHECK THE FUNCTIONS OF THE DISC-
35                ;       CONTROLLER AND DETECT HARDWARE-FAILURES. A RUN IS ALWAYS
36                ;       INITIATED WITH A SIMPLE BUS-TEST, FOLLOWED BY TESTS OF
37                ;       THE HARDWARE IN INCREASING COMPLEXITY.
38                ;       IT'S POSSIBLE TO SEPERATE AND RUN DIFFERENT PARTS OF
39                ;       THE TEST. BY SELECTING THE BELONGING STARTADDRESS, YOU
40                ;       MAY RUN THE COMPLETE TEST, TEST OF WRITE LOGIC, TEST OF
41                ;       READ LOGIC ETC.
42                ;
43                ;     * NOTICE, THAT SOME OF THE TESTS IMPLIES, THAT THE DRIVE
44                ;     * IS LOADED WITH THE PRE-WRITTEN TEST-DISCETTE (SEE SEC. 9)
45                ;
46                ;       THE REAL HARDWARE-ERRORS ARE ALWAYS ANNOUNCED, BUT IN
47                ;       CASE OF ERRORS, WHICH MAY BE CAUSED BY THE MEDIUM (E.G.
48                ;       PARITY-ERRORS) A NUMBER OF RETRIES MAY BE EXECUTED BEFORE
49                ;       ERROR-HALT (THIS NUMBER IS INITIALLY = 0).
50                ;
51                ;
52                ;2. MACHINE REQUIREMENTS
53                ;
54                ;       RC 3600 FAMILY PROCESSOR                  CPU
55                ;       MINIMUM 8K READ/WRITE MEMORY              MEM
56                ;       TELETYPE OR OPERATORS CONTROL PANEL       TTY/OCP *NOTE
57                ;       LINEPRINTER, OPTIONALLY                   LPT     *NOTE
58                ;
59                ;       *NOTE: IF THIS DEVICES IS SET ONLINE AFTER PROGRAM
60                ;       START, THE PROGRAM MUST BE RESTARTED TO GET OUTPUT.
```

```
| 0003  .MAIN
01                ;
02                ;
03                ;3. SWITCH SETTINGS AND INITIAL MESSAGES:
04                ;
05                ;       3.1 STARTING ADDRESSES - METHOD 3.3 START !
06                ;
07                ;               400 FDC-DIAGNOSTIC TEST. START WITH AN OPEN, UN-
08                ;                   LOADED DRIVE, AND INSERT THE PRE-WRITTEN
09                ;                   TEST DISCETTE WHEN DEMANDED BY PROGRAM.
10                ;                   THIS TEST WILL TEST ALL FUNCTIONS OF THE
11                ;                   CONTROLLER, INCLUDING THE WRITE/READ-TEST.
12                ;                   THE WRITETEST WILL DAMAGE THE PREWRIT-
13                ;                   TEN TEST DISCETTE, AND THEREFORE THE FINAL
14                ;                   "TESTLOOP" IN EACH PASS IS A RE-GENERATION
15                ;                   OF THE TESTDISCETTE.
16                ;                   A PASS WILL TAKE ABOUT 13 MIN.
17                ;
18                ;               401 SIMPLE CONTROLLER TEST. THE DRIVE HAS TO BE
19                ;                   OPEN AN UNLOADED. A PART (LITTLE) OF THE
20                ;                   CONTROLLER HARDWARE IS TESTED.
21                ;
22                ;               402 FDC-READTEST. LOAD THE DRIVE WITH THE PRE-
23                ;                   WRITTEN TEST DISCETTE.
24                ;
25                ;               403 FDC-WRITE/READ TEST. LOAD THE DRIVE WITH A
26                ;                   SCRATCH DISCETTE AND RELEASE THE "WRITE PRO"-
27                ;                   SWITCH. THE PROGRAM WILL EXECUTE A DIAGNOS-
28                ;                   TIC TEST, FOLLOWED BY A WRITE/READ-TEST.
29                ;
30                ;               404 FDC-DIAGNOSTIC TEST. LOAD THE DRIVE WITH THE
31                ;                   PRE-WRITTEN TEST DISCETTE AND RELEASE THE
32                ;                   "WRITE PRO"-SWITCH. THIS TEST IS BASICLY THE
33                ;                   THE SAME AS SA 400, WITH EXCEPTION OF A FEW
34                ;                   STATUS-TESTS.
35                ;
36                ;               405 PROGRAM TO WRITE AND CHECKREAD ALL SECTORS
37                ;                   OF THE DISCETTE. DATA IS ALTERNATING ZEROS
38                ;                   AND ONES. LOAD THE DRIVE WITH A SCRATCH-
39                ;                   DISCETTE.
40                ;
41                ;               406 PROGRAM TO READ ALL SECTORS OF THE DISCETTE.
42                ;                   NO DATACHECK IS PERFORMED, AND IT'S POSSIBLE
43                ;                   TO NEGLECT ALL ERRORS (FOR CONTINUESLY SCOPING
44                ;                   ANY DISCETTE MAY BE LOADED.
45                ;
46                ;               407 ROUTINE TO CHANGE DEVICE-CODE IN ALL I/O-
47                ;                   INSTRUCTIONS TO FDC. START AND ANSWER THE
48                ;                   QUESTION FOLLOWED BY NL AND WAIT FOR NEW
49                ;                   START ADDRESS QUESTION.
50                ;
51                ;               410 PROGRAM TO GENERATE THE PRE-WRITTEN TEST
52                ;                   DISCETTE ON A FAULT-FREE EQUIPMENT. LOAD A
53                ;                   SCRATCH DISCETTE AND RELEASE "WRITE PRO"-SW.
54                ;
55                ;              1400 RESTART PROGRAM AS WHEN LOADED, BUT NOW AN-
56                ;                   NOUNCING MEM. SIZE,CPU-TYPE ETC
57                ;              2202 GET A NEW PRINTER ALPHABETH (SEE 4.1.1)
58                ;              2204 SET TO 64K WORDS MODE, MEM SIZE ?
59                ;              2206 SET TO 32K WORDS MODE, MEM SIZE ?
60                ;              2210 EXAMINE MEMORY
61                ;              2212 DEPOSIT MEMORY
62                ;              2214 TROUBLE BREAKPOINT HALT
63                ;              2216 TROUBLE BREAKPOINT LOOP REPORT
64                ;              2220 TROUBLE BREAKPOINT RESET
65                ;              2222 START BINARY LOADER, READ FROM PTR/TTI (SW0)
```

```
| 0004  .MAIN
01
02                ;        3.2 CONTROL BY SWITCH SETTING, THE STATE IS REPORTED
03                ;            ON TELETYPE AND LINEPRINTER.
04                ;
05                ;                SW0:  LOOP IN TEST IN CASE OF ERROR.
06                ;                SW10: INHIBIT PRINTOUT IN TESTLOOP-PROGRAM.
07                ;                SW11: PRINT FAILURE RATE IN TESTLOOP-PROGRAM.
08                ;                SW12: NO HALT IN CASE OF ERROR IN TESTLOOP-PROGR.
09                ;                SW13: WAIT AFTER DIS MESSAGE.  DIS IS THE 16 CHAR
10                ;                SW14: HALT AFTER DIS MESSAGE.  DISPLAY AT OPERAT.
11                ;                SW15: CLEAR DIS AFTER MESSAGE. CONTROL PANEL. OCP
12                ;
13                ;            *   NOTICE: IF ANY OF SW 1 TO 9 ARE SET, THE CON-
14                ;            *   TROL BY SW 13 TO 15 IS DISABLED.
15                ;
16                ;
17                ;
18                ;        3.3 START
19                ;                OF PROGRAM AFTER LOADING:
20                ;                SET SWITCHES TO CONTROL. (3.2).
21                ;                ANSWER START ADDRESS QUESTION.
22                ;
23                ;
24                ;        3.3A RESTART
25                ;                THE BEST WAY TO INSURE CORRECT SWITCH-SETTINGS
26                ;                AFTER A RESTART:
27                ;
28                ;                RESET
29                ;                SET SWITCHES TO START ADDR. (3.1)
30                ;                EXAMINE
31                ;                SET SWITCHES TO CONTROL. (3.2)
32                ;                CONTINUE.
33                ;                THE START ADDR IS LISTED AT TTO, LPT AND DIS.
34                ;
35                ;
36                ;        3.3B RESTART OF PROG. IF RC3603 CPU
37                ;                SET DATA SWITCHS 0,10-15 UP (1)
38                ;                SET "RESET PARITY ERROR" DOWN (ON)
39                ;                PRESS AUTOLOAD AND RELEASE
40                ;                SET "RESET PARITY ERROR" UP (OFF)
41                ;                SET SWITCHES TO CONTROL (3.2)
42                ;                ANSWER THE QUESTION WRITTEN ON TTY/OCP
43                ;
44                ;
45                ;        3.3C POWER RESTART
46                ;                OF PROGRAM (ONLY IF POWER MONITOR OPTION):
47                ;
48                ;                SET KEY IN LOCK BEFORE REMOVING POWER
49                ;                AFTER POWERING UP, THE PROGRAM WRITES:
50                ;
51                ;                POWER
52                ;                ACTUAL PROGRAM NAME
53                ;                SET SWITCHES TO CONTROL. (3.2).
54                ;                ANSWER START ADDRESS QUESTION
```

```
I 0005  .MAIN
01
02                ;       3.4 LOADING MESSAGE:
03                ;               AFTER LOADING THE PROGRAM WILL RUN THROUGH SOME
04                ;               INITIALIZING CODE, ENDING UP WITH ANNOUNCING THE
05                ;               PROGRAM-NAME. THE MEM.SIZE, CPU TYPE ETC., WHICH
06                ;               ARE CALCULATED IN THIS CODE, MAY BE KNOWN BY
07                ;               STARTING THE TESTPROGRAM IN LOC. 1400, AND THE
08                ;               OUTPUT-FORMAT WILL BE LIKE THIS:
09                ;
10                ;               SWITCHES: 100073 ;THIS IS THE INITIAL STATE OF SW
11                ;               CPU TYPE: 000016 ;THIS IDENTIFIES WHICH CPU/MEM
12                ;               LAST LOC: 077777 ;THIS INFORMS THE USER: MEM-SIZE
13                ;               BINARY LOADER OK ;THIS IS THE TS VERSION PLACED
14                ;                                ;AT LAST LOC (MAX 32K) BUT
15                ;                                ;SLIGHTLY MODIFIED AND INCL.
16                ;                                ;BOOTSTRAP. THE LOADER HAS AN
17                ;                                ;ERRORHALT AT XX7752 AND READY-
18                ;                                ;HALT AT XX7676 IF NOT SELFSTRT.
19                ;               ACTUAL PROG NAME ;THIS IS THE LOADED PROGRAM.
20                ;               SET SWITCHES TO CONTROL.(3.2), STARTADDR 400 ?
21                ;
22                ;       3.5 START WITH BREAK OPTION (RC 3603).
23                ;               RESET
24                ;               SET SWITCHES TO HALT INSTR      063077
25                ;               SET REGISTER SELECT TO 6
26                ;               DEPOSIT INTO REGISTER
27                ;               SET SWITCHES TO ADDRESS         000003
28                ;               SET REGISTER SELECT TO 5
29                ;               DEPOSIT INTO REGISTER
30                ;               SET BREAK SWITCH TO ON
31                ;               GO TO 3.3 RESTART WITH SA = 1400
32                ;
33                ;       3.6 CPUNO
34                ;               FOR TIMING PURPOSE THE PROGRAM DETERMINES IN
35                ;               WHICH CPU IT RESIDUES. IF IT FAILS THE PROGRAM
36                ;               WILL TRY TO CONTINUE AFTER THE MESSAGE
37                ;               "MISERABLE TIMING". IF IMPOSSIBLE THE PROGRAM
38                ;               WILL ASK YOU TO IDENTIFY THE CPU WITH A NUMBER
39                ;               BETWEEN 0 AND 6.
40                ;                                       USE 2 FOR NOVA1200, RC3603-BREAK
41                ;                                           4 FOR RC3603
42                ;                                           5 FOR NOVA 2-16K
43                ;                                           6 FOR NOVA 2-8K WHERE
44                ;               THE MEMORY TYPE FOR THE FIRST 8K IS RELEVANT.
45                ;
46                ;               GIVE CPU-NO. TO PROGRAM IN THIS WAY:
47                ;
48                ;               RESET
49                ;               DEPOSIT # INTO AC2
50                ;               SET SWITCHES TO CONTROL (3.2).
51                ;               CONTINUE.
52                ;
53                ;               ...OR CORRECT PROGRAM,READ MESS IN END OF "LOCKA"
```

```
1 0006  .MAIN
01.
02                ;4. OPERATING PROCEDURE
03                ;
04                ;       LOAD THE PROGRAM USING THE SELFSTARTING BINARY LOADER
05                ;       IN FRONT OF THE PAPERTAPE (ONLY IF RC3600, IN CASE OF A
06                ;       RC7200 YOU HAVE TO LOAD ANOTHER BIN.LOADER AND PASS-BY
07                ;       THE FIRST SEGM. ON TAPE) OR, WHEN IT RESIDES ON DISC
08                ;       BY CALLING "FDCTEST" THROUGH HIPBOOT. AFTER LOADING
09                ;       SOME MESSAGES (SEC. 3.4) APPEARS, AND THE BINARY
10                ;       LOADER TS WILL BE CORRECTLY PLACED IN CORE. DURING THE
11                ;       INITIALIZING THE PROGRAM WILL FIND OUT IN WHICH CPU/MEM
12                ;       IT RESIDES (SEC 3.6).
13                ;
14                ;       IN CASE OF INPUT'ING ANSWERS TO THE QUESTIONS BEYOND
15                ;       THE LIMITS, THE QUESTIONS ARE REPEATED.IF THE SUGGESTED
16                ;       ANSWER IS OK, ANSWER ONLY NL (RETURN). THE LAST DIGIT
17                ;       ANSWERED MAY BE REMOVED WITH KEY "RUBOUT", ELSE,
18                ;       IF YOU TYPED WRONG, THE QUESTION IS REPEATED BY GIVING
19                ;       DIGITS UNTIL LIMIT IS PASSED.
20                ;
21                ;       THE INITIALIZING OF THE PROGRAM WILL END UP WITH A QUES-
22                ;       TION ABOUT THE STARTADDRESS, AND THE WANTED NUMBER OF
23                ;       RUNS. ANSWER THE QUESTIONS, AND WAIT (IF SA=400) FOR THE
24                ;       "DRIVE-LOAD"-INSTRUCTION ON TTY. THEN LOAD THE DRIVE,
25                ;       WITH THE DEMANDED DISCETTE, TYPE NL, AND THE PRO-
26                ;       GRAM WILL RUN WITHOUT FURTHER OPERATOR INVOLVEMENT.
27                ;
28                ;       THE TEST ST. 400 WILL START WITH A FEW LOOPS WHERE THE
29                ;       OFF-LINE STATUS ETC. ARE CHECKED. THEREFORE DON'T LOAD
30                ;       THE DRIVE WITH ANY DISCETTE, BEFORE THE PROGRAM DEMANDS
31                ;       IT.
32                ;
33                ;
34                ;       THE PROGRAM CONTAINS AN ASCII PRINTER TABLE. IF THE
35                ;       PRINTER HAS ANOTHER DRUM ALPHABETH READ FOLLOWING:
36                ;
37                ;
38                ;
39                ;
40                ;4.1 PRINTER ALPHABETH CHANGE:
41                ;
42                ;AT ANY TIME AFTER LOADING THIS PROGRAM IT IS POSSIBLE TO CHANGE
43                ;THE ALPHABETH USED ON THE LINEPRINTER. THERE ARE 2 METHODS:
44                ;
45                ;
46                ;4.1.1                  GET ONE OF THE BUILD-IN ALPHABETHS:
47                ;
48                ;       START PROGRAM IN 2202
49                ;       REMEMBER SWITCHES TO CONTROL, (3.2).
50                ;       ANSWER ALPHABET # (SEE EXISTING BELOW),
51                ;       THE PROGRAM WILL RESTART AFTER CHANGING THE ALPHABETH.
52                ;       ANSWER NEXT START ADDRESS.
```

```
| 0007  .MAIN
01
02              ;4.1.2                          GET -ADD ON- A NEW TAPE .
03              ;
04              ;LOAD THE NEW ALPHABETH FROM TAPE INTO CORE UPON THE PROG. BY
05              ;USE OF THE BINARY LOADER. THEN RESTART THE PROGRAM. THE CURRENT
06              ;TAPES WITH PRINTER TABLE ALPHABETH ARE DESCRIBED BELOW:
07              ;
08              ;EXISTING: #1   44-RT 535       ASCII
09              ;          #2   44-RT 529       RC STANDARD TYPE 71/78 STARTING .
10              ;          #3   44-RT 532       RC STANDARD TYPE 71/78 STARTING 0
11              ;                                       (SKEWED 4 POSITIONS)
12              ;          #4   44-RT 1213      PL 1, TYPE 70
13              ;
14              ;HOW TO PRODUCE A NEW TABLE:
15              ;
16              ;THE TABLE HAS 200 OKTAL (128 DECIMAL) BOXES. INPUT KEY
17              ;IS THE ASCII VALUE OF THE CHARACTER TO PRINT ADDED TO 2000.
18              ;THE RESULT IS ADDRESS OF A BOX. EACH BOX OCCUPIES A CORE WORD.
19              ;IT IS BUILT UP OF TWO FIGURES. THE FIRST IS THE CLASS OF THE
20              ;CHARACTER TO BE PRINTED: 0 FOR PRINT, 6 FOR BLIND. THE SECOND
21              ;IS THE CHARACTER VALUE AT THE PRESENT PRINT DRUM. BELOW
22              ;VALUE 40 OCTAL FOLLOWING CHARACTERS MAY BE USED: 11 TAB,
23              ;12 LF, 14 FF AND 15 CR. ALL OTHERS BELOW 40 WILL GIVE SPACE.
24              ;THE FIRST BOX SHOULD CONTAIN THE VALUE FOR THE NULL CHAR
25              ;AND THE LAST THE VALUE FOR THE DEL CHAR, WHICH BOTH NORMALLY
26              ;ARE BLIND. IF YOU COUNT 0,1,2, ,7,10, , THE BOX 101 SHALL
27              ;CONTAIN THE PRINT DRUM VALUE FOR AN A. IF THE DRUM DO NOT
28              ;HAVE SMALL LETTERS, FILL IN THE VALUE FOR BIG ONES. NOW PUNCH
29              ;AN ASCII TAPE LIKE THIS:
30              ;
31              ;   .LOC 2000
32              ;   .RDX 8          ;WHICH RDX YOU WANT
33              ;   .TXTE?
34              ;   <6><0>          ;(2000) FIRST BOX, BOX 0
35              ;   <6><0>
36              ;   .
37              ;   .
38              ;   .
39              ;   <0><101>        ;(2101) BOX 101 FOR A. FOR ASCII DRUM
40              ;   .               ;101 IS USED, FOR TYPE 71 137 IS USED.
41              ;   .
42              ;   .
43              ;   <6><0>?         ;(2177) BOX 177, LAST
44              ;   .RDX 8
45              ;   .END 101
46              ;
47              ;PRODUCE A BINARY TAPE AND LOAD THIS TO MEMORY WITH
48              ;BINARY LOADER AFTER LOADING OF MAIN PROGRAM.
49              ;
50              ;
51              ;
52              ;
53              ;5. PROGRAM DESCRIPTION
54              ;
55              ;       THIS SECTION CONTAINS:      1. TESTLOOP FAILURE RATE
56              ;                                  2. STRUCTURE OF PROGRAM
57              ;
58              ;
59              ;       5.1 TESTLOOP FAILURE RATE.
60              ;
61              ;THERE ARE TWO DIFFERENT WAYS TO USE THE ROUTINES FOR TESTLOOP:
62              ;SINGLE OR MULTIPLE ERRORHALT:
```

```
! 0008  .MAIN
01
02                ;SINGLE:          SETP1          MULTIPLE:    SETP1
03                ;                 ERRORHALT                   ERRORHALT
04                ;                 LOOP                        ERRORHALT
05                ;                                             ERRORHALT
06                ;                                             LOOP
07                ;
08                ;IN CASE OF A CONSTANT ERROR THE RATE WILL BE PRINTED
09                ;THIS WAY:
10                ;
11                ;PC XXXXXX 100 %                                PC XXXXXX 300 %
12                ;WERE THE LAST IS A MULTIPLE OF 100 %.
13                ;
14                ;THERE ARE THREE POSSIBILITIES FOR THE NUMBER OF LOOPS IN A
15                ;CYCLE, I. E. HOW MANY TIMES THE PROGRAM RUNS THROUGH THE
16                ;INSTRUCTIONS BETWEEN SETP1 AND LOOP. IF SWITCH 0 IS 0 FIRST
17                ;TIME AN ERROR IS DETECTED THE PROGRAM PROCEEDS TO NEXT
18                ;INSTRUCTION AFTER LOOP.
19                ;
20                ;NUMBER OF LOOPS:          SETP0:  1
21                ;                         SETP1:  10
22                ;                         SETP2:  100
23                ;
24                ;IF THE ERROR IS NOT CONSTANTLY IT IS POSSIBLE TO SEE THESE
25                ;FAILURE RATES IF SINGLE OPERATION:
26                ;
27                ;SETP0: 100 % ERROR IN THE ONE LOOP.
28                ;SETP1: 100,50,33,25,20,16,14,12,11,10 % FOR ERROR IN
29                ;         1, 2, 3, 4, 5, 6, 7, 8, 9, 10 -TH LOOP.
30                ;SETP2: LIKE SETP1, ONLY ADD 9...1 % FOR ERROR IN
31                ;        11...100 -TH LOOP.
32                ;
33                ;IF MULTIPLE OPERATION THE RATES DEPENDS ON HOW MANY OF THE
34                ;ERRORS ARE FOUND IN THE FIRST LOOP WITH ERROR:
35                ;       100 % COULD BE  1 ERROR IN FIRST LOOP
36                ;                 OR    2 ERRORS IN SECOND LOOP
37                ;                 OR    3 ERRORS IN THIRD LOOP ETC.
38                ;
39                ;IF SWITCH 0 IS 1 THE PROGRAM WILL REMAIN IN THE LOOP WITH
40                ;ERROR. FIRST TIME AN ERROR IS SEEN THE PROGRAM WILL HALT
41                ;(IF NOT SWITCH 12). THEN THE PROGRAM CONTINUES UNTILL ALL
42                ;NUMBER OF LOOPS ARE PERFORMED. THEN A NEW CYCLE IS
43                ;ENTERED CALLED ERRORCYCLE. AFTER EACH ERRORCYCLE A NEW IS
44                ;STARTED UNTILL SWITCH 0 IS SET TO 0.
45                ;
46                ;IN ALL CYCLES (FIRST OR ERROR) THE PC (PROGRAM COUNTER OF ERROR)
47                ;IS WRITTEN FIRST TIME AN ERROR IS SEEN AND IN MULTIPLE OPERATION
48                ;MORE THAN ONE ERRORHALT COULD WRITE THE PC. BUT ONLY WITHIN
49                ;THAT FIRST LOOP WITH ERROR. THE FAILURE RATE IS PRINTED WHEN
50                ;THE CYCLE IS FINISHED. THE PROGRAM ONLY HALTS IN THE FIRST CYCLE
51                ;(DEPENDING ON SW 12). THE ERRORCYCLES HAVE ANOTHER AMOUNT OF
52                ;LOOPS THAN FIRST CYCLE:
53                ;
54                ;NUMBER OF LOOPS:          FIRST CYCLE     ERROR CYCLE
55                ;                  SETP0:  1               10
56                ;                  SETP1:  10              100
57                ;                  SETP2:  100             100
58                ;
59                ;IN SINGLE OPERATION FAILURE RATE IS TELLING HOW OFTEN THE ERROR
60                ;APPEARS. IN MULTIPLE OPERATION FAILURE RATE SHOULD BE USED
61                ;VERY CAREFULLY.
```

```
I 0009  .MAIN
01
02                ;
03                ;
04                ;
05                ;
06                ;
07                ;
08                ;
09                ;
10                ;
11                ;
12                ;
13                ;
14                ;
15                ;
16                ;
17                ;
18                ;
19                ;
20                ;
21                ;
22                ;
23                ;
24                ;
25                ;
26                ;
27                ;
28                ;
29                ;
30                ;
31                ;
32                ;
33                ;
34                ;
35                ;
36                ;
37                ;
38                ;
39                ;
40                ;
41                ;
42                ;
43                ;
44                ;
45                ;
46                ;
47                ;
48                ;
49                ;
50                ;
51                ;
52                ;
53                ;
54                ;
55                ;
56                ;
57                ;
58                ;
59                ;
60                ;
61                ;
62                ;
63                ;
```

5.2 STRUCTURE OF PROGRAM.

THE PROGRAM IS A NORMAL DIAGNOSTIC PROGRAM, I. E. IN CASE OF ERROR THE PROGRAM WILL HALT WITH AC3 = PC. THE OPERATOR SHOULD THEN LOOK UP IN THE LISTNING TO KNOW WHAT WAS CAUSING THE ERRORHALT. THE PROGRAM INCLUDES ONLY FEW SUBROUTINES TO MAKE IT EASYER FOR THE OPERATOR.
THE PROGRAM ALSO CONTAINS NORMAL TESTLOOP FACILITIES, SEE 5.1. IN CASE OF SEVERAL ERRORHALTS ATTENTION SHOULD ONLY BE GIVEN TO THE FIRST (THE LOWEST PC) BECAUSE THE REST COULD BE DERIVED FROM THE FIRST. TIMING IN A LOOP IS AFFECTED BY PRINTING FIRST ERROR,TOO.

THE PROGRAM WILL TEST THE FDC WITH AID OF FOLLOWING TEST LOOPS:

A0000	CHECK SELB BUS LINE
A0010	CHECK SELD BUS LINE
A0020	CHECK DIA BUS LINE
A0030	CHECK DATALINES
A0040	CHECK DIB BUS LINE
A0050	CHECK DIC BUS LINE
A0060	CHECK NIOP-RESET OF FDC-DONE
A0070	CHECK CLEAR OF FDC-BUSY
A0080	CHECK I/O-RESET OF FDC-BUSY
A0090	CHECK CLEAR OF FDC-DONE
A0100	CHECK I/O-RESET OF FDC-DONE
A0110	CHECK OF CPU-SKIP
A0120	CHECK OF CPU-SKIP
A0130	CHECK OF CPU-INTERRUPT ON
A0140	CHECK RESET OF CPU-INTERRUPT ON
A0150	CHECK OF INTERRUPT REQUEST LINE
A0160	CHECK THAT STATUS IS OK
A0170	CHECK NIOP-RESET OF DATA BUFFER
A0180	CHECK DATA PATH TO/FROM BUFFER
A0190	CHECK DEVICE SELECT LOGIC
A0200	CHECK LOCAL-STATUS
A0210	CHECK WRITE-PRO-STATUS
A0220	CHECK SETTING OF BUSY AND DONE BY STARTPULSE
A0230	CHECK SETTING OF DONE AFTER RECALIBRATION
A0240	CHECK CLEAR OF FDC-DONE
A0250	CHECK NIOP-RESET OF FDC-DONE
A0260	CHECK INT. REQUEST,INTA-INSTR,CLEAR OF INT.REG.
A0270	CHECK INTERRUPT DISABLE FF
A0280	CHECK STATUS
A0290	CHECK STATUS
B0290	CHECK CLEAR OF TRACK-COUNTER
A0300	CHECK RECALIBRATION-COMMAND
A0310	CHECK TIME-OUT STATUS
A0320	CHECK TRACK-COUNTER,AUTO-RESET OF STATUS-REG.
A0330	CHECK OF DATA-BUFFER
A0340	CHECK OF READ-FUNCTION, I.E. READ OF TEST-MASTE DISCETTE WITHOUT DATACHECK
A0350	CHECK STATUS
A0360	READ OF 100 SECTORS IN TRACK 0 WITHOUT DATACHEC

```
| 0010  .MAIN
01
02                ;        A0370   READ 100 SEC. IN DIFF. TRACKS WITHOUT DATACHECK
03                ;
04                ;        A0380   CHECK STATUS
05                ;        A0390   READ 100 SECTORS IN DIFF. TRACKS WITH DATACHECK
06                ;
07                ;        A0400   CHECK STATUS
08                ;        A0410   READ SEC. 10 & 21 IN ALL TRACKS WITH DATACHECK
09                ;
10                ;        A0420   CHECK STATUS
11                ;        A0430   CHECK STATUS
12                ;        A0440   CHECK WRITE LOGIC, I.E. WRITE TRACK 0 WITHOUT
13                ;                DATACHECK
14                ;
15                ;        A0450   CHECK STATUS
16                ;        A0460   WRITE DIFF. TRACKS WITHOUT DATACHECK
17                ;
18                ;        A0470   CHECK STATUS
19                ;        A0480   WRITE DIFF. SECTORS IN TRACK 21&42 WITH DATACHECK
20                ;
21                ;        A0490   WRITE 100 RANDOM SECTORS IN RANDOM TRACKS WITH
22                ;                DATACHECK
23                ;
24                ;        A0500   WRITE ZERO'S IN ALL SECTORS OF THE DISCETTE
25                ;
26                ;        A0510   WRITE/READ LOOP. USED BY STARTADDRESS 405,406.
27                ;
28                ;
29                ;
30                ;
31                ;6. STRAPPINGS ETC.
32                ;
33                ;        THIS TEST DOESN'T DEMAND ANY SPECIAL STRAPPING.
34                ;
35                ;
36                ;7. STATUS BIT TABLE.
37                ;
38                ;
39                ;        BIT     MNEMONIC
40                ;
41                ;         0      HARDWARE ERROR
42                ;
43                ;         1      LOCAL
44                ;         2      0
45                ;         3      ADDRESS FIELD ERROR
46                ;
47                ;         4      0
48                ;         5      WRITE PROTECTION
49                ;         6      0
50                ;
51                ;         7      0
52                ;         8      0
53                ;         9      0
54                ;
55                ;        10      PARITY ERROR
56                ;        11      0
57                ;        12      POSITION ERROR
58                ;
59                ;        13      0
60                ;        14      TIME OUT
61                ;        15      0
```

```
| 0011   .MAIN
01
02                ;8. SOFTWARE SURVEY
03                ;
04                ;
05                ;  DOA   AC,FDC   ;LOAD DEVICE COMMAND REGISTER
06                ;                 ;AC(B6,B7)= COMMAND
07                ;                 ;AC(B9-B15)= PARAMETER
08                ;
09                ;                 COMMAND  MNENONIC                 PARAMETER
10                ;
11                ;                   0      READ ONE SECTOR          SECTOR NO.
12                ;                   1      WRITE ONE SECTOR         SECTOR NO.
13                ;                   2      RECALIBRATION               0
14                ;                   3      TRACK SEARCH             TRACK NO.
15                ;
16                ;  DOB   AC,FDC   ;LOAD DEVICE DATA-BUFFER WITH CHARACTER
17                ;                 ;AC(B8-B15)= CHARACTER
18                ;
19                ;  DOC   AC,FDC   ;SET THE DATA FIELD SYNC. OUTPUT REGISTER
20                ;                 ;AC(B8-B15)= DATA FIELD CHAR
21                ;
22                ;  DIA   AC,FDC   ;LOAD STATUS REG. TO AC
23                ;
24                ;  DIB   AC,FDC   ;LOAD CHARACTER FROM DEVICE DATA-BUFFER TO AC
25                ;
26                ;  DIC   AC,FDC   ;LOAD TRACKCOUNTER/DATA FIELD CHAR TO AC
27                ;                 ;AC(B0-B7)= TRACK COUNTER
28                ;                 ;AC(B8-B15)= DATA FIELD SYNC INPUT REG.
29                ;
30                ;  STARTPULSE     ;EXECUTE COMMAND IN COMMAND-REGISTER
31                ;
32                ;  CLEARPULSE     ;CLEARS BUSY AND DONE
33                ;
34                ;  P-PULSE        ;CLEARS DEVICE DATA BUFFER
35                ;
36                ;
37                ;
38                ;9. BELONGING TESTEQUIPMENT.
39                ;
40                ;       A SPECIAL PRE-WRITTEN DISCETTE (CALLED TEST DISCETTE
41                ;       OR TM-DISCETTE) HAS TO BE USED IN THE TESTS WHICH
42                ;       CHECKS THE READ-LOGIC. YOU MAY GENERATE THIS DISCETTE
43                ;       YOURSELF BY STARTING THE PROGRAM IN LOC. 410, OR YOU MAY
44                ;       ORDER IT FROM RC.
45                ;       NO SPECIAL TESTPLUGS OR CABLES HAS TO BE USED DURING THE
46                ;       TEST.
47                ;
48                ;
49                ;10. MESSAGES FROM THIS TEST
50                ;
51                ;       ALL MESSAGES ARE WRITTEN ON TTY, LPT AND THE 16 CHAR DIS
52                ;       ON OPERATORS CONTROL PANEL. IT IS POSSIBLE TO ANSWER
53                ;       QUESTIONS AT TTY OR NUK, NUMERIC KEYBOARD ON OPERATORS
54                ;       CONTROL PANEL.
55                ;       ABOUT THE NUMBERS:
56                ;               0-5 DIGITS IS A DECIMAL NUMBER
57                ;                   RANGE -32768 TO -1 AND 0 TO 32767
58                ;                 6 DIGITS IS AN OCTAL NUMBER
59                ;               0-6 DIGITS IS AN OCTAL NUMBER WITH
60                ;                   LEADING ZEROES SUPPRESSED, DON'T USE.
61                ;                 8 DIGITS IS A BINARY NUMBER.
```

```
! 0012  .MAIN
01
02                ;       MESSAGES:       LPT/TTY & DIS, IF SPEC DIS MESSAGE: ()
03                ;
04                ;       MISERABLE TIMING, RUN RC 3600 INSTRUCTION TIMER TEST
05                ;       (MISERABLE TIMING)
06                ;                               TIMING WITH BIG TOLERANCES, CPU
07                ;                               TYPE  IS GUESSED.
08                ;       RTC IS UNSTABLE,        RTC NOT RUNNING WITH CONSTANT
09                ;                               SPEED.
10                ;       SET CPUNO > AC2         HELP THE PROGRAM TO IDENTIFY CPU,
11                ;                               SEE 3.5
12                ;       SWITCHES: 100073        NO MESS AT DIS ABOUT SW. POSITION
13                ;       CPU TYPE: 000016
14                ;       LAST LOC. 077777
15                ;       BINARY LOADER OK
16                ;       RC3600 FDC-TEST
17                ;       SET SWITCHES TO CONTROL, (3.2), STARTADDR   400 ?
18                ;
19                ;       SWITCHES: 100030        NO MESS AT DIS ABOUT SW. POSITION
20                ;       000401   STARTADDR
21                ;
22                ;       NO. OF PASSES:   1 ?    ;ANSWER THE WANTED NO. OF PASSES,
23                ;       (# OF PASS   1 ?)       ;OR USE THE SUGGESTED ONE BY TYP-
24                ;                               ;ING A SINGLE NL.
25                ;
26                ;       DATACHECK (Y/N):  Y ?   ;IF YOU WANT DATACHECK AFTER SEC-
27                ;                               ;TORREAD THEN ANSWER NL, ELSE "N"
28                ;                               ;FOR "NO".
29                ;
30                ;       ERRORSKIP (Y/N):  N ?   ;IF ERRORS HAS TO BE ANNOUNCED IN
31                ;                               ;USUAL WAY,THEN ANSWER NL,IF THEY
32                ;                               ;HAS TO BE NEGLECTED (WHICH MAY BE
33                ;                               ;USEFUL IN CASE OF SCOP'ING READ-
34                ;                               ;SIGNALS, ANSWER "Y" FOR "YES".
35                ;
36                ;       INSERT TESTDISC.,"WRITE PRO" ON
37                ;
38                ;       INSERT TESTDISC.,"WRITE PRO" OFF
39                ;
40                ;       INSERT SCRATCHDISC.,"WRITE PRO" OFF
41                ;
42                ;                               ;INSERT THE DEMANDED DISCETTE,
43                ;                               ;AND PRESS WRITE-PRO SWITCH IF
44                ;                               ;REQUIRED.TYPE "ESCAPE" OR "NL"
45                ;                               ;TO CONTINUE.
46                ;
47                ;       RE-READ %= XXXXXX IN LOOP AYYYY
48                ;
49                ;                               ;IN SOME OF THE TEST-LOOPS, A
50                ;                               ;NUMBER OF RE-READS FROM THE DIS-
51                ;                               ;CETTE ARE ALLOWED,IF THE COMMAND
52                ;                               ;WASN'T SUCCESSFULL AT THE FIRST
53                ;                               ;TIME.THE ANNOUNCED PERCENT INDI-
54                ;                               ;CATES THE AMOUNT OF RE-READS
55                ;                               ;WHICH WERE NECESSARRY PER 100
56                ;                               ;SEC.-READS.ONLY TYPED IF % > 20.
57                ;
58                ;       DEV.CODE=               ;INPUT TO THE DEVICE-CODE CHANGE
59                ;                               ;ROUTINE. TYPE THE NEW NUMBER IN
60                ;                               ;OCTAL.THE PROG. WILL ASK FOR
61                ;                               ;NEW STARTADDRESS AFTER CHANGING.
```

```
! 0013  .MAIN
01
02                ;       037500  037503  000000  ERROR MESSAGE
03                ;       AC0     AC1     AC2
04                ;       PC 007464          100 %
05                ;
06                ;       (PC 007464   100 %)     EXAMINE FOR AC'S AFTER HALT.
07                ;
08                ;       SWITCHES: 100070        NO MESS AT DIS ABOUT SW. POSITION
09                ;                               PRINT INHIBIT SWITCH SET.
```

```
I 0014  .MAIN
01
02
03
04
05
06
07
08
09
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40                ;TAPE 2          PAGE ZERO FOR TAPE 3,4,5
41
42        000000 .LOC 0
43
44  00000 003160          2*MELOC          ;MESS AFTER RDOS LOAD AND STORE PC ON INTR
45  00001 000000          0                ;ADDR OF INTR. SERVICE ROUTINE
46  00002 001400          REBIN            ;SELFSTART ADDR FOR RDOS ETC.
47  00003 000000          0                ;0=HALT, 1=SELFSTART PROG AFTER REBIN
48  00004 000000          0                ;#ADDR FOR SELFSTART PROG AFTER REBIN
49  00005 000000          0                ;FOR LOAD RDOS, USED BY POW. INTR, FITYP
50
51        000020 .LOC 20
52
53  00020 000000 IDX0:    0                ;AUTO INCREMENT LOCATION
54  00021 000000 IDX1:    0                ;AUTO INCREMENT LOCATION
55  00022 000000 IDX2:    0                ;AUTO INCREMENT LOCATION
56  00023 000000 IDX3:    0                ;AUTO INCREMENT LOCATION
```

```
1 0015   .MAIN
01
02          000040 .LOC 40                    ;INDIRECT ADDRESSES
03                                            ;NOT IN AUTO INC,DEC LOC.
04 00040 000412 IMESS:  XMESS
05 00041 000664 ICHAR:  XCHAR
06 00042 000724 ITYPE:  XTYPE
07 00043 001043 ICRLF:  XCRLF
08 00044 001104 IDISP:  XDISP
09 00045 001134 IDOUT:  XDOUT
10 00046 001165 IDICL:  XDICL
11 00047 001176 IDATT:  DISATT
12 00050 001221 IHAAT:  HAATT
13 00051 001014 ITBIN:  XTBIN
14 00052 000560 ITOCT:  XTOCT
15 00053 000470 ITDEC:  XTDEC
16 00054 000550 ITZOC:  XTZOC
17 00055 001020 IDBIN:  XDBIN
18 00056 000564 IDOCT:  XDOCT
19 00057 000464 IDDEC:  XDDEC
20 00060 000554 IDZOC:  XDZOC
21 00061 002427 IWAIT:  XWAIT
22 00062 001244 IWAOP:  XWTOP
23 00063 002477 ITISK:  RTIME
24 00064 002553 ITIMS:  MSTIM
25 00065 002702 ITIRQ:  XTIMS
26 00066 002653 IMULT:  XMULT
27 00067 002665 IDIVS:  XDIVS
28 00070 002666 IDIVD:  XDIVD
29 00071 005211 IQUES:  XQUES
30 00072 001736 ISAMS:  XSAMS
31 00073 001312 IRESW:  XRESW
32 00074 000000 HMEND:  0                     ;TOP OF MEMORY, LOADER PROTECT
33 00075 000000 DIGIN:  0                     ;INPUT BUFFER FOR INPUT ROUTINES.
34
35              ;PRINTER TABLE HANDLING AND (POWER) RESTART:
36
37          000076 .LOC 76
38
39 00076 002100 POWZE:  JMP     @POWRE        ;INSTRUCTION TO BE STORED IN CELL ZERO
40 00077 004740 IRESA:  SWISA                 ;PROGRAM RESTART ADDR.
41 00100 004662 POWRE:  POWCN                 ;POWER RESTART ADDR
42 00101 063077 PRINT:  HALT                  ;IMPORTANT TO KEEP THIS AND NEXT IN
43 00102 002077 STOP:   JMP     @IRESA        ;101,102 BECAUSE PRINTER TABLE SELFSTART
44
45 00103 006054 IGTBI:  GETBI
46 00104 005733 IGTOK:  GETOK
47 00105 005430 IGTOC:  GETOC
48 00106 005645 IGTSC:  GETSC
49 00107 006163 IGTTX:  GETTX
50 00110 006420 ISTP0:  ENTP0
51 00111 006425 ISTP1:  ENTP1
52 00112 006432 ISTP2:  ENTP2
53 00113 006451 ILOOP:  CYCLE
54 00114 006621 IHALT:  ERROR
55 00115 006722 ISTAA:  XSTAA
56 00116 006736 ISTAN:  XSTAN
57 00117 006746 ISTAW:  XSTAW
58 00120 007000 ISTAS:  XSTAS
59 00121 006763 ISTAP:  XSTAP
60 00122 007010 ILORE:  XLORE
61 00123 007026 IPASS:  XPASS
```

```
| 0016  .MAIN
01
02                ;  DEFINITIONS
03
04                ;TTI=10
05                ;TTO=11
06                ;RTC=14
07                ;LPT=17
08          000032 FUN=32
09          000033 FUB=33
10          000034 NUK=34
11          000035 DIS=35
12          000017 XLPT=LPT
13          000010 XTTI=TTI
14          000011 XTTO=TTO
15          000014 XRTC=RTC
16
17          006040 CMESS=JSR        @IMESS  ;EACH DEFINITION BELOW CORRESPONDS
18          006041 CCHAR=JSR        @ICHAR  ;WITH A CALL OF A ROUTINE.
19          006042 CTYPE=JSR        @ITYPE
20          006043 CCRLF=JSR        @ICRLF
21          006044 CDISP=JSR        @IDISP
22          006045 CDOUT=JSR        @IDOUT
23          006046 CDICL=JSR        @IDICL
24          006047 CDATT=JSR        @IDATT
25          006050 CHAAT=JSR        @IHAAT
26          006051 CTBIN=JSR        @ITBIN
27          006052 CTOCT=JSR        @ITOCT
28          006053 CTDEC=JSR        @ITDEC
29          006054 CTZOC=JSR        @ITZOC
30          006055 CDBIN=JSR        @IDBIN
31          006056 CDOCT=JSR        @IDOCT
32          006057 CDDEC=JSR        @IDDEC
33          006060 CDZOC=JSR        @IDZOC
34          006061 CWAIT=JSR        @IWAIT
35          006062 WATOP=JSR        @IWAOP
36          006063 TIMSK=JSR        @ITISK
37          006064 TIMMS=JSR        @ITIMS
38          006065 TIMRO=JSR        @ITIRO
39          006066 MULTI=JSR        @IMULT
40          006067 DIVIS=JSR        @IDIVS
41          006070 DIVID=JSR        @IDIVD
42          006071 CQUES=JSR        @IQUES
43          006072 CSAMS=JSR        @ISAMS
44          006073 CRESW=JSR        @IRESW
45          006103 CGTBI=JSR        @IGTBI
46          006104 CGTOK=JSR        @IGTOK
47          006105 CGTDC=JSR        @IGTDC
48          006106 CGTSC=JSR        @IGTSC
49          006107 CGTTX=JSR        @IGTTX
50          006110 SETP0=JSR        @ISTP0
51          006111 SETP1=JSR        @ISTP1
52          006112 SETP2=JSR        @ISTP2
53          006113 LOOP=JSR         @ILOOP
54          006114 EHALT=JSR        @IHALT
55          006115 STATA=JSR        @ISTAA
56          006116 STATN=JSR        @ISTAN
57          006117 STATW=JSR        @ISTAW
58          006120 STATS=JSR        @ISTAS
59          006121 STATP=JSR        @ISTAP
60          006122 CLORE=JSR        @ILORE
61          006123 CPASS=JSR        @IPASS
```

```
I 0017  .MAIN
01
02                ; CONSTANTS AND PROCEDURE-CALLS USED BY OR AS EXTENSIONS FOR THE
03                ; TS-STANDARD-ROUTINES.
04
05        000061          FDD= 61
06        000061          XFDD= FDD
07        000061          DEV= XFDD
08
09
10 00124 000061 DEVICE: DEV                ;CURRENT DEVICE-CODE IN TEST.
11
12 00125 177777 INIMK:  177777             ;INIMK IS <> 0 AFTER PROG.LOAD, BUT IS
13                                         ;SET TO 0 AFTER FIRST CALL OF SW-READ-
14                                         ;ROUT.
15
16 00126 000010 SWREG:  000010             ;INITIAL SWITCH-SETTING, USED INSTEAD
17                                         ;OF THE ACTUAL SWITCHES UNTIL PROC.
18                                         ;"CSAMS" IS CALLED FIRST TIME.
19
20 00127 010113 CLLX1:  FDRST
21        006127          RESET=  JSR      @CLLX1
22 00130 010202 CLLX2:  RDSWI
23        006130          RDSWS=  JSR      @CLLX2
24 00131 010175 CLLX3:  ENBSW
25        006131          SWENB=  JSR      @CLLX3
26 00132 010144 CLLX4:  RINGB
27        006132          CBELL=  JSR      @CLLX4
28
29
30
31                        ;FDC 703 PROCEDURE CALL DEFINITIONS
32
33 00133 007510 CLL01:  STINT
34        006133          SETIN=  JSR      @CLL01
35 00134 007776 CLL02:  GERAN
36        006134          GRAND=  JSR      @CLL02
37 00135 010026 CLL03:  RTRSE
38        006135          RANTS=  JSR      @CLL03
39 00136 007620 CLL04:  RCALI
40        006136          RECAL=  JSR      @CLL04
41 00137 007664 CLL05:  TRAPO
42        006137          TRPOS=  JSR      @CLL05
43 00140 010273 CLL06:  REASE
44        006140          RDSEC=  JSR      @CLL06
45 00141 007437 CLL07:  CHDEV
46        006141          SETDV=  JSR      @CLL07
47 00142 010241 CLL08:  PATFL
48        006142          PTFLL=  JSR      @CLL08
49 00143 010364 CLL09:  WRISC
50        006143          WRSEC=  JSR      @CLL09
51 00144 010105 CLL10:  GCUTR
52        006144          CURTR=  JSR      @CLL10
53 00145 010336 CLL11:  DOSET
54        006145          SETDO=  JSR      @CLL11
55 00146 010603 CLL12:  REAWR
56        006146          RDWRI=  JSR      @CLL12
57 00147 010502 CLL13:  SFTER
58        006147          SOFER=  JSR      @CLL13
59 00150 010646 CLL14:  TRCAL
60        006150          CALTR=  JSR      @CLL14
61 00151 010231 CLL15:  PATCM
62        006151          COMPT=  JSR      @CLL15
63 00152 007742 CLL16:  MESTR
64        006152          TRMES=  JSR      @CLL16
65 00153 010460 CLL17:  CLSEC
66        006153          CALSEC= JSR      @CLL17
```

```
( 0018  .MAIN
01
02                          ;SAVELOCATIONS FOR AC'S AFTER CALL OF PROCEDURES
03
04 00154 000000 FSAV0:  0               ;FIRST
05 00155 000000 FSAV1:  0
06 00156 000000 FSAV2:  0
07 00157 000000 FPRET:  0
08
09 00160 000000 SSAV0:  0               ;SECOND
10 00161 000000 SSAV1:  0
11 00162 000000 SSAV2:  0
12 00163 000000 SPRET:  0
13
14 00164 000000 TSAV0:  0               ;THIRD
15 00165 000000 TSAV1:  0
16 00166 000000 TSAV2:  0
17 00167 000000 TPRET:  0
18
19 00170 000000 RETU0:  0               ;GENERAL RETURN SAVE
20 00171 000000 GRRET:  0               ;LOOP-GROUP RETURN SAVE
21
22
23
24
25                          ;CONSTANTS
26
27 00172 000001 FD001:  1
28 00173 000002 FD002:  2
29 00174 000003 FD003:  3
30 00175 000004 FD004:  4
31 00176 000005 FD005:  5
32 00177 000007 FD007:  7
33 00200 000012 FD010:  10.
34 00201 000024 FD020:  20.
35 00202 000025 FD021:  21.
36 00203 000046 FD038:  38.
37 00204 000052 FD042:  42.
38 00205 000062 FD050:  50.
39 00206 000100 FD064:  64.
40 00207 000106 FD070:  70.
41 00210 000144 FD100:  100.
42 00211 000200 FD128:  128.
43 00212 000372 FD250:  250.
44 00213 001750 FD1K0:  1000.
45 00214 005670 FD3K0:  3000.
46 00215 177600 MD128:  -128.
47 00216 000010 FO010:  10
48 00217 000040 FO040:  40
49 00220 000077 FO077:  77
50 00221 000207 FO207:  207
51 00222 000377 FO377:  377
52 00223 000400 FO400:  400
53 00224 001000 FO1K0:  1000
54 00225 001400 FO1K4:  1400
55 00226 002000 FO2K0:  2000
56 00227 010000 FO10K:  10000
57 00230 010052 FOTK5:  10052
58 00231 060000 FO60K:  60000
59 00232 140000 FOH14:  140000
```

```
I 0019  .MAIN
01
02 00233 000373 DFIEL:  373             ;FB(HEX) = DATA FIELD SYNC BYTE
03 00234 007236 AWBUF:  WRBUF           ;ADDRESS OF WRITE BUFFER
04 00235 000032 MXSEC:  26.             ;MAX LEGAL SECTOR NO.
05 00236 000114 MXTRA:  76.             ;MAX LEGAL TRACK NO.
06 00237 000000 INTOC:  0               ;INTERRUPT OCCOURED FLAG
07 00240 000000 REWCO:  0               ;RE-WRITE COUNTER (MAX 3)
08 00241 000000 RERCO:  0               ;RE-READ COUNTER (MAX "TRIES")
09 00242 000000 TOTCO:  0               ;TOTAL NUMBER OF RE-READS
10 00243 000000 PASCO:  0               ;PASS COUNTER
11 00244 006722 ATBEG:  XSTAA           ;START OF DEVICE CHANGE
12 00245 014334 ATEND:  TSEND           ;END OF PROGRAM
13 00246 000000 RUNMK:  0               ;= 0 DURING TEST-MASTER GENERATING,
14                                      ;ELSE <> 0.
15 00247 000061 DEVC1:  61
16 00250 000064 DEVC2:  64
17 00251 000000 DTINH:  0               ;= 0 IF DATACHECK, ELSE <> 0
18 00252 000000 NEGER:  0               ;=0 IF ERRORANNOUNCING, ELSE <> 0
19
20
21
22
23 00253 000000 WORK1:  0               ;COMMON WORK-CELLS
24 00254 000000 WORK2:  0
25
26
27 00255 000000 TRIES:  0               ;MAX. NO. OF RETRIES IN CASE OF "SOFT"
28                                      ;STATUS ERROR,I.E. TIME OUT,ADDRESS
29                                      ;FIELD ERROR,PARITY ERROR,POSITION ER-
30                                      ;ROR, DATA WRONG, DATA SYNC ERROR.
31 00256 000000 WTRIES: 0               ;MAX NO. OF RE-WRITES.
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55       000366         .LOC  366
56
57 00366 011102 FST00:  FTES0
58 00367 011144 FST01:  FTES1
59 00370 011157 FST02:  FTES2
60 00371 011175 FST03:  FTES3
61 00372 011233 FST04:  FTES4
62 00373 011330 FST05:  FTES5
63 00374 011346 FST06:  FTES6
64 00375 011252 FST07:  CDEVI
65 00376 011274 FST10:  GENTM
```

```
I 0020  .MAIN
01
02          000400          .LOC 400
03
04 00400 006366             JSR     @FST00
05 00401 006367             JSR     @FST01
06 00402 006370             JSR     @FST02
07 00403 006371             JSR     @FST03
08 00404 006372             JSR     @FST04
09 00405 006373             JSR     @FST05
10 00406 006374             JSR     @FST06
11 00407 006375             JSR     @FST07
12 00410 006376             JSR     @FST10
13
14
15
16
17                  ; CHANGES MADE IN TS-STANDARD-ROUTINES:
18                  ; ***************************************
19                  ;
20
21
22                  ;CHANGES MADE TC MAKE IT POSSIBLE TO DISABLE AND ENABLE PANEL-
23                  ;SWITCHES VIA PROGRAM FLAGS.
24                  ;
25                  ;
26                  ; THE "READS 2" - INSTR. IN "XRESW" ROUTINE ARE REPLACED WITH
27                  ; THE "RDSWS"-PROCEDURE CALL.
28                  ;
29                  ; LOCATIONS ARE: 1317 AND 1326
30                  ;
31                  ; THE "IORST"-INSTR IN "CSAMS"-ROUTINE IS REPLACED WITH THE
32                  ; PROCEDURECALL "SWENB" AND MOVED 3 PLACES FORWARD.
33                  ;
34                  ; LOCATION IS: 1762
35
36
37
38                  ;CHANGES MADE TO PREVENT THE STANDARDROUTINES "SETPX" AND
39                  ;"EHALT" FROM EXECUTING A "IORST", WHEN THE TRACKCOUNTER HAS
40                  ;TO STAY <> 0.
41                  ;
42                  ;
43                  ; THE "IORST" - INSTR. IN ERROR ROUTINES "ENTCO" AND "CYMOR"
44                  ; ARE REPLACED BY THE "RESET"-PROCEDURE CALL.
45                  ;
46                  ; LOCATIONS ARE: 6447 AND 6512
47
48
49                  ;CHANGES TO MAKE THE "QUESTION OUTPUT"-ROUTINE ABLE TO
50                  ;HANDLE CHARACTERS.
51                  ;
52                  ;
53                  ; JUST BEFORE THE "QUEST"-LABEL IS INSERTED: MOV  1,0.
54                  ;
55                  ; LOCATION IS: 5234
56
57
58                  ;CHANGE MADE TO LET THE TTO-BELL RING IN CASE OF ERRORS.
59                  ;
60                  ; IN LOC. QHAAT+2 THE CWAIT-CALL IS REPLACED BY THE ROUTINECALL:
61                  ; "CBELL".
62                  ;
63                  ; LOCATION IS: 1231
```

```
I 0021   .MAIN
01
02                              .EOT
```

```
  0022  .MAIN
01
02                ;TAPE 3          OUTPUT ROUTINES AND OTHERS.
03
04        000412 .LOC 412          ;DO NOT MOVE UP.
05
06                                 ;FOR THE USE SEE EACH ROUTINE.
07
08                ;MESS    TYPE A TEXT MESSAGE ON TTO AND LPT
09                ;CHAR    TYPE A CHAR ON TTO AND LPT, CALCULATE P-BIT
10                ;TYPE    TYPE A CHAR ON TTO AND LPT
11                ;CRLF    TYPE A CR AND A LF ON TTO AND LPT
12                ;DISP    DISPLAY A TEXT MESSAGE ON DIS
13                ;DOUT    DISPLAY A CHAR ON DIS
14                ;DICL    CLEAR DIS
15                ;DISATT DISPLAY ATTENTION: BEEP AND WAIT 3 SECONDS
16                ;HAATT   HALT ATTENTION: BEEP
17                ;TBIN    TYPE BINARY NUMBER ON TTO AND LPT
18                ;TOCT    TYPE OCTAL NUMBER ON TTO AND LPT
19                ;TDEC    TYPE DECIMAL NUMBER ON TTO AND LPT
20                ;TZOC    TYPE OCTAL NUMBER WITHOUT LEADING ZEROES.
21                ;DBIN    DISPLAY BINARY NUMBER ON DIS
22                ;DOCT    DISPLAY OCTAL NUMBER ON DIS
23                ;DDEC    DISPLAY DECIMAL NUMBER ON DIS
24                ;DZOC    DISPLAY OCTAL NUMBER WITHOUT LEADING ZEROES.
25                ;WAIT    WAIT SOME MILLISECONDS
26                ;WATOP   WAIT FOR OPERATOR ACTION
27                ;TIMSK   TIMER FOR SKIP INSTRUCTION
28                ;TIMRO   TIMER FOR ROUTINE
29                ;TIMMS   TIME MESSURE ROUTINE
30                ;MULTI   MULTIPLY
31                ;DIVIS   DIVIDE SINGLE
32                ;DIVID   DIVIDE DOUBLE
33                ;QUES    OUTPUT QUESTIONS
34                ;SAMS    START ADDR MESSAGE
35                ;RESW    READ SWITCHES, REPORT CHANGES
36                ;SAPTB   MOVE PRINTER TABLE
37                ;SAMEX   EXTEND TO 64K WORDS MEM MODE
38                ;SAMNM   RESET TO 32K WORDS MEM MODE
39                ;REBIN   INITIALIZE, RESTORE BIN LOADER
40
41
42                ;AN ADD ON PRINTERTABLE SHOULD HAVE THIS FORMAT:
43                ;    .LOC 2000
44                ;    .RDX Y
45                ;    .TXTE?
46                ;    <6><0>       ;TOTALLY 200 BOXES WITH
47                ;                 ;<TYPE OF ACTION><CHAR TO PRINT>
48                ;    <6><0>?
49                ;    .END 101
50                ;                 ;TYPE OF ACTION:        0=PRINT
51                ;                                         6=BLIND
```

```
! 0023  .MAIN
01
02              ;LPT, TTO AND DIS NON INTERRUPT PACKAGE
03              ;IF THE DEVICE(S) ARE NOT PRESENT THE ROUTINES CONTINUES.
04              ;IF A DEVICE IS CONNECTED (SET TO ONLINE) THE PROGRAM
05              ;MUST BE RESTARTED IN THE WANTED STARTADDRESS.
06
07              ;LPT AND TTO ROUTINES:
08              ;>MESS< PRINTS ASCII MESSAGES AS SPECIFIED BY ASSEMBLER.
09              ;>CHAR< PRINTS ASCII CHARACTOR, AC0-R.
10              ;AC0-L MUST BE 0. CORRECTS THE PARITY, 11 SIMULATE TAB.
11              ;>TYPE< PRINTS AC0-R. MUST HAVE PROPER PARITY. RETURN IS
12              ;TO CALL+1.REPLACE THIS ROUTINE WITH INTERRUPT TYPE IF DESIRED.
13              ;>CRLF< PRINTS A CARRIAGE RETURN FOLLOWED BY A LINE FEED.
14              ;>TOCT< PRINTS AC1 IN OCTAL, 6 DIGITS AND 6 PRINTPOSITIONS.
15              ;>TDEC< PRINTS AC1 IN DECIMAL, LEADING 0'S SUPPRESSED, WITH SIGN
16              ;RANGE -32768 TO -1 AND +0 TO +32767, 6 PRINTPOS, UP TO 5 DIGITS.
17              ;>TOCT< PRINTS AC1 IN OCTAL, LEADING 0'S SUPPRESSED,
18              ;6 PRINTPOSITIONS AND UP TO 6 DIGITS.
19              ;THE ROUTINES TOCT, TZOC AND TDEC ARE FOLLOWED BY A TAB
20              ;BUT ONLY IF NOT ALLREADY AT A TABPOINT.
21              ;>TBIN< PRINTS AC1 IN BINARY, AC1-R 8 DIGITS ONLY, 8 PRINTPOS.
22
23              ;DIS ROUTINES:
24              ;>DISP< PRINTS ACSII MESSAGES AS SPECIFIED BY ASSEMBLER.
25              ;>DOUT< PRINTS ASSII CHARACTOR, AC0-R. AC0-L MUST BE 0. RETURN
26              ;TO CALL+1. REPLACE THIS ROUTINE WITH INTERRUPT TYPE IF DESIRED.
27              ;>DICL< CLEARS THE DISPLAY.
28              ;>DISATT< ATTENTION DISPLAY, SEE ROUTINE
29              ;>HAATT<  ATTENTION HALT, SEE ROUTINE
30              ;>DOCT< DISPLAYS AC1 IN OCTAL, 6 DIGITS AND 6 PRINTPOSITIONS.
31              ;>DDEC< DISPLAYS AC1 IN DECIMAL LEADING 0'S SUPPRESSED, WITH SIGN
32              ;RANGE -32768 TO -1 AND +0 TO +32767, 6 PRINTPOS, UP TO 5 DIGITS.
33              ;>DZOC< DISPLAYS AC1 IN OCTAL, LEADING 0'S SUPPRESSED,
34              ;6 PRINTPOSITIONS AND UP TO 6 DIGITS.
35              ;>DBIN< DISPLAYS AC1 IN BINARY, AC1-R 8 DIGITS ONLY, 8 PRINTPOS.
36
37              ;       LDA     1,NUMBER ;ALL NUMBER ROUTINES RESTORE AC1
38              ;CALL   CTDEC
39              ;       CTOCT
40              ;       CTZOC
41              ;       CDDEC
42              ;       CDOCT
43              ;       CDZOC
44              ;       CTBIN
45              ;       CDBIN
46              ;       MOVS    1,1     ;FOR THE OTHER 8 BITS
47              ;       CTBIN
48              ;       CDBIN
49              ;       LDA     0,CHAR  ;ALL CHARACTER ROUTINES RESTORE AC0
50              ;       CTYPE
51              ;       CCHAR
52              ;       CDOUT
53              ;       CDICL
54              ;       CDISP
55              ;       TEXTLABEL
56              ;       CDATT
57              ;       CCRLF
58              ;       CMESS
59              ;       LABELTEXT
60              ;       CHAAT
61              ;       HALT
```

```
I 0024  .MAIN
01
02 00412 054450 XMESS:  STA     3,RPOUT ;PRINT A TEXT MESSAGE
03 00413 171000         MOV     3,2     ;SPEC RETURN IF PRINT SW
04 00414 010446         ISZ     RPOUT
05 00415 004556         JSR     PINHI
06 00416 002444         JMP     @RPOUT  ;NO PRINT
07 00417 034443         LDA     3,RPOUT ;AC3 POINTS TO MESSAGEPOINTER+1
08 00420 031777         LDA     2,-1,3  ;AC2 POINTS TO MESSAGE
09 00421 024440         LDA     1,CHMAS ;A 8 BIT MASK
10 00422 021000 MESSA:  LDA     0,0,2   ;AC0=DATA WORD
11 00423 125112         MOVL#   1,1,SZC
12 00424 123701         ANDS    1,0,SKP
13 00425 123401         AND     1,0,SKP ;AC0=DATA CHARACTOR RIGHT
14 00426 151400         INC     2,2     ;INC TO NEXT WORD
15 00427 124000         COM     1,1     ;FLIP MASK
16 00430 004404         JSR     MESCH   ;PRINT
17 00431 000771         JMP     MESSA   ;ANOTHER
18 00432 004405         JSR     BZOUT   ;TERMINATE MESS
19 00433 002427         JMP     @RPOUT  ;EXIT
20
21 00434 101015 MESCH:  MOV#    0,0,SNR ;TEST LAST CHAR
22 00435 001401         JMP     1,3     ;RETURN +2 IF NULL
23 00436 002420         JMP     @ZCHAR  ;TYPE CHAR
24
25 00437 054412 BZOUT:  STA     3,RBZOT
26 00440 006063         TIMSK           ;WAIT IF LPT/TTO BUSY
27 00441 001750 XLPTT:  1000.           ;MAX 1 SEC LPT
28 00442 063517         SKPBZ   XLPT
29 00443 044776         STA     1,XLPTT ;REMOVE WAITING, LPT NOT CONNECTED
30 00444 006063         TIMSK
31 00445 000454 XTTOT:  300.            ;MAX 300 MSEC TTO
32 00446 063511         SKPBZ   XTTO
33 00447 044776         STA     1,XTTOT ;REMOVE WAITING, TTO NOT CONNECTED
34 00450 002401         JMP     @RBZOT
35
36 00451 000000 RBZOT:  0
37 00452 000000 RINHI:  0
38 00453 000000 SADIG:  0
39 00454 000000 RXDEC:  0
40 00455 000040 CHINH:  40
41 00456 000672 ZCHAR:  YCHAR
42 00457 001142 ZDOUT:  YDOUT
43 00460 000712 ZTAB3:  YTAB3
44 00461 000377 CHMAS:  377
45 00462 000000 RPOUT:  0
46 00463 000000 PDECR:  0
47
48 00464 054776 XDDEC:  STA     3,RPOUT ;DISPLAY DECIMAL NUMBER
49 00465 004415         JSR     YPDEC
50 00466 000502         JMP     NXDIS   ;DISPLAY NEXT DIGIT
51 00467 000411         JMP     EXDIS   ;EXIT
52
53 00470 054772 XTDEC:  STA     3,RPOUT ;TYPE DECIMAL NUMBER
54 00471 004411         JSR     YPDEC
55 00472 000402         JMP     TYPNX   ;TYPE NEXT DIGIT
56 00473 000404         JMP     EXTYP   ;EXIT
57
58 00474 054760 TYPNX:  STA     3,RXDEC
59 00475 004575         JSR     YCHAR   ;TYPE DIGIT
60 00476 002756         JMP     @RXDEC  ;NEXT DIGIT, SIGPR OR DECPR
61
62 00477 006761 EXTYP:  JSR     @ZTAB3  ;YPDEC/ZOCT/POCT FINISH RETURN, TYPE TAB
63 00500 024753 EXDIS:  LDA     1,SADIG ;DISPLAY FINISHED, RESTORE PARAM
64 00501 002761         JMP     @RPOUT  ;EXIT
```

