```
0001  .MAIN
01
02
03
04                                                    ;RCSL:   44-RT 1647
05                                                    ;AUTOR:  FLEMMING KAM
06                                                    ;EDITED: 78,01,20
07
08
09
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27                          ;     RC3600
28                          ; EXTENDED MEMORY TEST
29                          ;    REVISION 1
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53                  ;S-BINARY(I)  TAPE:   RCSL  44-RT 1648  (SELFSTART AUTOLOAD HEAD
54                  ;S-BINARY(II) TAPE:   RCSL  44-RT 1595  (SELFSTART AUTOLOAD HEAD
55
56                  ;ASCII SOURCE(I):     RCSL  44-RT 1649  TAPE 1-5
57
58                  ;BINARY(I)  CARD:     RCSL  44-RT 1650
59                  ;BINARY(II) CARD:     RCSL  44-RT 1596
60
61                  ;KEYWORDS:          RC3603, MEMORY TEST, MEMORY ADDRESS TEST.
62
63                  ;ABSTRACT:          EXTENDED MEMORY TEST IS A MAINTENANCE PROGRAM
64                  ;                   DESIGNED TO DETECT MALFUNCTIONS IN THE
65                  ;                   MEMORY.
66                  ;NOTE:              THIS PROGRAM TEST ALSO THE MEMORY EXTENSION,
67                  ;                   IF ANY.
68
```

```
| 0002  .MAIN
01
02                ;                                                  PAGE 1 OF 11
03                ;
04                ;0. INDEX
05                ;       0.1 REVISION STATUS              PAGE 1
06                ;       1.  ABSTRACT                     PAGE 1
07                ;       2.  MACHINE REQUIREMENTS         PAGE 1
08                ;       3.  OPERATING PROCEDURE,SURVEY.  PAGE 2
09                ;       4.  SWITCH SETTINGS AND INIMESS  PAGE 3
10                ;       5.  OPERATING PROCEDURE          PAGE 4
11                ;       6.  PROGRAM DESCRIPTION          PAGE 7
12                ;       7.  SOFTWARE SURVEY              PAGE 9
13                ;       8.  MESSAGES FROM THIS TEST      PAGE 11
14                ;
15
16
17
18
19                ;0.1 REVISION STATUS
20                ;       REVISION        DATE            CHANGE
21                ;           0           77.10.06    ORIGINAL VERSION.
22                ;           1           78.01.20    REORGANISATION OF MAIN LOOP.
23                ;                                   PAGE ZERO TEST, X-Y LOAD TEST
24                ;                                   AND THE MOVE FUNCTION ARE
25                ;                                   APPENDED.
26
27
28
29
30
31
32                ;1. ABSTARCT
33                ;       EXTENDED MEMORY TEST IS A STAND-ALONE MAINTENANCE PROGRA
34                ;       DESIGNED TO DETECT MALFUNCTIONS IN THE DATARAM CORE
35                ;       MEMORY, WITH OR WITHOUT MEMORY ADAPTER.
36                ;       THE TEST CONSIST OF TWO BINARY VERSIONS: I & II.
37
38                ;       BIN. VERSION I:
39                ;       THE PROGRAM CAN TEST THE MEMORY EXTENSION, IF ANY.
40                ;       WITH THE PROGRAM IS IT ALSO POSSIBLE TO TEST PAGE ZERO
41                ;       AND THE DOMAIN CONTAINING THE BINARY LOADER.
42                ;       AS A SPECIAL FEATURE IS THE PROGRAM ABEL TO PERFORM A
43                ;       THERMAL LOAD OF THE X-Y DRIVERS IN THE CORE MODULS.
44                ;       THE TEST DO NOT PERFORM A TEST OF THE AREA FROM 400
45                ;       (256) TO C. 12000 (5120), CONTANING THE PROGRAM.
46                ;       IF THE SYSTEM HAVE TWO MEMORY MODULS IS IT STILL
47                ;       POSSIBLE TO TEST THE HOLE MEMORY, SEE SECTION 5.0.
48
49                ;       BINARY VERSION II:
50                ;       THIS PROGRAM IS A RELOCATED VERSION OF NO. I, BUT
51                ;       ONLY WITH THE MAIN TEST LOOP. IT IS ABEL TO TEST
52                ;       LOC. 400 TO 20000 (8K).
53
54
55
56
57
58
59                ;2. MACHINE REQUIREMENTS
60                ;       RC 3600 FAMILY PROCESSOR                    CPU
61                ;       MINIMUM 16K CORE MEMORY: MEM 809/711
62                ;                                MEM 717/719        MEM
63                ;       TELETYPE OR OPERATORS CONTROL PANEL         TTY/OCP *NOTE
64                ;       LINEPRINTER, OPTIONALLY                     LPT     *NOTE
65                ;
66                ;       *NOTE: IF THIS DEVICES IS SET ONLINE AFTER PROGRAM
67                ;       START, THE PROGRAM MUST BE RESTARTED TO GET OUTPUT.
```

```
| 0003  .MAIN
01                ;                                                  PAGE 2 OF 11
02
03
04                ;    3.  OPERATING PROCEDURE, SURVEY.
05                ;
06                ;
07                ;
08                ;
09                ;        START OF PROGRAM AFTER LOADING:
10                ;        -------------------------------
11                ;     SET SWITCHES TO ZERO.
12                ;     IF SPECIAL CONTROL IS WANTED, SEE SECTION 4.2.
13                ;
14                ;        STARTING ADDRESSES:
15                ;        -------------------
16                ;     400  START OF MAIN MEMORY TEST.
17                ;     401  PAGE ZERO AND DOMAIN OF BIN. LOADER TEST.
18                ;     402  X-Y DRIVER LOAD TEST, NOT FOR NOVA TYPES.
19                ;     403  RESTART OF TEST, NO QUESTIONS, LAST ANSWERS.
20                ;     404  MOVE THE PROGRAM TO THE OTHER MEMORY MODUL.
21                ;
22                ;        AUXILIARY ROUTINES, START ADDRESSES:
23                ;
24                ;    1400  RESTART PROGRAM AS WHEN LOADED.
25                ;    2202  GET A NEW PRINTER ALPHABETH (SEE 4.1.1).
26                ;    2204  SET TO 48/64K WORDS MODE, MEM SIZE ?
27                ;    2206  SET TO 32K WORDS MODE, MEM SIZE ?
28                ;    2210  EXAMINE MEMORY.
29                ;    2212  DEPOSIT MEMORY.
30                ;    2214  TROUBLE BREAKPOINTS HALT.
31                ;    2216  TROUBLE BREAKPOINTS LOOP REPORT.
32                ;    2220  TROUBLE BREAKPOINT RESET.
33                ;    2222  START BINARY LOADER, READ PROM PTR/TTI (SW 0).
34                ;
35                ;        ERROR MESSAGES:
36                ;        ---------------
37                ;        037500     037503      073000
38                ;          (AC0)      (AC1)       (AC2)
39                ;        PC 007464 (AC3)
40                ;
41                ;          AC0: CORRECT PATTERN.
42                ;          AC1: READ PATTERN.
43                ;          AC2: MEMORY ADDRESS.
44                ;          PC:  PROGRAM ADDRESS.
45                ;
46                ;        ERROR LOOPING:
47                ;        --------------
48                ;        IN CASE OF ERROR IS IT POSSIBLE TO TRIG THE SCOPE
49                ;        AT THE P-LINE. (1003 B33).
50                ;
51                ;
52
```

```
| 0004  .MAIN
01                ;                                                  PAGE 3 OF 11
02                ;4. SWITCH SETTINGS AND INITIAL MESSAGES:
03                ;       4.1 STARTING ADDRESSES - METHOD 4.3 START :
04                ;               400 START OF MEMORY TEST
05                ;               401 PAGE ZERO AND DOMAIN OF BIN. LOADER TEST.
06                ;               402 X-Y DRIVER LOAD TEST.
07                ;               403 RESTART OF TEST, NO QUESTIONS, LAST ANSWERS.
08                ;               404 MOVE THE PROGRAM TO THE OTHER MEM MODUL.
09                ;              1400 RESTART PROGRAM AS WHEN LOADED.
10                ;              2202 GET A NEW PRINTER ALPHABETH (SEE 4.1.1)
11                ;              2204 SET TO 48/64K WORDS MODE, MEM SIZE ?
12                ;              2206 SET TO 32K WORDS MODE, MEM SIZE ?
13                ;              2210 EXAMINE MEMORY
14                ;              2212 DEPOSIT MEMORY
15                ;              2214 TROUBLE BREAKPOINT HALT
16                ;              2216 TROUBLE BREAKPOINT LOOP REPORT
17                ;              2220 TROUBLE BREAKPOINT RESET
18                ;              2222 START BINARY LOADER, READ FROM PTR/TTI (SW 0
19                ;
20                ;       4.2 CONTROL BY SWITCH SETTING, THE STATE REPORTED
21                ;                    AT TELETYPE AND LINEPRINTER
22                ;               SW0:  LOOP IN TEST IN CASE OF ERROR.
23                ;               SW10: INHIBIT PRINTOUT IN TESTLOOP-PROGRAM.
24                ;               SW11: PRINT FAILURE RATE IN TESTLOOP-PROGRAM.
25                ;               SW12: NO HALT IN CASE OF ERROR IN TESTLOOP-PROGR
26                ;               SW13: WAIT AFTER DIS MESSAGE.  DIS IS THE 16 CHA
27                ;               SW14: HALT AFTER DIS MESSAGE.  DISPLAY AT OPERAT
28                ;               SW15: CLEAR DIS AFTER MESSAGE. CONTROL PANEL. OC
29                ;
30                ;       4.3 START
31                ;               OF PROGRAM AFTER LOADING:
32                ;               SET SWITCHES TO CONTROL, (4.2).
33                ;               ANSWER START ADDRESS QUESTION.
34                ;
35                ;       4.3 A RESTART
36                ;               OF PROGRAM THE BEST WAY TO INSURE
37                ;               CORRECT SWITCH SETTINGS:
38                ;               RESET
39                ;               SET SWITCHES TO START ADDR, (4.1).
40                ;               EXAMINE
41                ;               SET SWITCHES TO CONTROL, (4.2).
42                ;               CONTINUE.
43                ;               THE START ADDR IS LISTED AT TTO, LPT AND DIS.
44                ;
45                ;       4.3 B RESTART
46                ;               OF PROGRAM IF RC 3603 CPU:
47                ;               SET DATASWITCH 0, 10 - 15 UP (1)
48                ;               SET RESET PARITY ERROR DOWN (ON)
49                ;               PRESS AUTOLOAD AND RELEASE
50                ;               SET RESET PARITY ERROR UP (OFF)
51                ;               SET SWITCHES TO CONTROL (4.2).
52                ;               ANSWER THE QUESTION WRITTEN ON TTY/OCP.
53                ;
54                ;       4.3 C POWER RESTART
55                ;               OF PROGRAM ONLY IF POWER MONITOR OPTION:
56                ;               SET KEY IN LOCK BEFORE REMOVING POWER
57                ;               AFTER POWERING UP THE PROGRAM WRITES:
58                ;               POWER
59                ;               EXT. MEMORY TEST
60                ;               SET SWITCHES TO CONTROL, (4.2).
61                ;               ANSWER START ADDRESS QUESTION.
```

```
I 0005  .MAIN
01                ;                                                  PAGE 4 OF 11
02                ;
03                ;       4.4 LOADING MESSAGE:
04                ;               AFTER LOADING THE PROGRAM WILL WRITE FOLLOWING:
05                ;               SWITCHES: 100073 ;THIS IS THE INITIAL STATE OF S
06                ;               CPU TYPE: 000016 ;THIS IDENTIFIES WHICH CPU/MEM
07                ;               LAST LOC: 077777 ;THIS INFORMS THE USER: MEM-SIZ
08                ;               BINARY LOADER OK ;THIS IS THE TS VERSION PLACED
09                ;               AT LAST LOC (MAX 32K) BUT SLIGHTLY MODIFIED AND
10                ;               INCL BOOTSTRAP.                  ERRORHALT XX7752
11                ;               IF NOT SELFSTARTING PROG:        READYHALT XX7676
12                ;               ACTUAL PROG NAME ;THIS IS THE LOADED PROGRAM.
13                ;               SET SWITCHES TO CONTROL, (4.2), STARTADDR   400
14                ;
15                ;       4.5 START WITH BREAK OPTION (RC 3603).
16                ;               RESET
17                ;               SET SWITCHES TO HALT INSTR       063077
18                ;               SET REGISTER SELECT TO 6
19                ;               DEPOSIT INTO REGISTER
20                ;               SET SWITCHES TO ADDRESS          000003
21                ;               SET REGISTER SELECT TO 5
22                ;               DEPOSIT INTO REGISTER
23                ;               SET BREAK SWITCH TO ON
24                ;               GO TO 4.3 RESTART WITH SA = 1400
25                ;
26                ;       4.6 CPUNO
27                ;               FOR TIMING PURPOSE THE PROGRAM DETERMINES IN
28                ;               WHICH CPU IT RESIDUES. IF IT FAILS THE PROGRAM
29                ;               WILL TRY TO CONTINUE AFTER THE MESSAGE
30                ;               "MISERABLE TIMING". IF IMPOSSIBLE THE PROGRAM
31                ;               WILL ASK YOU TO IDENTIFY THE CPU WITH A NUMBER
32                ;               BETWEEN 0 AND 6. USE 2 FOR NOVA1200, RC3603-BREA
33                ;                                    4 FOR RC3603
34                ;                                    5 FOR NOVA 2-16K
35                ;                                    6 FOR NOVA 2-8K WHERE
36                ;               THE MEMORY TYPE FOR THE FIRST 8K IS RELEVANT:
37                ;               RESET
38                ;               DEPOSIT # INTO AC2
39                ;               SET SWITCHES TO CONTROL (4.2).
40                ;               CONTINUE.
41                ;               ...OR CORRECT PROGRAM, READ MESS IN END OF LOCKA
42                ;
43                ;5. OPERATING PROCEDURE
44                ;       LOAD THE PROGRAM AND ANSWER THE QUESTIONS GIVEN. IF THE
45                ;       ANSWER IS BEYOND LIMITS THE QUESTION IS REPEATED. IF THE
46                ;       SUGGESTED ANSWER IS OK, ANSWER ONLY NL (RETURN). THE LAS
47                ;       DIGIT ANSWERED CAN BE REMOVED WITH RUBOUT, ELSE IF YOU
48                ;       TYPED WRONG THE QUESTION IS REPEATED BY GIVING DIGITS
49                ;       UNTIL LIMIT IS PASSED.
50                ;
51                ;       THE PROGRAM ASK FOR FOLLOWING PARAMETERS:
52                ;       A. MAIN TEST
53                ;               FIRST TEST LOCATION.
54                ;               LAST TEST LOCATION.
55                ;               MEMORY TYPE:
56                ;                 NOVA 2 AND MEM 717/718 = 0
57                ;                 MEM 709/711            = 1
58
59                ;       B. PAGE ZERO AND DOMAIN OF BIN. LOADER TEST.
60                ;               NONE.
61                ;
62                ;       C. X-Y DRIVER LOAD TEST.
63                ;               MEMORY TYPE: SEE POINT A ABOVE.
64                ;               MEMORY MODUL SIZE 32KWORD (MEM 709/717)
65                ;                                 OR 16KWORD (MEM 711/718)
66                ;               IF 16KWORD THE PROG NEED FOLLOWING TOO:
67                ;               PERFORM LOAD IN LOWER MODUL OR
68                ;               IN UPPER MEMORY MODUL.
```

```
| 0006  .MAIN
01                ;                                                  PAGE 5 OF 11
02
03                ;       A FULL TEST OF A CORE MEMORY SYSTEM CAN BE:
04                ;
05                ;       1 X 16KW   2 X 16KW   1 X 32KW   2 X 32KW
06                ;       ---------------------------------------
07                ;         C/         C/C          C          C | C
08                ;         A           A           A          D | A
09                ;         B           B           B          C | B
10                ;         E         D | E         E          A | E
11                ;                   A |                      B | D
12                ;                                        E | D | C
13                ;                                            A | A
14                ;                                              | B
15                ;                                              | E
16                ;
17                ;       THE MEANING OF THE LETTERS ARE:
18                ;
19                ;               A: MAIN TEST                          SA. 400
20
21                ;               B: PAGE ZERO TEST.                    SA. 401
22                ;               C: X-Y DRIVER LOAD TEST(LOWER/
23                ;                                      UPPER)         SA. 402
24                ;               D: MOVE PROG. TO THE OTHER
25                ;                  MODUL, CHANGE SELECT SWITHES       SA. 403
26                ;               E: RUN THE RELOCATED PROGRAM,
27                ;                  RCSL: 44-RT 1595, SEE 7.2          SA. 20400
28
29                ;
30                ;
31                ;
32                ;
33
34
35                ;       IN CASE OF ERROR THE PROGRAM WILL PRINT THE PC AND AC'S,
36                ;       THEN THE FURTHER FLOW WILL DEPEND OF THE SWITCHES.
37                ;       A PASS MESSAGE IS PRINTED WITH C. 10 MINUTES INTERVAL,
38                ;       DEPEND OF CPU TYPE AND PARAMETERS.
39
40
41                ;       IF THE PROGRAM DESTROYES ITSELF OR IN ANOTHER WAY
42                ;       DISBEHAVE THERE IS A SUPERVISING METHOD BUILT IN: ALL
43                ;       LOOPS ARE STARTED WITH JMP .+1. BY HAND IT IS POSSIBLE TO
44                ;       SHORTEN THE PROGR., THEREBY NOT RUN ALL LOOPS BY CHANGING
45                ;       TO HALT OR JMP @RTEST (CAUSES PASS MESSAGE F. LOOPS UP TO
46                ;       THIS COMMAND). AUTOMATICALLY IT'S POSSIBLE TO INSERT HALT
47                ;       TO INSERT A CALL TO A ROUTINE, WHICH PRINTS PC LOOP ADDR,
48                ;       TO INSERT BACK THE JMP .+1 IN THE BREAKPOINTS CHOOSEN BY
49                ;       THE PROGRAMMER (TABLE TROTB) BY CALLING STARTADDR 2214,
50                ;       2216, 2220.
51
52                ;       THE PROGRAM CONTAINS AN ASCII PRINTER TABLE. IF THE
53                ;       PRINTER HAS ANOTHER DRUM ALPHABETH READ FOLLOWING:
54
55
```

```
| 0007  ,MAIN
01                ;                                                  PAGE 6 OF 11
02
03                ;5.1 PRINTER ALPHABETH CHANGE:
04                ;AT ANY TIME AFTER LOADING THIS PROGRAM IT IS POSSIBLE TO CHANGE
05                ;THE ALPHABETH USED ON THE LINEPRINTER. THERE ARE 2 METHODS:
06                ;5.1.1                    GET ONE OF THE BUILD-IN ALPHABETHS:
07                ;       START PROGRAM IN SA 2202
08                ;       REMEMBER SWITCHES TO CONTROL. (4.2).
09                ;       ANSWER ALPHABETH #, SEE EXISTING BELOW
10                ;THE PROGRAM WILL RESTART AFTER CHANGING THE ALPHABETH.
11                ;       ANSWER NEXT START ADDRESS.
12                ;5.1.2                          GET AN -ADD ON- TAPE WITH
13                ;A NEW ALPHABETH AND LOAD THIS UPON THE PROGRAM AND RESTART. THE
14                ;TAPES WITH PRINTER TABLE ALPHABETH ARE DESCRIBED BELOW:
15                ;EXISTING: #1   44-RT 535       ASCII
16                ;          #2   44-RT 529       RC STANDARD TYPE 71/78 STARTING
17                ;          #3   44-RT 532       RC STANDARD TYPE 71/78 STARTING
18                ;                                        (SKEWED 4 POSITIONS)
19                ;          #4   44-RT 1213      PL 1, TYPE 70
20                ;
21                ;HOW TO PRODUCE A NEW TABLE:
22                ;THE TABLE HAS 200 OKTAL (128 DECIMAL) BOXES. INPUT KEY
23                ;IS THE ASCII VALUE OF THE CHARACTER TO PRINT ADDED TO 2000.