```
I 0025  .MAIN
01
02 00502 044751 YPDEC:  STA     1,SADIG ;SAVE PARAM FOR REPEAT
03 00503 054760          STA     3,PDECR
04 00504 030756          LDA     2,RPOUT ;SPEC RETURN IF PRINT SW
05 00505 004466          JSR     PINHI
06 00506 002754          JMP     @RPOUT  ;NO PRINT
07 00507 020436          LDA     0,CHPLU ;SETUP PLUS
08 00510 040437          STA     0,CHFLG ;SIGN FLAG
09 00511 020552          LDA     0,CHSPA ;OR PLUS: CHPLU
10 00512 040432          STA     0,CHSIG ;PRINT SPACE OR PLUS
11 00513 102620          SUBZR   0,0     ;AC0:=100000
12 00514 106415          SUB#    0,1,SNR ;TEST FOR -32768
13 00515 000404          JMP     PDEC2
14 00516 107415          AND#    0,1,SNR ;TEST FOR NEGATIVE
15 00517 000404          JMP     PDEC3   ;POS
16 00520 124400          NEG     1,1     ;NEG
17 00521 020425 PDEC2:  LDA     0,CHMIN ;SETUP MINUS
18 00522 040422          STA     0,CHSIG
19 00523 020537 PDEC3:  LDA     0,CHSP  ;SUPPRESS LEADING ZEROES
20 00524 030516          LDA     2,DECTB ;PRINT AC1 IN DECIMAL
21 00525 000466          JMP     PDEC1
22
23 00526 054415 SIGN:   STA     3,SIGNR
24 00527 034420          LDA     3,CHFLG
25 00530 030415          LDA     2,CHPLU
26 00531 172414          SUB#    3,2,SZR ;TEST PRINT SIGN
27 00532 002411          JMP     @SIGNR  ;IF NO SIGN, RETURN
28 00533 034527          LDA     3,CHSP
29 00534 116415          SUB#    0,3,SNR ;TEST FIRST DIGIT
30 00535 002406          JMP     @SIGNR  ;IF LEADING SPACE, RETURN
31 00536 040411          STA     0,CHFLG ;STORE ASCII AND DESTROY
32 00537 020405          LDA     0,CHSIG ;PRINT SIGN FLAG
33 00540 006723          JSR     @PDECR  ;OUTPUT SIGN
34 00541 020406          LDA     0,CHFLG ;RESTORE ASCII DIGIT
35 00542 002401          JMP     @SIGNR  ;PRINT FIRST DIGIT
36
37 00543 000000 SIGNR:  0
38 00544 000000 CHSIG:  0
39 00545 000053 CHPLU:  53
40 00546 000055 CHMIN:  55
41 00547 000000 CHFLG:  0
42
43 00550 054712 XTZOC:  STA     3,RPOUT ;TYPE ZERO OCTAL NUMBER
44 00551 004430          JSR     YZOCT
45 00552 000722          JMP     TYPNX   ;TYPE NEXT DIGIT
46 00553 000724          JMP     EXTYP   ;EXIT
47
48 00554 054706 XDZOC:  STA     3,RPOUT ;DISPLAY ZERO OCTAL NUMBER
49 00555 004424          JSR     YZOCT
50 00556 000412          JMP     NXDIS   ;DISPLAY NEXT DIGIT
51 00557 000721          JMP     EXDIS   ;EXIT
52
53 00560 054702 XTOCT:  STA     3,RPOUT ;TYPE OCTAL NUMBER
54 00561 004422          JSR     YPOCT
55 00562 000712          JMP     TYPNX   ;TYPE NEXT DIGIT
56 00563 000714          JMP     EXTYP   ;EXIT
57
58 00564 054676 XDOCT:  STA     3,RPOUT ;DISPLAY OCTAL NUMBER
59 00565 004416          JSR     YPOCT
60 00566 000402          JMP     NXDIS   ;DISPLAY NEXT DIGIT
61 00567 000711          JMP     EXDIS   ;EXIT
62
63 00570 054664 NXDIS:  STA     3,RXDEC
64 00571 006666          JSR     @ZDOUT  ;DISPLAY DIGIT
65 00572 002662          JMP     @RXDEC  ;NEXT DIGIT
```

```
I 0026  .MAIN
01
02 00573 054657 PINHI:  STA     3,RINHI ;AFTER ACTIVATING SETPX IN TESTLOOP
03 00574 000403 SETAC:  JMP     NINHI   ;THIS IS CHANGED TO CRESW (READS 2 ROL)
04 00575 034660         LDA     3,CHINH ;SW 10 FOR NO PRINTING
05 00576 157405         AND     2,3,SNR ;RETURN+1 IF INHIBIT
06 00577 010653 NINHI:  ISZ     RINHI   ;RETURN+2 IF PRINTING
07 00600 002652         JMP     @RINHI  ;EXIT
08
09 00601 020461 YZOCT:  LDA     0,CHSP  ;ZERO OCTAL ROUTINE
10 00602 101001         MOV     0,0,SKP
11
12 00603 020565 YPOCT:  LDA     0,CHAR0 ;OCTAL ROUTINE
13 00604 044647         STA     1,SADIG ;SAVE PARAM FOR REPEAT
14 00605 054656         STA     3,PDECR
15 00606 030654         LDA     2,RPOUT ;SPEC RETURN IF PRINT SW
16 00607 004764         JSR     PINHI
17 00610 002652         JMP     @RPOUT  ;NO PRINT
18 00611 030440         LDA     2,OCTAB ;PRINT AC1 IN OCTAL
19 00612 040735         STA     0,CHFLG ;NO-SIGN FLAG
20 00613 040556 PDEC1:  STA     0,ZSUPP ;BOTH ENTRYS PRINT NUMBER
21 00614 050401         STA     2,.+1   ;THEN TAB TO NEXT POSITION
22 00615 000000 DECOCT: 0               ;A LDA 2,TABLE INSTRUCTION
23 00616 010777         ISZ     .-1
24 00617 151005         MOV     2,2,SNR ;IF TABLE ENTRY=0 THEN
25 00620 000420         JMP     DECEX   ;EXIT WITH TAB SPECIAL IF TYPE ROU.
26 00621 034550         LDA     3,ZSUPP ;ZEROS SUPPRESS STUF
27 00622 102400         SUB     0,0
28 00623 146512 DECOT:  SUBL#   2,1,SZC
29 00624 000405         JMP     DECP
30 00625 146400         SUB     2,1     ;FORM THE DIGIT
31 00626 034542         LDA     3,CHAR0 ;DO NOT SUPPRESS
32 00627 101400         INC     0,0     ;FOLLOWING ZEROES
33 00630 000773         JMP     DECOT
34 00631 151235 DECP:   MOVZR#  2,2,SNR ;IF LAST DIGIT THEN
35 00632 034536         LDA     3,CHAR0 ;AC3=ZERO, NOT SUPPR CHAR
36 00633 054536         STA     3,ZSUPP ;AC0=DIGIT
37 00634 163000         ADD     3,0     ;MAKE ASCII
38 00635 004671         JSR     SIGN    ;TEST SIGN
39 00636 006625         JSR     @PDECR  ;OUTPUT DIGIT
40 00637 000756         JMP     DECOCT  ;GET NEXT DIGIT
41 00640 010623 DECEX:  ISZ     PDECR   ;RETURN ADDR FOR JSR Y-ROUTINE
42 00641 002622         JMP     @PDECR  ;EXIT WITH TAB IF TYPING ROUTINE
43
44 00642 030426 DECTB:  LDA     2,.+1+.-DECOCT
45       000012 .RDX 10
46 00643 023420         10000
47 00644 001750         1000
48 00645 000144         100
49 00646 000012         10
50 00647 000001         1
51 00650 000000         0
52       000010 .RDX 8
53
54 00651 030435 OCTAB:  LDA     2,.+1+.-DECOCT
55 00652 100000         100000
56 00653 010000         10000
57 00654 001000         1000
58 00655 000100         100
59 00656 000010         10
60 00657 000001         1
61 00660 000000         0
62
63 00661 000000 CHRET:  0
64 00662 000240 CHSP:   240             ;LEADING ZERO SUPPRESS CHAR + TAB SIML
65 00663 000240 CHSPA:  240             ;A SPACE FOR + IN PDEC
```

```
| 0027  .MAIN
01
02 00664 040507 XCHAR:  STA     0,SACHA ;SAVE PARAM FOR REPEAT
03 00665 054774         STA     3,CHRET
04 00666 171000         MOV     3,2     ;SPEC RETURN IF PRINT SW
05 00667 004704         JSR     PINHI
06 00670 002771         JMP     @CHRET  ;NO TYPE
07 00671 000402         JMP     QCHAR
08
09 00672 054767 YCHAR:  STA     3,CHRET ;PRINT AC0 RIGHT
10 00673 101320 QCHAR:  MOVZS   0,0
11 00674 040500         STA     0,CHSAV
12 00675 176000 CHAR2:  ADC     3,3     ;COMPUTE THE PARITY
13 00676 117000         ADD     0,3     ;FOR EVEN
14 00677 163404         AND     3,0,SZR
15 00700 000775         JMP     CHAR2
16 00701 176660         SUBCR   3,3     ;COMBIND PARITY WITH CHAR
17 00702 020472         LDA     0,CHSAV
18 00703 163300         ADDS    3,0
19 00704 034463 CHAR1:  LDA     3,CHTAB ;IS THIS A TAB
20 00705 116405         SUB     0,3,SNR
21 00706 000413         JMP     CHAR4   ;YES
22 00707 004423         JSR     YTYPE   ;NO PRINT IT
23 00710 020463         LDA     0,SACHA ;RESTORE PARAM
24 00711 002750         JMP     @CHRET  ;EXIT
25
26 00712 054747 YTAB3:  STA     3,CHRET ;NUMBER ROUTINE TAB RETURN
27 00713 020462 CHAR3:  LDA     0,CHORZ ;SIMULATE A TAB
28 00714 034462         LDA     3,CHAR7 ;VIA 1 TO 8 SPACES
29 00715 117404         AND     0,3,SZR
30 00716 000403         JMP     CHAR4
31 00717 020454         LDA     0,SACHA ;RESTORE PARAM
32 00720 002741         JMP     @CHRET  ;EXIT AFTER TAB
33 00721 020741 CHAR4:  LDA     0,CHSP
34 00722 004410         JSR     YTYPE
35 00723 000770         JMP     CHAR3
36
37 00724 040554 XTYPE:  STA     0,REG0  ;SAVE PARAM FOR REPEAT
38 00725 054556         STA     3,REG3  ;SAVE RETURN
39 00726 171000         MOV     3,2     ;SPEC RETURN IF PRINT SW
40 00727 004644         JSR     PINHI
41 00730 002553         JMP     @REG3   ;NO TYPE
42 00731 000403         JMP     QTYPE
43
44 00732 054551 YTYPE:  STA     3,REG3  ;SAVE RETURN
45 00733 040545         STA     0,REG0  ;SAVE PARAM FOR TYPING
46 00734 010441 QTYPE:  ISZ     CHORZ   ;INC HORIZIONAL POSITION
47 00735 044544         STA     1,REG1  ;SAVE AC1 AND AC2 FOR NUMBER AND
48 00736 050544         STA     2,REG2  ;MESS ROUTINES
49 00737 006440         JSR     @TBZOT
50 00740 020540         LDA     0,REG0
51 00741 000437         JMP     XFORM
52 00742 061017 TYPE1:  DOA     0,XLPT  ;SEND CHAR
53 00743 060117         NIOS    XLPT    ;START LPT
54 00744 020534 TYPE2:  LDA     0,REG0
55 00745 024532         LDA     1,CHLF
56 00746 106415         SUB#    0,1,SNR ;LF ?
57 00747 000404         JMP     TYPE3   ;YES, LF
58 00750 024526         LDA     1,CHCR
59 00751 106414         SUB#    0,1,SZR ;CR ?
60 00752 000404         JMP     TYPE4   ;NO CR
61 00753 006061 TYPE3:  CWAIT           ;IF CR, LF WAIT FOR
62 00754 001237         SECM2           ;DATAPOINT 20 MSEC
63 00755 020523         LDA     0,REG0  ;RESTORE AC0
64 00756 061011 TYPE4:  DOA     0,XTTO  ;SEND CHAR
65 00757 060111         NIOS    XTTO    ;START TTO
```

```
I 0028  .MAIN
01
02 00760 152400 TYPES:  SUB     2,2
03 00761 024516         LDA     1,CHLF
04 00762 106415         SUB#    0,1,SNR ;IF LF
05 00763 050412         STA     2,CHORZ ;CLEAR HORZ POS
06 00764 030516         LDA     2,REG2
07 00765 024514         LDA     1,REG1  ;AC0 = REG0 FOR REPEAT
08 00766 002515         JMP     @REG3   ;EXIT
09
10 00767 000011 CHTAB:  11
11 00770 000060 CHAR0:  60
12 00771 000000 ZSUPP:  0
13 00772 177770 NN10:   -10
14 00773 000000 SACHA:  0
15 00774 000000 CHSAV:  0
16 00775 000000 CHORZ:  0
17 00776 000007 CHAR7:  7
18 00777 000437 TBZOT:  BZOUT
19
20 01000 030561 XFORM:  LDA     2,RMSK  ;TRANSFORM CHAR TO
21 01001 143400         AND     2,0     ;PRINTER ALPHABET
22 01002 030556         LDA     2,PTAB
23 01003 113000         ADD     0,2     ;AC2:=CHAR+TABLE
24 01004 025000         LDA     1,0,2
25 01005 030554         LDA     2,RMSK
26 01006 133400         AND     1,2     ;AC2:=CLASS
27 01007 020554         LDA     0,LMSK
28 01010 123700         ANDS    1,0     ;AC0:=CHAR
29 01011 024443         LDA     1,ACTN
30 01012 133000         ADD     1,2     ;AC2:=ACTN+1+CLASS
31 01013 003000         JMP     @0,2    ;GO TO ACTION
32 01014 054461 XTBIN:  STA     3,CRBIR
33 01015 004407         JSR     YPBIN
34 01016 004654         JSR     YCHAR   ;TYPE DIGIT
35 01017 000420         JMP     PBINN   ;NEXT DIGIT
36 01020 054455 XDBIN:  STA     3,CRBIR
37 01021 004403         JSR     YPBIN
38 01022 004520         JSR     YDOUT   ;DISPLAY DIGIT
39 01023 000414         JMP     PBINN   ;NEXT DIGIT
40 01024 044447 YPBIN:  STA     1,SABIN ;SAVE PARAM FOR REPEAT
41 01025 054447         STA     3,PBINR
42 01026 030447         LDA     2,CRBIR ;SPEC RETURN IF PRINT SW
43 01027 006535         JSR     @TINHI
44 01030 002445         JMP     @CRBIR  ;NO PRINT
45 01031 030741         LDA     2,NN10  ;8 TIMES
46 01032 125300         MOVS    1,1
47 01033 020735 PBINC:  LDA     0,CHAR0
48 01034 125102         MOVL    1,1,SZC
49 01035 101400         INC     0,0     ;AC0:="CHAR1"
50 01036 002436         JMP     @PBINR  ;OUTPUT DIGIT
51 01037 151404 PBINN:  INC     2,2,SZR
52 01040 000773         JMP     PBINC
53 01041 024432         LDA     1,SABIN ;RESTORE PARAM
54 01042 002433         JMP     @CRBIR
55
56 01043 054432 XCRLF:  STA     3,CRBIR ;SAVE RETURN
57 01044 171000         MOV     3,2     ;SPEC RETURN IF PRINT SW
58 01045 006517         JSR     @TINHI
59 01046 002427         JMP     @CRBIR  ;NO TYPE
60 01047 020427         LDA     0,CHCR
61 01050 004622         JSR     YCHAR
62 01051 020426         LDA     0,CHLF
63 01052 004620         JSR     YCHAR   ;PRINT CR,LF
64 01053 002422         JMP     @CRBIR  ;EXIT
```

```
[ 0029  .MAIN
01
02 01054 001055 ACTN:   .+1              ;ACTION ENTRY TABLE
03 01055 001071         ACT0             ;NORMAL ACTION
04 01056 001067         ILL
05 01057 001067         ILL
06 01060 001067         ILL
07 01061 001067         ILL
08 01062 001067         ILL
09 01063 001072         ACT6             ;BLIND
10 01064 001067         ILL
11 01065 001067         ILL
12 01066 001067         ILL
13
14 01067 063077 ILL:    HALT
15 01070 000777         JMP     .-1      ;ILLEGAL
16 01071 000651 ACT0:   JMP     TYPE1
17 01072 000652 ACT6:   JMP     TYPE2
18
19 01073 000000 SABIN:  0
20 01074 000000 PBINR:  0
21 01075 000000 CRBIR:  0
22 01076 000215 CHCR:   215
23 01077 000012 CHLF:   12
24 01100 000000 REG0:   0
25 01101 000000 REG1:   0
26 01102 000000 REG2:   0
27 01103 000000 REG3:   0
28
29 01104 054771 XDISP:  STA     3,CRBIR ;DISPLAY MESSAGE
30 01105 171000         MOV     3,2     ;SPEC RETURN IF PRINT SW
31 01106 010767         ISZ     CRBIR
32 01107 006455         JSR     @TINHI
33 01110 002765         JMP     @CRBIR  ;NO PRINT
34 01111 034764         LDA     3,CRBIR ;AC3=POINTS TO MESSAGEPOINTER+1
35 01112 031777         LDA     2,-1,3  ;AC2 POINTS TO MESSAGE
36 01113 024447         LDA     1,CMSK
37 01114 021000 DISP1:  LDA     0,0,2   ;AC0=DATAWORD
38 01115 125112         MOVL#   1,1,SZC
39 01116 123701         ANDS    1,0,SKP
40 01117 123401         AND     1,0,SKP ;AC0=CHAR. RIGHT
41 01120 151400         INC     2,2     ;INC TO NEXT WORD
42 01121 124000         COM     1,1     ;FLIP MASK
43 01122 004407         JSR     DDICH   ;GO DISPLAY
44 01123 000771         JMP     DISP1   ;ANOTHER
45 01124 006063         TIMSK           ;TERMINATE DISP
46 01125 000050 FDIST:  40.             ;MAX 40 MSEC
47 01126 063535         SKPBZ   DIS
48 01127 044776         STA     1,FDIST ;REMOVE WAITING, DIS NOT CONNECTED
49 01130 002745         JMP     @CRBIR  ;RETURN
50
51 01131 101015 DDICH:  MOV#    0,0,SNR ;TEST LAST CHAR
52 01132 001401         JMP     1,3     ;RETURN +2 IF NULL
53 01133 000407         JMP     YDOUT   ;DISPLAY
```

```
| 0030  .MAIN
01
02 01134 040744 XDOUT:  STA     0,REG0  ;SAVE PARAM FOR REPEAT
03 01135 054746         STA     3,REG3  ;SAVE RETURN
04 01136 171000         MOV     3,2     ;SPEC RETURN IF PRINT SW
05 01137 006425         JSR     @TINHI
06 01140 002743         JMP     @REG3   ;NO DISPLAY
07 01141 000403         JMP     QDOUT
08
09 01142 054741 YDOUT:  STA     3,REG3  ;SAVE RETURN
10 01143 040735         STA     0,REG0  ;SAVE PARAM FOR DISPLAYING
11 01144 044735 QDOUT:  STA     1,REG1  ;SAVE AC1 AND AC2 FOR NUMBER AND
12 01145 050735         STA     2,REG2  ;MESS ROUTINES
13 01146 006063         TIMSK           ;WAIT IF DIS BUSY
14 01147 000050 SDIST:  40.             ;MAX 40 MSEC
15 01150 063535         SKPBZ   DIS
16 01151 044776         STA     1,SDIST ;REMOVE WAITING, DIS NOT CONNECTED
17 01152 020726         LDA     0,REG0
18 01153 061035         DOA     0,DIS   ;SEND CHAR
19 01154 060135         NIOS    DIS     ;START DIS
20 01155 024724         LDA     1,REG1
21 01156 030724         LDA     2,REG2  ;AC0 = REG0 FOR REPEAT
22 01157 002724         JMP     @REG3   ;EXIT
23
24 01160 002000 PTAB:   TABLE
25 01161 000177 RMSK:   177
26 01162 000377 CMSK:   377
27 01163 077400 LMSK:   077400
28 01164 000573 TINHI:  PINHI
29
30 01165 054710 XDICL:  STA     3,CRBIR ;SAVE RETURN
31 01166 171000         MOV     3,2     ;SPEC RETURN IF PRINT SW
32 01167 006775         JSR     @TINHI
33 01170 002705         JMP     @CRBIR  ;NO DISPLAY
34 01171 000402         JMP     QDICL
35
36 01172 054703 YDICL:  STA     3,CRBIR
37 01173 020704 QDICL:  LDA     0,CHLF
38 01174 004746         JSR     YDOUT   ;DISPLAY CLEAR
39 01175 002700         JMP     @CRBIR  ;RETURN
```

```
I 0031  .MAIN
01
02                          ;RC 3600, ATTENTION DISPLAY OUTPUT
03                          ;BY MEANS OF ACCUSTIC ALARM FOR 50 MS
04                          ;AND WAIT FOR 3 SECONDS TO ALLOW
05                          ;THE OPERATOR TO READ THE MESSAGE.
06                          ;
07                          ;CALL   CDATT
08                          ;
09                          ;RC 3600, ATTENTION HALT BY MEANS OF
10                          ;ACOUSTIC ALARM FOR 50 MS BEFORE HALT.
11                          ;
12                          ;CALL   CHAAT
13                          ;
14 01176 054676 DISATT: STA     3,PBINR
15 01177 171000         MOV     3,2     ;SPEC RETURN IF PRINT SW
16 01200 006764         JSR     @TINHI
17 01201 002673         JMP     @PBINR  ;NO PRINT
18 01202 004424         JSR     YHAAT
19 01203 006073         CRESW           ;READS 2 ROUTINE
20 01204 024436         LDA     1,SECS4 ;SW13 WAITING DIS
21 01205 133415         AND#    1,2,SNR
22 01206 000403         JMP     DISSW
23 01207 006061         CWAIT
24 01210 001236         SEC3
25
26 01211 006073 DISSW:  CRESW           ;READS 2 ROUTINE
27 01212 024427         LDA     1,SECS2
28 01213 133414         AND#    1,2,SZR
29 01214 063077         HALT            ;HALT IF SWITCH 14
30 01215 126520         SUBZL   1,1     ;AC1:=1
31 01216 133414         AND#    1,2,SZR
32 01217 004753         JSR     YDICL   ;CLEAR DIS IF SWITCH 15
33 01220 002654         JMP     @PBINR
34
35 01221 054654 HAATT:  STA     3,CRBIR
36 01222 171000         MOV     3,2     ;SPEC RETURN IF PRINT SW
37 01223 006741         JSR     @TINHI
38 01224 002651         JMP     @CRBIR  ;NO "PRINT"
39 01225 000402         JMP     QHAAT
40
41 01226 054647 YHAAT:  STA     3,CRBIR
42 01227 020414 QHAAT:  LDA     0,SEC4K
43 01230 061032         DOA     0,FUN
44 01231 006132         CBELL           ;LET TTY BELL RING
45 01232 000401         JMP     .+1
46 01233 102400         SUB     0,0
47 01234 061032         DOA     0,FUN
48 01235 002640         JMP     @CRBIR
49
50       000012         .RDX 10         ;CHANGING SEC3 AFFECTS CRESW AND POWON%
51 01236 005670 SEC3:   3000            ;CHANGE TO 1000(0) FOR 1(0) SEC WAITING.
52 01237 000024 SECM2:  20              ;20 MSEC FOR TYPE ROUTINE DATAPOINT DELAY
53 01240 000062 SECM5:  50
54       000010         .RDX 8
55 01241 000002 SECS2:  2
56 01242 000004 SECS4:  4
57 01243 004000 SEC4K:  4000
```

```
I 0032  .MAIN
01
02                ;PROCEDURE WAIT OPERATOR
03                ;TURNS ON THE FUNCTION INDICATOR FROM ARG,
04                ;WAITS FOR THE OPERATOR TO PRESS BUTTON FROM ARG,
05                ;OR TO PRESS KEY AT TTY
06                ;AND TURNS OFF THE FUNCTION INDICATOR FROM ARG.
07                ;AND WRITES PROG NAME AT DIS (SEE LABEL: PROG),
08                ;OR TYPE $ AT TTY
09                ;UNCHANGED:      AC0
10                ;DESTROYED:      AC1,AC2,AC3,CARRY
11                ;CALL:  WATOP
12                ;       ARG
13
14 01244 025400 XWTOP:  LDA     1,0,3   ; FUNCTION MASK FROM ARGUMENT
15 01245 175400         INC     3,3     ; FOR INDICATORS AND BUTTONS
16 01246 054437         STA     3,WTORE ; SAVE RETURN
17 01247 034440         LDA     3,WHIGH
18 01250 040441         STA     0,WACSAV
19 01251 020435         LDA     0,WLOWL
20 01252 065032         DOA     1,FUN   ; TURN ON INDICATOR ARG
21 01253 063610 WTNOK:  SKPDN   XTTI
22 01254 000411         JMP     WTFUB
23 01255 070410         DIA     2,XTTI
24 01256 151300         MOVS    2,2     ; REMOVE P-BIT BY
25 01257 151120         MOVZL   2,2     ; MULTIPLYING WITH 512 DEC.
26 01260 060210         NIOC    XTTI
27 01261 172033         ADCZ#   3,2,SNC ; IS KEY BETWEEN OR
28 01262 142032         ADCZ#   2,0,SZC ; EQUAL TO LIMITS ?
29 01263 000402         JMP     WTFUB   ; NO
30 01264 000404         JMP     WTOK    ; YES
31 01265 070433 WTFUB:  DIA     2,FUB   ; SENSE BUTTON ARG
32 01266 147415         AND#    2,1,SNR ; IS ARG PRESSED?
33 01267 000764         JMP     WTNOK   ; NO - SENSE AGAIN
34 01270 152400 WTOK:   SUB     2,2
35 01271 071032         DOA     2,FUN   ; YES - TURN OFF INDICATOR ARG
36 01272 070433         DIA     2,FUB   ; WAIT TILL THE OPERATOR
37 01273 147414         AND#    2,1,SZR ; GETS HIS DIRTY FINGER
38 01274 000776         JMP     .-2     ; OFF THE BUTTON
39 01275 020413         LDA     0,WCH44
40 01276 006041         CCHAR
41 01277 006046         CDICL
42 01300 006044         CDISP
43 01301 007226         PROG            ; "ACTUAL PROGRAM NAME"
44 01302 006050         CHAAT
45 01303 020406         LDA     0,WACSAV
46 01304 002401         JMP     @WTORE  ; RETURN
47 01305 000000 WTORE:  0
48 01306 011000 WLOWL:  11*1000         ; LOW LIMIT CHAR * 219
49 01307 040000 WHIGH:  40*1000         ; HIGH LIMIT CHAR * 219
50 01310 000044 WCH44:  44
51 01311 000000 WACSAV: 0
52
53                ;INSERT ACTUAL PROGRAM NAME AFTER A LABEL "PROG:"
```

```
I 0033  .MAIN
01
02                ;ROUTINE READ SWITCHES.
03                ;SWITCHES ARE READ TO AC2. AC0 AND AC1 ARE SAVED.
04                ;THERE WILL BE A PAUSE FOR 3 SEC TO SEE IF MORE SWITCHES
05                ;ARE CHANGED. LAST SW REG CONTENT IN LSTSW.
06                ;SWITCH SETTINGS ARE REPORTED AT TTY/LPT.
07                ;CALL   CRESW
08                ;       RETURN
09
10 01312 040451 XRESW:  STA     0,SSWR0 ;SAVE AC'S
11 01313 044451         STA     1,SSWR1
12 01314 050451         STA     2,SSWR2 ;FOR PRINT INHIBIT RETURN
13 01315 054445         STA     3,RRESW ;SAVE RETURN
14 01316 024450         LDA     1,LSTSW ;LAST SW REG
15 01317 006130         RDSWS           ;NEW SW REG
16 01320 132415         SUB#    1,2,SNR ;CHANGED ?
17 01321 000431         JMP     NRESW   ;NO, RETURN
18 01322 050444 ARESW:  STA     2,LSTSW ;YES, SEE IF MORE CHANGE
19 01323 006061         CWAIT           ;AFTER 3 SECONDS
20 01324 001243         SEC4K           ;MODIFIED TO 2 SEC (HH)
21 01325 024441         LDA     1,LSTSW
22 01326 006130         RDSWS
23 01327 132414         SUB#    1,2,SZR ;CHANGED ?
24 01330 000772         JMP     ARESW   ;YES, WAIT MORE CHANGING
25 01331 050435         STA     2,LSTSW ;NO, CHANGE FINISHED
26 01332 026436         LDA     1,@XSTAC
27 01333 044436         STA     1,SSTAC ;SAVE PRINT INHIBIT SITU
28 01334 030436         LDA     2,KSTAC ;GET PRINT INHI INITIAL FOR PRINT
29 01335 052433         STA     2,@XSTAC ;SW REG WITHOUT FURTHER CALL CRESW
30 01336 006043         CCRLF
31 01337 006040         CMESS
32 01340 001522         MSWRG           ;SWITCHES: XXXXXX
33 01341 024425         LDA     1,LSTSW
34 01342 006052         CTOCT
35 01343 006424         JSR     @WBZOT  ;WAIT LPT,TTO
36 01344 030425         LDA     2,SSTAC
37 01345 052423         STA     2,@XSTAC ;RESTORE PRINT INHIBIT SITU
38 01346 034414         LDA     3,RRESW ;IF RRESW IS ALMOST EQUAL
39 01347 030421         LDA     2,XSTAC ;SETAC IT WAS PRINT INHIBIT
40 01350 156645         SUBOR   2,3,SNR ;WHO CALLED. DON'T RETURN
41 01351 000405         JMP     BRESW   ;BUT REPEAT THE OUTPUT ROUTINE
42 01352 020411 NRESW:  LDA     0,SSWR0
43 01353 024411         LDA     1,SSWR1 ;RESTORE AC'S
44 01354 030412         LDA     2,LSTSW ;NEW SW POSITION
45 01355 002405         JMP     @RRESW  ;RETURN
46
47 01356 014407 BRESW:  DSZ     SSWR2   ;REPEAT CALL OF ROUTINE
48 01357 020404         LDA     0,SSWR0 ;WHICH WAS INTERRUPTED
49 01360 024404         LDA     1,SSWR1 ;IN PRINT INHIBIT BY CRESW.
50 01361 002404         JMP     @SSWR2  ;RETURN TO MAIN PROG CALL.
51
52 01362 000000 RRESW:  0
53 01363 000000 SSWR0:  0
54 01364 000000 SSWR1:  0
55 01365 000000 SSWR2:  0
56 01366 000000 LSTSW:  0
57 01367 000437 WBZOT:  BZOUT
58 01370 000574 XSTAC:  SETAC
59 01371 000000 SSTAC:  0
60 01372 000403 KSTAC:  403             ;JMP NINHI, ASM VALUE FROM SETAC
61 01373 000000 LALOC:  0               ;LAST LOC IN MEM
62 01374 077777 LAPRG:  077777          ;LAST LOC FOR PROGRAM
63 01375 077635 BINFI:  77635           ;FIRST ADDR OF BINARY LOADER IF 32K.
64 01376 000177 BINLA:  177             ;CONSTANT FOR LAST ADDR IN MEM
65 01377 002756 XFITY:  FITYP
```

```
1 0034  .MAIN
01
02                ;SUBROUTINE TO RELOAD BOOTSTRAP LOADER AND BINARY LOADER.
03                ;DO USE IN HEAD OF ANY PROGRAM AS MANY THINGS ARE INITIALIZED.
04                ;FOR LOADERS INFO SEE LOADER BELOW.
05                ;
06                ;CALL   START IN REBIN AFTER LOADING PROGRAM SELFSTARTING.
07
08 01400 020076 REBIN:  LDA     0,POWZE ;SET RESTART ADDR TO CELL ZERO
09 01401 040000         STA     0,0     ;FOR POWER RESTART.
10 01402 006465         JSR     @XMEXT  ;TRY TO SET MEM EXT FLAG (RC3603 ONLY)
11 01403 006463         JSR     @XMEND  ;STORE MEM END LOC IN HMEND=AC2=XX7600
12 01404 024772         LDA     1,BINLA ;AC1:=177
13 01405 147000         ADD     2,1     ;AC1:=LAST LOC IN MEM=XX7777
14 01406 044765         STA     1,LALOC ;STORE IT FOR PRINT
15 01407 125112         MOVL#   1,1,SZC ;MORE THAN 32K ?
16 01410 024764         LDA     1,LAPRG ;YES, AC1:=077777, LAST LOC FOR PROG
17 01411 034764         LDA     3,BINFI ;NO, LAST LOC IN FIRST 32K
18 01412 137400         AND     1,3     ;AC3:=FIRST ADDR LOADER
19 01413 030534         LDA     2,BINAD ;AC2:=ADDR OF BIN DATA
20 01414 021000 MODAT:  LDA     0,0,2
21 01415 041400         STA     0,0,3   ;MOVE DATA
22 01416 166415         SUB#    3,1,SNR ;LAST DATA ?
23 01417 000404         JMP     GETYP   ;YES
24 01420 151400         INC     2,2
25 01421 175400         INC     3,3     ;NEXT DATA ADDR
26 01422 000772         JMP     MODAT
27 01423 126400 GETYP:  SUB     1,1
28 01424 044742         STA     1,LSTSW ;SET LAST SW REG TO ALL ZERO
29 01425 006752         JSR     @XFITY  ;FIND CPU TYPE
30 01426 006073         CRESW           ;OUTPUT OF SWITCH SETTINGS
31 01427 006436         JSR     @XPCPT  ;PRINT CPU TYPE
32 01430 006044         CDISP
33 01431 001501         MLLOC
34 01432 006040         CMESS
35 01433 001501         MLLOC           ;<15><12>LAST LOC. XX7777
36 01434 024737         LDA     1,LALOC
37 01435 006052         CTOCT
38 01436 006056         CDOCT
39 01437 006061         CWAIT           ;WAIT 3 SEC TO READ MESS. DON'T
40 01440 001236         SEC3            ;USE CDATT AS SW AREN'T SET.
41 01441 006044         CDISP
42 01442 001510         MBILO
43 01443 006040         CMESS
44 01444 001510         MBILO           ;<15><12>BINARY LOADER OK
45 01445 006061         CWAIT
46 01446 001236         SEC3
47 01447 006046 MSPRO:  CDICL
48 01450 006044         CDISP
49 01451 007226         PROG
50 01452 006043         CCRLF
51 01453 006040         CMESS
52 01454 007226         PROG            ;ACTUAL PROGRAM NAME
53 01455 006061         CWAIT
54 01456 001236         SEC3
55 01457 062701         DICP    0,1     ;TRY TO SET MEM EXT FLAG (RC3603 ONLY)
56 01460 020003         LDA     0,3     ;CELL 3 = 0 FOR HALT
57 01461 101004         MOV     0,0,SZR ;CELL 3 = 1 FOR
58 01462 002004         JMP     @4      ;SELFSTART ADDR 4
59 01463 002401         JMP     @,+1    ;CHOOSE YOUR OWN START ADDR,
60 01464 004740         SWISA           ;NORMAL HALT, BUT IF NO SWITCH PANEL ?
61
62 01465 003016 XPCPT:  PRTYP
63 01466 001716 XMEND:  GMEND
64 01467 002340 XMEXT:  TMEND
```

```
| 0035  .MAIN
01
02        000001          .TXTM 1          ;RDOS TEXT PACKING MODE
03
04                  MELOC:  .TXT !LOADING UNIT OFF! ;"LOADING UNIT OFF"
    01470 046117
    01471 040504
    01472 044516
    01473 043440
    01474 052516
    01475 044524
    01476 020117
    01477 043106
    01500 000000
05
06        000000          .TXTM 0          ;NORMAL TEXT PACKING MODE
07
08                  MLLOC:  .TXT !<15><12>LAST LOC. !        ;"<15><12>LAST LOC. "
    01501 005015
    01502 040514
    01503 052123
    01504 046040
    01505 041517
    01506 020056
    01507 000000
09
10                  MBILO:  .TXT !<15><12>BINARY LOADER OK!
    01510 005015
    01511 044502
    01512 040516
    01513 054522
    01514 046040
    01515 040517
    01516 042504
    01517 020122
    01520 045517
    01521 000000
11                          ;"<15><12>BINARY LOADER OK"
12
13                  MSWRG:  .TXT !SWITCHES: !        ;"SWITCHES: "
    01522 053523
    01523 052111
    01524 044103
    01525 051505
    01526 020072
    01527 000000
14
15                  MSAMS:  .TXT ! STARTADDR!        ;" STARTADDR"
    01530 051440
    01531 040524
    01532 052122
    01533 042101
    01534 051104
    01535 000000
16
17                  UNTIM:  .TXT !MISERABLE TIMING! ;"MISERABLE TIMING"
    01536 044515
    01537 042523
    01540 040522
    01541 046102
    01542 020105
    01543 044524
    01544 044515
    01545 043516
    01546 000000
```

```
| 0036  ,MAIN
01
02                ;BINARY LOADER TS
03                ;ERRORHALT XX7752 FOR OVERWRITE LOADER OR CHECKSUM ERROR.
04                ;READYHALT XX7676 IF LOADED PROG ISN'T SELFSTARTING.
05                ;ERRORBLOCK=IGNORE BLOCK
06                ;REPEAT BLOCK=MULTIPLE DATA BLOCK
07                ;COUNT=WORD COUNT IN BLOCK
08
09 01547 001550 BINAD:  .+1               ;ADDR OF BIN LOADER DATA
10                ;SUBROUTINE TO BUILD A WORD IN AC2
11 01550 054425 BUILD:  STA     3,TEMP1 ;SAVE RETURN
12 01551 004406         JSR     GTCHR   ;GET FIRST BYTE
13 01552 171300         MOVS    3,2     ;PUT INTO LH OF AC2
14 01553 004404         JSR     GTCHR   ;GET NEXT BYTE
15 01554 173300         ADDS    3,2     ;FORM WORD IN AC2
16 01555 143000         ADD     2,0     ;ADD INTO CHECKSUM
17 01556 002417         JMP     @TEMP1  ;AND RETURN
18
19                ;READ A BYTE INTO AC3
20                ;IF SWITCH0=0 USE TELETYPE ELSE USE PTR
21 01557 054417 GTCHR:  STA     3,TEMP2 ;SAVE RETURN
22 01560 034417         LDA     3,SAVE  ;TEST WHICH DEVICE
23 01561 175103         MOVL    3,3,SNC
24 01562 000406         JMP     GTTTI   ;TTI
25 01563 063612         SKPDN   PTR     ;PTR
26 01564 000777         JMP     .-1
27 01565 074412         DIA     3,PTR   ;READ AND START
28 01566 060112         NIOS    PTR
29 01567 002407         JMP     @TEMP2  ;AND RETURN
30
31 01570 063610 GTTTI:  SKPDN   TTI
32 01571 000777         JMP     .-1
33 01572 074410         DIA     3,TTI   ;READ AND START
34 01573 060110         NIOS    TTI
35 01574 002402         JMP     @TEMP2  ;AND RETURN
36 01575 000000 TEMP1:  0
37 01576 000000 TEMP2:  0
38 01577 000000 SAVE:   0
39
40                ;TEST BLOCK TYPE
41 01600 125224 BTEST:  MOVZR   1,1,SZR ;1=START BLOCK (.END XX)
42 01601 000411         JMP     IGNOR   ;NO, IGNORE BLOCK
43 01602 101004         MOV     0,0,SZR ;TEST THE CHECKSUM
44 01603 000461         JMP     CHKER   ;ERROR
45 01604 030505         LDA     2,ADDRS ;GET ADDR
46 01605 062677         IORST           ;DO A RESET
47 01606 151113         MOVL#   2,2,SNC ;TEST BIT 0
48 01607 001000         JMP     0,2     ;0=START PROGRAM
49 01610 063077         HALT            ;1=HALT
50 01611 000777         JMP     .-1     ;DON'T PROCEED
51
52                ;IGNORE BLOCK
53 01612 004745 IGNOR:  JSR     GTCHR   ;READ UNTIL AN ALL
54 01613 020404         LDA     0,BC377 ;ONES BYTE IS SEEN
55 01614 116404         SUB     0,3,SZR ;IGNORING ERROR MESS
56 01615 000775         JMP     IGNOR
57 01616 000407         JMP     BLOCK   ;OK, GO INTO BLOCK MODE
58 01617 000377 BC377:  377
59
60                ;START OF PROGRAM BINARY LOADER TS
61 01620 062677 START:  IORST           ;RESET
62 01621 060477         READS   0       ;READ THE SWITCH REGISTER
63 01622 040755         STA     0,SAVE  ;AND SAVE IT FOR GTCHR
64 01623 060110         NIOS TTI        ;START BOTH READERS
65 01624 060112         NIOS PTR
```

```
I 0037  .MAIN
01
02                ;READ IN A BLOCK
03 01625 004732 BLOCK:  JSR     GTCHR   ;GET A BYTE
04 01626 171305         MOVS    3,2,SNR ;AND TEST FOR NUL
05 01627 000776         JMP     BLOCK   ;YES, KEEP READING
06 01630 004727         JSR     GTCHR   ;OK. GET NEXT BYTE
07 01631 173300         ADDS    3,2     ;AND FORM COUNT. AC2:=# OF WORDS
08 01632 141000         MOV     2,0     ;SET CHECKSUM. AC0:=CHECKSUM TILL NOW
09 01633 145000         MOV     2,1     ;SET COUNTER
10 01634 004714         JSR     BUILD   ;GET ADDRESS
11 01635 050454         STA     2,ADDRS
12 01636 004712         JSR     BUILD   ;ADD IN THE CHECKSUM FROM TAPE
13 01637 125113         MOVL#   1,1,SNC ;TEST BLOCK TYPE
14 01640 000740         JMP     BTEST   ;NOT A DATABLOCK
15 01641 044427         STA     1,COUNT ;STORE WORD COUNT
16
17                ;READ IN THE DATA BLOCK
18 01642 030734 DATA:   LDA     2,TEMP2 ;LAST STA IN TEMP2 WAS JSR RETURN
19 01643 034423         LDA     3,DIFF  ;ADDR 4 CELLS AFTER PROGRAM START:
20 01644 172400         SUB     3,2     ;AC2:=FIRST ADDR IN LOADER
21 01645 034444         LDA     3,ADDRS ;ADDR IN WHICH TO STORE
22 01646 136400         SUB     1,3     ;ADD NEG WC TO CHECK SPACE
23 01647 172023         ADCZ    3,2,SNC ;FOR WHOLE BLOCK
24 01650 000414         JMP     CHKER   ;NO, HALT THE LOADER
25 01651 030416         LDA     2,BC20  ;IF WC > 20 (OCTAL, NEG)
26 01652 147033         ADDZ#   2,1,SNC ;IT IS A REPEAT BLOCK
27 01653 010415         ISZ     COUNT   ;WHERE WC IS ONE LESS THAN COUNT
28 01654 147022         ADDZ    2,1,SZC ;IF REPEAT BLOCK SKIP NEXT TO READ DATA
29 01655 125113 STORE:  MOVL#   1,1,SNC ;DON'T READ IN NEW DATA IF REPEAT BLOCK
30 01656 004672         JSR     BUILD
31 01657 052432         STA     2,@ADDRS
32 01660 010431         ISZ     ADDRS   ;NEXT ADDR
33 01661 010407         ISZ     COUNT   ;TEST COUNT
34 01662 000773         JMP     STORE   ;MORE DATA
35 01663 101004         MOV     0,0,SZR ;TEST CHECKSUM
36 01664 063077 CHKER:  HALT            ;ERROR IN CHECKSUM, AC0=VALUE
37 01665 000740         JMP     BLOCK   ;OK,GET NEXT BLOCK
38 01666 000004 DIFF:   4
39 01667 000020 BC20:   20              ;REPEAT BLOCKS HAVE WD>20
40 01670 000000 COUNT:  0
41
42                ;BOOTSTRAP LOADER TS FOR PTR:   ENTER AT BSTRP
43 01671 126440 GET:    SUBO    1,1
44 01672 063612         SKPDN   12      ;10 FOR TTI
45 01673 000777         JMP     .-1
46 01674 060412         DIA     0,12    ;10 FOR TTI
47 01675 060112         NIOS    12      ;10 FOR TTI
48 01676 127100         ADDL    1,1
49 01677 127100         ADCL    1,1
50 01700 107003         ADD     0,1,SNC
51 01701 000771         JMP     GET+1
52 01702 001400         JMP     0,3
53 01703 060112 BSTRP:  NIOS    12      ;10 FOR TTI
54 01704 004765         JSR     GET
55 01705 044402         STA     1,.+2
56 01706 004763         JSR     GET
57 01707 000000         0
58 01710 000000         0               ;FOR BOOTSTRAP
59 01711 000000 ADDRS:  0
60 01712 020706 BEND:   JMP     START   ;START OF BINARY LOADER
```

```
I 0038  .MAIN
01
02                              ;FIND THE TOP OF MEMORY (4K SEGMENTS)
03                              ;STORE HIGHEST USEABLE ADDRESS IN
04                              ;HMEND, PROTECTING THE BIN LOADER.
05                              ;THE CONTENTS IN MEMORY ARE LEFT UNCHANGED.
06                              ;CALL   JSR     GMEND
07                              ;       RETURN
08 01713 010000 DMEND:  10000           ;4K CONSTANT
09 01714 014000 EMEND:  14000           ;6K+1 ADDR
10 01715 004200 FMEND:  004200          ;2K+LOADER CONSTANT
11 01716 054451 GMEND:  STA     3,RSAMS
12 01717 030775         LDA     2,EMEND ;PASS 1 AND 2K SEGMENTS
13 01720 020773         LDA     0,DMEND
14 01721 035000 AMEND:  LDA     3,0,2   ;SAVE LOC CONTENT
15 01722 051000         STA     2,0,2   ;STORE INTO UPPER
16 01723 025000         LDA     1,0,2   ;GET IT BACK
17 01724 146404         SUB     2,1,SZR
18 01725 000405         JMP     BMEND
19 01726 055000         STA     3,0,2   ;RESTORE LOC CONTENT
20 01727 113020         ADDZ    0,2
21 01730 151013 JMEND:  MOV#    2,2,SNC ;CHECK LAST 64K/32K WITH L-SHIFT
22 01731 000770         JMP     AMEND
23 01732 024763 BMEND:  LDA     1,FMEND ;WENT TOO FAR
24 01733 132400         SUB     1,2     ;PROTECT LOADERS
25 01734 050074         STA     2,HMEND ;SET END MEM
26 01735 002432         JMP     @RSAMS
27
28                              ;ROUTINE TO PRINT WHICH STARTADDR SA OF MORE
29                              ;IS USED. INPUT:  SA IN AC1.
30                              ;CALL   CSAMS
31                              ;       RETURN
32 01736 054431 XSAMS:  STA     3,RSAMS
33 01737 044431         STA     1,SSAMS
34 01740 030431         LDA     2,CXLPT ;SET OUTPUT DEVICE ROUTINES
35 01741 052433         STA     2,@XXLPT ;TO WAIT FOR ONLINE DEVICES.
36 01742 030430         LDA     2,CXTTO
37 01743 052432         STA     2,@XXTTO
38 01744 030427         LDA     2,CDIS
39 01745 052431         STA     2,@XFDIS
40 01746 052431         STA     2,@XSDIS
41 01747 006046         CDICL
42 01750 006043         CCRLF
43 01751 024417         LDA     1,SSAMS
44 01752 152520         SUBZL   2,2     ;AC2:=1
45 01753 146400         SUB     2,1     ;SUBTRACT 1 FROM JSR ADDR
46 01754 006056         CDOCT
47 01755 006052         CTOCT           ;PRINT SA
48 01756 006044         CDISP
49 01757 001530         MSAMS
50 01760 006040         CMESS
51 01761 001530         MSAMS           ;XXXXXX STARTADDR
52 01762 006131         SWENB           ;ENABLE PANEL SWITCHES
53 01763 006047         CDATT           ;OUTPUT OF SWITCH SETTINGS OFTEN HERE
54 01764 006040         CMESS           ;PRINT CR,LF WITH MESS TO WAIT
55 01765 004743         MCRLF           ;FOR DEVICE READY BEFORE IORST
56 01766 002401         JMP     @RSAMS
57 01767 000000 RSAMS:  0
58 01770 000000 SSAMS:  0
59 01771 001750 CXLPT:  1000.           ;TIMEOUT CONSTANTS FOR
60 01772 000454 CXTTO:  300.            ;OUTPUT DEVICES
61 01773 000050 CDIS:   40.
62 01774 000441 XXLPT:  XLPTT
63 01775 000445 XXTTO:  XTTOT
64 01776 001125 XFDIS:  FDIST
65 01777 001147 XSDIS:  SDIST
```

```
| 0039  .MAIN
01                          ;TABLE TO BE CHANGED TO CORRECT PRINTER ALPHABET.
02
03          002000 .LOC     2000
04          000010 .RDX     8
05
06                  TABLE:
07
08                  .TXTE?
09 02000 000006 <6><0>
10 02001 000006 <6><0>
11 02002 000006 <6><0>
12 02003 000006 <6><0>
13 02004 000006 <6><0>
14 02005 000006 <6><0>
15 02006 000006 <6><0>
16 02007 000006 <6><0>
17 02010 000006 <6><0>
18 02011 004400 <0><11>
19 02012 005000 <0><12>
20 02013 000006 <6><0>
21 02014 006000 <0><14>
22 02015 106400 <0><15>
23 02016 000006 <6><0>
24 02017 000006 <6><0>
25 02020 000006 <6><0>
26 02021 000006 <6><0>
27 02022 000006 <6><0>
28 02023 000006 <6><0>
29 02024 000006 <6><0>
30 02025 000006 <6><0>
31 02026 000006 <6><0>
32 02027 000006 <6><0>
33 02030 000006 <6><0>
34 02031 000006 <6><0>
35 02032 000006 <6><0>
36 02033 000006 <6><0>
37 02034 000006 <6><0>
38 02035 000006 <6><0>
39 02036 000006 <6><0>
40 02037 000006 <6><0>
41 02040 117400 <0><37>
42 02041 020400 <0><41>
43 02042 021000 <0><42>
44 02043 121400 <0><43>
45 02044 022000 <0><44>
46 02045 122400 <0><45>
47 02046 123000 <0><46>
48 02047 023400 <0><47>
49 02050 024000 <0><50>
50 02051 124400 <0><51>
51 02052 125000 <0><52>
52 02053 025400 <0><53>
53 02054 126000 <0><54>
54 02055 026400 <0><55>
55 02056 027000 <0><56>
56 02057 127400 <0><57>
57 02060 030000 <0><60>
58 02061 130400 <0><61>
59 02062 131000 <0><62>
60 02063 031400 <0><63>
61 02064 132000 <0><64>
62 02065 032400 <0><65>
63 02066 033000 <0><66>
64 02067 133400 <0><67>
65 02070 134000 <0><70>
```

```
1 0040   .MAIN
02 02071 034400 <0><71>
03 02072 035000 <0><72>
04 02073 135400 <0><73>
05 02074 036000 <0><74>
06 02075 136400 <0><75>
07 02076 137000 <0><76>
08 02077 037400 <0><77>
09 02100 140000 <0><100>
10 02101 040400 <0><101>
11 02102 041000 <0><102>
12 02103 141400 <0><103>
13 02104 042000 <0><104>
14 02105 142400 <0><105>
15 02106 143000 <0><106>
16 02107 043400 <0><107>
17 02110 044000 <0><110>
18 02111 144400 <0><111>
19 02112 145000 <0><112>
20 02113 045400 <0><113>
21 02114 146000 <0><114>
22 02115 046400 <0><115>
23 02116 047000 <0><116>
24 02117 147400 <0><117>
25 02120 050000 <0><120>
26 02121 150400 <0><121>
27 02122 151000 <0><122>
28 02123 051400 <0><123>
29 02124 152000 <0><124>
30 02125 052400 <0><125>
31 02126 053000 <0><126>
32 02127 153400 <0><127>
33 02130 154000 <0><130>
34 02131 054400 <0><131>
35 02132 055000 <0><132>
36 02133 155400 <0><133>
37 02134 056000 <0><134>
38 02135 156400 <0><135>
39 02136 157000 <0><136>
40 02137 057400 <0><137>
41 02140 006000 <0><14>
42 02141 040400 <0><101>
43 02142 041000 <0><102>
44 02143 141400 <0><103>
45 02144 042000 <0><104>
46 02145 142400 <0><105>
47 02146 143000 <0><106>
48 02147 043400 <0><107>
49 02150 044000 <0><110>
50 02151 144400 <0><111>
51 02152 145000 <0><112>
52 02153 045400 <0><113>
53 02154 146000 <0><114>
54 02155 046400 <0><115>
55 02156 047000 <0><116>
56 02157 147400 <0><117>
57 02160 050000 <0><120>
58 02161 150400 <0><121>
59 02162 151000 <0><122>
60 02163 051400 <0><123>
61 02164 152000 <0><124>
62 02165 052400 <0><125>
63 02166 053000 <0><126>
64 02167 153400 <0><127>
65 02170 154000 <0><130>
```

```
| 0041   .MAIN
01 02171 054400 <0><131>
02 02172 055000 <0><132>
03 02173 140000 <0><100>
04 02174 121400 <0><43>
05 02175 156400 <0><135>
06 02176 004400 <0><11>
07 02177 000006 <6><0>?
   02200 000000
08       000010 .RDX 8
09                              ;START ADDR FOR "HELP" PROGRAMS
10       002201 .LOC 2201
11
12 02201 002403          SAPTB
13 02202 006777          JSR     @.-1    ;START MOVE PRINTER TABLE
14 02203 002260          SAMEX
15 02204 006777          JSR     @.-1    ;START SET TO 64K MODE, MEM SIZE ?
16 02205 002265          SAMNM
17 02206 006777          JSR     @.-1    ;START SET TO 32K MODE, MEM SIZE ?
18 02207 005011          EXMEM
19 02210 006777          JSR     @.-1    ;START EXAMINE MEMORY
20 02211 005127          DPMEM
21 02212 006777          JSR     @.-1    ;START DEPOSIT MEMORY
22 02213 007077          TROHA
23 02214 006777          JSR     @.-1    ;START TROUBLE BREAKPOINT HALT
24 02215 007111          TROLO
25 02216 006777          JSR     @.-1    ;START TROUBLE BREAKPOINT LOOP REPORT
26 02217 007123          TRORE
27 02220 006777          JSR     @.-1    ;START TROUBLE BREAKPOINT RESET
28 02221 002314          LOADB
29 02222 006777          JSR     @.-1    ;START BINARY LOADER, READ FROM PTR/TTI
30
31                MLPTT: .TXT !LPT TABLE!        ;"LPT TABLE"
   02223 050114
   02224 020124
   02225 040524
   02226 046102
   02227 000105
32                       ;ROUTINE TO MOVE PRINTER TABLE.
33                       ;INPUT: TABLE # IN AC2
34                       ;CALL:  JSR     MOPTB
35
36 02230 054415 MOPTB:   STA     3,MOPTR ;SAVE RETIRE
37 02231 034417          LDA     3,MOGTT ;ADDR OF POINTER
38 02232 157000          ADD     2,3     ;AC3:=TABLE ENTRY
39 02233 031400          LDA     2,0,3   ;AC2:=ADDR OF TABLE, START
40 02234 024413          LDA     1,MOLAD ;LAST STORE ADDR
41 02235 034411          LDA     3,MOFID ;FIRST STORE ADDR
42 02236 021000 MOREP:   LDA     0,0,2
43 02237 041400          STA     0,0,3   ;MOVE CHAR
44 02240 166415          SUB#    3,1,SNR ;LAST CHAR ?
45 02241 002404          JMP     @MOPTR  ;YES, RETIRE
46 02242 151400          INC     2,2
47 02243 175400          INC     3,3     ;NEXT CHAR ADDR
48 02244 000772          JMP     MOREP   ;REPEAT MOVE CHAR
49 02245 000000 MOPTR:   0               ;RETURN ADDR
50 02246 002000 MOFID:   2000            ;FIRST STORE ADDR
51 02247 002200 MOLAD:   2200            ;LAST STORE ADDR
52 02250 002250 MOGTT:   .               ;GET TABLE ADDR
53 02251 003656          PTAB1           ;ASCII TABLE ADDR
54 02252 004057          PTAB2           ;RC STANDARD 71/78 START .
55 02253 004260          PTAB3           ;RC STANDARD 71/78 START 0
56 02254 004461          PTAB4           ;PL1 TYPE 70 TABLE ADDR
57 02255 000004 ALPTT:   4               ;SUGGESTED ANSWER TABLE #
58 02256 000004 ULPTT:   4               ;UPPER LIMIT
59 02257 000001 LLPTT:   1               ;LOWER LIMIT
```

```
 0042  .MAIN
01
02                        ;PROGRAMS TO SET MAX MEMORY LOCATION. THEY WILL SET
03                        ;MAX MEM LOC IN HMEND BOTH FOR 32K AND 64K MAX MODE.
04                        ;BUT 64K MAX (MEMORY EXTENSION) MODE WILL BE RESET
05                        ;AFTER EACH ICRST. THEREFORE IF THE PROGRAMS SHOULD
06                        ;USE THIS MODE IT SHOULD BE MADE TO CHECK HMEND TO SEE
07                        ;IF IT IS BIGGER THAN 32K. IF SO YOU SHOULD FIRE THE
08                        ;DICP 0,1 COMMAND BEFORE USING MEMORY LOC BEYOND 32 K.
09
10                        ;PROGRAM TO SET MAX MEM LOC TO 64K WORDS MODE.
11
12 02260 165000 SAMEX:    MOV     3,1
13 02261 006072           CSAMS             ;START ADDR MESSAGE
14 02262 004456           JSR     TMEND     ;TRY SET MEM EXT FLAG (RC3603 ONLY)
15 02263 000411           JMP     SAMMS
16 02264 151013 MCMEX:    MOV#    2,2,SNC
17
18                        ;PROGRAM TO SET MAX MEM LOC TO 32K WORDS MODE.
19 02265 165000 SAMNM:    MOV     3,1
20 02266 006072           CSAMS             ;START ADDR MESSAGE
21 02267 024404           LDA     1,MCMNM
22 02270 046423           STA     1,@MIMEX
23 02271 062677           ICRST             ;RESET MEM EXT FLAG (RC3603 ONLY)
24 02272 000402           JMP     SAMMS
25 02273 151113 MCMNM:    MOVL#   2,2,SNC
26 02274 006416 SAMMS:    JSR     @IMEND    ;AC2=HMEND=XX7600
27 02275 006044           CDISP
28 02276 001501           MLLCC
29 02277 006040           CMESS
30 02300 001501           MLLOC             ;<15><12>LAST LOC. XX7777
31 02301 020074           LDA     0,HMEND
32 02302 026407           LDA     1,@SAMCO  ;AC1:=177
33 02303 107000           ADD     0,1
34 02304 006056           CDOCT
35 02305 006052           CTOCT
36 02306 006047           CDATT
37 02307 002401           JMP     @.+1      ;PROGRAM FINISHED
38 02310 004740           SWISA             ;RESTART MAIN PROGRAM.
39 02311 001376 SAMCO:    BINLA
40 02312 001716 IMEND:    GMEND
41 02313 001730 MIMEX:    JMEND
42
43                        ;ROUTINE RESTORE BINARY LOADER AND
44                        ;START LOADING PTR/TTI DEPENDING ON SWITCH 0.
45
46 02314 165000 LOADB:    MOV     3,1
47 02315 006072           CSAMS             ;START ADDR MESSAGE
48 02316 024755           LDA     1,MCMNM
49 02317 046774           STA     1,@MIMEX  ;MAX 32K MODE
50 02320 062677           ICRST             ;RESET MEM EXT FLAG
51 02321 006771           JSR     @IMEND    ;AC2=HMEND=0X7600
52 02322 026767           LDA     1,@SAMCO  ;AC1:=177
53 02323 147000           ADD     2,1       ;LAST LOC = 0X7777
54 02324 036412           LDA     3,@LOADF  ;FIRST ADDR LOADER
55 02325 137400           AND     1,3       ;IN THIS MEM
56 02326 032411           LDA     2,@LOADA  ;ADDR OF BINARY DATA
57 02327 021000 LOADR:    LDA     0,0,2
58 02330 041400           STA     0,0,3     ;MOVE DATA
59 02331 166415           SUB#    3,1,SNR   ;LAST DATA ?
60 02332 001400           JMP     0,3       ;YES, START BINARY LOADER
61 02333 151400           INC     2,2
62 02334 175400           INC     3,3       ;NEXT DATA ADDR
63 02335 000772           JMP     LOADR
64 02336 001375 LOADF:    BINF1             ;FIRST ADDR BIN LOADER (32K)
65 02337 001547 LOADA:    BINAD             ;ADDR POINTER TO BIN DATA
```

```
I 0043  .MAIN
01
02                          ;ROUTINE TO CHECK FOR MORE THAN 32K WORDS MEMORY.
03                          ;(MEMORY EXTEND OPTION).
04
05 02340 054435 TMEND:  STA     3,RMEND
06 02341 062677         IORST               ;RESET MEM EXT FLAG
07 02342 062701         DICP    0,1         ;TRY TO SET MEM EXT FLAG (RC3603 ONLY).
08 02343 030433         LDA     2,OMEND     ;DON'T USE SKPDN 1 TO CHECK FLAG
09 02344 034433         LDA     3,PMEND     ;AS NOT IMPLEMENTED IN ALL CPU'S
10 02345 021000         LDA     0,0,2       ;SAVE LOWER LOC CONTENT
11 02346 040432         STA     0,QMEND
12 02347 051000         STA     2,0,2       ;STORE LOWER ADDR
13 02350 025400         LDA     1,0,3       ;SAVE UPPER LOC CONTENT
14 02351 044430         STA     1,SMEND
15 02352 055400         STA     3,0,3       ;STORE UPPER ADDR
16 02353 021400         LDA     0,0,3       ;GET IT BACK
17 02354 116414         SUB#    0,3,SZR
18 02355 000411         JMP     UMEND       ;NOT MORE THAN 32K BUT MEM EXT OPTION
19 02356 025000         LDA     1,0,2       ;MORE THAN 32K OR NOT MEM EXT OPTION
20 02357 132415         SUB#    1,2,SNR     ;UPPER STORED IN LOWER (15 BIT ADDR) ?
21 02360 000406         JMP     UMEND       ;NO, MEM EXT OPTION
22 02361 020417         LDA     0,QMEND     ;YES, NOT MEM EXT OPTION, MAX 32K
23 02362 041000         STA     0,0,2       ;RESTORE LOWER LOC CONTENT
24 02363 024710         LDA     1,MCMNM     ;SET 32K MAX
25 02364 046727         STA     1,@MIMEX
26 02365 002410         JMP     @RMEND      ;EXIT
27 02366 020412 UMEND:  LDA     0,QMEND
28 02367 041000         STA     0,0,2       ;RESTORE LOWER LOC CONTENT
29 02370 024411         LDA     1,SMEND
30 02371 045400         STA     1,0,3       ;RESTORE UPPER LOC CONTENT
31 02372 030672         LDA     2,MCMEX     ;SET 64K MAX
32 02373 052720         STA     2,@MIMEX
33 02374 002401         JMP     @RMEND      ;EXIT
34 02375 000000 RMEND:  0                   ;RETURN ADDR
35 02376 002402 OMEND:  VMEND               ;LOWER ADDR
36 02377 102402 PMEND:  @VMEND              ;UPPER ADDR 16 BIT, NOT INDIRECT
37 02400 000000 QMEND:  0                   ;SAVE LOWER CONTENT
38 02401 000000 SMEND:  0                   ;SAVE UPPER CONTENT
39 02402 000000 VMEND:  0                   ;WORK CELL LOWER ADDR
40
41                          ;PROGRAM TO GET NEW PRINTER TABLE
42
43 02403 165000 SAPTB:  MOV     3,1
44 02404 006072         CSAMS               ;START ADDR MESSAGE
45 02405 006071 RLPTT:  CQUES
46 02406 002223         MLPTT               ;LPT TABLE
47 02407 002223         MLPTT
48 02410 002255         ALPTT               ;SUGGESTED ANSWER
49 02411 006053         CTDEC
50 02412 006057         CDDEC
51 02413 006105         CGTDC               ;READ ANSWER
52 02414 000402         JMP     .+2         ;SUGGESTED ACCEPTED
53 02415 000770         JMP     RLPTT       ;ERROR RETURN
54 02416 030075         LDA     2,DIGIN     ;ANSWER INPUT'ED
55 02417 024637         LDA     1,ULPTT     ;UPPER LIMIT
56 02420 020637         LDA     0,LLPTT     ;LOWER LIMIT
57 02421 132033         ADCZ#   1,2,SNC
58 02422 142032         ADCZ#   2,0,SZC     ;AC0=<AC2=<AC1
59 02423 000762         JMP     RLPTT       ;OUTSIDE LIMITS
60 02424 004604         JSR     MOPTB       ;INPUT ACCEPTED, MOVE TABLE
61 02425 002401         JMP     @.+1        ;PROGRAM FINISHED
62 02426 004740         SWISA               ;RESTART MAIN PROGRAM
```

```
! 0044   .MAIN
01
02                          .EOT
```

```
  0045  .MAIN
01
02                        ;DELAY SUBROUTINE
03                        ;ARGUMENT FOLLOWING CALL IS ADDRESS
04                        ;OF DELAY CONSTANT.
05                        ;DELAY IN INCREMENTS OF 1 MSEC
06                        ;AC2 & 3 ARE USED, DEFINE WAIT, ARG
07                        ;
08                        ;CALL   CWAIT
09                        ;       ARG
10                        ;
11 02427 054416 XWAIT:    STA     3,WARET ;SAVE RETURN
12 02430 010415           ISZ     WARET   ;PASS ARG
13 02431 033400           LDA     2,@0,3  ;FETCH ARG
14 02432 050414           STA     2,WIRET ;STORE ARG
15 02433 030414           LDA     2,KINDI ;FETCH KNOVA ADDRESS
16 02434 036537           LDA     3,@XCPN ;FETCH CPUNO
17 02435 157000           ADD     2,3     ;COMPUTE KADDRESS
18 02436 031400           LDA     2,0,3   ;FETCH CPU CONSTANT
19 02437 050437 NWAIT:    STA     2,KINC  ;STORE IT FOR INC OF 1 MSEC
20 02440 014436           DSZ     KINC    ;X USEC LOOP
21 02441 000777           JMP     .-1     ;X USEC
22 02442 014404           DSZ     WIRET   ;NO. OF MS STILL TO WAIT
23 02443 000774           JMP     NWAIT   ;COUNT ARG NOT ENDED
24 02444 002401           JMP     @WARET  ;COUNT ARG ENDED, RETURN
25 02445 000000 WARET:    0
26 02446 000000 WIRET:    0
27
28 02447 002450 KINDI:    .+1             ;ADDRESS OF KNOVA
29       000012           .RDX 10
30 02450 000175 KCP0:     125             ;NOVA
31 02451 000001 KCP1:     1               ;
32 02452 000341 KCP2:     225             ;1200
33 02453 000567 KCP3:     375             ;SUPER
34 02454 000505 KCP4:     325             ;SUPER SC/830
35 02455 000536 KCP5:     350             ;800/NOVA 2-16K
36 02456 000620 KCP6:     400             ;NOVA 2-8K
37 02457 000001 KCP7:     1               ;CONSTANTS ABOVE ARE FOR GUESSED CPU TYPE
38 02460 000001 KCP10:    1               ;CONSTANTS BELOW ARE EXACTLY.
39 02461 000200 KCP11:    128             ;NOVA    7,8 USEC
40 02462 000336 KCP12:    222             ;1200    4,5 USEC
41 02463 000601 KCP13:    385             ;800     2,6 USEC
42 02464 000601 KCP14:    385             ;SUPER   2,6 USEC
43 02465 000764 KCP15:    500             ;SUPER SC  2,0 USEC
44 02466 000620 KCP16:    400             ;NOVA2/8K  2,5 USEC
45 02467 000502 KCP17:    322             ;NOVA2/16K 3,1 USEC
46 02470 000531 KCP20:    345             ;RC3603/RC3609           2,90 USEC
47 02471 000334 KCP21:    220             ;RC3603/RC3609/BREAK ON 4,55 USEC
48 02472 000463 KCP22:    307             ;RC3603/RC3608           3,26 USEC
49 02473 000314 KCP23:    204             ;RC3603/RC3608/BREAK ON 4,90 USEC
50 02474 000515 KCP24:    333             ;NOVA2/DATARAM 900 NSEC 3,00 USEC
51 02475 000001 KCP25:    1               ;FOR TROUBLE, PUT HERE RELEVANT VALUE.
52       000010           .RDX 8
53 02476 000000 KINC:     0
```