24                ;THE RESULT IS ADDRESS OF A BOX. EACH BOX OCCUPIES A CORE WORD.
25                ;IT IS BUILT UP OF TWO FIGURES. THE FIRST IS THE CLASS OF THE
26                ;CHARACTER TO BE PRINTED: 0 FOR PRINT, 6 FOR BLIND. THE SECOND
27                ;IS THE CHARACTER VALUE AT THE PRESENT PRINT DRUM. BELOW
28                ;VALUE 40 OCTAL FOLLOWING CHARACTERS MAY BE USED: 11 TAB,
29                ;12 LF, 14 FF AND 15 CR. ALL OTHERS BELOW 40 WILL GIVE SPACE.
30                ;THE FIRST BOX SHOULD CONTAIN THE VALUE FOR THE NULL CHAR
31                ;AND THE LAST THE VALUE FOR THE DEL CHAR, WHICH BOTH NORMALLY
32                ;ARE BLIND. IF YOU COUNT 0,1,2, ,7,10, , THE BOX 101 SHALL
33                ;CONTAIN THE PRINT DRUM VALUE FOR AN A. IF THE DRUM DO NOT
34                ;HAVE SMALL LETTERS, FILL IN THE VALUE FOR BIG ONES. NOW PUNCH
35                ;AN ASCII TAPE LIKE THIS:
36                ;    .LOC 2000
37                ;    .RDX 8          ;WHICH RDX YOU WANT
38                ;    .TXTE?
39                ;    <6><0>          ;(2000) FIRST BOX, BOX 0
40                ;    <6><0>
41                ;    .
42                ;    .
43                ;    .
44                ;    <0><101>        ;(2101) BOX 101 FOR A. FOR ASCII DRUM
45                ;    .               ;101 IS USED, FOR TYPE 71 137 IS USED.
46                ;    .
47                ;    .
48                ;    <6><0>?         ;(2177) BOX 177, LAST
49                ;    .RDX 8
50                ;    .END 101
51                ;PRODUCE A BINARY TAPE AND LOAD THIS TO MEMORY WITH
52                ;BINARY LOADER AFTER LOADING OF MAIN PROGRAM.
53
54
55                ;IN THE RELOCATED VERSION (RCSL  44-RT 1595) OF THE PROGRAM,
56                ;IT IS NECESSARY TO START IN SA. 20,000 + 404 , AFTER
57                ;LOADING THE NEW PRINTER ALPHABET.
58
```

```
| 0008  .MAIN
01                      ;                                                PAGE 7 OF 11
02                      ;6. PROGRAM DESCRIPTION
03                      ;       6.1 TESTLOOP FAILURE RATE
04                      ;       6.2 STRUCTURE OF PROGRAM, NEXT PAGE.
05                      ;               6.1 TESTLOOP FAILURE RATE:
06                      ;THERE ARE TWO DIFFERENT WAYS TO USE THE ROUTINES FOR TESTLOOP:
07                      ;SINGLE OR MULTIPLE ERRORHALT:
08
09                      ;SINGLE:        SETP1       MULTIPLE:   SETP1
10                      ;               ERRORHALT               ERRORHALT
11                      ;               LOOP                    ERRORHALT
12                      ;                                       ERRORHALT
13                      ;                                       LOOP
14
15                      ;IN CASE OF A CONSTANT ERROR THE RATE WILL BE PRINTED
16                      ;THIS WAY:
17                      ;PC XXXXXX 100 %                        PC XXXXXX 300 %
18                      ;WERE THE LAST IS A MULTIPLE OF 100 %.
19
20                      ;THERE ARE THREE POSSIBILITIES FOR THE NUMBER OF LOOPS IN A
21                      ;CYCLE, I. E. HOW MANY TIMES THE PROGRAM RUNS THROUGH THE
22                      ;INSTRUCTIONS BETWEEN SETP1 AND LOOP. IF SWITCH 0 IS 0 FIRST
23                      ;TIME AN ERROR IS DETECTED THE PROGRAM PROCEEDS TO NEXT
24                      ;INSTRUCTION AFTER LOOP.
25
26                      ;NUMBER OF LOOPS:       SETP0:  1
27                      ;                       SETP1:  10
28                      ;                       SETP2:  100
29
30                      ;IF THE ERROR IS NOT CONSTANTLY IT IS POSSIBLE TO SEE THESE
31                      ;FAILURE RATES IF SINGLE OPERATION:
32
33                      ;SETP0: 100 % ERROR IN THE ONE LOOP.
34                      ;SETP1: 100,50,33,25,20,16,14,12,11,10 % FOR ERROR IN
35                      ;         1, 2, 3, 4, 5, 6, 7, 8, 9, 10 -TH LOOP.
36                      ;SETP2: LIKE SETP1, ONLY ADD 9...1 % FOR ERROR IN
37                      ;        11...100 -TH LOOP.
38
39                      ;IF MULTIPLE OPERATION THE RATES DEPENDS ON HOW MANY OF THE
40                      ;ERRORS ARE FOUND IN THE FIRST LOOP WITH ERROR:
41                      ;       100 % COULD BE  1 ERROR IN FIRST LOOP
42                      ;               OR      2 ERRORS IN SECOND LOOP
43                      ;               OR      3 ERRORS IN THIRD LOOP ETC.
44
45                      ;IF SWITCH 0 IS 1 THE PROGRAM WILL REMAIN IN THE LOOP WITH
46                      ;ERROR, FIRST TIME AN ERROR IS SEEN THE PROGRAM WILL HALT
47                      ;(IF NOT SWITCH 12), THEN THE PROGRAM CONTINUES UNTILL ALL
48                      ;NUMBER OF LOOPS ARE PERFORMED. THEN A NEW CYCLE IS
49                      ;ENTERED CALLED ERRORCYCLE. AFTER EACH ERRORCYCLE A NEW IS
50                      ;STARTED UNTILL SWITCH 0 IS SET TO 0.
51
52                      ;IN ALL CYCLES (FIRST OR ERROR) THE PC (PROGRAM COUNTER OF ERROR
53                      ;IS WRITTEN FIRST TIME AN ERROR IS SEEN AND IN MULTIPLE OPERATIO
54                      ;MORE THAN ONE ERRORHALT COULD WRITE THE PC. BUT ONLY WITHIN
55                      ;THAT FIRST LOOP WITH ERROR. THE FAILURE RATE IS PRINTED WHEN
56                      ;THE CYCLE IS FINISHED. THE PROGRAM ONLY HALTS IN THE FIRST CYCL
57                      ;(DEPENDING ON SW 12). THE ERRORCYCLES HAVE ANOTHER AMOUNT OF
58                      ;LOOPS THAN FIRST CYCLE:
59
60                      ;NUMBER OF LOOPS:       FIRST CYCLE     ERROR CYCLE
61                      ;               SETP0:  1               10
62                      ;               SETP1:  10              100
63                      ;               SETP2:  100             100
64
65                      ;IN SINGLE OPERATION FAILURE RATE IS TELLING HOW OFTEN THE ERROR
66                      ;APPEARS. IN MULTIPLE OPERATION FAILURE RATE SHOULD BE USED
67                      ;VERY CAREFULLY.
```

```
;            6.2 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 WHA
;   WAS CAUSING THE ERRORHALT. THE PROGRAM INCLUDES ONLY FEW
;   SUBROUTINES TO MAKE IT EASYER FOR THE OPERATOR.
;   THE PROGRAM ALSO CONTAINS NORMAL TESTLOOP FACILITIES, SE
;   6.1. 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 MEMORY WITH AID OF FOLLOWING
;   TEST LOOPS:


;   BINARY VERSION I.


;   A. MAIN TEST ROUTINE. SA 400

;      B0  ADDRESS SELECTION TEST
;      B1  CHECK READ ADDRESS
;      B3  COMPLEMENT ADDRESS TEST.
;      B4  BIT SELECTION TEST (FLOATINT ONE AND ZERO).
;      B10 FILL MEMORY WITH ALTANATE:
;              10101...10
;              010101..01
;          PERFORM THE TEST AGAIN WITH THE COMPLEMENT PATT.
;      B35 FILL MEMORY WITH "WORST CASE PATTERN"
;      B60 CHECK READ THE PATTERN
;          PERFORM THE TEST AGAIN WITH THE COMPLEMENT PATT.
;      B91 FAST BIT SELECTION TEST,
;          WITHOUT THE POSSIBLITY TO SETUP ERROR LOOPS.
;     B210 TEST ACCES TO MEM <=32KW VIA AUTO INC/DEC LOCATION

;   B. PAGE ZERO AND DOMAIN OF BIN. LOADER TEST. SA 401

;      E10 ADDRESS SELECTION TEST
;      E30 BIT SELECTION TEST

;
;   C. X-Y DRIVER LOAD TEST. SA 402

;      AGAIN  THE LOAD IS PERFORMED BY A JSR  . ,
;             IN THE SELECTED DRIVER ADDRESS.

;
;   BINARY VERSION II.
;
;   A. MAIN TEST ROUTINE. SA 20400

;      EQUIVALENT TO TEST LOOP A IN VERSION I.
;      (SEE SECTION 7.2).
```

```
I 2010  .MAIN
01                ;
02                ;7. SOFTWARE SURVEY
03                ;
04                ; 7.1 BINARY VERSION I.
05                ;
06                ;       EXTENDED MEMORY TEST, VERSION I, IS DIVIDED UP IN 4
07                ;       PARTS DESCRIBED IN THE FOLLOWING SUBSECTIONS.
08                ;
09                ; 7.1.1 MAIN TEST LOOP - SA 400
10                ;       THIS LOOP TEST THE MEMORY FROM FIRST TO LAST, EXCLUSIVE
11                ;       THE LOCATIONS CONTAINING THE BINARY LOADER, THE
12                ;       UPPER 200 LOCATIONES LESSER THAN 32K. THE
13                ;       OPERATOR HAVE TO TYPE IN, ON THE CONSOL (TTY/OCP)
14                ;       THE PARAMETERS FIRST, LAST AND THE MEMORY TYPE.
15                ;       (THE PROGRAM ASK FOR THEM).
16                ;       THE PARAMETERS SHALL MEET FOLLOWING CONDITIONS:
17                ;               FIRST > LAST PROG. LOC (=XXXX)
18                ;               LAST  > FIRST
19                ;               MEMORY TYPE: DR 103 OR DR 106/NOVE 2
20                ;                            DR 103: MEM 709/711
21                ;                            DR 106: MEM 717/718/NOVE 2
22                ;
23                ;       THE TEST IS PERFORMED BY MEANS OF FOLLOWING LOOPS:
24                ;
25                ;       1. MEMORY(ADDRESS):=ADDRESS; ADDRESS SELECTION TEST
26                ;       2. BIT SELECTION TEST,EQUIVALENT TO FLOATING ZERO/ONE TE
27                ;       3. COMPLEMENT ADDRESS TEST:
28                ;          MEM(C):= -,C;
29                ;       4. MEMORY(.):=1010...10/0101..01; ALTERNATE PATT. TEST
30                ;       5. WORST CASE AND COMPLEMENT WORST CASE TEST.
31                ;          THIS TEST PROUDUCE WORST CASE NOICE CONDITIONS TO THE
32                ;          SENSE/INHIBIT WIRES AT THE MEM 709/711 OR
33                ;          MEM 717/718 , EQUIVALENT TO NOVA-TYPES.
34                ;
35                ;          THE WORST CASE PATTERN IS GENERATED IN FOLLOWING WAY:
36                ;          DR 103:
37                ;           IF ADRBIT(3)=ADRBIT(15) THEN MEM:= -1
38                ;                                   ELSE MEM:= 0;
39                ;          DR 106:
40                ;           IF ADRBIT(7)=ADRBIT(11) THEN MEM:= -1
41                ;                                   ELSE MEM:= 0;
42                ;          THE '-1' AND '0' ABOVE IS RESPECTIVELY A
43                ;          FLOATING 0 AND 1. IN THIS WAY IS THE POSSIBLITY
44                ;          TO CATCH A ADDRESS SELECTION ERROR BIGGER.
45                ;          IN THE COMPLEMENT TEST IS THE COMPLEMENT PATTERN
46                ;          GENERATED.
47                ;
48                ;       6. FAST BIT SELECTION TEST.
49                ;          THE TEST IS THE SAME AS 2, EXCEPT THAT THE
50                ;          MEMORY IS FILLED WITH WORSE CASE PATTERN AND IT
51                ;          IS NOT POSSIBLE TO PERFORM ERROR LOOPS.
52                ;
53                ;       7. ACCES VIA AUTO INC/DEC LOCATIONS.
54                ;          THE MEMORY <=32K WORD IS FILLED WITH THE COMP.
55                ;          ADDRESS VIA ALL THE AUTI INC/DEC LOC.
56                ;          THE CONT. OF BOTH MEM AND AUTO LOC ARE TESTED.
57                ;
58                ;  7.1.2 PAGE ZERO AND DOMAIN OF BIN. LOADER TEST - SA 401
59                ;       THE LOOP CHECK LOC 0 - 377 (PAGE ZERO) AND X7600 -
60                ;       X7777 (THE BINARY LOADER).
61                ;       THE TEST IS PERFORMED WITH AID OF:
62                ;       1. ADDRESS SELECTION TEST
63                ;       2. BIT SELECTION TEST.
64                ;
65                ; *NOTE THAT IN THIS TEST IS IT ONLY POSSIBLE TO STOP THE PROGRA
66                ;       WITHOUT DESTROYING IT, BY STRIKE A KEY AT TTY OR OCP.
67
```

```
! 0011  .MAIN
01                ;                                                  PAGE 10 OF 11
02                ;  7.1.3 X-Y DRIVER LOAD TEST - SA 402, NOT NOVE TYPES.
03                ;       THE PROGRAM NEED FOLLOWING PARAMETERS:
04                ;
05                ;         MEMORY TYPE.
06                ;         32KWORD MODUL OR    (MEM 709/717)
07                ;         16KWORD MODUL.      (MEM 711/718)
08                ;         IF 16 KWORD MODUL THE PROGRAM ALSO ASK FOR
09                ;         UPPER OR LOWER MODUL.
10                ;
11                ;                                        PAGE 10 OF 11
12
13                ;       THIS TEST CHECKS AND HEAT ALL THE X AND Y DRIVERS
14                ;       AND THE SELECTION DIODES IN THE DATARAM CORE MEMORY
15                ;       MODULS BY MEANS OF 16 DRIVER ADDRESSES (SEE PAGE  106 IN
16                ;       PROG. LIST). IN ALL THE DRIVER ADRESSES A JSR  . IS
17                ;       PERFORMED IN: 40, 80, 160, .. MSEC.
18                ;
19                ;
20                ;
21                ;  7.1.4 PROGRAM DISPLACEMENT - SA 404
22                ;       THE PROGRAM MOVE THE EXTENDED MEMORY TEST TO
23                ;       THE OTHER MEMORY MODUL, IF ANY:
24                ;               2 X 16KWORD MOD. TO ADDRESS 16 - 22 K
25                ;               2 X 32KWORD MOD. TO ADDRESS 32 - 38 K
26                ;
27                ;       AFTER THE MOVE THE PROGRAM WRITES:
28                ;         SET SWITCHES TO CONTROL, (4,2), START ADDR 400 ?
29                ;       TURN THE POWER OFF AND INTERCHANGE THE ADDRESS
30                ;       SELECTION SWITCHES AT THE MEMORY MODULS:
31                ;               16KW: 0 == 2
32                ;               32KW: 0 == 4
33                ;       TURN THE POWER ON AND MAKE A RESTART SEE SEC 4.3.
34
35                ;
36                ; 7.2 BINARY VERSION II.
37                ;
38                ; 7.2.1 MAIN TEST LOOP - SA 20400
39                ;       THE TEST IS EQUIVALENT TO 6.1.1, EXCEPT THAT ALL THE
40                ;       PROGRAM ADDRESES HAVE TO BE ADD WITH 20000 (8K) AND THE
41                ;       PARAMETERS SHALL MEET FOLLOWING CONDITIONS:
42                ;         FIRST > 377 (255)
43                ;         LAST  > FIRST
44                ;         LAST  < 20000 (8K).
45                ;       CONSEQUENTLY IS THIS LISTNING ALSO VALID FOR THE
46                ;       BINARY VERSION II. (LISTNING II: RCSL 44-RT 1651).
47                ;
48                ; 7.2.2 MOVE ROUTINE - SA 20404
49                ;       THE ROUTINE MOVE A LOADED PRINTER TABLE TO THE RIGHT
50                ;       PLACE IN THE RELOCATED PROGRAM.
```

```
| 0012  .MAIN
01                ;                                                PAGE 11 OF 11
02
03                ;8. MESSAGES FROM THIS TEST
04                ;       ALL MESSAGES ARE WRITTEN ON TTY, LPT AND THE 16 CHAR DIS
05                ;       ON OPERATORS CONTROL PANEL. IT IS POSSIBLE TO ANSWER
06                ;       QUESTIONS AT TTY OR NUK, NUMERIC KEYBOARD ON OPERATORS
07                ;       CONTROL PANEL.
08                ;       ABOUT THE NUMBERS:
09                ;               0-5 DIGITS IS A DECIMAL NUMBER
10                ;                   RANGE -32768 TO -1 AND 0 TO 32767
11                ;                 6 DIGITS IS AN OCTAL NUMBER
12                ;               0-6 DIGITS IS AN OCTAL NUMBER WITH
13                ;                   LEADING ZEROES SUPPRESSED, DON'T USE.
14                ;                 8 DIGITS IS A BINARY NUMBER.
15                ;
16                ;       MESSAGES:       LPT/TTY & DIS, IF SPEC DIS MESSAGE: ()
17                ;
18                ;       MISERABLE TIMING, RUN RC 3600 INSTRUCTION TIMER TEST
19                ;       (MISERABLE TIMING)      TIMING WITH BIG TOLERANCES, CPU
20                ;                               TYPE IS GUESSED.
21                ;       RTC IS UNSTABLE,        RTC NOT RUNNING WITH CONSTANT
22                ;                               SPEED.
23                ;       SET CPUNO > AC2         HELP THE PROGRAM TO IDENTIFY CPU
24                ;                               SEE 4,5
25                ;       SWITCHES: 100073        NO MESS AT DIS ABOUT SW. POSITIO
26                ;       LAST LOC. 077777
27                ;       PROG NAME               IDENTIFIKATION
28                ;       SET SWITCHES TO CONTROL, (2,2), STARTADDR   400 ?
29
30                ;       SWITCHES: 000000        NO MESS AT DIS ABOUT SW. POSITIO
31                ;       000400   STARTADDR
32                ;
33                ;  MAIN AND PAGE ZERO TEST.
34                ;
35                ;       1, PASS OF  5 RUNS      WITH/WITHOUT ERRORS
36                ;       (1,PASS  5 R)
37                ;
38                ;       037500  037503  000000  ERROR MESSAGE
39                ;       AC0     AC1     AC2
40
41                ;           THE AC'S CONTAIN THE FOLLOWING:
42                ;               AC0: CORRECT PATTERN
43                ;               AC1: ERROR PATTERN
44                ;               AC2: MEMORY ADDRESS
45                ;               AC3: PROGRAM ADDRESS (PC)
46                ;
47                ;
48                ;       PC 007464           100 %
49                ;
50                ;       (PC 007464   100 %)     EXAMINE FOR AC'S AFTER HALT.