```
1 0046   .MAIN
01
02
03                          ;PROCEDURE TIMER ON SKP
04                          ;CALL:  TIMSK
05                          ;       MSEC (>0) TO WAIT MAX FOR
06                          ;       SKP INSTR.
07                          ;       TIME OUT RETURN
08                          ;       NORMAL RETURN
09
10 02477 025400 RTIME:  LDA     1,0,3   ;AC1:=TIME
11 02500 125015         MOV#    1,1,SNR ;IF TIME = ZERO
12 02501 125400         INC     1,1     ;GIVE IT A CHANGE
13 02502 021401         LDA     0,1,3   ;AC0:=INSTR
14 02503 040410         STA     0,STSKP ;STORE INSTR
15 02504 124400         NEG     1,1     ;AC1:=-TIME
16 02505 020417         LDA     0,CPU00 ;FETCH CPU0-ADDR.
17 02506 032465         LDA     2,@XCPN ;FETCH CPUNO
18 02507 113000         ADD     0,2     ;COMPUTE CPU-ADDR.
19 02510 021000         LDA     0,0,2   ;FETCH CPU-CONSTANT
20 02511 040412         STA     0,CPUINC;STORE IT FOR INC OF 1MS
21 02512 020411 STINC:  LDA     0,CPUINC;# OF LOOPS FOR 1 MSEC
22 02513 000000 STSKP:  0               ; X USEC (SKP INSTR.)
23 02514 000402         JMP     .+2     ; X USEC
24 02515 001403         JMP     3,3     ;NORMAL RETURN
25 02516 101404         INC     0,0,SZR ; X USEC
26 02517 000774         JMP     STSKP   ; X USEC
27 02520 125404         INC     1,1,SZR ;COUNT # OF MSEC
28 02521 000771         JMP     STINC   ;MORE MSEC
29 02522 001402         JMP     2,3     ;TIME OUT RETURN, AC0=AC1=0
30
31 02523 000000 CPUINC: 0
32 02524 002525 CPU00:  .+1             ;ADDR. OF CPU0
33       000012         .RDX 10
34 02525 177677 CPU0:   -65             ;NOVA
35 02526 177777 CPU1:   -1
36 02527 177552 CPU2:   -150            ;1200
37 02530 177470 CPU3:   -200            ;SUPER
38 02531 177437 CPU4:   -225            ;SUPER SC/830
39 02532 177406 CPU5:   -250            ;800/NOVA 2-16K
40 02533 177324 CPU6:   -300            ;NOVA 2-8K
41 02534 177777 CPU7:   -1              ;
42 02535 177777 CPU10:  -1              ;
43 02536 177676 CPU11:  -66             ;NOVA    15,2 USEC
44 02537 177550 CPU12:  -152            ;1200     6,6 USEC
45 02540 177371 CPU13:  -263            ;800      3,8 USEC
46 02541 177504 CPU14:  -188            ;SUPER    5,3 USEC
47 02542 177427 CPU15:  -233            ;SUPER SC  4,3 USEC
48 02543 177343 CPU16:  -285            ;NOVA2/8K  3,5 USEC
49 02544 177422 CPU17:  -238            ;NOVA2/16K 4,2 USEC
50 02545 177371 CPU20:  -263            ;RC3603/RC3609           3,80 USEC
51 02546 177565 CPU21:  -139            ;RC3603/RC3609/BREAK ON 7,20 USEC
52 02547 177416 CPU22:  -242            ;RC3603/RC3608           4,14 USEC
53 02550 177571 CPU23:  -135            ;RC3603/RC3608/BREAK ON 7,40 USEC
54 02551 177371 CPU24:  -263            ;NOVA2/DATARAM 900 NSEC 3,80 USEC
55 02552 177777 CPU25:  -1              ;FOR TROUBLE, PUT HERE RELEVANT VALUE.
56       000010         .RDX 8
```

```
| 0047  ,MAIN
01
02                        ;ROUTINE TO MESSURE TIME.
03                        ;CALL    TIMMS
04                        ;        SKP INSTR. TO BE MESSURED
05                        ;        TIMEOUTRETURN    TIMEM:=MAX TIME
06                        ;        NORMAL RETURN    TIMEM:=MESSURED TIME
07                        ;THE TIME IS MESSURED IN TENS OF USEC.
08                        ;LOOPS IN THE INSTRUCTIONS MARKED X USEC UNTIL
09                        ;BIT 0 IS SET (TIMEOUT, MAX MESSUREABLE TIME)
10                        ;COUNTING FROM -1 IN THE INC INSTRUCTION OR UNTIL
11                        ;SKP INSTRUCTION TO BE MESSURED IS EFFECTIVE.
12                        ;COULD BE EXTENDED TO TWICE THE TIME POSSIBLE HERE
13                        ;IF CARRY IS USED INSTEAD OF BIT 0, BUT NO PRINT ROUT.
14
15 02553 054477 MSTIM:    STA      3,MSTIR ;SAVE RETURN
16 02554 021400           LDA      0,0,3   ;GET SKP INSTR.
17 02555 040405           STA      0,MSTIB ;TO BE MESSURED.
18 02556 102040           ADCO     0,0     ;AC0:=177777, C:=1
19 02557 101400 MSTIA:    INC      0,0     ;TO AVOID OVERFLOW       X USEC
20 02560 101112           MOVL#    0,0,SZC ;INCREASE LOOP TIME      X USEC
21 02561 000443           JMP      MSTIO   ;TIME OUT, BIT 0=1
22 02562 000000 MSTIB:    0                ;SKP INSTR. TO-          X USEC
23 02563 000774           JMP      MSTIA   ;BE TIMED.               X USEC
24 02564 101400           INC      0,0     ;COMPENSATE INITIALIZATION 4 INSTR.
25 02565 101400           INC      0,0     ;COMPENSATE STARTING AT -1
26 02566 040402           STA      0,TIMEX ;TIMECOUNT MESSURED
27 02567 000441           JMP      MSTID   ;CALCULATE, BIT 0=0
28
29 02570 000000 TIMEX:    0                ;TIMECOUNT USED ABOVE
30 02571 000000 TIMCT:    0                ;TIMECONSTANT FROM TABLE BELOW
31 02572 000000 TIMEM:    0                ;TIME MESSURED
32 02573 003015 XCPN:     CPUNO
33
34 02574 002575 TCP00:    .+1              ;100 MSEC CONSTANTS:
35        000012          .RDX 10
36 02575 012574 TCP0:     5500             ;NOVA
37 02576 077777 TCP1:     32767            ;        WILL GIVE TIMEOUT = 100 MSEC
38 02577 030324 TCP2:     12500            ;1200
39 02600 040164 TCP3:     16500            ;SUPER
40 02601 051774 TCP4:     21500            ;SUPER SC/830
41 02602 055714 TCP5:     23500            ;800/NOVA2-16K
42 02603 063604 TCP6:     26500            ;NOVA2-8K
43 02604 077777 TCP7:     32767            ;
44 02605 077777 TCP10:    32767            ;                X USEC:
45 02606 012566 TCP11:    5494             ;NOVA            18,2 USEC
46 02607 030442 TCP12:    12578            ;1200             7,95 USEC
47 02610 060650 TCP13:    25000            ;800              4,0 USEC
48 02611 040011 TCP14:    16393            ;SUPER            6,1 USEC
49 02612 055327 TCP15:    23255            ;SUPER SC         4,6 USEC
50 02613 063314 TCP16:    26316            ;NOVA 2 - 8K      3,8 USEC
51 02614 054307 TCP17:    22727            ;NOVA 2 - 16K     4,4 SEC
52 02615 052001 TCP20:    21505            ;RC3603/RC3609           4,65 USEC
53 02616 030206 TCP21:    12422            ;RC3603/RC3609/BREAK ON 8,05 USEC
54 02617 047545 TCP22:    20325            ;RC3603/RC3608           4,92 USEC
55 02620 027531 TCP23:    12121            ;RC3603/RC3608/BREAK ON 8,25 USEC
56 02621 060650 TCP24:    25000            ;NOVA2/DATARAM 900 NSEC 4,00 USEC
57 02622 077777 TCP25:    32767            ;FOR TROUBLE, PUT HERE RELEVANT VALUE.
58
59 02623 023420 MST10:    10000
60        000010          .RDX 8
61
62 02624 102220 MSTIC:    ADCZR    0,0     ;AC0:=077777, CARRY:=1
63 02625 040743           STA      0,TIMEX ;TIMEOUT TIMECOUNT = 32767
64 02626 000402           JMP      MSTID   ;CALCULATE
```

```
I 0048  .MAIN
01
02 02627 000000 MSTIC:  0                ;CARRY FLAG
03 02630 101200 MSTID:  MOVR     0,0     ;
04 02631 040776         STA      0,MSTIC ;SAVE CARRY
05 02632 026741         LDA      1,@XCPN ;
06 02633 030741         LDA      2,TCP00 ;
07 02634 133000         ADD      1,2     ;
08 02635 025000         LDA      1,0,2   ;FETCH CPU CONSTANT
09 02636 044733         STA      1,TIMCT ;
10 02637 024731         LDA      1,TIMEX ;
11 02640 030763         LDA      2,MST10 ;
12 02641 006066         MULTI            ;TIMECOUNT X 10000
13 02642 030727         LDA      2,TIMCT ;
14 02643 006070         DIVID            ;DIVIDED BY CPU CONSTANT
15 02644 044726         STA      1,TIMEM ;TIME IN TENS OF USEC
16 02645 020762         LDA      0,MSTIC ;GET CARRY FLAG
17 02646 010404         ISZ      MSTIR   ;PASS SKP ARG.
18 02647 101103         MOVL     0,0,SNC ;IF CARRY THEN TIMEOUT
19 02650 010402         ISZ      MSTIR   ;PASS TIMEOUT RETURN
20 02651 002401         JMP      @MSTIR  ;RETURN
21 02652 000000 MSTIR:  0                ;RETURN ADDR.
22
23                      ;ROUTINE TO MULTIPLY
24                      ;CALL    MULTI
25                      ;        RETURN
26                      ;
27                      ;        AC0, AC1:=AC1*AC2
28                      ;
29 02653 102460 XMULT:  SUBC     0,0     ;MULTIPLY C(1)*(2)
30 02654 054425         STA      3,MSAV  ;RESULT TO C(0),(1)
31 02655 034423         LDA      3,MDCTR ;AC2 UNCHANGED
32 02656 125203 MLOOP:  MOVR     1,1,SNC ;CARRY UNCHANGED
33 02657 101201         MOVR     0,0,SKP
34 02660 143220         ADDZR    2,0
35 02661 175404         INC      3,3,SZR
36 02662 000774         JMP      MLOOP
37 02663 125260         MOVCR    1,1
38 02664 002415         JMP      @MSAV
39
40                      ;ROUTINES TO DIVIDE
41                      ;CALL    DIVIS
42                      ;        RETURN
43                      ;
44                      ;        AC0:=REMAINDER
45                      ;        AC1:=QUOTIENT FOR AC1/AC2
46                      ;
47                      ;CALL    DIVID
48                      ;        RETURN
49                      ;
50                      ;        AC0:=REMAINDER
51                      ;        AC1:=QUOTIENT FOR AC0, AC1/AC2
52                      ;
53 02665 102400 XDIVS:  SUB      0,0     ;DIVIDE      C(1)/C(2)
54 02666 054413 XDIVD:  STA      3,MSAV  ;DIVIDE C(0),C(1)/C(2)
55 02667 034411         LDA      3,MDCTR ;C(0)=REMAINDER
56 02670 125120         MOVZL    1,1     ;C(1)=QUOTIENT
57 02671 101100 DLOOP:  MOVL     0,0     ;AC2 UNCHANGED
58 02672 142412         SUB#     2,0,SZC ;CARRY...?
59 02673 142400         SUB      2,0
60 02674 125100         MOVL     1,1
61 02675 175404         INC      3,3,SZR
62 02676 000773         JMP      DLOOP
63 02677 002402         JMP      @MSAV
64 02700 177760 MDCTR:  -20
65 02701 000000 MSAV:   0
```

```
| 0049  .MAIN
01
02                          ;SUBROUTINE TIMER ON ROUTINE
03                          ;
04                          ;CALL    TIMRO
05                          ;        ADDR. OF TIME TO WAIT MAX (MSEC)
06                          ;        JSR @PAGE ZERO (DEFINITION)      OR JMP .+1
07                          ;        ARGUMENT FOR JSR INSTR.          OR SKP INSTR.
08                          ;        TIMEOUT RETURN
09                          ;        EVENT RETURN
10
11 02702 054430 XTIMS:  STA      3,XTIMR ;
12 02703 033400         LDA      2,@0,3  ;GET TIME, # OF MSEC
13 02704 050425         STA      2,XTIMC ;TO COUNT
14 02705 010425         ISZ      XTIMR
15 02706 032424         LDA      2,@XTIMR ;GET CALL DEFINITION
16 02707 050406         STA      2,XTIMD ;TO USE FOR TIMER
17 02710 010422         ISZ      XTIMR
18 02711 032421         LDA      2,@XTIMR ;GET ARGUMENT
19 02712 050404         STA      2,XTIMA ;FOR CALLED ROUTINE
20 02713 010417         ISZ      XTIMR    ;COUNT RETURN ADDR TO PASS ARG.
21 02714 010415         ISZ      XTIMC    ;COUNT TIME FOR CORRECT USE
22 02715 000401 XTIMD:  JMP      .+1      ;JSR    XX
23 02716 000401 XTIMA:  JMP      .+1      ;ARGUMENT YY
24 02717 000403         JMP      XTIMT    ;NOT YET RETURN, TEST TIMER
25 02720 010412         ISZ      XTIMR    ;EVENT RETURN, COUNT RETURN ADDR.
26 02721 002411         JMP      @XTIMR   ;TO PASS ERROR RETURN.
27 02722 014407 XTIMT:  DSZ      XTIMC    ;TIMECOUNT STILL NOT FINISHED ?
28 02723 000402         JMP      XTIMW    ;STALL 1 MSEC
29 02724 002406         JMP      @XTIMR   ;TIMEOUT RETURN
30 02725 006061 XTIMW:  CWAIT
31 02726 002730         XTIM1
32 02727 000766         JMP      XTIMD    ;TEST EVENT
33 02730 000001 XTIM1:  1                 ;1 MSEC CONSTANT
34 02731 000000 XTIMC:  0                 ;X MSEC COUNTER
35 02732 000000 XTIMR:  0                 ;RETURN ADDR.
36
37              UNTST:  .TXT !, RUN RC 3600 INSTRUCTION TIMER TEST!
   02733 020054
   02734 052522
   02735 020116
   02736 041522
   02737 031440
   02740 030066
   02741 020060
   02742 047111
   02743 052123
   02744 052522
   02745 052103
   02746 047511
   02747 020116
   02750 044524
   02751 042515
   02752 020122
   02753 042524
   02754 052123
   02755 000000
38                          ;", RUN RC 3600 INSTRUCTION TIMER TEST"
```

```
I 0050  .MAIN
01
02                      ;ROUTINE TO FIND TYPE OF CPU.
03                      ;CALLED BY REBIN
04                      ;THIS ROUTINE IS SETTING A NUMBER INDICATING
05                      ;TYPE AND SPEED OF CPU INTO CPUNO. SEE
06                      ;ROUTINE TO MESSURE TIME FOR DEFINITION OF #.
07
08 02756 054432 FITYP:  STA     3,RETYP
09 02757 062c77         IORST
10 02760 020434         LDA     0,NUTYP ;NUMBER OF TRYING AGAIN
11 02761 040432         STA     0,AGTYP
12 02762 004447 TRTYP:  JSR     TYMER
13 02763 125214         MOV#    1,1,SZR ;AC1=0, SEARCH MORE
14 02764 000422         JMP     STTYP
15 02765 006424         JSR     @XOMER
16 02766 125014         MOV#    1,1,SZR ;AC1=0, SEARCH MORE
17 02767 000417         JMP     STTYP
18 02770 014423         DSZ     AGTYP   ;COUNT DOWN # OF
19 02771 000771         JMP     TRTYP   ;RETRY TO IDENTIFY CPU
20 02772 006420         JSR     @XWTYP
21 02773 044422         STA     1,CPUNO
22 02774 006044         CDISP
23 02775 001536         UNTIM
24 02776 006040         CMESS
25 02777 001536         UNTIM           ;MISERABLE TIMING
26 03000 006040         CMESS
27 03001 002733         UNTST           ;RUN INSTR TIMER TEST
28 03002 006043         CCRLF
29 03003 006061         CWAIT           ;WAIT 3 SEC TO READ MESS. DON'T
30 03004 001236         SEC3            ;USE CDATT AS SW AREN'T SET.
31 03005 002403         JMP     @RETYP  ;EXIT
32 03006 044407 STTYP:  STA     1,CPUNO
33 03007 002401         JMP     @RETYP  ;EXIT
34 03010 000000 RETYP:  0
35 03011 003275 XOMER:  TOMER
36 03012 003506 XWTYP:  NWTYP
37 03013 000000 AGTYP:  0
38 03014 000007 NUTYP:  7               ;# OF CALLS OF TYMER, TOMER.
39 03015 000006 CPUNO:  6               ;TYPE OF CPU, NOVA 2-8K SHOWN
40
41 03016 054772 PRTYP:  STA     3,RETYP ;PRINT CPU TYPE NUMBER
42 03017 006044         CDISP
43 03020 005002         MCPUT
44 03021 006040         CMESS
45 03022 005002         MCPUT           ;<15><12>CPU TYPE:
46 03023 024772         LDA     1,CPUNO
47 03024 006052         CTOCT
48 03025 006056         CDOCT
49 03026 006061         CWAIT           ;WAIT 3 SEC TO READ MESS. DON'T
50 03027 001236         SEC3            ;USE CDATT AS SW AREN'T SET.
51 03030 002760         JMP     @RETYP  ;EXIT
```

```
1 0051  .MAIN
01
02 03031 054572 TYMER:  STA     3,SVTYME
03 03032 020576         LDA     0,XTYME ;GET ADDRESS OF TYME LIST.
04 03033 040021         STA     0,IDX1  ;STORE IN AUTO POINTER.
05 03034 020575         LDA     0,XINST ;GET ADDR OF INSTRUCTION LIST
06 03035 040020         STA     0,IDX0  ;STORE IN AUTO POINTER.
07 03036 030567         LDA     2,INXW5
08 03037 050023         STA     2,IDX3  ;INITIALIZE IDX3 FOR LDA INST.
09 03040 152440         SUBO    2,2     ;AC2:=0, NULL CHAR
10 03041 071011         DOA     2,XTTO  ;SEND CHAR
11 03042 060111         NIOS    XTTO    ;START TTO AND
12 03043 063511         SKPBZ   XTTO    ;SYNCHRONIZE
13 03044 000777         JMP     .-1     ;WITH TTO CLOCK
14 03045 071011         DOA     2,XTTO  ;SEND A CHAR
15 03046 060111         NIOS    XTTO    ;AND MESSURE TIME, NOT ACCURATE
16 03047 006064         TIMMS           ;CPU TYPE NOT KNOWN BUT TO GIVE AN IDEA
17 03050 063511         SKPBZ   XTTO    ;OF TTO SPEED
18 03051 000401         JMP     .+1     ;AC0 IS MESSURED TIME
19 03052 022544         LDA     0,@TYMEM
20 03053 024544         LDA     1,TYLIM ;LIMIT TO DESTINGV. SPEED
21 03054 034544         LDA     3,TYTTS ;SLOW TTY CONSTANT
22 03055 106432         SUBZ#   0,1,SZC ;IS TTY FAST ?
23 03056 034543         LDA     3,TYTTF ;YES, FAST TTY CONSTANT
24 03057 054535         STA     3,TYMEN ;STORE # OF COUNTS
25 03060 152440         SUBO    2,2     ;AC2:=0, NULL CHAR
26 03061 071011         DOA     2,XTTO  ;SEND CHAR
27 03062 060111         NIOS    XTTO    ;START TTO AND
28 03063 063511         SKPBZ   XTTO    ;SYNCHRONIZE PROGRAM
29 03064 000777         JMP     .-1     ;WITH TTY CLOCK.
30 03065 152440 TYMA:   SUBO    2,2     ;CLEAR AC2
31 03066 141000         MOV     2,0     ;CLEAR AC0 ALSO.
32 03067 026020         LDA     1,@IDX0 ;GET INST. FROM LIST
33 03070 125015         MOV#    1,1,SNR ;ZERO MARKS END OF INSTR. LIST
34 03071 000455         JMP     SCORE   ;
35 03072 071011         DOA     2,XTTO  ;SEND NULL CHAR, STARTING LATER
36 03073 034451         LDA     3,CSKP  ;GET ALC-SKP MASK AND
37 03074 137400         AND     1,3     ;AND WITH INSTR.
38 03075 175123         MOVZL   3,3,SNC ;CKN BIT 0, THE ALC BIT
39 03076 000403         JMP     TYMD    ;BIT 0 = 0 MEANS NO ALC CODE.
40 03077 175004         MOV     3,3,SZR ;CKN 3 LSB'S FOR SKP CODE.
41 03100 000411         JMP     TYMB    ;FOUND ALC-SKP CODE.
42 03101 044403 TYMD:   STA     1,TYMJ  ;STORE INSTR. IN TIME LOOP.
43 03102 060111         NIOS    XTTO    ;START TTO, FOR NON ALC-SKKP INSTR.
44 03103 151400 TYMF:   INC     2,2     ;THESE **********************
45 03104 000000 TYMJ:   0               ;   INSTRUCTIONS ************
46 03105 063511         SKPBZ   XTTO    ;      FORM THE *************
47 03106 000775         JMP     TYMF    ;         TIMING LOOP *******
48 03107 052021 TYMC:   STA     2,@IDX1 ;STORE COUNT INTO TYME LIST.
49 03110 000755         JMP     TYMA    ;LOOP.
50 03111 044403 TYMB:   STA     1,TYMH  ;STORE ALC-SKP IN TIME LOOP.
51 03112 060111         NIOS    XTTO    ;START TTO
52 03113 151400 TYMG:   INC     2,2     ;THESE **********************
53 03114 000000 TYMH:   0               ;   INSTRUCTIONS ***********
54 03115 000401         JMP     .+1     ;      FORM THE ************
55 03116 063511         SKPBZ   XTTO    ;         TIMING ***********
56 03117 000774         JMP     TYMG    ;            LOOP **********
57 03120 000767         JMP     TYMC    ;FOR ALC-SKP INSTRUCTIONS.
58
59 03121 000000 INSW1:  0               ;HERE WORKS ISZ INSAD FROM TYMJ
60 03122 100005 INDW1:  @5              ;HERE WORKS LDA @INDAD FROM TYMJ
```

```
0052  .MAIN
01
02 03123 101000 INSTR:  MOV     0,0       ;THIS IS THE 16 INSTRUCTION
03 03124 103000         ADD     0,0       ;LIST, SELECTED TO DEVELOPE
04 03125 103401         AND     0,0,SKP   ;AN IDENTITY PROFILE OF THE
05 03126 020005         LDA     0,5       ;PROCESSOR IN THE COURSE OF
06 03127 040005         STA     0,5       ;BEING EXECUTED. THE INSTR. ARE
07 03130 010415         ISZ     INSAD     ;LATER STORED IN TYMJ,TYMH
08 03131 000401         JMP     INSTA
09 03132 004401 INSTA:  JSR     INSTB
10 03133 022005 INSTB:  LDA     0,@5
11 03134 022023         LDA     0,@IDX3
12 03135 022416         LDA     0,@INDAD  ;LABEL INDAD PLACED .+16
13 03136 060400         DIA     0,0
14 03137 061000         DOA     0,0
15 03140 061477         INTA    0
16 03141 063400         SKPBN   0
17 03142 060100         NIOS    0
18 03143 000000         0                 ;END OF INSTR. LIST MARKER
19 03144 100007 CSKP:   100007            ;ONLY FOR ASSEMBLING ISZ INSAD
20 03145 000000 INSAD:  0                 ;TO WORK IN INSW1 AND INSW2.
21
22 03146 102520 SCORE:  SUBZL   0,0       ;SET THE
23 03147 040455         STA     0,CRDINAL ;ORDINAL COUNTER TO +1.
24 03150 020460 SCORA:  LDA     0,XTYME   ;GET TYME LIST INITIAL
25 03151 040021         STA     0,IDX1    ;ADDRESS FOR AUTO INC.
26 03152 020455         LDA     0,XC20    ;SET UP THE X16 COUNTER.
27 03153 040453 INDAD:  STA     0,XX16    ;WITH A COUNT OF 16 DEC.
28 03154 026021         LDA     1,@IDX1   ;GET TYME ENTRY INTO AC1 AS FIRST
29                                        ;BIG TYME. THEN SEARCH FOR BIGGER TYME.
30 03155 022021 SCORB:  LDA     0,@IDX1   ;CK MAGNITYDE OF NX TYME ENTRY.
31 03156 106033         ADCZ#   0,1,SNC   ;SKPS IF AC0 < AC1
32 03157 105000         MOV     0,1       ;AC0 > OR = AC1, AC1 = BIGGEST TYME.
33 03160 014446         DSZ     XX16      ;COUNT DOWN # OF TYMES.
34 03161 000774         JMP     SCORB     ;STILL MORE TYMES, SO LOOP.
35
36                      ;REVIEW CLEARS ALL BIG TYMES. AC1 = THE BIGGEST TYME NOW.
37                      ;LOC'S = AC1 OR (AC1-1) OR (AC1-2) ARE CLEARED AND THERE
38                      ;POS IN RANK LIST ARE SET TO THE (C) OF THE ORDINAL COUNT
39 03162 125015 REVU:   MOV#    1,1,SNR   ;IF A SCORE PASS IS COMPL WITH
40 03163 000566         JMP     KEYS      ;AC1 = 0, TYME IS CLEAR, SO GO FORM KEY.
41 03164 020444         LDA     0,XTYME   ;REINITIALIZE RVTMP WITH
42 03165 040435         STA     0,RVTMP   ;TYME - 1.
43 03166 020444         LDA     0,XRANK   ;INITIALIZE IDX2 WITH
44 03167 040022         STA     0,IDX2    ;RANK - 1.
45 03170 020437         LDA     0,XC20    ;RESET X16 COUNTER
46 03171 040435         STA     0,XX16    ;BACK TO 16 DECIMAL.
47 03172 010430 REVUA:  ISZ     RVTMP     ;INC TYMES LIST POINTER.
48 03173 030421         LDA     2,TYMEN   ;GET TOLERANCE COUNT
49 03174 022426         LDA     0,@RVTMP  ;GET TYME ENTRY AND
50 03175 106415 REVUD:  SUB#    0,1,SNR   ;COMP WITH LARGEST TYME ENTRY.
51 03176 000411         JMP     REVUB     ;IF BIGGEST TYME, STORE ORDINAL.
52 03177 101400         INC     0,0       ;ADD +1 AND COMP AGAIN WITH BIG TYME.
53 03200 151404         INC     2,2,SZR   ;TEST FOR TOLERANCE BIG TYME 0,-1,-2,...
54 03201 000774         JMP     REVUD     ;IF BIGGEST TYME -1,-2,-3...-N, STORE ORD
55 03202 022022         LDA     0,@IDX2   ;ENTRY OUT OF RANGE, INC RANK POINTER.
56 03203 014423 REVUC:  DSZ     XX16      ;DECREMENT THE X16 POINTER.
57 03204 000766         JMP     REVUA     ;STILL MORE TYMES, SO LOOP.
58 03205 010417         ISZ     ORDINAL   ;INC ORDINAL COUNT.
59 03206 000742         JMP     SCORA     ;GO TO SCORA TO FIND NX BIG TYME.
60 03207 030415 REVUB:  LDA     2,ORDINAL ;GET ORDINAL COUNT
61 03210 052022         STA     2,@IDX2   ;AND STORE IT IN PRESENT RANK LOC.
62 03211 176440         SUBO    3,3       ;CLEARING AC3.
63 03212 056410         STA     3,@RVTMP  ;CLEARS LOC. IN TYME LIST.
64 03213 000770         JMP     REVUC     ;
```

```
I 0053  .MAIN
01
02 03214 177775 TYMEN:  -3              ;-# OF COUNTS, TOLERENCE FOR TYMES
03 03215 177767 TYRTC:  -9.             ;-3 WILL GIVE BIG, BIG-1, BIG-2 IS EQUAL.
04 03216 002572 TYMEM:  TIMEM           ;MESSURED TIME FOR ONE TTO CHAR
05 03217 001212 TYLIM:  650.            ;LIMIT TO TELL SPEED > 1600 BAUD
06 03220 177772 TYTTS:  -6              ;COUNT TOLERANCE SLOW TTY
07 03221 177775 TYTTF:  -3              ;COUNT TOLERANCE FAST TTY
08 03222 000000 RVTMP:  0
09 03223 000000 SVTYME: 0
10 03224 000000 ORDINAL: 0
11 03225 000005 INXW5:  5               ;START LDA 0,@IDX3 IN CELL 5
12 03226 000000 XX16:   0
13 03227 000020 XC20:   20
14 03230 003232 XTYME:  TYME-1
15 03231 003122 XINST:  INSTR-1
16 03232 003253 XRANK:  RANK-1
17       000020 TYME:   .BLK 20
18 03253 000000 ENTYM:  0               ;END OF TYME LIST MARKER.
19       000020 RANK:   .BLK 20
20 03274 000000 ENRNK:  0               ;END OF RANK LIST MARKER.
21
22 03275 054726 TOMER:  STA     3,SVTYME
23 03276 020717         LDA     0,TYRTC ;GET RTC TOLERANCE
24 03277 040715         STA     0,TYMEN ;STORE # OF COUNTS
25 03300 020730         LDA     0,XTYME ;GET ADDRESS OF TYME LIST.
26 03301 040021         STA     0,IDX1  ;STORE IN AUTO POINTER.
27 03302 020727         LDA     0,XINST ;GET ADDR OF INSTRUCTION LIST
28 03303 040020         STA     0,IDX0  ;STORE IN AUTO POINTER.
29 03304 030721         LDA     2,INXW5
30 03305 050023         STA     2,IDX3  ;INITIALIZE IDX3 FOR LDA INST.
31 03306 102520         SUBZL   0,0
32 03307 101120         MOVZL   0,0     ;AC0:=2
33 03310 061014         DOA     0,XRTC  ;SET RTC FREQUENCY TO 100 HZ (10 MSEC).
34 03311 060114         NIOS    XRTC
35 03312 063514         SKPBZ   XRTC    ;SYNCHRONIZE PROGRAM
36 03313 000777         JMP     .-1     ;WITH RTC.
37 03314 152440 TOMA:   SUBO    2,2     ;CLEAR AC2
38 03315 141000         MOV     2,0     ;CLEAR AC0 ALSO.
39 03316 026020         LDA     1,@IDX0 ;GET INST. FROM LIST
40 03317 125015         MOV#    1,1,SNR ;ZERO MARKS END OF INSTR. LIST
41 03320 000626         JMP     SCORE   ;
42 03321 034623         LDA     3,CSKP  ;GET ALC-SKP MASK AND
43 03322 137400         AND     1,3     ;AND WITH INSTR.
44 03323 175123         MOVZL   3,3,SNC ;CKN BIT 0, THE ALC BIT
45 03324 000403         JMP     TOMD    ;BIT 0 = 0 MEANS NO ALC CODE.
46 03325 175004         MOV     3,3,SZR ;CKN 3 LSB'S FOR SKP CODE.
47 03326 000411         JMP     TOMB    ;FOUND ALC-SKP CODE.
48 03327 044403 TOMD:   STA     1,TOMJ  ;STORE INSTR. IN TIME LOOP.
49 03330 060114         NIOS    XRTC    ;START RTC
50 03331 151400 TOMF:   INC     2,2     ;THESE **********************
51 03332 000000 TOMJ:   0               ;   INSTRUCTIONS ************
52 03333 063514         SKPBZ   XRTC    ;       FORM THE *************
53 03334 000775         JMP     TOMF    ;          TIMING LOOP *******
54                                      ;FOR NON ALC-SKP INSTR.
```

```
I 0054  .MAIN
01
02 03335 052021 TOMC:   STA     2,@IDX1 ;STORE COUNT INTO TYME LIST.
03 03336 000756         JMP     TOMA    ;LOOP.
04 03337 044403 TOMB:   STA     1,TOMH  ;STORE ALC-SKP IN TIME LOOP.
05 03340 060114         NIOS    XRTC    ;START RTC
06 03341 151400 TOMG:   INC     2,2     ;THESE **********************
07 03342 000000 TOMH:   0               ;   INSTRUCTIONS ************
08 03343 000401         JMP     .+1     ;      FORM THE *************
09 03344 063514         SKPBZ   XRTC    ;         TIMING ************
10 03345 000774         JMP     TOMG    ;            LOOP ***********
11 03346 000767         JMP     TOMC    ;FOR ALC-SKP INSTRUCTIONS.
12 03347 000000 INSW2:  0               ;HERE WORKS ISZ INSAD FROM TOMJ.
13 03350 100000 INDW2:  @0              ;HERE WORKS LDA @INDAD FROM TOMJ.
14
15 03351 020661 KEYS:   LDA     0,XRANK ;REINITIALIZE RANK LIST
16 03352 040022         STA     0,IDX2  ;AUTO INC POINTER.
17 03353 102520         SUBZL   0,0     ;BIT 15:=1 AS END OF KEY MARKER.
18
19 03354 026022 KEYA:   LDA     1,@IDX2 ;GET ORDINAL COUNT FROM RANK LIST.
20 03355 125223         MOVZR   1,1,SNC ;CK IF # IS ODD OR EVEN.
21 03356 000404         JMP     KEYB    ;EVEN
22 03357 101122         MOVZL   0,0,SZC ;ROTATE 0 INTO BIT 15 FOR ODD ORDINAL.
23 03360 000405         JMP     SESAME  ;CARRY BIT SET MEANS END OF KEY.
24 03361 000773         JMP     KEYA    ;MORE-ON-KEY.
25 03362 101142 KEYB:   MOVCL   0,0,SZC ;ROTATE 1 INTO BIT 15 FOR EVEN ORDINAL.
26 03363 000402         JMP     SESAME  ;CARRY BIT SET MEANS END OF KEY.
27 03364 000770         JMP     KEYA    ;MORE-ON-KEY.
28
29 03365 000401 SESAME: JMP     .+1     ;FOR TROUBLE HALT
30 03366 024415         LDA     1,SES11 ;GET FIRST CPUNO
31 03367 034426         LDA     3,LOCKA ;ADDR OF LOCK TABLE
32 03370 031400 SESA1:  LDA     2,0,3   ;GET KEY FROM TABLE
33 03371 151015         MOV#    2,2,SNR ;ZERO ?
34 03372 000407         JMP     SESOUT  ;YES, END MARK: KEY NOT KNOWN
35 03373 150015         COM#    2,2,SNR ;ALL ONES ?
36 03374 125400         INC     1,1     ;YES, NEXT CPU MARK, INC CPUNO
37 03375 112435         SUBZ#   0,2,SNR ;COMPARE KEY WITH LOCK ENTRY
38 03376 000404         JMP     SESEX   ;A HIT ! RETURN WITH CPUNO IN AC1
39 03377 175400         INC     3,3     ;INC TO NEXT LOCK ENTRY
40 03400 000770         JMP     SESA1   ;AND LOOP
41 03401 126440 SESOUT: SUBO    1,1     ;CLEAR AC1 AS KEY INDICATOR FOR NO
42 03402 002621 SESEX:  JMP     @SVTYME ;CPU FOUND, RETURN.
43 03403 000011 SES11:  11
44
45
46              NOTYM:  .TXT    ! SET CPUNO > AC2!
   03404 051440
   03405 052105
   03406 041440
   03407 052520
   03410 047516
   03411 037040
   03412 040440
   03413 031103
   03414 000000
```

```
I 0055  .MAIN
01
02 03415 003416 LOCKA:  .+1
03 03416 064450         064450          ;NOVA KEY.                  CPU #   11
04 03417 177777         177777
05 03420 016137         016137          ;NOVA 1200 KEY.                     12
06 03421 177777         177777
07 03422 034174         034174          ;NOVA 800 KEY.                      13
08 03423 177777         177777
09 03424 002544         002544          ;SUPERNOVA KEY.                     14
10 03425 177777         177777
11 03426 007126         007126          ;SUPERNOVA SC KEY.                  15
12 03427 177777         177777
13 03430 034652         034652          ;NOVA 2 - 8K KEY.                   16
14 03431 177777         177777
15 03432 022512         022512          ;NOVA 2 - 16K KEY.                  17
16 03433 177777         177777
17 03434 170225         170225          ;RC3603/RC3609, 16K KEY             20
18 03435 170265         170265
19 03436 170365         170365
20 03437 172225         172225
21 03440 172265         172265
22 03441 177777         177777
23 03442 144567         144567          ;RC3603/RC3609/BREAK ON KEY         21
24 03443 146427         146427
25 03444 166610         166610
26 03445 177777         177777
27 03446 166065         166065          ;RC3603/RC3608, 32K KEY             22
28 03447 166165         166165
29 03450 166225         166225
30 03451 166325         166325
31 03452 172065         172065
32 03453 172325         172325
33 03454 174050         174050
34 03455 174150         174150
35 03456 174210         174210
36 03457 174250         174250
37 03460 174310         174310
38 03461 176010         176010
39 03462 177777         177777
40 03463 144352         144352          ;RC3603/RC3608/BREAK ON KEY         23
41 03464 154352         154352
42 03465 160135         160135
43 03466 162025         162025
44 03467 162202         162202
45 03470 162225         162225
46 03471 166025         166025
47 03472 170312         170312
48 03473 170352         170352
49 03474 172012         172012
50 03475 172052         172052
51 03476 172202         172202
52 03477 176025         176025
53 03500 176050         176050
54 03501 177777         177777
55 03502 020653         020653          ;NOVA 2 - DATARAM 900 NSEC KEY      24
56 03503 177777         177777
57 03504 000000         0               ;SPACE IF TROUBLES                  25
58                                      ;CORRECT JMP .+1 IN SESAME TO A HALT
59                                      ;RESTART IN 1400 (REBIN). PUT HERE
60                                      ;KEY FROM AC0 EXAMINED WHEN STOPPED,
61                                      ;AND CORRECT CPUNO 25'S CONSTANTS IN THE
62                                      ;3 ROUTINES CWAIT, TIMSK, TIMMS
63 03505 000000         0               ;END OF TABLE, NO CPU FOUND, #  0
```

```
| 0056  .MAIN
01
02                          ;FIND TYPE OF CPU USING RTC
03                          ;PART OF CPUTYP
04 03506 054444 NWTYP:  STA     3,NIRET ;SAVE RETURN
05 03507 102520         SUBZL   0,0     ;AC0:=1
06 03510 101120         MOVZL   0,0     ;AC0:=2
07 03511 126400         SUB     1,1
08 03512 061014         DOA     0,XRTC  ;SET RTC FREQUENCY
09 03513 060114         NIOS    XRTC    ;TO 100 HZ (10 MSEC)
10 03514 063514         SKPBZ   XRTC
11 03515 000777         JMP     .-1     ;SYNCHRONIZE RTC
12 03516 060114         NIOS    XRTC    ;START RTC
13 03517 125400         INC     1,1     ;COUNTS          ;X USEC
14 03520 063514         SKPBZ   XRTC    ;                ;X USEC
15 03521 000776         JMP     .-2     ;LOOP FOR 10 MS  ;X USEC
16 03522 004454         JSR     NTTYP   ;TEST RTC STABILITY
17 03523 030507         LDA     2,NN500 ;STEP -500
18 03524 034507         LDA     3,NN9   ;8 TIMES
19 03525 141000         MOV     2,0     ;STARTING AT -500=SPEED
20 03526 175405 NXTYP:  INC     3,3,SNR ;NEXT STEP OR
21 03527 000413         JMP     NFTYP   ;TYPE OF NOVA NOT FOUND
22 03530 143020         ADDZ    2,0     ;SPEED:=SPEED-500
23 03531 107013         ADD#    0,1,SNC ;NO. OF COUNTS>=(-SPEED)
24 03532 000774         JMP     NXTYP   ;THEN GO TO NXTYP ELSE
25 03533 030501         LDA     2,NC8   ;CONVERT TIMES TO CPUNO
26 03534 173000         ADD     3,2     ;AS MENTIONED BELOW
27 03535 034416         LDA     3,NEXIS ;CHECK IN
28 03536 157000         ADD     2,3     ;THE TABEL THAT
29 03537 025400         LDA     1,0,3   ;THE FOUND CPUNO
30 03540 125004         MOV     1,1,SZR ;EXIST
31 03541 000403         JMP     NRTYP   ;YES, RETURN
32 03542 006043 NFTYP:  CCRLF           ;NO
33 03543 004421 NITYP:  JSR     NOTYP   ;GET OPERATORS CPUNO
34 03544 050405 NRTYP:  STA     2,NCTYP ;SAVE CPUNO
35 03545 006046         CDICL
36 03546 006043         CCRLF
37 03547 024402         LDA     1,NCTYP ;RESTORE CPUNO
38 03550 002402         JMP     @NIRET  ;EXIT
39 03551 000000 NCTYP:  0
40 03552 000000 NIRET:  0
41
42                                      ;RANGE          LOOPS OF X USEC CPUNO   TYPE
43 03553 003554 NEXIS:  .+1             ;   0- 500
44 03554 000001         1               ; 500-1000       794    12,6    0      NOVA
45 03555 000000         0               ;1000-1500                      1
46 03556 000001         1               ;1500-2000      1724    5,80    2      3608BREAK
47                                      ;               1770    5,65    2      3609BREAK
48                                      ;               1905    5,25    2      12XX
49 03557 000001         1               ;2000-2500      2222    4,5     3      SUPER
50 03560 000001         1               ;2500-3000      2702    3,7     4      SUPER SC
51                                      ;               2778    3,6     4      830
52 03561 000001         1               ;3000-3500      3012    3,32    5      3603/3608
53                                      ;               3125    3,2     5      NOVA2-16K
54                                      ;               3226    3,10    5      3603/3609
55                                      ;               3333    3,0     5   800,820,840
56                                      ;               3448    2,9     5   N-2-DATARAM
57 03562 000001         1               ;3500-4000      3703    2,7     6      NOVA2-8K
58 03563 000000         0               ;4000-4500                      7
```

```
| 0057  .MAIN
01
02 03564 054411 NOTYP:  STA     3,NOTRE
03 03565 006046         CDICL           ;CLEAR DIS
04 03566 006044         CDISP           ;DIS - MESSAGE
05 03567 003404         NOTYM
06 03570 006040         CMESS
07 03571 003404         NOTYM           ; SET CPUNO > AC2
08 03572 006043         CCRLF           ;NO DISATT, WAIT ROUT. NO CONSTANT
09 03573 063077         HALT            ;PUT CPUNO INTO AC2
10 03574 002401         JMP     @NOTRE  ;CONTINUE
11 03575 000000 NOTRE:  0
12
13 03576 020437 NTTYP:  LDA     0,NC125 ;TEST STABILITY
14 03577 040431         STA     0,NTDEC ;125 TIMES
15 03600 044431         STA     1,NTRES ;STORE FIRST RESULT
16 03601 054426         STA     3,NTRET
17 03602 126440 NTREP:  SUBO    1,1     ;AC1:=0
18 03603 060114         NIOS    XRTC    ;START RTC
19 03604 125400         INC     1,1     ;COUNTS
20 03605 063514         SKPBZ   XRTC    ;LOOP FOR 10 MSEC
21 03606 000776         JMP     .-2
22 03607 034422         LDA     3,NTRES ;GET FIRST COUNT
23 03610 161220         MOVZR   3,0     ;50 %
24 03611 101220         MOVZR   0,0     ;25 %
25 03612 101220         MOVZR   0,0     ;12,5 %
26 03613 101220         MOVZR   0,0     ;6,25 %
27 03614 101220         MOVZR   0,0     ;AC0:=3,12 %
28 03615 117000         ADD     0,3     ;AC3:=103,12 %
29 03616 030413         LDA     2,NTRES
30 03617 112400         SUB     0,2     ;AC2:=96,88 %
31 03620 166433         SUBZ#   3,1,SNC
32 03621 132432         SUBZ#   1,2,SZC
33 03622 000414         JMP     NYTYP   ;OUTSIDE TOLERANCE
34 03623 014405         DSZ     NTDEC   ;97 %<COUNT<103 %
35 03624 000756         JMP     NTREP   ;TRY 125 TIMES
36 03625 024404         LDA     1,NTRES ;GET FIRST RESULT
37 03626 002401         JMP     @NTRET  ;EXIT
38 03627 000000 NTRET:  0
39 03630 000000 NTDEC:  0
40 03631 000000 NTRES:  0
41       000012         .RDX 10
42 03632 177014 NN500:  -500
43 03633 177767 NN9:    -9
44 03634 000010 NC8:    8
45 03635 000175 NC125:  125
46       000010         .RDX 8
47 03636 006046 NYTYP:  CDICL
48 03637 006044         CDISP
49 03640 003645         NZTYP           ;RTC IS UNSTABLE,
50 03641 006043         CCRLF
51 03642 006040         CMESS
52 03643 003645         NZTYP           ;NO DISATT, WAIT ROUT. NO CONSTANT
53 03644 000677         JMP     NITYP
54
55              NZTYP:  .TXT !RTC IS UNSTABLE,! ;"RTC IS UNSTABLE,"
   03645 052122
   03646 020103
   03647 051511
   03650 052440
   03651 051516
   03652 040524
   03653 046102
   03654 026105
   03655 000000
```

```
I 0058  .MAIN
01
02          000010 .RDX     8
03                 PTAB1:
04                 .TXTE?
05 03656 000006 <6><0>
06 03657 000006 <6><0>
07 03660 000006 <6><0>
08 03661 000006 <6><0>
09 03662 000006 <6><0>
10 03663 000006 <6><0>
11 03664 000006 <6><0>
12 03665 000006 <6><0>
13 03666 000006 <6><0>
14 03667 004400 <0><11>
15 03670 005000 <0><12>
16 03671 000006 <6><0>
17 03672 006000 <0><14>
18 03673 106400 <0><15>
19 03674 000006 <6><0>
20 03675 000006 <6><0>
21 03676 000006 <6><0>
22 03677 000006 <6><0>
23 03700 000006 <6><0>
24 03701 000006 <6><0>
25 03702 000006 <6><0>
26 03703 000006 <6><0>
27 03704 000006 <6><0>
28 03705 000006 <6><0>
29 03706 000006 <6><0>
30 03707 000006 <6><0>
31 03710 000006 <6><0>
32 03711 000006 <6><0>
33 03712 000006 <6><0>
34 03713 000006 <6><0>
35 03714 000006 <6><0>
36 03715 000006 <6><0>
37 03716 117400 <0><37>
38 03717 020400 <0><41>
39 03720 021000 <0><42>
40 03721 121400 <0><43>
41 03722 022000 <0><44>
42 03723 122400 <0><45>
43 03724 123000 <0><46>
44 03725 023400 <0><47>
45 03726 024000 <0><50>
46 03727 124400 <0><51>
47 03730 125000 <0><52>
48 03731 025400 <0><53>
49 03732 126000 <0><54>
50 03733 026400 <0><55>
51 03734 027000 <0><56>
52 03735 127400 <0><57>
53 03736 030000 <0><60>
54 03737 130400 <0><61>
55 03740 131000 <0><62>
56 03741 031400 <0><63>
57 03742 132000 <0><64>
58 03743 032400 <0><65>
59 03744 033000 <0><66>
60 03745 133400 <0><67>
61 03746 134000 <0><70>
62 03747 034400 <0><71>
63 03750 035000 <0><72>
64 03751 135400 <0><73>
65 03752 036000 <0><74>
```

```
| 0059   .MAIN
02 03753 136400 <0><75>
03 03754 137000 <0><76>
04 03755 037400 <0><77>
05 03756 140000 <0><100>
06 03757 040400 <0><101>
07 03760 041000 <0><102>
08 03761 141400 <0><103>
09 03762 042000 <0><104>
10 03763 142400 <0><105>
11 03764 143000 <0><106>
12 03765 043400 <0><107>
13 03766 044000 <0><110>
14 03767 144400 <0><111>
15 03770 145000 <0><112>
16 03771 045400 <0><113>
17 03772 146000 <0><114>
18 03773 046400 <0><115>
19 03774 047000 <0><116>
20 03775 147400 <0><117>
21 03776 050000 <0><120>
22 03777 150400 <0><121>
23 04000 151000 <0><122>
24 04001 051400 <0><123>
25 04002 152000 <0><124>
26 04003 052400 <0><125>
27 04004 053000 <0><126>
28 04005 153400 <0><127>
29 04006 154000 <0><130>
30 04007 054400 <0><131>
31 04010 055000 <0><132>
32 04011 155400 <0><133>
33 04012 056000 <0><134>
34 04013 156400 <0><135>
35 04014 157000 <0><136>
36 04015 057400 <0><137>
37 04016 006000 <0><14>
38 04017 040400 <0><101>
39 04020 041000 <0><102>
40 04021 141400 <0><103>
41 04022 042000 <0><104>
42 04023 142400 <0><105>
43 04024 143000 <0><106>
44 04025 043400 <0><107>
45 04026 044000 <0><110>
46 04027 144400 <0><111>
47 04030 145000 <0><112>
48 04031 045400 <0><113>
49 04032 146000 <0><114>
50 04033 046400 <0><115>
51 04034 047000 <0><116>
52 04035 147400 <0><117>
53 04036 050000 <0><120>
54 04037 150400 <0><121>
55 04040 151000 <0><122>
56 04041 051400 <0><123>
57 04042 152000 <0><124>
58 04043 052400 <0><125>
59 04044 053000 <0><126>
60 04045 153400 <0><127>
61 04046 154000 <0><130>
62 04047 054400 <0><131>
63 04050 055000 <0><132>
64 04051 140000 <0><100>
65 04052 121400 <0><43>
```

```
1 0060   .MAIN
02 04053 156400 <0><135>
03 04054 004400 <0><11>
04 04055 000006 <6><0>?
   04056 000000
05
06       000012 .RDX     10
07              PTAB2:
08              .TXTE?
09 04057 000006 <6><0>
10 04060 000006 <6><0>
11 04061 000006 <6><0>
12 04062 000006 <6><0>
13 04063 000006 <6><0>
14 04064 000006 <6><0>
15 04065 000006 <6><0>
16 04066 000006 <6><0>
17 04067 000006 <6><0>
18 04070 004400 <0><9>
19 04071 005000 <0><10>
20 04072 000006 <6><0>
21 04073 006000 <0><12>
22 04074 106400 <0><13>
23 04075 000006 <6><0>
24 04076 000006 <6><0>
25 04077 000006 <6><0>
26 04100 000006 <6><0>
27 04101 000006 <6><0>
28 04102 000006 <6><0>
29 04103 000006 <6><0>
30 04104 000006 <6><0>
31 04105 000006 <6><0>
32 04106 000006 <6><0>
33 04107 000006 <6><0>
34 04110 000006 <6><0>
35 04111 000006 <6><0>
36 04112 000006 <6><0>
37 04113 000006 <6><0>
38 04114 000006 <6><0>
39 04115 000006 <6><0>
40 04116 000006 <6><0>
41 04117 117400 <0><31>
42 04120 036000 <0><60>
43 04121 137000 <0><62>
44 04122 040400 <0><65>
45 04123 131000 <0><50>
46 04124 034400 <0><57>
47 04125 035000 <0><58>
48 04126 136400 <0><61>
49 04127 030000 <0><48>
50 04130 130400 <0><49>
51 04131 135400 <0><59>
52 04132 027000 <0><46>
53 04133 020400 <0><33>
54 04134 127400 <0><47>
55 04135 120000 <0><32>
56 04136 133400 <0><55>
57 04137 022000 <0><36>
58 04140 122400 <0><37>
59 04141 123000 <0><38>
60 04142 023400 <0><39>
61 04143 024000 <0><40>
62 04144 124400 <0><41>
63 04145 125000 <0><42>
64 04146 025400 <0><43>
65 04147 126000 <0><44>
```

```
: 0061   .MAIN
02 04150 026400 <0><45>
03 04151 021000 <0><34>
04 04152 121400 <0><35>
05 04153 032400 <0><53>
06 04154 132000 <0><52>
07 04155 033000 <0><54>
08 04156 037400 <0><63>
09 04157 041000 <0><66>
10 04160 057400 <0><95>
11 04161 157000 <0><94>
12 04162 156400 <0><93>
13 04163 056000 <0><92>
14 04164 155400 <0><91>
15 04165 055000 <0><90>
16 04166 054400 <0><89>
17 04167 154000 <0><88>
18 04170 153400 <0><87>
19 04171 053000 <0><86>
20 04172 052400 <0><85>
21 04173 152000 <0><84>
22 04174 051400 <0><83>
23 04175 151000 <0><82>
24 04176 150400 <0><81>
25 04177 050000 <0><80>
26 04200 147400 <0><79>
27 04201 047000 <0><78>
28 04202 046400 <0><77>
29 04203 146000 <0><76>
30 04204 045400 <0><75>
31 04205 145000 <0><74>
32 04206 144400 <0><73>
33 04207 142400 <0><69>
34 04210 042000 <0><68>
35 04211 141400 <0><67>
36 04212 044000 <0><72>
37 04213 043400 <0><71>
38 04214 143000 <0><70>
39 04215 031400 <0><51>
40 04216 134000 <0><56>
41 04217 006000 <0><12>
42 04220 057400 <0><95>
43 04221 157000 <0><94>
44 04222 156400 <0><93>
45 04223 056000 <0><92>
46 04224 155400 <0><91>
47 04225 055000 <0><90>
48 04226 054400 <0><89>
49 04227 154000 <0><88>
50 04230 153400 <0><87>
51 04231 053000 <0><86>
52 04232 052400 <0><85>
53 04233 152000 <0><84>
54 04234 051400 <0><83>
55 04235 151000 <0><82>
56 04236 150400 <0><81>
57 04237 050000 <0><80>
58 04240 147400 <0><79>
59 04241 047000 <0><78>
60 04242 046400 <0><77>
61 04243 146000 <0><76>
62 04244 045400 <0><75>
63 04245 145000 <0><74>
64 04246 144400 <0><73>
65 04247 142400 <0><69>
```

```
1 0062   .MAIN
02 04250 042000 <0><68>
03 04251 141400 <0><67>
04 04252 041000 <0><66>
05 04253 040400 <0><65>
06 04254 143000 <0><70>
07 04255 004400 <0><9>
08 04256 000006 <6><0>?
   04257 000000
09
10         000012 .RDX     10
11                PTAB3:
12                .TXTE?
13 04260 000006 <6><0>
14 04261 000006 <6><0>
15 04262 000006 <6><0>
16 04263 000006 <6><0>
17 04264 000006 <6><0>
18 04265 000006 <6><0>
19 04266 000006 <6><0>
20 04267 000006 <6><0>
21 04270 000006 <6><0>
22 04271 004400 <0><9>
23 04272 005000 <0><10>
24 04273 000006 <6><0>
25 04274 006000 <0><12>
26 04275 106400 <0><13>
27 04276 000006 <6><0>
28 04277 000006 <6><0>
29 04300 000006 <6><0>
30 04301 000006 <6><0>
31 04302 000006 <6><0>
32 04303 000006 <6><0>
33 04304 000006 <6><0>
34 04305 000006 <6><0>
35 04306 000006 <6><0>
36 04307 000006 <6><0>
37 04310 000006 <6><0>
38 04311 000006 <6><0>
39 04312 000006 <6><0>
40 04313 000006 <6><0>
41 04314 000006 <6><0>
42 04315 000006 <6><0>
43 04316 000006 <6><0>
44 04317 000006 <6><0>
45 04320 117400 <0><31>
46 04321 134000 <0><56>
47 04322 035000 <0><58>
48 04323 136400 <0><61>
49 04324 027000 <0><46>
50 04325 032400 <0><53>
51 04326 033000 <0><54>
52 04327 034400 <0><57>
53 04330 126000 <0><44>
54 04331 026400 <0><45>
55 04332 133400 <0><55>
56 04333 125000 <0><42>
57 04334 156400 <0><93>
58 04335 025400 <0><43>
59 04336 056000 <0><92>
60 04337 031400 <0><51>
61 04340 120000 <0><32>
62 04341 020400 <0><33>
63 04342 021000 <0><34>
64 04343 121400 <0><35>
65 04344 022000 <0><36>
```

```
| 0063  .MAIN
02 04345 122400 <0><37>
03 04346 123000 <0><38>
04 04347 023400 <0><39>
05 04350 024000 <0><40>
06 04351 124400 <0><41>
07 04352 157000 <0><94>
08 04353 057400 <0><95>
09 04354 130400 <0><49>
10 04355 030000 <0><48>
11 04356 131000 <0><50>
12 04357 135400 <0><59>
13 04360 137000 <0><62>
14 04361 155400 <0><91>
15 04362 055000 <0><90>
16 04363 054400 <0><89>
17 04364 154000 <0><88>
18 04365 153400 <0><87>
19 04366 053000 <0><86>
20 04367 052400 <0><85>
21 04370 152000 <0><84>
22 04371 051400 <0><83>
23 04372 151000 <0><82>
24 04373 150400 <0><81>
25 04374 050000 <0><80>
26 04375 147400 <0><79>
27 04376 047000 <0><78>
28 04377 046400 <0><77>
29 04400 146000 <0><76>
30 04401 045400 <0><75>
31 04402 145000 <0><74>
32 04403 144400 <0><73>
33 04404 044000 <0><72>
34 04405 043400 <0><71>
35 04406 143000 <0><70>
36 04407 142400 <0><69>
37 04410 040400 <0><65>
38 04411 140000 <0><64>
39 04412 037400 <0><63>
40 04413 042000 <0><68>
41 04414 141400 <0><67>
42 04415 041000 <0><66>
43 04416 127400 <0><47>
44 04417 132000 <0><52>
45 04420 006000 <0><12>
46 04421 155400 <0><91>
47 04422 055000 <0><90>
48 04423 054400 <0><89>
49 04424 154000 <0><88>
50 04425 153400 <0><87>
51 04426 053000 <0><86>
52 04427 052400 <0><85>
53 04430 152000 <0><84>
54 04431 051400 <0><83>
55 04432 151000 <0><82>
56 04433 150400 <0><81>
57 04434 050000 <0><80>
58 04435 147400 <0><79>
59 04436 047000 <0><78>
60 04437 046400 <0><77>
61 04440 146000 <0><76>
62 04441 045400 <0><75>
63 04442 145000 <0><74>
64 04443 144400 <0><73>
65 04444 044000 <0><72>
```

```
1 0064   .MAIN
02 04445 043400 <0><71>
03 04446 143000 <0><70>
04 04447 142400 <0><69>
05 04450 040400 <0><65>
06 04451 140000 <0><64>
07 04452 037400 <0><63>
08 04453 137000 <0><62>
09 04454 136400 <0><61>
10 04455 041000 <0><66>
11 04456 004400 <0><9>
12 04457 000006 <6><0>?
   04460 000000
13
14          000010 .RDX     8
15                 PTAB4:
16                 .TXTE?
17 04461 000006 <6><0>
18 04462 000006 <6><0>
19 04463 000006 <6><0>
20 04464 000006 <6><0>
21 04465 000006 <6><0>
22 04466 000006 <6><0>
23 04467 000006 <6><0>
24 04470 000006 <6><0>
25 04471 000006 <6><0>
26 04472 004400 <0><11>
27 04473 005000 <0><12>
28 04474 000006 <6><0>
29 04475 006000 <0><14>
30 04476 106400 <0><15>
31 04477 000006 <6><0>
32 04500 000006 <6><0>
33 04501 000006 <6><0>
34 04502 000006 <6><0>
35 04503 000006 <6><0>
36 04504 000006 <6><0>
37 04505 000006 <6><0>
38 04506 000006 <6><0>
39 04507 000006 <6><0>
40 04510 000006 <6><0>
41 04511 000006 <6><0>
42 04512 000006 <6><0>
43 04513 000006 <6><0>
44 04514 000006 <6><0>
45 04515 000006 <6><0>
46 04516 000006 <6><0>
47 04517 000006 <6><0>
48 04520 000006 <6><0>
49 04521 117400 <0><37>
50 04522 022000 <0><44>
51 04523 025400 <0><53>
52 04524 157000 <0><136>
53 04525 047000 <0><116>
54 04526 156400 <0><135>
55 04527 136400 <0><75>
56 04530 127400 <0><57>
57 04531 026400 <0><55>
58 04532 027000 <0><56>
59 04533 045400 <0><113>
60 04534 137000 <0><76>
61 04535 146000 <0><114>
62 04536 145000 <0><112>
63 04537 044000 <0><110>
64 04540 046400 <0><115>
65 04541 037400 <0><77>
```

```
1 0065   .MAIN
02 04542 140000 <0><100>
03 04543 040400 <0><101>
04 04544 041000 <0><102>
05 04545 141400 <0><103>
06 04546 042000 <0><104>
07 04547 142400 <0><105>
08 04550 143000 <0><106>
09 04551 043400 <0><107>
10 04552 144400 <0><111>
11 04553 126000 <0><54>
12 04554 122400 <0><45>
13 04555 021000 <0><42>
14 04556 030000 <0><60>
15 04557 124400 <0><51>
16 04560 125000 <0><52>
17 04561 057400 <0><137>
18 04562 147400 <0><117>
19 04563 050000 <0><120>
20 04564 150400 <0><121>
21 04565 151000 <0><122>
22 04566 051400 <0><123>
23 04567 152000 <0><124>
24 04570 052400 <0><125>
25 04571 053000 <0><126>
26 04572 153400 <0><127>
27 04573 154000 <0><130>
28 04574 054400 <0><131>
29 04575 055000 <0><132>
30 04576 155400 <0><133>
31 04577 056000 <0><134>
32 04600 130400 <0><61>
33 04601 131000 <0><62>
34 04602 031400 <0><63>
35 04603 132000 <0><64>
36 04604 032400 <0><65>
37 04605 033000 <0><66>
38 04606 133400 <0><67>
39 04607 134000 <0><70>
40 04610 034400 <0><71>
41 04611 035000 <0><72>
42 04612 135400 <0><73>
43 04613 036000 <0><74>
44 04614 020400 <0><41>
45 04615 123000 <0><46>
46 04616 023400 <0><47>
47 04617 024000 <0><50>
48 04620 121400 <0><43>
49 04621 006000 <0><14>
50 04622 147400 <0><117>
51 04623 050000 <0><120>
52 04624 150400 <0><121>
53 04625 151000 <0><122>
54 04626 051400 <0><123>
55 04627 152000 <0><124>
56 04630 052400 <0><125>
57 04631 053000 <0><126>
58 04632 153400 <0><127>
59 04633 154000 <0><130>
60 04634 054400 <0><131>
61 04635 055000 <0><132>
62 04636 155400 <0><133>
63 04637 056000 <0><134>
64 04640 130400 <0><61>
65 04641 131000 <0><62>
```

```
1 0066   .MAIN
02 04642 031400 <0><63>
03 04643 132000 <0><64>
04 04644 032400 <0><65>
05 04645 033000 <0><66>
06 04646 133400 <0><67>
07 04647 134000 <0><70>
08 04650 034400 <0><71>
09 04651 035000 <0><72>
10 04652 135400 <0><73>
11 04653 036000 <0><74>
12 04654 057400 <0><137>
13 04655 157000 <0><136>
14 04656 023400 <0><47>
15 04657 004400 <0><11>
16 04660 000006 <6><0>?
   04661 000000
17
18                          ;POWER RESTART ROUTINE
19
20 04662 026454 POWON:  LDA     1,@PSTAC;GET PRINT INHI INITIAL FOR PRINT
21 04663 046454          STA     1,@PSETP
22 04664 062677          IORST
23 04665 006061          CWAIT           ;WAIT 3 SECONDS TO TERMINAL READY
24 04666 001236          SEC3
25 04667 020442          LDA     0,PCOTT ;PRINT 5 CR, LF FOR TTY, SILENT
26 04670 040440          STA     0,PCOUN
27 04671 006043          CCRLF
28 04672 014436          DSZ     PCOUN
29 04673 000776          JMP     .-2
30 04674 020436          LDA     0,PCH14 ;FF FOR LPT AND CLEARING SOME CRT'S
31 04675 006041          CCHAR
32 04676 006061          CWAIT           ;20 MSEC FOR CRT
33 04677 001237          SECM2
34 04700 020433          LDA     0,PCH35 ;HOME UP FOR CRT
35 04701 006041          CCHAR
36 04702 006061          CWAIT           ;20 MSEC FOR CRT
37 04703 001237          SECM2
38 04704 020430          LDA     0,PCH37 ;ERASE EOF FOR CRT
39 04705 006041          CCHAR
40 04706 006061          CWAIT           ;20 MSEC FOR CRT
41 04707 001237          SECM2
42 04710 006040          CMESS
43 04711 004745          MPOWO           ;POWER
44 04712 006043          CCRLF
45 04713 006040          CMESS
46 04714 007226          PROG            ;ACTUAL PROG NAME
47 04715 006071 RPSAG:   CQUES
48 04716 004755          MSAGU           ;SET SWITCHES, START ADDR
49 04717 004750          DSAGU
50 04720 004735          PSAAN           ;SUGGESTED ANSWER
51 04721 006054          CTZCC
52 04722 006060          CDZOC
53 04723 006104          CGTOK           ;READ ANSWER
54 04724 000402          JMP     .+2     ;SUGGESTED ACCEPTED BY OPERATOR
55 04725 000770          JMP     RPSAQ   ;ERROR RETURN
56 04726 030075          LDA     2,DIGIN ;ANSWER INPUT'ED
57 04727 001000          JMP     0,2     ;START PROG
```

```
| 0067  .MAIN
01
02 04730 000000 PCOUN:  0                ;COUNTER
03 04731 000005 PCOTT:  5                ;5 CR,LF
04 04732 000014 PCH14:  14               ;FF
05 04733 000035 PCH35:  35               ;HOME UP
06 04734 000037 PCH37:  37               ;ERASE EOF
07 04735 000400 PSAAN:  400              ;SUGGESTED START ADDR
08 04736 001372 PSTAC:  KSTAC            ;ASM VALUE FOR JMP      NINHI
09 04737 000574 PSETP:  SETAC
10
11                      ;INITIAL START ADDRESS ROUTINE
12                      ;USED TO HELP START WITHOUT SWITCHES
13
14 04740 026776 SWISA:  LDA     1,@PSTAC
15 04741 046776         STA     1,@PSETP
16 04742 000753         JMP     RPSAQ    ;USE POWER RESTART ROUTINE
17
18              MCRLF:  .TXT !<15><12>! ;"<15><12>"
   04743 005015
   04744 000000
19
20              MPOWO:  .TXT !POWER!    ;"POWER"
   04745 047520
   04746 042527
   04747 000122
21
22              DSAQU:  .TXT !SEE3.2,SA!        ;"SEE3.2,SA"
   04750 042523
   04751 031505
   04752 031056
   04753 051454
   04754 000101
23
24              MSAQU:  .TXT !SET SWITCHES TO CONTROL, (3.2), STARTADDR!
   04755 042523
   04756 020124
   04757 053523
   04760 052111
   04761 044103
   04762 051505
   04763 052040
   04764 020117
   04765 047503
   04766 052116
   04767 047522
   04770 026114
   04771 024040
   04772 027063
   04773 024462
   04774 020054
   04775 052123
   04776 051101
   04777 040524
   05000 042104
   05001 000122
25                      ;"SET SWITCHES TO CONTROL, (3.2), STARTADDR"
26
27              MCPUT:  .TXT !<15><12>CPU TYPE: !       ;"<15><12>CPU TYPE: "
   05002 005015
   05003 050103
   05004 020125
   05005 054524
   05006 042520
   05007 020072
   05010 000000
```