51
52                ;       SWITCHES: 170000        NO MESS AT DIS ABOUT SW. POSITIO
53                ;                               PRINT INHIBIT SWITCH SET.
54                ;
55                ;
56                ;  X-Y DRIVER LOAD TEST.
57                ;
58                ;       DELAY  40 MSEC
59                ;       DRIVER ADDRESS NO: 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,1
60                ;       DELAY  80 MSEC
61                ;       DRIVER ADDRESS NO: 1,2,3, ....
62                ;       ........
63                ;
64                ;       TO EATCH NUMBER CORRESPOND A DRIVER ADDRESS,
65                ;       GIVEN AT PAGE 101 IN THE PROG. LISTNIG.
66
```

```
| 0013  .MAIN
01
02                ;                    PAGE ZERO FOR TAPE 2,3,4,5
03
04        000000 .LOC 0
05
06 00000 003142          2*MELOC          ;MESS AFTER RDOS LOAD AND STORE PC ON IN
07 00001 011343          INTPT            ;ADDR OF INTR. SERVICE ROUTINE
08 00002 001400          REBIN            ;SELFSTART ADDR FOR RDOS ETC.
09 00003 000000          0                ;0=HALT, 1=SELFSTART PROG AFTER REBIN
10 00004 000000          0                ;#ADDR FOR SELFSTART PROG AFTER REBIN
11 00005 000000          0                ;FOR LOAD RDOS, USED BY POW. INTR, FITYP
12
13        000020 .LOC 20
14
15 00020 000000 IDX0:    0                ;AUTO INCREMENT LOCATION
16 00021 000000 IDX1:    0                ;AUTO INCREMENT LOCATION
17 00022 000000 IDX2:    0                ;AUTO INCREMENT LOCATION
18 00023 000000 IDX3:    0                ;AUTO INCREMENT LOCATION
19 00024 000000 IDX4:    0                ;AUTO INCREMEMT LOCATION
20 00025 000000 IDX5:    0                ;AUTO INCREMEMT LOCATION
21 00026 000000 IDX6:    0                ;AUTO INCREMEMT LOCATION
22 00027 000000 IDX7:    0                ;AUTO INCREMEMT LOCATION
23 00030 000000 DDX0:    0                ;AUTO DECREMENT LOCATION
24 00031 000000 DDX1:    0                ;AUTO DECREMEMT LOCATION
25 00032 000000 DDX2:    0                ;AUTO DECREMEMT LOCATION
26 00033 000000 DDX3:    0                ;AUTO DECREMEMT LOCATION
27 00034 000000 DDX4:    0                ;AUTO DECREMEMT LOCATION
28 00035 000000 DDX5:    0                ;AUTO DECREMEMT LOCATION
29 00036 000000 DDX6:    0                ;AUTO DECREMEMT LOCATION
30 00037 000000 DDX7:    0                ;AUTO DECREMEMT LOCATION
```

```
| 0014   .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 001727 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:  POWON            ;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 006052 IGTBI:  GETBI
46  00104 005731 IGTOK:  GETOK
47  00105 005426 IGTDC:  GETDC
48  00106 005643 IGTSC:  GETSC
49  00107 006161 IGTTX:  GETTX
50  00110 006416 ISTP0:  ENTP0
51  00111 006423 ISTP1:  ENTP1
52  00112 006430 ISTP2:  ENTP2
53  00113 006447 ILOOP:  CYCLE
54  00114 006617 IHALT:  ERROR
55  00115 006720 ISTAA:  XSTAA
56  00116 006734 ISTAN:  XSTAN
57  00117 006744 ISTAW:  XSTAW
58  00120 006776 ISTAS:  XSTAS
59  00121 006761 ISTAP:  XSTAP
60  00122 007006 ILORE:  XLORE
61  00123 007024 IPASS:  XPASS
62  00124 007233 IDIST:  XDIST
```

```
| 0015  .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          000017 DEV=XLPT                    ;DEFINE DEVICE TO TEST FOR STATUS
17
18          006040 CMESS=JSR        @IMESS     ;EACH DEFINITION BELOW CORRESPONDS
19          006041 CCHAR=JSR        @ICHAR     ;WITH A CALL OF A ROUTINE.
20          006042 CTYPE=JSR        @ITYPE
21          006043 CCRLF=JSR        @ICRLF
22          006044 CDISP=JSR        @IDISP
23          006045 CDOUT=JSR        @IDOUT
24          006046 CDICL=JSR        @IDICL
25          006047 CDATT=JSR        @IDATT
26          006050 CHAAT=JSR        @IHAAT
27          006051 CTBIN=JSR        @ITBIN
28          006052 CTOCT=JSR        @ITOCT
29          006053 CTDEC=JSR        @ITDEC
30          006054 CTZOC=JSR        @ITZOC
31          006055 CDBIN=JSR        @IDBIN
32          006056 CDOCT=JSR        @IDOCT
33          006057 CDDEC=JSR        @IDDEC
34          006060 CDZOC=JSR        @IDZOC
35          006061 CWAIT=JSR        @IWAIT
36          006062 WATOP=JSR        @IWAOP
37          006063 TIMSK=JSR        @ITISK
38          006064 TIMMS=JSR        @ITIMS
39          006065 TIMRO=JSR        @ITIRO
40          006066 MULTI=JSR        @IMULT
41          006067 DIVIS=JSR        @IDIVS
42          006070 DIVID=JSR        @IDIVD
43          006071 CQUES=JSR        @IQUES
44          006072 CSAMS=JSR        @ISAMS
45          006073 CRESW=JSR        @IRESW
46          006103 CGTBI=JSR        @IGTBI
47          006104 CGTOK=JSR        @IGTOK
48          006105 CGTDC=JSR        @IGTDC
49          006106 CGTSC=JSR        @IGTSC
50          006107 CGTTX=JSR        @IGTTX
51          006110 SETP0=JSR        @ISTP0
52          006111 SETP1=JSR        @ISTP1
53          006112 SETP2=JSR        @ISTP2
54          006113 LOOP=JSR         @ILOOP
55          006114 EHALT=JSR        @IHALT
56          006115 STATA=JSR        @ISTAA
57          006116 STATN=JSR        @ISTAN
58          006117 STATW=JSR        @ISTAW
59          006120 STATS=JSR        @ISTAS
60          006121 STATP=JSR        @ISTAP
61          006122 CLORE=JSR        @ILORE
62          006123 CPASS=JSR        @IPASS
63          006124 DISTB=JSR        @ICIST
```

```
| 0016  .MAIN
01
02 00125 000000 FIRST:  0
03 00126 000000 LAST:   0
04 00127 000000 LPOS:   0
05 00130 000000 RETES:  0
06 00131 000000 CTR:    0
07 00132 000000 FLAG:   0
08 00133 000000 FIPOS:  0
09 00134 000000 BMASK:  0
10 00135 000020 C20:    20
11 00136 010565 XPBASE: PBASE
12 00137 010565 XPPBASE: PBASE
13 00140 010606 XPCBASE: PCBASE
14 00141 000000 RTEST:  0               ;RETURN ADDR FROM TESTLOOPS
15
16       000400 .LOC 400
17
18 00400 006405         JSR     @IMTEST ;MEMORY TEST
19 00401 006406         JSR     @ISTES  ;PAGE ZERO AND BIN. LOADER TEST.
20 00402 006404         JSR     @IDRIV  ;X-Y DRIVER LOAD TEST.
21 00403 006405         JSR     @IANSW  ;RESTART, NO QUESTIONS, LAST ANSW.
22 00404 006405         JSR     @IMOVE  ;MOVE PROG TO THE OTHER MEM. MODUL.
23
24 00405 007201 IMTEST: XMTEST
25 00406 010627 IDRIV:  XDRIV
26 00407 011371 ISTES:  XSTES
27 00410 007212 IANSW:  XANSW
28 00411 011570 IMOVE:  XMOVE
29
30              ;THE STANDARD ROUTINES ARE NOT MODIFIED.
31              ;IF MODIFICATIONS, LIST HERE:
32              ;
33              ;       NUMBER OF RUNS IN A PASS IS CHANGED FROM
34              ;       10 TO 5.
35              ;       IN THE SETUP AND LOOP SUBROUTINE IS THE
36              ;       IORST, I/O RESET, REPLACED BY NIOP  0.
37              ;       THE PRINTOUT UNDER LOADING OF PROGRAM
38              ;       IS REDUCED.
39
40
41              ;TAPE 1
42
43              .EOT
```

```
  0017  .MAIN
01
02                ;TAPE 2          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                ;DCL     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
```

```
| 0018  .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 SIG
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
```

```
| 0019  .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/PCCT FINISH RETURN, TYPE TAB
63 00500 024753 EXDIS:  LDA     1,SADIG ;DISPLAY FINISHED, RESTORE PARAM
64 00501 002761         JMP     @RPOUT  ;EXIT
```

```
I 0020   .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
```

```
| 0021  .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 ROU)
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 SIMU
65 00663 000240 CHSPA:  240             ;A SPACE FOR + IN PDEC
```

```
| 0022  .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     XFCRM
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
```

```
| 0023  .MAIN
01
02 00760 152400 TYPE5:  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
33 01014 054461 XTBIN:  STA     3,CRBIR
34 01015 004407         JSR     YPBIN
35 01016 004654         JSR     YCHAR   ;TYPE DIGIT
36 01017 000420         JMP     PBINN   ;NEXT DIGIT
37
38 01020 054455 XDBIN:  STA     3,CRBIR
39 01021 004403         JSR     YPBIN
40 01022 004520         JSR     YDOUT   ;DISPLAY DIGIT
41 01023 000414         JMP     PBINN   ;NEXT DIGIT
42
43 01024 044447 YPBIN:  STA     1,SABIN ;SAVE PARAM FOR REPEAT
44 01025 054447         STA     3,PBINR
45 01026 030447         LDA     2,CRBIR ;SPEC RETURN IF PRINT SW
46 01027 006535         JSR     @TINHI
47 01030 002445         JMP     @CRBIR  ;NO PRINT
48 01031 030741         LDA     2,NN10  ;8 TIMES
49 01032 125300         MOVS    1,1
50 01033 020735 PBINC:  LDA     0,CHAR0
51 01034 125102         MOVL    1,1,SZC
52 01035 101400         INC     0,0     ;AC0:="CHAR1"
53 01036 002436         JMP     @PBINR  ;OUTPUT DIGIT
54 01037 151404 PBINN:  INC     2,2,SZR
55 01040 000773         JMP     PBINC
56 01041 024432         LDA     1,SABIN ;RESTORE PARAM
57 01042 002433         JMP     @CRBIR
58
59 01043 054432 XCRLF:  STA     3,CRBIR ;SAVE RETURN
60 01044 171000         MOV     3,2     ;SPEC RETURN IF PRINT SW
61 01045 006517         JSR     @TINHI
62 01046 002427         JMP     @CRBIR  ;NO TYPE
63 01047 020427         LDA     0,CHCR
64 01050 004622         JSR     YCHAR
65 01051 020426         LDA     0,CHLF
66 01052 004620         JSR     YCHAR   ;PRINT CR,LF
67 01053 002422         JMP     @CRBIR  ;EXIT
```

```
| 0024  .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      YDCUT   ;DISPLAY
```

```
| 0025  .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 0026  .MAIN
01
02                      ;RC 3600, ATTENTION DISPLAY OUTPUT
03                      ;BY MEANS OF ACOUSTIC 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 006061         CWAIT
45 01232 001240         SECM5
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 DELA
53 01240 000062 SECM5:  50
54       000010         .RDX 8
55 01241 000002 SECS2:  2
56 01242 000004 SECS4:  4
57 01243 004000 SEC4K:  4000
```

```
| 0027  .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 007167         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 0028  .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 070477         READS   2       ;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 001236         SEC3
21 01325 024441         LDA     1,LSTSW
22 01326 070477         READS   2
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 001513         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
```

```
| 0029  .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 006456         JSR     @XMEXT  ;TRY TO SET MEM EXT FLAG (RC3603 ONLY)
11 01403 006454         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 030525         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                ;**     JSR     @XPCPT  ;PRINT CPU TYPE
32 01427 006044         CDISP
33 01430 001472         MLLOC
34 01431 006040         CMESS
35 01432 001472         MLLOC           ;<15><12>LAST LOC: XX7777
36 01433 024740         LDA     1,LALOC
37 01434 006052         CTOCT
38 01435 006056         CDOCT
39 01436 006061         CWAIT           ;WAIT 3 SEC TO READ MESS. DON'T
40 01437 001236         SEC3            ;USE CDATT AS SW AREN'T SET.
41                ;**     CDISP
42                ;**     MBILO
43                ;**     CMESS
44                ;**     MBILO           ;<15><12>BINARY LOADER OK
45                ;**     CWAIT
46                ;**     SEC3
47 01440 006046         CDICL
48 01441 006044         CDISP
49 01442 007167         PROG
50 01443 006043         CCRLF
51 01444 006040         CMESS
52 01445 007167         PROG            ;ACTUAL PROGRAM NAME
53 01446 006061         CWAIT
54 01447 001236         SEC3
55 01450 062701         DICP    0,1     ;TRY TO SET MEM EXT FLAG (RC3603 ONLY)
56 01451 020003         LDA     0,3     ;CELL 3 = 0 FOR HALT
57 01452 101004         MOV     0,0,SZR ;CELL 3 = 1 FOR
58 01453 002004         JMP     @4      ;SELFSTART ADDR 4
59 01454 002401         JMP     @.+1    ;CHOOSE YOUR OWN START ADDR,
60 01455 004740         SWISA           ;NORMAL HALT, BUT IF NO SWITCH PANEL ?
61
62 01456 003016 XPCPT:  PRTYP
63 01457 001707 XMEND:  GMEND
64 01460 002340 XMEXT:  TMEND
```

```
I 0030  .MAIN
01
02          000001          .TXTM 1          ;RDOS TEXT PACKING MODE
03
04                  MELOC:  .TXT !LOADING UNIT OFF! ;"LOADING UNIT OFF"
    01461 046117
    01462 040504
    01463 044516
    01464 043440
    01465 052516
    01466 044524
    01467 020117
    01470 043106
    01471 000000
05
06          000000          .TXTM 0          ;NORMAL TEXT PACKING MODE
07
08                  MLLOC:  .TXT !<15><12>LAST LOC. !        ;"<15><12>LAST LOC. "
    01472 005015
    01473 040514
    01474 052123
    01475 046040
    01476 041517
    01477 020056
    01500 000000
09
10                  MBILO:  .TXT !<15><12>BINARY LOADER OK!
    01501 005015
    01502 044502
    01503 040516
    01504 054522
    01505 046040
    01506 040517
    01507 042504
    01510 020122
    01511 045517
    01512 000000
11                          ;"<15><12>BINARY LOADER OK"
12
13                  MSWRG:  .TXT !SWITCHES: !        ;"SWITCHES: "
    01513 053523
    01514 052111
    01515 044103
    01516 051505
    01517 020072
    01520 000000
14
15                  MSAMS:  .TXT ! STARTADDR!        ;" STARTADDR"
    01521 051440
    01522 040524
    01523 052122
    01524 042101
    01525 051104
    01526 000000
16
17                  UNTIM:  .TXT !MISERABLE TIMTNG! ;"MISERABLE TIMING"
    01527 044515
    01530 042523
    01531 040522
    01532 046102
    01533 020105
    01534 044524
    01535 044515
    01536 043516
    01537 000000
```

```
| 0031  .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 01540 001541 BINAD:  .+1               ;ADDR OF BIN LOADER DATA
10
11                ;SUBROUTINE TO BUILD A WORD IN AC2
12 01541 054425 BUILD:  STA     3,TEMP1 ;SAVE RETURN
13 01542 004406         JSR     GTCHR   ;GET FIRST BYTE
14 01543 171300         MOVS    3,2     ;PUT INTO LH OF AC2
15 01544 004404         JSR     GTCHR   ;GET NEXT BYTE
16 01545 173300         ADDS    3,2     ;FORM WORD IN AC2
17 01546 143000         ADD     2,0     ;ADD INTO CHECKSUM
18 01547 002417         JMP     @TEMP1  ;AND RETURN
19
20                ;READ A BYTE INTO AC3
21                ;IF SWITCH0=0 USE TELETYPE ELSE USE PTR
22 01550 054417 GTCHR:  STA     3,TEMP2 ;SAVE RETURN
23 01551 034417         LDA     3,SAVE  ;TEST WHICH DEVICE
24 01552 175103         MOVL    3,3,SNC
25 01553 000406         JMP     GTTTI   ;TTI
26 01554 063612         SKPDN   PTR     ;PTR
27 01555 000777         JMP     .-1
28 01556 074412         DIA     3,PTR   ;READ AND START
29 01557 060112         NIOS    PTR
30 01560 002407         JMP     @TEMP2  ;AND RETURN
31
32 01561 063610 GTTTI:  SKPDN   TTI
33 01562 000777         JMP     .-1
34 01563 074410         DIA     3,TTI   ;READ AND START
35 01564 060110         NIOS    TTI
36 01565 002402         JMP     @TEMP2  ;AND RETURN
37 01566 000000 TEMP1:  0
38 01567 000000 TEMP2:  0
39 01570 000000 SAVE:   0
40
41                ;TEST BLOCK TYPE
42 01571 125224 BTEST:  MOVZR   1,1,SZR ;1=START BLOCK (.END XX)
43 01572 000411         JMP     IGNOR   ;NO, IGNORE BLOCK
44 01573 101004         MOV     0,0,SZR ;TEST THE CHECKSUM
45 01574 000461         JMP     CHKER   ;ERROR
46 01575 030505         LDA     2,ADDRS ;GET ADDR
47 01576 062677         ICRST           ;DO A RESET
48 01577 151113         MOVL#   2,2,SNC ;TEST BIT 0
49 01600 001000         JMP     0,2     ;0=START PROGRAM
50 01601 063077         HALT            ;1=HALT
51 01602 000777         JMP     .-1     ;DON'T PROCEED
52
53                ;IGNORE BLOCK
54 01603 004745 IGNOR:  JSR     GTCHR   ;READ UNTIL AN ALL
55 01604 020404         LDA     0,BC377 ;ONES BYTE IS SEEN
56 01605 116404         SUB     0,3,SZR ;IGNORING ERROR MESS
57 01606 000775         JMP     IGNOR
58 01607 000407         JMP     BLOCK   ;OK, GO INTO BLOCK MODE
59 01610 000377 BC377:  377
```

```
 0032  .MAIN
01
02                    ;START OF PROGRAM BINARY LOADER TS
03 01611 062677 START:  IORST             ;RESET
04 01612 060477         READS   0         ;READ THE SWITCH REGISTER
05 01613 040755         STA     0,SAVE    ;AND SAVE IT FOR GTCHR
06 01614 060110         NIOS TTI          ;START BOTH READERS
07 01615 060112         NIOS PTR
08
09                    ;READ IN A BLOCK
10 01616 004732 BLOCK:  JSR     GTCHR     ;GET A BYTE
11 01617 171305         MOVS    3,2,SNR   ;AND TEST FOR NUL
12 01620 000776         JMP     BLOCK     ;YES, KEEP READING
13 01621 004727         JSR     GTCHR     ;OK, GET NEXT BYTE
14 01622 173300         ADDS    3,2       ;AND FORM COUNT. AC2:=# OF WORDS
15 01623 141000         MOV     2,0       ;SET CHECKSUM. AC0:=CHECKSUM TILL NOW
16 01624 145000         MOV     2,1       ;SET COUNTER
17 01625 004714         JSR     BUILD     ;GET ADDRESS
18 01626 050454         STA     2,ADDRS
19 01627 004712         JSR     BUILD     ;ADD IN THE CHECKSUM FROM TAPE
20 01630 125113         MOVL#   1,1,SNC   ;TEST BLOCK TYPE
21 01631 000740         JMP     BTEST     ;NOT A DATABLOCK
22 01632 044427         STA     1,COUNT   ;STORE WORD COUNT
23
24                    ;READ IN THE DATA BLOCK
25 01633 030734 DATA:   LDA     2,TEMP2   ;LAST STA IN TEMP2 WAS JSR RETURN
26 01634 034423         LDA     3,DIFF    ;ADDR 4 CELLS AFTER PROGRAM START:
27 01635 172400         SUB     3,2       ;AC2:=FIRST ADDR IN LOADER
28 01636 034444         LDA     3,ADDRS   ;ADDR IN WHICH TO STORE
29 01637 136400         SUB     1,3       ;ADD NEG WC TO CHECK SPACE
30 01640 172023         ADCZ    3,2,SNC   ;FOR WHOLE BLOCK
31 01641 000414         JMP     CHKER     ;NO, HALT THE LOADER
32 01642 030416         LDA     2,BC20    ;IF WC > 20 (OCTAL, NEG)
33 01643 147033         ADDZ#   2,1,SNC   ;IT IS A REPEAT BLOCK
34 01644 010415         ISZ     COUNT     ;WHERE WC IS ONE LESS THAN COUNT
35 01645 147022         ADDZ    2,1,SZC   ;IF REPEAT BLOCK SKIP NEXT TO READ DATA
36 01646 125113 STORE:  MOVL#   1,1,SNC   ;DON'T READ IN NEW DATA IF REPEAT BLOCK
37 01647 004672         JSR     BUILD
38 01650 052432         STA     2,@ADDRS
39 01651 010431         ISZ     ADDRS     ;NEXT ADDR
40 01652 010407         ISZ     COUNT     ;TEST COUNT
41 01653 000773         JMP     STORE     ;MORE DATA
42 01654 101004         MOV     0,0,SZR   ;TEST CHECKSUM
43 01655 063077 CHKER:  HALT              ;ERROR IN CHECKSUM, AC0=VALUE
44 01656 000740         JMP     BLOCK     ;OK,GET NEXT BLOCK
45 01657 000004 DIFF:   4
46 01660 000020 BC20:   20                ;REPEAT BLOCKS HAVE WD>20
47 01661 000000 COUNT:  0
48
49                    ;BOOTSTRAP LOADER TS FOR PTR:   ENTER AT BSTRP
50 01662 126440 GET:    SUBC    1,1
51 01663 063612         SKPDN   12        ;10 FOR TTI
52 01664 000777         JMP     .-1
53 01665 060412         DIA     0,12      ;10 FOR TTI
54 01666 060112         NIOS    12        ;10 FOR TTI
55 01667 127100         ADDL    1,1
56 01670 127100         ADDL    1,1
57 01671 107003         ADD     0,1,SNC
58 01672 000771         JMP     GET+1
59 01673 001400         JMP     0,3
60 01674 060112 BSTRP:  NIOS    12        ;10 FOR TTI
61 01675 004765         JSR     GET
62 01676 044402         STA     1,.+2
63 01677 004763         JSR     GET
64 01700 000000         0
65 01701 000000         0                 ;FOR BOOTSTRAP
66 01702 000000 ADDRS:  0
67 01703 000706 BEND:   JMP     START     ;START OF BINARY LOADER
```

```
| 0033  .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                          ;
07                          ;CALL   JSR     GMEND
08                          ;       RETURN
09 01704 010000 DMEND:  10000           ;4K CONSTANT
10 01705 014000 EMEND:  14000           ;6K+1 ADDR
11 01706 004200 FMEND:  004200          ;2K+LOADER CONSTANT
12 01707 054451 GMEND:  STA     3,RSAMS
13 01710 030775         LDA     2,EMEND ;PASS 1 AND 2K SEGMENTS
14 01711 020773         LDA     0,DMEND
15 01712 035000 AMEND:  LDA     3,0,2   ;SAVE LOC CONTENT
16 01713 051000         STA     2,0,2   ;STORE INTO UPPER
17 01714 025000         LDA     1,0,2   ;GET IT BACK
18 01715 146404         SUB     2,1,SZR
19 01716 000405         JMP     BMEND
20 01717 055000         STA     3,0,2   ;RESTORE LOC CONTENT
21 01720 113020         ADDZ    0,2
22 01721 151013 JMEND:  MOV#    2,2,SNC ;CHECK LAST 64K/32K WITH L-SHIFT
23 01722 000770         JMP     AMEND
24 01723 024763 BMEND:  LDA     1,FMEND ;WENT TOO FAR
25 01724 132400         SUB     1,2     ;PROTECT LOADERS
26 01725 050074         STA     2,HMEND ;SET END MEM
27 01726 002432         JMP     @RSAMS
28
29                          ;ROUTINE TO PRINT WHICH STARTADDR SA OF MORE
30                          ;IS USED. INPUT:  SA IN AC1.