```
| 2068  .MAIN
01
02                          ;ROUTINE TO EXAMINE MEMORY.
03
04 05011 165000 EXMEM:  MOV     3,1         ;
05 05012 006272         CSAMS               ;START ADDR MESSAGE
06 05013 006071 EXMMF:  CQUES
07 05014 005070         MXMMF               ;EXAMINE MEM FROM
08 05015 005101         DXMMF               ;SEE NEXT QUESTION, TO (INCL)
09 05016 005064         XFROM               ;SUGGESTED ANSWER
10 05017 006052         CTOCT
11 05020 006056         CDOCT
12 05021 006104         CGTOK               ;READ ANSWER
13 05022 000402         JMP     .+2         ;SUGGESTED ACCEPTED
14 05023 000770         JMP     EXMMF       ;ERROR RETURN
15 05024 020075         LDA     0,DIGIN     ;ANSWER INPUT'ED
16                                          ;MOVR# 0,0,SZC ;EVEN ?
17                                          ;JMP   EXMMF   ;NO, ERROR
18                                          ;LDA 1,UPPERLIMIT
19                                          ;LDA 2,LOWERLIMIT
20                                          ;ADCZ# 1,0,SNC
21                                          ;ADCZ# 0,2,SZC ;AC2=<AC0=<AC1 ?
22                                          ;JMP   EXMMF   ;OUTSIDE LIMITS
23 05025 040440         STA     0,FMADR     ;INPUT ACCEPTED
24 05026 006071 EXMMT:  CQUES
25 05027 005106         MXMMT               ;TO INCL.
26 05030 005106         MXMMT               ;USE THE SAME MESS AT DIS
27 05031 005066         XTOIN               ;SUGGESTED ANSWER
28 05032 006052         CTOCT
29 05033 006056         CDOCT
30 05034 006104         CGTOK               ;READ ANSWER
31 05035 000402         JMP     .+2         ;SUGGESTED ACCEPTED
32 05036 000770         JMP     EXMMT       ;ERROR RETURN
33 05037 020075         LDA     0,DIGIN     ;ANSWER INPUT'ED
34 05040 040427         STA     0,LMADR     ;INPUT ACCEPTED
35 05041 006043 EXPRT:  CCRLF
36 05042 006046         CDICL
37 05043 024422         LDA     1,FMADR
38 05044 006052         CTOCT
39 05045 006056         CDOCT
40 05046 030417         LDA     2,FMADR
41 05047 025000         LDA     1,0,2
42 05050 006052         CTOCT
43 05051 006056         CDOCT
44 05052 006047         CDATT
45 05053 024414         LDA     1,LMADR
46 05054 030411         LDA     2,FMADR
47 05055 010410         ISZ     FMADR       ;TO NEXT LOC
48 05056 000401         JMP     .+1
49 05057 132414         SUB#    1,2,SZR     ;LAST LOC ?
50 05060 000761         JMP     EXPRT       ;NO, NEXT
51 05061 006043         CCRLF               ;YES, PROGRAM FINISHED
52 05062 002401         JMP     @.+1
53 05063 004740         SWISA               ;RESTART MAIN PROGRAM
54
55 05064 000034 XFROM:  34                  ;FIRST MEM LOC QUES
56 05065 000000 FMADR:  0                   ;ANSWER
57 05066 000037 XTOIN:  37                  ;LAST MEM LOC QUES, INCL
58 05067 000000 LMADR:  0                   ;ANSWER
```

```
| 0069  .MAIN
01
02                  MXMMF:  .TXT !EXAMINE MEM FROM !        ;"EXAMINE MEM FROM "
   05070 054105
   05071 046501
   05072 047111
   05073 020105
   05074 042515
   05075 020115
   05076 051106
   05077 046517
   05100 000040
03
04                  DXMMF:  .TXT !X M FROM !                ;"X M FROM "
   05101 020130
   05102 020115
   05103 051106
   05104 046517
   05105 000040
05
06                  MXMMT:  .TXT !TO INCL. !                ;"TO INCL. "
   05106 047524
   05107 044440
   05110 041516
   05111 027114
   05112 000040
07
08                  MDMMC:  .TXT !DEPOSIT: !                ;"DEPOSIT: "
   05113 042504
   05114 047520
   05115 044523
   05116 035124
   05117 000040
09
10                  MDMMF:  .TXT !FROM LOC !                ;"FROM LOC "
   05120 051106
   05121 046517
   05122 046040
   05123 041517
   05124 000040
11
12 05125 177777 DPCON:  177777          ;DEPOSIT CONTENT QUES
13 05126 000000 CMADR:  0               ;ANSWER
```

```
| 0372  .MAIN
01
02                          ;ROUTINE TO DEPOSIT IN MEMORY.
03
04 05127 165000 DPMEM:  MOV     3,1
05 05130 006072         CSAMS           ;START ADDR MESSAGE
06 05131 006071 DPMMC:  CGUES
07 05132 005113         MDMMC           ;DEPOSIT:
08 05133 005113         MDMMC
09 05134 005125         DPCCN           ;SUGGESTED ANSWER
10 05135 006052         CTOCT
11 05136 006056         CDOCT
12 05137 006104         CGTCK           ;READ ANSWER
13 05140 000402         JMP     .+2     ;SUGGESTED ACCEPTED
14 05141 000770         JMP     DPMMC   ;ERROR RETURN
15 05142 020075         LDA     0,DIGIN ;ANSWER INPUT'ED
16 05143 040763         STA     0,CMADR ;INPUT ACCEPTED
17 05144 006071 DPMMF:  CGUES
18 05145 005120         MDMMF           ;FROM LOC
19 05146 005120         MDMMF
20 05147 005064         XFROM           ;SUGGESTED ANSWER
21 05150 006052         CTOCT
22 05151 006056         CDOCT
23 05152 006104         CGTCK           ;READ ANSWER
24 05153 000402         JMP     .+2     ;SUGGESTED ACCEPTED
25 05154 000770         JMP     DPMMF   ;ERROR RETURN
26 05155 020075         LDA     0,DIGIN ;ANSWER INPUT'ED
27 05156 040707         STA     0,FMADR ;INPUT ACCEPTED
28 05157 006071 DPMMT:  CGUES
29 05160 005106         MXMMT           ;TO INCL.
30 05161 005106         MXMMT
31 05162 005066         XTOIN           ;SUGGESTED ANSWER
32 05163 006052         CTOCT
33 05164 006056         CDOCT
34 05165 006104         CGTCK           ;READ ANSWER
35 05166 000402         JMP     .+2     ;SUGGESTED ACCEPTED
36 05167 000770         JMP     DPMMT   ;ERROR RETURN
37 05170 020075         LDA     0,DIGIN ;ANSWER INPUT'ED
38 05171 040676         STA     0,LMADR ;INPUT ACCEPTED
39 05172 024734         LDA     1,CMADR
40 05173 030672 DPPRT:  LDA     2,FMADR
41 05174 045000         STA     1,0,2
42 05175 034672         LDA     3,LMADR
43 05176 010667         ISZ     FMADR   ;TO NEXT LOC
44 05177 000401         JMP     .+1
45 05200 172414         SUB#    3,2,SZR ;LAST LOC ?
46 05201 000772         JMP     DPPRT   ;NO, NEXT
47 05202 002401         JMP     @.+1    ;YES, PROGRAM FINISHED
48 05203 004740         SWISA           ;RESTART MAIN PROGRAM
```

```
| 0071  .MAIN
01
02 05204 000000 RQUES:  0                ;RETURN ADDR QUES ROUTINE
03 05205 000000 QUESA:  0                ;SUGG. ANSWER
04
05              MXQUE:  .TXT !?!         ;"?"
   05206 000077
06
07              MX2SP:  .TXT !  !        ;"2 SPACE"
   05207 020040
   05210 000000
08
09                      ;ROUTINE TO OUTPUT QUESTIONS.
10                      ;HOW TO USE, SEE EXMEM.
11                      ;CALL   CQUES
12                      ;       MQUES   ;LABEL TEXT TTO/LPT 1,9,17,25 LETTERS
13                      ;       DQUES   ;LABEL TEXT DIS ALLWAYS 9 LETTERS
14                      ;       AQUES   ;LABEL SUGGESTED ANSWER
15                      ;       CTOCT   ;PRINT ROUTINE FOR SUGG. ANSWER
16                      ;       CDOCT   ;DISPLAY ROUTINE FOR SUGG. ANSWER
17                      ;       RETURN  ;TO RELEVANT CALL OF INPUT ROUTINE.
18
19 05211 054773 XQUES:  STA     3,RQUES
20 05212 025400         LDA     1,0,3   ;1. PARAM
21 05213 044417         STA     1,QUESM
22 05214 025401         LDA     1,1,3   ;2. PARAM
23 05215 044413         STA     1,QUESD
24 05216 027402         LDA     1,02,3  ;3. PARAM
25 05217 044075         STA     1,DIGIN
26 05220 044765         STA     1,QUESA
27 05221 025403         LDA     1,3,3   ;4. PARAM
28 05222 044413         STA     1,QUEST
29 05223 025404         LDA     1,4,3   ;5. PARAM
30 05224 044412         STA     1,QUESS
31 05225 006046         CDICL
32 05226 006043         CCRLF
33 05227 006044         CDISP
34 05230 000000 QUESD:  0                ;DISPLAY MESSAGE
35 05231 006040         CMESS
36 05232 000000 QUESM:  0                ;PRINT MESSAGE
37 05233 024752         LDA     1,QUESA
38 05234 121000         MOV     1,0
39 05235 000000 QUEST:  0                ;NUMBER PRINT ROUTINE TTO/LPT
40 05236 000000 QUESS:  0                ;NUMBER PRINT ROUTINE DIS
41 05237 006044         CDISP
42 05240 005206         MXQUE
43 05241 006040         CMESS
44 05242 005206         MXQUE            ;?
45 05243 006040         CMESS
46 05244 005207         MX2SP            ;2 SPACE
47 05245 034737         LDA     3,RQUES
48 05246 001405         JMP     5,3      ;BYPASS PARAM., RETURN
```

```
! 0072  .MAIN
01
02                          .EOT
```

```
  0073  ,MAIN
01
02                ;TAPE 4
03
04                ; GENERAL INPUT-ROUTINES.          VERSION  760119 BY HH
05
06                ; GETDC          GET DECIMAL NUMBER
07                ; GETOK          GET OKTAL NUMBER
08                ; GETBI          GET BINARY NUMBER
09                ; GETSC          GET SINGLE CHARACTER
10                ; GETTX          GET TEXT STRING
11
12
13                ; ABSTRACTS:
14
15                ;    AFTER CALLING DIFFERENT ROUTINES THE OPERATER CAN KEY-IN A
16                ; DEC.-, OCT.- OR A BINARY NUMBER, A CHARACTER OR A TEXT-STRING.
17                ;    THE NUMBER OR THE CHARACTER WILL AFTER A TERMINATION-INPUT
18                ; BE STORED IN "DIGIN". THE TEXT-STRING IS PACKED IN A BUFFER
19                ; CALLED "TEXIN". THE ADDRESS OF TEXIN IS STORED IN "DIGIN"
20                ; FOR INDIRECT USE.
21
22
23
24                ; DESCRIPTION:
25
26                ; "NUMBER"-INPUTROUTINES:
27
28                ; >GETDC< FOR A DECIMALNUMBER DX, WHERE -32768 <= DX <=32767.
29                ; ACCEPTED INPUTS ARE +,-,0,1 ....9.
30
31                ; >GETOK< FOR AN OCTALNUMBER OX, WHERE 0 <= OX <= 177777.
32                ; ACCEPTED INPUTS ARE 0,1, .... 7.
33
34                ; >GETBI< FOR A BINARY NUMBER BX, WHERE 0 <= BX <= 11111111.
35                ; ACCEPTED INPUTS ARE 0,1.
36
37                ; THE SPACE CHAR IS BLIND.
38
39
40                ; A TERMINATION-INPUT WILL STORE THE ENTIRE NUMBER IN THE PAGE-
41                ; ZERO-ADRESS "DIGIN".
42
43
44
45                ; "CHARACTER"-INPUTROUTINE:
46
47                ; >GETSC< FOR A SINGLE CHARACTER, ACCEPTED INPUT IS
48                ; CHARACTERS WITH THE OCTAL ASCII-CODE 11, 33, 40 - 176.
49
50                ; A TERMINATION-INPUT WILL STORE THE CHARACTER-ASCII-CODE
51                ; (WITHOUT PARITY) IN THE RIGTH HALF OF "DIGIN".
52
53
54
55                ; "TEXT"-INPUTROUTINE:
56
57                ; >GETTX< FOR A CHARACTERSTRING OF MAX. 80 CHARACTERS.
58                ; ACCEPTED INPUT IS CHARACTERS WITH THE OCTAL CODE 11, 40 - 176.
59
60                ; AFTER A TERMINATION THE STRING IS PACKED R-L AND THE 3 LAST
61                ; BYTES IN THE STRING WILL ALWAYS CONTENT CR,LF,0. THEN IT IS
62                ; STORED IN "TEXIN", FIRST UP TO 80 BYTES OF INPUT, FOLLOWED
63                ; BY THE 3 TERMINATION BYTES.
```

```
| 0074  .MAIN
01
02
03
04              ; OPERATING PROCEDURE:
05
06              ;    THE ROUTINES ARE CALLED BY THE INSTRUCTION: C>NAME<,
07              ; WHICH EQUALS JSR "ROUTINE". AFTER RECEIVING A LEGAL NUMBER,
08              ; CHARACTER OR TEXT-STRING FOLLOWED BY A TERMINATIOR, THE
09              ; PROGRAM RETURNS TO CALL+3. WHEN A FORMAT- OR OVERFLOWERROR
10              ; OCCURS, THE RETURN IS CALL+2. WHEN ONLY A TERMINATOR IS
11              ; INPUT, THE RETURN IS TO CALL+1.
12
13              ; CALL  CGTDC           ;OR CGTOK, CGTBI, CGTSC, CGTTX
14              ;       TERM RETURN
15              ;       ERROR RETURN
16              ;       NORMAL RETURN
17
18              ; TERMINATOR:
19              ;   IN ALL ROUTINES THE INPUTS: NL, LF, CR OR FF  WILL
20              ; TERMINATE THE MESSAGE.
21
22              ;   EVERY ERRORRETURN WILL INITIALISE THE BUFFERS, COUNTERS
23              ; ETC., AND THE RE-INPUT'ING THEN HAVE TO START AT THE LAST
24              ; TERMINATION POINT. AN * IS PRINTED.
25
26              ; CANCEL:
27              ;   IT'S POSSIBLE TO CANCEL THE LAST-KEYED DIGIT OR CHARACTER
28              ; BY INPUT'ING A RUBOUT, DEL OR CAN CHARACTER .THE PROGRAM
29              ; ECHOES THEN A "$", AND WAITS FOR A NEW INPUT. IT IS NOT
30              ; POSSIBLE TO CANCEL MORE THAN ONE CHAR. AN ATTEMT WILL
31              ; GIVE ERROR RETURN.
32
33              ; THE NULL CHAR IS BLIND.
34
35              ; IF A CHARACTER IS NOT TERMINATOR, BLIND, CANCEL OR
36              ; ACCEPTED, IT IS ILLEGAL AND CAUSES ERROR RETURN.
```

```
I 0075  ,MAIN
01
02                ; SUBROUTINES USED BY ALL INPUTROUTINES
03
04                ; GETCH GETS A CHARACTER FROM TTI OR NUK TO AC0
05
06 05247 054456 GETCH:  STA     3,GETRE
07 05250 004443         JSR     BZNUK   ; CHECK BUSY NUK
08 05251 004431         JSR     INTTI   ; INPUTDEVICE = TTI
09 05252 004446         JSR     INNUK   ; INPUTDEVICE = NUK
10 05253 024461         LDA     1,RAZER ; READ ASCII FROM NUK
11 05254 034454         LDA     3,RALIF ; AC0=KEY
12 05255 054452         STA     3,RADYN
13 05256 101122 RANDI:  MOVZL   0,0,SZC ;
14 05257 000421         JMP     RAEND   ; DIGIT FOUND
15 05260 125400         INC     1,1     ; AC1 = DIGIT
16 05261 014446         DSZ     RADYN   ; 0-9
17 05262 000774         JMP     RANDI
18 05263 126400         SUB     1,1
19 05264 101122         MOVZL   0,0,SZC
20 05265 024445         LDA     1,RAPLU ; AC1 = +
21 05266 101122         MOVZL   0,0,SZC
22 05267 024444         LDA     1,RAMIN ; AC1 = -
23 05270 101122         MOVZL   0,0,SZC
24 05271 024437         LDA     1,RALIF ; AC1 = LF
25 05272 101122         MOVZL   0,0,SZC
26 05273 024436         LDA     1,RASPA ; AC1 = SP
27 05274 101122         MOVZL   0,0,SZC
28 05275 024440         LDA     1,RACAN ; AC1 = CAN
29 05276 127405         AND     1,1,SNR
30 05277 063077         HALT            ; NO KEYS
31 05300 121000 RAEND:  MOV     1,0     ; AC1 TO AC0
32 05301 002424         JMP     @GETRE
33
34 05302 063610 INTTI:  SKPDN   XTTI    ;IS TTI KEY PRESSED ?
35 05303 001400         JMP     0,3     ; NO, CHECK OTHER INPUT DEVICE
36 05304 024422         LDA     1,HC177 ; YES, GET CHAR
37 05305 060410         DIA     0,XTTI
38 05306 123400         AND     1,0
39 05307 060110         NIOS    XTTI
40 05310 101015         MOV#    0,0,SNR ; NULL CHAR ?
41 05311 000771         JMP     INTTI
42 05312 002413         JMP     @GETRE
43
44 05313 126400 BZNUK:  SUB     1,1     ; AC1:=0
45 05314 060434 AANUK:  DIA     0,NUK   ; TEST BUZY NUK
46 05315 122414         SUB#    1,0,SZR ; IS AC0=0  ?
47 05316 000776         JMP     AANUK   ; NO - KEYBOARD NOT READY
48 05317 001400         JMP     0,3     ; YES, RETURN.
49
50 05320 126400 INNUK:  SUB     1,1     ; AC1:= 0
51 05321 060434 BBNUK:  DIA     0,NUK   ; GET KEY BITS
52 05322 122415         SUB#    1,0,SNR ; ARE THEY ZERO ?
53 05323 001776         JMP     -2,3    ; YES, OTHER INPUT, TEST TTI
54 05324 001400         JMP     0,3     ; NO, KEY DEPRESSED
55
56 05325 000000 GETRE:  0
57 05326 000177 HC177:  177
58 05327 000000 RADYN:  0
59 05330 000012 RALIF:  12
60 05331 000040 RASPA:  40
61 05332 000053 RAPLU:  53
62 05333 000055 RAMIN:  55
63 05334 000060 RAZER:  60
64 05335 000030 RACAN:  30
```

```
| 0076   .MAIN
01
02 05336 030551 DELTE:  LDA     2,HC30   ; THE DELTE ROUTINE LOOKS FOR RUBOUT
03 05337 112415         SUB#    0,2,SNR  ; IS AC0= 30 ?
04 05340 000405         JMP     DELOV    ; YES - RUBOUT
05 05341 030765         LDA     2,HC177  ; NO
06 05342 112415         SUB#    0,2,SNR  ; IS AC0= 177 ?
07 05343 000402         JMP     DELOV    ; YES - RUBOUT
08 05344 001401         JMP     1,3      ; NO
09
10 05345 054462 DELOV:  STA     3,DELRE  ; CHECK FOR TOO MANY RUBOUTS
11 05346 030460         LDA     2,LASTN  ; OR THE FIRST CHAR IS RUBOUT
12 05347 151102         MOVL    2,2,SZC  ; IS LASTN = 177777 ?
13 05350 000417         JMP     ILLGR    ; YES - FIRST IS RUBOUT
14 05351 151102         MOVL    2,2,SZC  ; IS LASTN = 077777 ?
15 05352 000415         JMP     ILLGR    ; YES - TOO MANY RUBOUTS
16 05353 152220         ADCZR   2,2      ; NO - PLACE 077777 IN LASTN
17 05354 050452         STA     2,LASTN  ; TO INDICATE RUBOUT
18 05355 020534         LDA     0,HC44   ; TYPE $
19 05356 004402         JSR     TYPIN
20 05357 002450         JMP     @DELRE   ; RETURN + 1
21
22 05360 054406 TYPIN:  STA     3,TYPRE  ; TYPE THE CHAR IN AC0 ON TTY/DIS
23 05361 004430         JSR     TERMT    ; TERMINATOR ?
24 05362 002404         JMP     @TYPRE   ; YES, DO NOT COPY
25 05363 006045         CDOUT            ; TRY TO TYPE CHAR ON DISPLAY
26 05364 006041         CCHAR            ; TRY TO TYPE CHAR ON TTO
27 05365 002401         JMP     @TYPRE
28 05366 000000 TYPRE:  0
29
30 05367 020522 ILLGR:  LDA     0,HC44   ; ILLEGAL ROUTINE. TYPE $
31 05370 004770 ILLEG:  JSR     TYPIN    ; COPY LAST CHAR
32 05371 020521         LDA     0,HC52
33 05372 004766         JSR     TYPIN    ; TYPE *
34 05373 126400         SUB     1,1      ; AC1:= 0
35 05374 044075         STA     1,DIGIN  ; DIGIN:= 0
36 05375 046427         STA     1,@XTXCO ; TEXIN:= CR,LF,0
37 05376 006427         JSR     @XTXND   ; FOR EMPTY TEXTSTRING
38 05377 010424         ISZ     INRET    ; ILLEGAL OR OVERFLOW RETURN
39 05400 000403         JMP     ONTER    ; IS TO CALL + 2
40
41 05401 010422 INTER:  ISZ     INRET    ; TERMINATION RETURN TO CALL+3
42 05402 010421         ISZ     INRET    ; FOR ACCEPTED INPUT
43 05403 006040 ONTER:  CMESS            ; TYPE CR,LF WITH MESS TO WAIT
44 05404 004743         MCRLF            ; FOR DEVICE READY.
45 05405 006047         CDATT            ; RETURN IS TO CALL + 1
46 05406 006044         CDISP            ; FOR NO INPUT, ONLY TERMINATION
47 05407 004743         MCRLF            ; (FOR ANSWER QUE. WITH AN OKAY).
48 05410 002413         JMP     @INRET   ; RETURN TO MAIN PROGRAM
49
50 05411 030474 TERMT:  LDA     2,HC13   ; THE TERMT ROUTINE LOOKS FOR TERM.
51 05412 112415         SUB#    0,2,SNR  ; IS AC0=13 ?
52 05413 001401         JMP     1,3      ; YES - IT IS NOT A TERM
53 05414 030472         LDA     2,HC15   ; NO
54 05415 112433         SUBZ#   0,2,SNC  ; IS AC0=<15 ?
55 05416 001401         JMP     1,3      ; NO, IT'S NOT A TERMINATOR
56 05417 030465         LDA     2,HC11   ; YES
57 05420 112432         SUBZ#   0,2,SZC  ; IS AC0=< 11 ?
58 05421 001401         JMP     1,3      ; YES - IT IS NOT A TERMINATOR
59 05422 001400         JMP     0,3      ; NO - IT IS A TERMINATOR, RETURN
60
61 05423 000000 INRET:  0                ; RETURN ADDRESS TO MAIN PROGRAM
62 05424 006332 XTXCO:  TXCCU            ; ADDRESS OF TEXTBUFFER COUNTER
63 05425 006403 XTXND:  TXEND            ; ADDRESS OF TEXT TERM ROUTINE
64 05426 000000 LASTN:  0                ; X77777 FOR FIRST/RUBOUT, ELSE = CHAR
65 05427 000000 DELRE:  0                ; RETURN ADDRESS FOR DELTE
```

```
| 0077  .MAIN
01
02                ; DECIMAL-NUMBER-INPUTROUTINE.
03
04 05430 054773 GETDC:  STA     3,INRET ; INITIALIZE
05 05431 126000         ADC     1,1     ; AC1:=177777
06 05432 046540         STA     1,@LAST4 ; SET LAST4
07 05433 044534         STA     1,DSIGN ; SET DSIGN TO +
08 05434 126400         SUB     1,1     ; AC1:= 0
09 05435 044534         STA     1,NUMB4 ; CLEAR NUMBER
10 05436 004611 GETD1:  JSR     GETCH   ; GET CHARACTER TO AC0
11 05437 004752         JSR     TERMT   ; TEST FOR TERMINATOR
12 05440 000456         JMP     TERMDC  ; IT IS A TERMINATOR
13 05441 004675         JSR     DELTE   ; IT'S NOT A TERMINATOR- IS IT A DEL ?
14 05442 000774         JMP     GETD1   ; IT IS A DELETE-CHAR
15 05443 004412         JSR     LETE4   ; IT'S NOT A DELETE-CHAR
16 05444 004510         JSR     CHRAN   ; PUT CHAR IN RANGE 0-9
17 05445 004527         JSR     OFTDC   ; TEST FOR OVERFLOW
18 05446 000403         JMP     DELDC   ; FIRST
19 05447 000402         JMP     DELDC   ; LAST IS RUBOUT
20 05450 004462         JSR     PLADC   ; ADD LAST4 TO MUMB4 * 10-DEC.
21 05451 024517 DELDC:  LDA     1,DCDIG ; STORE NEW DIGIT IN LAST4
22 05452 046520         STA     1,@LAST4
23 05453 004705 ECHO4:  JSR     TYPIN   ; TYPE NEW CHARACTER
24 05454 000762         JMP     GETD1   ; GET NEXT CHARACTER
25
26                ; SUBROUTINES USED BY GETDC.
27
28 05455 054426 LETE4:  STA     3,LERE4 ; LEGAL TEST
29 05456 034432         LDA     3,HC40
30 05457 111000         MOV     0,2     ; AC2:= CHAR
31 05460 024513         LDA     1,HC60
32 05461 116415         SUB#    0,3,SNR ; IS AC0 = 40 ?
33 05462 000771         JMP     ECHO4   ; YES - THE CHAR IS A SPACE
34 05463 034427         LDA     3,HC52  ; NO
35 05464 116432         SUBZ#   0,3,SZC ; IS AC0 > 52 ?
36 05465 004703         JSR     ILLEG   ; NO - THE CHAR IS NOT LEGAL
37 05466 034427         LDA     3,HC72  ; YES
38 05467 116033         ADCZ#   0,3,SNC ; IS AC0 < 72 ?
39 05470 004700         JSR     ILLEG   ; NO - NOT LEGAL
40 05471 034422         LDA     3,HC53  ; YES
41 05472 116415         SUB#    0,3,SNR ; IS AC0 >< 53 ?
42 05473 133000         ADD     1,2     ; NO - IT'S A PLUSSIGN, ADD 60
43 05474 034420         LDA     3,HC55  ; YES
44 05475 116415         SUB#    0,3,SNR ; IS AC0 >< 55 ?
45 05476 133000         ADD     1,2     ; NO - IT'S A MINUSSIGN ADD 60
46 05477 034474         LDA     3,HC60  ; YES
47 05500 156032         ADCZ#   2,3,SZC ; IS AC0 >= 60 ?
48 05501 004667         JSR     ILLEG   ; NO NOT LEGAL
49 05502 002401         JMP     @LERE4  ; YES - IT'S A VALID DIGIT
50 05503 000000 LERE4:  0
51
52 05504 000011 HC11:   11
53 05505 000013 HC13:   13
54 05506 000015 HC15:   15
55 05507 000030 HC30:   30
56 05510 000040 HC40:   40
57 05511 000044 HC44:   44
58 05512 000052 HC52:   52
59 05513 000053 HC53:   53
60 05514 000055 HC55:   55
61 05515 000072 HC72:   72
```

```
| 0078  .MAIN
01
02 05516 004456 TERMDC: JSR     OFTDC   ; TERMINATION, TEST OVERFLOW
03 05517 000664         JMP     ONTER   ; TERM IS FIRST
04 05520 000402         JMP     TERDC   ; LAST IS RUBOUT
05 05521 004411         JSR     PLADC   ; ADD LAST DIGIT TO NUMB4
06 05522 024447 TERDC:  LDA     1,NUMB4
07 05523 030444         LDA     2,DSIGN
08 05524 125015         MOV#    1,1,SNR ; IS AC1 = 0 ?
09 05525 000403         JMP     TERM1   ; YES - ZERO ALLWAYS POSITIVE
10 05526 151014         MOV#    2,2,SZR ; IS DSIGN = 0 ?
11 05527 124400         NEG     1,1     ; NO - IT IS A NEGATIVE NUMBER
12 05530 044075 TERM1:  STA     1,DIGIN ; YES - IT'S A POSITIVE NUMBER
13 05531 000650         JMP     INTER   ; OUTPUT TERM CHAR
14
15 05532 054421 PLADC:  STA     3,REPL4 ; PLACE LAST DIGIT/SIGN
16 05533 026437         LDA     1,@LAST4
17 05534 152400         SUB     2,2     ; AC2:= 0
18 05535 034756         LDA     3,HC53  ; LAST = + ?
19 05536 136415         SUB#    1,3,SNR ; NO
20 05537 000412         JMP     PLASI   ; YES, SET DSIGN
21 05540 152520         SUBZL   2,2     ; AC2:= 1
22 05541 034753         LDA     3,HC55  ; LAST = - ?
23 05542 136415         SUB#    1,3,SNR ; NO
24 05543 000406         JMP     PLASI   ; YES, SET DSIGN
25 05544 030425         LDA     2,NUMB4
26 05545 004413         JSR     MULTE   ; MULTIPLY NUMB4 WITH 10-DEC.
27 05546 133000         ADD     1,2     ; ADD LAST4 TO NUMB4*10
28 05547 050422         STA     2,NUMB4 ; PLACE NEW NUMBER
29 05550 002403         JMP     @REPL4
30 05551 050416 PLASI:  STA     2,DSIGN ; PLACE NEW SIGN
31 05552 002401         JMP     @REPL4  ; RETURN
32 05553 000000 REPL4:  0
33
34 05554 024417 CHRAN:  LDA     1,HC60  ; PUT CHAR IN RANGE 0 - 9 OR
35 05555 132400         SUB     1,2     ; PUT CHAR "+" TO 53 OR
36 05556 050412         STA     2,DCDIG ; PUT CHAR "-" TO 55
37 05557 001400         JMP     0,3     ; + AND - WAS ADDED 60 IN LEGAL TEST
38
39 05560 044406 MULTE:  STA     1,MULSA ; NUMBER TO BE MULT. BY 10 IN AC2
40 05561 145120         MOVZL   2,1     ; AC2=NUM, AC1=2*NUM
41 05562 125120         MOVZL   1,1     ; AC2=NUM, AC1=4*NUM
42 05563 133120         ADDZL   1,2     ; AC2=10*NUM, AC1=4*NUM
43 05564 024402         LDA     1,MULSA
44 05565 001400         JMP     0,3     ; RESULT IN AC2
45
46 05566 000000 MULSA:  0
47 05567 000000 DSIGN:  0               ; SIGNFLAG, 0=+, 1=-, 177777=NO SIGN=+
48 05570 000000 DCDIG:  0
49 05571 000000 NUMB4:  0
50 05572 005426 LAST4:  LASTN
51 05573 000060 HC60:   60
```

```
| 0079  .MAIN
01
02 05574 054450 OFTOC:  STA     3,RETOF ; OVERFLOW TEST FOR SIGN AND
03 05575 026775         LDA     1,@LAST4 ; (PREV*10)+LAST < LIMIT OX
04 05576 125102         MOVL    1,1,SZC ; IS THE LAST RUBBED OUT ?
05 05577 002445         JMP     @RETOF  ; OR IS IT THE FIRST ? YES, FIRST
06 05600 010444         ISZ     RETOF
07 05601 125102         MOVL    1,1,SZC
08 05602 002442         JMP     @RETOF  ; YES, RUBOUT
09 05603 034710         LDA     3,HC53  ; NO, IS LAST A + ?
10 05604 136415         SUB#    1,3,SNR ; NO
11 05605 000432         JMP     OFTSI   ; YES
12 05606 034706         LDA     3,HC55  ; IS LAST A - ?
13 05607 136415         SUB#    1,3,SNR ; NO
14 05610 000427         JMP     OFTSI   ; YES
15 05611 176400         SUB     3,3     ; AC3:= 0
16 05612 030755         LDA     2,DSIGN ; IS DSIGN = 177777
17 05613 151112         MOVL#   2,2,SZC ; NO
18 05614 054753         STA     3,DSIGN ; YES, SET FIRST +
19 05615 030754         LDA     2,NUMB4 ; AC2:=PREVIOUS (PREV)
20 05616 155120         MOVZL   2,3     ; PREV*2
21 05617 175120         MOVZL   3,3     ; PREV*4
22 05620 175112         MOVL#   3,3,SZC ; PREV*8>=65536 IF
23 05621 006556         JSR     @XILLG  ; PREV >= 8192
24 05622 173122         ADDZL   3,2,SZC ; PREV*10>=65540 IF
25 05623 006554         JSR     @XILLG  ; PREV >= 6554
26 05624 151112         MOVL#   2,2,SZC ; PREV*10>=32770 IF
27 05625 006552         JSR     @XILLG  ; PREV >= 3277
28 05626 133000         ADD     1,2     ; (PREV*10)+LAST>32767
29 05627 151113         MOVL#   2,2,SNC ; (<=32760)+0...9>32767 ?
30 05630 000412         JMP     OFTRE   ; NO, RETURN
31 05631 034736         LDA     3,DSIGN ; YES, TEST +- 32768,32769
32 05632 175005         MOV     3,3,SNR ; TEST SIGN
33 05633 006544         JSR     @XILLG  ; +32768, +32769
34 05634 151134         MOVZL#  2,2,SZR ; -32768, OK RETURN
35 05635 006542         JSR     @XILLG  ; -32769
36 05636 000404         JMP     OFTRE   ; RETURN
37
38 05637 030730 OFTSI:  LDA     2,DSIGN ; OVERFLOW TEST SIGN
39 05640 151113         MOVL#   2,2,SNC ; IS IT FIRST SIGN ? YES
40 05641 006536         JSR     @XILLG  ; NO
41 05642 010402 OFTRE:  ISZ     RETOF   ; PASS RUBOUT/FIRST RETURN
42 05643 002401         JMP     @RETOF  ; RETURN
43
44 05644 000000 RETOF:  0               ; RETURN ADDRESS
```

```
| 0080  .MAIN
01
02                ; CHARACTER-INPUT-ROUTINE
03
04 05645 056537 GETSC:  STA     3,@XINRT
05 05646 126000         ADC     1,1      ; AC1:=177777
06 05647 046440         STA     1,@LAST1 ; SET LAST1
07 05650 126400         SUB     1,1      ; AC1:= 0
08 05651 044435         STA     1,NUMSC  ; CLEAR NUMSC
09 05652 006530 GETS1:  JSR     @XGTCH   ; GET CHAR TO AC0
10 05653 006526         JSR     @XTRMT   ; IS IT A TERMINATOR ?
11 05654 000450         JMP     TERMSC   ; YES
12 05655 006523         JSR     @XDLTE   ; NO - IS IT A DEL CHAR ?
13 05656 000774         JMP     GETS1    ; YES
14 05657 004410         JSR     LETE1    ; NO - IS IT LEGAL ?
15 05660 004430         JSR     OFTSC    ; IT'S A LEGAL CHAR, TEST OVERFLOW
16 05661 000403         JMP     DELSC    ; FIRST
17 05662 000402         JMP     DELSC    ; LAST IS RUBOUT
18 05663 004436         JSR     PLASC    ; PLACE LAST IN NUMSC
19 05664 042423 DELSC:  STA     0,@LAST1 ; STORE NEW CHAR IN LAST1
20 05665 006516         JSR     @XTYPN   ; TYPE NEW CHARACTER
21 05666 000764         JMP     GETS1    ; GET NEXT INPUT
22
23                ; SUBROUTINES USED BY GETSC
24
25 05667 054413 LETE1:  STA     3,LERE1 ; LEGAL TEST
26 05670 030413         LDA     2,H1C40
27 05671 034414         LDA     3,H1C11
28 05672 024412         LDA     1,H1C33
29 05673 106415         SUB#    0,1,SNR ; IS AC0 = 33 ?
30 05674 002406         JMP     @LERE1  ; YES
31 05675 116415         SUB#    0,3,SNR ; NO - IS AC0 = 11 ?
32 05676 002404         JMP     @LERE1  ; YES
33 05677 112032         ADCZ#   0,2,SZC ; NO - IS AC0 >= 40 ?
34 05700 006477         JSR     @XILLG  ; NO
35 05701 002401         JMP     @LERE1  ; YES - IT IS A LEGAL CHAR
36 05702 000000 LERE1:  0
37
38 05703 000040 H1C40:  40
39 05704 000033 H1C33:  33
40 05705 000011 H1C11:  11
41 05706 000000 NUMSC:  0
42 05707 005426 LAST1:  LASTN
43
44 05710 026777 OFTSC:  LDA     1,@LAST1 ; OVERFLOW TEST
45 05711 125102         MOVL    1,1,SZC  ; IS THE LAST CHAR RUBBED OUT ?
46 05712 001400         JMP     0,3      ; OR IS IT THE FIRST ? YES, FIRST
47 05713 125102         MOVL    1,1,SZC
48 05714 001401         JMP     1,3      ; YES, RUBOUT
49 05715 030771         LDA     2,NUMSC  ; NUMSC = 0 FOR FIRST
50 05716 151004         MOV     2,2,SZR  ; IS IT THE FIRST CHAR ?
51 05717 006460         JSR     @XILLG   ; NO, OVERFLOW
52 05720 001402         JMP     2,3      ; YES, RETURN
53
54 05721 032766 PLASC:  LDA     2,@LAST1 ; PLACE LAST CHAR
55 05722 050764         STA     2,NUMSC
56 05723 001400         JMP     0,3
57
58 05724 004764 TERMSC: JSR     OFTSC    ; TERMINATION, TEST OVERFLOW
59 05725 002461         JMP     @XONTR   ; TERM IS FIRST
60 05726 000402         JMP     TERSC    ; LAST IS RUBOUT
61 05727 004772         JSR     PLASC    ; STORE LAST CHAR IN NUMSC
62 05730 024756 TERSC:  LDA     1,NUMSC  ; MOVE CHAR TO DIGIN
63 05731 044075         STA     1,DIGIN
64 05732 002453         JMP     @XINTR   ; OUTPUT TERM CHAR
```

```
| 0081  .MAIN
01
02                ; OCTAL-NUMBER-INPUTROUTINE
03
04 05733 056451 GETOK:  STA     3,@XINRT
05 05734 126000         ADC     1,1     ; AC1:=177777
06 05735 046536         STA     1,@LAST2
07 05736 126400         SUB     1,1     ; AC1:= 0
08 05737 044505         STA     1,NUMB2
09 05740 006442 GETO1:  JSR     @XGTCH  ; GET CHAR TO AC0
10 05741 006440         JSR     @XTRMT  ; IS IT A TERMINATOR ?
11 05742 000503         JMP     TERMOK  ; YES
12 05743 006435         JSR     @XDLTE  ; NO - IS IT A DEL CHAR ?
13 05744 000774         JMP     GETO1   ; YES
14 05745 004412         JSR     LETE2   ; NO - IS IT LEGAL ?
15 05746 004441         JSR     CHRA2   ; IT IS A LEGAL DIGIT
16 05747 004445         JSR     OFTE2   ; TEST FOR OVERFLOW
17 05750 000403         JMP     DELOK   ; FIRST
18 05751 000402         JMP     DELOK   ; LAST IS RUBOUT
19 05752 004460         JSR     PLAOK   ; ADD LAST2 TO NUMB2 * 8-DEC
20 05753 024423 DELOK:  LDA     1,OKDIG ; STORE NEW DIGIT IN LAST2
21 05754 046467         STA     1,@LAST2
22 05755 006426 ECHO2:  JSR     @XTYPN  ; TYPE NEW CHARACTER
23 05756 000762         JMP     GETO1   ; GET NEXT CHARACTER
24
25                ; SUBROUTINES USED BY GETOK
26
27 05757 054413 LETE2:  STA     3,LERE2 ; LEGAL TEST
28 05760 034413         LDA     3,H2C40
29 05761 116415         SUB#    0,3,SNR ; IS AC0 = 40 ?
30 05762 000773         JMP     ECHO2   ; YES - THE CHAR IS A SPACE
31 05763 034411         LDA     3,H2C60 ; NO
32 05764 116032         ADCZ#   0,3,SZC ; IS AC0 >= 60 ?
33 05765 006412         JSR     @XILLG  ; NO - NOT LEGAL
34 05766 034407         LDA     3,H2C70 ; YES
35 05767 116033         ADCZ#   0,3,SNC ; IS AC0 < 70 ?
36 05770 006407         JSR     @XILLG  ; NO - NOT LEGAL
37 05771 002401         JMP     @LERE2  ; YES - IT IS A VALID DIGIT
38 05772 000000 LERE2:  0
39
40 05773 000040 H2C40:  40
41 05774 000060 H2C60:  60
42 05775 000070 H2C70:  70
43 05776 000000 OKDIG:  0
44
45 05777 005370 XILLG:  ILLEG
46 06000 005336 XDLTE:  DELTE
47 06001 005411 XTRMT:  TERMT
48 06002 005247 XGTCH:  GETCH
49 06003 005360 XTYPN:  TYPIN
50 06004 005423 XINRT:  INRET
51 06005 005401 XINTR:  INTER
52 06006 005403 XONTR:  ONTER
53
54 06007 024765 CHRA2:  LDA     1,H2C60
55 06010 111000         MOV     0,2
56 06011 132400         SUB     1,2     ; PUT CHAR IN RANGE 0-7
57 06012 050764         STA     2,OKDIG
58 06013 001400         JMP     0,3
```

```
| 0082  .MAIN
01
02 06014 054415 OFTE2:  STA     3,REOF2 ; OVERFLOW TEST
03 06015 030427         LDA     2,NUMB2
04 06016 026425         LDA     1,@LAST2 ; IS THE LAST DIGIT RUBBED OUT ?
05 06017 125102         MOVL    1,1,SZC ; OR IS IT THE FIRST ?
06 06020 002411         JMP     @REOF2  ; YES, FIRST
07 06021 010410         ISZ     REOF2
08 06022 125102         MOVL    1,1,SZC
09 06023 002406         JMP     @REOF2  ; YES, RUBOUT
10 06024 034416         LDA     3,OF2CO ; NO
11 06025 156433         SUBZ#   2,3,SNC ; IS NUMB2 <= 17777 ?
12 06026 006751         JSR     @XILLG  ; NO - OVERFLOW
13 06027 010402         ISZ     REOF2   ; YES
14 06030 002401         JMP     @REOF2  ; RETURN
15 06031 000000 REOF2:  0
16
17 06032 026411 PLAOK:  LDA     1,@LAST2 ; PLACE LAST DIGIT
18 06033 030411         LDA     2,NUMB2
19 06034 151120         MOVZL   2,2     ; MULTIPLY NUMB2 WITH 8-DEC
20 06035 151120         MOVZL   2,2
21 06036 151120         MOVZL   2,2
22 06037 133000         ADD     1,2     ; ADD LAST DIGIT
23 06040 050404         STA     2,NUMB2
24 06041 001400         JMP     0,3
25
26 06042 017777 OF2CO:  17777
27 06043 005426 LAST2:  LASTN
28 06044 000000 NUMB2:  0
29
30 06045 004747 TERMOK: JSR     OFTE2   ; TERMINATION, TEST OVERFLOW
31 06046 002740         JMP     @XCNTR  ;TERM IS FIRST
32 06047 000402         JMP     TEROK   ; LAST IS RUBOUT
33 06050 004762         JSR     PLAOK   ; ADD LAST DIGIT TO NUMB2
34 06051 024773 TEROK:  LDA     1,NUMB2
35 06052 044075         STA     1,DIGIN
36 06053 002732         JMP     @XINTR  ; OUTPUT TERM CHAR
```

```
I 0083  .MAIN
01
02                ; BINARY-NUMBER-INPUTROUTINE
03
04 06054 056730 GETBI:  STA     3,@XINRT
05 06055 126000         ADC     1,1     ; AC1:=177777
06 06056 046474         STA     1,@LAST3
07 06057 126400         SUB     1,1     ; AC1:= 0
08 06060 044473         STA     1,NUMB3
09 06061 006721 GETB1:  JSR     @XGTCH  ; GET CHAR TO AC0
10 06062 006717         JSR     @XTRMT  ; IS IT A TERMINATOR ?
11 06063 000471         JMP     TERMBI  ; YES
12 06064 006714         JSR     @XDLTE  ; NO - IS IT A DEL CHAR ?
13 06065 000774         JMP     GETB1   ; YES
14 06066 004412         JSR     LETE3   ; NO - IS IT LEGAL ?
15 06067 004431         JSR     CHRA3   ; IT IS A LEGAL DIGIT
16 06070 004435         JSR     OFTE3   ; TEST FOR OVERFLOW
17 06071 000403         JMP     DELBI   ; FIRST
18 06072 000402         JMP     DELBI   ; LAST IS RUBCUT
19 06073 004450         JSR     PLABI   ; ADD LAST3 TO NUMB3 * 2-DEC
20 06074 024423 DELBI:  LDA     1,BIDIG ; STORE NEW DIGIT IN LAST3
21 06075 046455         STA     1,@LAST3
22 06076 006705 ECHO3:  JSR     @XTYPN  ; TYPE NEW CHARACTER
23 06077 000762         JMP     GETB1   ; GET NEXT CHARACTER
24
25                ; SUBROUTINES USED BY GETBI
26
27 06100 054413 LETE3:  STA     3,LERE3 ; LEGAL TEST
28 06101 034413         LDA     3,H3C40
29 06102 116415         SUB#    0,3,SNR ; IS AC0 = 40 ?
30 06103 000773         JMP     ECHO3   ; YES - THE CHAR IS A SPACE
31 06104 034411         LDA     3,H3C60 ; NO
32 06105 116032         ADCZ#   0,3,SZC ; IS AC0 >= 60 ?
33 06106 006671         JSR     @XILLG  ; NO - NOT LEGAL
34 06107 034407         LDA     3,H3C62 ; YES
35 06110 116033         ADCZ#   0,3,SNC; IS AC0 < 62 ?
36 06111 006666         JSR     @XILLG  ; NO - NOT LEGAL
37 06112 002401         JMP     @LERE3  ; YES
38 06113 000000 LERE3:  0
39
40 06114 000040 H3C40:  40
41 06115 000060 H3C60:  60
42 06116 000062 H3C62:  62
43 06117 000000 BIDIG:  0
44
45
46 06120 024775 CHRA3:  LDA     1,H3C60 ; PUT CHAR IN RANGE 0-1
47 06121 111000         MOV     0,2
48 06122 132400         SUB     1,2
49 06123 050774         STA     2,BIDIG
50 06124 001400         JMP     0,3
```

```
| 0084  .MAIN
01
02 06125 054415 OFTE3:  STA     3,REOF3 ; OVERFLOW TEST
03 06126 030425         LDA     2,NUMB3
04 06127 026423         LDA     1,@LAST3 ; IS THE LAST DIGIT RUBBED OUT ?
05 06130 125102         MOVL    1,1,SZC ; OR IS IT THE FIRST ?
06 06131 002411         JMP     @REOF3  ; YES, FIRST
07 06132 010410         ISZ     REOF3
08 06133 125102         MOVL    1,1,SZC
09 06134 002406         JMP     @REOF3  ; YES, RUBOUT
10 06135 034414         LDA     3,OF3CO ; NO
11 06136 156433         SUBZ#   2,3,SNC ; IS NUMB3 <= 177 ?
12 06137 006640         JSR     @XILLG  ; NO - OVERFLOW
13 06140 010402         ISZ     REOF3   ; YES
14 06141 002401         JMP     @REOF3  ; RETURN
15 06142 000000 REOF3:  0
16
17 06143 026407 PLABI:  LDA     1,@LAST3 ; PLACE LAST DIGIT
18 06144 030407         LDA     2,NUMB3
19 06145 151120         MOVZL   2,2     ; MULTIPLY WITH 2-DEC
20 06146 133000         ADD     1,2     ; ADD LAST DIGIT
21 06147 050404         STA     2,NUMB3
22 06150 001400         JMP     0,3
23
24 06151 000177 OF3CO:  177
25 06152 005426 LAST3:  LASTN
26 06153 000000 NUMB3:  0
27
28 06154 004751 TERMBI: JSR     OFTE3   ; TERMINATION, TEST OVERFLOW
29 06155 002631         JMP     @XCNTR  ; TERM IS FIRST
30 06156 000402         JMP     TERBI   ; LAST IS RUBOUT
31 06157 004764         JSR     PLABI   ; ADD LAST DIGIT TO NUMB3
32 06160 024773 TERBI:  LDA     1,NUMB3 ; MOVE THE BINARY NUMBER TO DIGIN
33 06161 044075         STA     1,DIGIN
34 06162 002623         JMP     @XINTR  ; OUTPUT TERM CHAR
```

```
| 0085  .MAIN
01
02                ; TEXT-INPUT-ROUTINE
03
04 06163 056555 GETTX:  STA     3,@YINRT
05 06164 126000         ADC     1,1       ; AC1:=177777
06 06165 046543         STA     1,@LAST5
07 06166 126400         SUB     1,1       ; AC1:= 0
08 06167 044543         STA     1,TXCOU
09 06170 006543 GETT1:  JSR     @YGTCH    ; GET CHAR TO AC0
10 06171 006543         JSR     @YTRMT    ; IS IT A TERMINATOR ?
11 06172 000551         JMP     TERMXT    ; YES
12 06173 006542         JSR     @YDLTE    ; NO - IS IT A DEL CHAR ?
13 06174 000774         JMP     GETT1     ; YES
14 06175 004410         JSR     LETE5     ; NO - IS IT LEGAL ?
15 06176 004423         JSR     OFTTX     ; TEST FOR OVERFLOW
16 06177 000403         JMP     DELTX     ; FIRST
17 06200 000402         JMP     DELTX     ; LAST IS RUBOUT
18 06201 004505         JSR     PLATX     ; STORE LAST5 IN TEXTBUFFER
19 06202 042526 DELTX:  STA     0,@LAST5  ; STORE NEW CHAR IN LAST5
20 06203 006533         JSR     @YTYPN    ; TYPE NEW CHAR
21 06204 000764         JMP     GETT1     ; GET NEXT CHARACTER
22
23                ; SUBROUTINES USED BY GETTX
24
25 06205 054410 LETE5:  STA     3,LERE5 ; LEGAL TEST
26 06206 034410         LDA     3,H5C11
27 06207 030410         LDA     2,H5C40
28 06210 116415         SUB#    0,3,SNR ; IS AC0 = 11 ?
29 06211 002404         JMP     @LERE5  ; YES
30 06212 112032         ADCZ#   0,2,SZC ; NO - IS AC0 >= 40 ?
31 06213 006524         JSR     @YILLG  ; NO - THE CHAR IS NOT LEGAL
32 06214 002401         JMP     @LERE5  ; YES
33 06215 000000 LERE5:  0
34
35 06216 000011 H5C11:  11
36 06217 000040 H5C40:  40
37 06220 000117 H5C79:  117               ; TEXTBUFFER LENGTH-1, OKTAL
38
39 06221 026507 OFTTX:  LDA     1,@LAST5 ; OVERFLOW TEST
40 06222 125102         MOVL    1,1,SZC ; IS THE LAST CHAR RUBBED OUT ?
41 06223 001400         JMP     0,3     ; OR IS IT THE FIRST ? YES, FIRST
42 06224 125102         MOVL    1,1,SZC
43 06225 001401         JMP     1,3     ; YES, RUBOUT
44 06226 024504         LDA     1,TXCOU ; NO
45 06227 030771         LDA     2,H5C79
46 06230 132433         SUBZ#   1,2,SNC ; IS TXCOU <= 79-DEC
47 06231 006506         JSR     @YILLG  ; NO - OVERFLOW
48 06232 001402         JMP     2,3     ; YES, RETURN
49
50 06233 006234 TEXIA:  .+1               ; ADDRESS OF TEXTBUFFER
51       000051 TEXIN:  .BLK 51           ; TEXTBUFFER 80 BYTES + CR,LF
52 06305 000000 TEXEN:  0                 ; END OF TEXTBUFFER: NUL CHAR
```

```
| 0086  .MAIN
01
02                                          ; PLACE LAST CHAR
03 06306 054421 PLATX:  STA     3,REPL5 ; LAST5 TO CHAHA OR
04 06307 026421         LDA     1,@LAST5 ; LAST5 + CHAHA TO BUFFER
05 06310 010422         ISZ     TXCOU   ; INCREMENT CHARACTER-COUNTER
06 06311 030421         LDA     2,TXCOU
07 06312 151213         MOVR#   2,2,SNC ; IS TXCOU EVEN ?
08 06313 000403         JMP     STOTX   ; YES
09 06314 044415         STA     1,CHAHA ; NO - MOVE LAST5 TO CHAHA
10 06315 002412         JMP     @REPL5
11 06316 030413 STOTX:  LDA     2,CHAHA ; CHAHA IS CHAR HALF BUFFER
12 06317 125320         MOVZS   1,1
13 06320 147000         ADD     2,1     ; AC1 = "LAST5,CHAHA"
14 06321 030411         LDA     2,TXCOU ; CALCULATE ADRESS OF BUFFEREND+1
15 06322 151220         MOVZR   2,2
16 06323 034710         LDA     3,TEXIA
17 06324 173000         ADD     3,2     ; AC2:= TEXIN+TXCOU/2
18 06325 045377         STA     1,-1,2  ; STORE LAST TWO CHAR INTO BUFFEREND+1
19 06326 002401         JMP     @REPL5
20 06327 000000 REPL5:  0
21
22 06330 005426 LAST5:  LASTN
23 06331 000000 CHAHA:  0               ; THE CHAR BEFORE LAST5
24 06332 000000 TXCOU:  0               ; TEXT COUNTER BYTE ADDRESS
25
26 06333 005247 YGTCH:  GETCH
27 06334 005411 YTRMT:  TERMT
28 06335 005336 YDLTE:  DELTE
29 06336 005360 YTYPN:  TYPIN
30 06337 005370 YILLG:  ILLEG
31 06340 005423 YINRT:  INRET
32 06341 005401 YINTR:  INTER
33 06342 005403 YONTR:  ONTER
34
35 06343 004656 TERMXT: JSR     OFTTX   ; TERMINATION, TEST OVERFLOW
36 06344 002776         JMP     @YONTR  ; TERM IS FIRST
37 06345 000402         JMP     TERTX   ; LAST IS RUBOUT
38 06346 004740         JSR     PLATX   ; PLACE THE CHAR BEFORE TERM CHAR
39 06347 024763 TERTX:  LDA     1,TXCOU ; IN LAST5
40 06350 125213         MOVR#   1,1,SNC ; IS TXCOU EVEN ?
41 06351 000417         JMP     NBEVEN  ; YES
42 06352 020443         LDA     0,H5C15 ; NO
43 06353 101320         MOVZS   0,0
44 06354 030755         LDA     2,CHAHA
45 06355 113000         ADD     0,2     ; AC2 = "CR,CHAR"
46 06356 010754         ISZ     TXCOU   ; INCREMENT CHAR COUNT FOR CR
47 06357 004413         JSR     COUDI
48 06360 004415         JSR     STABU   ; STORE AC2 INTO BUFFER
49 06361 030435         LDA     2,H5C12 ; AC2 = " 0,LF"
50 06362 004410         JSR     COUDI
51 06363 125400         INC     1,1     ; INCREMENT BUFF ADDR FOR 0,LF
52 06364 004411         JSR     STABU   ; STORE AC2 INTO BUFFER
53 06365 030646 OUT5:   LDA     2,TEXIA
54 06366 050075         STA     2,DIGIN ; ADDRESS OF TEXTBUFFER
55 06367 002752         JMP     @YINTR  ; OUTPUT TERM CHAR
56
57 06370 004413 NBEVEN: JSR     TXEND   ; STORE LF,CR,0,0
58 06371 000774         JMP     OUT5    ; TERMINATE
```

```
| 0087  .MAIN
01
02 06372 024740 COUDI:  LDA     1,TXCOU
03 06373 125220         MOVZR   1,1     ; DIVIDE TXCOU WITH 2
04 06374 001400         JMP     0,3
05
06 06375 054405 STABU:  STA     3,RET5
07 06376 034635         LDA     3,TEXIA ; CALCULATE ADRESS OF BUFFEREND
08 06377 137000         ADD     1,3
09 06400 051777         STA     2,-1,3  ; STORE AC2 INTO BUFFER
10 06401 002401         JMP     @RET5
11 06402 000000 RET5:   0
12
13 06403 054411 TXEND:  STA     3,TXNDR
14 06404 030413         LDA     2,HLFCR ; AC2 = "LF,CR"
15 06405 004765         JSR     COUDI
16 06406 125400         INC     1,1
17 06407 004766         JSR     STABU   ; STORE AC2 INTO BUFFER
18 06410 125400         INC     1,1
19 06411 152400         SUB     2,2     ; AC2:= 0
20 06412 004763         JSR     STABU   ; STORE "0,0" INTO BUFFER
21 06413 002401         JMP     @TXNDR
22 06414 000000 TXNDR:  0
23
24 06415 000015 H5C15:  15
25 06416 000012 H5C12:  12
26 06417 005015 HLFCR:  5015
```

```
1 0088  .MAIN
01
02                      ;TESTLOOP ROUTINE
03                      ;CALL   SETPX    ;SETP0,SETP1,SETP2
04                      ;                ;PROGRAM LOOP
05                      ;       EHALT    ;ERROR HALT ROUTINE
06                      ;       LOOP     ;CYCLE LOOP ROUTINE
07                      ;
08                      ;                ;NEXT TEST CYCLE
09
10              ;SETPX: ;IORST AND SET # OF LOOPS IN
11                      ;FIRST CYCLUS    ERROR CYCLUS IF SWITCH 0
12                      ;SETP0  10!0     10!1
13                      ;SETP1  10!1     10!2
14                      ;SETP2  10!2     10!2
15              ;EHALT: ;IF NOT FIRST LOOP WITH ERROR IN A CYCLE:
16                      ;       DO NOTHING, LOOP
17                      ;IF FIRST LOOP WITH ERROR IN A CYCLE:
18                      ;       PRINT AC0, AC1, AC2 (NOT ON DIS) AND
19                      ;       PRINT PC XXXXXX IF NOT SWITCH 10 (INHIBIT PRINT)
20                      ;IF FIRST LOOP WITH ERROR AT ALL:
21                      ;       HALT IN EACH ERROR WITH AC3=PC
22                      ;       OTHER ACS RELEVANT INFO.
23                      ;ELSE DO NOTHING, LOOP
24              ;LOOP:  ;IF CYCLE NOT FINISHED (# OF LOOP NOT FINISHED):
25                      ;       IF NO ERRORS AT ALL UNTIL NOW:
26                      ;               IORST, LOOP
27                      ;       IF ERROR AND SWITCH 0 = 0:
28                      ;               PRINT FAILURE RATE OF LAST CYCLE IF
29                      ;               SWITCH 11 = 1 AND SWITCH 10 = 0
30                      ;               PROCEED TO NEXT TEST CYCLE
31                      ;       ELSE:   IORST, LOOP
32                      ;IF CYCLE FINISHED:
33                      ;       IF NO ERRORS AT ALL:
34                      ;               PROCEED TO NEXT TEST CYCLE
35                      ;       ELSE:   PRINT FAILURE RATE OF LAST CYCLE IF
36                      ;               SWITCH 11 = 1 AND SWITCH 10 = 0
37                      ;       IF SWITCH 0 = 0:
38                      ;               PROCEED TO NEXT TEST CYCLE
39                      ;       ELSE:   IORST, LOOP.
40 06420 054522 ENTP0:  STA     3,LOOPR ;INITIALIZE EACH TEST
41 06421 034504         LDA     3,ITRP1
42 06422 054505         STA     3,ITRAG
43 06423 176520         SUBZL   3,3      ;AC3:= 1
44 06424 000412         JMP     ENTCO
45 06425 054515 ENTP1:  STA     3,LOOPR
46 06426 034500         LDA     3,ITRP2
47 06427 054500         STA     3,ITRAG
48 06430 034475         LDA     3,ITRP1
49 06431 000405         JMP     ENTCO
50 06432 054510 ENTP2:  STA     3,LOOPR
51 06433 034473         LDA     3,ITRP2
52 06434 054473         STA     3,ITRAG
53 06435 034471         LDA     3,ITRP2
54 06436 054472 ENTCO:  STA     3,ITR
55 06437 054472         STA     3,ITRCT
56 06440 176400         SUB     3,3      ;AC3:= 0
57 06441 054471         STA     3,EFLAG ;SET FIRST ERROR FLAG = 0
58 06442 054471         STA     3,ERRCT ;SET ERROR COUNT = 0
59 06443 054471         STA     3,ERBCT ;SET ERROR BUFFER COUNT = 0
60 06444 054471         STA     3,EBFLG ;SET FIRST ERROR FLAG BUFFER = 0
61 06445 034501         LDA     3,SETSW ;SET PRINT INHIBIT ON SW 10
62 06446 056477         STA     3,@ISTAC ;IN FUNCTION
63 06447 006127         RESET            ;I/O RESET
64 06450 002472         JMP     @LOOPR  ;LOOP ITERATE RETURN
```

```
| 0089  .MAIN
01
02 06451 054465 CYCLE:  STA     3,RETUR ;END OF TEST ITERATION ROUTINE
03 06452 050465         STA     2,CSAV2
04 06453 044465         STA     1,CSAV1 ;SAVE THE ACS'
05 06454 040465         STA     0,CSAV0
06 06455 020457         LDA     0,ERBCT ;ADD BUFFER COUNT
07 06456 024455         LDA     1,ERRCT ;TO ERROR COUNT
08 06457 107000         ADD     0,1
09 06460 044453         STA     1,ERRCT
10 06461 020454         LDA     0,EBFLG ;MOVE FIRST FLAG
11 06462 040450         STA     0,EFLAG ;BUFFER TO FLAG
12 06463 014446         DSZ     ITRCT
13 06464 000416         JMP     CYCTS   ;NOT N TIMES ITERATED, LOOP
14 06465 030445         LDA     2,EFLAG ;CYCLUS FINISHED
15 06466 151005         MOV     2,2,SNR ;ERRORS AT ALL ?
16 06467 000432         JMP     NOEX    ;NO ERRORS
17 06470 004457         JSR     FRATE   ;YES, PRINT FAILURE RATE
18 06471 102400         SUB     0,0
19 06472 040441         STA     0,ERRCT ;RESET ERROR COUNT
20 06473 034434         LDA     3,ITRAG
21 06474 054434         STA     3,ITR
22 06475 054434         STA     3,ITRCT
23 06476 006073         CRESW           ;READS 2 ROUTINE
24 06477 151112         MOVL#   2,2,SZC ;SWITCH 0 ?
25 06500 000412         JMP     CYMOR   ;(1) = LOOP IN ERROR
26 06501 000420         JMP     NOEX    ;(0) = PROCEED TO NEXT TEST
27
28 06502 034430 CYCTS:  LDA     3,EFLAG ;LOOP FINISHED
29 06503 175005         MOV     3,3,SNR ;ERRORS UNTIL NOW ?
30 06504 000406         JMP     CYMOR   ;NO, LOOP
31 06505 006073         CRESW           ;YES, READS 2 ROUTINE
32 06506 151112         MOVL#   2,2,SZC ;SWITCH 0 ?
33 06507 000403         JMP     CYMOR   ;(1) = LOOP IN ERROR
34 06510 004437         JSR     FRATE   ;PRINT FAILURE RATE
35 06511 000410         JMP     NOEX    ;(0)=PROCEED TO NEXT TEST
36
37 06512 006127 CYMOR:  RESET           ;I/O RESET
38 06513 176400         SUB     3,3     ;AC3:=0
39 06514 054420         STA     3,ERBCT ;RESET ERROR BUFFER COUNTER
40 06515 020424         LDA     0,CSAV0
41 06516 024422         LDA     1,CSAV1
42 06517 030420         LDA     2,CSAV2 ;RESTORE AC'S
43 06520 002422         JMP     @LOOPR  ;LOOP MORE
44
45 06521 020420 NOEX:   LDA     0,CSAV0 ;EXIT TO NEXT TEST
46 06522 024416         LDA     1,CSAV1
47 06523 030414         LDA     2,CSAV2 ;RESTORE AC'S
48 06524 002412         JMP     @RETUR  ;PROCEED TO NEXT TEST
49
50 06525 000012 ITRP1:  12              ;1011 LOOP CONSTANT
51 06526 000144 ITRP2:  144             ;1012 LOOP CONSTANT
52 06527 000000 ITRAG:  0               ;# OF LOOP IF ERROR
53 06530 000000 ITR:    0               ;# OF LOOP IN CYCLUS ACTUAL
54 06531 000000 ITRCT:  0               ;LOOP COUNTER
55 06532 000000 EFLAG:  0               ;FIRST ERROR FLAG FOR ALL CYCLES
56 06533 000000 ERRCT:  0               ;ERROR COUNTER
57 06534 000000 ERBCT:  0               ;ERROR BUFFER COUNTER
58 06535 000000 EBFLG:  0               ;FIRST ERROR FLAG BUFFER
59 06536 000000 RETUR:  0
60 06537 000000 CSAV2:  0
61 06540 000000 CSAV1:  0
62 06541 000000 CSAV0:  0
63 06542 000000 LOOPR:  0
64 06543 000010 ERHSW:  10              ;SWITCH 12
```

```
| 0090  .MAIN
01
02 06544 000437 IBZOT:  BZOUT
03 06545 000574 ISTAC:  SETAC
04 06546 006073 SETSW:  CRESW           ;FOR PRINT INHIBIT ROUTINE
05
06 06547 054432 FRATE:  STA     3,FRATR ;PRINT FAILURE RATE
07 06550 006073         CRESW           ;IF SWITCH 10 = 0
08 06551 024433         LDA     1,INHSW ;AND SWITCH 11 = 1
09 06552 133414         AND#    1,2,SZR
10 06553 002426         JMP     @FRATR  ;INHIBIT PRINTOUT
11 06554 024431         LDA     1,FRASW
12 06555 133415         AND#    1,2,SNR
13 06556 000420         JMP     NORAT   ;PRINT CRLF ONLY
14 06557 020423         LDA     0,CH40
15 06560 006041         CCHAR
16 06561 024752         LDA     1,ERRCT
17 06562 030744         LDA     2,ITRP2
18 06563 006066         MULTI           ;AC1*AC2
19 06564 030744         LDA     2,ITR
20 06565 034744         LDA     3,ITRCT
21 06566 172400         SUB     3,2     ;AC2:=# OF LOOPS TILL NOW
22 06567 006070         DIVID           ;AC0,AC1/AC2
23 06570 006053         CTDEC
24 06571 006057         CDDEC           ;PRINT VALUE
25 06572 020411         LDA     0,PCENT ;EXAMPLE:       89%
26 06573 006045         CDOUT           ;VALUE=ERRCT*100/(ITR-ITRCT)
27 06574 006041         CCHAR
28 06575 006047         CDATT
29 06576 006040 NORAT:  CMESS           ;PRINT CRLF WITH MESS TO WAIT
30 06577 004743         MCRLF           ;FOR DEVICE READY BEFORE NEXT IORST.
31 06600 002401         JMP     @FRATR
32 06601 000000 FRATR:  0
33 06602 000040 CH40:   40
34 06603 000245 PCENT:  245
35 06604 000040 INHSW:  40              ;SWITCH 10
36 06605 000020 FRASW:  20              ;SWITCH 11
37
38               MHEAD: .TXT !AC0<11>AC1<11>AC2<15><12>PC<40>!  ;"AC0<11>AC1<11>AC
   06606 041501
   06607 004460
   06610 041501
   06611 004461
   06612 041501
   06613 006462
   06614 050012
   06615 020103
   06616 000000
39
40               DHEAD: .TXT !PC<40>!   ;"PC<40>"
   06617 041520
   06620 000040
41
42 06621 054715 ERROR:  STA     3,RETUR ;ERROR SUBROUTINE
43 06622 034711         LDA     3,ERRCT
44 06623 175005         MOV     3,3,SNR ;FIRST ERROR LOOP ?
45 06624 000404         JMP     ERR01
46 06625 010707 ERRET:  ISZ     ERBCT   ;COUNT BUFFER ERROR COUNTER
47 06626 002710         JMP     @RETUR  ;EXIT
48 06627 002707         JMP     @RETUR  ;EXIT IF SKIP
```