31                          ;CALL   CSAMS
32                          ;       RETURN
33 01727 054431 XSAMS:  STA     3,RSAMS
34 01730 044431         STA     1,SSAMS
35 01731 030431         LDA     2,CXLPT ;SET OUTPUT DEVICE ROUTINES
36 01732 052433         STA     2,@XXLPT ;TO WAIT FOR ONLINE DEVICES.
37 01733 030430         LDA     2,CXTTO
38 01734 052432         STA     2,@XXTTO
39 01735 030427         LDA     2,CDIS
40 01736 052431         STA     2,@XFDIS
41 01737 052431         STA     2,@XSDIS
42 01740 006046         CDICL
43 01741 006043         CCRLF
44 01742 024417         LDA     1,SSAMS
45 01743 152520         SUBZL   2,2     ;AC2:=1
46 01744 146400         SUB     2,1     ;SUBTRACT 1 FROM JSR ADDR
47 01745 006056         CDOCT
48 01746 006052         CTOCT           ;PRINT SA
49 01747 006044         CDISP
50 01750 001521         MSAMS
51 01751 006040         CMESS
52 01752 001521         MSAMS           ;XXXXXX STARTADDR
53 01753 006073         CRESW           ;ONLY FOR PRINT SWITCH SETTINGS IF NEW
54 01754 006040         CMESS           ;PRINT CR,LF WITH MESS TO WAIT
55 01755 004743         MCRLF           ;FOR DEVICE READY BEFORE IORST
56 01756 062677         IORST
57 01757 002401         JMP     @RSAMS
58 01760 000000 RSAMS:  0
59 01761 000000 SSAMS:  0
60 01762 001750 CXLPT:  1000.           ;TIMEOUT CONSTANTS FOR
61 01763 000454 CXTTO:  300.            ;OUTPUT DEVICES
62 01764 000050 CDIS:   40.
63 01765 000441 XXLPT:  XLPTT
64 01766 000445 XXTTO:  XTTOT
65 01767 001125 XFDIS:  FDIST
66 01770 001147 XSDIS:  SDIST
```

```
i 0034   .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 0035  .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>
```

```
: 0036   .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 007105          TROHA
23 02214 006777          JSR     @.-1    ;START TROUBLE BREAKPOINT HALT
24 02215 007117          TROLO
25 02216 006777          JSR     @.-1    ;START TROUBLE BREAKPOINT LOOP REPORT
26 02217 007131          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
33                       ;ROUTINE TO MOVE PRINTER TABLE.
34                       ;INPUT: TABLE # IN AC2
35                       ;CALL:  J3R     MOPTB
36
37 02230 054415 MOPTB:   STA     3,MOPTR ;SAVE RETIRE
38 02231 034417          LDA     3,MOGTT ;ADDR OF POINTER
39 02232 157000          ADD     2,3     ;AC3:=TABLE ENTRY
40 02233 031400          LDA     2,0,3   ;AC2:=ADDR OF TABLE, START
41 02234 024413          LDA     1,MOLAD ;LAST STORE ADDR
42 02235 034411          LDA     3,MOFID ;FIRST STORE ADDR
43 02236 021000 MOREP:   LDA     0,0,2
44 02237 041400          STA     0,0,3   ;MOVE CHAR
45 02240 166415          SUB#    3,1,SNR ;LAST CHAR ?
46 02241 002404          JMP     @MOPTR  ;YES, RETIRE
47 02242 151400          INC     2,2
48 02243 175400          INC     3,3     ;NEXT CHAR ADDR
49 02244 000772          JMP     MOREP   ;REPEAT MOVE CHAR
50 02245 000000 MOPTR:   0               ;RETURN ADDR
51 02246 002000 MOFID:   2000            ;FIRST STORE ADDR
52 02247 002200 MOLAD:   2200            ;LAST STORE ADDR
53 02250 002250 MOGTT:   .               ;GET TABLE ADDR
54 02251 003656          PTAB1           ;ASCII TABLE ADDR
55 02252 004057          PTAB2           ;RC STANDARD 71/78 START .
56 02253 004260          PTAB3           ;RC STANDARD 71/78 START @
57 02254 004461          PTAB4           ;PL1 TYPE 72 TABLE ADDR
58
59 02255 000004 ALPTT:   4               ;SUGGESTED ANSWER TABLE #
60 02256 000004 ULPTT:   4               ;UPPER LIMIT
61 02257 000001 LLPTT:   1               ;LOWER LIMIT
```

```
[ 0037  .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 IORST. 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
20 02265 165000 SAMNM:  MOV     3,1
21 02266 006072         CSAMS           ;START ADDR MESSAGE
22 02267 024404         LDA     1,MCMNM
23 02270 046423         STA     1,@MIMEX
24 02271 062677         IORST           ;RESET MEM EXT FLAG (RC3603 ONLY)
25 02272 000402         JMP     SAMMS
26 02273 151113 MCMNM:  MOVL#   2,2,SNC
27
28 02274 006416 SAMMS:  JSR     @IMEND  ;AC2=HMEND=XX7600
29 02275 006044         CDISP
30 02276 001472         MLLOC
31 02277 006040         CMESS
32 02300 001472         MLLOC           ;<15><12>LAST LOC. XX7777
33 02301 020074         LDA     0,HMEND
34 02302 026407         LDA     1,@SAMCO ;AC1:=177
35 02303 107000         ADD     0,1
36 02304 006056         CDOCT
37 02305 006052         CTOCT
38 02306 006047         CDATT
39 02307 002401         JMP     @.+1    ;PROGRAM FINISHED
40 02310 004740         SWISA           ;RESTART MAIN PROGRAM.
41 02311 001376 SAMCO:  BINLA
42 02312 001707 IMEND:  GMEND
43 02313 001721 MIMEX:  JMEND
44
45                      ;ROUTINE RESTORE BINARY LOADER AND
46                      ;START LOADING PTR/TTI DEPENDING ON SWITCH 0.
47
48 02314 165000 LOADB:  MOV     3,1
49 02315 006072         CSAMS           ;START ADDR MESSAGE
50 02316 024755         LDA     1,MCMNM
51 02317 046774         STA     1,@MIMEX ;MAX 32K MODE
52 02320 062677         IORST           ;RESET MEM EXT FLAG
53 02321 006771         JSR     @IMEND  ;AC2=HMEND=0X7600
54 02322 026767         LDA     1,@SAMCC ;AC1:=177
55 02323 147000         ADD     2,1     ;LAST LOC = 0X7777
56 02324 036412         LDA     3,@LOADF ;FIRST ADDR LOADER
57 02325 137400         AND     1,3     ;IN THIS MEM
58 02326 032411         LDA     2,@LOADA ;ADDR OF BINARY DATA
59 02327 021000 LOADR:  LDA     0,0,2
60 02330 041400         STA     0,0,3   ;MOVE DATA
61 02331 166415         SUB#    3,1,SNR ;LAST DATA ?
62 02332 001400         JMP     0,3     ;YES, START BINARY LOADER
63 02333 151400         INC     2,2
64 02334 175400         INC     3,3     ;NEXT DATA ADDR
65 02335 000772         JMP     LOADR
66 02336 001375 LOADF:  BINFI           ;FIRST ADDR BIN LOADER (32K)
67 02337 001540 LOADA:  BINAD           ;ADDR POINTER TO BIN DATA
```

```
| 0038  .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         ICRST               ;RESET MEM EXT FLAG
07 02342 062701         DICP    0,1         ;TRY TO SET MEM EXT FLAG (RC3603 ONLY).
08 02343 030433         LDA     2,CMEND     ;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
63
64
65                  ;TAPE 2
66
67                  .EOT
```

```
  0039  .MAIN
01
02                ;TAPE 3B
03
04                        ;DELAY SUBROUTINE
05                        ;ARGUMENT FOLLOWING CALL IS ADDRESS
06                        ;OF DELAY CONSTANT.
07                        ;DELAY IN INCREMENTS OF 1 MSEC
08                        ;AC2 & 3 ARE USED, DEFINE WAIT, ARG
09                        ;
10                        ;CALL   CWAIT
11                        ;       ARG
12                        ;
13 02427 054416 KWAIT:    STA     3,WARET ;SAVE RETURN
14 02430 010415           ISZ     WARET   ;PASS ARG
15 02431 033400           LDA     2,@0,3  ;FETCH ARG
16 02432 050414           STA     2,WIRET ;STORE ARG
17 02433 030414           LDA     2,KINDI ;FETCH KNOVA ADDRESS
18 02434 036537           LDA     3,.MCPN ;FETCH CPUNO
19 02435 157000           ADD     2,3     ;COMPUTE KADDRESS
20 02436 031400           LDA     2,0,3   ;FETCH CPU CONSTANT
21 02437 050437 NWAIT:    STA     2,KINC  ;STORE IT FOR INC OF 1 MSEC
22 02440 014436           DSZ     KINC    ;X USEC LOOP
23 02441 000777           JMP     .-1     ;X USEC
24 02442 014404           DSZ     WIRET   ;NO. OF MS STILL TO WAIT
25 02443 000774           JMP     NWAIT   ;COUNT ARG NOT ENDED
26 02444 002401           JMP     @WARET  ;COUNT ARG ENDED, RETURN
27 02445 000000 WARET:    0
28 02446 000000 WIRET:    0
29
30 02447 002450 KINDI:    .+1             ;ADDRESS OF KNOVA
31       000012           .RDX 10
32 02450 000175 KCP0:     125             ;NOVA
33 02451 000001 KCP1:     1               ;
34 02452 000341 KCP2:     225             ;1200
35 02453 000567 KCP3:     375             ;SUPER
36 02454 000505 KCP4:     325             ;SUPER SC/830
37 02455 000536 KCP5:     350             ;800/NOVA 2-16K
38 02456 000620 KCP6:     400             ;NOVA 2-8K
39 02457 000001 KCP7:     1               ;CONSTANTS ABOVE ARE FOR GUESSED CPU TYPE
40 02460 000001 KCP10:    1               ;CONSTANTS BELOW ARE EXACTLY.
41 02461 000200 KCP11:    128             ;NOVA    7,8 USEC
42 02462 000336 KCP12:    222             ;1200    4,5 USEC
43 02463 000601 KCP13:    385             ;800     2,6 USEC
44 02464 000601 KCP14:    385             ;SUPER   2,6 USEC
45 02465 000764 KCP15:    500             ;SUPER SC  2,0 USEC
46 02466 000620 KCP16:    400             ;NOVA2/8K  2,5 USEC
47 02467 000502 KCP17:    322             ;NOVA2/16K 3,1 USEC
48 02470 000531 KCP20:    345             ;RC3603/RC3609           2,90 USEC
49 02471 000334 KCP21:    220             ;RC3603/RC3609/BREAK ON 4,55 USEC
50 02472 000463 KCP22:    307             ;RC3603/RC3608           3,26 USEC
51 02473 000314 KCP23:    204             ;RC3603/RC3608/BREAK ON 4,90 USEC
52 02474 000515 KCP24:    333             ;NOVA2/DATARAM 900 NSEC 3,00 USEC
53 02475 000001 KCP25:    1               ;FOR TROUBLE, PUT HERE RELEVANT VALUE.
54       000010           .RDX 8
55 02476 000000 KINC:     0
```

```
| 0040  .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
```

```
| 0041  .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 MSTIO:  ADCZR   0,0     ;AC0:=077777, CARRY:=1
63 02625 040743         STA     0,TIMEX ;TIMEOUT TIMECOUNT = 32767
64 02626 000402         JMP     MSTID   ;CALCULATE
```

```
I 0042  .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
```

```
| 0043  .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"
```

```
| 0044   ,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 062677         ICRST
10 02760 020434         LDA     0,NUTYP ;NUMBER OF TRYING AGAIN
11 02761 040432         STA     0,AGTYP
12 02762 004447 TRTYP:  JSR     TYMER
13 02763 125014         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 001527         UNTIM
24 02776 006040         CMESS
25 02777 001527         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 003475 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
```

```
| 0045  .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
```

```
| 0046  .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,ORDINAL ;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 COUN
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 OR
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         SUBC    3,3       ;CLEARING AC3.
63 03212 056410         STA     3,@RVTMP  ;CLEARS LOC. IN TYME LIST.
64 03213 000770         JMP     REVUC     ;
```

```
| 0047  .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,
55 03335 052021 TOMC:   STA     2,@IDX1 ;STORE COUNT INTO TYME LIST.
56 03336 000756         JMP     TOMA    ;LOOP.
57 03337 044403 TOMB:   STA     1,TOMH  ;STORE ALC-SKP IN TIME LOOP.
58 03340 060114         NIOS    XRTC    ;START RTC
59 03341 151400 TOMG:   INC     2,2     ;THESE **********************
60 03342 000000 TOMH:   0               ;   INSTRUCTIONS ***********
61 03343 000401         JMP     .+1     ;      FORM THE ************
62 03344 063514         SKPBZ   XRTC    ;         TIMING ***********
63 03345 000774         JMP     TOMG    ;            LOOP **********
64 03346 000767         JMP     TOMC    ;FOR ALC-SKP INSTRUCTIONS.
65 03347 000000 INSW2:  0               ;HERE WORKS ISZ INSAD FROM TOMJ.
66 03350 100000 INDW2:  @0              ;HERE WORKS LDA @INDAD FROM TOMJ.
```

```
| 0048  .MAIN
01
02 03351 020661 KEYS:   LDA     0,xRANK ;REINITIALIZE RANK LIST
03 03352 040022         STA     0,IDX2  ;AUTO INC POINTER.
04 03353 102520         SUBZL   0,0     ;BIT 15:=1 AS END OF KEY MARKER.
05
06 03354 026022 KEYA:   LDA     1,@IDX2 ;GET ORDINAL COUNT FROM RANK LIST.
07 03355 125223         MOVZR   1,1,SNC ;CK IF # IS ODD OR EVEN.
08 03356 000404         JMP     KEYB    ;EVEN
09 03357 101122         MOVZL   0,0,SZC ;ROTATE 0 INTO BIT 15 FOR ODD ORDINAL.
10 03360 000405         JMP     SESAME  ;CARRY BIT SET MEANS END OF KEY.
11 03361 000773         JMP     KEYA    ;MORE-ON-KEY.
12 03362 101142 KEYB:   MOVOL   0,0,SZC ;ROTATE 1 INTO BIT 15 FOR EVEN ORDINAL.
13 03363 000402         JMP     SESAME  ;CARRY BIT SET MEANS END OF KEY.
14 03364 000770         JMP     KEYA    ;MORE-ON-KEY.
15
16 03365 000401 SESAME: JMP     .+1     ;FOR TROUBLE HALT
17 03366 024415         LDA     1,SES11 ;GET FIRST CPUNO
18 03367 034415         LDA     3,LOCKA ;ADDR OF LOCK TABLE
19 03370 031400 SESA1:  LDA     2,0,3   ;GET KEY FROM TABLE
20 03371 151015         MOV#    2,2,SNR ;ZERO ?