```
| 0091  .MAIN
01
02 06630 040711 ERR01:  STA     0,CSAV0 ;FIRST ERROR LOOP !
03 06631 044707          STA     1,CSAV1 ;(IN FIRST CYCLUS
04 06632 050705          STA     2,CSAV2 ;OR ERROR CYCLUS)
05 06633 006073          CRESW           ;READS 2 ROUTINE
06 06634 024750          LDA     1,INHSW ;SWITCH 10
07 06635 133414          AND#    1,2,SZR
08 06636 000455          JMP     ERR03   ;INHIBIT PRINTOUT
09 06637 006043          CCRLF
10 06640 024701          LDA     1,CSAV0
11 06641 006052          CTOCT
12 06642 024676          LDA     1,CSAV1
13 06643 006052          CTOCT
14 06644 024673          LDA     1,CSAV2
15 06645 006052          CTOCT           ;PRINT AC'S ONLY AT TTY, LPT
16 06646 006043          CCRLF           ;PRINT CARRIAGE
17 06647 006040          CMESS           ;PRINT HEADER
18 06650 006606          MHEAD
19 06651 006046          CDICL
20 06652 006044          CDISP
21 06653 006617          DHEAD
22 06654 020662          LDA     0,RETUR
23 06655 126000          ADC     1,1
24 06656 107000          ADD     0,1
25 06657 006052          CTOCT
26 06660 006056          CDOCT           ;PRINT PC OF ERROR
27 06661 006663          JSR     @IBZOT  ;WAIT FOR LPT/TTY BEFORE NEXT IORST
28 06662 020650          LDA     0,EFLAG
29 06663 101005          MOV     0,0,SNR
30 06664 000406          JMP     ERR02   ;FIRST ERROR LOOP AT ALL
31 06665 006047          CDATT
32 06666 020653          LDA     0,CSAV0
33 06667 024651          LDA     1,CSAV1
34 06670 030647          LDA     2,CSAV2 ;RESTORE ACS
35 06671 000734          JMP     ERRET
36
37 06672 006050 ERR02:  CHAAT           ;FIRST ERROR LOOP AT ALL: HALT
38 06673 126000          ADC     1,1     ;AC1:=177777
39 06674 044641          STA     1,EBFLG ;SET FIRST ERROR BUFFER FLAG
40 06675 006073          CRESW           ;READS 2 ROUTINE
41 06676 034645          LDA     3,ERHSW
42 06677 157404          AND     2,3,SZR ;SW 12 ?
43 06700 000407          JMP     ERRNH   ;NO HALT
44 06701 034635          LDA     3,RETUR
45 06702 137000          ADD     1,3     ;ERROR, AC3=PC OF ERROR
46 06703 030634          LDA     2,CSAV2
47 06704 024634          LDA     1,CSAV1
48 06705 020634          LDA     0,CSAV0
49 06706 063077          HALT            ;OPERATOR-SET SWITCHES!
50 06707 020632 ERRNH:  LDA     0,CSAV0
51 06710 024630          LDA     1,CSAV1
52 06711 030626          LDA     2,CSAV2
53 06712 000713          JMP     ERRET
54
55 06713 020617 ERR03:  LDA     0,EFLAG ;NO PRINTING
56 06714 101005          MOV     0,0,SNR
57 06715 000755          JMP     ERR02   ;FIRST ERROR LOOP AT ALL
58 06716 020623          LDA     0,CSAV0
59 06717 024621          LDA     1,CSAV1
60 06720 030617          LDA     2,CSAV2
61 06721 000704          JMP     ERRET
```

```
I 0092  .MAIN
01
02                          ;GET ARGUMENT STATUS
03                          ;AND CHECK THAT ALL OF
04                          ;THE BITS ARE PRESENT IN THE ACTUAL STATUS
05                          ;RETURN+2 IF BITS ARE PRESENT, OTHERWISE +1
06                          ;EXIT WITH AC1=EXPECTED, AC0=ACTUAL STATUS
07                          ;
08                          ;CALL   STATA
09                          ;       ARG
10
11 06722 054412 XSTAA:  STA     3,RXAST
12 06723 010411         ISZ     RXAST
13 06724 025400         LDA     1,0,3   ;AC1=EXPECTED BITS
14 06725 060461 SDEV1:  DIA     0,DEV
15 06726 040407         STA     0,SXAST ;SAVE AC0
16 06727 123400         AND     1,0
17 06730 122415         SUB#    1,0,SNR
18 06731 010403         ISZ     RXAST   ;OK
19 06732 020403         LDA     0,SXAST ;RESTORE STATUS
20 06733 002401         JMP     @RXAST
21 06734 000000 RXAST:  0
22 06735 000000 SXAST:  0
23
24                          ;GET ARGUMENT STATUS
25                          ;AND CHECK THAT NONE OF
26                          ;THE BITS ARE PRESENT IN ACTUAL STATUS
27                          ;RETURN +1 IF BITS PRESENT, OTHERWISE +2
28                          ;EXIT WITH AC1=NON-EXPECTED, AC0=ACTUAL STATUS
29                          ;
30                          ;CALL   STATN
31                          ;       ARG
32
33
34 06736 054407 XSTAN:  STA     3,RXNST
35 06737 010406         ISZ     RXNST
36 06740 025400         LDA     1,0,3   ;AC1=NONEXPECTED STATUS
37 06741 060461 SDEV2:  DIA     0,DEV   ;AC0=ACTUAL STATUS
38 06742 107415         AND#    0,1,SNR
39 06743 010402         ISZ     RXNST   ;OK
40 06744 002401         JMP     @RXNST
41 06745 000000 RXNST:  0
42
43                          ;COMBINE ARGUMENT STATUS WITH (SWITCH STATUS
44                          ;MASK) AND CHECK WHOLE AGAINST ACTUAL.
45                          ;RETURN+2 IF STATUS MATCH, +1 OTHERWISE
46                          ;EXIT WITH AC1=EXPECTED STATUS, AC0=ACTUAL
47                          ;
48                          ;CALL   STATW
49                          ;       ARG
50                          ;
51 06746 054413 XSTAW:  STA     3,RXWST
52 06747 010412         ISZ     RXWST
53 06750 025400         LDA     1,0,3
54 06751 006073         CRESW           ;READS 2 ROUTINE
55 06752 034410         LDA     3,SMASK ;MASK FOR SWITCH REGISTER
56 06753 157400         AND     2,3
57 06754 167000         ADD     3,1     ;AC1=EXP STATUS
58 06755 060461 SDEV3:  DIA     0,DEV   ;AC0=ACTUAL STATUS
59 06756 106415         SUB#    0,1,SNR
60 06757 010402         ISZ     RXWST   ;OK
61 06760 002401         JMP     @RXWST
62 06761 000000 RXWST:  0
63 06762 000000 SMASK:  000000          ;CHANGE MASK TO ONES FOR
64                                      ;THOSE BITS TO USE FROM SW.
```

```
I 0093  .MAIN
01
02                          ;GET ARGUMENT STATUS AND ARGUMENT MASK
03                          ;AND CHECK FOR EQUALITY AGAINST ACTUAL,
04                          ;EXCEPT THOSE BIT, WHERE MASK BIT ARE NULL.
05                          ;RETURN +3 IF STATUS MATCH, OTHERWISE +2.
06                          ;EXIT WITH AC0 = ACTUAL, AC1 = EXPECTED, AC2 = MASK.
07                          ;
08                          ;CALL   STATP
09                          ;       ARG
10                          ;       MASK
11
12 06763 054414 XSTAP:      STA     3,RXPST
13 06764 010413             ISZ     RXPST
14 06765 010412             ISZ     RXPST
15 06766 025400             LDA     1,0,3   ;AC1:=EXPECTED PART
16 06767 031401             LDA     2,1,3   ;AC2:=MASK
17 06770 147400             AND     2,1
18 06771 060461 SDEV4:      DIA     0,DEV   ;AC0:=ACTUAL STATUS
19 06772 115000             MOV     0,3     ;AC3:=ACTUAL STATUS FOR MASK
20 06773 157400             AND     2,3
21 06774 136415             SUB#    1,3,SNR
22 06775 010402             ISZ     RXPST   ;OK
23 06776 002401             JMP     @RXPST
24 06777 000000 RXPST:      0
25
26                          ;GET ARGUMENT STATUS
27                          ;AND CHECK THAT SOME OF
28                          ;THE BITS ARE PRESENT IN THE ACTUAL STATUS
29                          ;RETURN+2 IF BITS ARE PRESENT, OTHERWISE +1
30                          ;EXIT WITH AC1=EXPECTED, AC0=ACTUAL STATUS
31                          ;
32                          ;CALL   STATS
33                          ;       ARG
34
35 07000 054407 XSTAS:      STA     3,RXSST
36 07001 010406             ISZ     RXSST
37 07002 025400             LDA     1,0,3   ;AC1:=EXPECTED STATUS
38 07003 060461 SDEV5:      DIA     0,DEV   ;AC0:=ACTUAL STATUS
39 07004 107414             AND#    0,1,SZR
40 07005 010402             ISZ     RXSST   ;OK
41 07006 002401             JMP     @RXSST
42 07007 000000 RXSST:      0
43
44                          ;ROUTINE LOOP REPORT
45
46 07010 054454 XLORE:      STA     3,RPASS
47 07011 006046             CDICL
48 07012 006043             CCRLF
49 07013 024451             LDA     1,RPASS
50 07014 152520             SUBZL   2,2     ;AC2:=1
51 07015 146400             SUB     2,1     ;SUBTRACT 1 FROM JSR ADDR
52 07016 006056             CDOCT
53 07017 006052             CTOCT           ;PRINT ADDR
54 07020 006044             CDISP
55 07021 007071             MLORE
56 07022 006040             CMESS
57 07023 007071             MLORE           ;XXXXXX LOOP-ADDR
58 07024 006047             CDATT
59 07025 006437             JSR     @RPASS  ;RETURN TO START LOOP
```

```
| 0094  .MAIN
01
02                          ;ROUTINE TO HANDLE PASS #
03                          ;CORRECT PASSC TO YOUR CHOICE OF # OF RUNS BETWEEN
04                          ;EACH PASS MESSAGE (2 MINUTES INTERVAL IS CONVIENIENT),
05                          ;CORRECT TEXT MPASS, DPASS ACCORDINGLY,
06                          ;INITIALIZE WHEN PROGRAM STARTED/RESTARTED:
07                          ;       PASSN TO 0
08                          ;       PASSB TO # OF RUNS BETWEEN MESS = PASSC
09                          ;CALL:  CPASS
10                          ;       RETURN
11
12 07026 054436 XPASS:  STA     3,RPASS
13 07027 014432         DSZ     PASSB   ;RUN COUNT DOWN
14 07030 002434         JMP     @RPASS  ;NO MESSAGE
15 07031 023431         LDA     0,PASSC
16 07032 040427         STA     0,PASSB ;INITIALIZE RUN COUNT
17 07033 006046         CDICL           ;WRITE PASS MESSAGE
18 07034 006043         CCRLF
19 07035 006040         CMESS
20 07036 005207         MX2SP
21 07037 010424         ISZ     PASSN   ;COUNT PASS #
22 07040 024423         LDA     1,PASSN
23 07041 006053         CTDEC
24 07042 006057         CDDEC
25 07043 006044         CDISP
26 07044 007055         DPASS
27 07045 006040         CMESS
28 07046 007051         MPASS
29 07047 006047         CDATT
30 07050 002414         JMP     @RPASS  ;RETURN
31
32                  MPASS:  .TXT    !. PASS!
   07051 023056
   07052 040520
   07053 051523
   07054 000000
33
34                  DPASS:  .TXT    !. PASS!
   07055 023056
   07056 040520
   07057 051523
   07060 000000
35
36
37                  ;PASS-ADMINISTRATOR CONSTANTS AND COUNTERS (HAS TO BE SET BE-
38                  ;FORE START OF TESTEXECUTION).
39
40 07061 000000 PASSB:  0               ;COUNTER FOR # OF RUNS BETWEEN PASS MESS
41 07062 000012 PASSC:  10.             ;# OF RUNS BETWEEN PASS MESS
42 07063 000000 PASSN:  0               ;# OF PASS COUNTER
43
44 07064 000000 RPASS:  0               ;RETURN ADDR
45 07065 007137 TROTA:  TROTB           ;POINTER TO TABLE FOR BREAKPOINT
46 07066 063077 TROHC:  HALT            ;TROUBLE HALT COMMAND
47 07067 006122 TROLC:  CLORE           ;TROUBLE LOOP REPORT COMMAND
48 07070 000401 TRORC:  JMP     .+1     ;TROUBLE RESET COMMAND
49
50                  MLORE:  .TXT ! LOOP-ADDR!       ;" LOOP-ADDR"
   07071 046040
   07072 047517
   07073 026520
   07074 042101
   07075 051104
   07076 000000
```

```
I 0095  .MAIN
01
02                ;ROUTINES TO HELP LOCATING THE LOOP WHICH MAY DESTROY THE
03                ;RUNNING PROGRAM.
04                ;ALL LOOPS SHOULD START THIS WAY:
05                ;A#:    JMP     .+1         ;FOR PROGRAM TROUBLESHOOTING.
06                ;       MOV     0,1         ;INITIALIZE LOOP CODE
07                ;       SETP1               ;SETUP 1,10,100 LOOPS
08                ;THIS GIVES 3 POSSIBILITIES FOR CHANGING THE JMP .+1 TO MORE
09                ;SUITABLE CONTENT IN CASE OF TROUBLES:
10                ;A:  MANUAL INSERT JMP @RTEST FROM LAST LOOP AXX, THE PROGRAM
11                ;    THEN RUNS THE FIRST LOOPS UNTILL THIS COMMAND IS SEEN,
12                ;    FOLLOWED BY PASS MESSAGE AFTER N RUNS, THE BREAKPOINT
13                ;    IS UP TO YOU.
14                ;B:  START PROGRAM IN SA 2214 WHICH CHANGES THE JMP .+1 TO HALT
15                ;    BUT ONLY IN THE BREAKPOINTS CHOOSEN BY THE PROGRAMMER
16                ;    IN THE FOLLOWING TABLE. THE PROGRAM THEN RUNS FROM
17                ;    BREAKPOINT TO BREAKPOINT IF CONTINUE IS DEPRESSED.
18                ;C:  START PROGRAM IN SA 2216 WHICH CHANGES THE JMP .+1 TO CLORE,
19                ;    A ROUTINE, CALL LOOP REPORT, WHICH PRINTS THE PC EACH TIME
20                ;    THE BREAKPOINTS CHOOSEN BY THE PROGRAMMER IN
21                ;    FOLLOWING TABLE IS PASSED.
22                ;D:  STARTING PROGRAM IN SA 2220 WILL RESET ALL BREAKPOINTS
23                ;    MENTIONED IN THE TABLE TO JMP .+1.
24
25                          ;TROUBLE HALT
26 07077 165000 TROHA:  MOV     3,1
27 07100 006072         CSAMS               ;START ADDR MESSAGE
28 07101 036764         LDA     3,@TROTA
29 07102 054023         STA     3,IDX3  ;INIT TABLEPOINTER
30 07103 024763         LDA     1,TROHC ;HALT COMMAND
31 07104 032023 TROHL:  LDA     2,@IDX3
32 07105 151015         MOV#    2,2,SNR ;END OF TABLE ?
33 07106 000427         JMP     TROEN   ;YES, RETURN
34 07107 045000         STA     1,0,2   ;STORE HALT IN LOOP START
35 07110 000774         JMP     TROHL   ;AGAIN
36
37                          ;TROUBLE LOOP REPORT
38 07111 165000 TROLO:  MOV     3,1
39 07112 006072         CSAMS               ;START ADDR MESSAGE
40 07113 036752         LDA     3,@TROTA
41 07114 054023         STA     3,IDX3  ;INIT TABLEPOINTER
42 07115 024752         LDA     1,TROLC ;LOOP REPORT COMMAND
43 07116 032023 TROLL:  LDA     2,@IDX3
44 07117 151015         MOV#    2,2,SNR ;END OF TABLE ?
45 07120 000415         JMP     TROEN   ;YES, RETURN
46 07121 045000         STA     1,0,2   ;STORE CLORE IN LOOP START
47 07122 000774         JMP     TROLL   ;AGAIN
48
49                          ;TROUBLE RESET
50 07123 165000 TRORE:  MOV     3,1
51 07124 006072         CSAMS               ;START ADDR MESSAGE
52 07125 036740         LDA     3,@TROTA
53 07126 054023         STA     3,IDX3  ;INIT TABLEPOINTER
54 07127 024741         LDA     1,TRORC ;JMP .+1 COMMAND
55 07130 032023 TRORL:  LDA     2,@IDX3
56 07131 151015         MOV#    2,2,SNR ;END OF TABLE ?
57 07132 000403         JMP     TROEN   ;YES, RETURN
58 07133 045000         STA     1,0,2   ;STORE JMP .+1 IN LOOP START
59 07134 000774         JMP     TRORL   ;AGAIN
60 07135 002401 TROEN:  JMP     @.+1
61 07136 004740         SWISA               ;RESTART MAIN PROGRAM
```

```
I 0096  .MAIN
01
02                          .EOT
```

```
  0097   ,MAIN
01
02 07137 007137 TROTB:  .                      ;TABLE WITH ADDRESSES OF ALL TESTLOOPS
03 07140 011777          A0000
04 07141 012004          A0010
05 07142 012011          A0020
06 07143 012020          A0030
07 07144 012027          A0040
08 07145 012036          A0050
09 07146 012045          A0060
10 07147 012053          A0070
11 07150 012061          A0080
12 07151 012067          A0090
13 07152 012075          A0100
14 07153 012103          A0110
15 07154 012112          A0120
16 07155 012121          A0130
17 07156 012132          A0140
18 07157 012144          A0150
19 07160 012171          A0160
20 07161 012201          A0170
21 07162 012241          A0180
22 07163 012256          A0190
23 07164 012321          A0200
24 07165 012327          A0210
25 07166 012340          A0220
26 07167 012367          A0230
27 07170 012377          A0240
28 07171 012407          A0250
29 07172 012417          A0260
30 07173 012453          A0270
31 07174 012476          A0280
32 07175 012506          A0290
33 07176 012514          B0290
34 07177 012526          A0300
35 07200 012557          A0310
36 07201 012605          A0320
37 07202 012743          A0330
38 07203 013025          A0340
39 07204 013140          A0350
40 07205 013154          A0360
41 07206 013226          A0370
42 07207 013355          A0380
43 07210 013365          A0390
44 07211 013433          A0400
45 07212 013443          A0410
46 07213 013525          A0420
47 07214 013535          A0430
48 07215 013543          A0440
49 07216 013617          A0450
50 07217 013626          A0460
51 07220 013713          A0470
52 07221 013722          A0480
53 07222 014063          A0490
54 07223 014134          A0500
55 07224 014237          A0510
56 07225 000000          0
```

```
| 0098  .MAIN
01
02                PROG:    .TXTE   !RC3600 FDC-TEST!        ;NAME OF PROGRAM
   07226 141722
   07227 033063
   07230 030060
   07231 143240
   07232 141504
   07233 152055
   07234 051705
   07235 000324
03
04        000200 WRBUF:    .BLK   128.       ;WRITE BUFFER CONTENTS 128 CHARS
05                                           ;TO BE WRITTEN BY CALL OF "WRSEC"
06
07 07436 000000 TSBEG:     0                 ;START OF PROGRAM
```

```
| 0099  .MAIN
01
02                ;PROCEDURE CALL: SETDV
03                ;
04                ;THE PROCEDURE WILL CHANGE THE DEVICECODE IN ALL I/O-INSTRUC-
05                ;TIONS TO THE DEVICE SPECIFIED IN "DEVICE" (PAGE ZERO). THE
06                ;CHANGE WILL TAKE PLACE IN LOCATIONS FROM LOC. SPECIFIED IN AC1
07                ;TO LOC. SPECIFIED IN AC2. THE NEW DEVICE-CODE WILL BE SET =
08                ;B10 - B15  OF AC0.
09                ;
10                ;ENTRY:   AC0= NEW DEVICE CODE,AC1= FIRST LOCATION,AC2= LAST
11                ;         LOCATION  (LOC. "DEVICE" = CURRENT DEVICE NO., IT
12                ;         HAS TO BE SET EQUAL TO THE ASSEMBLED DEVICE NO.
13                ;         IN THE ASCII-SOURCE PROGRAM.)
14                ;RETURN: .+1
15                ;
16                ;DESTROYED: AC3
17
18 07437 054157 CHDEV:  STA     3,FPRET
19 07440 044155         STA     1,FSAV1
20 07441 024220         LDA     1,F0077 ;AC1:= 77
21 07442 107400         AND     0,1
22 07443 034124         LDA     3,DEVICE;LAST USED CODE FOR DEVICE
23 07444 166415         SUB#    3,1,SNR ;SKIP IF LAST USED CODE <> NEW CODE
24 07445 000435         JMP     CIMOU   ;CHANGE IS NOT NECESSARY - EXIT
25
26 07446 050156         STA     2,FSAV2 ;CHANGE ALL I/O-INSTR. TO "DEVICE"
27 07447 040154         STA     0,FSAV0
28 07450 024155         LDA     1,FSAV1
29 07451 044433         STA     1,INSPOI;FIRST ADDRESS TO EXAMINE
30 07452 132420         SUBZ    1,2
31 07453 151400         INC     2,2
32 07454 050431         STA     2,INSCOU;# OF INSTRUCTIONS TO EXAMINE
33 07455 030220         LDA     2,F0077 ;AC2:= 77
34 07456 143400         AND     2,0
35 07457 040124         STA     0,DEVICE;UPDATE DEVICE
36 07460 024231         LDA     1,F060K ;AC1:= 060000
37 07461 123000         ADD     1,0     ;AC0:= NEW CODE WITH BIT 1,2
38 07462 167000         ADD     3,1     ;AC1:= LAST USED CODE WITH BIT 1,2
39
40 07463 010421 NXINS:  ISZ     INSPOI  ;INCREMENT INSTRUCTION POINTER
41 07464 032420         LDA     2,@INSPOI;GET INSTRUCTION
42 07465 034421         LDA     3,IOMSK ;AC3:= 160077
43 07466 014417         DSZ     INSCOU
44 07467 157401         AND     2,3,SKP ;AC3:= CURR. INSTR. WITH OUTMASKED BIT
45                                      ;3,4,5,6,7,8,9
46 07470 000410         JMP     SDOUT   ;ALL INSTRUCTIONS  EXAMINED  - EXIT
47 07471 166414         SUB#    3,1,SZR ;SKIP IF INSTR. IS I/O-INSTR TO LAST
48                                      ;USED DEVICE
49 07472 000771         JMP     NXINS   ;NOT A I/O-INSTR. GET NEXT
50 07473 034414         LDA     3,FUMSK ;AC3:= 017700
51 07474 173400         AND     3,2     ;AC2:= CURRENT FUNCTIONBITS
52 07475 113000         ADD     0,2     ;AC2:= NEW I/O-INSTR
53 07476 052406         STA     2,@INSPOI;STORE INSTR.
54 07477 000764         JMP     NXINS   ;GET NEXT INSTR.
55
56 07500 020154 SDOUT:  LDA     0,FSAV0
57 07501 030156         LDA     2,FSAV2
58 07502 024155 CIMOU:  LDA     1,FSAV1
59 07503 002157         JMP     @FPRET
60
61 07504 000000 INSPOI: 0               ;POINTER TO CURRENT INSTR.
62 07505 000000 INSCOU: 0               ;INSTRUCTION COUNTER
63 07506 160077 IOMSK:  160077
64 07507 017700 FUMSK:  017700
```

```
| 0100  .MAIN
01
02                ;PROCEDURE CALL: SETIN
03                ;
04                ;THE ROUTINE WILL TAKE THE CONTENT OF AC1 AND USE IT AS MASK
05                ;IN A MASKOUT-INSTRUCTION.IT WILL FURTHERMORE PLACE THE ADDRESS
06                ;OF THE INTERRUPT-SERVICE ROUTINE IN LOC. 1, AND SET THE "IN-
07                ;TERRUPT HANDLE FLAG" TO -1 (IMMEDIATLY RETURN) IF AC0 = -1,
08                ;ELSE TO ZERO.
09                ;
10                ;ENTRY:  AC0 = INTERRUPT HANDLE FLAG, AC1 = INTERRUPT MASK
11                ;
12                ;RETURN: .+1
13                ;
14                ;DESTROYED: AC3
15
16
17
18 07510 054157 STINT:  STA     3,FPRET
19 07511 040154         STA     0,FSAV0
20 07512 044155         STA     1,FSAV1
21 07513 050156         STA     2,FSAV2
22
23 07514 006063         TIMSK           ;WAIT FOR BUSY = 0 (MAX 2 SEC)
24 07515 003720         2000.
25 07516 063561         SKPBZ   XFDD
26 07517 000402         JMP     .+2     ;ERROR
27 07520 000403         JMP     STIN1   ;OK
28 07521 030157         LDA     2,FPRET
29 07522 006114         EHALT           ;BUSY DOESN'T RETURN TO ZERO
30                                      ;AC2= CALL OF "SETIN"
31
32 07523 006116 STIN1:  STATN           ;CHECK STATUS
33 07524 135775         135775
34 07525 000402         JMP     .+2     ;ERROR
35 07526 000403         JMP     STIN2   ;OK
36 07527 030157         LDA     2,FPRET
37 07530 006114         EHALT           ;FDD-STATUS ISN'T OK
38                                      ;AC0=RECEIVED STATUS,AC1="NON"-EXPECTED
39                                      ;STATUS, AC2= CALL OF "SETIN".
40 07531 024155 STIN2:  LDA     1,FSAV1
41 07532 066077         MSKO    1       ;MASK OUT
42
43 07533 024414         LDA     1,INTAD
44 07534 044001         STA     1,1     ;SET INTERRUPT SERVICE ROUT. ADDRESS
45 07535 102440         SUBC    0,0     ;AC0:= 0
46 07536 040237         STA     0,INTOC ;CLEAR INTERRUPT OCCOURED FLAG
47
48 07537 020154         LDA     0,FSAV0
49 07540 126000         ADC     1,1     ;AC1:= -1
50 07541 106414         SUB#    0,1,SZR ;SKIP IF AC0 = -1
51 07542 126440         SUBC    1,1     ;AC1:= 0
52 07543 044451         STA     1,INFLG ;SET INTERRUPT HANDLE FLAG
53
54 07544 024155         LDA     1,FSAV1
55 07545 030156         LDA     2,FSAV2
56
57 07546 002157         JMP     @FPRET
58
59 07547 007550 INTAD:  INTSV           ;ADDRESS OF INTERRUPT SERV.ROUT.
```

```
| 0101   .MAIN
01
02                ;INTERRUPT - SERVICEROUTINE
03                ;
04                ;ANY INTERRUPT WILL CAUSE A JUMP TO THIS ROUTINE, IF PROCEDURE
05                ;SETIN WAS CALLED PREVICUSLY. IF INTERRUPT-HANDLE-FLAG (SET
06                ;BY SETIN) = -1 AT INTERRUPTIME, THEN AN IMMEDIATE JUMP BACK
07                ;TO THE INTERRUPTED ROUTINE WILL BE EXECUTED. IF NOT, THE IN-
08                ;TERRUPT-SOURCE WILL BE RECOGNIZED BY INTA.
09                ;IF IT'S FDD, THEN A JUMP TO INTERRUPTED INSTRUCTION  WILL
10                ;BE EXECUTED, IF NOT FDD, AN ERRORHALT WILL BE THE RESULT.
11                ;
12                ;ENTRY AND RETURN:
13                ;
14                ;  IF INTERRUPT-HANDLE-FLAG = -1: RETURN IMMEDIATLY TO .+ 0
15                ;
16                ;  IF INTERRUPT-HANDLE-FLAG <> -1: CHECK INTERRUPTDEVICE AND
17                ;                                  RETURN TO .+ 0
18
19 07550 040437 INTSV:  STA     0,INSAV
20 07551 044437         STA     1,INSAV+1
21 07552 050437         STA     2,INSAV+2
22 07553 054437         STA     3,INSAV+3
23 07554 101200         MOVR    0,0
24 07555 040436         STA     0,INSAV+4
25
26 07556 063777         SKPDZ   CPU     ;SKIP IF NOT POWER INTR.
27 07557 000436         JMP     POWIN
28 07560 020434         LDA     0,INFLG ;LOAD INTERRUPT-HANDLE-FLAG
29 07561 126000         ADC     1,1     ;AC1:= -1
30 07562 106415         SUB#    0,1,SNR ;SKIP IF AC0<>-1
31 07563 000412         JMP     INTOU   ;RETURN IMMEDIATLY
32 07564 024124         LDA     1,DEVICE;AC1:= CURRENT FDD DEVICE-CODE
33 07565 061477         INTA    0       ;GET INTERRUPT DEVICE TO AC0
34 07566 106415         SUB#    0,1,SNR ;SKIP IF <> FDD
35 07567 000403         JMP     INTS1
36 07570 006114         EHALT           ;INTERRUPT FROM UNKNOWN DEVICE
37                                      ;DEVICE-CODE IN AC0
38 07571 126441         SUBO    1,1,SKP ;AC1:= 0
39 07572 126000 INTS1:  ADC     1,1     ;AC1:= -1
40 07573 034417         LDA     3,INSAV+3
41 07574 030415         LDA     2,INSAV+2
42 07575 044237 INTOU:  STA     1,INTOC ;SET INTERRUPT OCCOURED FLAG = -1 IF
43 07576 024412         LDA     1,INSAV+1;INTERRUPT OCCOURED ELSE 0
44 07577 020414         LDA     0,INSAV+4
45 07600 101100         MOVL    0,0
46 07601 020000         LDA     0,0     ;GET RETURN ADDRESS AND STORE IT
47 07602 040005         STA     0,5     ;IN LOC. 5.
48 07603 020076         LDA     0,POWZE ;RESTORE POWER-UP INSTR.
49 07604 040000         STA     0,0
50 07605 020402         LDA     0,INSAV
51 07606 002005         JMP     @5      ;RETURN TO INTERRUPTET PROGRAM
52
53 07607 000000 INSAV:  0               ;AC - SAVE LOCATIONS
54 07610 000000         0
55 07611 000000         0
56 07612 000000         0
57 07613 000000         0
58 07614 177777 INFLG:  -1              ;INTERRUPT HANDLE FLAG
59
60 07615 024076 POWIN:  LDA     1,POWZE ;POWER INTR. OCCOURED
61 07616 044000         STA     1,0     ;SET POWER-UP INSTR. IN LOC 0.
62 07617 063077         HALT            ;WAIT FOR POWER RETURN, I.E. EXE-
63                                      ;CUTION OF INSTR. IN LOC 0.
```

```
| 0102  .MAIN
01
02                ;PROCEDURE CALL: RECAL
03                ;
04                ;THE ROUTINE WILL RECALIBRATE THE DISC-DRIVE, AND WAIT UN-
05                ;TIL DONE IS SET (MAX 2 SEC)
06                ;
07                ;RETURN: .+1
08                ;
09                ;DESTROYED: AC3
10
11
12 07620 054157 RCALI:  STA     3,FPRET
13 07621 040154         STA     0,FSAV0
14 07622 044155         STA     1,FSAV1
15 07623 050156         STA     2,FSAV2
16
17 07624 006063         TIMSK           ;WAIT FOR BUSY = 0 (MAX 2 SEC)
18 07625 003720         2000.
19 07626 063561         SKPBZ   XFDD
20 07627 000402         JMP     .+2     ;ERROR
21 07630 000403         JMP     RCAL1   ;OK
22 07631 030157         LDA     2,FPRET
23 07632 006114         EHALT           ;BUSY DOESN'T RETURN TO ZERO
24                                      ;AC2= CALL OF "RECAL"
25
26 07633 006116 RCAL1:  STATN           ;CHECK STATUS
27 07634 040000         040000
28 07635 000402         JMP     .+2     ;ERROR
29 07636 000403         JMP     RCAL2   ;OK
30 07637 030157         LDA     2,FPRET
31 07640 006114         EHALT           ;DEVICE IS OFF-LINE
32                                      ;AC2= CALL OF "RECAL"
33
34 07641 020224 RCAL2:  LDA     0,FO1K0 ;AC0:= 1000 = RECAL-COMMAND
35 07642 061161         DOAS    0,XFDD
36
37 07643 006063         TIMSK           ;WAIT FOR DONE = 1 (MAX 2 SEC)
38 07644 003720         2000.
39 07645 063661         SKPDN   XFDD
40 07646 000402         JMP     .+2     ;ERROR
41 07647 000403         JMP     RCAL3   ;OK
42 07650 030157         LDA     2,FPRET
43 07651 006114         EHALT           ;DONE ISN'T SET AFTER A RECALIBRATE-
44                                      ;COMMAND, AC2= CALL OF "RECAL".
45
46 07652 006116 RCAL3:  STATN           ;CHECK STATUS
47 07653 175777         175777
48 07654 000402         JMP     .+2     ;ERROR
49 07655 000403         JMP     RCAL4   ;OK
50 07656 030157         LDA     2,FPRET
51 07657 006114         EHALT           ;STATUS ISN'T CORRECT AFTER A RECALI-
52                                      ;BRATION, AC0=RECEIVED STATUS, AC1=
53                                      ;"NON"-EXPECTED STATUS, AC2= CALL OF
54                                      ;"RECAL".
55
56 07660 020154 RCAL4:  LDA     0,FSAV0
57 07661 024155         LDA     1,FSAV1
58 07662 030156         LDA     2,FSAV2
59 07663 002157         JMP     @FPRET
```

```
1 0103   .MAIN
01
02                ;PROCEDURE CALL: TRPOS
03                ;
04                ;THE PROCEDURE WILL POSITION THE HEAD TO THE TRACK NO. SPE-
05                ;CIFIED IN AC1 AND WAIT UNTIL DONE IS SET (MAX 2 SEC).
06                ;
07                ;ENTRY:  AC1= TRACK NO.
08                ;
09                ;RETURN:  .+1
10                ;
11                ;DESTROYED: AC3
12
13
14 07664 054157 TRAP0:  STA     3,FPRET
15 07665 040154         STA     0,FSAV0
16 07666 044155         STA     1,FSAV1
17 07667 050156         STA     2,FSAV2
18
19 07670 006063         TIMSK                   ;WAIT FOR BUSY = 0 (MAX 2 SEC)
20 07671 003720         2000.
21 07672 063561         SKPBZ   XFDD
22 07673 000402         JMP     .+2             ;ERROR
23 07674 000403         JMP     TRAP1           ;OK
24 07675 030157         LDA     2,FPRET
25 07676 006114         EHALT                   ;BUSY DOESN'T RETURN TO ZERO
26                                              ;AC2= CALL OF "TRPOS"
27
28 07677 006116 TRAP1:  STATN                   ;CHECK STATUS
29 07700 040000         040000
30 07701 000402         JMP     .+2             ;ERROR
31 07702 000403         JMP     TRAP2           ;OK
32 07703 030157         LDA     2,FPRET
33 07704 006114         EHALT                   ;DEVICE IS OFF-LINE
34                                              ;AC2= CALL OF "TRPOS".
35
36 07705 030155 TRAP2:  LDA     2,FSAV1 ;WANTED TRACK NO.
37 07706 024225         LDA     1,F01K4 ;AC1:= 1400 = TRACK SEARCH COMMAND
38 07707 133000         ADD     1,2
39 07710 071161         DOAS    2,XFDD
40 07711 006063         TIMSK                   ;WAIT FOR DONE = 1 (MAX 2 SEC)
41 07712 003720         2000.
42 07713 063661         SKPDN   XFDD
43 07714 000402         JMP     .+2             ;ERROR
44 07715 000403         JMP     TRAP3           ;OK
45 07716 030157         LDA     2,FPRET
46 07717 006114         EHALT                   ;DONE ISN'T SET AFTER A TRACK SEARCH
47                                              ;COMMAND, AC2= CALL OF "TRPOS"
48
49 07720 006116 TRAP3:  STATN                   ;CHECK STATUS
50 07721 175777         175777
51 07722 000402         JMP     .+2             ;ERROR
52 07723 000403         JMP     TRAP4           ;OK
53 07724 030157         LDA     2,FPRET
54 07725 006114         EHALT                   ;STATUS ISN'T CORRECT AFTER A TRACK
55                                              ;SEARCH COMMAND. AC0=RECEIVED STATUS,
56                                              ;AC1= "NON"-EXPECTED STATUS, AC2=
57                                              ;CALL OF "TRPOS".
58
59 07726 062461 TRAP4:  DIC     0,XFDD  ;GET TRACKCOUNTER
60 07727 101300         MOVS    0,0
61 07730 024222         LDA     1,F0377 ;AC1:= 377
62 07731 123400         AND     1,0
```

```
| 0104   .MAIN
01
02 07732 024155          LDA     1,FSAV1 ;AC1:= WANTED TRACK NO.
03 07733 030157          LDA     2,FPRET
04 07734 106414          SUB#    0,1,SZR
05 07735 006114          EHALT           ;TRACKCOUNTER <> WANTED TRACK NO.
06                                       ;AFTER POSITIONING.AC0= TRACK COUN-
07                                       ;TER,AC1= WANTED TRACK NO.,AC2=
08                                       ;CALL OF "TRPOS".
09 07736 020154 TRAPS:   LDA     0,FSAV0
10 07737 024155          LDA     1,FSAV1
11 07740 030156          LDA     2,FSAV2
12 07741 002157          JMP     @FPRET
13
14
15
16
17               ;PROCEDURE CALL:  TRMES
18               ;
19               ;THE ROUTINE WILL MESS THE STATE OF THE TOTAL RE-READ COUNTER
20               ;RELATIVE TO THE NO. OF SECTOR READS. I.E., IF 200 SECTORS ARE
21               ;READ, AND 448 READ-TRIES AT ALL WERE NECESSARY, THE NUMBER
22               ;MESSED WILL BE 224.
23               ;
24               ;ENTRY:     AC0= NO. OF SECTORS READ
25               ;           "TOTCO"= TOTAL NO. OF READ-TRIES.
26               ;
27               ;RETURN:    .+1 IF RE-READ PERCENT < 20
28               ;           .+2 IF RE-READ PERCENT >= 20
29               ;
30               ;DESTROYED:   AC3
31
32 07742 054157 MESTR:   STA     3,FPRET
33 07743 040154          STA     0,FSAV0
34 07744 044155          STA     1,FSAV1
35 07745 050156          STA     2,FSAV2
36
37 07746 024242          LDA     1,TOTCO ;AC1:= TOTAL NO. OF READTRIES
38 07747 030210          LDA     2,FD100 ;AC2:= 100.
39 07750 006066          MULTI
40                                       ;AC0,AC1:= AC1 * AC2
41 07751 030154          LDA     2,FSAV0 ;AC2:= NO. OF SECTORS READ
42 07752 006070          DIVID
43                                       ;AC1:= AC0,AC1/AC2
44 07753 044253          STA     1,WORK1
45 07754 030201          LDA     2,FD020 ;AC2:= 20.
46 07755 132032          ADCZ#   1,2,SZC ;SKIP IF AC1 >= AC2
47 07756 000414          JMP     MESTO   ;RETURN +1
48 07757 010157          ISZ     FPRET   ;MESSAGE AND RETURN +2
49
50 07760 006043          CCRLF
51 07761 006046          CDICL
52 07762 006040          CMESS
53 07763 011632          RRPER           ;"RE-READ %="
54 07764 006044          CDISP
55 07765 011632          RRPER
56 07766 024253          LDA     1,WORK1
57 07767 006053          CTDEC           ;TYPE RE-READ PERCENT
58 07770 006057          CDDEC
59 07771 006047          CDATT
60
61 07772 020154 MESTO:   LDA     0,FSAV0
62 07773 024155          LDA     1,FSAV1
63 07774 030156          LDA     2,FSAV2
64 07775 002157          JMP     @FPRET
```

```
| 0105  .MAIN
01
02                ;PROCEDURE CALL: GRAND
03                ;
04                ;THE ROUTINE WILL GENERATE A RANDOM NUMBER, USING THE VALUE
05                ;IN AC0 AS "OLD" NUMBER. THE RANDOM NO. IS DELIVERED IN AC0.
06                ;
07                ;ENTRY:  AC0= LAST USED RANDOM NO.
08                ;
09                ;RETURN: .+1
10                ;
11                ;DESTROYED: AC0,AC3
12
13 07776 054157 GERAN:  STA     3,FPRET
14 07777 044155         STA     1,FSAV1
15 10000 050156         STA     2,FSAV2
16
17 10001 024422         LDA     1,RANCA
18 10002 044423         STA     1,RANCO ;SET COUNTER
19 10003 105120         MOVZL   0,1
20 10004 125120         MOVZL   1,1
21
22 10005 014420         DSZ     RANCO   ;DECREMENT COUNTER
23 10006 000776         JMP     .-2
24
25 10007 107000         ADD     0,1     ;8 TIMES SHIFTED
26 10010 125120         MOVZL   1,1
27 10011 125120         MOVZL   1,1
28 10012 123000         ADD     1,0
29 10013 034411         LDA     3,RANCB
30 10014 163000         ADD     3,0
31
32 10015 101112         MOVL#   0,0,SZC
33 10016 174220         COMZR   3,3
34 10017 054405         STA     3,RANCB
35
36 10020 024155         LDA     1,FSAV1
37 10021 030156         LDA     2,FSAV2
38 10022 002157         JMP     @FPRET
39
40 10023 000010 RANCA:  10
41 10024 033031 RANCB:  33031
42 10025 000000 RANCO:  0
43
44
45
46                ;PROCEDURE CALL: RANTS
47                ;
48                ;THE ROUTINE WILL, BY USE OF PROCEDURE "GRAND", GENERATE A
49                ;RANDOM TRACK NO. AND A RANDOM SECTOR NO. THE TRACK NO. WILL
50                ;BE DELIVERED IN AC1, THE SECTOR NO. IN AC0.
51                ;
52                ;RETURN: .+1
53                ;
54                ;DESTROYED: AC0,AC1,AC3
55
56 10026 054163 RTRSE:  STA     3,SPRET
57 10027 050162         STA     2,SSAV2
58
59 10030 020447         LDA     0,LARAN ;LAST USED RANDOM NO.
60 10031 006134         GRAND           ;GET NEW RANDOM NO. TO AC0
61 10032 040445         STA     0,LARAN
62 10033 040445         STA     0,RANWK
```

```
| 0106  .MAIN
01
02 10034 024445 RESEC:  LDA     1,FMSK4 ;AC1:= 37
03 10035 107400         AND     0,1     ;AC1:= #
04 10036 030235         LDA     2,MXSEC ;AC2:= 26.
05 10037 146032         ADCZ#   2,1,SZC ;SKIP IF MAX. SEC. >= #
06 10040 000405         JMP     NWSEC   ;GET A NEW NUMBER
07 10041 125005         MOV     1,1,SNR ;SECTOR 0 NOT ALLOWED
08 10042 000403         JMP     NWSEC
09 10043 044440         STA     1,RASEC ;SECTOR NO IS OK
10 10044 000405         JMP     GETTR   ;GET TRACK NO.
11
12 10045 024433 NWSEC:  LDA     1,RANWK ;TRY TO GET A NEW #
13 10046 121220         MOVZR   1,0
14 10047 040431         STA     0,RANWK
15 10050 000764         JMP     RESEC
16
17 10051 020426 GETTR:  LDA     0,LARAN ;GET A RANDOM TRACK NO.
18 10052 040426         STA     0,RANWK
19 10053 024427 RETRA:  LDA     1,FMSK5 ;AC1:= 007740
20 10054 107400         AND     0,1
21 10055 125220         MOVZR   1,1
22 10056 125220         MOVZR   1,1
23 10057 125220         MOVZR   1,1
24 10060 125220         MOVZR   1,1
25 10061 125220         MOVZR   1,1     ;AC1:= #
26 10062 030236         LDA     2,MXTRA ;AC2:= 76.
27 10063 146032         ADCZ#   2,1,SZC ;SKIP IF MAX. TRACK >= #
28 10064 000403         JMP     NWTRA   ;TRY TO GET A NEW NO.
29 10065 044417         STA     1,RATRA ;TRACK NO. IS OK
30 10066 000405         JMP     RANTO   ;RETURN
31
32 10067 024411 NWTRA:  LDA     1,RANWK ;TRY TO GET A NEW #
33 10070 121220         MOVZR   1,0
34 10071 040407         STA     0,RANWK
35 10072 000761         JMP     RETRA
36
37 10073 020410 RANTO:  LDA     0,RASEC ;AC0:= SECTOR NO.
38 10074 024410         LDA     1,RATRA ;AC1:= TRACK NO.
39 10075 030162         LDA     2,SSAV2
40 10076 002163         JMP     @SPRET
41
42 10077 012345 LARAN:  12345           ;LAST GENERATED RANDOM NO.
43 10100 000000 RANWK:  0
44 10101 000037 FMSK4:  000037
45 10102 007740 FMSK5:  007740
46 10103 000000 RASEC:  0               ;GENERATED SECTOR NO.
47 10104 000000 RATRA:  0               ;GENERATED TRACK NO.
48
49
50              ;PROCEDURE CALL: CURTR
51              ;
52              ;THE ROUTINE WILL, BY USE OF DIC-INSTR., GET THE CURRENT STATE
53              ;OF FDD-TRACKCOUNTER TO AC2.
54              ;
55              ;RETURN:  AC2= CURRENT STATE OF TRACK COUNTER
56              ;
57              ;DESTROYED: AC2,AC3
58
59 10105 054167 GCUTR:  STA     3,TPRET
60
61 10106 072461         DIC     2,XFDD  ;GET TRACKCOUNTER,DATA FIELD CHAR
62 10107 151300         MOVS    2,2     ;SWAP AND MASK OUT
63 10110 034222         LDA     3,FO377 ;AC3:= 377
64 10111 173400         AND     3,2
65
66 10112 002167         JMP     @TPRET
```

```
 I 0107  .MAIN
 01
 02
 03                ;PROCEDURE CALL: RESET
 04                ;
 05                ;THE ROUTINE WILL SEND A CLEAR-PULSE TO FDC IF IT'S NOT BUSY.
 06                ;IF IT'S BUSY THE ROUTINE WILL WAIT UP TO 2 SEC. BEFORE ER-
 07                ;ORHALT.
 08 10113 054447 FDRST:  STA     3,SXRT0
 09 10114 050454         STA     2,SXSV2
 10 10115 044452         STA     1,SXSV1
 11 10116 040450         STA     0,SXSV0
 12 10117 006063         TIMSK
 13 10120 003720         2000.                   ;WAIT FOR FDD-BUSY = 0 (MAX 2 SEC)
 14 10121 063561         SKPBZ   XFDD
 15 10122 000416         JMP     RSTER           ;ERROR
 16 10123 060261         NIOC    XFDD            ;CLEAR FDC
 17 10124 020442 RSTOU:  LDA     0,SXSV0
 18 10125 024442         LDA     1,SXSV1
 19 10126 030442         LDA     2,SXSV2
 20 10127 063711         SKPDZ   XTTO
 21 10130 060211         NIOC    XTTO
 22 10131 063717         SKPDZ   XLPT
 23 10132 060217         NIOC    XLPT
 24 10133 063710         SKPDZ   XTTI
 25 10134 060210         NIOC    XTTI
 26 10135 063714         SKPDZ   XRTC
 27 10136 060214         NIOC    XRTC
 28 10137 002423         JMP     @SXRT0          ;RETURN
 29
 30 10140 032403 RSTER:  LDA     2,@ADLOP
 31 10141 006114         EHALT
 32 10142 000762         JMP     RSTOU           ;BUSY DIDN'T RETURN TO 0 WITHIN 2 SEC.
 33                                              ;AC2= ADDRESS OF TESTLOOP WHICH CAU-
 34 10143 006542 ADLOP:  LOOPR                   ;SED THE ERROR.
 35
 36
 37                ;PROCEDURE CALL: CBELL
 38                ;
 39                ;THE ROUTINE WILL RING THE TTY-BELL AND WAIT FOR 50 MS.
 40 10144 054417 RINGB:  STA     3,SXRT1
 41 10145 040424         STA     0,SXSV3
 42 10146 044424         STA     1,SXSV4
 43 10147 050424         STA     2,SXSV5
 44 10150 020221         LDA     0,F0207
 45 10151 063511         SKPBZ   XTTO
 46 10152 000777         JMP     .-1
 47 10153 061111         DOAS    0,XTTO
 48 10154 006061         CWAIT
 49 10155 000205         FD050
 50 10156 020413         LDA     0,SXSV3
 51 10157 024413         LDA     1,SXSV4
 52 10160 030413         LDA     2,SXSV5
 53 10161 002402         JMP     @SXRT1
 54
 55
 56 10162 000000 SXRT0:  0
 57 10163 000000 SXRT1:  0
 58 10164 000000 SXRT2:  0
 59 10165 000000 SXRT3:  0
 60 10166 000000 SXSV0:  0
 61 10167 000000 SXSV1:  0
 62 10170 000000 SXSV2:  0
 63 10171 000000 SXSV3:  0
 64 10172 000000 SXSV4:  0
 65 10173 000000 SXSV5:  0
 66 10174 000000 SXSV6:  0
```

```
| 0108  .MAIN
01
02                ;PROCEDURE CALL: SWENB
03                ;
04                ;THE PROCEDURE WILL ENABLE THE PANEL-SWITCHES BY MAKING
05                ;"SWREG" = 0. IT'S CALLED ONLY FROM STANDARD ROUTINE "CSAMS".
06                ;
07                ;RETURN:     .+1
08                ;
09                ;DESTROYED:   AC3
10
11 10175 054767 ENBSW:  STA     3,SXRT2
12 10176 176440         SUBO    3,3      ;AC3:= 0
13 10177 054126         STA     3,SWREG
14 10200 062677         ICRST
15 10201 002763         JMP     @SXRT2
16
17                ;PROCEDURE CALL:  RDSWS
18                ;
19                ;THE PROCEDURE IS AN EXTENSION OF THE STANDARD PROCEDURE "XRESW",
20                ;WHICH WILL READ AND ANNOUNCE THE SWITCH-SETTING. "RDSWS" WILL
21                ;USE "SWREG" AS SWITCHCONTROL IF IT'S <> 0, ELSE THE PROC. WILL
22                ;READ THE SWITCHES AND CHECK IF ANY OF THE SWITCHES 1-9 ARE SET.
23                ;IF THAT'S THE FACT, THE SETTING OF SWITCH 13 - 15 WILL BE DIS-
24                ;ABLED TO PROHIBIT PROGRAM-HALTS,CAUSED BY THIS ERRONEOUS SWITCH-
25                ;SETTING.
26                ;
27                ;RETURN:    .+1 IF "INIMK" = 0
28                ;               IF "SWREG" <> 0 THEN AC2= "SWREG", ELSE
29                ;               IF NONE OF SW 1-9 ARE SET, THEN AC2= SWITCHES READ.
30                ;               IF ANY OF SW 1-9 ARE SET, THEN AC2= SWITCHES READ,
31                ;               WITH OUTMASKED BIT 13-15
32                ;
33                ;          TO "MSPRO" IF "INIMK" <> 0.
34                ;
35                ;DESTROYED:  AC3,AC2  (AND AC1 IF "SWREG" <> 0)
36
37 10202 054763 RDSWI:  STA     3,SXRT3
38 10203 040771         STA     0,SXSV6
39
40 10204 030126         LDA     2,SWREG ;AC2:= COMMON SWITCH REGISTER
41 10205 151014         MOV#    2,2,SZR ;SKIP IF REG = 0
42 10206 145000         MOV     2,1     ;THIS WILL FORCE "UNCHANGED"? YES.
43 10207 034125         LDA     3,INIMK ;AC3 <> 0 IF FIRST CALL AFTER LOAD
44 10210 175014         MOV#    3,3,SZR ;SKIP IF NOT FIRST CALL AFTER PROG.LOAD
45 10211 000411         JMP     INIO1
46
47 10212 020415         LDA     0,SWMS1 ;AC0:= 077700
48 10213 034415         LDA     3,SWMS2 ;AC3:= 177770
49 10214 151015         MOV#    2,2,SNR ;SKIP IF SWITCH REG. <> 0
50
51 10215 070477         READS   2
52 10216 143414         AND#    2,0,SZR
53 10217 173400         AND     3,2     ;MASK OUT SW 13-15
54 10220 020754         LDA     0,SXSV6
55 10221 002744         JMP     @SXRT3  ;RETURN TO PROC. "XRESW"
56
57 10222 176440 INIO1:  SUBO    3,3
58 10223 054125         STA     3,INIMK ;CLEAR INI-MARK
59 10224 145000         MOV     2,1
60 10225 002401         JMP     @IMSPR  ;GO TO "REBIN" AND MESS PROG. NAME
61
62 10226 001447 IMSPR:  MSPRO
63 10227 077700 SWMS1:  077700
64 10230 177770 SWMS2:  177770
```

```
| 0109  .MAIN
01
02                ;PROCEDURE CALL: CCMPT
03                ;
04                ;THE ROUTINE WILL CHANGE EVERY BYTE IN THE WRITE BUFFER TO IT'S
05                ;COMPLEMENT.
06                ;
07                ;RETURN:     ,+1
08                ;
09                ;DESTROYED:  AC3
10
11
12 10231 054163 PATCM:  STA     3,SPRET
13 10232 044161         STA     1,SSAV1
14 10233 050162         STA     2,SSAV2
15
16 10234 024433         LDA     1,PATT1 ;CHANGE PATTERNWORDS
17 10235 030433         LDA     2,PATT2
18 10236 044432         STA     1,PATT2
19 10237 050430         STA     2,PATT1
20
21 10240 000410         JMP     PATFC   ;USE PTFLL-ROUTINE TO CHANGE BUFFER.
22
23
24                ;PROCEDURE CALL: PTFLL
25                ;
26                ;THE ROUTINE WILL FILL THE WRITE BUFFER (WRBUF) WITH 128 BYTES
27                ;TAKEN ALTERNATELY FROM PATT1 AND PATT2.
28                ;
29                ;RETURN: .+1
30                ;
31                ;DESTROYED: AC3
32
33 10241 054163 PATFL:  STA     3,SPRET
34 10242 044161         STA     1,SSAV1
35 10243 050162         STA     2,SSAV2
36
37 10244 024425         LDA     1,PAT1S
38 10245 044422         STA     1,PATT1
39 10246 024424         LDA     1,PAT2S
40 10247 044421         STA     1,PATT2
41 10250 024206 PATFC:  LDA     1,FD064 ;AC1:= 64.
42 10251 044415         STA     1,CHACP ;INITIALIZE CHARCOUNTER
43 10252 030234         LDA     2,AWBUF ;AC2:= ADDRESS OF WRITE BUFFER
44
45 10253 024414 PASWB:  LDA     1,PATT1
46 10254 045000         STA     1,0,2
47 10255 151400         INC     2,2
48 10256 024412         LDA     1,PATT2
49 10257 045000         STA     1,0,2
50 10260 151400         INC     2,2
51 10261 014405         DSZ     CHACP
52 10262 000771         JMP     PASWB   ;MOVE MORE PATTERNS
53
54 10263 024161         LDA     1,SSAV1 ;WRITE BUFFER IS FULL - EXIT
55 10264 030162         LDA     2,SSAV2
56 10265 002163         JMP     @SPRET
57
58 10266 000000 CHACP:  0               ;CHARCOUNTER
59 10267 000125 PATT1:  125             ;PATTERN: 0000000001010101
60 10270 000252 PATT2:  252             ;PATTERN: 0000000010101010
61
62 10271 000125 PAT1S:  125
63 10272 000252 PAT2S:  252
```

```
 0110  .MAIN
01
02                ;PROCEDURE CALL: RDSEC
03                ;
04                ;THE ROUTINE WILL READ ONE SECTOR FROM DISCETTE INTO CON-
05                ;TROLLERS FIFO-BUFFER. IF SUCCESSFUL THE RETURN IS CALL+3,
06                ;ELSE THE RETURN IS +1 OR +2 DEPENDING ON THE ERROR. THE
07                ;TRACK-POSITIONING HAS TO BE DONE PREVIOUSLY.
08                ;
09                ;ENTRY:    AC0= SECTOR NO. TO BE READ
10                ;
11                ;RETURN:   .+1    HARD ERROR, NO DATA IS AVAILABLE IN BUFFER.
12                ;                 AC1= RECEIVED ERROR-STATUS.
13                ;          .+2    ADDRESS FIELD, POS.,TIME OUT OR PARITY ERROR.
14                ;                 AC1= RECEIVED ERROR-STATUS.
15                ;          .+3    NORMAL RETURN. DATA IS READY. AC2= DATA
16                ;                 SYNC BYTE READ.
17                ;
18                ;DESTROYED:       AC1,AC2,AC3
19
20 10273 054157 REASE:  STA      3,FPRET
21 10274 040154          STA      0,FSAV0
22 10275 044155          STA      1,FSAV1
23 10276 050156          STA      2,FSAV2
24
25 10277 006063          TIMSK             ;WAIT FOR BUSY = 0 (MAX 2 SEC)
26 10300 003720          2000.
27 10301 063561          SKPBZ    XFDD
28 10302 000402          JMP      .+2      ;ERROR
29 10303 000403          JMP      REAS1    ;OK
30 10304 030157          LDA      2,FPRET
31 10305 006114          EHALT             ;BUSY DOESN'T RETURN TO ZERO
32                                         ;AC2= CALL OF "RDSEC".
33
34 10306 060361 REAS1:   NIOP     XFDD     ;CLEAR DATA BUFFER
35 10307 126440          SUBC     1,1      ;AC1:= 0 = READ COMMAND
36 10310 020154          LDA      0,FSAV0  ;AC0:= SECTOR NO.
37 10311 123000          ADD      1,0
38 10312 061161          DOAS     0,XFDD   ;READ SECTOR
39
40 10313 006063          TIMSK             ;WAIT FOR DONE = 1 (MAX 1.5 SEC)
41 10314 002734          1500.
42 10315 063661          SKPDN    XFDD
43 10316 000402          JMP      .+2      ;ERROR
44 10317 000403          JMP      REAS2    ;OK
45 10320 030157          LDA      2,FPRET
46 10321 006114          EHALT             ;DONE ISN'T SET AFTER A READ
47                                         ;AC2= CALL OF "RDSEC".
48
49 10322 006116 REAS2:   STATN             ;CHECK STATUS
50 10323 140000          140000
51 10324 000407          JMP      ROUT2    ;HARD ERROR, AC0:= RECEIVED STATUS
52
53 10325 024230          LDA      1,FOTKS  ;AC1:= 010052
54 10326 107404          AND      0,1,SZR
55 10327 000403          JMP      ROUT1    ;ADR. FIELD/POS./TIME OUT/PARITY ERROR
56
57 10330 072461          DIC      2,XFDD   ;GET DATA FIELD SYNC BYTE
58 10331 010157          ISZ      FPRET
59 10332 010157 ROUT1:   ISZ      FPRET
60 10333 105000 ROUT2:   MOV      0,1      ;MOVE STATUS
61 10334 020154          LDA      0,FSAV0
62 10335 002157          JMP      @FPRET   ;RETURN
```

```
| 0111  .MAIN
01
02                ;PROCEDURE CALL:  SETDO
03                ;
04                ;THE ROUTINE WILL SET FDD-DONE BY FIRST WAITING FOR BUSY = 0
05                ;AND THEN TRANSMITTING A RECALIBRATE COMMAND. NORMAL RETURN FROM
06                ;THE ROUTINE IS CALL+2 WHEN DONE IS SET WITHIN 2 SEC.
07                ;
08                ;RETURN:      .+1 IF DONE ISN'T SET
09                ;             .+2 IF DONE IS SET
10                ;
11                ;DESTROYED:  AC3
12
13 10336 054157 DOSET:  STA     3,FPRET
14 10337 040154         STA     0,FSAV0
15 10340 044155         STA     1,FSAV1
16 10341 050156         STA     2,FSAV2
17
18 10342 010157         ISZ     FPRET   ;INCREMENT RETURN ADDRESS
19
20 10343 006063         TIMSK           ;WAIT FOR BUSY = 0 (MAX. 2 SEC)
21 10344 003720         2000.
22 10345 063561         SKPBZ   XFDD
23 10346 000402         JMP     .+2     ;ERROR
24 10347 000403         JMP     DOSE1   ;OK
25 10350 030157         LDA     2,FPRET
26 10351 006114         EHALT           ;BUSY DOESN'T RETURN TO ZERO
27                                      ;AC2= CALL OF "SETDO".
28 10352 020224 DOSE1:  LDA     0,FO1K0 ;AC0:= 1000 = RECAL-COMMAND
29 10353 061161         DOAS    0,XFDD
30
31 10354 006063         TIMSK           ;WAIT FOR DONE = 1 (MAX. 2 SEC)
32 10355 003720         2000.
33 10356 063661         SKPDN   XFDD
34 10357 014157         DSZ     FPRET   ;DONE ISN'T SET WITHIN 2 SEC, DECRE-
35                                      ;MENT RETURN ADDRESS TO CALL+1
36 10360 020154         LDA     0,FSAV0
37 10361 024155         LDA     1,FSAV1
38 10362 030156         LDA     2,FSAV2
39 10363 002157         JMP     @FPRET
40
41
42
43
44
45                ;PROCEDURE CALL: WRSEC
46                ;
47                ;THE ROUTINE WILL TRANSMIT THE DATA FIELD CHAR. IN "DFIEL",
48                ;THE 128 DATA-BYTES IN "WRBUF", TO THE FDC, AND THEN WRITE
49                ;IN SECTOR NO. SPECIFIED IN AC0.
50                ;FDC-BUSY/DONE AND STATUS ARE CHECKED BEFORE AND AFTER WRITING.
51                ;IN CASE OF TIME OUT (SECTOR NOT FOUND), THE WRITE COMMAND WILL
52                ;BE REPEATED UP TO "TRIES" TIMES.
53                ;
54                ;ENTRY:   AC0 = SECTOR NO. TO WRITE
55                ;
56                ;RETURN:  .+1
57                ;
58                ;DESTROYED: AC3
59
60 10364 054163 WRISC:  STA     3,SPRET
61 10365 040160         STA     0,SSAV0
62 10366 044161         STA     1,SSAV1
63 10367 050162         STA     2,SSAV2
```

```
! 0112  .MAIN
01
02 10370 006063          TIMSK            ;WAIT FOR BUSY = 0 (MAX 2 SEC.)
03 10371 003720          2000.
04 10372 063561          SKPBZ   XFDD
05 10373 000402          JMP     .+2      ;ERROR
06 10374 000403          JMP     WRIS1    ;OK
07 10375 030163          LDA     2,SPRET
08 10376 006114          EHALT            ;BUSY DOESN'T RETURN TO ZERO
09                                        ;AC2= CALL OF "WRSEC".
10 10377 006116 WRIS1:   STATN            ;CHECK STATUS
11 10400 177777          177777
12 10401 000402          JMP     .+2      ;ERROR
13 10402 000403          JMP     WRIS2    ;OK
14 10403 030163          LDA     2,SPRET
15 10404 006114          EHALT            ;STATUS ISN'T CORRECT. AC0 = RE-
16                                        ;CEIVED STATUS, AC1 = "NON"-EXPEC-
17                                        ;TED STATUS, AC2= CALL OF "WRSEC".
18 10405 024233 WRIS2:   LDA     1,DFIEL  ;DATA FIELD CHAR.
19 10406 067061          DOC     1,XFDD   ;SET DATA FIELD CHAR. REGISTER
20 10407 024206          LDA     1,FD064  ;AC1:= 64.
21 10410 044446          STA     1,CHACW  ;INITIALIZE CHARCOUNTER
22 10411 030234          LDA     2,AWBUF  ;AC2:= ADDRESS OF WRITE-BUFFER
23
24 10412 025000 WRIS3:   LDA     1,0,2    ;FILL DATA-BUFFER
25 10413 066061          DOB     1,XFDD
26 10414 151400          INC     2,2
27 10415 025000          LDA     1,0,2
28 10416 066061          DOB     1,XFDD
29 10417 151400          INC     2,2
30 10420 014436          DSZ     CHACW
31 10421 000771          JMP     WRIS3    ;FILL ON
32 10422 020255          LDA     0,TRIES
33 10423 040434          STA     0,WRICO  ;SET RE-WRITE COUNTER
34
35 10424 020160 WRIS4:   LDA     0,SSAV0  ;AC0:= SECTOR NO.
36 10425 024223          LDA     1,FD400  ;AC1:= 400 = WRITE COMMAND
37 10426 107000          ADD     0,1
38 10427 065161          DOAS    1,XFDD   ;WRITE ONE SECTOR
39 10430 006063          TIMSK            ;WAIT FOR DONE = 1 (MAX 3 SEC)
40 10431 005670          3000.
41 10432 063661          SKPDN   XFDD
42 10433 000402          JMP     .+2      ;ERROR
43 10434 000403          JMP     WRIS5    ;OK
44 10435 030163          LDA     2,SPRET
45 10436 006114          EHALT            ;DONE ISN'T SET AFTER EXECUTION OF
46                                        ;A WRITE SECTOR COMMAND, AC2= CALL OF
47                                        ;"WRSEC".
48
49 10437 006116 WRIS5:   STATN            ;CHECK STATUS
50 10440 177777          177777
51 10441 000402          JMP     .+2      ;ERROR
52 10442 000410          JMP     WRIS7    ;OK
53 10443 034173          LDA     3,FD002  ;AC3:= 2
54 10444 117415          AND#    0,3,SNR  ;SKIP IF TIME OUT
55 10445 000404          JMP     WRIS6
56 10446 014411          DSZ     WRICO    ;SKIP IF GIVE UP
57 10447 000755          JMP     WRIS4    ;TRY TO WRITE AGAIN
58 10450 030163          LDA     2,SPRET
59 10451 006114 WRIS6:   EHALT            ;STATUS ISN'T CORRECT AFTER WRITE-
60                                        ;COMMAND. AC0= RECEIVED STATUS, AC1=
61                                        ;"NON"-EXPECTED STATUS, AC2= CALL OF
62                                        ;"WRSEC".
```

```
| 0113  .MAIN
01
02 10452 020160 WRIS7:  LDA     0,SSAV0
03 10453 024161         LDA     1,SSAV1
04 10454 030162         LDA     2,SSAV2
05 10455 002163         JMP     @SPRET
06
07 10456 000000 CHACN:  0               ;CHARCOUNTER
08 10457 000005 WRICO:  5               ;RE-WRITE COUNTER
09
10
11
12              ;PROCEDURE CALL: CALSEC
13              ;
14              ;ROUTINE TO CALCULATE A NEW SECTOR NO. FROM THE PREVIOUS ONE.
15              ;THE CALCULATION IS DONE IN THE FOLLOWING WAY (IF FIRST
16              ;ENTRY = 1):
17              ;       1,3,5,7,.....25,2,4,6,8,.....24,26.
18              ;
19              ;ENTRY:         AC0= PREVIOUS SECTOR NO.
20              ;
21              ;RETURN:        CALL+2 IF ALL 26. SECTORS OF THE TRACK ARE
22              ;               ACCESSED ELSE
23              ;               CALL+1 WITH AC0= NEW SECTOR NO.
24              ;
25              ;DESTROYED:     AC3
26
27
28 10460 044420 CLSEC:  STA     1,CSEC1
29 10461 050420         STA     2,CSEC2
30 10462 030173         LDA     2,FD002
31 10463 143000         ADD     2,0
32 10464 024235         LDA     1,MXSEC
33 10465 147000         ADD     2,1
34 10466 106415         SUB#    0,1,SNR ;SKIP IF NEW SECTOR <> 28.
35 10467 000405         JMP     CSOU1
36 10470 024235         LDA     1,MXSEC
37 10471 106433         SUBZ#   0,1,SNC ;SKIP IF NEW SECTOR <= 26.
38 10472 020173         LDA     0,FD002
39 10473 000402         JMP     .+2
40 10474 175400 CSOU1:  INC     3,3
41 10475 024403         LDA     1,CSEC1
42 10476 030403         LDA     2,CSEC2
43 10477 001400         JMP     0,3
44
45 10500 000000 CSEC1:  0
46 10501 000000 CSEC2:  0
```