21 03372 000407         JMP     SESOUT  ;YES, END MARK: KEY NOT KNOWN
22 03373 150015         COM#    2,2,SNR ;ALL ONES ?
23 03374 125400         INC     1,1     ;YES, NEXT CPU MARK, INC CPUNO
24 03375 112435         SUBZ#   0,2,SNR ;COMPARE KEY WITH LOCK ENTRY
25 03376 000404         JMP     SESEX   ;A HIT ! RETURN WITH CPUNO IN AC1
26 03377 175400         INC     3,3     ;INC TO NEXT LOCK ENTRY
27 03400 000770         JMP     SESA1   ;AND LOOP
28 03401 126440 SESOUT: SUBO    1,1     ;CLEAR AC1 AS KEY INDICATOR FOR NO
29 03402 002621 SESEX:  JMP     @SVTYME ;CPU FOUND, RETURN.
30 03403 000011 SES11:  11
31
32 03404 003405 LOCKA:  .+1
33 03405 064450         064450          ;NOVA KEY.                  CPU #   11
34 03406 177777         177777
35 03407 016137         016137          ;NOVA 1200 KEY.                     12
36 03410 177777         177777
37 03411 034174         034174          ;NOVA 800 KEY.                      13
38 03412 177777         177777
39 03413 002544         002544          ;SUPERNOVA KEY.                     14
40 03414 177777         177777
41 03415 007126         007126          ;SUPERNOVA SC KEY.                  15
42 03416 177777         177777
43 03417 034652         034652          ;NOVA 2 - 8K KEY.                   16
44 03420 177777         177777
45 03421 022512         022512          ;NOVA 2 - 16K KEY.                  17
46 03422 177777         177777
47 03423 170225         170225          ;RC3603/RC3609, 16K KEY             20
48 03424 170265         170265
49 03425 170365         170365
50 03426 172225         172225
51 03427 172265         172265
52 03430 177777         177777
53 03431 144567         144567          ;RC3603/RC3609/BREAK ON KEY         21
54 03432 146427         146427
55 03433 166610         166610
56 03434 177777         177777
```

```
| 0049   .MAIN
01
02 03435 166065          166065          ;RC3603/RC3608, 32K KEY          22
03 03436 166165          166165
04 03437 166225          166225
05 03440 166325          166325
06 03441 172065          172065
07 03442 172325          172325
08 03443 174050          174050
09 03444 174150          174150
10 03445 174210          174210
11 03446 174250          174250
12 03447 174310          174310
13 03450 176010          176010
14 03451 177777          177777
15 03452 144352          144352          ;RC3603/RC3608/BREAK ON KEY      23
16 03453 154352          154352
17 03454 160135          160135
18 03455 162025          162025
19 03456 162202          162202
20 03457 162225          162225
21 03460 166025          166025
22 03461 170312          170312
23 03462 170352          170352
24 03463 172012          172012
25 03464 172052          172052
26 03465 172202          172202
27 03466 176025          176025
28 03467 176050          176050
29 03470 177777          177777
30 03471 020653          020653          ;NOVA 2 - DATARAM 900 NSEC KEY   24
31 03472 177777          177777
32 03473 000000          0               ;SPACE IF TROUBLES               25
33                                       ;CORRECT JMP .+1 IN SESAME TO A HALT
34                                       ;RESTART IN 1400 (REBIN). PUT HERE
35                                       ;KEY FROM AC2 EXAMINED WHEN STOPPED,
36                                       ;AND CORRECT CPUNO 25'S CONSTANTS IN THE
37                                       ;3 ROUTINES CWAIT, TIMSK, TIMMS
38 03474 000000          0               ;END OF TABLE, NO CPU FOUND, #   0
```

```
| 0050  .MAIN
01
02                          ;FIND TYPE OF CPU USING RTC
03                          ;PART OF CPUTYP
04 03475 054444 NWTYP:  STA     3,NIRET ;SAVE RETURN
05 03476 102520         SUBZL   0,0     ;AC0:=1
06 03477 101120         MOVZL   0,0     ;AC0:=2
07 03500 126400         SUB     1,1
08 03501 061014         DOA     0,XRTC  ;SET RTC FREQUENCY
09 03502 060114         NIOS    XRTC    ;TO 100 HZ (10 MSEC)
10 03503 063514         SKPBZ   XRTC
11 03504 000777         JMP     .-1     ;SYNCHRONIZE RTC
12 03505 060114         NIOS    XRTC    ;START RTC
13 03506 125400         INC     1,1     ;COUNTS          ;X USEC
14 03507 063514         SKPBZ   XRTC    ;                ;X USEC
15 03510 000776         JMP     .-2     ;LOOP FOR 10 MS  ;X USEC
16 03511 004465         JSR     NTTYP   ;TEST RTC STABILITY
17 03512 030520         LDA     2,NN500 ;STEP -500
18 03513 034520         LDA     3,NN9   ;8 TIMES
19 03514 141000         MOV     2,0     ;STARTING AT -500=SPEED
20 03515 175405 NXTYP:  INC     3,3,SNR ;NEXT STEP OR
21 03516 000413         JMP     NFTYP   ;TYPE OF NOVA NOT FOUND
22 03517 143020         ADDZ    2,0     ;SPEED:=SPEED-500
23 03520 107013         ADD#    0,1,SNC ;NO. OF COUNTS>=(-SPEED)
24 03521 000774         JMP     NXTYP   ;THEN GO TO NXTYP ELSE
25 03522 030512         LDA     2,NC8   ;CONVERT TIMES TO CPUNO
26 03523 173000         ADD     3,2     ;AS MENTIONED BELOW
27 03524 034427         LDA     3,NEXIS ;CHECK IN
28 03525 157000         ADD     2,3     ;THE TABEL THAT
29 03526 025400         LDA     1,0,3   ;THE FOUND CPUNO
30 03527 125004         MOV     1,1,SZR ;EXIST
31 03530 000403         JMP     NRTYP   ;YES, RETURN
32 03531 006043 NFTYP:  CCRLF           ;NO
33 03532 004432 NITYP:  JSR     NOTYP   ;GET OPERATORS CPUNO
34 03533 050405 NRTYP:  STA     2,NCTYP ;SAVE CPUNO
35 03534 006046         CDICL
36 03535 006043         CCRLF
37 03536 024402         LDA     1,NCTYP ;RESTORE CPUNO
38 03537 002402         JMP     @NIRET  ;EXIT
39 03540 000000 NCTYP:  0
40 03541 000000 NIRET:  0
41              NOTYM:  .TXT | SET CPUNO > AC2| ;" SET CPUNO > AC2"
   03542 051440
   03543 052105
   03544 041440
   03545 052520
   03546 047516
   03547 037040
   03550 040440
   03551 031103
   03552 000000
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       3608BREA
47                              ;               1770     5,65   2       3609BREA
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       3603/360
53                              ;               3125     3,2    5       NOVA2-16
54                              ;               3226     3,10   5       3603/360
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
```

```
| 0051  .MAIN
01
02 03564 054411 NOTYP:  STA     3,NOTRE
03 03565 006046         CDICL           ;CLEAR DIS
04 03566 006044         CDISP           ;DIS = MESSAGE
05 03567 003542         NOTYM
06 03570 006040         CMESS
07 03571 003542         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 000666         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 0052  .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>
66  03753  136400 <0><75>
67  03754  137000 <0><76>
68  03755  037400 <0><77>
```

```
I 0053   .MAIN
02 03756 140000 <0><100>
03 03757 040400 <0><101>
04 03760 041000 <0><102>
05 03761 141400 <0><103>
06 03762 042000 <0><104>
07 03763 142400 <0><105>
08 03764 143000 <0><106>
09 03765 043400 <0><107>
10 03766 044000 <0><110>
11 03767 144400 <0><111>
12 03770 145000 <0><112>
13 03771 045400 <0><113>
14 03772 146000 <0><114>
15 03773 046400 <0><115>
16 03774 047000 <0><116>
17 03775 147400 <0><117>
18 03776 050000 <0><120>
19 03777 150400 <0><121>
20 04000 151000 <0><122>
21 04001 051400 <0><123>
22 04002 152000 <0><124>
23 04003 052400 <0><125>
24 04004 053000 <0><126>
25 04005 153400 <0><127>
26 04006 154000 <0><130>
27 04007 054400 <0><131>
28 04010 055000 <0><132>
29 04011 155400 <0><133>
30 04012 056000 <0><134>
31 04013 156400 <0><135>
32 04014 157000 <0><136>
33 04015 057400 <0><137>
34 04016 006000 <0><14>
35 04017 040400 <0><101>
36 04020 041000 <0><102>
37 04021 141400 <0><103>
38 04022 042000 <0><104>
39 04023 142400 <0><105>
40 04024 143000 <0><106>
41 04025 043400 <0><107>
42 04026 044000 <0><110>
43 04027 144400 <0><111>
44 04030 145000 <0><112>
45 04031 045400 <0><113>
46 04032 146000 <0><114>
47 04033 046400 <0><115>
48 04034 047000 <0><116>
49 04035 147400 <0><117>
50 04036 050000 <0><120>
51 04037 150400 <0><121>
52 04040 151000 <0><122>
53 04041 051400 <0><123>
54 04042 152000 <0><124>
55 04043 052400 <0><125>
56 04044 053000 <0><126>
57 04045 153400 <0><127>
58 04046 154000 <0><130>
59 04047 054400 <0><131>
60 04050 055000 <0><132>
61 04051 140000 <0><100>
62 04052 121400 <0><43>
63 04053 156400 <0><135>
64 04054 004400 <0><11>
65 04055 000006 <6><0>?
   04056 000000
```

```
| 0054   .MAIN
01
02          000012 .RDX     10
03                 PTAB2:
04                 .TXTE?
05  04057  000006 <6><0>
06  04060  000006 <6><0>
07  04061  000006 <6><0>
08  04062  000006 <6><0>
09  04063  000006 <6><0>
10  04064  000006 <6><0>
11  04065  000006 <6><0>
12  04066  000006 <6><0>
13  04067  000006 <6><0>
14  04070  004400 <0><9>
15  04071  005000 <0><10>
16  04072  000006 <6><0>
17  04073  006000 <0><12>
18  04074  106400 <0><13>
19  04075  000006 <6><0>
20  04076  000006 <6><0>
21  04077  000006 <6><0>
22  04100  000006 <6><0>
23  04101  000006 <6><0>
24  04102  000006 <6><0>
25  04103  000006 <6><0>
26  04104  000006 <6><0>
27  04105  000006 <6><0>
28  04106  000006 <6><0>
29  04107  000006 <6><0>
30  04110  000006 <6><0>
31  04111  000006 <6><0>
32  04112  000006 <6><0>
33  04113  000006 <6><0>
34  04114  000006 <6><0>
35  04115  000006 <6><0>
36  04116  000006 <6><0>
37  04117  117400 <0><31>
38  04120  036000 <0><60>
39  04121  137000 <0><62>
40  04122  040400 <0><65>
41  04123  131000 <0><50>
42  04124  034400 <0><57>
43  04125  035000 <0><58>
44  04126  136400 <0><61>
45  04127  030000 <0><48>
46  04130  130400 <0><49>
47  04131  135400 <0><59>
48  04132  027000 <0><46>
49  04133  020400 <0><33>
50  04134  127400 <0><47>
51  04135  120000 <0><32>
52  04136  133400 <0><55>
53  04137  022000 <0><36>
54  04140  122400 <0><37>
55  04141  123000 <0><38>
56  04142  023400 <0><39>
57  04143  024000 <0><40>
58  04144  124400 <0><41>
59  04145  125000 <0><42>
60  04146  025400 <0><43>
61  04147  126000 <0><44>
62  04150  026400 <0><45>
63  04151  021000 <0><34>
64  04152  121400 <0><35>
65  04153  032400 <0><53>
66  04154  132000 <0><52>
67  04155  033000 <0><54>
68  04156  037400 <0><63>
```

```
1 0055   .MAIN
02 04157 041000 <0><66>
03 04160 057400 <0><95>
04 04161 157000 <0><94>
05 04162 156400 <0><93>
06 04163 056000 <0><92>
07 04164 155400 <0><91>
08 04165 055000 <0><90>
09 04166 054400 <0><89>
10 04167 154000 <0><88>
11 04170 153400 <0><87>
12 04171 053000 <0><86>
13 04172 052400 <0><85>
14 04173 152000 <0><84>
15 04174 051400 <0><83>
16 04175 151000 <0><82>
17 04176 150400 <0><81>
18 04177 050000 <0><80>
19 04200 147400 <0><79>
20 04201 047000 <0><78>
21 04202 046400 <0><77>
22 04203 146000 <0><76>
23 04204 045400 <0><75>
24 04205 145000 <0><74>
25 04206 144400 <0><73>
26 04207 142400 <0><69>
27 04210 042000 <0><68>
28 04211 141400 <0><67>
29 04212 044000 <0><72>
30 04213 043400 <0><71>
31 04214 143000 <0><70>
32 04215 031400 <0><51>
33 04216 134000 <0><56>
34 04217 006000 <0><12>
35 04220 057400 <0><95>
36 04221 157000 <0><94>
37 04222 156400 <0><93>
38 04223 056000 <0><92>
39 04224 155400 <0><91>
40 04225 055000 <0><90>
41 04226 054400 <0><89>
42 04227 154000 <0><88>
43 04230 153400 <0><87>
44 04231 053000 <0><86>
45 04232 052400 <0><85>
46 04233 152000 <0><84>
47 04234 051400 <0><83>
48 04235 151000 <0><82>
49 04236 150400 <0><81>
50 04237 050000 <0><80>
51 04240 147400 <0><79>
52 04241 047000 <0><78>
53 04242 046400 <0><77>
54 04243 146000 <0><76>
55 04244 045400 <0><75>
56 04245 145000 <0><74>
57 04246 144400 <0><73>
58 04247 142400 <0><69>
59 04250 042000 <0><68>
60 04251 141400 <0><67>
61 04252 041000 <0><66>
62 04253 040400 <0><65>
63 04254 143000 <0><70>
64 04255 004400 <0><9>
65 04256 000006 <6><0>?
   04257 000000
```

```
| 0056   .MAIN
01
02          000012 .ROX     12
03                 PTAB3:
04                 .TXTE?
05  04260  000006 <6><0>
06  04261  000006 <6><0>
07  04262  000006 <6><0>
08  04263  000006 <6><0>
09  04264  000006 <6><0>
10  04265  000006 <6><0>
11  04266  000006 <6><0>
12  04267  000006 <6><0>
13  04270  000006 <6><0>
14  04271  004400 <0><9>
15  04272  005000 <0><10>
16  04273  000006 <6><0>
17  04274  006000 <0><12>
18  04275  106400 <0><13>
19  04276  000006 <6><0>
20  04277  000006 <6><0>
21  04300  000006 <6><0>
22  04301  000006 <6><0>
23  04302  000006 <6><0>
24  04303  000006 <6><0>
25  04304  000006 <6><0>
26  04305  000006 <6><0>
27  04306  000006 <6><0>
28  04307  000006 <6><0>
29  04310  000006 <6><0>
30  04311  000006 <6><0>
31  04312  000006 <6><0>
32  04313  000006 <6><0>
33  04314  000006 <6><0>
34  04315  000006 <6><0>
35  04316  000006 <6><0>
36  04317  000006 <6><0>
37  04320  117400 <0><31>
38  04321  134000 <0><56>
39  04322  035000 <0><58>
40  04323  136400 <0><61>
41  04324  027000 <0><46>
42  04325  032400 <0><53>
43  04326  033000 <0><54>
44  04327  034400 <0><57>
45  04330  126000 <0><44>
46  04331  026400 <0><45>
47  04332  133400 <0><55>
48  04333  125000 <0><42>
49  04334  156400 <0><93>
50  04335  025400 <0><43>
51  04336  056000 <0><92>
52  04337  031400 <0><51>
53  04340  120000 <0><32>
54  04341  020400 <0><33>
55  04342  021000 <0><34>
56  04343  121400 <0><35>
57  04344  022000 <0><36>
58  04345  122400 <0><37>
59  04346  123000 <0><38>
60  04347  023400 <0><39>
61  04350  024000 <0><40>
62  04351  124400 <0><41>
63  04352  157000 <0><94>
64  04353  057400 <0><95>
65  04354  130400 <0><49>
66  04355  030000 <0><48>
67  04356  131000 <0><50>
68  04357  135400 <0><59>
```

```
| 0057   .MAIN
02  04360  137000  <0><62>
03  04361  155400  <0><91>
04  04362  055000  <0><90>
05  04363  054400  <0><89>
06  04364  154000  <0><88>
07  04365  153400  <0><87>
08  04366  053000  <0><86>
09  04367  052400  <0><85>
10  04370  152000  <0><84>
11  04371  051400  <0><83>
12  04372  151000  <0><82>
13  04373  150400  <0><81>
14  04374  050000  <0><80>
15  04375  147400  <0><79>
16  04376  047000  <0><78>
17  04377  046400  <0><77>
18  04400  146000  <0><76>
19  04401  045400  <0><75>
20  04402  145000  <0><74>
21  04403  144400  <0><73>
22  04404  044000  <0><72>
23  04405  043400  <0><71>
24  04406  143000  <0><70>
25  04407  142400  <0><69>
26  04410  040400  <0><65>
27  04411  140000  <0><64>
28  04412  037400  <0><63>
29  04413  042000  <0><68>
30  04414  141400  <0><67>
31  04415  041000  <0><66>
32  04416  127400  <0><47>
33  04417  132000  <0><52>
34  04420  006000  <0><12>
35  04421  155400  <0><91>
36  04422  055000  <0><90>
37  04423  054400  <0><89>
38  04424  154000  <0><88>
39  04425  153400  <0><87>
40  04426  053000  <0><86>
41  04427  052400  <0><85>
42  04430  152000  <0><84>
43  04431  051400  <0><83>
44  04432  151000  <0><82>
45  04433  150400  <0><81>
46  04434  050000  <0><80>
47  04435  147400  <0><79>
48  04436  047000  <0><78>
49  04437  046400  <0><77>
50  04440  146000  <0><76>
51  04441  045400  <0><75>
52  04442  145000  <0><74>
53  04443  144400  <0><73>
54  04444  044000  <0><72>
55  04445  043400  <0><71>
56  04446  143000  <0><70>
57  04447  142400  <0><69>
58  04450  040400  <0><65>
59  04451  140000  <0><64>
60  04452  037400  <0><63>
61  04453  137000  <0><62>
62  04454  136400  <0><61>
63  04455  041000  <0><66>
64  04456  004400  <0><9>
65  04457  000006  <6><0>?
    04460  000000
```

```
I 0058   .MAIN
01
02          000010 .RDX    8
03                 PTAB4:
04                 .TXTE?
05  04461  000006 <6><0>
06  04462  000006 <6><0>
07  04463  000006 <6><0>
08  04464  000006 <6><0>
09  04465  000006 <6><0>
10  04466  000006 <6><0>
11  04467  000006 <6><0>
12  04470  000006 <6><0>
13  04471  000006 <6><0>
14  04472  004400 <0><11>
15  04473  005000 <0><12>
16  04474  000006 <6><0>
17  04475  006000 <0><14>
18  04476  106400 <0><15>
19  04477  000006 <6><0>
20  04500  000006 <6><0>
21  04501  000006 <6><0>
22  04502  000006 <6><0>
23  04503  000006 <6><0>
24  04504  000006 <6><0>
25  04505  000006 <6><0>
26  04506  000006 <6><0>
27  04507  000006 <6><0>
28  04510  000006 <6><0>
29  04511  000006 <6><0>
30  04512  000006 <6><0>
31  04513  000006 <6><0>
32  04514  000006 <6><0>
33  04515  000006 <6><0>
34  04516  000006 <6><0>
35  04517  000006 <6><0>
36  04520  000006 <6><0>
37  04521  117400 <0><37>
38  04522  022000 <0><44>
39  04523  025400 <0><53>
40  04524  157000 <0><136>
41  04525  047000 <0><116>
42  04526  156400 <0><135>
43  04527  136400 <0><75>
44  04530  127400 <0><57>
45  04531  026400 <0><55>
46  04532  027000 <0><56>
47  04533  045400 <0><113>
48  04534  137000 <0><76>
49  04535  146000 <0><114>
50  04536  145000 <0><112>
51  04537  044000 <0><110>
52  04540  046400 <0><115>
53  04541  037400 <0><77>
54  04542  140000 <0><100>
55  04543  040400 <0><101>
56  04544  041000 <0><102>
57  04545  141400 <0><103>
58  04546  042000 <0><104>
59  04547  142400 <0><105>
60  04550  143000 <0><106>
61  04551  043400 <0><107>
62  04552  144400 <0><111>
63  04553  126000 <0><54>
64  04554  122400 <0><45>
65  04555  021000 <0><42>
66  04556  030000 <0><60>
67  04557  124400 <0><51>
68  04560  125000 <0><52>
```

```
I 0059   .MAIN
02 04561 057400 <0><137>
03 04562 147400 <0><117>
04 04563 050000 <0><120>
05 04564 150400 <0><121>
06 04565 151000 <0><122>
07 04566 051400 <0><123>
08 04567 152000 <0><124>
09 04570 052400 <0><125>
10 04571 053000 <0><126>
11 04572 153400 <0><127>
12 04573 154000 <0><130>
13 04574 054400 <0><131>
14 04575 055000 <0><132>
15 04576 155400 <0><133>
16 04577 056000 <0><134>
17 04600 130400 <0><61>
18 04601 131000 <0><62>
19 04602 031400 <0><63>
20 04603 132000 <0><64>
21 04604 032400 <0><65>
22 04605 033000 <0><66>
23 04606 133400 <0><67>
24 04607 134000 <0><70>
25 04610 034400 <0><71>
26 04611 035000 <0><72>
27 04612 135400 <0><73>
28 04613 036000 <0><74>
29 04614 020400 <0><41>
30 04615 123000 <0><46>
31 04616 023400 <0><47>
32 04617 024000 <0><50>
33 04620 121400 <0><43>
34 04621 006000 <0><14>
35 04622 147400 <0><117>
36 04623 050000 <0><120>
37 04624 150400 <0><121>
38 04625 151000 <0><122>
39 04626 051400 <0><123>
40 04627 152000 <0><124>
41 04630 052400 <0><125>
42 04631 053000 <0><126>
43 04632 153400 <0><127>
44 04633 154000 <0><130>
45 04634 054400 <0><131>
46 04635 055000 <0><132>
47 04636 155400 <0><133>
48 04637 056000 <0><134>
49 04640 130400 <0><61>
50 04641 131000 <0><62>
51 04642 031400 <0><63>
52 04643 132000 <0><64>
53 04644 032400 <0><65>
54 04645 033000 <0><66>
55 04646 133400 <0><67>
56 04647 134000 <0><70>
57 04650 034400 <0><71>
58 04651 035000 <0><72>
59 04652 135400 <0><73>
60 04653 036000 <0><74>
61 04654 057400 <0><137>
62 04655 157000 <0><136>
63 04656 023400 <0><47>
64 04657 004400 <0><11>
65 04660 000006 <6><0>?
   04661 000000
```

```
| 0060  .MAIN
01
02                          ;POWER RESTART ROUTINE
03
04 04662 026454 POWON:  LDA     1,@PSTAC ;GET PRINT INHI INITIAL FOR PRINT
05 04663 046454         STA     1,@PSETP
06 04664 062677         IORST
07 04665 006061         CWAIT           ;WAIT 3 SECONDS TO TERMINAL READY
08 04666 001236         SEC3
09 04667 020442         LDA     0,PCOTT ;PRINT 5 CR, LF FOR TTY, SILENT
10 04670 040440         STA     0,PCOUN
11 04671 006043         CCRLF
12 04672 014436         DSZ     PCOUN
13 04673 000776         JMP     .-2
14 04674 020436         LDA     0,PCH14 ;FF FOR LPT AND CLEARING SOME CRT'S
15 04675 006041         CCHAR
16 04676 006061         CWAIT           ;20 MSEC FOR CRT
17 04677 001237         SECM2
18 04700 020433         LDA     0,PCH35 ;HOME UP FOR CRT
19 04701 006041         CCHAR
20 04702 006061         CWAIT           ;20 MSEC FOR CRT
21 04703 001237         SECM2
22 04704 020430         LDA     0,PCH37 ;ERASE EOF FOR CRT
23 04705 006041         CCHAR
24 04706 006061         CWAIT           ;20 MSEC FOR CRT
25 04707 001237         SECM2
26 04710 006040         CMESS
27 04711 004745         MPOWO           ;POWER
28 04712 006043         CCRLF
29 04713 006040         CMESS
30 04714 007167         PROG            ;ACTUAL PROG NAME
31 04715 006071 RPSAQ:  CQUES
32 04716 004755         MSAGU           ;SET SWITCHES, START ADDR
33 04717 004750         DSAGU
34 04720 004735         PSAAN           ;SUGGESTED ANSWER
35 04721 006054         CTZOC
36 04722 006060         CDZOC
37 04723 006104         CGTCK           ;READ ANSWER
38 04724 000402         JMP     .+2     ;SUGGESTED ACCEPTED BY OPERATOR
39 04725 000770         JMP     RPSAQ   ;ERROR RETURN
40 04726 030075         LDA     2,DIGIN ;ANSWER INPUT'ED
41 04727 001000         JMP     0,2     ;START PROG
42
43 04730 000000 PCOUN:  0               ;COUNTER
44 04731 000005 PCOTT:  5               ;5 CR,LF
45 04732 000014 PCH14:  14              ;FF
46 04733 000035 PCH35:  35              ;HOME UP
47 04734 000037 PCH37:  37              ;ERASE EOF
48 04735 000400 PSAAN:  400             ;SUGGESTED START ADDR
49 04736 001372 PSTAC:  KSTAC           ;ASM VALUE OF JMP NINHI
50 04737 000574 PSETP:  SETAC           ;PRINT INHIBIT ROUTINE
51
52                          ;INITIAL START ADDRESS ROUTINE
53                          ;USED TO HELP START WITHOUT SWITCHES
54
55 04740 026776 SWISA:  LDA     1,@PSTAC ;GET PRINT INHI INITIAL FOR PRINT
56 04741 046776         STA     1,@PSETP
57 04742 000753         JMP     RPSAQ   ;USE POWER RESTART ROUTINE
```

```
| 0061  .MAIN
01
02                MCRLF:  .TXT !<15><12>! ;"<15><12>"
   04743 005015
   04744 000000
03
04                MPOWO:  .TXT !POWER!    ;"POWER"
   04745 047520
   04746 042527
   04747 000122
05
06                DSAQU:  .TXT !SEE4,2,SA!        ;"SEE4,2,SA"
   04750 042523
   04751 032105
   04752 031056
   04753 051454
   04754 000101
07
08                MSAQU:  .TXT !SET SWITCHES TO CONTROL, (4,2), 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 027064
   04773 024462
   04774 020054
   04775 052123
   04776 051101
   04777 040524
   05000 042104
   05001 000122
09                      ;"SET SWITCHES TO CONTROL, (4,2), STARTADDR"
10
11                MCPUT:  .TXT !<15><12>CPU TYPE: !       ;"<15><12>CPU TYPE: "
   05002 005015
   05003 050103
   05004 020125
   05005 054524
   05006 042520
   05007 020072
   05010 000000
```

```
| 0062  .MAIN
01
02                          ;ROUTINE TO EXAMINE MEMORY.
03
04 05011 165000 EXMEM:  MOV     3,1       ;
05 05012 006072         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
```

```
| 0063  .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
```

```
| 0064   .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:  CQUES
07 05132 005113         MDMMC               ;DEPOSIT:
08 05133 005113         MDMMC
09 05134 005125         DPCON               ;SUGGESTED ANSWER
10 05135 006052         CTOCT
11 05136 006056         CDOCT
12 05137 006104         CGTOK               ;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:  CQUES
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         CGTOK               ;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:  CQUES
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         CGTOK               ;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
```

```
| 0065  .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,@2,3   ;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 044412         STA     1,QUEST
29 05223 025404         LDA     1,4,3    ;5. PARAM
30 05224 044411         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 000000 QUEST:  0                ;NUMBER PRINT ROUTINE TTO/LPT
39 05235 000000 QUESS:  0                ;NUMBER PRINT ROUTINE DIS
40 05236 006044         CDISP
41 05237 005206         MXQUE
42 05240 006040         CMESS
43 05241 005206         MXQUE            ;?
44 05242 006040         CMESS
45 05243 005207         MX2SP            ;2 SPACE
46 05244 034740         LDA     3,RQUES
47 05245 001405         JMP     5,3      ;BYPASS PARAM., RETURN
48
49
50              ;TAPE 3
51
52              .EOT
```

```
0066  .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.
```

```
| 0067  .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.
```

```
| 0068  .MAIN
01
02                ; SUBROUTINES USED BY ALL INPUTROUTINES
03
04                ; GETCH GETS A CHARACTER FROM TTI OR NUK TO AC0
05
06 05246 054456 GETCH:  STA     3,GETRE
07 05247 004443         JSR     BZNUK    ; CHECK BUSY NUK
08 05250 004431         JSR     INTTI    ; INPUTDEVICE = TTI
09 05251 004446         JSR     INNUK    ; INPUTDEVICE = NUK
10 05252 024461         LDA     1,RAZER  ; READ ASCII FROM NUK
11 05253 034454         LDA     3,RALIF  ; AC0=KEY
12 05254 054452         STA     3,RADYN
13 05255 101122 RANDI:  MOVZL   0,0,SZC  ;
14 05256 000421         JMP     RAEND    ; DIGIT FOUND
15 05257 125400         INC     1,1      ; AC1 = DIGIT
16 05260 014446         DSZ     RADYN    ; 0-9
17 05261 000774         JMP     RANDI
18 05262 126400         SUB     1,1
19 05263 101122         MOVZL   0,0,SZC
20 05264 024445         LDA     1,RAPLU  ; AC1 = +
21 05265 101122         MOVZL   0,0,SZC
22 05266 024444         LDA     1,RAMIN  ; AC1 = -
23 05267 101122         MOVZL   0,0,SZC
24 05270 024437         LDA     1,RALIF  ; AC1 = LF
25 05271 101122         MOVZL   0,0,SZC
26 05272 024436         LDA     1,RASPA  ; AC1 = SP
27 05273 101122         MOVZL   0,0,SZC
28 05274 024440         LDA     1,RACAN  ; AC1 = CAN
29 05275 127405         AND     1,1,SNR
30 05276 063077         HALT             ; NO KEYS
31 05277 121000 RAEND:  MOV     1,0      ; AC1 TO AC0
32 05300 002424         JMP     @GETRE
33
34 05301 063610 INTTI:  SKPDN   XTTI     ;IS TTI KEY PRESSED ?
35 05302 001400         JMP     0,3      ; NO, CHECK OTHER INPUT DEVICE
36 05303 024422         LDA     1,HC177  ; YES, GET CHAR
37 05304 060410         DIA     0,XTTI
38 05305 123400         AND     1,0
39 05306 060110         NIOS    XTTI
40 05307 101015         MOV#    0,0,SNR  ; NULL CHAR ?
41 05310 000771         JMP     INTTI
42 05311 002413         JMP     @GETRE
43
44 05312 126400 BZNUK:  SUB     1,1      ; AC1:=0
45 05313 060434 AANUK:  DIA     0,NUK    ; TEST BUZY NUK
46 05314 122414         SUB#    1,0,SZR  ; IS AC0=0  ?
47 05315 000776         JMP     AANUK    ; NO - KEYBOARD NOT READY
48 05316 001400         JMP     0,3      ; YES, RETURN.
49
50 05317 126400 INNUK:  SUB     1,1      ; AC1:= 0
51 05320 060434 BBNUK:  DIA     0,NUK    ; GET KEY BITS
52 05321 122415         SUB#    1,0,SNR  ; ARE THEY ZERO ?
53 05322 001776         JMP     -2,3     ; YES, OTHER INPUT, TEST TTI
54 05323 001400         JMP     0,3      ; NO, KEY DEPRESSED
55
56 05324 000000 GETRE:  0
57 05325 000177 HC177:  177
58 05326 000000 RADYN:  0
59 05327 000012 RALIF:  12
60 05330 000040 RASPA:  40
61 05331 000053 RAPLU:  53
62 05332 000055 RAMIN:  55
63 05333 000060 RAZER:  60
64 05334 000030 RACAN:  30
```

```
I 0069  .MAIN
01
02 05335 030550 DELTE:  LDA     2,HC30  ; THE DELTE ROUTINE LOOKS FOR RUBOUT
03 05336 112415         SUB#    0,2,SNR ; IS AC0= 30 ?
04 05337 000405         JMP     DELOV   ; YES - RUBOUT
05 05340 030765         LDA     2,HC177 ; NO
06 05341 112415         SUB#    0,2,SNR ; IS AC0= 177 ?
07 05342 000402         JMP     DELOV   ; YES - RUBOUT
08 05343 001401         JMP     1,3     ; NO
09
10 05344 054461 DELOV:  STA     3,DELRE ; CHECK FOR TOO MANY RUBOUTS
11 05345 030457         LDA     2,LASTN ; OR THE FIRST CHAR IS RUBOUT
12 05346 151102         MOVL    2,2,SZC ; IS LASTN = 177777 ?
13 05347 000417         JMP     ILLGR   ; YES - FIRST IS RUBOUT
14 05350 151102         MOVL    2,2,SZC ; IS LASTN = 077777 ?
15 05351 000415         JMP     ILLGR   ; YES - TOO MANY RUBOUTS
16 05352 152220         ADCZR   2,2     ; NO - PLACE 077777 IN LASTN
17 05353 050451         STA     2,LASTN ; TO INDICATE RUBOUT
18 05354 020533         LDA     0,HC44  ; TYPE $
19 05355 004402         JSR     TYPIN
20 05356 002447         JMP     @DELRE  ; RETURN + 1
21
22 05357 054406 TYPIN:  STA     3,TYPRE ; TYPE THE CHAR IN AC0 ON TTY/DIS
23 05360 004427         JSR     TERMT   ; TERMINATOR ?
24 05361 002404         JMP     @TYPRE  ; YES, DO NOT COPY
25 05362 006045         CDOUT           ; TRY TO TYPE CHAR ON DISPLAY
26 05363 006041         CCHAR           ; TRY TO TYPE CHAR ON TTO
27 05364 002401         JMP     @TYPRE
28 05365 000000 TYPRE:  0
29
30 05366 020521 ILLGR:  LDA     0,HC44  ; ILLEGAL ROUTINE, TYPE $
31 05367 004770 ILLEG:  JSR     TYPIN   ; COPY LAST CHAR
32 05370 020520         LDA     0,HC52
33 05371 004766         JSR     TYPIN   ; TYPE *
34 05372 126400         SUB     1,1     ; AC1:= 0
35 05373 044075         STA     1,DIGIN ; DIGIN:= 0
36 05374 046426         STA     1,@XTXCO ; TEXIN:= CR,LF,0
37 05375 006426         JSR     @XTXND  ; FOR EMPTY TEXTSTRING
38 05376 010423         ISZ     INRET   ; ILLEGAL OR OVERFLOW RETURN
39 05377 000403         JMP     ONTER   ; IS TO CALL + 2
40
41 05400 010421 INTER:  ISZ     INRET   ; TERMINATION RETURN TO CALL+3
42 05401 010420         ISZ     INRET   ; FOR ACCEPTED INPUT
43 05402 006040 ONTER:  CMESS           ; TYPE CR,LF WITH MESS TO WAIT FOR
44 05403 004743         MCRLF           ; DEVICE READY, RETURN IS TO CALL + 1
45 05404 006044         CDISP           ; FOR NO INPUT, ONLY TERMINATION
46 05405 004743         MCRLF           ; (FOR ANSWER QUE. WITH AN OKAY).
47 05406 002413         JMP     @INRET  ; RETURN TO MAIN PROGRAM
48
49 05407 030474 TERMT:  LDA     2,HC13  ; THE TERMT ROUTINE LOOKS FOR TERM.
50 05410 112415         SUB#    0,2,SNR ; IS AC0=13 ?
51 05411 001401         JMP     1,3     ; YES - IT IS NOT A TERM
52 05412 030472         LDA     2,HC15  ; NO
53 05413 112433         SUBZ#   0,2,SNC ; IS AC0=<15 ?
54 05414 001401         JMP     1,3     ; NO, IT'S NOT A TERMINATOR
55 05415 030465         LDA     2,HC11  ; YES
56 05416 112432         SUBZ#   0,2,SZC ; IS AC0=< 11 ?
57 05417 001401         JMP     1,3     ; YES - IT IS NOT A TERMINATOR
58 05420 001400         JMP     0,3     ; NO - IT IS A TERMINATOR, RETURN
59
60 05421 000000 INRET:  0               ; RETURN ADDRESS TO MAIN PROGRAM
61 05422 006330 XTXCO:  TXCOU           ; ADDRESS OF TEXTBUFFER COUNTER
62 05423 006401 XTXND:  TXEND           ; ADDRESS OF TEXT TERM ROUTINE
63 05424 000000 LASTN:  0               ; X77777 FOR FIRST/RUBOUT, ELSE = CHAR
64 05425 000000 DELRE:  0               ; RETURN ADDRESS FOR DELTE
```

```
| 0070  ,MAIN
01
02                ; DECIMAL-NUMBER-INPUTROUTINE,
03
04 05426 054773 GETDC:  STA     3,INRET ; INITIALIZE
05 05427 126000         ADC     1,1     ; AC1:=177777
06 05430 046540         STA     1,@LAST4 ; SET LAST4
07 05431 044534         STA     1,DSIGN ; SET DSIGN TO +
08 05432 126400         SUB     1,1     ; AC1:= 0
09 05433 044534         STA     1,NUMB4 ; CLEAR NUMBER
10 05434 004612 GETD1:  JSR     GETCH   ; GET CHARACTER TO AC0
11 05435 004752         JSR     TERMT   ; TEST FOR TERMINATOR
12 05436 000456         JMP     TERMDC  ; IT IS A TERMINATOR
13 05437 004676         JSR     DELTE   ; IT'S NOT A TERMINATOR- IS IT A DEL ?
14 05440 000774         JMP     GETD1   ; IT IS A DELETE-CHAR
15 05441 004412         JSR     LETE4   ; IT'S NOT A DELETE-CHAR
16 05442 004510         JSR     CHRAN   ; PUT CHAR IN RANGE 0-9
17 05443 004527         JSR     OFTDC   ; TEST FOR OVERFLOW
18 05444 000403         JMP     DELDC   ; FIRST
19 05445 000402         JMP     DELDC   ; LAST IS RUBOUT
20 05446 004462         JSR     PLADC   ; ADD LAST4 TO NUMB4 * 10-DEC,
21 05447 024517 DELDC:  LDA     1,DCDIG ; STORE NEW DIGIT IN LAST4
22 05450 046520         STA     1,@LAST4
23 05451 004706 ECHO4:  JSR     TYPIN   ; TYPE NEW CHARACTER
24 05452 000762         JMP     GETD1   ; GET NEXT CHARACTER
25
26                ; SUBROUTINES USED BY GETDC,
27
28 05453 054426 LETE4:  STA     3,LERE4 ; LEGAL TEST
29 05454 034432         LDA     3,HC40
30 05455 111000         MOV     0,2     ; AC2:= CHAR
31 05456 024513         LDA     1,HC60
32 05457 116415         SUB#    0,3,SNR ; IS AC0 = 40 ?