```
| 0114  .MAIN
01
02                ;PROCEDURE CALL: SOFER
03                ;
04                ;THE PROCEDURE IS USED TO ADMINSTER THE HANDLING OF RE-READS
05                ;(AND RE-WRITES) OF A SECTOR. UP TO "WTRIES" RE-WRITES, EACH
06                ;FOLLOWED BY UP TO "TRIES" RE-READS, ARE ALLOWED BEFORE AN ERROR-
07                ;HALT WILL APPEAR. THE TOTAL NUMBER OF RE-READS SINCE INITIALIZE-
08                ;ING WILL BE COUNTED IN "TOTCO" (PAGE ZERO).
09                ;BEFORE ROUTINE-USE, "RERCO" HAS TO BE SET TO 0, "REWCO" TO 0
10                ;IF "WTRIES" RE-WRITES ARE ALLOWED. IF THE "USER"-PROGRAM ONLY
11                ;READS, "REWCO" HAS TO BE SET TO "WTRIES", TO INDICATE, THAT
12                ;NO RE-WRITES ARE ALLOWED. "TOTCO" HAS TO BE INITIALIZED TO 0.
13                ;
14                ;ENTRY:          AC1= ERROR INDICATOR (AS DELIVERED BY PROCE-
15                ;                DURE "RDWRI").
16                ;
17                ;RETURN:         .+1  GIVE UP RETURN (AFTER ERRORHALT)
18                ;                .+2  RE-READ COUNTER IS STILL <= "TRIES"
19                ;                .+3  RE-READ COUNTER= "TRIES" AND RE-WRITE COUN-
20                ;                     TER IS STILL <= "WTRIES".
21                ;
22                ;DESTROYED:      AC3
23
24
25 10502 054163 SFTER:  STA      3,SPRET
26 10503 040160         STA      0,SSAV0
27 10504 044161         STA      1,SSAV1
28 10505 050162         STA      2,SSAV2
29
30 10506 010241         ISZ      RERCO    ;INCREMENT READ COUNTER
31 10507 010242         ISZ      TOTCO    ;INCREMENT TOTAL READ COUNTER
32 10510 030242         LDA      2,TOTCO
33 10511 151113         MOVL#    2,2,SNC ;SKIP IF TOTAL READ COU. > 32567
34 10512 000404         JMP      SFTE1
35
36 10513 030163         LDA      2,SPRET
37 10514 006114         EHALT             ;NO OF READ-TRIES > 32567. DIS-
38                                        ;CETTE OR DRIVE TOO BAD.AC2= CALL
39                                        ;OF "SOFER".
40 10515 000461         JMP      SFCU1    ;RETURN CALL + 1
41
42 10516 034241 SFTE1:  LDA      3,RERCO ;AC3:= RE-READ COUNTER
43 10517 030255         LDA      2,TRIES ;AC2:= "TRIES"
44 10520 172432         SUBZ#    3,2,SZC ;SKIP IF READ COUNTER > "TRIES"
45
46 10521 000454         JMP      SFCU2    ;RETURN +2
47
48 10522 010240         ISZ      REWCO    ;INCREMENT RE-WRITE COUNTER
49 10523 034240         LDA      3,REWCO ;AC3:= RE-WRITE COUNTER
50 10524 030256         LDA      2,WTRIES        ;AC2:= 3
51 10525 172432         SUBZ#    3,2,SZC ;SKIP IF WRITE COUNTER > WTRIES
52
53 10526 000446         JMP      SFCU3    ;RETURN +3
54
55 10527 006144         CURTR             ;GET CURRENT TRACK NO TO AC2
56 10530 145000         MOV      2,1
57
58 10531 030161         LDA      2,SSAV1 ;LOAD ERROR-INDICATOR
59 10532 151014         MOV#     2,2,SZR
60 10533 000404         JMP      SFTE2
```

```
I 0115  .MAIN
01
02 10534 030163          LDA     2,SPRET
03 10535 006114          EHALT           ;DATA WRONG ERROR. THE DATA
04                                       ;READ IS STILL WRONG AFTER "TRIES"*
05                                       ;CHECKREAD. AC0= SECTOR NO,
06 10536 000440          JMP     SFOU1   ;AC1= TRACK NO,AC2= CALL OF "SOFER".
07                                       ;*************  IT MAY BE A FAULTHY
08                                       ;TEST DISCETTE  ***********
09
10 10537 034227 SFTE2:   LDA     3,F010K ;AC3:= 10000
11 10540 157405          AND     2,3,SNR
12 10541 000404          JMP     SFTE3
13
14 10542 030163          LDA     2,SPRET
15 10543 006114          EHALT           ;ADDRESS FIELD ERROR AFTER "TRIES"*
16                                       ;CHECKREAD. AC0= SECTOR NO.,
17                                       ;AC1= TRACK NO,AC2= CALL OF "SOFER".
18 10544 000432          JMP     SFOU1
19
20 10545 034217 SFTE3:   LDA     3,F0040 ;AC3:= 000040
21 10546 157405          AND     2,3,SNR
22 10547 000404          JMP     SFTE4
23
24 10550 030163          LDA     2,SPRET
25 10551 006114          EHALT           ;PARITY ERROR AFTER "TRIES"* CHECK-
26                                       ;READ. AC0= SECTOR NO.,AC1= TRACK NO.
27                                       ;AC2= CALL OF "SOFER".
28 10552 000424          JMP     SFOU1
29
30 10553 034216 SFTE4:   LDA     3,F0010 ;AC3:= 000010
31 10554 157405          AND     2,3,SNR
32 10555 000404          JMP     SFTE5
33
34 10556 030163          LDA     2,SPRET
35 10557 006114          EHALT           ;POSITION ERROR AFTER "TRIES"* CHECK-
36                                       ;READ.AC0= SECTOR NO., AC1= TRACK
37 10560 000416          JMP     SFOU1   ;NO., AC2= CALL OF "SOFER".
38
39 10561 034173 SFTE5:   LDA     3,F0002 ;AC3:= 000002
40 10562 157405          AND     2,3,SNR
41 10563 000404          JMP     SFTE6
42 10564 030163          LDA     2,SPRET
43 10565 006114          EHALT           ;TIME OUT AFTER "TRIES" * CHECKREAD
44                                       ;AC0= SECTOR NO,AC1= TRACK NO
45 10566 000410          JMP     SFOU1   ;AC2= CALL OF "SOFER".
46
47 10567 034162 SFTE6:   LDA     3,SSAV2 ;AC3:= SYNCBYTE
48 10570 054412          STA     3,RECSY ;STORE SYNCBYTE
49 10571 030163          LDA     2,SPRET
50 10572 006114          EHALT           ;DATA FIELD SYNC ERROR AFTER "TRIES"*
51                                       ;CHECKREAD.AC0= SECTOR NO, AC1= TRACK
52                                       ;NO.AC2= CALL OF "SOFER",LOC. "RECSY"
53                                       ;= SYNC BYTE RECEIVED (SHOULD
54 10573 000403          JMP     SFOU1   ;BE 373)
55
56 10574 010163 SFOU3:   ISZ     SPRET
57 10575 010163 SFOU2:   ISZ     SPRET
58 10576 020160 SFOU1:   LDA     0,SSAV0
59 10577 024161          LDA     1,SSAV1
60 10600 030162          LDA     2,SSAV2
61 10601 002163          JMP     @SPRET
62
63 10602 000000 RECSY:   0               ;ERRONEOUS SYNC BYTE RECEIVED
```

```
| 0116  .MAIN
01
02                ;PROCEDURE CALL: RDWRI
03                ;
04                ;THE ROUTINE WILL TRY TO READ THE SECTOR NO. SPECIFIED IN AC0,
05                ;AND COMPARE THE DATA WITH THE DATA IN WRITE-BUFFER, BYTE BY
06                ;BYTE. DEPENDING ON THE RESULT, DIFFERENT RETURNS ARE USED.
07                ;
08                ;ENTRY:     AC0= SECTOR NO. TO READ
09                ;
10                ;RETURN:    .+1  HARD ERROR AFTER READ. AC1= RECEIVED
11                ;               STATUS.
12                ;           .+2  SOFT ERROR AFTER READ
13                ;                ADDRESS FIELD ERROR - AC1= 10000
14                ;                PARITY ERROR - AC1= 40
15                ;                POSITION ERROR - AC1= 10
16                ;                TIME OUT -  AC1= 2
17                ;                DATA WRONG ERROR - AC1= 0
18                ;                DATA SYNC ERROR - AC1= -1,AC2= SYNC BYTE REC.
19                ;           .+3  NORMAL RETURN - READ OK
20                ;
21                ;DESTROYED: AC1,AC2,AC3
22
23 10603 054163 REAWR:  STA     3,SPRET
24 10604 040160          STA     0,SSAV0
25 10605 044161          STA     1,SSAV1
26 10606 050162          STA     2,SSAV2
27
28 10607 006140          RDSEC           ;READ SECTOR SPEC. IN AC0
29 10610 000433          JMP     HAERR   ;HARD ERROR - AC1= STATUS
30 10611 000431          JMP     REERR   ;ADDRESS FIELD/PARITY/POS./T.OUT ERROR
31                                       ;OK - AC2= DATA SYNC BYTE READ
32 10612 024251          LDA     1,DTINH ;AC1= 0 IF DATACHECK ELSE <> 0
33 10613 125014          MOV#    1,1,SZR
34 10614 000422          JMP     REDOU   ;NO DATACHECK
35 10615 024222          LDA     1,FO377 ;AC1:= 377
36 10616 133400          AND     1,2
37 10617 024233          LDA     1,DFIEL ;DATA FIELD SYNC WRITTEN
38 10620 132414          SUB#    1,2,SZR
39 10621 000420          JMP     SYNCE   ;DATA FIELD SYNC ERROR
40
41 10622 024211          LDA     1,FD128 ;AC1:= 128.
42 10623 044422          STA     1,CHACR
43 10624 034234          LDA     3,AWBUF ;ADDRESS OF WRITE BUFFER
44
45 10625 024222 REDAG:  LDA     1,FO377 ;AC1:= 377
46 10626 071461          DIB     2,XFDC
47 10627 133400          AND     1,2     ;BYTE READ
48 10630 025400          LDA     1,0,3   ;BYTE WRITTEN
49 10631 146414          SUB#    2,1,SZR
50 10632 000406          JMP     DTERR   ;DATA NOT EQUAL - ERROR
51
52 10633 175400          INC     3,3
53 10634 014411          DSZ     CHACR
54 10635 000770          JMP     REDAG   ;READ NEXT BYTE
55
56                                       ;128 CHRS READ AND CHECKED
57 10636 010163 REDOU:  ISZ     SPRET   ;INCREMENT RETURN
58 10637 000403          JMP     REERR
59
60 10640 126441 DTERR:  SUBO    1,1,SKP ;AC1:= 0
61 10641 126000 SYNCE:  ACC     1,1     ;AC1:= -1
62 10642 010163 REERR:  ISZ     SPRET   ;INCREMENT RETURN
63
64 10643 020160 HAERR:  LDA     0,SSAV0
65 10644 002163          JMP     @SPRET  ; RETURN
66
67 10645 000000 CHACR:  0               ;READ CHARACTER COUNTER
```

```
| 0117  .MAIN
01
02                ;PROCEDURE CALL: CALTR
03                ;
04                ;THE PROCEDURE "TRCAL" WILL CALCULATE A NEW TRACK NO. TO AC1
05                ;DEPENDING ON THE PREVIOUS TRACK NO. (IN AC2 AT ENTRY).THE
06                ;CALCULATION IS DONE IN A MANNER TO PRODUCE WORST CASE HEAD-
07                ;MOVEMENT, AND GOES LIKE THIS:
08                ;
09                ;       STARTING WITH 0
10                ;       NEXT:  76,1,75,2,74,3,73,4,.....36,40,37,39,38.
11                ;
12                ;ENTRY:       AC2= LAST USED TRACK NO.
13                ;
14                ;RETURN:      +1  IF AC2 AT ENTRY <> 38, WITH AC1= NEW TRACK NO
15                ;
16                ;             +2  IF AC2 AT ENTRY = 38,  NO NEW NO. IS CALCUL.
17                ;
18                ;DESTROYED:   AC1,AC3
19
20
21 10646 054167 TRCAL:  STA     3,TPRET
22
23 10647 034203         LDA     3,FD038 ;AC3:= 38.
24 10650 172415         SUB#    3,2,SNR ;SKIP IF OLD TRACK NO <> 38
25 10651 000416         JMP     TCOU2   ;INCREMENT RETURN AND EXIT
26 10652 156033         ADCZ#   2,3,SNC ;SKIP IF OLD TRACK NO < 38.
27
28 10653 000404         JMP     TRCA1   ;OLD TRACK NO > 38.
29
30 10654 024236         LDA     1,MXTRA ;OLD TRACK NO < 38.,AC1:= 76.
31 10655 146400         SUB     2,1     ;AC1:= 76 - AC2
32 10656 000412         JMP     TCOU1   ;RETURN
33
34 10657 175400 TRCA1:  INC     3,3     ;AC3:= 39.
35 10660 156414         SUB#    2,3,SZR ;SKIP IF OLD TRACK NO = 39.
36
37 10661 000403         JMP     TRCA2   ;OLD TRACK NO > 39.
38
39 10662 024203         LDA     1,FD038 ;OLD TRACK NO = 39., AC1:= 38.
40 10663 000405         JMP     TCOU1   ;RETURN
41
42 10664 024236 TRCA2:  LDA     1,MXTRA ;AC1:= 76.
43 10665 125400         INC     1,1
44 10666 146401         SUB     2,1,SKP ;AC1:= 77. - AC2
45
46 10667 010167 TCOU2:  ISZ     TPRET   ;INCREMENT RETURN ADDRESS
47 10670 002167 TCOU1:  JMP     @TPRET
```

```
| 0118  .MAIN
01
02 10671 000000 NOPAS:  0
03 10672 007026 AXPAS:  XPASS
04 10673 007061 APASB:  PASSB
05
06              ; ROUTINE TO SET-UP THE STANDARD PASS-ADMINISTRATOR
07
08 10674 054170 SETPA:  STA     3,RETU0
09 10675 030776         LDA     2,APASB
10 10676 102520         SUBZL   0,0     ;AC0:= 1
11 10677 041000         STA     0,0,2   ;SET NO. OF RUNS BEFORE MESS (COUNTER)
12 10700 041001         STA     0,1,2   ;SET NO. OF RUNS BEFORE MESS (CONST.)
13 10701 102440         SUBO    0,0     ;AC0:= 0
14 10702 040251         STA     0,DTINH ;ENABLE DATACHECK
15 10703 040252         STA     0,NEGER ;ENABLE ERRORANNOUNCING IN LOOP A0510
16 10704 041002         STA     0,2,2   ;CLEAR PASS COUNTER
17 10705 050246         STA     2,RUNMK ;MAKE RUN-MARK <> 0
18 10706 006071 RNMES:  CGUES
19 10707 011461         TNRUN           ;"NO. OF PASSES: "
20 10710 011471         DNRUN           ;"# OF PASS:"
21 10711 010723         DFRUN           ;SUGGESTED ANSWER
22 10712 006053         CTDEC           ;PRINT SUGGESTED ANSWER
23 10713 006057         CDDEC
24 10714 006105         CGTDC           ;GET DECIMAL INPUT
25 10715 000402         JMP     .+2     ;SUGGESTED ANSWER OK
26 10716 000770         JMP     RNMES   ;ILLEGAL INPUT
27 10717 020075         LDA     0,CIGIN
28 10720 040751         STA     0,NOPAS
29 10721 062677         IORST           ;CLEAR STATUS REG.
30 10722 002170         JMP     @RETU0  ;RETURN
31
32 10723 000001 DFRUN:  1               ;PRE-DEFINED NO. OF PASSES
33
34
35
36
37
38              ; PROCEDURE TO ADMINSTER THE PASS-EXECUTION
39
40 10724 054170 PAADM:  STA     3,RETU0
41 10725 006745         JSR     @AXPAS  ;USE STANDARD PASS-ADMINISTATOR
42 10726 030745         LDA     2,APASB
43 10727 025002         LDA     1,2,2   ;AC1:= NO OF PASSES UNTIL NOW
44 10730 020741         LDA     0,NOPAS ;AC0:= WANTED NO OF PASSES
45 10731 122032         ADCZ#   1,0,SZC ;SKIP IF AC1 >= AC0
46 10732 002170         JMP     @RETU0  ;GO TO NEXT PASS
47 10733 006043         CCRLF
48 10734 006050         CHAAT
49 10735 006061         CWAIT
50 10736 000213         FD1K0           ;WAIT 1 SEC
51 10737 006050         CHAAT
52 10740 002401         JMP     @.+1    ;ASK FOR NEW STARTADDRESS
53 10741 004740         SWISA
```

```
| 0119  .MAIN
01
02 10742 054170 OPDRV:  STA     3,RETU0
03 10743 006043         CCRLF
04 10744 006046         CDICL
05 10745 006040         CMESS
06 10746 010764         TOPDR
07 10747 006044         CDISP
08 10750 011001         DOPDR
09 10751 006043         CCRLF
10 10752 002170         JMP     @RETU0
11
12 10753 054170 CLDRV:  STA     3,RETU0
13 10754 006043         CCRLF
14 10755 006046         CDICL
15 10756 006040         CMESS
16 10757 011011         TCLDR
17 10760 006044         CDISP
18 10761 011027         DCLDR
19 10762 006043         CCRLF
20 10763 002170         JMP     @RETU0
21
22              TOPDR:  .TXTE   !*** START WITH OPEN DRIVE!
   10764 125252
   10765 120252
   10766 152123
   10767 151101
   10770 120324
   10771 144727
   10772 044324
   10773 147640
   10774 142520
   10775 120116
   10776 151104
   10777 053311
   11000 000305
23
24              DOPDR:  .TXTE   !*** OPEN DRIVE!
   11001 125252
   11002 120252
   11003 050317
   11004 047305
   11005 042240
   11006 144722
   11007 142526
   11010 000000
25
26              TCLDR:  .TXTE   !*** START WITH LOADED DRIVE!
   11011 125252
   11012 120252
   11013 152123
   11014 151101
   11015 120324
   11016 144727
   11017 044324
   11020 146240
   11021 040717
   11022 142504
   11023 120104
   11024 151104
   11025 053311
   11026 000305
```

```
| 0120  .MAIN
01
02                DCLDR:  .TXTE   !*** LOAD DRIVE!
   11027 125252
   11030 120252
   11031 147714
   11032 042101
   11033 042240
   11034 144722
   11035 142526
   11036 000000
03
04 11037 054170 DCMES:  STA     3,RETU0
05 11040 006071         CQUES
06 11041 011063         TDACH
07 11042 011074         DDACH
08 11043 011231         CHARY
09 11044 006041         CCHAR
10 11045 006045         CDOUT
11 11046 006106         CGTSC
12 11047 000413         JMP     DCMEO   ;SUGGESTED ANSWER OK
13 11050 000770         JMP     DCMES+1 ;ILLEGAL INPUT
14 11051 176000         ADC     3,3
15 11052 020075         LDA     0,DIGIN
16 11053 024556         LDA     1,CHARY
17 11054 030556         LDA     2,CHARN
18 11055 106415         SUB#    0,1,SNR
19 11056 000404         JMP     DCMEO   ;DATACHECK WANTED
20 11057 112414         SUB#    0,2,SZR
21 11060 000760         JMP     DCMES+1 ;NOT Y OR N
22 11061 054251         STA     3,DTINH ;DATACHECK NOT WANTED
23 11062 002170 DCMEO:  JMP     @RETU0
24
25
26                TDACH:  .TXTE   !DATACHECK (Y/N): !
   11063 040504
   11064 040724
   11065 044303
   11066 141705
   11067 120113
   11070 054450
   11071 047257
   11072 035251
   11073 000240
27
28                DDACH:  .TXTE   !DATACHECK: !
   11074 040504
   11075 040724
   11076 044303
   11077 141705
   11100 035113
   11101 000240
```

```
I 0121  .MAIN
01
02                ;TESTLOOP-BUILDER FOR START OF TEST IN STARTADDRESS  400
03
04
05 11102 165000 FTES0:  MOV     3,1
06 11103 006272         CSAMS           ;TYPE START ADDRESS
07 11104 006523         JSR     @ICPDR
08 11105 006516         JSR     @ISTPA  ;SET UP PASS-ADMINISTRATOR
09 11106 006503         JSR     @IGR0A  ;EXECUTE DIAGNOSTIC TEST PART 0A
10 11107 006503         JSR     @IGR0B  ;EXECUTE DIAGNOSTIC TEST PART 0B
11 11110 006503         JSR     @IGR0C  ;EXECUTE DIAGNOSTIC TEST PART 0C
12 11111 006503         JSR     @IGR0D  ;EXECUTE DIAGNOSTIC TEST PART 0D
13
14 11112 006043         CCRLF
15 11113 006046         CDICL
16 11114 006040         CMESS
17 11115 011525         DIDIS           ;"INSERT TESTDISC."
18 11116 006040         CMESS
19 11117 011457         TCSPA           ;", "
20 11120 006040         CMESS
21 11121 011603         DWPOF           ;""WRITE PRO" OFF"
22 11122 006044         CDISP
23 11123 011525         DIDIS           ;"INSERT TESTDISC."
24 11124 006047         CDATT
25 11125 006046         CDICL
26 11126 006044         CDISP
27 11127 011603         DWPOF           ;""WRITE PRO" OFF
28
29 11130 006062         WATOP           ;WAIT FOR OPERATOR ACTION
30 11131 020000         020000
31 11132 006043         CCRLF
32
33 11133 006462 FTS01:  JSR     @IGR01  ;EXECUTE DIAGNOSTICTEST PART 1
34 11134 006462         JSR     @IGR02  ;EXECUTE DIAGNOSTICTEST PART 2
35 11135 006462         JSR     @IGR03  ;EXECUTE MASTERREAD (RANDOMLY)
36 11136 006462         JSR     @IGR04  ;EXECUTE MASTERREAD (HEADOSCILLATING)
37 11137 006462         JSR     @IGR05  ;EXECUTE WRITE/READ TEST
38 11140 006466         JSR     @IGNDS  ;RE-GENERATE TEST-DISCETTE
39 11141 006136         RECAL
40 11142 006463         JSR     @IPADM  ;GOTO PASS-ADMINISTRATOR
41 11143 000770         JMP     FTS01
42
43
44
45
46                ;TESTLOOP-BUILDER FOR START IN STARTADDRESS 401
47
48 11144 165000 FTES1:  MOV     3,1
49 11145 006272         CSAMS           ;TYPE STARTADDRESS
50 11146 006461         JSR     @ICPDR
51 11147 006454         JSR     @ISTPA  ;SET UP PASS-ADMINISTATOR
52
53 11150 006441 FTS11:  JSR     @IGR0A
54 11151 006441         JSR     @IGR0B
55 11152 006441         JSR     @IGR0C
56 11153 006441         JSR     @IGR0D
57 11154 006441         JSR     @IGR01
58
59 11155 006450         JSR     @IPADM  ;GOTO PASS ADMINISTATOR
60 11156 000772         JMP     FTS11   ;REPEAT PASS
```

```
1 0122  .MAIN
01
02
03                ;TESTLOOP-BUILDER FOR START IN STARTADDRESS 402
04 11157 165000 FTES2:  MOV     3,1
05 11160 006072         CSAMS
06 11161 006447         JSR     @ICLDR          ;TYPE START ADDRESS
07 11162 006441         JSR     @ISTPA
08 11163 006441         JSR     @IDCME  ;SET UP PASS ADMINISTATOR
09                                              ;ASK FOR DATACHECK
10 11164 006425 FTS21:  JSR     @IGR0A
11 11165 006426         JSR     @IGR0C
12 11166 006427         JSR     @IGR01
13 11167 006427         JSR     @IGR02
14 11170 006427         JSR     @IGR03
15 11171 006427         JSR     @IGR04
16
17 11172 006136         RECAL
18 11173 006432         JSR     @IPADM  ;RECALIBRATE DISC-DRIVE
19 11174 000770         JMP     FTS21   ;GOTO PASS ADMINISTATOR
20                                      ;REPEAT PASS
21
22
23                ;TESTLOOP-BUILDER FOR START IN STARTADDRESS 403
24 11175 165000 FTES3:  MOV     3,1
25 11176 006072         CSAMS
26 11177 006431         JSR     @ICLDR          ;TYPE START ADDRESS
27 11200 006423         JSR     @ISTPA  ;SET UP PASS ADMINISTRATOR
28
29 11201 006410 FTS31:  JSR     @IGR0A
30 11202 006411         JSR     @IGR0C
31 11203 006412         JSR     @IGR01
32 11204 006412         JSR     @IGR02
33 11205 006414         JSR     @IGR05
34 11206 006136         RECAL
35 11207 006416         JSR     @IPADM  ;RECALIBRATE DISC-DRIVE
36 11210 000771         JMP     FTS31   ;GOTO PASS ADMINISTRATOR
37                                      ;REPEAT PASS
38 11211 011776 IGR0A:  GR02A
39 11212 012165 IGR0B:  GR02B
40 11213 012200 IGR0C:  GR02C
41 11214 012320 IGR0D:  GR02D
42 11215 012336 IGR01:  GR021
43 11216 012475 IGR02:  GR022
44 11217 013017 IGR03:  GR023
45 11220 013432 IGR04:  GR024
46 11221 013524 IGR05:  GR025
47 11222 014236 IGR06:  GR026
48 11223 010674 ISTPA:  SETPA
49 11224 011037 IDCME:  DCMES
50 11225 010724 IPADM:  PAADM
51 11226 011651 IGNDS:  GNDSC
52 11227 010742 IOPDR:  OPDRV
53 11230 010753 ICLDR:  CLDRV
54 11231 000131 CHARY:  131
55 11232 000116 CHARN:  116
```

```
! 0123  .MAIN
01
02                ;TESTLOOP-BUILDER FOR START IN STARTADDRESS 404
03
04 11233 165000 FTES4:  MOV     3,1
05 11234 006072         CSAMS           ;TYPE START ADDRESS
06 11235 006773         JSR     @ICLDR
07 11236 006765         JSR     @ISTPA  ;SET UP PASS ADMINISTRATOR
08
09 11237 006752 FTS41:  JSR     @IGR0A
10 11240 006753         JSR     @IGR0C
11 11241 006754         JSR     @IGR01
12 11242 006754         JSR     @IGR02
13 11243 006754         JSR     @IGR03
14 11244 006754         JSR     @IGR04
15 11245 006754         JSR     @IGR05
16 11246 006760         JSR     @IGNDS  ;RE-GENERATE TEST DISCETTE
17 11247 006136         RECAL           ;RECALIBRATE DISC DRIVE
18 11250 006755         JSR     @IPADM  ;GOTO PASS ADMINISTRATOR
19 11251 000766         JMP     FTS41   ;REPEAT PASS
20
21                ;PROGRAM FOR CHANGE OF DEVICE CODE AFTER START OF PROG IN
22                ;STARTADDRESS  407
23
24 11252 020216 CDEVI:  LDA     0,F0010 ;AC0:= 000010
25 11253 040126         STA     0,SWREG ;SET COMMON SWITCH REG, AND DISABLE
26                                      ;PANEL-SWITCHES.
27 11254 006043         CCRLF
28 11255 006046         CDICL
29 11256 006040         CMESS
30 11257 011614         SETDC           ;"DEV.CODE= "
31 11260 006044         CDISP
32 11261 011614         SETDC
33 11262 006050         CHAAT
34 11263 006104         CGTOK           ;GET OCTAL NUMBER
35 11264 000766         JMP     CDEVI   ;ILLEGAL INPUT
36 11265 000765         JMP     CDEVI   ;ILLEGAL INPUT
37 11266 020075         LDA     0,DIGIN ;AC0:= NEW DEVICE CODE
38 11267 024244         LDA     1,ATBEG ;AC1:= ADDRESS OF PROG START
39 11270 030245         LDA     2,ATEND ;AC2:= ADDRESS OF PROG END
40
41 11271 006141         SETDV           ;CHANGE DEVICE CODE IN PROGRAM
42 11272 002401         JMP     @.+1
43 11273 004740         SWISA           ;ASK FOR NEW START ADDRESS
```

```
I 0124   .MAIN
01
02                ;PROGRAM FOR GENERATING OF THE TM-DISCETTE, STARTADDRESS 410
03
04 11274 020216 GENTM:  LDA     0,F0010 ;AC0:= 000010
05 11275 040126         STA     0,SWREG ;SET COMMON SWITCH REG, AND DISABLE
06                                      ;PANEL-SWITCHES.
07 11276 006043         CCRLF
08 11277 006046         CDICL
09 11300 006040         CMESS
10 11301 011431         MTGTE           ;"*** RC3600 TEST DISCETTE GENE-
11 11302 006043         CCRLF           ;RATOR ***"
12 11303 006040         CMESS
13 11304 011622         TINCD           ;"INSERT DISCETTE"
14 11305 006040         CMESS
15 11306 011457         TCSPA           ;", "
16 11307 006040         CMESS
17 11310 011603         DWPOF           ;""WRITE PRO" OFF"
18 11311 006044         CDISP
19 11312 011622         TINCD
20 11313 006047         CDATT
21 11314 006046         CDICL
22 11315 006044         CDISP
23 11316 011603         DWPOF
24 11317 102440         SUBO    0,0     ;AC0:= 0
25 11320 040246         STA     0,RUNMK ;CLEAR RUN-MARK
26 11321 006062         WATOP           ;WAIT FOR OPERATOR ACTION
27 11322 020000         020000
28 11323 006043         CCRLF
29
30 11324 006702         JSR     @IGNDS  ;GENERATE DISCETTE
31 11325 006136         RECAL           ;RECALIBRATE DISC DRIVE
32 11326 002401         JMP     @.+1
33 11327 004740         SWISA           ;ASK FOR NEW START ADDRESS
34
35                ;TEST LOOP BUILDER FOR START IN STARTADDRESS 405. IT WILL
36                ;WRITE AND CHECKREAD ALL SECTORS.
37
38 11330 165000 FTES5:  MOV     3,1
39 11331 006072         CSAMS
40 11332 102440         SUBO    0,0
41 11333 040242         STA     0,TOTCO
42 11334 006674         JSR     @ICLDR
43 11335 006666         JSR     @ISTPA
44 11336 004425         JSR     NEGLE   ;ASK FOR ERRORACTION
45
46 11337 006142 FTS51:  PTFLL
47 11340 006662         JSR     @IGR06
48 11341 006151         COMPT
49 11342 006660         JSR     @IGR06
50 11343 006136         RECAL
51 11344 006661         JSR     @IPADM
52 11345 000772         JMP     FTS51
```

```
I 0125  .MAIN
01
02                ;TESTLOOP BUILDER FOR START IN  STARTADDRESS 406. IT WILL READ
03                ;ALL SECTORS WITOUT DATACHECK.
04
05 11346 165000 FTES6:  MOV     3,1
06 11347 006072         CSAMS
07 11350 102440         SUBO    0,0
08 11351 040242         STA     0,TOTCO
09 11352 006656         JSR     @ICLDR
10 11353 006650         JSR     @ISTPA
11 11354 102000         ADC     0,0
12 11355 040251         STA     0,DTINH ;INHIBIT DATACHECK
13 11356 004405         JSR     NEGLE   ;ASK FOR ERRORACTION
14
15 11357 006643 FTS61:  JSR     @IGR06
16 11360 006136         RECAL
17 11361 006644         JSR     @IPADM
18 11362 000775         JMP     FTS61
19
20
21
22 11363 054170 NEGLE:  STA     3,RETU0
23 11364 006071         CQUES
24 11365 011407         TACERR
25 11366 011476         DACERR
26 11367 011232         CHARN
27 11370 006041         CCHAR
28 11371 006045         CCOUT
29 11372 006106         CGTSC
30 11373 000413         JMP     NEGLO
31 11374 000770         JMP     NEGLE+1
32 11375 176000         ADC     3,3
33 11376 020075         LDA     0,DIGIN
34 11377 024632         LDA     1,CHARY
35 11400 030632         LDA     2,CHARN
36 11401 106415         SUB#    0,1,SNR
37 11402 000404         JMP     NEGLO
38 11403 112414         SUB#    0,2,SZR
39 11404 000760         JMP     NEGLE+1
40 11405 054252         STA     3,NEGER
41 11406 002170 NEGLO:  JMP     @RETU0
42
43
44                TACERR: .TXTE   !ERRORSKIP (Y/N): !
   11407 151305
   11410 147722
   11411 051722
   11412 144513
   11413 120120
   11414 054450
   11415 047257
   11416 035251
   11417 000240
45
46                MGERR:  .TXTE   @FAULTY DISCETTE!@
   11420 040706
   11421 146125
   11422 054724
   11423 042240
   11424 051711
   11425 142703
   11426 152324
   11427 020705
   11430 000000
```

```
! 0126  .MAIN
01
02
03
04                MTGTE:  .TXTE   !<15><12>*** RC3600 TEST DISCETTE GENERATOR ***<15
   11431 005215
   11432 125252
   11433 120252
   11434 141722
   11435 033063
   11436 030060
   11437 152240
   11440 051705
   11441 120324
   11442 144504
   11443 141523
   11444 152305
   11445 142724
   11446 043640
   11447 047305
   11450 151305
   11451 152101
   11452 151317
   11453 125240
   11454 125252
   11455 005215
   11456 000000
05
06                TCSPA:  .TXTE   !, !
   11457 120254
   11460 000000
07
08                TNRUN:  .TXTE   !NO. OF PASSES: !
   11461 147516
   11462 120056
   11463 143317
   11464 050240
   11465 051501
   11466 142523
   11467 035123
   11470 000240
09
10                DNRUN:  .TXTE   !# OF PASS!
   11471 120243
   11472 143317
   11473 050240
   11474 051501
   11475 000123
11
12                DACERR: .TXTE   !ERRORSKIP: !
   11476 151305
   11477 147722
   11500 051722
   11501 144513
   11502 035120
   11503 000240
```

```
! 0127  .MAIN
01
02
                    TIDIS:  .TXTE   !INSERT TESTDISC,, "WRITE PRO" ON !
   11504 047311
   11505 142523
   11506 152322
   11507 152240
   11510 051705
   11511 042324
   11512 051711
   11513 027303
   11514 120254
   11515 153442
   11516 144722
   11517 142724
   11520 050240
   11521 147722
   11522 120042
   11523 047317
   11524 000240
03
04
                    DIDIS:  .TXTE   !INSERT TESTDISC.! _
   11525 047311
   11526 142523
   11527 152322
   11530 152240
   11531 051705
   11532 042324
   11533 051711
   11534 027303
   11535 000000
05
06
                    DWPON:  .TXTE   !"WRITE PRO" ON!
   11536 153442
   11537 144722
   11540 142724
   11541 050240
   11542 147722
   11543 120042
   11544 047317
   11545 000000
07
08
                    TISCD:  .TXTE   !INSERT SCRATCHDISC.,!
   11546 047311
   11547 142523
   11550 152322
   11551 051640
   11552 151303
   11553 152101
   11554 044303
   11555 144504
   11556 141523
   11557 126056
   11560 000000
09
10
                    LOP39:  .TXTE   !   IN LOOP A0390!
   11561 120240
   11562 144640
   11563 120116
   11564 147714
   11565 050317
   11566 040640
   11567 031460
   11570 032071
   11571 000000
```

```
| 0128  .MAIN
01
02                DISCD:  .TXTE   !INSERT SCR.DISC.!
   11572 047311
   11573 142523
   11574 152322
   11575 051640
   11576 151303
   11577 042056
   11600 051711
   11601 027303
   11602 000000
03
04                DWPOF:  .TXTE   !"WRITE PRO" OFF !
   11603 153442
   11604 144722
   11605 142724
   11606 050240
   11607 147722
   11610 120042
   11611 143317
   11612 120306
   11613 000000
05
06                SETDC:  .TXTE   !DEV.CODE= !
   11614 142504
   11615 027126
   11616 147703
   11617 142504
   11620 120275
   11621 000000
07
08                TINCD:  .TXTE   !INSERT DISCETTE!
   11622 047311
   11623 142523
   11624 152322
   11625 042240
   11626 051711
   11627 142703
   11630 152324
   11631 000305
09
10                RRPER:  .TXTE   !RE-READ %=!
   11632 142722
   11633 151055
   11634 040705
   11635 120104
   11636 136645
   11637 000000
11
12                LOP41:  .TXTE   !   IN LOOP A0410!
   11640 120240
   11641 144640
   11642 120116
   11643 147714
   11644 050317
   11645 040640
   11646 132060
   11647 030261
   11650 000000
```

```
| 0129  .MAIN
01
02                ;TESTDISCETTE(MASTER) GENERATOR PROGRAM
03                ;
04                ;THE PROGRAM WILL GENERATE A TESTDISCETTE BY WRITING THE PRE-
05                ;SCRIBED TESTPATTERN IN ALL TRACKS. AFTER WRITING A TRACK,IT
06                ;IS CHECK-READ UP TO "TRIES" TIMES IF ERROR,AND IF STILL ERROR,
07                ;IT IS RE-WRITTEN AND CHECKED, MAX 3 TIMES. DURING THE GENERA-
08                ;TION ALL RE-READS ARE COUNTED, AND THE FINAL NUMBER WILL BE
09                ;STATED AFTER A SUCCESSFUL GENERATION, TO INFORM ABOUT THE CON-
10                ;DITION OF THE DISCETTE (AND DRIVE).
11
12
13 11651 054170 GNDSC:  STA     3,RETU0
14
15 11652 006110         SETP0
16 11653 062677         IORST
17 11654 006061         CWAIT           ;WAIT 1 SEC
18 11655 000213         FD1K0
19 11656 006065         TIMRO           ;TEST STATUS AND WAIT UNTIL DRIVE
20 11657 000214         FD3K0           ;GOES ON-LINE (MAX 3 SEC)
21 11660 006116         STATN
22 11661 040000         040000
23 11662 006114         EHALT           ;DEVICE DIDN'T GO ON-LINE WITHIN 3 SEC
24
25 11663 006136         RECAL           ;RECALIBRATE DISC-DRIVE
26
27 11664 006142         PTFLL           ;FILL WRITE-BUFFER WITH PATTERNS
28
29 11665 126440         SUB0    1,1     ;AC1:= 0 = FIRST TRACK TO WRITE
30 11666 044242         STA     1,TOTCO ;CLEAR TOTAL RE-READ COUNTER
31 11667 044253         STA     1,WORK1 ;CLEAR SECTOR COUNTER
32
33 11670 006137 WPOS1:  TRPOS           ;POSITION HEAD TO TRACK SPEC. IN AC1
34 11671 102440         SUB0    0,0
35 11672 040240         STA     0,REWCO ;CLEAR RE-WRITE COUNTER
36
37 11673 102520 GWRS0:  SUBZL   0,0     ;AC0:= 1 = FIRST SECTOR TO WRITE
38
39 11674 042234 GWRS1:  STA     0,@AWBUF;WRITE SECTOR NO. AS FIRST BYTE IN
40                                      ;DATA BUFFER.
41 11675 010253         ISZ     WORK1   ;INCREMENT SECTOR COUNTER
42
43 11676 006143 GWRS2:  WRSEC           ;WRITE SECTOR SPEC. IN AC0, DATA IS
44                                      ;FETCHED FROM "WRBUF".
45
46 11677 006153         CALSEC          ;CALCULATE NEXT SECTOR
47 11700 000774         JMP     GWRS1   ;GO ON
48                                      ;TRACK FINISHED
49 11701 102520         SUBZL   0,0     ;AC0:= 1 = FIRST SECTOR TO READ
50 11702 042234 GWRS3:  STA     0,@AWBUF
51 11703 126440         SUB0    1,1
52 11704 044241         STA     1,RERCO ;CLEAR RE-READ COUNTER
53 11705 006146 GWRS4:  RDWRI           ;CHECKREAD SECTORS
54 11706 000435         JMP     GNDE1   ;HARD ERROR AFTER READ
55 11707 000437         JMP     GNDE2   ;SOFT ERROR
56
57 11710 006153         CALSEC          ;CALCULATE NEXT SECTOR NO.
58 11711 000771         JMP     GWRS3   ;GO ON
59                                      ;TRACK FINISHED
60
61 11712 006144         CURTR           ;GET CURRENT TRACK NO. TO AC2
62 11713 145400         INC     2,1     ;INCREMENT TRACK NO.
63 11714 020236         LDA     0,MXTRA ;AC1:= 76.
64 11715 122432         SUBZ#   1,0,SZC ;SKIP IF TRACK NO. > 76.
65 11716 000752         JMP     WPOS1   ;GO TO NEXT TRACK
```

```
| 0130  .MAIN
01
02 11717 034246          LDA     3,RUNMK
03 11720 175014          MOV#    3,3,SZR ;SKIP IF GEN-PROG ISN'T CALLED DURING TEST
04 11721 000420          JMP     GWOUT
05
06                                       ;ALL TRACKS ARE WRITTEN
07 11722 006043          CCRLF
08 11723 006046          CDICL
09 11724 006040          CMESS
10 11725 011766          MGEND           ;"MASTER-GEN. END"
11 11726 006043          CCRLF
12 11727 006044          CDISP
13 11730 011766          MGEND
14 11731 006047          CDATT
15
16 11732 020242          LDA     0,TOTCO
17 11733 024214          LDA     1,FD3K0 ;AC1:= 3000.
18 11734 106032          ADCZ#   0,1,SZC ;SKIP IF AC0 >= AC1
19 11735 000404          JMP     GWOUT
20 11736 006152          TRMES           ;TYPE TOTAL RE-READ PERCENT,RET +1 OR +2
21 11737 000401          JMP     .+1     ;NO SPECIAL RETURN IF % < 20
22 11740 006043          CCRLF
23
24 11741 006113 GWOUT:   LOOP
25 11742 002170          JMP     @RETU0
26
27
28
29
30 11743 121000 GNDE1:   MOV     1,0
31 11744 006114          EHALT           ;HARD ERROR AFTER CHECKREAD. AC0=
32                                       ;RECEIVED STATUS
33 11745 000405          JMP     GWERR   ;GIVE UP
34
35 11746 006147 GNDE2:   SOFER           ;EXAMINE THE READ AND WRITE COUNTERS.
36                                       ;HALT AND/OR RETURN +1,+2,+3.
37
38 11747 000403          JMP     GWERR   ;GIVE UP
39 11750 000735          JMP     GWRS4   ;TRY TO READ SECTOR AGAIN
40 11751 000722          JMP     GWRS0   ;TRY TO RE-WRITE TRACK AND READ SECTOR
41                                       ;AGAIN
42
43 11752 034246 GWERR:   LDA     3,RUNMK
44 11753 175014          MOV#    3,3,SZR ;SKIP IF GEN-PROG ISN'T CALLED DURING TEST
45 11754 000765          JMP     GWOUT
46 11755 006043          CCRLF
47 11756 006046          CDICL
48 11757 006040          CMESS
49 11760 011420          MGERR           ;"FAULTY DISCETTE!"
50 11761 006043          CCRLF
51 11762 006044          CDISP
52 11763 011420          MGERR
53 11764 006047          CDATT
54 11765 000754          JMP     GWOUT
55
56
57             MGEND:    .TXTE   !MASTER-GEN. END!
   11766 040515
   11767 152123
   11770 151305
   11771 043455
   11772 047305
   11773 120056
   11774 047305
   11775 020104
```

```
! 0131   .MAIN
01
02                          .EOT
```

```
  0132  .MAIN
01
02                ;LOOP: A0000 - A0150
03                ;
04                ;TEST OF BUS,CPU-SKIP,CPU I/O LOGIC,BUSY- DONE RESET.
05                ;
06                ;FDD - CONDITIONS: POWER ON
07
08 11776 054171 GR00A:  STA     3,GRRET
09
10 11777 000401 A0000:  JMP     .+1       ;DUMMY INSTR. FOR DEBUG-USE
11 12000 006112         SETP2
12 12001 063500         SKPBZ   0
13 12002 006114         EHALT             ;THE SELB-LINE IS GROUNDED
14 12003 006113         LOOP
15
16 12004 000401 A0010:  JMP     .+1       ;DUMMY INSTR. FOR DEBUG-USE
17 12005 006112         SETP2
18 12006 063700         SKPDZ   0
19 12007 006114         EHALT             ;THE SELD-LINE IS GROUNDED
20 12010 006113         LOOP
21
22 12011 000401 A0020:  JMP     .+1       ;DUMMY INSTR. FOR DEBUG-USE
23 12012 006112         SETP2
24 12013 102000         ADC     0,0       ;AC0:= 177777
25 12014 060400         DIA     0,0
26 12015 100015         COM#    0,0,SNR
27 12016 006114         EHALT             ;DIA DIDN'T READ ANYTHING (CPU-ERROR)
28 12017 006113         LOOP
29
30 12020 000401 A0030:  JMP     .+1       ;DUMMY INSTR. FOR DEBUG-USE
31 12021 006112         SETP2
32 12022 102440         SUBO    0,0       ;AC0:= 0
33 12023 060400         DIA     0,0
34 12024 101004         MOV     0,0,SZR
35 12025 006114         EHALT             ;ONE OR MORE DATALINES ARE GROUNDED -
36 12026 006113         LOOP              ;A ONE IN AC0 SHOWS WHICH ONE(S)
37
38 12027 000401 A0040:  JMP     .+1       ;DUMMY INSTR. FOR DEBUG-USE
39 12030 006112         SETP2
40 12031 102000         ADC     0,0       ;AC0:= 177777
41 12032 061400         DIB     0,0
42 12033 100015         COM#    0,0,SNR
43 12034 006114         EHALT             ;DIB DIDN'T READ ANYTHING (CPU-ERROR)
44 12035 006113         LOOP
45
46 12036 000401 A0050:  JMP     .+1       ;DUMMY INSTR. FOR DEBUG-USE
47 12037 006112         SETP2
48 12040 102000         ADC     0,0       ;AC0:= 177777
49 12041 062400         DIC     0,0
50 12042 100015         COM#    0,0,SNR
51 12043 006114         EHALT             ;DIC DIDN'T READ ANYTHING (CPU-ERROR)
52 12044 006113         LOOP
53
54 12045 000401 A0060:  JMP     .+1       ;DUMMY INSTR. FOR DEBUG-USE
55 12046 006112         SETP2
56 12047 060361         NIOP    XFDD
57 12050 063761         SKPDZ   XFDD
58 12051 006114         EHALT             ;I/O-PULSE FAILED TO CLEAR FDC-DONE
59 12052 006113         LOOP              ;CHECK "RESET BUFFER"-SIGNAL
```

```
| 0133  ,MAIN
01
02 12053 000401 A0070:  JMP     .+1     ;DUMMY INSTR. FOR DEBUG-USE
03 12054 006112         SETP2
04 12055 060261         NIOC    XFDD
05 12056 063561         SKPBZ   XFDD
06 12057 006114         EHALT           ;CLEAR-PULSE FAILED TO CLEAR FDC-BUSY
07 12060 006113         LOOP
08
09 12061 000401 A0080:  JMP     .+1     ;DUMMY INSTR. FOR DEBUG-USE
10 12062 006112         SETP2
11 12063 062677         IORST
12 12064 063561         SKPBZ   XFDD
13 12065 006114         EHALT           ;IOPST FAILED TO CLEAR FDC-BUSY
14 12066 006113         LOOP
15
16 12067 000401 A0090:  JMP     .+1     ;DUMMY INSTR. FOR DEBUG-USE
17 12070 006112         SETP2
18 12071 060261         NIOC    XFDD
19 12072 063761         SKPDZ   XFDD
20 12073 006114         EHALT           ;CLEAR-PULSE FAILED TO CLEAR FDC-DONE
21 12074 006113         LOOP
22
23 12075 000401 A0100:  JMP     .+1     ;DUMMY INSTR. FOR DEBUG-USE
24 12076 006112         SETP2
25 12077 062677         IORST
26 12100 063761         SKPDZ   XFDD
27 12101 006114         EHALT           ;IORST FAILED TO CLEAR FDC-DONE
28 12102 006113         LOOP
29
30 12103 000401 A0110:  JMP     .+1     ;DUMMY INSTR. FOR DEBUG-USE
31 12104 006112         SETP2
32 12105 062677         IORST
33 12106 063461         SKPBN   XFDD
34 12107 000402         JMP     .+2
35 12110 006114         EHALT           ;CPU-SKIP-ERROR
36 12111 006113         LOOP
37
38 12112 000401 A0120:  JMP     .+1     ;DUMMY INSTR. FOR DEBUG-USE
39 12113 006112         SETP2
40 12114 062677         IORST
41 12115 063661         SKPDN   XFDD
42 12116 000402         JMP     .+2
43 12117 006114         EHALT           ;CPU-SKIP-ERROR
44 12120 006113         LOOP
45
46 12121 000401 A0130:  JMP     .+1     ;DUMMY INSTR. FOR DEBUG-USE
47 12122 006112         SETP2
48 12123 062677         IORST
49 12124 060177         INTEN
50 12125 000401         JMP     .+1
51 12126 063477         SKPBN   CPU
52 12127 006114         EHALT           ;INTERRUPT ON (CPU BUSY) SHOULD BE
53 12130 060277         INTDS           ;SET AFTER INTEN-INSTR,
54 12131 006113         LOOP
55
56 12132 000401 A0140:  JMP     .+1     ;DUMMY INSTR. FOR DEBUG-USE
57 12133 006112         SETP2
58 12134 062677         IORST
59 12135 060177         INTEN
60 12136 000401         JMP     .+1
61 12137 060277         INTDS
62 12140 063477         SKPBN   CPU
63 12141 000402         JMP     .+2
64 12142 006114         EHALT           ;INTERRUPT ON (CPU BUSY) SHOULD BE
65 12143 006113         LOOP            ;RESET AFTER INTDS-INSTR.
```

```
| 0134  .MAIN
01
02 12144 000401 A0150:  JMP     .+1       ;DUMMY INSTR. FOR DEBUG-USE
03 12145 006112         SETP2
04 12146 062677         IORST
05 12147 102000         ADC     0,0       ;AC0:= -1,IMMEDIATLY RETURN IF INTR.
06 12150 105000         MOV     0,1       ;AC1:= 177777, INTERRUPT MASK
07
08 12151 006133         SETIN             ;SET INTERRUPT ADDRESS, FLAG AND
09                                        ;MASK OUT
10 12152 060177         INTEN             ;ENABLE INTERRUPT
11 12153 006061         CWAIT             ;WAIT 10 MS
12 12154 000200         FD010
13
14 12155 063477         SKPBN   CPU       ;LOOK IF INTERRUPT ON STILL IS SET
15 12156 000402         JMP     .+2       ;IT ISN'T
16 12157 000403         JMP     A0151     ;IT IS
17 12160 061477         INTA    0
18 12161 006114         EHALT             ;AN INTERRUPT OCCURED EVEN IF ALL DE-
19                                        ;VICES ARE MASKED OUT. IF AC0 = 0 THEN
20                                        ;NO DEVICE-CODE WAS READ BY INTA-
21                                        ;INSTR./CHECK INTERRUPT-REQUEST-LINE.
22                                        ;IF AC0=NUMBER, THEN NUMBER MAY BE THE
23 12162 060277 A0151:  INTDS             ;CODE OF THE ILLEGAL INTERRUPT-DEVICE.
24 12163 006113         LOOP
25
26
27 12164 002171         JMP     @GRRET    ;RETURN FROM THIS LOOP-GROUP
```

```
[ 0135  .MAIN
01
02                ;LOOP: A0160 - A0210
03                ;
04                ;FIRST TEST OF STATUSREG.(BIT 1&5),DEVICE SELECT LOGIC, DATA
05                ;FIFO-BUFFER.
06                ;
07                ;FDD-CONDITIONS: POWER ON, DEVICE EMPTY, WRITE PROTECT ON
08
09 12165 054171 GR00B:  STA     3,GRRET
10 12166 000403         JMP     A0160   ;JUMP OVER CONSTANTS
11
12 12167 000000 A17C0:  0               ;COUNTER
13 12170 000125 XBYTE:  125             ;TEST-BYTE FOR FIFO BUFFER
14
15 12171 000401 A0160:  JMP     .+1     ;DUMMY INSTR. FOR DEBUG-USE
16 12172 006112         SETP2
17 12173 060461         DIA     0,XFDD  ;GET STATUS
18 12174 101015         MOV#    0,0,SNR
19 12175 006114         EHALT           ;RECEIVED STATUS = 0.("LOCAL" AND
20                                      ;"WRITE PRO" SHOULD BE SET). CHECK
21 12176 006113         LOOP            ;SELECT LOGIC.
22
23 12177 002171         JMP     @GRRET  ;RETURN FROM THIS LOOP-GROUP
24
25 12200 054171 GR00C:  STA     3,GRRET
26
27 12201 000401 A0170:  JMP     .+1     ;DUMMY INSTR. FOR DEBUG USE
28 12202 006112         SETP2
29 12203 062677         IORST
30 12204 024206         LDA     1,FD064 ;AC1:= 64.
31 12205 044762         STA     1,A17C0 ;SET COUNTER
32 12206 020222         LDA     0,F0377 ;AC0:= 377
33 12207 062061 A0171:  DOB     0,XFDD  ;FILL FIFO-BUFFER WITH CHARS (377)
34 12210 062061         DOB     0,XFDD
35 12211 014756         DSZ     A17C0
36 12212 000775         JMP     A0171   ;FILL ON
37 12213 061461         DIB     0,XFDD
38 12214 101005         MOV     0,0,SNR
39 12215 006114         EHALT           ;FIFO-BUFFER WASN'T CLEARED AFTER
40 12216 062061         DOB     0,XFDD  ;POWER-UP OR AFTER IORST
41 12217 060361         NIOP    XFDD    ;CLEAR FIFO BUFFER WITH I/O-PULSE
42 12220 024206         LDA     1,FD064 ;AC1:= 64.
43 12221 044746         STA     1,A17C0 ;SET COUNTER
44 12222 102440         SUBO    0,0     ;AC0:= 0
45 12223 062061 A0172:  DOB     0,XFDD
46 12224 062061         DOB     0,XFDD  ;FILL FIFO-BUFFER WITH CHARS (0)
47 12225 014742         DSZ     A17C0
48 12226 000775         JMP     A0172   ;FILL ON
49 12227 061461         DIB     0,XFDD  ;GET FIRST WRITTEN BYTE BACK
50 12230 101004         MOV     0,0,SZR
51 12231 000406         JMP     A0173
52 12232 061461         DIB     0,XFDD
53 12233 101004         MOV     0,0,SZR
54 12234 000403         JMP     A0173
55 12235 061461         DIB     0,XFDD
56 12236 101004         MOV     0,0,SZR
57 12237 006114 A0173:  EHALT           ;IT'S IMPOSSIBLE TO CLEAR THE
58 12240 006113         LOOP            ;DATA FIFO-BUFFER WITH AN I/O-PULSE
```

```
I 0136  .MAIN
01
02 12241 000401 A0180:  JMP     .+1         ;DUMMY INSTR. FOR DEBUG-USE
03 12242 006112         SETP2
04 12243 060361         NIOP    XFDD        ;CLEAR DATA BUFFER
05 12244 060261         NIOC    XFDD        ;CLEAR COMMAND BUFFER
06 12245 020723         LDA     0,XBYTE     ;AC0:= 125
07 12246 062061         DOB     0,XFDD      ;WRITE BYTE INTO FIFO
08 12247 006061         CWAIT               ;WAIT 1 MS, TO LET BYTE APPEAR ON
09 12250 000172         FD001               ;FIFO'S OUTPUT.
10 12251 061461         DIB     0,XFDD      ;GET BYTE AGAIN FROM FIFO
11 12252 024716         LDA     1,XBYTE
12 12253 122414         SUB#    1,0,SZR     ;SKIP IF RECEIVED BYTE = TRANSMITTED
13 12254 006114         EHALT               ;ERROR IN FIFO-BUFFER OR DATA PATH.
14                                          ;AC1= BYTE TRANSMITTED, AC0= BYTE RE-
15 12255 006113         LOOP                ;CEIVED.
16
17 12256 000401 A0190:  JMP     .+1         ;DUMMY INSTR. FOR DEBUG-USE
18 12257 006112         SETP2
19 12260 020176         LDA     0,FD005     ;AC0:= 5
20 12261 040253         STA     0,WORK1
21 12262 060361 A0191:  NIOP    XFDD        ;CLEAR DATA BUFFER
22 12263 060261         NIOC    XFDD        ;CLEAR COMMAND REG.
23 12264 010253         ISZ     WORK1       ;INCREMENT DEVICECODE
24 12265 024253         LDA     1,WORK1     ;AC1:= NUMBER FROM 6 TO 77
25 12266 030247         LDA     2,DEVC1     ;AC2:= 61
26 12267 132415         SUB#    1,2,SNR
27 12270 000772         JMP     A0191       ;IF # = 61 THEN GET NEXT
28 12271 030250         LDA     2,DEVC2     ;AC2:= 64
29 12272 132415         SUB#    1,2,SNR
30 12273 000767         JMP     A0191       ;IF # = 64 THEN GET NEXT
31 12274 030220         LDA     2,FD077     ;AC2:= 77
32 12275 132033         ADCZ#   1,2,SNC     ;SKIP IF C(WORK1) < 77
33 12276 000420         JMP     A0192       ;EXIT
34 12277 020671         LDA     0,XBYTE     ;AC0:= 125
35 12300 062061         DOB     0,XFDD      ;WRITE BYTE INTO FDD-DATA-BUFFER
36 12301 020407         LDA     0,DIINS     ;AC0:= DATA-IN INSTRUCTION
37 12302 150000         COM     2,2         ;AC2:= 177700
38 12303 143400         AND     2,0         ;AC0:= DATA-IN INSTR. WITH OUT-
39                                          ;MASKED DEVICE BITS
40 12304 123000         ADD     1,0         ;AC0:= DATA-IN INSTR. WITH NEW
41                                          ;DEVICE CODE
42 12305 040403         STA     0,DIINS     ;SET DIB INSTR. WITH DEVICECODE=C(WORK)
43 12306 006061         CWAIT               ;WAIT 1 MS FOR FIFO-BUFFER
44 12307 000172         FD001
45 12310 061400 DIINS:  DIB     0,2         ;DATA-IN INSTRUCTION (DEVICE CODE IS
46                                          ;CHANGED FROM 6 TO 76 DURING THE
47                                          ;LOOPS, BUT CURRENT FDD-DEVICE-CODE
48                                          ;IS SKIPED).
49 12311 030253         LDA     2,WORK1
50 12312 024656         LDA     1,XBYTE
51 12313 106415         SUB#    0,1,SNR     ;SKIP IF RECEIVED BYTE <> TRANSMITTED
52 12314 006114         EHALT               ;RECEIVED BYTE FROM DEVICE <> FDD =
53                                          ;TRANSMITTED BYTE TO FDD. IT MEANS
54                                          ;THAT THE FDC REACTS ON A DEVICE-CODE
55                                          ;DIFFERENT FROM THE CORRECT ONE. AC2=
56                                          ;ERRONEOUS DEVICE CODE.
57 12315 000745         JMP     A0191       ;REPEAT AND TRY NEXT NUMBER AS DEV.CODE
58 12316 006113 A0192:  LOOP                ;ALL NUMBERS FROM 6 TO 77 ARE TRIED.
59
60 12317 002171         JMP     @GRRET      ;RETURN FROM THIS LOOP-GROUP
```

```
| 0137  .MAIN
01
02 12320 054171 GR00D:  STA     3,GRRET
03
04 12321 000401 A0200:  JMP     .+1         ;DUMMY INSTR. FOR DEBUG-USE
05 12322 006112         SETP2
06 12323 006115         STATA               ;GET STATUS AND CHECK THAT "LOCAL"-BIT
07 12324 040000         040000              ;IS SET.
08 12325 006114         EHALT               ;WRONG STATUS RECEIVED - "LOCAL" ISN'T
09                                          ;SET. AC0= RECEIVED STATUS, AC1= EXPEC-
10 12326 006113         LOOP                ;TED STATUS.
11
12 12327 000401 A0210:  JMP     .+1         ;DUMMY INSTR. FOR DEBUG-USE
13 12330 006112         SETP2
14 12331 006115         STATA               ;GET STATUS AND CHECK THAT "WRITE PRO"-
15 12332 002000         002000              ;BIT IS SET.
16 12333 006114         EHALT               ;WRONG STATUS RECEIVED - "WRITE PRO"
17                                          ;ISN'T SET. AC0= RECEIVED STATUS, AC1=
18 12334 006113         LOOP                ;EXPECTED STATUS.
19
20 12335 002171         JMP     @GRRET      ;RETURN FROM THIS LOOP-GROUP
```

```
| 0138  .MAIN
01
02                ;LOOP: A0220 - A0270
03                ;
04                ;TEST OF BUSY/DONE - SET AND RESET. TEST OF INT-REQ AND
05                ;INT-DIS FLIP-FLOPS.
06                ;
07                ;FDD-CONDITIONS: POWER ON, DEVICE EMPTY OR LOADED WITH
08                ;TM-DISCETTE OR SCRATCHDISCETTE, WRITE PROTECT ON OR OFF.
09
10 12336 054171 GR001:  STA     3,GRRET
11 12337 062677         IORST
12
13 12340 000401 A0220:  JMP     .+1       ;DUMMY INSTR. FOR DEBUG-USE
14 12341 006061         CWAIT             ;WAIT 3 SEC
15 12342 000214         FD3K0
16 12343 006112         SETP2
17 12344 020224         LDA     0,FO1K0 ;AC0:= 1000 = RECALIBRATE COMMAND
18 12345 061161         DOAS    0,XFDD
19 12346 006063         TIMSK             ;WAIT FOR DONE = 1 (MAX 2 SEC)
20 12347 003720         2000.
21 12350 063661         SKPDN   XFDD
22 12351 006114         EHALT             ;DONE ISN'T SET WITHIN 2 SEC AFTER EXE-
23                                        ;CUTION OF A RECALIBRATE-COMMAND. IT
24                                        ;MAY AT THIS TIME BE BUSY-FF WHICH WAS-
25                                        ;N'T SET BY THE START-PULSE, OR IT MAY
26                                        ;BE CAUSED BY A FAULTY ADJUSTMENT OF
27                                        ;THE TRACK ZERO SWITCH.
28 12352 006116         STATN             ;GET STATUS, AND LOOK IF DEVICE IS
29 12353 040000         040000            ;EMPTY
30 12354 000412         JMP     A0221   ;DEVICE EMPTY - SKIP LAST PART OF LOOP
31 12355 020225         LDA     0,FO1K4 ;AC0:= 1400
32 12356 101400         INC     0,0       ;AC0:= 1401 = SEARCH TR. 1 COMMAND
33 12357 061161         DOAS    0,XFDD
34 12360 063461         SKPBN   XFDD
35 12361 006114         EHALT             ;IT'S IMPOSSIBLE TO SET BUSY-FF
36                                        ;WITH A START PULSE.
37 12362 006063         TIMSK             ;WAIT FOR BUSY = 0 (MAX 2 SEC)
38 12363 003720         2000.
39 12364 063561         SKPBZ   XFDD
40 12365 006114         EHALT             ;BUSY DIDN'T RETURN TO 0 WITH-
41 12366 006113 A0221:  LOOP              ;IN 2 SEC.
42
43 12367 000401 A0230:  JMP     .+1       ;DUMMY INSTR. FOR DEBUG-USE
44 12370 006111         SETP1
45 12371 006145         SETDO             ;SET DONE BY USE OF A RECALIBRATE
46                                        ;COMMAND
47 12372 006114         EHALT             ;DONE ISN'T SET AFTER RECALIB.
48 12373 062677         IORST
49 12374 063761         SKPDZ   XFDD
50 12375 006114         EHALT             ;IT'S IMPOSSIBLE TO CLEAR DONE BY
51 12376 006113         LOOP              ;USE OF I/O-RESET
52
53 12377 000401 A0240:  JMP     .+1       ;DUMMY INSTR. FOR DEBUG-USE
54 12400 006111         SETP1
55 12401 006145         SETDO             ;SET DONE BY USE OF A RECALIBRATE
56                                        ;COMMAND
57 12402 006114         EHALT             ;DONE ISN'T SET AFTER RECALIB.
58 12403 060261         NIOC    XFDD
59 12404 063761         SKPDZ   XFDD
60 12405 006114         EHALT             ;IT'S IMPOSSIBLE TO CLEAR DONE BY
61 12406 006113         LOOP              ;USE OF CLEAR PULSE
```

```
| 0139  .MAIN
01
02 12407 000401 A0250:  JMP     .+1         ;DUMMY INSTR. FOR DEBUG-USE
03 12410 006111         SETP1
04 12411 006145         SETDO               ;SET DONE BY USE OF A RECALIB. COMM.
05 12412 006114         EHALT               ;DONE ISN'T SET AFTER RECALIB.
06 12413 060361         NIOP    XFDD
07 12414 063761         SKPDZ   XFDD
08 12415 006114         EHALT               ;IT'S IMPOSSIBLE TO CLEAR DONE BY
09 12416 006113         LOOP                ;USE OF NIOP-PULSE.
10
11 12417 000401 A0260:  JMP     .+1         ;DUMMY INSTR. FOR DEBUG-USE
12 12420 006112         SETP2
13 12421 062677         IORST
14 12422 102000         ADC     0,0         ;AC0:= -1 = INTERRUPT FLAG.MEANS
15                                          ;RETURN IMMEDIATLY AFTER INTERR.
16 12423 024220         LDA     1,F0077     ;AC1:= 000077 = INTERRUPT MASK
17 12424 006133         SETIN               ;SET INTERRUPT ADDRESS,FLAG AND
18                                          ;MASK OUT.
19 12425 060177         INTEN               ;ENABLE INTERRUPTS
20 12426 006145         SETDO               ;SET DONE AND INT-REQ BY USE
21                                          ;OF A RECALIBRATE COMMAND.
22 12427 006114         EHALT               ;DONE ISN'T SET AFTER RECALIB.
23 12430 020237         LDA     0,INTOC     ;AC0:= INTERRUPT OCCOURED FLAG = -1
24                                          ;IF INTERRUPT OCCOURED,ELSE 0.
25 12431 101005         MOV     0,0,SNR     ;SKIP IF INTERRUPT OCCOURED
26 12432 006114         EHALT               ;NO INTERRUPT OCCOURED FROM FDD, EVEN
27                                          ;IF DONE IS SET AND INT-DIS IS RESET.
28 12433 024124         LDA     1,DEVICE    ;AC1:= CURRENT FDD-CODE
29 12434 061477         INTA    0
30 12435 106414         SUB#    0,1,SZR     ;SKIP IF INTERRUPT-CODE = FDD
31 12436 006114         EHALT               ;INTA-INSTR. DIDN'T READ THE FDD-
32                                          ;CODE, EVEN IF INT-REQ FF IS SET.
33                                          ;AC0= DEVICE CODE READ.
34 12437 060261         NIOC    XFDD        ;CLEAR INTERRUPT REQUEST FF
35 12440 102000         ADC     0,0         ;AC0:= -1 = INTERRUPT FLAG
36 12441 024220         LDA     1,F0077     ;AC1:= 000077 = INTERRUPT MASK
37 12442 006133         SETIN               ;SET UP INTERRUPT SERVICE ROUT.
38 12443 060177         INTEN               ;ENABLE INTERRUPT
39 12444 006061         CWAIT               ;WAIT 10 MS
40 12445 000200         FD010
41 12446 020237         LDA     0,INTOC     ;AC0= INTERRUPT OCCOURED FLAG
42 12447 101004         MOV     0,0,SZR     ;SKIP IF NO INTR. OCCOURED
43 12450 006114         EHALT               ;AN INTERRUPT OCCOURED AGAIN
44                                          ;EVEN IF INT-REQ FF WAS CLEARED
45 12451 060277         INTDS               ;BY CLEAR-PULSE.
46 12452 006113         LOOP
47
48 12453 000401 A0270:  JMP     .+1         ;DUMMY INSTR. FOR DEBUG-USE
49 12454 006111         SETP1
50 12455 062677         IORST
51 12456 102440         SUBO    0,0         ;AC0:= 0 = INTERRUPT FLAG
52 12457 024223         LDA     1,F0400     ;AC1:= 400 = BIT7
53 12460 030220         LDA     2,F0077     ;AC2:= 000077
54 12461 147000         ADD     2,1         ;AC1:= INTERRUPT MASK
55 12462 006133         SETIN               ;SET UP INTR. SERVICE ROUT.
56                                          ;MASK OUT FDD (I.E. SET INT-DIS FF).
57 12463 060177         INTEN               ;ENABLE INTERRUPT
58 12464 006145         SETDO               ;SET FDD DONE BY USE OF A RECAL. COMM.
59 12465 006114         EHALT               ;DONE ISN'T SET AFTER RECALIB.
60 12466 020237         LDA     0,INTOC     ;AC0:= INTERRUPT OCCOURED FLAG
61 12467 101004         MOV     0,0,SZR     ;SKIP IF NO INTR. OCCOURED
62 12470 006114         EHALT               ;AN INTERRUPT OCCOURED FROM FDD
63 12471 060277         INTDS               ;EVEN IF INT-DIS FF SHOULD BE SET.
64 12472 006113         LOOP
65 12473 002171         JMP     @GRRET      ;RETURN FROM THIS LOOP-GROUP
```