33 05460 000771         JMP     ECHO4   ; YES - THE CHAR IS A SPACE
34 05461 034427         LDA     3,HC52  ; NO
35 05462 116432         SUBZ#   0,3,SZC ; IS AC0 > 52 ?
36 05463 004704         JSR     ILLEG   ; NO - THE CHAR IS NOT LEGAL
37 05464 034427         LDA     3,HC72  ; YES
38 05465 116033         ADCZ#   0,3,SNC ; IS AC0 < 72 ?
39 05466 004701         JSR     ILLEG   ; NO - NOT LEGAL
40 05467 034422         LDA     3,HC53  ; YES
41 05470 116415         SUB#    0,3,SNR ; IS AC0 >< 53 ?
42 05471 133000         ADD     1,2     ; NO - IT'S A PLUSSIGN, ADD 60
43 05472 034420         LDA     3,HC55  ; YES
44 05473 116415         SUB#    0,3,SNR ; IS AC0 >< 55 ?
45 05474 133000         ADD     1,2     ; NO - IT'S A MINUSSIGN ADD 60
46 05475 034474         LDA     3,HC60  ; YES
47 05476 156032         ADCZ#   2,3,SZC ; IS AC0 >= 60 ?
48 05477 004670         JSR     ILLEG   ; NO NOT LEGAL
49 05500 002401         JMP     @LERE4  ; YES - IT'S A VALID DIGIT
50 05501 000000 LERE4:  0
51
52 05502 000011 HC11:   11
53 05503 000013 HC13:   13
54 05504 000015 HC15:   15
55 05505 000030 HC30:   30
56 05506 000040 HC40:   40
57 05507 000044 HC44:   44
58 05510 000052 HC52:   52
59 05511 000053 HC53:   53
60 05512 000055 HC55:   55
61 05513 000072 HC72:   72
```

```
| 0071  .MAIN
01
02 05514 004456 TERMDC: JSR     OFTDC   ; TERMINATION, TEST OVERFLOW
03 05515 000665         JMP     ONTER   ; TERM IS FIRST
04 05516 000402         JMP     TERDC   ; LAST IS RUBOUT
05 05517 004411         JSR     PLADC   ; ADD LAST DIGIT TO NUMB4
06 05520 024447 TERDC:  LDA     1,NUMB4
07 05521 030444         LDA     2,DSIGN
08 05522 125015         MOV#    1,1,SNR ; IS AC1 = 0 ?
09 05523 000403         JMP     TERM1   ; YES - ZERO ALLWAYS POSITIVE
10 05524 151014         MOV#    2,2,SZR ; IS DSIGN = 0 ?
11 05525 124400         NEG     1,1     ; NO - IT IS A NEGATIVE NUMBER
12 05526 044075 TERM1:  STA     1,DIGIN ; YES - IT'S A POSITIVE NUMBER
13 05527 000651         JMP     INTER   ; OUTPUT TERM CHAR
14
15 05530 054421 PLADC:  STA     3,REPL4 ; PLACE LAST DIGIT/SIGN
16 05531 026437         LDA     1,@LAST4
17 05532 152400         SUB     2,2     ; AC2:= 0
18 05533 034756         LDA     3,HC53  ; LAST = + ?
19 05534 136415         SUB#    1,3,SNR ; NO
20 05535 000412         JMP     PLASI   ; YES, SET DSIGN
21 05536 152520         SUBZL   2,2     ; AC2:= 1
22 05537 034753         LDA     3,HC55  ; LAST = - ?
23 05540 136415         SUB#    1,3,SNR ; NO
24 05541 000406         JMP     PLASI   ; YES, SET DSIGN
25 05542 030425         LDA     2,NUMB4
26 05543 004413         JSR     MULTE   ; MULTIPLY NUMB4 WITH 10-DEC.
27 05544 133000         ADD     1,2     ; ADD LAST4 TO NUMB4*10
28 05545 050422         STA     2,NUMB4 ; PLACE NEW NUMBER
29 05546 002403         JMP     @REPL4
30 05547 050416 PLASI:  STA     2,DSIGN ; PLACE NEW SIGN
31 05550 002401         JMP     @REPL4  ; RETURN
32 05551 000000 REPL4:  0
33
34 05552 024417 CHRAN:  LDA     1,HC60  ; PUT CHAR IN RANGE 0 - 9 OR
35 05553 132400         SUB     1,2     ; PUT CHAR "+" TO 53 OR
36 05554 050412         STA     2,DCDIG ; PUT CHAR "-" TO 55
37 05555 001400         JMP     0,3     ; + AND - WAS ADDED 60 IN LEGAL TEST
38
39 05556 044406 MULTE:  STA     1,MULSA ; NUMBER TO BE MULT. BY 10 IN AC2
40 05557 145120         MOVZL   2,1     ; AC2=NUM, AC1=2*NUM
41 05560 125120         MOVZL   1,1     ; AC2=NUM, AC1=4*NUM
42 05561 133120         ADDZL   1,2     ; AC2=10*NUM, AC1=4*NUM
43 05562 024402         LDA     1,MULSA
44 05563 001400         JMP     0,3     ; RESULT IN AC2
45
46 05564 000000 MULSA:  0
47 05565 000000 DSIGN:  0               ; SIGNFLAG, 0=+, 1=-, 177777=NO SIGN=+
48 05566 000000 DCDIG:  0
49 05567 000000 NUMB4:  0
50 05570 005424 LAST4:  LASTN
51 05571 000060 HC60:   60
```

```
| 0072  .MAIN
01
02 05572 054450 OFTOC:  STA     3,RETOF ; OVERFLOW TEST FOR SIGN AND
03 05573 026775         LDA     1,@LAST4 ; (PREV*10)+LAST < LIMIT DX
04 05574 125102         MOVL    1,1,SZC ; IS THE LAST RUBBED OUT ?
05 05575 002445         JMP     @RETOF  ; OR IS IT THE FIRST ? YES, FIRST
06 05576 010444         ISZ     RETOF
07 05577 125102         MOVL    1,1,SZC
08 05600 002442         JMP     @RETOF  ; YES, RUBOUT
09 05601 034710         LDA     3,HC53  ; NO, IS LAST A + ?
10 05602 136415         SUB#    1,3,SNR ; NO
11 05603 000432         JMP     OFTSI   ; YES
12 05604 034706         LDA     3,HC55  ; IS LAST A - ?
13 05605 136415         SUB#    1,3,SNR ; NO
14 05606 000427         JMP     OFTSI   ; YES
15 05607 176400         SUB     3,3     ; AC3:= 0
16 05610 030755         LDA     2,DSIGN ; IS DSIGN = 177777
17 05611 151112         MOVL#   2,2,SZC ; NO
18 05612 054753         STA     3,DSIGN ; YES, SET FIRST +
19 05613 030754         LDA     2,NUMB4 ; AC2:=PREVIOUS (PREV)
20 05614 155120         MOVZL   2,3     ; PREV*2
21 05615 175120         MOVZL   3,3     ; PREV*4
22 05616 175112         MOVL#   3,3,SZC ; PREV*8>=65536 IF
23 05617 006556         JSR     @XILLG  ; PREV >= 8192
24 05620 173122         ADDZL   3,2,SZC ; PREV*10>=65540 IF
25 05621 006554         JSR     @XILLG  ; PREV >= 6554
26 05622 151112         MOVL#   2,2,SZC ; PREV*10>=32770 IF
27 05623 006552         JSR     @XILLG  ; PREV >= 3277
28 05624 133000         ADD     1,2     ; (PREV*10)+LAST>32767
29 05625 151113         MOVL#   2,2,SNC ; (<=32760)+0...9>32767 ?
30 05626 000412         JMP     OFTRE   ; NO, RETURN
31 05627 034736         LDA     3,DSIGN ; YES, TEST +- 32768,32769
32 05630 175005         MOV     3,3,SNR ; TEST SIGN
33 05631 006544         JSR     @XILLG  ; +32768, +32769
34 05632 151134         MOVZL#  2,2,SZR ; -32768, OK RETURN
35 05633 006542         JSR     @XILLG  ; -32769
36 05634 000404         JMP     OFTRE   ; RETURN
37
38 05635 030730 OFTSI:  LDA     2,DSIGN ; OVERFLOW TEST SIGN
39 05636 151113         MOVL#   2,2,SNC ; IS IT FIRST SIGN ? YES
40 05637 006536         JSR     @XILLG  ; NO
41 05640 010402 OFTRE:  ISZ     RETOF   ; PASS RUBOUT/FIRST RETURN
42 05641 002401         JMP     @RETOF  ; RETURN
43
44 05642 000000 RETOF:  0               ; RETURN ADDRESS
```

```
| 0073  .MAIN
01
02                ; CHARACTER-INPUT-ROUTINE
03
04 05643 056537 GETSC:  STA     3,@XINRT
05 05644 126000         ADC     1,1       ; AC1:=177777
06 05645 046440         STA     1,@LAST1  ; SET LAST1
07 05646 126400         SUB     1,1       ; AC1:= 0
08 05647 044435         STA     1,NUMSC   ; CLEAR NUMSC
09 05650 006530 GETS1:  JSR     @XGTCH    ; GET CHAR TO AC0
10 05651 006526         JSR     @XTRMT    ; IS IT A TERMINATOR ?
11 05652 000450         JMP     TERMSC    ; YES
12 05653 006523         JSR     @XDLTE    ; NO - IS IT A DEL CHAR ?
13 05654 000774         JMP     GETS1     ; YES
14 05655 004410         JSR     LETE1     ; NO - IS IT LEGAL ?
15 05656 004430         JSR     OFTSC     ; IT'S A LEGAL CHAR, TEST OVERFLOW
16 05657 000403         JMP     DELSC     ; FIRST
17 05660 000402         JMP     DELSC     ; LAST IS RUBOUT
18 05661 004436         JSR     PLASC     ; PLACE LAST IN NUMSC
19 05662 042423 DELSC:  STA     0,@LAST1  ; STORE NEW CHAR IN LAST1
20 05663 006516         JSR     @XTYPN    ; TYPE NEW CHARACTER
21 05664 000764         JMP     GETS1     ; GET NEXT INPUT
22
23                ; SUBROUTINES USED BY GETSC
24
25 05665 054413 LETE1:  STA     3,LERE1   ; LEGAL TEST
26 05666 030413         LDA     2,H1C40
27 05667 034414         LDA     3,H1C11
28 05670 024412         LDA     1,H1C33
29 05671 106415         SUB#    0,1,SNR   ; IS AC0 = 33 ?
30 05672 002406         JMP     @LERE1    ; YES
31 05673 116415         SUB#    0,3,SNR   ; NO - IS AC0 = 11 ?
32 05674 002404         JMP     @LERE1    ; YES
33 05675 112032         ADCZ#   0,2,SZC   ; NO - IS AC0 >= 40 ?
34 05676 006477         JSR     @XILLG    ; NO
35 05677 002401         JMP     @LERE1    ; YES - IT IS A LEGAL CHAR
36 05700 000000 LERE1:  0
37
38 05701 000040 H1C40:  40
39 05702 000033 H1C33:  33
40 05703 000011 H1C11:  11
41 05704 000000 NUMSC:  0
42 05705 005424 LAST1:  LASTN
43
44 05706 026777 OFTSC:  LDA     1,@LAST1  ; OVERFLOW TEST
45 05707 125102         MOVL    1,1,SZC   ; IS THE LAST CHAR RUBBED OUT ?
46 05710 001400         JMP     0,3       ; OR IS IT THE FIRST ? YES, FIRST
47 05711 125102         MOVL    1,1,SZC
48 05712 001401         JMP     1,3       ; YES, RUBOUT
49 05713 030771         LDA     2,NUMSC   ; NUMSC = 0 FOR FIRST
50 05714 151004         MOV     2,2,SZR   ; IS IT THE FIRST CHAR ?
51 05715 006460         JSR     @XILLG    ; NO, OVERFLOW
52 05716 001402         JMP     2,3       ; YES, RETURN
53
54 05717 032766 PLASC:  LDA     2,@LAST1  ; PLACE LAST CHAR
55 05720 050764         STA     2,NUMSC
56 05721 001400         JMP     0,3
57
58 05722 004764 TERMSC: JSR     OFTSC     ; TERMINATION, TEST OVERFLOW
59 05723 002461         JMP     @XCNTR    ; TERM IS FIRST
60 05724 000402         JMP     TERSC     ; LAST IS RUBOUT
61 05725 004772         JSR     PLASC     ; STORE LAST CHAR IN NUMSC
62 05726 024756 TERSC:  LDA     1,NUMSC   ; MOVE CHAR TO DIGIN
63 05727 044075         STA     1,DIGIN
64 05730 002453         JMP     @XINTR    ; OUTPUT TERM CHAR
```

```
| 0074  .MAIN
01
02                ; OCTAL-NUMBER-INPUTROUTINE
03
04 05731 056451 GETOK:  STA     3,@XINRT
05 05732 126000         ADC     1,1      ; AC1:=177777
06 05733 046506         STA     1,@LAST2
07 05734 126400         SUB     1,1      ; AC1:= 0
08 05735 044505         STA     1,NUMB2
09 05736 006442 GETO1:  JSR     @XGTCH   ; GET CHAR TO AC0
10 05737 006440         JSR     @XTRMT   ; IS IT A TERMINATOR ?
11 05740 000503         JMP     TERMOK   ; YES
12 05741 006435         JSR     @XDLTE   ; NO - IS IT A DEL CHAR ?
13 05742 000774         JMP     GETO1    ; YES
14 05743 004412         JSR     LETE2    ; NO - IS IT LEGAL ?
15 05744 004441         JSR     CHRA2    ; IT IS A LEGAL DIGIT
16 05745 004445         JSR     OFTE2    ; TEST FOR OVERFLOW
17 05746 000403         JMP     DELOK    ; FIRST
18 05747 000402         JMP     DELOK    ; LAST IS RUBOUT
19 05750 004460         JSR     PLAOK    ; ADD LAST2 TO NUMB2 * 8-DEC
20 05751 024423 DELOK:  LDA     1,OKDIG  ; STORE NEW DIGIT IN LAST2
21 05752 046467         STA     1,@LAST2
22 05753 006426 ECHO2:  JSR     @XTYPN   ; TYPE NEW CHARACTER
23 05754 000762         JMP     GETO1    ; GET NEXT CHARACTER
24
25                ; SUBROUTINES USED BY GETOK
26
27 05755 054413 LETE2:  STA     3,LERE2  ; LEGAL TEST
28 05756 034413         LDA     3,H2C40
29 05757 116415         SUB#    0,3,SNR  ; IS AC0 = 40 ?
30 05760 000773         JMP     ECHO2    ; YES - THE CHAR IS A SPACE
31 05761 034411         LDA     3,H2C60  ; NO
32 05762 116032         ADCZ#   0,3,SZC  ; IS AC0 >= 60 ?
33 05763 006412         JSR     @XILLG   ; NO - NOT LEGAL
34 05764 034407         LDA     3,H2C70  ; YES
35 05765 116033         ADCZ#   0,3,SNC  ; IS AC0 < 70 ?
36 05766 006407         JSR     @XILLG   ; NO - NOT LEGAL
37 05767 002401         JMP     @LERE2   ; YES - IT IS A VALID DIGIT
38 05770 000000 LERE2:  0
39
40 05771 000040 H2C40:  40
41 05772 000060 H2C60:  60
42 05773 000070 H2C70:  70
43 05774 000000 OKDIG:  0
44
45 05775 005367 XILLG:  ILLEG
46 05776 005335 XDLTE:  DELTE
47 05777 005407 XTRMT:  TERMT
48 06000 005246 XGTCH:  GETCH
49 06001 005357 XTYPN:  TYPIN
50 06002 005421 XINRT:  INRET
51 06003 005400 XINTR:  INTER
52 06004 005402 XONTR:  ONTER
53
54 06005 024765 CHRA2:  LDA     1,H2C60
55 06006 111000         MOV     0,2
56 06007 132400         SUB     1,2      ; PUT CHAR IN RANGE 0-7
57 06010 050764         STA     2,OKDIG
58 06011 001400         JMP     0,3
```

```
| 0075  .MAIN
01
02 06012 054415 OFTE2:  STA     3,REOF2 ; OVERFLOW TEST
03 06013 030427         LDA     2,NUMB2
04 06014 026425         LDA     1,@LAST2 ; IS THE LAST DIGIT RUBBED OUT ?
05 06015 125102         MOVL    1,1,SZC ; OR IS IT THE FIRST ?
06 06016 002411         JMP     @REOF2  ; YES, FIRST
07 06017 010410         ISZ     REOF2
08 06020 125102         MOVL    1,1,SZC
09 06021 002406         JMP     @REOF2  ; YES, RUBOUT
10 06022 034416         LDA     3,OF2CO ; NO
11 06023 156433         SUBZ#   2,3,SNC ; IS NUMB2 <= 17777 ?
12 06024 006751         JSR     @XILLG  ; NO - OVERFLOW
13 06025 010402         ISZ     REOF2   ; YES
14 06026 002401         JMP     @REOF2  ; RETURN
15 06027 000000 REOF2:  0
16
17 06030 026411 PLAOK:  LDA     1,@LAST2 ; PLACE LAST DIGIT
18 06031 030411         LDA     2,NUMB2
19 06032 151120         MOVZL   2,2     ; MULTIPLY NUMB2 WITH 8-DEC
20 06033 151120         MOVZL   2,2
21 06034 151120         MOVZL   2,2
22 06035 133000         ADD     1,2     ; ADD LAST DIGIT
23 06036 050404         STA     2,NUMB2
24 06037 001400         JMP     0,3
25
26 06040 017777 OF2CO:  17777
27 06041 005424 LAST2:  LASTN
28 06042 000000 NUMB2:  0
29
30 06043 004747 TERMOK: JSR     OFTE2   ; TERMINATION, TEST OVERFLOW
31 06044 002740         JMP     @XCNTR  ;TERM IS FIRST
32 06045 000402         JMP     TEROK   ; LAST IS RUBOUT
33 06046 004762         JSR     PLAOK   ; ADD LAST DIGIT TO NUMB2
34 06047 024773 TEROK:  LDA     1,NUMB2
35 06050 044075         STA     1,DIGIN
36 06051 002732         JMP     @XINTR  ; OUTPUT TERM CHAR
```

```
| 0076  .MAIN
01
02                ; BINARY-NUMBER-INPUTROUTINE
03
04 06052 056730 GETBI:  STA     3,@XINRT
05 06053 126000         ADC     1,1       ; AC1:=177777
06 06054 046474         STA     1,@LAST3
07 06055 126400         SUB     1,1       ; AC1:= 0
08 06056 044473         STA     1,NUMB3
09 06057 006721 GETB1:  JSR     @XGTCH    ; GET CHAR TO AC0
10 06060 006717         JSR     @XTRMT    ; IS IT A TERMINATOR ?
11 06061 000471         JMP     TERMBI    ; YES
12 06062 006714         JSR     @XDLTE    ; NO - IS IT A DEL CHAR ?
13 06063 000774         JMP     GETB1     ; YES
14 06064 004412         JSR     LETE3     ; NO - IS IT LEGAL ?