```
! 0140  .MAIN
01
02                ;LOOP: A0280 - A0320
03                ;
04                ;FIRST TEST OF ONLINE - STATUS, TEST OF RECALIBRATE-COMMAND.
05                ;TEST OF TIME-OUT STATUS BY SEARCHING SECTOR 0 (WHICH DOES-
06                ;N'T EXIST). FIRST TEST OF TRACK-COUNT REG., AND STATUS REG.-
07                ;AUTO RESET, BY SEARCHING TRACK 76 AND 0, FOLLOWED BY READ-
08                ;ING SECTOR 0 AND 2 TIMES DIA-INSTR.
09                ;
10                ;FDD-CONDITIONS: POWER ON, DEVICE LOADED WITH TM-DISCETTE OR
11                ;SCRATCHDISCETTE, WRITE PROTECT ON OR OFF.
12
13 12474 001130 MS600:  600.
14
15 12475 054171 GR002:  STA     3,GRRET
16 12476 000401 A0282:  JMP     .+1         ;DUMMY INSTR. FOR DEBUG-USE
17 12477 006110         SETP0
18 12500 006065         TIMRO               ;TEST STATUS AND WAIT UNTIL DRIVE
19 12501 000214         FD3K0               ;GOES ON-LINE (MAX 3 SEC).
20 12502 006116         STATN
21 12503 040000         040000
22 12504 006114         EHALT               ;DEVICE DIDN'T GO ON-LINE WITHIN
23 12505 006113         LOOP                ;3 SEC.
24
25 12506 000401 A0290:  JMP     .+1         ;DUMMY INSTR. FOR DEBUG-USE
26 12507 006112         SETP2
27 12510 006116         STATN               ;CHECK STATUS
28 12511 175777         175777
29 12512 006114         EHALT               ;STATUS ERROR-OTHER THAN BIT 5 IS SET.
30 12513 006113         LOOP                ;AC0=RECEIVED STATUS,AC1="NON"-EXPECTED.
31
32 12514 000401 B0290:  JMP     .+1         ;DUMMY INSTR. FOR DEBUG-USE
33 12515 006112         SETP2
34 12516 062677         IORST
35 12517 062461         DIC     0,XFDD      ;AC0:= TRACK COUNTER
36 12520 101300         MOVS    0,0
37 12521 024222         LDA     1,F0377     ;AC1:= 377
38 12522 123400         AND     1,0
39 12523 101014         MCV#    0,0,SZR
40 12524 006114         EHALT               ;IT'S IMPOSSIBLE TO CLEAR TRACK
41                                          ;COUNTER BY USE OF I/O-RESET. AC0=
42 12525 006113         LOOP                ;RECEIVED STATE OF COUNTER
43
44 12526 000401 A0300:  JMP     .+1         ;DUMMY INSTR. FOR DEBUG-USE
45 12527 006111         SETP1
46 12530 062677         IORST
47 12531 006063         TIMSK               ;WAIT FOR BUSY = 0 (MAX 2 SEC)
48 12532 003720         2000.
49 12533 063561         SKPBZ   XFDD
50 12534 006114         EHALT               ;BUSY DOESN'T RETURN TO ZERO
51 12535 102440         SUBO    0,0         ;AC0:= 0 = INTERRUPT FLAG
52 12536 024220         LDA     1,F0077     ;AC1:= 000077 = INTERRUPT MASK
53 12537 006133         SETIN               ;SET INTERRUPT ADDRESS,FLAG, MASK OUT.
54 12540 060177         INTEN               ;ENABLE INTERRUPTS
55 12541 020224         LDA     0,F01K0     ;AC0:= 1000 = RECAL.-COMMAND
56 12542 061161         DOAS    0,XFDD
57 12543 006063         TIMSK               ;WAIT FOR DONE = 1 (MAX 2 SEC)
58 12544 003720         2000.
59 12545 063661         SKPDN   XFDD
60 12546 006114         EHALT               ;DONE ISN'T SET AFTER RECALIB.
```

```
! 0141  .MAIN
01
02 12547 020237          LDA     0,INTOC ;AC0:= INTERRUPT OCCOURED FLAG
03 12550 101005          MOV     0,0,SNR ;SKIP IF INTERRUPT OCCOURED
04 12551 006114          EHALT           ;NO INTERRUPT OCCOURED WHEN DONE
05                                       ;IS SET AFTER A RECALIB.
06 12552 006116          STATN           ;CHECK STATUS
07 12553 175777          175777
08 12554 006114          EHALT           ;STATUS ERROR AFTER RECALIB. OTHER
09                                       ;THAN BIT 5 (WRITE PRO) IS SET.
10 12555 060277          INTDS           ;AC0=RECEIVED STATUS,AC1="NON"-EXPECTED
11 12556 006113          LOOP
12
13 12557 000401 A0310:   JMP     .+1     ;DUMMY INSTR. FOR DEBUG-USE
14 12560 006111          SETP1
15 12561 006136          RECAL           ;RECALIBRATE DISC-DRIVE
16 12562 102440          SUBC    0,0     ;AC0:= 0 = READ SECTOR 0 COMM.
17 12563 061061          DOA     0,XFDD
18 12564 060161          NIOS    XFDD    ;READ SECTOR 0 (SECTOR 0 DOESN'T EXIST)
19 12565 006063          TIMSK           ;WAIT FOR DONE = 1 (MAX 1 SEC). TIME-
20 12566 001750          1000.           ;OUT MUST APPEAR WITHIN 4 REVOLUTIONS
21 12567 063661          SKPDN   XFDD    ;(700 MS)
22 12570 006114          EHALT           ;DONE WASN'T SET WITHIN 1 SEC AFTER
23                                       ;START.
24 12571 006115          STATA           ;GET STATUS
25 12572 000002          000002
26 12573 006114          EHALT           ;"TIME OUT" (BIT 14) WASN'T SET. AC0=
27                                       ;RECEIVED STATUS,AC1= EXPECTED.
28 12574 030226          LDA     2,FO2K0 ;AC2:= 2000
29 12575 147000          ADD     2,1
30 12576 130000          COM     1,2     ;AC2:= 175775
31 12577 143414          AND#    2,0,SZR
32 12600 006114          EHALT           ;OTHERS THAN BIT 14 OR 5 ARE SET. AC0=
33                                       ;RECEIVED STATUS,AC1= EXPECTED
34 12601 006116          STATN           ;GET STATUS AGAIN,CHECK AUTO-CLEAR
35 12602 000002          000002
36 12603 006114          EHALT           ;"TIME OUT" IS STILL SET AFTER 2*DIA
37                                       ;CHECK AUTO RESET OF STATUS-REG. AC0=
38 12604 006113          LOOP            ;RECEIVED STATUS,AC1="NON"-EXPECTED
39
40 12605 000401 A0320:   JMP     .+1     ;DUMMY INSTR. FOR DEBUG-USE
41 12606 006111          SETP1
42 12607 006136          RECAL           ;RECALIBRATE DISC-DRIVE
43 12610 006063          TIMSK           ;WAIT FOR BUSY = 0 (MAX 2 SEC)
44 12611 003720          2000.
45 12612 063561          SKPBZ   XFDD
46 12613 006114          EHALT           ;BUSY DIDN'T RETURN TO 0 WITHIN 2 SEC
47 12614 006116          STATN           ;CHECK STATUS
48 12615 175777          175777
49 12616 006114          EHALT           ;STATUS ERROR.AC0= RECEIVED STATUS
50                                       ;AC1= "NON"-EXPECTED STATUS.
51 12617 024236          LDA     1,MXTRA ;AC1:= 76.
52 12620 030225          LDA     2,FO1K4 ;AC2:= 1400 = TRACK SEARCH COMMAND
53 12621 133000          ADD     1,2
54 12622 071161          DOAS    2,XFDD
55 12623 063461          SKPBN   XFDD
56 12624 006114          EHALT           ;IT'S IMPOSSIBLE TO SET FDD-BUSY WITH
57                                       ;A TRACK-SEARCH COMMAND.
```

```
I 0142  .MAIN
01
02 12625 006061          CWAIT             ;WAIT 600 MS
03 12626 012474          MS600
04 12627 063461          SKPBN   XFDD
05 12630 006114          EHALT             ;TOO FAST HEAD-MOVING. THE HEAD WAS
06                                         ;MOVED FROM TRACK 0 TO 76. IN A
07                                         ;TIME < 600 MS (SHOULD BE 770 MS).
08 12631 006063          TIMSK             ;WAIT FOR DONE = 1 (MAX 800 MSEC)
09 12632 001440          800.
10 12633 063661          SKPDN   XFDD
11 12634 006114          EHALT             ;TOO SLOW HEAD-MOVING. THE HEAD WAS
12                                         ;MOVED FROM TRACK 0 TO 76. IN A
13                                         ;TIME > 1.4 SEC. (SHOULD BE 770 MS)
14
15 12635 006116          STATN             ;CHECK STATUS
16 12636 175777          175777
17 12637 006114          EHALT             ;STATUS ERROR.AC0= RECEIVED STATUS,
18                                         ;AC1= "NON"-EXPECTED STATUS.
19 12640 102440          SUBC    0,0       ;AC0:= 0
20 12641 061061          DOA     0,XFDD
21 12642 060161          NIOS    XFDD      ;READ SECTOR 0 (SEC. 0 DOESN'T EXIST)
22 12643 006063          TIMSK             ;WAIT FOR DONE = 1 (MAX 1 SEC).TIME-
23 12644 001750          1000.             ;OUT MUST APPEAR WITHIN 4 REVOLU-
24 12645 063661          SKPDN   XFDD      ;TIONS (700 MS).
25 12646 006114          EHALT             ;DONE WASN'T SET WITHIN 1 SEC. AFTER
26                                         ;START.
27 12647 062461          DIC     0,XFDD    ;GET TRACK COUNTER
28 12650 101300          MOVS    0,0
29 12651 024222          LDA     1,FO377   ;AC1:= 377
30 12652 123400          AND     1,0
31 12653 024236          LDA     1,MXTRA   ;AC1:= 76.
32 12654 106414          SUB#    0,1,SZR   ;SKIP IF TRACK COUNTER = 76.
33 12655 006114          EHALT             ;AFTER SEARCH OF TRACK 76., THE TRACK
34                                         ;COUNTER WASN'T = 76. AC0= CURRENT STATE
35                                         ;OF TRACKCOUNTER
36 12656 006115          STATA             ;GET STATUS
37 12657 000002          000002
38 12660 006114          EHALT             ;"TIME OUT" (BIT 14) WASN'T SET.
39                                         ;AC0= RECEIVED STATUS, AC1= EXPECTED
40 12661 006116          STATN             ;GET STATUS AGAIN, CHECK AUTO-CLEAR
41 12662 000002          000002
42 12663 006114          EHALT             ;"TIME OUT" IS STILL SET AFTER 2*DIA.
43                                         ;CHECK AUTO-RESET OF STATUS REGISTER.
44                                         ;AC0= RECEIVED STATUS,AC1= "NON"-EXPEC.
45 12664 006063          TIMSK             ;WAIT FOR BUSY = 0 (MAX 2 SEC)
46 12665 003720          2000.
47 12666 063561          SKPBZ   XFDD
48 12667 006114          EHALT             ;BUSY DIDN'T RETURN TO 0 WITHIN 2 SEC
49 12670 006116          STATN             ;CHECK STATUS
50 12671 175777          175777
51 12672 006114          EHALT             ;STATUS ERROR.AC0= RECEIVED STATUS,
52                                         ;AC1= "NON"-EXPECTED STATUS.
53 12673 126440          SUBC    1,1       ;AC1:= 0
54 12674 030225          LDA     2,FO1K4   ;AC2:= 1400 = TRACK SEARCH COMMAND
55 12675 133000          ADD     1,2
56 12676 071161          DOAS    2,XFDD
57 12677 063461          SKPBN   XFDD
58 12700 006114          EHALT             ;IT'S IMPOSSIBLE TO SET FDD-BUSY WITH
59                                         ;A TRACK-SEARCH COMMAND.
60 12701 006063          TIMSK             ;WAIT FOR DONE = 1 (MAX 2 SEC)
61 12702 003720          2000.
62 12703 063661          SKPDN   XFDD
63 12704 006114          EHALT             ;DONE WASN'T SET WITHIN 2 SEC
```

```
| 0143   .MAIN
01
02 12705 006116          STATN              ;CHECK STATUS
03 12706 175777          175777
04 12707 006114          EHALT              ;STATUS ERROR. AC0= RECEIVED STATUS,
05                                          ;AC1= "NON"-EXPECTED STATUS.
06 12710 102440          SUBO     0,0       ;AC0:= 0
07 12711 061061          DOA      0,XFDD
08 12712 060161          NIOS     XFDD      ;READ SECTOR 0 (SEC. 0 DOESN'T EXIST)
09 12713 006063          TIMSK              ;WAIT FOR DONE = 1 (MAX 1 SEC).TIME-
10 12714 001750          1000.              ;OUT MUST APPEAR WITHIN 4 REVOLUTIONS
11 12715 063661          SKPDN    XFDD      ;(700 MS).
12 12716 006114          EHALT              ;DONE WASN'T SET WITHIN 1 SEC AFTER
13                                          ;START.
14
15 12717 062461          DIC      0,XFDD    ;GET TRACK COUNTER
16 12720 101300          MOVS     0,0
17 12721 024222          LDA      1,F0377   ;AC1:= 377
18 12722 123400          AND      1,0
19 12723 101014          MOV#     0,0,SZR   ;SKIP IF TRACK COUNTER = 0
20 12724 006114          EHALT              ;AFTER TRACKSEARCH OF TRACK 0, THE TRACK
21                                          ;COUNTER WASN'T = 0. AC0= CURRENT
22                                          ;STATE OF TRACKCOUNTER.
23 12725 006115          STATA              ;GET STATUS
24 12726 000002          000002
25 12727 006114          EHALT              ;"TIME OUT" WASN'T SET.
26                                          ;AC0= RECEIVED STATUS,AC1= EXPECTED
27 12730 006116          STATN              ;GET STATUS AGAIN, AND CHECK AUTO-CLEAR
28 12731 000002          000002
29 12732 006114          EHALT              ;"TIME OUT" IS STILL SET AFTER 2*DIA.
30                                          ;CHECK AUTO-RESET OF STATUS REG. AC0=
31 12733 006113          LOOP               ;RECEIVED STATUS,AC1="NON"-EXPECTED.
```

```
| 0144  .MAIN
01
02                ;LOOP A0330
03                ;
04                ;TEST OF THE FIFO-DATABUFFER IN CONTROLLER. 128 RANDOM GENERATED
05                ;BYTES ARE WRITTEN INTO THE FIFO AND A CORE BUFFER. LATER ON
06                ;THE BYTES ARE READ BACK AND COMPARED, CHAR BY CHAR. DURING THE
07                ;LOOPS NO STARTPULSE IS SEND, AND NO DATA IS WRITTEN ON DIS-
08                ;CETTE.
09                ;
10                ;FDD-CONDITIONS: POWER ON, DEVICE LOADED WITH TM-DISCETTE OR
11                ;SCRATCHDISCETTE, WRITE PROTECT ON OR OFF.
12
13 12734 000404           JMP     A0330    ;JUMP OVER CONSTANTS
14
15 12735 013333 LARA1:    13333            ;RANDOM #
16 12736 000077 A33C0:    63.              ;CHARCOUNTER
17 12737 000000 A033R:    0                ;INTERNAL RETURN SAVE
18
19 12740 000401 A0330:    JMP     .+1      ;DUMMY INSTR. FOR DEBUG-USE
20 12741 006112           SETP2
21 12742 060361           NIOP    XFDD     ;CLEAR FIFO BUFFER
22 12743 024206           LDA     1,FD064  ;AC1:= 64.
23 12744 044772           STA     1,A33C0  ;SET CHARCOUNTER
24 12745 030234           LDA     2,AWBUF  ;AC2:= ADDRESS OF WRITE BUFFER
25 12746 020767 A0331:    LDA     0,LARA1  ;AC0:= LAST USED RANDOM #
26 12747 006134           GRAND            ;CALCULATE NEW RANDOM #
27 12750 040765           STA     0,LARA1  ;AC0= NEW RANDOM #
28 12751 105000           MOV     0,1
29 12752 101300           MOVS    0,0
30 12753 034222           LDA     3,F0377  ;AC3:= 377
31 12754 163400           AND     3,0
32 12755 167400           AND     3,1
33 12756 041000           STA     0,0,2    ;STORE RANDOM BYTE IN WRITE BUFFER
34 12757 151400           INC     2,2
35 12760 045000           STA     1,0,2    ;STORE RANDOM BYTE IN WRITE BUFFER
36 12761 151400           INC     2,2
37 12762 062061           DOB     0,XFDD   ;WRITE RANDOM BYTES INTO FIFO
38 12763 066061           DOB     1,XFDD
39 12764 014752           DSZ     A33C0    ;DECREMENT CHAR COUNTER
40 12765 000761           JMP     A0331    ;FILL ON
41 12766 024206           LDA     1,FD064  ;AC1:=64.-WRITE BUFF. AND FIFO ARE FULL
42 12767 044747           STA     1,A33C0
43 12770 030234           LDA     2,AWBUF
44 12771 021000 A0332:    LDA     0,0,2    ;AC0:= BYTE WRITTEN
45 12772 151400           INC     2,2
46 12773 065461           DIB     1,XFDD   ;AC1:= BYTE FROM FIFO
47 12774 106414           SUB#    0,1,SZR  ;SKIP IF BYTE WRITTEN=BYTE RECEIVED
48 12775 004411           JSR     A0333    ;ERROR
49 12776 021000           LDA     0,0,2    ;OK - AC0:= BYTE WRITTEN
50 12777 151400           INC     2,2
51 13000 065461           DIB     1,XFDD   ;AC1:= BYTE FROM FIFO
52 13001 106414           SUB#    0,1,SZR  ;SKIP IF BYTE WRITTEN=BYTE RECEIVED
53 13002 004404           JSR     A0333    ;ERROR
54 13003 014733           DSZ     A33C0    ;OK
55 13004 000765           JMP     A0332    ;TEST ON
56 13005 000410           JMP     A0334    ;ALL BYTES ARE EXAMINED
```

```
I 0145  .MAIN
01
02 13006 054731 A0333:  STA     3,A033R
03 13007 034234         LDA     3,AWBUF
04 13010 172400         SUB     3,2     ;AC2:= BYTE NO.
05 13011 006114         EHALT           ;ERROR IN FIFO - DATA BUFFER, OR DATA-
06                                      ;PATHS. DATABYTE READ <> PREVIOUSLY
07                                      ;WRITTEN BYTE. AC0= BYTE WRITTEN, AC1=
08 13012 034234         LDA     3,AWBUF ;BYTE READ, AC2= BYTE NO. WITH ERROR
09 13013 173000         ADD     3,2     ;(1<=  NO. <= 128)
10 13014 002723         JMP     @A033R
11
12 13015 006113 A0334:  LOOP
13
14 13016 002171         JMP     @GRRET  ;RETURN FROM THIS LOOP-GROUP
```

```
! 0146  .MAIN
01
02                ;LOOP  A0342
03                ;
04                ;FIRST CHECK OF THE READING FROM THE ADDRESS FIELD. RANDOM SE-
05                ;LECTED SECTORS ARE READ FROM TRACK 0. NO DATA-CHECK IS PERFOR-
06                ;MED. THE NO. OF POSITION-ERRORS (TRACKCOUNTER ISN'T = 0 OR
07                ;TRACK NO. IN ADDRESS FIELD ISN'T = 0) AND ADDRESS FIELD ERRORS
08                ;(CRC CHECK BYTES READ AND GENERATED DIDN'T EQUAL) ARE COUN-
09                ;TED DURING 100 READS, AND DIFFERENT ERRORHALTS ARE SELECTED DE-
10                ;PENDING ON ERROR - AMOUNT AND KIND.
11                ;
12                ;FDD-CONDITIONS: POWER ON, DEVICE LOADED WITH TM-DISCETTE, WRITE
13                ;PROTECT ON OR OFF.
14
15
16 13017 054171 GR0@3:  STA     3,GRRET
17 13020 000405         JMP     A@340   ;JUMP OVER CONSTANTS
18
19 13021 000000 A34E1:  0               ;ERROR COUNTER (POSITION ERRORS)
20 13022 000000 A34E2:  0               ;ERROR COUNTER (ADDRESS FIELD ERRORS)
21 13023 000000 A34E3:  0               ;ERROR COUNTER (TIME OUTS)
22 13024 000144 A34C0:  100.            ;READ COUNTER
23
24
25 13025 000401 A@340:  JMP     .+1     ;DUMMY INSTR. FOR DEBUG-USE
26 13026 006110         SETP0
27 13027 102440         SUB0    0,0     ;AC0:= 0
28 13030 040771         STA     0,A34E1 ;CLEAR ERROR COUNTERS
29 13031 040771         STA     0,A34E2
30 13032 040771         STA     0,A34E3
31 13033 020210         LDA     0,FD100 ;AC0:= 100.
32 13034 040770         STA     0,A34C0 ;SET READ COUNTER
33 13035 006116         STATN           ;CHECK STATUS
34 13036 175777         175777
35 13037 006114         EHALT           ;STATUS ERROR - OTHER THAN BIT 5 IS
36                                      ;SET.AC0= RECEIVED STATUS, AC1= EX-
37                                      ;PECTED STATUS.
38 13040 006136         RECAL           ;RECALIBRATE DISC-DRIVE
39 13041 006063         TIMSK           ;WAIT FOR BUSY = 0 (MAX 2 SEC)
40 13042 003720         2000.
41 13043 063561         SKPBZ   XFDD
42 13044 006114         EHALT           ;BUSY DOESN'T RETURN TO ZERO
43 13045 062677 A@341:  IORST           ;CLEAR STATUS REG.
44 13046 006135         RANTS           ;GET A RANDOM SECTOR NO TO AC0
45 13047 126440         SUB0    1,1     ;AC1:= 0 = READ COMMAND
46 13050 123000         ADD     1,0     ;AC0:= READ SECTOR COMMAND
47 13051 040253         STA     0,WORK1
48 13052 061161         DOAS    0,XFDD  ;READ SECTOR
49 13053 006063         TIMSK           ;WAIT FOR BUSY = 0 (MAX 1.5 SEC)
50 13054 002734         1500.
51 13055 063561         SKPBZ   XFDD
52 13056 006114         EHALT           ;BUSY DOESN'T RETURN TO ZERO
53 13057 060461         DIA     0,XFDD  ;GET STATUS
54 13060 024232         LDA     1,F0H14 ;AC1:= 140000
55 13061 107415         AND#    0,1,SNR ;SKIP IF LOCAL OR HARD ERROR
56 13062 000404         JMP     A@342
57 13063 024253         LDA     1,WORK1
58 13064 006114         EHALT           ;THE SECTOR-SEARCH CAUSED
59                                      ;LOCAL OR HARD-ERROR. AC0= RECEIVED
60 13065 000412         JMP     A@343   ;STATUS, AC1= SECTOR SEARCHED.
```

```
| 0147  .MAIN
01
02 13066 024216 A0342:  LDA     1,F0010 ;AC1:= 000010
03 13067 107414         AND#    0,1,SZR ;SKIP IF NOT POSITION ERROR
04 13070 010731         ISZ     A34E1   ;INCREMENT ERROR COUNTER
05 13071 024227         LDA     1,F010K ;AC1:= 010002
06 13072 107414         AND#    0,1,SZR ;SKIP IF NOT ADDRESS FIELD ERROR
07 13073 010727         ISZ     A34E2   ;INCREMENT ERROR COUNTER
08 13074 024173         LDA     1,FD002 ;AC1:= 2
09 13075 107414         AND#    0,1,SZR ;SKIP IF NOT TIME-OUT
10 13076 010725         ISZ     A34E3   ;INCREMENT ERROR COUNTER
11 13077 014725 A0343:  DSZ     A34C0   ;DECREMENT LOOP COUNTER
12 13100 000745         JMP     A0341   ;READ AGAIN
13                                      ;100 SECTORS READ .
14 13101 020720         LDA     0,A34E1 ;AC0:= NO OF POSITION ERRORS
15 13102 024207         LDA     1,FD070 ;AC1:= 70.
16 13103 106432         SUBZ#   0,1,SZC ;SKIP IF # PE > 70.
17 13104 000403         JMP     A0344
18 13105 006114         EHALT           ;MORE THAN 70% POSITION ERRORS
19                                      ;DURING 100 SECTOR-READS IN TRACK 0.
20                                      ;CHECK TRACK SEARCH - AND TRACK COUN-
21 13106 000405         JMP     A0345   ;TER LOGIC. AC0= ERRORPERCENT.
22 13107 024200 A0344:  LDA     1,FD010 ;AC1:= 10.
23 13110 106432         SUBZ#   0,1,SZC ;SKIP IF # PE > 10.
24 13111 000402         JMP     A0345
25 13112 006114         EHALT           ;MORE THAN 10% POSITION ERRORS
26                                      ;DURING 100 SECTOR-READS IN TRACK 0.
27                                      ;CHECK READ-CIRCUIT,AC0= ERRORPERCENT
28 13113 020707 A0345:  LDA     0,A34E2 ;AC0:= NO OF ADDRESS FIELD ERRORS
29 13114 024207         LDA     1,FD070 ;AC1:= 70.
30 13115 106432         SUBZ#   0,1,SZC ;SKIP IF # AE > 70.
31 13116 000403         JMP     A0346
32 13117 006114         EHALT           ;MORE THAN 70% ADDRESS FIELD ERRORS
33                                      ;DURING 100 SECTOR READS IN TRACK 0.
34                                      ;CHECK CRC LOGIC.AC0= ERRORPERCENT
35 13120 000405         JMP     A0347
36 13121 024200 A0346:  LDA     1,FD010 ;AC1:= 10.
37 13122 106432         SUBZ#   0,1,SZC ;SKIP IF # AE > 10.
38 13123 000402         JMP     A0347
39 13124 006114         EHALT           ;MORE THAN 10% ADDRESS FIELD ERRORS
40                                      ;DURING 100 SECTOR-READS IN TRACK 0.
41                                      ;CHECK READ-CIRCUIT AND CRC-LOGIC.
42                                      ;AC0= ERRORPERCENT.
43 13125 020676 A0347:  LDA     0,A34E3 ;AC0:= NO OF TIME OUTS
44 13126 024207         LDA     1,FD070 ;AC1:= 70.
45 13127 106432         SUBZ#   0,1,SZC ;SKIP IF # TO > 70.
46 13130 000403         JMP     A0348
47 13131 006114         EHALT           ;MORE THAN 70% TIME OUTS DURING
48                                      ;100 SECTOR-READS IN TRACK 0. AC0=
49 13132 000405         JMP     A0349   ;ERRORPERCENT.
50 13133 024200 A0348:  LDA     1,FD010 ;AC1:= 10.
51 13134 106432         SUBZ#   0,1,SZC ;SKIP IF # TO > 10.
52 13135 000402         JMP     A0349
53 13136 006114         EHALT           ;MORE THAN 10% TIME OUTS DURING
54                                      ;100 SECTOR-READS IN TRACK 0.AC0=
55 13137 006113 A0349:  LOOP            ;ERRORPERCENT.
```

```
I 0148  ,MAIN
01
02                ;LOOP  A0352 - A0360
03                ;
04                ;FIRST READ OF A SECTOR WITH FOLLOWING DATACHECK. 100 RANDOM-
05                ;LY SELECTED SECTORS IN TRACK 0 ARE READ AND CHECKED. IN CASE
06                ;OF "SOFT"-ERRORS, THE SECTOR WILL BE RE-READ UP TO "TRIES"
07                ;TIMES BEFORE ERROR-HALT.
08                ;
09                ;FDD-CONDITIONS: POWER ON, DEVICE LOADED WITH IM-DISCETTE, WRITE
10                ;PROTECT ON OR OFF.
11
12 13140 000401 A0350:  JMP     .+1       ;DUMMY INSTR. FOR DEBUG-USE
13 13141 006110         SETP0
14 13142 006116         STATN             ;CHECK STATUS
15 13143 175777         175777
16 13144 006114         EHALT             ;STATUS ERROR - OTHER THAN BIT 5 IS
17                                        ;SET.AC0= RECEIVED STATUS,AC1= EXPEC-
18                                        ;TED STATUS.
19 13145 006136         RECAL             ;RECALIBRATE DISC-DRIVE
20 13146 006063         TIMSK             ;WAIT FOR BUSY = 0 (MAX 2 SEC)
21 13147 003720         2000.
22 13150 063561         SKPBZ   XFDD
23 13151 006114         EHALT             ;BUSY DOESN'T RETURN TO ZERO
24 13152 006113         LOOP
25
26 13153 006142         PTFLL             ;FILL WRITE BUFFER WITH PATTERNS
27                                        ;FOR COMPARISON
28
29 13154 000401 A0360:  JMP     .+1       ;DUMMY INSTR. FOR DEBUG-USE
30 13155 006112         SETP2
31 13156 102440         SUBO    0,0       ;AC0:= 0
32 13157 040242         STA     0,TOTCO   ;RESET READ ERROR COUNTERS
33 13160 040241         STA     0,RERCO
34 13161 020256         LDA     0,WTRIES
35 13162 040240         STA     0,REWCO   ;SET REWCO = "WTRIES" TO INHIBIT REWRITE-
36 13163 006135         RANTS             ;TRIES. GET A RANDOM SECTOR NO TO AC0
37 13164 042234         STA     0,@AWBUF  ;STORE SECTOR NO AS 1. BYTE IN BUFFER
38 13165 022234 A0361:  LDA     0,@AWBUF
39 13166 006146         RDWRI             ;READ SECTOR SPEC IN AC0 AND CHECK
40                                        ;STATUS AND DATA.
41 13167 000403         JMP     A0362     ;HARD ERROR,HALT AND GIVE UP
42 13170 000405         JMP     A0363     ;SOFT ERROR,READ AGAIN IF TRIES < "TRIES"
43 13171 000410         JMP     A0364     ;READ OK
44
45 13172 121000 A0362:  MOV     1,0
46 13173 006114         EHALT             ;HARD ERROR AFTER READ. AC0=
47                                        ;RECEIVED STATUS.
48 13174 000405         JMP     A0364     ;GIVE UP
49 13175 006147 A0363:  SOFER             ;SOFT ERROR AFTER READ.READ AGAIN IF
50                                        ;TRIES < "TRIES" ELSE HALT AND RET. +1.
51 13176 000403         JMP     A0364     ;GIVE UP
52 13177 000766         JMP     A0361     ;TRY TO READ AGAIN
53 13200 063077         HALT              ;ERRONEOUS RETURN FROM "SOFER". NOT
54                                        ;USED WHILE "REWCO" = 3.
55 13201 006113 A0364:  LOOP
```

```
| 0149  .MAIN
01
02                ;LOOP  A0370
03                ;
04                ;FIRST READ OF A RANDOM SELECTED SECTOR IN DIFF. TRACKS <> 0.
05                ;THE ADDRESS FIELD INFORMATION AND DATA FIELD INF. ARE CHECKED.
06                ;THE READ-ERRORS ARE COUNTED AND DIFFERENT ERROR-HALTS ARE POS-
07                ;SIBLE, LIKE IN LOOP A0340.
08                ;
09                ;FDD-CONDITIONS: POWER ON, DEVICE LOADED WITH TM-DISCETTE, WRITE
10                ;PROTECT ON OR OFF.
11
12 13202 000424           JMP      A0370    ;JUMP OVER CONSTANTS
13
14 13203 000000 A37E1:    0                 ;ERRORCOUNTER (POSITION ERRORS)
15 13204 000000 A37E2:    0                 ;ERRORCOUNTER (ADDRESS FIELD ERRORS)
16 13205 000000 A37E3:    0                 ;ERRORCOUNTER (PARITY ERRORS)
17 13206 000000 A37E4:    0                 ;ERRORCOUNTER (DATA ERRORS)
18 13207 000000 A37E5:    0                 ;ERRORCOUNTER (DATA SYNC ERRORS)
19 13210 000000 A37E6:    0                 ;ERRORCOUNTER (TIME OUTS)
20 13211 000012 A37C0:    10.               ;READCOUNTER
21 13212 013213 TRA37:    .+1               ;POINTER TO TABLE WITH TRACKNO'S
22 13213 000001           1                 ;WHICH HAVE TO BE READ IN THIS LOOP
23 13214 000002           2
24 13215 000004           4
25 13216 000010           8.
26 13217 000020           16.
27 13220 000040           32.
28 13221 000100           64.
29 13222 000025           21.
30 13223 000052           42.
31 13224 000112           74.
32 13225 000000           0
33
34 13226 000401 A0370:    JMP      .+1      ;DUMMY INSTR. FOR DEBUG-USE
35 13227 006110           SETP0
36 13230 102440           SUBO     0,0      ;AC0:= 0
37 13231 040752           STA      0,A37E1  ;RESET ERROR COUNTERS
38 13232 040752           STA      0,A37E2
39 13233 040752           STA      0,A37E3
40 13234 040752           STA      0,A37E4
41 13235 040752           STA      0,A37E5
42 13236 040752           STA      0,A37E6
43 13237 020200           LDA      0,FD010  ;AC0:= 10.
44 13240 040751           STA      0,A37C0  ;SET READ COUNTER
45 13241 006116           STATN              ;CHECK STATUS
46 13242 175777           175777
47 13243 006114           EHALT              ;STATUS ERROR-OTHER THAN BIT 5 IS SET.
48                                           ;AC0=RECEIVED STATUS,AC1=EXPECTED.
49 13244 006136           RECAL              ;RECALIBRATE DISC-DRIVE
50 13245 006063           TIMSK              ;WAIT FOR BUSY = 0 (MAX 2 SEC)
51 13246 003720           2000.
52 13247 063561           SKPBZ    XFDD
53 13250 006114           EHALT              ;BUSY DOESN'T RETURN TO ZERO
54 13251 006142           PTFLL              ;FILL WRITE BUFFER WITH PATTERNS USED
55                                           ;DURING WRITE OF TESTDISC FOR COMPARISON
56 13252 030740 A0371:    LDA      2,TRA37  ;AC2:= ADDRESS OF TRACK TABLE
57 13253 025000 A0372:    LDA      1,0,2    ;AC1:= TRACK TO READ FROM
58 13254 006137           TRPOS              ;POSITION HEAD TO TRACK SPEC IN AC1
59 13255 125014           MOV#     1,1,SZR  ;SKIP IF LAST TRACK IN TABLE
60 13256 000404           JMP      A0373    ;NOT LAST TRACK
```

```
I 0150  .MAIN
01
02 13257 014732            DSZ     A37C0   ;DECREMENT READ COUNTER
03 13260 000772            JMP     A0371
04 13261 000443            JMP     A0378   ;ALL TRACKS IN TABLE READ 10 TIMES
05 13262 050254 A0373:     STA     2,WORK2
06 13263 006135            RANTS           ;GET A RANDOM SECTOR NO TO AC0
07 13264 042234            STA     0,@AWBUF;STORE SECTOR NO AS 1. BYTE IN BUFFER
08 13265 006146            RDWRI           ;READ SECTOR,CHECK STATUS AND DATA
09 13266 000403            JMP     A0375   ;HARD ERROR, AC1= STATUS
10 13267 000407            JMP     A0376   ;"SOFT" ERROR, AC1= STATUS
11 13270 000431            JMP     A0377   ;READ OK
12 13271 032254 A0375:     LDA     2,@WORK2
13 13272 121000            MOV     1,0
14 13273 026234            LDA     1,@AWBUF
15 13274 006114            EHALT           ;HARD ERROR AFTER READ,AC0= RECEIVED
16 13275 000424            JMP     A0377   ;STATUS,AC1=SECTOR NO,AC2= TRACK NO.
17 13276 125015 A0376:     MOV#    1,1,SNR ;SKIP IF NOT DATA WRONG
18 13277 010707            ISZ     A37E4   ;INCREMENT ERROR COUNTER
19 13300 102000            ADC     0,0     ;AC0:= -1
20 13301 122414            SUB#    1,0,SZR ;SKIP IF DATA SYNC ERROR
21 13302 000403            JMP     .+3
22 13303 010704            ISZ     A37E5   ;INCREMENT ERROR COUNTER
23 13304 000415            JMP     A0377
24 13305 020216            LDA     0,F0010 ;AC0:= 10
25 13306 107414            AND#    0,1,SZR ;SKIP IF NOT POSITION ERROR
26 13307 010674            ISZ     A37E1   ;INCREMENT ERRORCOUNTER
27 13310 020227            LDA     0,F010K ;AC0:= 010000
28 13311 107414            AND#    0,1,SZR ;SKIP IF NOT ADDRESS FIELD ERROR
29 13312 010672            ISZ     A37E2   ;INCREMENT ERRORCOUNTER
30 13313 020217            LDA     0,F0040 ;AC0:= 40
31 13314 107414            AND#    0,1,SZR ;SKIP IF NOT PARITY ERROR
32 13315 010670            ISZ     A37E3   ;INCREMENT ERRORCOUNTER
33 13316 020173            LDA     0,FD002 ;AC0:= 2
34 13317 107414            AND#    0,1,SZR ;SKIP IF NOT TIME OUT
35 13320 010670            ISZ     A37E6   ;INCREMENT ERRORCOUNTER
36 13321 030254 A0377:     LDA     2,WORK2
37 13322 151400            INC     2,2
38 13323 000730            JMP     A0372
39                                         ;100 SECTORS ARE READ FROM 10 DIFF.
40                                         ;TRACKS.
41 13324 020657 A0378:     LDA     0,A37E1 ;AC0:= NO OF POS. ERRORS
42 13325 024200            LDA     1,FD010 ;AC1:= 10.
43 13326 106033            ADCZ#   0,1,SNC ;SKIP IF # PE < 10.
44 13327 006114            EHALT           ;MORE THAN 10% POSITION ERRORS
45                                         ;DUPING 100 SECTOR READS IN DIFF.
46                                         ;TRACKS, AC0= ERRORPERCENT.
47 13330 020654            LDA     0,A37E2
48 13331 024200            LDA     1,FD010 ;AC1:= 10.
49 13332 106033            ADCZ#   0,1,SNC ;SKIP IF # ADE < 10%
50 13333 006114            EHALT           ;MORE THAN 10% ADDRESS FIELD ERRORS
51                                         ;DURING 100 SECTOR READS IN DIFF.
52                                         ;TRACKS, AC0= ERRORPERCENT.
53 13334 020651            LDA     0,A37E3
54 13335 024200            LDA     1,FD010 ;AC1:= 10.
55 13336 106033            ADCZ#   0,1,SNC ;SKIP IF # PAE < 10%
56 13337 006114            EHALT           ;MORE THAN 10% PARITY ERRORS DURING
57                                         ;100 SECTOR READS IN DIFF TRACKS,
58                                         ;AC0= ERRORPERCENT.
```

```
I 0151  .MAIN
01
02 13340 020646          LDA     0,A37E4
03 13341 024200          LDA     1,FD010 ;AC1:= 10.
04 13342 106033          ADCZ#   0,1,SNC ;SKIP IF # DE < 10%
05 13343 006114          EHALT           ;MORE THAN 10% DATA ERRORS DURING
06                                       ;100 SECTOR READS IN DIFF. TRACKS.
07                                       ;AC0= ERROR PERCENT.
08 13344 020643          LDA     0,A37E5
09 13345 024200          LDA     1,FD010 ;AC1:= 10.
10 13346 106033          ADCZ#   0,1,SNC ;SKIP IF # DSE < 10%
11 13347 006114          EHALT           ;MORE THAN 10% DATA SYNC ERRORS
12                                       ;DURING 100 SECTOR READS IN DIFF.
13                                       ;TRACKS. AC0= ERRORPERCENT.
14 13350 020640          LDA     0,A37E6
15 13351 024200          LDA     1,FD010 ;AC1:= 10.
16 13352 106033          ADCZ#   0,1,SNC ;SKIP IF # TO < 10.
17 13353 006114          EHALT           ;MORE THAN 10% TIME OUTS DURING
18                                       ;100 SECTOR READS IN DIFF. TRACKS.
19 13354 006113          LOOP            ;AC0= ERRORPERCENT
```

```
I 0152  .MAIN
01
02                ;LOOP  A0380 - A0390
03                ;
04                ;RANDOMLY READ LOOP. 100 RANDOMLY SELECTED SECTORS IN RANDOMLY
05                ;SELECTED TRACKS ARE READ. STATUS AND DATA ARE CHECKED. IN CA-
06                ;SE OF "SOFT" ERROR, THE SECTOR IS RE-READ UP TO "TRIES" TIMES
07                ;BEFORE ERRORHALT.
08                ;
09                ;FOD-CONDITIONS: POWER ON,DEVICE LOADED WITH TM-DISCETTE, WRITE
10                ;PROTECT ON OR OFF.
11
12
13 13355 000401 A0380:  JMP     .+1       ;DUMMY INSTR. FOR DEBUG-USE
14 13356 006110         SETP0
15 13357 006116         STATN             ;CHECK STATUS
16 13360 175777         175777
17 13361 006114         EHALT             ;STATUS ERROR - OTHER THAN BIT 5 IS
18                                        ;SET. AC0= RECEIVED STATUS,AC1= EXPEC-
19                                        ;TED STATUS.
20 13362 006136         RECAL             ;RECALIBRATE DISC-DRIVE
21 13363 006113         LOOP
22
23 13364 006142         PTFLL             ;FILL WRITE BUFFER WITH PATTERNS
24                                        ;USED DURING GENERATION FOR COM-
25                                        ;PARISION.
26
27 13365 000401 A0390:  JMP     .+1       ;DUMMY INSTR. FOR DEBUG-USE
28 13366 102440         SUBO    0,0       ;AC0:= 0
29 13367 040242         STA     0,TOTCO   ;CLEAR TOTAL RE-READ COUNTER
30 13370 040253         STA     0,WORK1   ;CLEAR SECTOR COUNTER
31 13371 006112         SETP2
32 13372 102440         SUBO    0,0       ;AC0:= 0
33 13373 040241         STA     0,RERCO
34 13374 020256         LDA     0,WTRIES  ;AC0:= "WTRIES"
35 13375 040240         STA     0,REWCO   ;SET "REWCO" = "WTRIES" TO INHIBIT RE-
36                                        ;WRITE-TRIES.
37 13376 010253         ISZ     WORK1     ;INCREMENT SECTOR COUNTER
38 13377 006135         RANTS             ;GET A RANDOM SECTOR NO. TO AC0
39                                        ;AND A RANDOM TRACK NO. TO AC1.
40 13400 042234         STA     0,@AWBUF  ;STORE SECTOR NO AS 1. BYTE IN BUFFER
41 13401 006137         TRPOS             ;POSITION HEAD TO TRACK SPEC. IN AC1
42 13402 022234 A0391:  LDA     0,@AWBUF
43 13403 006146         RDWRI             ;READ SECTOR SPEC. IN AC0 AND CHECK
44                                        ;STATUS AND DATA.
45 13404 000403         JMP     A0392     ;HARD ERROR, HALT AND GIVE UP
46 13405 000405         JMP     A0393     ;SOFT ERROR,READ AGAIN IF TRIES <"TRIES"
47 13406 000410         JMP     A0394     ;READ OK
48 13407 121000 A0392:  MOV     1,0
49 13410 006114         EHALT             ;HARD ERROR AFTER READ, AC0= RECEIVED
50                                        ;STATUS.
51 13411 000405         JMP     A0394     ;GIVE UP
52 13412 006147 A0393:  SOFER             ;SOFT ERROR AFTER READ, READ AGAIN IF
53                                        ;TRIES < "TRIES" ELSE HALT AND RET. +1
54 13413 000403         JMP     A0394     ;TRIES = "TRIES", GIVE UP
55 13414 000766         JMP     A0391     ;TRY TO READ AGAIN
56 13415 063077         HALT              ;ERRONEOUS RETURN FROM "SOFER". NOT
57                                        ;USED WHILE "REWCO" = 3.
58 13416 006113 A0394:  LOOP
```

```
I 0153  .MAIN
01
02 13417 020253          LDA     0,WORK1 ;AC0:= NO. OF SECTORS READ
03 13420 006152          TRMES           ;TYPE TOTAL RE-READ PERCENT,RET +1 OR +2
04 13421 000410          JMP     A0395   ;PERCENT < 20, DONT ANNOUNCE IT.
05                                       ;PERCENT >= 20
06 13422 006046          CDICL
07 13423 006040          CMESS
08 13424 011561          LOP39           ;"    IN LOOP A0390"
09 13425 006043          CCRLF
10 13426 006044          CDISP
11 13427 011561          LOP39
12 13430 006047          CDATT
13
14 13431 002171 A0395:   JMP     @GRRET  ;RETURN FROM THIS LOOP-GROUP
```

```
 0154   .MAIN
01
02                ;LOOP  A0400 - A0410
03                ;
04                ;THIS LOOP WILL READ SECTOR 10. AND 21. IN ALL TRACKS.
05                ;UP TO "TRIES" TIMES RE-READ WILL BE EXECUTED BEFORE ERRORHALT.
06                ;THE TRACK NO. CALCULATION WILL BE DONE TO PRODUCE WORST-CASE
07                ;HEAD MOVEMENT (SEE PROCEDURE "TRCAL").
08                ;
09                ;FDD-CONDITIONS: POWER ON, DEVICE LOADED WITH TM-DISCETTE, WRITE
10                ;PROTECT ON OR OFF.
11
12
13 13432 054171 GR0004:  STA     3,GRRET
14 13433 000401 A0400:   JMP     .+1       ;DUMMY INSTR. FOR DEBUG-USE
15 13434 006110          SETP0
16 13435 006116          STATN             ;CHECK STATUS
17 13436 175777          175777
18 13437 006114          EHALT             ;STATUS ERROR - OTHER THAN BIT 5 IS
19                                         ;SET. AC0= RECEIVED STATUS,AC1= EX-
20                                         ;PECTED STATUS.
21 13440 006136          RECAL             ;RECALIBRATE DISC DRIVE
22 13441 006113          LOOP
23
24 13442 006142          PTFLL             ;FILL WRITE BUFFER WITH PATTERNS USED
25                                         ;DURING GENERATION, FOR COMPARISION.
26
27 13443 000401 A0410:   JMP     .+1       ;DUMMY INSTR. FOR DEBUG-USE
28 13444 102440          SUBO    0,0       ;AC0:= 0
29 13445 040242          STA     0,TOTCO   ;CLEAR RE-READ COUNTER
30 13446 040253          STA     0,WORK1   ;CLEAR SECTOR COUNTER
31 13447 020256          LDA     0,WTRIES  ;AC0:= "WTRIES"
32 13450 040240          STA     0,REWCO   ;SET REWCO = "WTRIES" TO INHIBIT WRITE
33                                         ;RETRIES
34 13451 006110          SETP0
35 13452 006136          RECAL             ;RECALIBRATE DISC DRIVE
36 13453 126440          SUBO    1,1       ;AC1:= 0
37 13454 006137 A0411:   TRPOS             ;POSITION HEAD TO TRACK SPEC. IN AC1
38 13455 020200          LDA     0,FD010   ;AC0:= 10. = FIRST SECTOR TO READ
39 13456 126440 A0412:   SUBO    1,1       ;AC1:= 0
40 13457 044241          STA     1,RERCO   ;CLEAR RE-READ COUNTER
41 13460 010253          ISZ     WORK1     ;INCREMENT SECTOR COUNTER
42 13461 042234          STA     0,@AWBUF  ;STORE SECTOR NO AS 1. BYTE IN BUFFER
43 13462 022234 A0413:   LDA     0,@AWBUF
44 13463 006146          RDWRI             ;READ SECTOR SPEC. IN AC0 AND CHECK
45                                         ;STATUS AND DATA.
46 13464 000403          JMP     A0414     ;HARD ERROR, HALT AND GIVE UP
47 13465 000405          JMP     A0415     ;SOFT ERROR,READ AGAIN IF TRIES <"TRIES"
48 13466 000410          JMP     A0416     ;READ OK
49 13467 121000 A0414:   MOV     1,0
50 13470 006114          EHALT             ;HARD ERROR AFTER READ. AC0= RECEIVED
51                                         ;STATUS
52 13471 000405          JMP     A0416     ;GIVE UP
53 13472 006147 A0415:   SOFER             ;SOFT ERROR AFTER READ. READ AGAIN IF
54                                         ;TRIES < "TRIES" ELSE HALT AND RET. +1.
55 13473 000403          JMP     A0416     ;GIVE UP
56 13474 000766          JMP     A0413     ;READ AGAIN
57 13475 063077          HALT              ;ERRONEOUS RETURN FROM "SOFER". NOT
58                                         ;USED WHILE REWCO = 3
```

```
I 0155  .MAIN
01
02 13476 022234 A0416:  LDA     0,@AWBUF;AC0:= SECTOR READ
03 13477 024200         LDA     1,FD010 ;AC1:= 10.
04 13500 106414         SUB#    0,1,SZR ;SKIP IF SECTOR NO = 10.
05 13501 000403         JMP     A0417   ;SECTOR 10. AND 21. ARE READ IN CUR.
06                                      ;TRACK.
07 13502 020202         LDA     0,FD021 ;AC0:= 21.
08 13503 000753         JMP     A0412   ;READ SECTOR 21. IN CUR. TRACK
09 13504 006144 A0417:  CURTR           ;GET CURRENT TRACK TO AC2
10 13505 006150         CALTR           ;CALCULATE NEW TRACK NO TO AC1. RE-
11                                      ;TURN +2 IF AC2= 38, ELSE +1.
12 13506 000746         JMP     A0411   ;GO TO NEXT TRACK
13 13507 006113         LOOP            ;ALL TRACKS ARE READ.AND CHECKED
14 13510 020253         LDA     0,WORK1 ;AC0:= NO. OF SECTORS READ
15 13511 006152         TRMES           ;TYPE TOTAL RE-READ PERCENT,RET +1 OR +2
16 13512 000410         JMP     A0418   ;PERCENT < 20, DONT ANNOUNCE IT.
17                                      ;PERCENT >= 20
18 13513 006046         CDICL
19 13514 006040         CMESS
20 13515 011640         LOP41           ;"   IN LOOP A0410"
21 13516 006043         CCRLF
22 13517 006044         CDISP
23 13520 011640         LOP41
24 13521 006047         CDATT
25
26 13522 002171 A0418:  JMP     @GRRET  ;RETURN FROM THIS LOOP-GROUP
```

```
I 0156  .MAIN
01
02                ;LOOP  A0420 - A0440
03                ;
04                ;FIRST TEST OF WRITE LOGIC. THIS LOOPS WILL TRY TO WRITE ZEROS
05                ;IN ALL SECTORS IN TRACK 0. THE LOOP ONLY VERIFY, THAT THE WRITE
06                ;MECHANISM WORKS, I.E. THAT NO HARDWARE BITS ARE SET IN STATUS
07                ;AND THAT THE "WRITE-PRO" BIT HAS DISAPPEARED. NO DATA-
08                ;CHECK IN THIS LOOP.
09                ;
10                ;FDD-CONDITIONS: POWER ON, DEVICE LOADED WITH SCRATCH DISCETTE,
11                ;WRITE PROTECT OFF.
12
13 13523 000000 A44SE:  0                       ;SECTOR COUNTER
14
15 13524 054171 GR005:  STA     3,GRRET
16 13525 000401 A0420:  JMP     .+1             ;DUMMY INSTR. FOR DEBUG-USE
17 13526 006110         SETP0
18 13527 006065         TIMRO                   ;TEST STATUS AND WAIT UNTIL DRIVE
19 13530 000214         FD3K0                   ;GOES ON-LINE (MAX 3 SEC)
20 13531 006116         STATN
21 13532 040000         040000
22 13533 006114         EHALT                   ;DEVICE DIDN'T GO ON-LINE WITHIN
23 13534 006113         LOOP                    ;3 SEC
24
25 13535 000401 A0430:  JMP     .+1             ;DUMMY INSTR. FOR DEBUG-USE
26 13536 006112         SETP2
27 13537 060461         DIA     0,XFDD          ;CHECK STATUS
28 13540 101014         MOV#    0,0,SZR
29 13541 006114         EHALT                   ;STATUS <> 0, AC0= RECEIVED STATUS
30 13542 006113         LOOP
31
32 13543 000401 A0440:  JMP     .+1             ;DUMMY INSTR. FOR DEBUG-USE
33 13544 006111         SETP1
34 13545 006136         RECAL                   ;RECALIBRATE DISC DRIVE
35 13546 152520         SUBZL   2,2             ;AC2:= 1 = FIRST SECTOR TO WRITE
36 13547 050754         STA     2,A44SE         ;SET SECTOR COUNTER
37 13550 006116         STATN                   ;CHECK STATUS
38 13551 177777         177777
39 13552 006114         EHALT                   ;STATUS ERROR, AC0= RECEIVED STATUS
40                                              ;AC1= "NON"-EXPECTED STATUS
41 13553 024233 A0441:  LDA     1,DFIEL         ;AC1:= DATA FIELD CHAR.
42 13554 067061         DOC     1,XFDD          ;SET DATA FIELD CHAR. REGISTER
43 13555 024215         LDA     1,MD128         ;AC1:= -128.
44 13556 102440         SUBC    0,0             ;AC0:= 0
45 13557 062061 A0442:  DOB     0,XFDD          ;SEND CHAR TO FIFO
46 13560 125404         INC     1,1,SZR
47 13561 000776         JMP     A0442           ;FILL ON
48                                              ;FIFO IS FULL
49 13562 020741         LDA     0,A44SE         ;AC0:= SECTOR TO WRITE
50 13563 024223         LDA     1,FO400         ;AC1:= 400 = WRITE COMMAND
51 13564 107000         ADD     0,1
52 13565 065161         DOAS    1,XFDD
53 13566 006063         TIMSK                   ;WAIT FOR DONE = 1 (MAX 2 SEC)
54 13567 003720         2000.
55 13570 063661         SKPDN   XFDD
56 13571 006114         EHALT                   ;DONE ISN'T SET WITHIN 2 SEC. AFTER
57                                              ;WRITING A SECTOR IN TRACK 0.
58 13572 060461         DIA     0,XFDD          ;GET STATUS
59 13573 030730         LDA     2,A44SE
60 13574 101014         MOV#    0,0,SZR
61 13575 006114         EHALT                   ;STATUS <> 0 AFTER WRITING A SEC-
62                                              ;TOR IN TRACK 0. AC0= RECEIVED
63                                              ;STATUS, AC2= SECTOR WRITTEN
```

```
I 0157   .MAIN
01
02 13576 010725          ISZ     A44SE
03 13577 020724          LDA     0,A44SE ;AC0:= NEW SECTOR TO WRITE
04 13600 024235          LDA     1,MXSEC ;AC1:= 26.
05 13601 106432          SUBZ#   0,1,SZC ;SKIP IF NEW SECTOR NO. > 26.
06 13602 000751          JMP     A0441   ;WRITE NEXT SECTOR
07 13603 006113          LOOP             ;ALL SECTORS IN TRACK 0 ARE WRITTEN.
```

```
I 0158  .MAIN
01
02                ;LOOP  A0450 - A0460
03                ;
04                ;THIS LOOP TRIES TO WRITE ZEROS IN SECTOR 1,4,8,16,24,2 IN
05                ;DIFFERENT RANDOMLY SELECTED TRACKS. LIKE IN LOOP A0440 NO
06                ;DATA-CHECK IS PERFORMED, ONLY HARDWARE-STATUS IS CHECKED.
07                ;
08                ;FDD-CONDITIONS: POWER ON, DEVICE LOADED WITH SCRATCH-DIS-
09                ;CETTE, WRITE PROTECT OFF.
10
11 13604 000413           JMP     A0450    ;JUMP OVER CONSTANTS
12
13 13605 013606 SEC46:    .+1              ;POINTER TO TABLE WITH SECTOR-NO.'S
14 13606 000001           1                ;WHICH HAVE TO BE WRITTEN IN THIS LOOP
15 13607 000004           4
16 13610 000010           8.
17 13611 000020           16.
18 13612 000030           24.
19 13613 000002           2
20 13614 000000           0
21 13615 000000 A46SE:    0                ;SECTOR COUNTER
22 13616 000005 A46WC:    5                ;RE-WRITE COUNTER
23
24 13617 000401 A0450:    JMP     .+1      ;DUMMY INSTR. FOR DEBUG-USE
25 13620 006110           SETP0
26 13621 006136           RECAL            ;RECALIBRATE DISC-DRIVE
27 13622 006116           STATN            ;CHECK STATUS
28 13623 177777           177777
29 13624 006114           EHALT            ;STATUS ISN'T CORRECT. AC0= RECEIVED
30 13625 006113           LOOP             ;STATUS, AC1="NON"-EXPECTED STATUS
31
32 13626 000401 A0460:    JMP     .+1      ;DUMMY INSTR FOR DEBUG-USE
33 13627 006111           SETP1
34 13630 006135           RANTS            ;GET A RANDOM TRACKNO TO AC1
35 13631 044253           STA     1,WORK1
36 13632 006137           TRPOS            ;POSITION HEAD TO TRACK SPEC IN AC1
37 13633 030752           LDA     2,SEC46  ;AC2:= POINTER TO SECTOR TABLE
38 13634 021000 A0461:    LDA     0,0,2    ;AC0:= SECTOR TO WRITE
39 13635 101015           MOV#    0,0,SNR  ;SKIP IF NOT LAST SECTOR TO WRITE
40 13636 000443           JMP     A0466
41 13637 050254           STA     2,WORK2
42 13640 040755           STA     0,A46SE
43 13641 024233           LDA     1,DFIEL  ;AC1:= DATA FIELD CHAR.
44 13642 067061           DOC     1,XFDD   ;SET DATA FIELD CHAR REGISTER
45 13643 024215           LDA     1,MD128  ;AC1:= -128.
46 13644 102440           SUBO    0,0      ;AC0:= 0
47 13645 062061 A0462:    DOB     0,XFDD   ;SEND CHAR TO FIFO
48 13646 125404           INC     1,1,SZR  ;SKIP IF 128 CHARS ARE TRANFERRED
49 13647 000776           JMP     A0462    ;FILL ON
50                                         ;FIFO IS FULL
51 13650 020256           LDA     0,WTRIES
52 13651 040745           STA     0,A46WC  ;SET RE-WRITE COUNTER
53 13652 020743 A0463:    LDA     0,A46SE  ;AC0:= SECTOR TO WRITE
54 13653 024223           LDA     1,FO400  ;AC1:= 400 = WRITE COMMAND
55 13654 107000           ADD     0,1
56 13655 065161           DOAS    1,XFDD
57 13656 006063           TIMSK            ;WAIT FOR DONE = 1 (MAX 2 SEC)
58 13657 003720           2002.
59 13660 063661           SKPDN   XFDD
60 13661 006114           EHALT            ;DONE ISN'T SET WITHIN 2 SEC AFTER
61                                         ;WRITING.
```

```
I 0159  .MAIN
01
02 13662 060461          DIA     0,XFDD  ;GET STATUS
03 13663 024732          LDA     1,A46SE
04 13664 030253          LDA     2,WORK1
05 13665 034173          LDA     3,FD002 ;AC3:= 2
06 13666 117414          AND#    0,3,SZR ;SKIP IF NOT TIME OUT
07 13667 000406          JMP     A0465
08 13670 101014          MOV#    0,0,SZR
09 13671 006114          EHALT           ;STATUS <> 0 AFTER WRITING A SECTOR.
10                                       ;AC0= RECEIVED STATUS,AC1= SECTOR
11                                       ;NO.,AC2= TRACK NO.
12 13672 030254 A0464:   LDA     2,WORK2
13 13673 151400          INC     2,2     ;INCREMENT TO NEXT SECTOR IN TABLE
14 13674 000740          JMP     A0461   ;WRITE NEXT SECTOR
15 13675 014721 A0465:   DSZ     A46WC   ;SKIP IF WRITE-TRIES = "WTRIES"
16 13676 000754          JMP     A0463   ;TRY TO WRITE AGAIN
17 13677 006114          EHALT           ;STATUS ERROR.STILL TIME-OUT AFTER
18                                       ;"TRIES" - RE-WRITES.AC0= RECEIVED
19                                       ;STATUS, AC1= SECTOR NO, AC2=
20 13700 000772          JMP     A0464   ;TRACK NO.
21 13701 006113 A0466:   LOOP
```

```
                .MAIN
                ;LOOP  A0470 - A0480
                ;
                ;FIRST TRY TO READ THE SECTORS AFTER WRITING. ALL SECTORS IN
                ;TRACK 21. AND 42. ARE WRITTEN WITH PATTERNS FOLLOWED
                ;BY STATUSCHECK. AFTER WRITING A SECTOR, IT IS CHECKREAD. WHERE
                ;STATUS AND DATA ARE CHECKED. AFTER 100 WRITE/READS THE AM-
                ;OUNT OF ERRORS ARE EXAMINED, AND IF > 10% AN ERROR-HALT IS THE
                ;RESULT.
                ;
                ;FDD-CONDITIONS: POWER ON, DEVICE LOADED WITH SCRATCH DISCETTE,
                ;WRITE PROTECT OFF.

13762 000411            JMP     A0470   ;JUMP OVER CONSTANTS

13763 000000 A48E1:     0               ;ERRORCOUNTER (POSITION ERRORS)
13764 000000 A48E2:     0               ;ERRORCOUNTER (ADDRESS FIELD ERRORS)
13765 000000 A48E3:     0               ;ERRORCOUNTER (PARITY ERRORS)
13766 000000 A48E4:     0               ;ERRORCOUNTER (DATA ERRORS)
13767 000000 A48E5:     0               ;ERRORCOUNTER (DATA SYNC ERRORS)
13770 000000 A48E6:     0               ;ERRORCOUNTER (TIME OUTS)
13771 000004 A48C0:     4               ;READ COUNTER
13772 000001 A48SE:     1               ;SECTOR COUNTER

13773 000401 A0470:     JMP     .+1     ;DUMMY INSTR. FOR DEBUG-USE
13774 006110            SETP0
13775 006136            RECAL           ;RECALIBRATE DISCDRIVE
13776 006116            STATN           ;CHECK STATUS
13777 177777            177777
13720 006114            EHALT           ;STATUS <> 0. AC0= RECEIVED STATUS,
13721 006113            LOOP            ;AC1= "NON"-EXPECTED STATUS.

13722 000401 A0480:     JMP     .+1     ;DUMMY INSTR. FOR DEBUG-USE
13723 006110            SETP0
13724 024202            LDA     1,FD021 ;AC1:= 21. = FIRST TRACK TO WRITE
13725 102440 A0481:     SUB0    0,0     ;AC0:= 0
13726 040755            STA     0,A48E1 ;CLEAR ERRORCOUNTERS
13727 040755            STA     0,A48E2
13730 040755            STA     0,A48E3
13731 040755            STA     0,A48E4
13732 040755            STA     0,A48E5
13733 040755            STA     0,A48E6
13734 020175            LDA     0,FD004 ;AC0:= 4
13735 040754            STA     0,A48C0 ;SET COUNTER
13736 006142            PTFLL           ;FILL WRITE BUFFER WITH PATTERNS
13737 006137            TRPOS           ;POSITION HEAD TO TRACK SPEC IN AC1
13740 044253            STA     1,WORK1
13741 102520 A0482:     SUBZL   0,0     ;AC0:= 1 = FIRST SECTOR TO WRITE
13742 040750            STA     0,A48SE
13743 042234 A0483:     STA     0,@AWBUF;STORE SECTOR NO. AS 1. BYTE IN BUFFER
13744 006143            WRSEC           ;WRITE SECTOR NO. SPEC IN AC0
13745 006146            RDWRI           ;READ SECTOR SPEC. IN AC0, CHECK
                                        ;STATUS AND DATA
13746 000403            JMP     A0484   ;HARD ERROR, AC1= STATUS
13747 000407            JMP     A0485   ;"SOFT" ERROR, AC1= STATUS
13750 000431            JMP     A0486   ;READ OK
13751 030253 A0484:     LDA     2,WORK1
13752 121000            MOV     1,0
13753 026234            LDA     1,@AWBUF
13754 006114            EHALT           ;HARD ERROR AFTER CHECKREAD, AC0=
                                        ;RECEIVED STATUS, AC1= SECTOR NO.,
13755 000424            JMP     A0486   ;AC2= TRACK NO.
```

```
| 0161  .MAIN
01
02 13756 125015 A0485:  MOV#    1,1,SNR ;SKIP IF NOT DATA WRONG
03 13757 010727         ISZ     A48E4   ;INCREMENT ERROR COUNTER
04 13760 102030         ADC     0,0     ;AC0:= -1
05 13761 122414         SUB#    1,0,SZR ;SKIP IF DATA SYNC ERROR
06 13762 000403         JMP     .+3
07 13763 010724         ISZ     A48E5   ;INCREMENT ERRORCOUNTER
08 13764 000415         JMP     A0486
09 13765 020216         LDA     0,F0010 ;AC0:= 10
10 13766 107414         AND#    0,1,SZR ;SKIP IF NOT POSITION ERROR
11 13767 010714         ISZ     A48E1   ;INCREMENT ERRORCOUNTER
12 13770 020227         LDA     0,F010K ;AC0:= 10000
13 13771 107414         AND#    0,1,SZR ;SKIP IF NOT ADDRESS FIELD ERROR
14 13772 010712         ISZ     A48E2   ;INCREMENT ERRORCOUNTER
15 13773 020217         LDA     0,F0040 ;AC0:= 40
16 13774 107414         AND#    0,1,SZR ;SKIP IF NOT PARITY ERROR
17 13775 010713         ISZ     A48E3   ;INCREMENT ERRORCOUNTER
18 13776 020173         LDA     0,FD002 ;AC0:= 000002
19 13777 107414         AND#    0,1,SZR ;SKIP IF NOT TIME OUT
20 14000 010710         ISZ     A48E6   ;INCREMENT ERRORCOUNTER
21 14001 010711 A0486:  ISZ     A48SE   ;INCREMENT SECTOR NO
22 14002 020710         LDA     0,A48SE
23 14003 024235         LDA     1,MXSEC ;AC1:= 26.
24 14004 106432         SUBZ#   0,1,SZC ;SKIP IF NEW SECTOR NO > 26.
25 14005 000736         JMP     A0483   ;WRITE/READ NEXT SECTOR
26 14006 006151         COMPT           ;COMPLEMENT PATTERNS IN WRITE BUFF.
27 14007 014702         DSZ     A48C0   ;DECREMENT FOUR-COUNTER
28 14010 000731         JMP     A0482   ;WRITE/READ TRACK AGAIN
29                                      ;FOUR TIMES WRITE/READ FINISHED
30 14011 020175         LDA     0,FD004 ;AC0:= 4
31 14012 040677         STA     0,A48C0 ;SET COUNTER
32 14013 030253         LDA     2,WORK1
33 14014 020667         LDA     0,A48E1 ;AC0:= NO. OF POSITION ERRORS
34 14015 024200         LDA     1,FD010 ;AC1:= 10.
35 14016 106033         ADCZ#   0,1,SNC ;SKIP IF # PE < 10.
36 14017 006114         EHALT           ;MORE THAN 10% POSITION ERRORS
37                                      ;DURING 100 SECTOR WRITE/READ. AC0=
38                                      ;ERRORPERCENT, AC2= TRACK NO.
39 14020 020664         LDA     0,A48E2 ;AC0:= NO OF ADDRESS FIELD ERRORS
40 14021 024200         LDA     1,FD010 ;AC1:= 10.
41 14022 030253         LDA     2,WORK1
42 14023 106033         ADCZ#   0,1,SNC ;SKIP IF # ADE < 10.
43 14024 006114         EHALT           ;MORE THAN 10% ADDRESS FIELD ERRORS
44                                      ;DURING 100 SECTOR WRITE/READ. AC0=
45                                      ;ERRORPERCENT, AC2= TRACK NO.
46 14025 020660         LDA     0,A48E3 ;AC0:= NO OF PARITY ERRORS
47 14026 024200         LDA     1,FD010 ;AC1:= 10.
48 14027 030253         LDA     2,WORK1
49 14030 106033         ADCZ#   0,1,SNC ;SKIP IF # PAE < 10
50 14031 006114         EHALT           ;MORE THAN 10% PARITY ERRORS
51                                      ;DURING 100 SECTOR WRITE/READ. AC0=
52                                      ;ERRORPERCENT, AC2= TRACK NO.
53 14032 020654         LDA     0,A48E4 ;AC0:= NO OF DATA ERRORS
54 14033 024200         LDA     1,FD010 ;AC1:= 10.
55 14034 030253         LDA     2,WORK1
56 14035 106033         ADCZ#   0,1,SNC ;SKIP IF # DE < 10.
57 14036 006114         EHALT           ;MORE THAN 10% DATA ERRORS
58                                      ;DURING 100 SECTOR WRITE/READ. AC0=
59                                      ;ERRORPERCENT, AC2= TRACK NO.
```

```
I 0162  .MAIN
01
02 14037 020650          LDA     0,A48E5 ;AC0:= NO OF DATA SYNC ERRORS
03 14040 024200          LDA     1,FD010 ;AC1:= 10.
04 14041 030253          LDA     2,WORK1
05 14042 106033          ADCZ#   0,1,SNC ;SKIP IF # DSE < 10.
06 14043 006114          EHALT           ;MORE THAN 10% DATA SYNC ERRORS
07                                       ;DURING 100 SECTOR WRITE/READ. AC0=
08                                       ;ERRORPERCENT, AC2= TRACK NO.
09 14044 020644          LDA     0,A48E6 ;AC0:= NO OF TIME OUTS
10 14045 024200          LDA     1,FD010 ;AC1:= 10.
11 14046 030253          LDA     2,WORK1
12 14047 106033          ADCZ#   0,1,SNC ;SKIP IF # TC < 10.
13 14050 006114          EHALT           ;MORE THAN 10% TIME OUTS DURING
14                                       ;100 SECTOR WRITE/READ. AC0=
15                                       ;ERRORPERCENT,AC2= TRACK NO.
16
17 14051 024253          LDA     1,WORK1 ;AC1:= CURRENT TRACK NO.
18 14052 020202          LDA     0,FD021 ;AC0:= 21.
19 14053 106414          SUB#    0,1,SZR ;SKIP IF CURRENT TRACK = 21.
20 14054 000403          JMP     A0487   ;CURRENT TRACK = 42. EXIT
21 14055 024204          LDA     1,FD042 ;AC1:= 42.
22 14056 000647          JMP     A0481   ;WRITE/READ TRACK 42.
23 14057 006113 A0487:   LOOP
```

```
I 0163  .MAIN
01
02                ;LOOP  A0490
03                ;
04                ;THIS LOOP WILL WRITE A RANDOMLY SELECTED SECTOR IN A TRACK,
05                ;STARTING WITH TRACK 0, AND THEN 76,1,75,2,74 . . .
06                ;AFTER WRITING A SECTOR, THE HEAD IS POSITIONED TO THE NEXT
07                ;TRACK, AND THEN BACK AGAIN FOR CHECK-READ. DURING
08                ;CHECK-READ UP TO "TRIES" RETRIES WILL BE ALLOWED, AND IF STILL
09                ;NOT OK, MAX "WTRIES" RE-WRITES WILL TAKE PLACE.
10                ;
11                ;FDD-CONDITIONS: POWER ON, DEVICE LOADED WITH SCRATCH-DISCETTE,
12                ;WRITE PROTECT OFF.
13
14 14060 000403          JMP      A0490    ;JUMP OVER CONSTANTS
15
16 14061 000000 A49WT:   0                 ;LAST WRITTEN TRACK
17 14062 000000 A49WS:   0                 ;LAST WRITTEN SECTOR
18
19 14063 000401 A0490:   JMP      .+1      ;DUMMY INSTR. FOR DEBUG-USE
20 14064 006110          SETP0
21 14065 006136          RECAL             ;RECALIBRATE DISC-DRIVE
22 14066 126440          SUBO     1,1      ;AC1:= 0
23 14067 006137 A0491:   TRPOS             ;POSITION HEAD TO TRACK SPEC IN AC1
24 14070 044771          STA      1,A49WT
25 14071 006135          RANTS             ;GET A RANDOM SECTOR NO TO AC0
26 14072 040770          STA      0,A49WS
27 14073 042234          STA      0,@AWBUF;STORE SECTOR NO. AS 1. BYTE IN BUFFER
28 14074 126440 A0492:   SUBO     1,1      ;AC1:= 0
29 14075 044241          STA      1,RERCO  ;CLEAR RE-READ COUNTERS
30 14076 044242          STA      1,TOTCO
31 14077 044240          STA      1,REWCO  ;CLEAR RE-WRITE COUNTER
32 14100 020762 A0493:   LDA      0,A49WS  ;AC0:= SECTOR TO WRITE
33 14101 006143          WRSEC             ;WRTTE SECTOR SPEC. IN AC0
34 14102 030757          LDA      2,A49WT  ;AC2:= CURRENT TRACK NO.
35 14103 006150          CALTR             ;CALCULATE NEW TRACK NO. TO AC1.
36                                         ;RETURN +2 IF AC2= 38.,ELSE +1
37 14104 000402          JMP      A0494
38 14105 126440          SUBO     1,1      ;AC1:= 0
39 14106 006137 A0494:   TRPOS             ;POSITION HEAD TO TRACK SPEC. IN AC1
40 14107 024752          LDA      1,A49WT  ;AC1:= LAST WRITTEN TRACK
41 14110 006137          TRPOS             ;POSITION HEAD TO TRACK SPEC IN AC1
42 14111 020751 A0495:   LDA      0,A49WS  ;AC0:= LAST WRITTEN SECTOR
43 14112 006146          RDWRI             ;READ SECTOR SPEC. IN AC0 AND CHECK
44                                         ;STATUS AND DATA
45 14113 000403          JMP      A0496    ;HARD ERROR, HALT AND GIVE UP
46 14114 000405          JMP      A0497    ;"SOFT" ERROR,READ AGAIN IF TRIES <"TRIES"
47 14115 000410          JMP      A0498    ;READ OK
48 14116 121600 A0496:   MOV      1,0
49 14117 006114          EHALT             ;HARD ERROR AFTER READ, AC0= RECEIVED
50                                         ;STATUS.
51 14120 000405          JMP      A0498    ;GIVE UP, GO TO NEXT TRACK.
52 14121 006147 A0497:   SOFER             ;SOFT ERROR AFTER READ. READ AGAIN IF
53                                         ;TRIES <"TRIES" ELSE WRITE AGAIN IF WRITE-
54                                         ;TRIES < "WTRIES" ELSE HALT AND GIVE UP.
55 14122 000403          JMP      A0498    ;GIVE UP. GO TO NEXT TRACK
56 14123 000766          JMP      A0495    ;READ AGAIN
57 14124 000754          JMP      A0493    ;WRITE AGAIN
58 14125 030734 A0498:   LDA      2,A49WT  ;AC2:= LAST WRITTEN TRACK
59 14126 006150          CALTR             ;CALCULATE NEW TRACK NO TO AC1
60                                         ;RETURN +2 IF AC2= 38, ELSE +1
61 14127 000740          JMP      A0491    ;WRITE/READ NEW TRACK
62 14130 026113          LOOP
```

```
1 0164  .MAIN
01
02                ;LOOP  A0500
03                ;
04                ;THIS LOOP WILL WRITE ZEROS IN ALL SECTORS OF THE DISCETTE.
05                ;THIS INITIALIZING IS NECESSARY IN CASE OF RESTART OF THE WRITE-
06                ;TEST LOOPS. DRIVE-TERMINATION IS DETECTED BY INTERRUPT. NO
07                ;READ-CHECK IS PERFORMED.
08                ;
09                ;FDD-CONDITIONS: POWER ON, DEVICE LOADED WITH SCRATCHDISCETTE,
10                ;WRITE PROTECT OFF.
11
12 14131 000403           JMP     A0500   ;JUMP OVER CONSTANTS
13
14 14132 000000 A50TR:    0               ;TRACK COUNTER
15 14133 000001 A50SE:    1               ;SECTOR COUNTER
16
17 14134 000401 A0500:    JMP     .+1     ;DUMMY INSTR. FOR DEBUG-USE
18 14135 006110           SETP0
19 14136 062677           IORST
20 14137 006136           RECAL           ;RECALIBRATE DISC-DRIVE
21 14140 060361           NIOP    XFDD    ;CLEAR DATA BUFFER
22 14141 102520           SUBZL   0,0     ;AC0:= 1 = FIRST SECTOR TO WRITE
23 14142 040771           STA     0,A50SE ;SET SECTOR COUNTER
24 14143 126440           SUBO    1,1     ;AC1:= 0 = FIRST TRACK TO WRITE
25 14144 044766           STA     1,A50TR ;SET TRACK COUNTER
26 14145 102440 A0501:    SUBO    0,0     ;AC0:= 0 = INTERRUPT HANDLE FLAG
27 14146 024220           LDA     1,F0077 ;AC1:= 000077 = INTERRUPT MASK
28 14147 006133           SETIN           ;SET UP INTR. SERVICE ROUT.
29 14150 060177           INTEN           ;ENABLE INTERRUPTS
30 14151 024761           LDA     1,A50TR ;AC1:= TRACK NO.
31 14152 006137           TRPOS           ;POSITION HEAD TO TRACK SPEC. IN AC1
32 14153 034237           LDA     3,INTOC ;AC3:= INTERRUPT OCCOURED FLAG
33 14154 175005           MOV     3,3,SNR ;SKIP IF INTERRUPT OCCOURED
34 14155 006114           EHALT           ;NO INTERRUPT OCCOURED FROM FDD
                                          ;AFTER EXECUTION OF TRACK SEARCH.
35                                        ;AC1= TRACK SEARCHED.
36
37 14156 006063 A0502:    TIMSK           ;WAIT FOR BUSY = 0 (MAX 2 SEC)
38 14157 003720           2000.
39 14160 063561           SKPBZ   XFDD
40 14161 006114           EHALT           ;BUSY DIDN'T RETURN TO 0 WITHIN 2 SEC
41 14162 006116           STATN           ;CHECK STATUS
42 14163 177777           177777
43 14164 006114           EHALT           ;STATUS <> 0, AC0= RECEIVED STATUS,
                                          ;AC1= "NON"-EXPECTED STATUS.
44
45 14165 102440           SUBO    0,0     ;AC0:= 0 = INTERRUPT HANDLE FLAG
46 14166 024220           LDA     1,F0077 ;AC1:= 000077 = INTERRUPT MASK
47 14167 006133           SETIN           ;SET UP INTR. SERVICE ROUT.
48 14170 060177           INTEN           ;ENABLE INTERRUPTS
49 14171 024233           LDA     1,DFIEL ;AC1:= DATA FIELD CHAR.
50 14172 067061           DOC     1,XFDD  ;SET DATA FIELD CHAR REGISTER
51 14173 020740           LDA     0,A50SE ;AC0:= SECTOR NO.
52 14174 024223           LDA     1,F0400 ;AC1:= 400 = WRITE COMMAND
53 14175 107000           ADD     0,1
54 14176 065061           DOA     1,XFDD
55 14177 060161           NIOS    XFDD    ;WRITE ONE SECTOR
56 14200 006063           TIMSK           ;WAIT FOR BUSY = 0 (MAX 2 SEC)
57 14201 003720           2000.
58 14202 063561           SKPBZ   XFDD
59 14203 006114           EHALT           ;BUSY DIDN'T RETURN TO 0 WITHIN 2 SEC
```

```
I 0165  .MAIN
01
02 14204 020727          LDA     0,A50SE ;AC0:= SECTOR NO
03 14205 024725          LDA     1,A50TR ;AC1:= TRACK NO
04 14206 034237          LDA     3,INTOC ;AC3:= INTERRUPT OCCOURED FLAG
05 14207 175005          MOV     3,3,SNR ;SKIP IF INTERRUPT OCCOURED
06 14210 006114          EHALT           ;NO INTERRUPT OCCOURED FROM FDD AFTER
07                                       ;EXECUTION OF SECTOR WRITE. AC0=
08                                       ;SECTOR NO, AC1= TRACK NO.
09 14211 006116          STATN           ;CHECK STATUS
10 14212 177777          177777
11 14213 006114          EHALT           ;STATUS <> 0 AFTER WRITE. AC0=
12                                       ;RECEIVED STATUS,AC1= "NON"-EXPECTED
13 14214 020717          LDA     0,A50SE
14 14215 006153          CALSEC          ;CALCULATE SECTOR NO
15 14216 000402          JMP     .+2     ;GO ON
16 14217 000403          JMP     .+3     ;TRACK FINISHED
17 14220 040713          STA     0,A50SE
18 14221 000735          JMP     A0502   ;WRITE NEXT SECTOR
19 14222 102520          SUBZL   0,0     ;AC0:= 1
20 14223 040710          STA     0,A50SE ;RESET SECTOR COUNTER
21 14224 010706          ISZ     A50TR   ;INCREMENT TRACK NO.
22 14225 020705          LDA     0,A50TR
23 14226 024236          LDA     1,MXTRA ;AC1:= 76.
24 14227 106432          SUBZ#   0,1,SZC ;SKIP IF NEW TRACK NO > 76.
25 14230 000715          JMP     A0501   ;WRITE NEXT TRACK
26 14231 006113          LOOP
27 14232 060277          INTDS
28 14233 002171          JMP     @GRRET  ;RETURN FROM THIS LOOP-GROUP
```

```
! 0166  .MAIN
01
02                ;LOOP A0510
03                ;
04                ;THIS LOOP IS A RELIABILITY LOOP, CALLED ONLY FROM START-
05                ;ADDRESS 405 AND 406. IT WILL WRITE/READ OR ONLY READ ALL
06                ;SECTORS ON THE DISCETTE. THE LOOP IS USEFUL FOR SCOPING,
07                ;WHILE ERROR-ACTION IS INHIBITED.
08
09 14234 000000 A51WS:  0
10 14235 000000 A51WT:  0
11
12 14236 054171 GR006:  STA     3,GRRET
13
14 14237 000401 A0510:  JMP     .+1       ;DUMMY INSTR. FOR DEBUG-USE
15 14240 126440         SUBO    1,1
16 14241 044241         STA     1,RERCO
17 14242 044240         STA     1,REWCO
18
19 14243 006110         SETP0
20 14244 006065         TIMRO             ;WAIT UNTIL DRIVE GOES ON-LINE (MAX 3 SEC)
21 14245 000214         FD3K0
22 14246 006116         STATN
23 14247 040000         040000
24 14250 006114         EHALT             ;DRIVE DIDN'T GO ON-LINE
25 14251 006116         STATN
26 14252 175777         175777
27 14253 006114         EHALT             ;STATUS ERROR, AC0= STATUS
28
29 14254 006136         RECAL
30 14255 126440         SUBO    1,1       ;AC1:= 0
31 14256 006137 A0511:  TRPOS
32 14257 044756         STA     1,A51WT
33 14260 102520         SUBZL   0,0       ;AC0:= 1
34 14261 040753 A0512:  STA     0,A51WS
35 14262 042234         STA     0,@AWBUF
36 14263 020751 A051W:  LDA     0,A51WS
37 14264 034251         LDA     3,DTINH
38 14265 175014         MOV#    3,3,SZR
39 14266 000413         JMP     A051R     ;TESTLOOP IS CALLED FROM SA 406
40 14267 006143         WRSEC
41
42 14270 024745         LDA     1,A51WT ;MOVE HEAD AND MOVE IT BACK
43 14271 176520         SUBZL   3,3       ;AC3:= 1
44 14272 125014         MOV#    1,1,SZR
45 14273 125212         MOVR#   1,1,SZC
46 14274 125401         INC     1,1,SKP
47 14275 166400         SUB     3,1
48 14276 006137         TRPOS
49 14277 024736         LDA     1,A51WT
50 14300 006137         TRPOS
51 14301 020733 A051R:  LDA     0,A51WS
52 14302 006146         RDWRI             ;CHECKREAD SECTOR
53 14303 000415         JMP     A0513     ;HARD ERROR
54 14304 000417         JMP     A0514     ;SOFT ERROR
55                                        ;READ OK
56 14305 020727 A0510:  LDA     0,A51WS
57 14306 101400         INC     0,0
58 14307 024235         LDA     1,MXSEC
59 14310 106432         SUBZ#   0,1,SZC ;SKIP IF SEC > 26.
60 14311 000750         JMP     A0512
61 14312 024723         LDA     1,A51WT
62 14313 125400         INC     1,1
63 14314 020236         LDA     0,MXTRA
64 14315 122432         SUBZ#   1,0,SZC ;SKIP IF TRACK > 76.
65 14316 000740         JMP     A0511
66 14317 000413         JMP     A0515
```

```
| 0167  .MAIN
01
02 14320 121000 A0513:  MOV     1,0
03 14321 006114         EHALT           ;HARD ERROR AFTER READ, AC0= STATUS
04 14322 000410         JMP     A0515
05 14323 034252 A0514:  LDA     3,NEGER
06 14324 175014         MOV#    3,3,SZR
07 14325 000760         JMP     A0510   ;NEGLECT ERRORS
08 14326 006147         SOFER
09 14327 000403         JMP     A0515   ;GIVE UP
10 14330 000751         JMP     A051R   ;READ AGAIN
11 14331 000732         JMP     A051W   ;WRITE AGAIN
12 14332 006113 A0515:  LOOP
13 14333 002171         JMP     @GRRET
14
15 14334 000000 TSEND:  0               ;END OF TEST
```

```
I 0168   .MAIN
01
02        001400              .END     REBIN
  0169   .MAIN

A0000   011777      97/03   132/10
A0010   012004      97/04   132/16
A0020   012011      97/05   132/22
A0030   012020      97/06   132/30
A0040   012027      97/07   132/38
A0050   012036      97/08   132/46
A0060   012045      97/09   132/54
A0070   012053      97/10   133/02
A0080   012061      97/11   133/09
A0090   012067      97/12   133/16
A0100   012075      97/13   133/23
A0110   012103      97/14   133/30
A0120   012112      97/15   133/38
A0130   012121      97/16   133/46
A0140   012132      97/17   133/56
A0150   012144      97/18   134/02
A0151   012162     134/16   134/23
A0160   012171      97/19   135/10   135/15
A0170   012201      97/20   135/27
A0171   012207     135/33   135/36
A0172   012223     135/45   135/48
A0173   012237     135/51   135/54   135/57
A0180   012241      97/21   136/02
A0190   012256      97/22   136/17
A0191   012262     136/21   136/27   136/30   136/57
A0192   012316     136/33   136/58
A0200   012321      97/23   137/04
A0210   012327      97/24   137/12
A0220   012340      97/25   138/13
A0221   012366     138/30   138/41
A0230   012367      97/26   138/43
A0240   012377      97/27   138/53
A0250   012407      97/28   139/02
A0260   012417      97/29   139/11
A0270   012453      97/30   139/48
A0280   012476      97/31   140/16
A0290   012506      97/32   140/25
A0300   012526      97/34   140/44
A0310   012557      97/35   141/13
A0320   012605      97/36   141/40
A0330   012740      97/37   144/13   144/19
A0331   012746     144/25   144/40
A0332   012771     144/44   144/55
A0333   013006     144/48   144/53   145/02
A0334   013015     144/56   145/12
A033R   012737     144/17   145/02   145/10
A0340   013025      97/38   146/17   146/25
A0341   013045     146/43   147/12
A0342   013066     146/56   147/02
A0343   013077     146/60   147/11
A0344   013107     147/17   147/22
A0345   013113     147/21   147/24   147/28
A0346   013121     147/31   147/36
A0347   013125     147/35   147/38   147/43
A0348   013133     147/46   147/50
A0349   013137     147/49   147/52   147/55
A0350   013140      97/39   148/12
A0360   013154      97/40   148/29
A0361   013165     148/38   148/52
```

0170 .MAIN

A0362	013172	148/41	148/45		
A0363	013175	148/42	148/49		
A0364	013201	148/43	148/48	148/51	148/55
A0370	013226	97/41	149/12	149/34	
A0371	013252	149/56	150/03		
A0372	013253	149/57	150/38		
A0373	013262	149/60	150/05		
A0375	013271	150/09	150/12		
A0376	013276	150/10	150/17		
A0377	013321	150/11	150/16	150/23	150/36
A0378	013324	150/04	150/41		
A0380	013355	97/42	152/13		
A0390	013365	97/43	152/27		
A0391	013402	152/42	152/55		
A0392	013407	152/45	152/48		
A0393	013412	152/46	152/52		
A0394	013416	152/47	152/51	152/54	152/58
A0395	013431	153/04	153/14		
A0400	013433	97/44	154/14		
A0410	013443	97/45	154/27		
A0411	013454	154/37	155/12		
A0412	013456	154/39	155/08		
A0413	013462	154/43	154/56		
A0414	013467	154/46	154/49		
A0415	013472	154/47	154/53		
A0416	013476	154/48	154/52	154/55	155/02
A0417	013504	155/05	155/09		
A0418	013522	155/16	155/26		
A0420	013525	97/46	156/16		
A0430	013535	97/47	156/25		
A0440	013543	97/48	156/32		
A0441	013553	156/41	157/06		
A0442	013557	156/45	156/47		
A0450	013617	97/49	158/11	158/24	
A0460	013626	97/50	158/32		
A0461	013634	158/38	159/14		
A0462	013645	158/47	158/49		
A0463	013652	158/53	159/16		
A0464	013672	159/12	159/20		
A0465	013675	159/07	159/15		
A0466	013701	158/40	159/21		
A0470	013713	97/51	160/14	160/25	
A0480	013722	97/52	160/33		
A0481	013725	160/36	162/22		
A0482	013741	160/48	161/28		
A0483	013743	160/50	161/25		
A0484	013751	160/54	160/57		
A0485	013756	160/55	161/02		
A0486	014001	160/56	160/62	161/08	161/21
A0487	014057	162/20	162/23		
A0490	014063	97/53	163/14	163/19	
A0491	014067	163/23	163/61		
A0492	014074	163/28			
A0493	014100	163/32	163/57		
A0494	014106	163/37	163/39		
A0495	014111	163/42	163/56		
A0496	014116	163/45	163/48		
A0497	014121	163/46	163/52		
A0498	014125	163/47	163/51	163/55	163/58

0171 .MAIN

A0500	014134	97/54	164/12	164/17					
A0501	014145	164/26	165/25						
A0502	014156	164/37	165/18						
A051Ø	014237	97/55	166/14						
A0511	014256	166/31	167/01						
A0512	014261	166/34	166/60						
A0513	014320	166/53	167/02						
A0514	014323	166/54	167/05						
A0515	014332	167/02	167/04	167/09	167/12				
A051U	014305	166/56	167/07						
A051R	014301	166/39	166/51	167/10					
A051W	014263	166/36	167/11						
A17CO	012167	135/12	135/31	135/35	135/43	135/47			
A33CO	012736	144/16	144/23	144/39	144/42	144/54			
A34CO	013024	146/22	146/32	147/11					
A34E1	013021	146/19	146/29	147/04	147/14				
A34E2	013022	146/20	146/29	147/07	147/28				
A34E3	013023	146/21	146/30	147/10	147/43				
A37CO	013211	149/20	149/44	150/02					
A37E1	013203	149/14	149/37	150/26	150/41				
A37E2	013204	149/15	149/38	150/29	150/47				
A37E3	013205	149/16	149/39	150/32	150/53				
A37E4	013206	149/17	149/40	150/18	151/02				
A37E5	013207	149/18	149/41	150/22	151/08				
A37E6	013210	149/19	149/42	150/35	151/14				
A44SE	013523	156/13	156/36	156/49	156/59	157/02	157/03		
A46SE	013615	158/21	158/42	158/53	159/03				
A46WC	013616	158/22	158/52	159/15					
A48CO	013711	160/22	160/44	161/27	161/31				
A48E1	013703	160/16	160/37	161/11	161/33				
A48E2	013704	160/17	160/38	161/14	161/39				
A48E3	013705	160/18	160/39	161/17	161/46				
A48E4	013706	160/19	160/40	161/03	161/53				
A48E5	013707	160/20	160/41	161/07	162/02				
A48E6	013710	160/21	160/42	161/20	162/09				
A48SE	013712	160/23	160/49	161/21	161/22				
A49WS	014062	163/17	163/26	163/32	163/42				
A49WT	014061	163/16	163/24	163/34	163/40	163/58			
A50SE	014133	164/15	164/23	164/51	165/02	165/13	165/17	165/20	
A50TR	014132	164/14	164/25	164/30	165/03	165/21	165/22		
A51WS	014234	166/09	166/34	166/36	166/51	166/56			
A51WT	014235	166/10	166/32	166/42	166/49	166/61			
AANUK	005314	75/45	75/47						
ACT0	001071	29/03	29/16						
ACT6	001072	29/09	29/17						
ACTN	001054	28/29	29/02						
ADDRS	001711	36/45	37/11	37/21	37/31	37/32	37/59		
ADLOP	010143	107/30	107/34						
AGTYP	003013	50/11	50/18	50/37					
ALPTT	002255	41/57	43/48						
AMEND	001721	38/14	38/22						
APASB	010673	118/04	118/09	118/42					
ARESW	001322	33/18	33/24						
ATBEG	000244	19/11	123/38						
ATEND	000245	19/12	123/39						
AWBUF	000234	19/03	109/43	112/22	116/43	129/39	129/50	144/24	144/43
		145/03	145/08	148/37	148/38	150/07	150/14	152/40	152/42
		154/42	154/43	155/02	160/50	160/59	163/27	166/35	
AXPAS	010672	118/03	118/41						

CHAR4	000721	27/21	27/30	27/33					
CHAR7	000776	27/28	28/17						
CHARN	011232	120/17	122/55	125/26	125/35				
CHARY	011231	120/08	120/16	122/54	125/34				
CHCR	001076	27/58	28/60	29/22					
CHDEV	007437	17/45	99/18						
CHFLG	000547	25/08	25/24	25/31	25/34	25/41	26/19		
CHINH	000455	24/40	26/04						
CHKER	001664	36/44	37/24	37/36					
CHLF	001077	27/55	28/03	28/62	29/23	30/37			
CHMAS	000461	24/09	24/44						
CHMIN	000546	25/17	25/40						
CHORZ	000775	27/27	27/46	28/05	28/16				
CHPLU	000545	25/07	25/25	25/39					
CHRA2	006007	81/15	81/54						
CHRA3	006120	83/15	83/46						
CHRAN	005554	77/16	78/34						
CHRET	000661	26/63	27/03	27/06	27/09	27/24	27/26	27/32	
CHSAV	000774	27/11	27/17	28/15					
CHSIG	000544	25/10	25/18	25/32	25/38				
CHSP	000662	25/19	25/28	26/09	27/00	27/33			
CHSPA	000663	25/09	27/01						
CHTAB	000767	27/19	28/10						
CIMOU	007502	99/24	99/58						
CLDRV	010753	119/12	122/53						
CLL01	000133	17/33	17/34						
CLL02	000134	17/35	17/36						
CLL03	000135	17/37	17/38						
CLL04	000136	17/39	17/40						
CLL05	000137	17/41	17/42						
CLL06	000140	17/43	17/44						
CLL07	000141	17/45	17/46						
CLL08	000142	17/47	17/48						
CLL09	000143	17/49	17/50						
CLL10	000144	17/51	17/52						
CLL11	000145	17/53	17/54						
CLL12	000146	17/55	17/56						
CLL13	000147	17/57	17/58						
CLL14	000150	17/59	17/60						
CLL15	000151	17/61	17/62						
CLL16	000152	17/63	18/00						
CLL17	000153	18/01	18/02						
CLLX1	000127	17/20	17/21						
CLLX2	000130	17/22	17/23						
CLLX3	000131	17/24	17/25						
CLLX4	000132	17/26	17/27						
CLORE	006122	16/60	94/47						
CLSEC	010460	18/01	113/28						
CMADR	005126	69/13	70/16	70/39					
CMESS	006040	16/17	33/31	34/34	34/43	34/51	38/50	38/54	42/29
		50/24	50/26	50/44	57/06	57/51	66/42	66/45	71/35
		71/43	71/45	76/43	90/29	91/17	93/56	94/19	94/27
		104/52	119/05	119/15	121/16	121/18	121/20	123/29	124/09
		124/12	124/14	124/16	130/09	130/48	153/07	155/19	
CMSK	001162	29/36	30/26						
COMPT	006151	17/62	124/48	161/26					
COUDI	006372	86/47	86/50	87/02	87/15				
COUNT	001670	37/15	37/27	37/33	37/40				
CPASS	006123	16/61							

0174 .MAIN

CPU0	002525	46/34							
CPU00	002524	46/16	46/32						
CPU1	002526	46/35							
CPU10	002535	46/42							
CPU11	002536	46/43							
CPU12	002537	46/44							
CPU13	002540	46/45							
CPU14	002541	46/46							
CPU15	002542	46/47							
CPU16	002543	46/48							
CPU17	002544	46/49							
CPU2	002527	46/36							
CPU20	002545	46/50							
CPU21	002546	46/51							
CPU22	002547	46/52							
CPU23	002550	46/53							
CPU24	002551	46/54							
CPU25	002552	46/55							
CPU3	002530	46/37							
CPU4	002531	46/38							
CPU5	002532	46/39							
CPU6	002533	46/40							
CPU7	002534	46/41							
CPUIN	002523	46/20	46/21	46/31					
CPUNO	003015	47/32	50/21	50/32	50/39	50/46			
CQUES	006071	16/42	43/45	66/47	68/06	68/24	70/06	70/17	70/28
		118/18	120/05	125/23					
CRBIR	001075	28/32	28/36	28/42	28/44	28/54	28/56	28/59	29/00
		29/21	29/29	29/31	29/33	29/34	29/49	30/30	30/33
		30/36	30/39	31/35	31/38	31/41	31/48		
CRESW	006073	16/44	31/19	31/26	34/30	89/23	89/31	90/04	90/07
		91/05	91/40	92/54					
CSAMS	006072	16/43	42/13	42/20	42/47	43/44	68/05	70/05	95/27
		95/39	95/51	121/06	121/49	122/05	122/25	123/05	124/39
		125/06							
CSAV0	006541	89/05	89/43	89/45	89/62	91/02	91/10	91/32	91/48
		91/50	91/58						
CSAV1	006540	89/04	89/41	89/46	89/61	91/03	91/12	91/33	91/47
		91/51	91/59						
CSAV2	006537	89/03	89/42	89/47	89/60	91/04	91/14	91/34	91/46
		91/52	91/60						
CSEC1	010500	113/28	113/41	113/45					
CSEC2	010501	113/29	113/42	113/46					
CSKP	003144	51/36	52/19	53/42					
CSOU1	010474	113/35	113/40						
CTBIN	006051	16/26							
CTDEC	006053	16/28	43/49	90/23	94/23	104/57	118/22		
CTOCT	006052	16/27	33/34	34/37	38/47	42/35	50/47	68/10	68/28
		68/38	68/42	70/10	70/21	70/32	91/11	91/13	91/15
		91/25	93/53						
CTYPE	006042	16/19							
CTZOC	006054	16/29	66/51						
CURTR	006144	17/52	114/55	129/61	155/09				
CWAIT	006061	16/34	27/61	31/23	33/19	34/39	34/45	34/53	49/30
		50/29	50/49	66/23	66/32	66/36	66/40	107/48	118/49
		129/17	134/11	136/08	136/43	138/14	139/39	142/02	
CXLPT	001771	38/34	38/59						
CXTTO	001772	38/36	38/60						
CYCLE	006451	15/53	89/02						

0176 ,MAIN

DTINH	000251	19/17	116/32	118/14	120/22	125/12	166/37		
DWPOF	011603	121/21	121/27	124/17	124/23	128/04			
DWPON	011536	127/06							
DXMMF	005101	68/08	69/04						
EBFLG	006535	88/60	89/10	89/58	91/39				
ECHO2	005755	81/22	81/30						
ECHO3	006076	83/22	83/30						
ECHO4	005453	77/23	77/33						
EFLAG	006532	88/57	89/11	89/14	89/28	89/55	91/28	91/55	
EHALT	006114	16/54	100/29	100/37	101/36	102/23	102/31	102/43	102/51
		103/25	103/33	103/46	103/54	104/05	107/31	110/31	110/46
		111/26	112/08	112/15	112/45	112/59	114/37	115/03	115/15
		115/25	115/35	115/43	115/50	129/23	130/31	132/13	132/19
		132/27	132/35	132/43	132/51	132/58	133/06	133/13	133/20
		133/27	133/35	133/43	133/52	134/00	134/18	135/19	135/39
		135/57	136/13	136/52	137/08	137/16	138/22	138/35	138/40
		138/47	138/53	138/57	138/60	139/05	139/08	139/22	139/26
		139/31	139/43	139/59	139/62	140/22	140/29	140/40	140/50
		140/60	141/04	141/08	141/22	141/26	141/32	141/36	141/46
		141/49	141/56	142/05	142/11	142/17	142/25	142/33	142/38
		142/42	142/48	142/51	142/58	142/63	143/04	143/12	143/20
		143/25	143/29	145/05	146/35	146/42	146/52	146/58	147/18
		147/25	147/32	147/39	147/47	147/53	148/16	148/23	148/46
		149/47	149/53	150/15	150/44	150/50	150/56	151/05	151/11
		151/17	152/17	152/49	154/18	154/50	156/22	156/29	156/39
		156/56	156/61	158/29	158/60	159/09	159/17	160/30	160/60
		161/36	161/43	161/50	161/57	162/06	162/13	163/49	164/34
		164/40	164/43	164/59	165/06	165/11	166/24	166/27	167/03
EMEND	001714	38/09	38/12						
ENBSW	010175	17/24	108/11						
ENRNK	003274	53/20							
ENTCO	006436	88/44	88/49	88/54					
ENTP0	006420	15/50	88/40						
ENTP1	006425	15/51	88/45						
ENTP2	006432	15/52	88/50						
ENTYM	003253	53/18							
ERBCT	006534	88/59	89/06	89/39	89/57	90/46			
ERHSW	006543	90/00	91/41						
ERRCT	006533	88/58	89/07	89/09	89/19	89/56	90/16	90/43	
ERRET	006625	90/46	91/35	91/53	91/61				
ERRNH	006707	91/43	91/50						
ERRO1	006630	90/45	91/02						
ERRO2	006672	91/30	91/37	91/57					
ERRO3	006713	91/08	91/55						
ERROR	006621	15/54	90/42						
EXDIS	000500	24/51	24/63	25/51	25/61				
EXMEM	005011	41/18	68/04						
EXMMF	005013	68/06	68/14						
EXMMT	005026	68/24	68/32						
EXPRT	005041	68/35	68/50						
EXTYP	000477	24/56	24/62	25/46	25/56				
FD001	000172	18/27	136/09	136/44					
FD002	000173	18/28	112/53	113/30	113/38	115/39	147/08	150/33	159/05
		161/18							
FD003	000174	18/29							
FD004	000175	18/30	160/43	161/30					
FD005	000176	18/31	136/19						
FD007	000177	18/32							
FD010	000200	18/33	134/12	139/40	147/22	147/36	147/50	149/43	150/42

FST05	000373	19/62	20/09						
FST06	000374	19/63	20/10						
FST07	000375	20/00	20/11						
FST10	000376	20/01	20/12						
FTES0	011102	19/57	121/05						
FTES1	011144	19/58	121/48						
FTES2	011157	19/59	122/04						
FTES3	011175	19/60	122/24						
FTES4	011233	19/61	123/04						
FTES5	011330	19/62	124/38						
FTES6	011346	19/63	125/05						
FTS01	011133	121/33	121/41						
FTS11	011150	121/53	121/60						
FTS21	011164	122/10	122/19						
FTS31	011201	122/29	122/36						
FTS41	011237	123/09	123/19						
FTS51	011337	124/46	124/52						
FTS61	011357	125/15	125/18						
FUB	000033	16/09	32/31	32/36					
FUMSK	007507	99/50	100/00						
FUN	000032	16/08	31/43	31/47	32/20	32/35			
GCUTR	010105	17/51	106/59						
GENTM	011274	20/01	124/04						
GERAN	007776	17/35	105/13						
GET	001671	37/43	37/51	37/54	37/56				
GETB1	006061	83/09	83/13	83/23					
GETBI	006054	15/45	83/04						
GETCH	005247	75/06	77/10	81/48	86/26				
GETD1	005436	77/10	77/14	77/24					
GETDC	005430	15/47	77/04						
GETO1	005740	81/09	81/13	81/23					
GETOK	005733	15/46	81/04						
GETRE	005325	75/06	75/32	75/42	75/56				
GETS1	005652	80/09	80/13	80/21					
GETSC	005645	15/48	80/04						
GETT1	006170	85/09	85/13	85/21					
GETTR	010051	106/10	106/17						
GETTX	006163	15/49	85/04						
GETYP	001423	34/23	34/27						
GMEND	001716	34/63	38/11	42/40					
GNDE1	011743	129/54	130/30						
GNDE2	011746	129/55	130/35						
GNDSC	011651	122/51	129/13						
GRAND	006134	17/36	105/60	144/26					
GR001	012336	122/42	138/10						
GR002	012475	122/43	140/15						
GR003	013017	122/44	146/16						
GR004	013432	122/45	154/13						
GR005	013524	122/46	156/15						
GR006	014236	122/47	166/12						
GR00A	011776	122/38	132/08						
GR00B	012165	122/39	135/09						
GR00C	012200	122/40	135/25						
GR00D	012320	122/41	137/02						
GRRET	000171	18/20	132/08	134/27	135/09	135/23	135/25	136/60	137/02
		137/20	138/10	140/01	140/15	145/14	146/16	153/14	154/13
		155/26	156/15	165/28	166/12	167/13			
GTCHR	001557	36/12	36/14	36/21	36/53	37/03	37/06		
GTTTI	001570	36/24	36/31						

GWERR	011752	130/33	130/38	130/43					
GWOUT	011741	130/04	130/19	130/24	130/45	130/54			
GWRS0	011673	129/37	130/40						
GWRS1	011674	129/39	129/47						
GWRS2	011676	129/43							
GWRS3	011702	129/50	129/58						
GWRS4	011705	129/53	130/39						
H1C11	005705	80/27	80/40						
H1C33	005704	80/28	80/39						
H1C40	005703	80/26	80/38						
H2C40	005773	81/28	81/40						
H2C60	005774	81/31	81/41	81/54					
H2C70	005775	81/34	81/42						
H3C40	006114	83/28	83/40						
H3C60	006115	83/31	83/41	83/46					
H3C62	006116	83/34	83/42						
H5C11	006216	85/26	85/35						
H5C12	006416	86/49	87/25						
H5C15	006415	86/42	87/24						
H5C40	006217	85/27	85/36						
H5C79	006220	85/37	85/45						
HAATT	001221	15/12	31/35						
HAERR	010643	116/29	117/00						
HC11	005504	76/56	77/52						
HC13	005505	76/50	77/53						
HC15	005506	76/53	77/54						
HC177	005326	75/36	75/57	76/05					
HC30	005507	76/02	77/55						
HC40	005510	77/29	77/56						
HC44	005511	76/18	76/30	77/57					
HC52	005512	76/32	77/34	77/58					
HC53	005513	77/40	77/59	78/18	79/09				
HC55	005514	77/43	77/60	78/22	79/12				
HC60	005573	77/31	77/46	78/34	78/51				
HC72	005515	77/37	77/61						
HLFCR	006417	87/14	87/26						
HMEND	000074	15/32	38/25	42/31					
IBZOT	006544	90/02	91/27						
ICHAR	000041	15/05	16/18						
ICLDR	011230	122/06	122/26	122/53	123/06	124/42	125/09		
ICRLF	000043	15/07	16/20						
IDATI	000047	15/11	16/24						
IDBIN	000055	15/17	16/30						
IDCME	011224	122/08	122/49						
IDDEC	000057	15/19	16/32						
IDICL	000046	15/10	16/23						
IDISP	000044	15/08	16/21						
IDIVD	000070	15/28	16/41						
IDIVS	000067	15/27	16/40						
IDOCT	000056	15/18	16/31						
IDOUT	000045	15/09	16/22						
IDX0	000020	14/53	51/06	51/32	53/28	53/39			
IDX1	000021	14/54	51/04	51/48	52/25	52/29	52/30	53/26	54/02
IDX2	000022	14/55	52/44	52/55	52/61	54/16	54/19		
IDX3	000023	14/56	51/08	52/11	53/30	95/29	95/31	95/41	95/43
		95/53	95/55						
IDZOC	000060	15/20	16/33						
IGNDS	011226	121/38	122/51	123/16	124/30				
IGNOR	001612	36/42	36/53	36/56					

0180 .MAIN

IGR01	011215	121/33	121/57	122/12	122/31	122/42	123/11		
IGR02	011216	121/34	122/13	122/32	122/43	123/12			
IGR03	011217	121/35	122/14	122/44	123/13				
IGR04	011220	121/36	122/15	122/45	123/14				
IGR05	011221	121/37	122/33	122/46	123/15				
IGR06	011222	122/47	124/47	124/49	125/15				
IGR0A	011211	121/09	121/53	122/10	122/29	122/38	123/09		
IGR0B	011212	121/10	121/54	122/39					
IGR0C	011213	121/11	121/55	122/11	122/30	122/40	123/10		
IGR0D	011214	121/12	121/56	122/41					
IGTBI	000103	15/45	16/45						
IGTDC	000105	15/47	16/47						
IGTOK	000104	15/46	16/46						
IGTSC	000106	15/48	16/48						
IGTTX	000107	15/49	16/49						
IHAAI	000050	15/12	16/25						
IHALT	000114	15/54	16/54						
ILL	001067	29/04	29/05	29/06	29/07	29/08	29/10	29/11	29/12
		29/14							
ILLEG	005370	76/31	77/36	77/39	77/48	81/45	86/30		
ILLGR	005367	76/13	76/15	76/30					
ILOOP	000113	15/53	16/53						
ILORE	000122	15/60	16/60						
IMEND	002312	42/26	42/40	42/51					
IMESS	000040	15/04	16/17						
IMSPR	010226	108/60	108/62						
IMULT	000066	15/26	16/39						
INDAD	003153	52/12	52/27						
INDW1	003122	51/60							
INDW2	003350	54/13							
INFLG	007614	100/52	101/28	101/58					
INHSW	006604	90/08	90/35	91/06					
INIMK	000125	17/12	108/43	108/58					
INIO1	010222	108/45	108/57						
INNUK	005320	75/09	75/50						
INRET	005423	76/38	76/41	76/42	76/48	76/61	77/04	81/50	86/31
INSAD	003145	52/07	52/20						
INSAV	007607	101/19	101/20	101/21	101/22	101/24	101/40	101/41	101/43
		101/44	101/50	101/53					
INSCO	007505	99/32	99/43	99/62					
INSPO	007504	99/29	99/40	99/41	99/53	99/61			
INSTA	003132	52/08	52/09						
INSTB	003133	52/09	52/10						
INSTR	003123	52/02	53/15						
INSW1	003121	51/59							
INSW2	003347	54/12							
INTAD	007547	100/43	100/59						
INTER	005401	76/41	78/13	81/51	86/32				
INTOC	000237	19/06	100/46	101/42	139/23	139/41	139/60	141/02	164/32
		165/04							
INTOU	007575	101/31	101/42						
INTS1	007572	101/35	101/39						
INTSV	007550	100/59	101/19						
INTTI	005302	75/08	75/34	75/41					
INXWS	003225	51/07	53/11	53/29					
ICMSK	007506	99/42	99/63						
ICPDR	011227	121/07	121/50	122/52					
IPADM	011225	121/40	121/59	122/18	122/35	122/50	123/18	124/51	125/17
IPASS	000123	15/61	16/61						

0181 ,MAIN

IQUES	000071	15/29	16/42						
IRESA	000077	15/40	15/43						
IRESW	000073	15/31	16/44						
ISAMS	000072	15/30	16/43						
ISTAA	000115	15/55	16/55						
ISTAC	006545	88/62	90/03						
ISTAN	000116	15/56	16/56						
ISTAP	000121	15/59	16/59						
ISTAS	000120	15/58	16/58						
ISTAW	000117	15/57	16/57						
ISTP0	000110	15/50	16/50						
ISTP1	000111	15/51	16/51						
ISTP2	000112	15/52	16/52						
ISTPA	011223	121/08	121/51	122/07	122/27	122/48	123/07	124/43	125/10
ITBIN	000051	15/13	16/26						
ITDEC	000053	15/15	16/28						
ITIMS	000064	15/24	16/37						
ITIRQ	000065	15/25	16/38						
ITISK	000063	15/23	16/36						
ITOCT	000052	15/14	16/27						
ITR	006530	88/54	89/21	89/53	90/19				
ITRAG	006527	88/42	88/47	88/52	89/20	89/52			
ITRCT	006531	88/55	89/12	89/22	89/54	90/20			
ITRP1	006525	88/41	88/48	89/50					
ITRP2	006526	88/46	88/51	88/53	89/51	90/17			
ITYPE	000042	15/06	16/19						
ITZOC	000054	15/16	16/29						
IWAIT	000061	15/21	16/34						
IWAOP	000062	15/22	16/35						
JMEND	001730	38/21	42/41						
KCP0	002450	45/30							
KCP1	002451	45/31							
KCP10	002460	45/38							
KCP11	002461	45/39							
KCP12	002462	45/40							
KCP13	002463	45/41							
KCP14	002464	45/42							
KCP15	002465	45/43							
KCP16	002466	45/44							
KCP17	002467	45/45							
KCP2	002452	45/32							
KCP20	002470	45/46							
KCP21	002471	45/47							
KCP22	002472	45/48							
KCP23	002473	45/49							
KCP24	002474	45/50							
KCP25	002475	45/51							
KCP3	002453	45/33							
KCP4	002454	45/34							
KCP5	002455	45/35							
KCP6	002456	45/36							
KCP7	002457	45/37							
KEYA	003354	54/19	54/24	54/27					
KEYB	003362	54/21	54/25						
KEYS	003351	52/40	54/15						
KINC	002476	45/19	45/22	45/53					
KINDI	002447	45/15	45/28						
KSTAC	001372	33/28	33/60	67/08					
LALOC	001373	33/61	34/14	34/36					

0182 ,MAIN

LAPRG	001374	33/62	34/16						
LARA1	012735	144/15	144/25	144/27					
LARAN	012077	105/59	105/61	106/17	106/42				
LAST1	005707	80/06	80/19	80/42	80/44	80/54			
LAST2	006043	81/06	81/21	82/04	82/17	82/27			
LAST3	006152	83/06	83/21	84/04	84/17	84/25			
LAST4	005572	77/06	77/22	78/16	78/50	79/03			
LAST5	006330	85/06	85/19	85/39	86/04	86/22			
LASTN	005426	76/11	76/17	77/00	78/50	80/42	82/27	84/25	86/22
LERE1	005702	80/25	80/30	80/32	80/35	80/36			
LERE2	005772	81/27	81/37	81/38					
LERE3	006113	83/27	83/37	83/38					
LERE4	005503	77/28	77/49	77/50					
LERE5	006215	85/25	85/29	85/32	85/33				
LETE1	005667	80/14	80/25						
LETE2	005757	81/14	81/27						
LETE3	006100	83/14	83/27						
LETE4	005455	77/15	77/28						
LETE5	006205	85/14	85/25						
LLPTT	002257	41/59	43/56						
LMADR	005067	68/34	68/45	68/58	70/38	70/42			
LMSK	001163	28/27	30/27						
LOADA	002337	42/56	43/01						
LOADB	002314	41/28	42/46						
LOADF	002336	42/54	43/00						
LOADR	002327	42/57	42/63						
LOCKA	003415	54/31	55/02						
LOOP	006113	16/53	130/24	132/14	132/20	132/28	132/36	132/44	132/52
		132/59	133/07	133/14	133/21	133/28	133/36	133/44	133/54
		134/01	134/24	135/21	135/58	136/15	136/58	137/10	137/18
		138/41	138/51	138/61	139/09	139/46	140/00	140/23	140/30
		140/42	141/11	141/38	143/31	145/12	147/55	148/24	148/55
		151/19	152/21	152/58	154/22	155/13	156/23	156/32	157/07
		158/30	159/21	160/31	162/23	163/62	165/26	167/12	
LOOPR	006542	88/40	88/45	88/50	89/00	89/43	89/63	107/34	
LOP39	011561	127/10	153/08	153/11					
LOP41	011640	128/12	155/20	155/23					
LSTSW	001366	33/14	33/18	33/21	33/25	33/33	33/44	33/56	34/28
MBILO	001510	34/42	34/44	35/10					
MCMEX	002264	42/16	43/31						
MCMNM	002273	42/21	42/25	42/48	43/24				
MCPUT	005002	50/43	50/45	67/27					
MCRLF	004743	38/55	67/18	76/44	76/47	90/39			
MD128	000215	18/46	156/43	158/45					
MDCTR	002700	48/31	48/55	49/00					
MDMNC	005113	69/08	70/07	70/08					
MDMMF	005120	69/10	70/18	70/19					
MELOC	001470	14/44	35/04						
MESCH	000434	24/16	24/21						
MESSA	000422	24/10	24/17						
MESTO	007772	104/47	104/61						
MESTR	007742	17/63	104/32						
MGEND	011766	130/10	130/13	130/57					
MGERR	011420	125/46	130/49	130/52					
MHEAD	006606	90/38	91/18						
MIMEX	002313	42/22	42/41	42/49	43/25	43/32			
MLLOC	001501	34/33	34/35	35/08	42/28	42/30			
MLOOP	002656	48/32	48/36						
MLORE	007071	93/55	93/57	94/50					

NTREP	003602	57/17	57/35				
NTRES	003631	57/15	57/22	57/29	57/36	57/40	
NTRET	003627	57/16	57/37	57/38			
NTTYP	003576	56/16	57/13				
NUK	000034	16/10	75/45	75/51			
NUMB2	006044	81/08	82/03	82/18	82/23	82/28	82/34
NUMB3	006153	83/08	84/03	84/18	84/21	84/26	84/32
NUMB4	005571	77/09	78/06	78/25	78/28	78/49	79/19
NUMSC	005706	80/08	80/41	80/49	80/55	80/62	
NUTYP	003014	50/10	50/38				
NWAIT	002437	45/19	45/23				
NWSEC	010045	106/06	106/08	106/12			
NWTRA	010067	106/28	106/32				
NWTYP	003506	50/36	56/04				
NXDIS	000570	24/50	25/50	25/60	25/63		
NXINS	007463	99/40	99/49	99/54			
NXTYP	003526	56/20	56/24				
NYTYP	003636	57/33	57/47				
NZTYP	003645	57/49	57/52	57/55			
OCTAB	000651	26/18	26/54				
OF2CO	006042	82/10	82/26				
OF3CO	006151	84/10	84/24				
OFTDC	005574	77/17	78/02	79/02			
OFTE2	006014	81/16	82/02	82/30			
OFTE3	006125	83/16	84/02	84/28			
OFTRE	005642	79/30	79/36	79/41			
OFTSC	005710	80/15	80/44	80/58			
OFTSI	005637	79/11	79/14	79/38			
OFTTX	006221	85/15	85/39	86/35			
OKDIG	005776	81/20	81/43	81/57			
OMEND	002376	43/08	43/35				
ONTER	005403	76/39	76/43	78/03	81/52	86/33	
OPDRV	010742	119/02	122/52				
ORDIN	003224	52/23	52/58	52/60	53/10		
OUT5	006365	86/53	86/58				
PAADM	010724	118/40	122/50				
PASCO	000243	19/10					
PASSB	007061	94/13	94/16	94/40	118/04		
PASSC	007062	94/15	94/41				
PASSN	007063	94/21	94/22	94/42			
PASWB	010253	109/45	109/52				
PAT1S	010271	109/37	109/62				
PAT2S	010272	109/39	109/63				
PATCM	010231	17/61	109/12				
PATFC	010250	109/21	109/41				
PATFL	010241	17/47	109/33				
PATT1	010267	109/16	109/19	109/38	109/45	109/59	
PATT2	010270	109/17	109/18	109/40	109/48	109/60	
PBINC	001033	28/47	28/52				
PBINN	001037	28/35	28/39	28/51			
PBINR	001074	28/41	28/50	29/20	31/14	31/17	31/33
PCENT	006603	90/25	90/34				
PCH14	004732	66/30	67/04				
PCH35	004733	66/34	67/05				
PCH37	004734	66/38	67/06				
PCOTT	004731	66/25	67/03				
PCOUN	004730	66/26	66/28	67/02			
PDEC1	000613	25/21	26/20				
PDEC2	000521	25/13	25/17				

0186 ,MAIN

RCAL4	007660	102/49	102/56						
RCALI	007620	17/39	102/12						
RDSEC	006140	17/44	116/28						
RDSWI	010202	17/22	108/37						
RDSWS	006130	17/23	33/15	33/22					
RDWRI	006146	17/56	129/53	148/39	150/08	152/43	154/44	160/52	163/43
		166/52							
REAS1	010306	110/29	110/34						
REAS2	010322	110/44	110/49						
REASE	010273	17/43	110/20						
REAWR	010603	17/55	116/23						
REBIN	001400	14/46	34/08	168/02					
RECAL	006136	17/40	121/39	122/17	122/34	123/17	124/31	124/50	125/16
		129/25	141/15	141/42	146/38	148/19	149/49	152/20	154/21
		154/35	156/34	158/26	160/27	163/21	164/20	166/29	
RECSY	010602	115/48	115/63						
REDAG	010625	116/45	116/54						
REDOU	010636	116/34	116/57						
REERR	010642	116/30	116/58	116/62					
REG0	001100	27/37	27/45	27/50	27/54	27/63	29/24	30/02	30/10
		30/17							
REG1	001101	27/47	28/07	29/25	30/11	30/20			
REG2	001102	27/48	28/06	29/26	30/12	30/21			
REG3	001103	27/38	27/41	27/44	28/08	29/27	30/03	30/06	30/09
		30/22							
REOF2	006031	82/02	82/06	82/07	82/09	82/13	82/14	82/15	
REOF3	006142	84/02	84/06	84/07	84/09	84/13	84/14	84/15	
REPL4	005553	78/15	78/29	78/31	78/32				
REPL5	006327	86/03	86/10	86/19	86/20				
RERCO	000241	19/08	114/39	114/42	129/52	148/33	152/33	154/40	163/29
		166/16							
RESEC	010034	106/02	106/15						
RESET	006127	17/21	88/63	89/37					
RETS	006402	87/06	87/10	87/11					
RETOF	005644	79/02	79/05	79/06	79/08	79/41	79/42	79/44	
RETRA	010053	106/19	106/35						
RETU0	000170	18/19	118/08	118/30	118/40	118/46	119/02	119/10	119/12
		119/20	120/04	120/23	125/22	125/41	129/13	130/25	
RETUR	006536	89/02	89/48	89/59	90/42	90/47	90/48	91/22	91/44
RETYP	003010	50/08	50/31	50/33	50/34	50/41	50/51		
REVU	003162	52/39							
REVUA	003172	52/47	52/57						
REVUB	003207	52/51	52/60						
REVUC	003203	52/56	53/00						
REVUD	003175	52/50	52/54						
REWCO	000240	19/07	114/48	114/49	129/35	148/35	152/35	154/32	163/31
		166/17							
RINGB	010144	17/26	107/40						
RINHI	000452	24/37	26/02	26/06	26/07				
RLPTI	002405	43/45	43/53	43/59					
RMEND	002375	43/05	43/26	43/33	43/34				
RMSK	001161	28/20	28/25	30/25					
RNMES	010706	118/18	118/26						
ROUT1	010332	110/55	110/59						
ROUT2	010333	110/51	110/60						
RPASS	007064	93/46	93/49	93/59	94/12	94/14	94/30	94/44	
RPOUT	000462	24/02	24/04	24/06	24/07	24/19	24/45	24/48	24/53
		25/00	25/04	25/06	25/43	25/48	25/53	25/58	26/15
		26/17							

SWREG	000126	17/16	108/13	108/40	123/25	124/05
SXAST	006735	92/15	92/19	92/22		
SXRT0	010162	107/08	107/28	107/56		
SXRT1	010163	107/40	107/53	107/57		
SXRT2	010164	107/58	108/11	108/15		
SXRT3	010165	107/59	108/37	108/55		
SXSV0	010166	107/11	107/17	107/60		
SXSV1	010167	107/10	107/18	107/61		
SXSV2	010170	107/09	107/19	107/62		
SXSV3	010171	107/41	107/50	107/63		
SXSV4	010172	107/42	107/51	108/00		
SXSV5	010173	107/43	107/52	108/01		
SXSV6	010174	108/02	108/38	108/54		
SYNCE	010641	116/39	116/61			
TABLE	002000	30/24	39/06			
TACER	011407	125/24	125/44			
TBZOT	000777	27/49	28/18			
TCLDR	011011	119/16	119/26			
TCOU1	010670	117/32	117/40	117/47		
TCOU2	010667	117/25	117/46			
TCP0	002575	47/36				
TCP00	002574	47/34	48/06			
TCP1	002576	47/37				
TCP10	002605	47/44				
TCP11	002606	47/45				
TCP12	002607	47/46				
TCP13	002610	47/47				
TCP14	002611	47/48				
TCP15	002612	47/49				
TCP16	002613	47/50				
TCP17	002614	47/51				
TCP2	002577	47/38				
TCP20	002615	47/52				
TCP21	002616	47/53				
TCP22	002617	47/54				
TCP23	002620	47/55				
TCP24	002621	47/56				
TCP25	002622	47/57				
TCP3	002600	47/39				
TCP4	002601	47/40				
TCP5	002602	47/41				
TCP6	002603	47/42				
TCP7	002604	47/43				
TCSPA	011457	121/19	124/15	126/06		
TDACH	011063	120/06	120/26			
TEMP1	001575	36/11	36/17	36/36		
TEMP2	001576	36/21	36/29	36/35	36/37	37/18
TERBI	006160	84/30	84/32			
TERDC	005522	78/04	78/06			
TERM1	005530	78/09	78/12			
TERMB	006154	83/11	84/28			
TERMD	005516	77/12	78/02			
TERMO	006045	81/11	82/30			
TERMS	005724	80/11	80/58			
TERMT	005411	76/23	76/50	77/11	81/47	86/27
TERMX	006343	85/11	86/35			
TEROK	006051	82/32	82/34			
TERSC	005730	80/60	80/62			
TERTX	006347	86/37	86/39			

0190 .MAIN

TEXEN	006305	85/52							
TEXIA	006233	85/50	86/16	86/53	87/07				
TEXIN	006234	85/51							
TIDIS	011504	127/02							
TIMCI	002571	47/30	48/09	48/13					
TIMEM	002572	47/31	48/15	53/04					
TIMEX	002570	47/26	47/29	47/63	48/10				
TIMMS	006064	16/37	51/16						
TIMRO	006065	16/38	129/19	140/18	156/18	166/20			
TIMSK	006063	16/36	24/26	24/30	29/45	30/13	100/23	102/17	102/37
		103/19	103/40	107/12	110/25	110/40	111/20	111/31	112/02
		112/39	138/19	138/37	140/47	140/57	141/19	141/43	142/08
		142/22	142/45	142/60	143/09	146/39	146/49	148/20	149/50
		156/53	158/57	164/37	164/56				
TINCD	011622	124/13	124/19	128/08					
TINHI	001164	28/43	28/58	29/32	30/05	30/28	30/32	31/16	31/37
TISCD	011546	127/08							
TMEND	002340	35/00	42/14	43/05					
TNRUN	011461	118/19	126/08						
TOMA	003314	53/37	54/03						
TOMB	003337	53/47	54/04						
TOMC	003335	54/02	54/11						
TOMD	003327	53/45	53/48						
TOMER	003275	50/35	53/22						
TOMF	003331	53/50	53/53						
TOMG	003341	54/06	54/10						
TOMH	003342	54/04	54/07						
TOMJ	003332	53/48	53/51						
TOPDR	010764	119/06	119/22						
TOTCO	000242	19/09	104/37	114/31	114/32	124/41	125/08	129/30	130/16
		148/32	152/29	154/29	163/30				
TPRET	000167	18/17	106/59	107/02	117/21	117/46	117/47		
TRA37	013212	149/21	149/56						
TRAP1	007677	103/23	103/28						
TRAP2	007705	103/31	103/36						
TRAP3	007720	103/44	103/49						
TRAP4	007726	103/52	103/59						
TRAP5	007736	104/09							
TRAPU	007664	17/41	103/14						
TRCA1	010657	117/28	117/34						
TRCA2	010664	117/37	117/42						
TRCAL	010646	17/59	117/21						
TRIES	000255	19/27	112/32	114/43					
TRMES	006152	18/00	130/20	153/03	155/15				
TROEN	007135	95/33	95/45	95/57	95/60				
TROHA	007077	41/22	95/26						
TROHC	007066	94/46	95/30						
TROHL	007104	95/31	95/35						
TROLC	007067	94/47	95/42						
TROLL	007116	95/43	95/47						
TROLO	007111	41/24	95/38						
TRORC	007070	94/48	95/54						
TRORE	007123	41/26	95/50						
TRORL	007130	95/55	95/59						
TROTA	007065	94/45	95/28	95/40	95/52				
TROTB	007137	94/45	97/02						
TRPOS	006137	17/42	129/33	149/58	152/41	154/37	158/36	160/46	163/23
		163/39	163/41	164/31	166/31	166/48	166/50		
TRTYP	002762	50/12	50/19						

0192 .MAIN

WRIS6	010451	112/55	112/59						
WRIS7	010452	112/52	113/02						
WRISC	010364	17/49	111/60						
WRSEC	006143	17/50	129/43	160/51	163/33	166/40			
WTFUB	001265	32/22	32/29	32/31					
WTNOK	001253	32/21	32/33						
WTOK	001270	32/30	32/34						
WTORE	001305	32/16	32/46	32/47					
WTRIE	000256	19/31	114/50	148/34	152/34	154/31	158/51		
XBYTE	012170	135/13	136/06	136/11	136/34	136/50			
XC20	003227	52/26	52/45	53/13					
XCHAR	000664	15/05	27/02						
XCPN	002573	45/16	46/17	47/32	48/05				
XCRLF	001043	15/07	28/56						
XDBIN	001020	15/17	28/36						
XDDEC	000464	15/19	24/48						
XDICL	001165	15/10	30/30						
XDISP	001104	15/08	29/29						
XDIVD	002666	15/28	48/54						
XDIVS	002665	15/27	48/53						
XDLTE	006000	80/12	81/12	81/46	83/12				
XDOCT	000564	15/18	25/58						
XDOUT	001134	15/09	30/02						
XDZOC	000554	15/20	25/48						
XFDD	000061	17/06	17/07	100/25	102/19	102/35	102/39	103/21	103/39
		103/42	103/59	106/61	107/14	107/16	110/27	110/34	110/38
		110/42	110/57	111/22	111/29	111/33	112/04	112/19	112/25
		112/28	112/38	112/41	116/46	132/56	132/57	133/04	133/05
		133/12	133/18	133/19	133/26	133/33	133/41	135/17	135/33
		135/34	135/37	135/40	135/41	135/45	135/46	135/49	135/52
		135/55	136/04	136/05	136/07	136/10	136/21	136/22	136/35
		138/18	138/21	138/33	138/34	138/39	138/49	138/58	138/59
		139/06	139/07	139/34	140/35	140/49	140/56	140/59	141/17
		141/18	141/21	141/45	141/54	141/55	142/04	142/10	142/20
		142/21	142/24	142/27	142/47	142/56	142/57	142/62	143/07
		143/08	143/11	143/15	144/21	144/37	144/38	144/46	144/51
		146/41	146/48	146/51	146/53	148/22	149/52	156/27	156/42
		156/45	156/52	156/55	156/58	158/44	158/47	158/56	158/59
		159/02	164/21	164/39	164/50	164/54	164/55	164/58	
XFDIS	001776	38/39	39/00						
XFITY	001377	34/01	34/29						
XFORM	001000	27/51	28/20						
XFROM	005064	68/09	68/55	70/20					
XGTCH	006002	80/09	81/09	81/48	83/09				
XILLG	005777	79/23	79/25	79/27	79/33	79/35	79/40	80/34	80/51
		81/33	81/36	81/45	82/12	83/33	83/36	84/12	
XINRT	006004	80/04	81/04	81/50	83/04				
XINST	003231	51/05	53/15	53/27					
XINTR	006005	81/00	81/51	82/36	84/34				
XLCRE	007010	15/60	93/46						
XLPT	000017	16/12	24/28	27/52	27/53	107/22	107/23		
XLPTT	000441	24/27	24/29	38/62					
XMEND	001466	34/11	34/63						
XMESS	000412	15/04	24/02						
XMEXT	001467	34/10	35/00						
XMULT	002653	15/26	46/29						
XOMER	003011	50/15	50/35						
XONTR	006006	80/59	81/52	82/31	84/29				
XPASS	007026	15/61	94/12	118/03					

0193 .MAIN

XPCPT	001465	34/31	34/62						
XQUES	005211	15/29	71/19						
XRANK	003232	52/43	53/16	54/15					
XRESW	001312	15/31	33/10						
XRTC	000014	16/15	53/33	53/34	53/35	53/49	53/52	54/05	54/09
		56/08	56/09	56/10	56/12	56/14	57/18	57/20	107/26
		107/27							
XSAMS	001736	15/30	38/32						
XSDIS	001777	38/40	39/01						
XSTAA	006722	15/55	19/11	92/11					
XSTAC	001370	33/26	33/29	33/37	33/39	33/58			
XSTAN	006736	15/56	92/34						
XSTAP	006763	15/59	93/12						
XSTAS	007000	15/58	93/35						
XSTAW	006746	15/57	92/51						
XTBIN	001014	15/13	28/32						
XTDEC	000470	15/15	24/53						
XTIM1	002730	49/31	49/33						
XTIMA	002716	49/19	49/23						
XTIMC	002731	49/13	49/21	49/27	49/34				
XTIMD	002715	49/16	49/22	49/32					
XTIMR	002732	49/11	49/14	49/15	49/17	49/18	49/20	49/25	49/26
		49/29	49/35						
XTIMS	002702	15/25	49/11						
XTIMT	002722	49/24	49/27						
XTIMW	002725	49/28	49/30						
XTOCT	000560	15/14	25/53						
XTOIN	005066	68/27	68/57	70/31					
XTRMT	006001	80/10	81/10	81/47	83/10				
XTTI	000010	16/13	32/21	32/23	32/26	75/34	75/37	75/39	107/24
		107/25							
XTTO	000011	16/14	24/32	28/00	28/01	51/10	51/11	51/12	51/14
		51/15	51/17	51/26	51/27	51/28	51/35	51/43	51/46
		51/51	51/55	107/20	107/21	107/45	107/47		
XTTOT	000445	24/31	24/33	38/63					
XTXCO	005424	76/36	76/62						
XTXND	005425	76/37	76/63						
XTYME	003230	51/03	52/24	52/41	53/14	53/25			
XTYPE	000724	15/06	27/37						
XTYPN	006003	80/20	81/22	81/49	83/22				
XTZOC	000550	15/16	25/43						
XWAIT	002427	15/21	45/11						
XWTOP	001244	15/22	32/14						
XWTYP	003012	50/20	50/36						
XX16	003226	52/27	52/33	52/46	52/56	53/12			
XXLPT	001774	38/35	38/62						
XXTTO	001775	38/37	38/63						
YCHAR	000672	24/41	24/59	27/09	28/34	28/61	28/63		
YDICL	001172	30/36	31/32						
YDLTE	006335	85/12	86/28						
YDOUT	001142	24/42	28/38	29/53	30/09	30/38			
YGTCH	006333	85/09	86/26						
YHAAT	001226	31/18	31/41						
YILLG	006337	85/31	85/47	86/30					
YINRT	006340	85/04	86/31						
YINTR	006341	86/32	86/55						
YONTR	006342	86/33	86/36						
YPBIN	001024	28/33	28/37	28/40					
YPDEC	000502	24/49	24/54	25/02					

0194 .MAIN

YPOCT	000603	25/54	25/59	26/12	
YTAB3	000712	24/43	27/26		
YTRMT	006334	85/10	86/27		
YTYPE	000732	27/22	27/34	27/44	
YTYPN	006336	85/20	86/29		
YZOCT	000601	25/44	25/49	26/09	
ZCHAR	000456	24/23	24/41		
ZDOUT	000457	24/42	26/00		
ZSUPP	000771	26/20	26/26	26/36	28/12
ZTAB3	000460	24/43	24/62		