15 06065 004431         JSR     CHRA3     ; IT IS A LEGAL DIGIT
16 06066 004435         JSR     OFTE3     ; TEST FOR OVERFLOW
17 06067 000403         JMP     DELBI     ; FIRST
18 06070 000402         JMP     DELBI     ; LAST IS RUBOUT
19 06071 004450         JSR     PLABI     ; ADD LAST3 TO NUMB3 * 2-DEC
20 06072 024423 DELBI:  LDA     1,BIDIG ; STORE NEW DIGIT IN LAST3
21 06073 046455         STA     1,@LAST3
22 06074 006705 ECHO3:  JSR     @XTYPN    ; TYPE NEW CHARACTER
23 06075 000762         JMP     GETB1     ; GET NEXT CHARACTER
24
25                ; SUBROUTINES USED BY GETBI
26
27 06076 054413 LETE3:  STA     3,LERE3 ; LEGAL TEST
28 06077 034413         LDA     3,H3C40
29 06100 116415         SUB#    0,3,SNR ; IS AC0 = 40 ?
30 06101 000773         JMP     ECHO3     ; YES - THE CHAR IS A SPACE
31 06102 034411         LDA     3,H3C60 ; NO
32 06103 116032         ADCZ#   0,3,SZC ; IS AC0 >= 60 ?
33 06104 006671         JSR     @XILLG    ; NO - NOT LEGAL
34 06105 034407         LDA     3,H3C62 ; YES
35 06106 116033         ADCZ#   0,3,SNC; IS AC0 < 62 ?
36 06107 006666         JSR     @XILLG    ; NO - NOT LEGAL
37 06110 002401         JMP     @LERE3    ; YES
38 06111 000000 LERE3:  0
39
40 06112 000040 H3C40:  40
41 06113 000060 H3C60:  60
42 06114 000062 H3C62:  62
43 06115 000000 BIDIG:  0
44
45
46 06116 024775 CHRA3:  LDA     1,H3C60 ; PUT CHAR IN RANGE 0-1
47 06117 111000         MOV     0,2
48 06120 132400         SUB     1,2
49 06121 050774         STA     2,BIDIG
50 06122 001400         JMP     0,3
```

```
| 0077  .MAIN
01
02 06123 054415 OFTE3:  STA     3,REOF3 ; OVERFLOW TEST
03 06124 030425         LDA     2,NUMB3
04 06125 026423         LDA     1,@LAST3 ; IS THE LAST DIGIT RUBBED OUT ?
05 06126 125102         MOVL    1,1,SZC ; OR IS IT THE FIRST ?
06 06127 002411         JMP     @REOF3  ; YES, FIRST
07 06130 010410         ISZ     REOF3
08 06131 125102         MOVL    1,1,SZC
09 06132 002406         JMP     @REOF3  ; YES, RUBOUT
10 06133 034414         LDA     3,OF3CO ; NO
11 06134 156433         SUBZ#   2,3,SNC ; IS NUMB3 <= 177 ?
12 06135 006640         JSR     @XILLG  ; NO - OVERFLOW
13 06136 010402         ISZ     RECF3   ; YES
14 06137 002401         JMP     @REOF3  ; RETURN
15 06140 000000 REOF3:  0
16
17 06141 026407 PLABI:  LDA     1,@LAST3 ; PLACE LAST DIGIT
18 06142 030407         LDA     2,NUMB3
19 06143 151120         MOVZL   2,2     ; MULTIPLY WITH 2-DEC
20 06144 133000         ADD     1,2     ; ADD LAST DIGIT
21 06145 050404         STA     2,NUMB3
22 06146 001400         JMP     0,3
23
24 06147 000177 OF3CO:  177
25 06150 005424 LAST3:  LASTN
26 06151 000000 NUMB3:  0
27
28 06152 004751 TERMBI: JSR     OFTE3   ; TERMINATION, TEST OVERFLOW
29 06153 002631         JMP     @XONTR  ; TERM IS FIRST
30 06154 000402         JMP     TERBI   ; LAST IS RUBOUT
31 06155 004764         JSR     PLABI   ; ADD LAST DIGIT TO NUMB3
32 06156 024773 TERBI:  LDA     1,NUMB3 ; MOVE THE BINARY NUMBER TO DIGIN
33 06157 044075         STA     1,DIGIN
34 06160 002623         JMP     @XINTR  ; OUTPUT TERM CHAR
```

```
I 0078  .MAIN
01
02                ; TEXT-INPUT-ROUTINE
03
04 06161 056555 GETTX:  STA     3,@YINRT
05 06162 126000         ADC     1,1       ; AC1:=177777
06 06163 046543         STA     1,@LAST5
07 06164 126400         SUB     1,1       ; AC1:= 0
08 06165 044543         STA     1,TXCOU
09 06166 006543 GETT1:  JSR     @YGTCH    ; GET CHAR TO AC0
10 06167 006543         JSR     @YTRMT    ; IS IT A TERMINATOR ?
11 06170 000551         JMP     TERMXT    ; YES
12 06171 006542         JSR     @YDLTE    ; NO - IS IT A DEL CHAR ?
13 06172 000774         JMP     GETT1     ; YES
14 06173 004410         JSR     LETE5     ; NO - IS IT LEGAL ?
15 06174 004423         JSR     OFTTX     ; TEST FOR OVERFLOW
16 06175 000403         JMP     DELTX     ; FIRST
17 06176 000402         JMP     DELTX     ; LAST IS RUBOUT
18 06177 004505         JSR     PLATX     ; STORE LAST5 IN TEXTBUFFER
19 06200 042526 DELTX:  STA     0,@LAST5  ; STORE NEW CHAR IN LAST5
20 06201 006533         JSR     @YTYPN    ; TYPE NEW CHAR
21 06202 000764         JMP     GETT1     ; GET NEXT CHARACTER
22
23                ; SUBROUTINES USED BY GETTX
24
25 06203 054410 LETE5:  STA     3,LERE5 ; LEGAL TEST
26 06204 034410         LDA     3,H5C11
27 06205 030410         LDA     2,H5C40
28 06206 116415         SUB#    0,3,SNR ; IS AC0 = 11 ?
29 06207 002404         JMP     @LERE5  ; YES
30 06210 112032         ADCZ#   0,2,SZC ; NO - IS AC0 >= 40 ?
31 06211 006524         JSR     @YILLG  ; NO - THE CHAR IS NOT LEGAL
32 06212 002401         JMP     @LERE5  ; YES
33 06213 000000 LERE5:  0
34
35 06214 000011 H5C11:  11
36 06215 000040 H5C40:  40
37 06216 000117 H5C79:  117               ; TEXTBUFFER LENGTH-1, OKTAL
38
39 06217 026507 OFTTX:  LDA     1,@LAST5 ; OVERFLOW TEST
40 06220 125102         MOVL    1,1,SZC ; IS THE LAST CHAR RUBBED OUT ?
41 06221 001400         JMP     0,3     ; OR IS IT THE FIRST ? YES, FIRST
42 06222 125102         MOVL    1,1,SZC
43 06223 001401         JMP     1,3     ; YES, RUBOUT
44 06224 024504         LDA     1,TXCOU ; NO
45 06225 030771         LDA     2,H5C79
46 06226 132433         SUBZ#   1,2,SNC ; IS TXCOU <= 79-DEC
47 06227 006506         JSR     @YILLG  ; NO - OVERFLOW
48 06230 001402         JMP     2,3     ; YES, RETURN
49
50 06231 006232 TEXIA:  .+1               ; ADDRESS OF TEXTBUFFER
51       000051 TEXIN:  .BLK 51           ; TEXTBUFFER 80 BYTES + CR,LF
52 06303 000000 TEXEN:  0                 ; END OF TEXTBUFFER: NUL CHAR
```

```
| 0079  .MAIN
01
02                                          ; PLACE LAST CHAR
03 06304 054421 PLATX:  STA     3,REPL5 ; LAST5 TO CHAHA OR
04 06305 026421         LDA     1,@LAST5 ; LAST5 + CHAHA TO BUFFER
05 06306 010422         ISZ     TXCOU   ; INCREMENT CHARACTER-COUNTER
06 06307 030421         LDA     2,TXCOU
07 06310 151213         MOVR#   2,2,SNC ; IS TXCOU EVEN ?
08 06311 000403         JMP     STOTX   ; YES
09 06312 044415         STA     1,CHAHA ; NO - MOVE LAST5 TO CHAHA
10 06313 002412         JMP     @REPL5
11 06314 030413 STOTX:  LDA     2,CHAHA ; CHAHA IS CHAR HALF BUFFER
12 06315 125320         MOVZS   1,1
13 06316 147000         ADD     2,1     ; AC1 = "LAST5,CHAHA"
14 06317 030411         LDA     2,TXCOU ; CALCULATE ADRESS OF BUFFEREND+1
15 06320 151220         MOVZR   2,2
16 06321 034710         LDA     3,TEXIA
17 06322 173000         ADD     3,2     ; AC2:= TEXIN+TXCOU/2
18 06323 045377         STA     1,-1,2  ; STORE LAST TWO CHAR INTO BUFFEREND+1
19 06324 002401         JMP     @REPL5
20 06325 000000 REPL5:  0
21
22 06326 005424 LAST5:  LASTN
23 06327 000000 CHAHA:  0               ; THE CHAR BEFORE LAST5
24 06330 000000 TXCOU:  0               ; TEXT COUNTER BYTE ADDRESS
25
26 06331 005246 YGTCH:  GETCH
27 06332 005407 YTRMT:  TERMT
28 06333 005335 YDLTE:  DELTE
29 06334 005357 YTYPN:  TYPIN
30 06335 005367 YILLG:  ILLEG
31 06336 005421 YINRT:  INRET
32 06337 005400 YINTR:  INTER
33 06340 005402 YONTR:  ONTER
34
35 06341 004656 TERMXT: JSR     OFTTX   ; TERMINATION, TEST OVERFLOW
36 06342 002776         JMP     @YONTR  ; TERM IS FIRST
37 06343 000402         JMP     TERTX   ; LAST IS RUBOUT
38 06344 004740         JSR     PLATX   ; PLACE THE CHAR BEFORE TERM CHAR
39 06345 024763 TERTX:  LDA     1,TXCOU ; IN LAST5
40 06346 125213         MOVR#   1,1,SNC ; IS TXCOU EVEN ?
41 06347 000417         JMP     NBEVEN  ; YES
42 06350 020443         LDA     0,H5C15 ; NO
43 06351 101320         MOVZS   0,0
44 06352 030755         LDA     2,CHAHA
45 06353 113000         ADD     0,2     ; AC2 = "CR,CHAR"
46 06354 010754         ISZ     TXCOU   ; INCREMENT CHAR COUNT FOR CR
47 06355 004413         JSR     COUDI
48 06356 004415         JSR     STABU   ; STORE AC2 INTO BUFFER
49 06357 030435         LDA     2,H5C12 ; AC2 = " 0,LF"
50 06360 004410         JSR     COUDI
51 06361 125400         INC     1,1     ; INCREMENT BUFF ADDR FOR 0,LF
52 06362 004411         JSR     STABU   ; STORE AC2 INTO BUFFER
53 06363 030646 OUT5:   LDA     2,TEXIA
54 06364 050075         STA     2,DIGIN ; ADDRESS OF TEXTBUFFER
55 06365 002752         JMP     @YINTR  ; OUTPUT TERM CHAR
56
57 06366 004413 NBEVEN: JSR     TXEND   ; STORE LF,CR,0,0
58 06367 000774         JMP     OUT5    ; TERMINATE
```

```
! 0080  .MAIN
01
02 06370 024740 COUDI:  LDA     1,TXCOU
03 06371 125220         MOVZR   1,1       ; DIVIDE TXCOU WITH 2
04 06372 001400         JMP     0,3
05
06 06373 054405 STABU:  STA     3,RET5
07 06374 034635         LDA     3,TEXIA ; CALCULATE ADRESS OF BUFFEREND
08 06375 137000         ADD     1,3
09 06376 051777         STA     2,-1,3  ; STORE AC2 INTO BUFFER
10 06377 002401         JMP     @RET5
11 06400 000000 RET5:   0
12
13 06401 054411 TXEND:  STA     3,TXNDR
14 06402 030413         LDA     2,HLFCR ; AC2 = "LF,CR"
15 06403 004765         JSR     COUDI
16 06404 125400         INC     1,1
17 06405 004766         JSR     STABU   ; STORE AC2 INTO BUFFER
18 06406 125400         INC     1,1
19 06407 152400         SUB     2,2     ; AC2:= 0
20 06410 004763         JSR     STABU   ; STORE "0,0" INTO BUFFER
21 06411 002401         JMP     @TXNDR
22 06412 000000 TXNDR:  0
23
24 06413 000015 H5C15:  15
25 06414 000012 H5C12:  12
26 06415 005015 HLFCR:  5015
27
28
29
```

```
| 0081  .MAIN
01
02                ;                TEST ROUTINES
03
04                        ;TESTLOOP ROUTINE
05                        ;CALL    SETPX    ;SETP0,SETP1,SETP2
06                        ;                 ;PROGRAM LOOP
07                        ;
08                        ;        EHALT    ;ERROR HALT ROUTINE
09                        ;        LOOP     ;CYCLE LOOP ROUTINE
10                        ;
11                        ;                 ;NEXT TEST CYCLE
12
13                ;SETPX: ;IORST AND SET # OF LOOPS IN
14                        ;FIRST CYCLUS   ERROR CYCLUS IF SWITCH 0
15                        ;SETP0  10!0     10!1
16                        ;SETP1  10!1     10!2
17                        ;SETP2  10!2     10!2
18                ;EHALT: ;IF NOT FIRST LOOP WITH ERROR IN A CYCLE:
19                        ;        DO NOTHING, LOOP
20                        ;IF FIRST LOOP WITH ERROR IN A CYCLE:
21                        ;        PRINT AC0, AC1, AC2 (NOT ON DIS) AND
22                        ;        PRINT PC XXXXXX IF NOT SWITCH 10 (INHIBIT PRINT)
23                        ;IF FIRST LOOP WITH ERROR AT ALL:
24                        ;        HALT IN EACH ERROR WITH AC3=PC
25                        ;        OTHER ACS RELEVANT INFO.
26                        ;ELSE DO NOTHING, LOOP
27                ;LOOP:  ;IF CYCLE NOT FINISHED (# OF LOOP NOT FINISHED):
28                        ;        IF NO ERRORS AT ALL UNTIL NOW:
29                        ;                 IORST, LOOP
30                        ;        IF ERROR AND SWITCH 0 = 0:
31                        ;                 PRINT FAILURE RATE OF LAST CYCLE IF
32                        ;                 SWITCH 11 = 1 AND SWITCH 10 = 0
33                        ;                 PROCEED TO NEXT TEST CYCLE
34                        ;        ELSE:    IORST, LOOP
35                        ;IF CYCLE FINISHED:
36                        ;        IF NO ERRORS AT ALL:
37                        ;                 PROCEED TO NEXT TEST CYCLE
38                        ;        ELSE:    PRINT FAILURE RATE OF LAST CYCLE IF
39                        ;                 SWITCH 11 = 1 AND SWITCH 10 = 0
40                        ;        IF SWITCH 0 = 0:
41                        ;                 PROCEED TO NEXT TEST CYCLE
42                        ;        ELSE:    IORST, LOOP.
43  06416  054522 ENTP0:  STA      3,LOOPR ;INITIALIZE EACH TEST
44  06417  034504         LDA      3,ITRP1
45  06420  054505         STA      3,ITRAG
46  06421  176520         SUBZL    3,3      ;AC3:= 1
47  06422  000412         JMP      ENTCO
48  06423  054515 ENTP1:  STA      3,LOOPR
49  06424  034500         LDA      3,ITRP2
50  06425  054500         STA      3,ITRAG
51  06426  034475         LDA      3,ITRP1
52  06427  000405         JMP      ENTCO
53  06430  054510 ENTP2:  STA      3,LOOPR
54  06431  034473         LDA      3,ITRP2
55  06432  054473         STA      3,ITRAG
56  06433  034471         LDA      3,ITRP2
57  06434  054472 ENTCO:  STA      3,ITR
58  06435  054472         STA      3,ITRCT
59  06436  176400         SUB      3,3      ;AC3:= 0
60  06437  054471         STA      3,EFLAG ;SET FIRST ERROR FLAG = 0
61  06440  054471         STA      3,ERRCT ;SET ERROR COUNT = 0
62  06441  054471         STA      3,ERBCT ;SET ERROR BUFFER COUNT = 0
63  06442  054471         STA      3,EBFLG ;SET FIRST ERROR FLAG BUFFER = 0
64  06443  034501         LDA      3,SETSW ;SET PRINT INHIBIT ON SW 10
65  06444  056477         STA      3,@ISTAC ;IN FUNCTION
66  06445  060300         NIOP     0        ;I/O RESET REPLACED BY NIOP  0.
67  06446  002472         JMP      @LOOPR   ;LOOP ITERATE RETURN
```

```
| 0082  .MAIN
01
02 06447 054465 CYCLE:  STA     3,RETUR ;END OF TEST ITERATION ROUTINE
03 06450 050465         STA     2,CSAV2
04 06451 044465         STA     1,CSAV1 ;SAVE THE ACS'
05 06452 040465         STA     0,CSAV0
06 06453 020457         LDA     0,ERBCT ;ADD BUFFER COUNT
07 06454 024455         LDA     1,ERRCT ;TO ERROR COUNT
08 06455 107000         ADD     0,1
09 06456 044453         STA     1,ERRCT
10 06457 020454         LDA     0,EBFLG ;MOVE FIRST FLAG
11 06460 040450         STA     0,EFLAG ;BUFFER TO FLAG
12 06461 014446         DSZ     ITRCT
13 06462 000416         JMP     CYCTS   ;NOT N TIMES ITERATED, LOOP
14 06463 030445         LDA     2,EFLAG ;CYCLUS FINISHED
15 06464 151005         MOV     2,2,SNR ;ERRORS AT ALL ?
16 06465 000432         JMP     NOEX    ;NO ERRORS
17 06466 004457         JSR     FRATE   ;YES, PRINT FAILURE RATE
18 06467 102400         SUB     0,0
19 06470 040441         STA     0,ERRCT ;RESET ERROR COUNT
20 06471 034434         LDA     3,ITRAG
21 06472 054434         STA     3,ITR
22 06473 054434         STA     3,ITRCT
23 06474 006073         CRESW           ;READS 2 ROUTINE
24 06475 151112         MOVL#   2,2,SZC ;SWITCH 0 ?
25 06476 000412         JMP     CYMOR   ;(1) = LOOP IN ERROR
26 06477 000420         JMP     NOEX    ;(0) = PROCEED TO NEXT TEST
27
28 06500 034430 CYCTS:  LDA     3,EFLAG ;LOOP FINISHED
29 06501 175005         MOV     3,3,SNR ;ERRORS UNTIL NOW ?
30 06502 000406         JMP     CYMOR   ;NO, LOOP
31 06503 006073         CRESW           ;YES, READS 2 ROUTINE
32 06504 151112         MOVL#   2,2,SZC ;SWITCH 0 ?
33 06505 000403         JMP     CYMOR   ;(1) = LOOP IN ERROR
34 06506 004437         JSR     FRATE   ;PRINT FAILURE RATE
35 06507 000410         JMP     NOEX    ;(0)=PROCEED TO NEXT TEST
36
37 06510 060300 CYMOR:  NIOP    0       ;** I/O RESET REPLACED BY A NIOP.
38 06511 176400         SUB     3,3     ;AC3:=0
39 06512 054420         STA     3,ERBCT ;RESET ERROR BUFFER COUNTER
40 06513 020424         LDA     0,CSAV0
41 06514 024422         LDA     1,CSAV1
42 06515 030420         LDA     2,CSAV2 ;RESTORE AC'S
43 06516 002422         JMP     @LOOPR  ;LOOP MORE
44
45 06517 020420 NOEX:   LDA     0,CSAV0 ;EXIT TO NEXT TEST
46 06520 024416         LDA     1,CSAV1
47 06521 030414         LDA     2,CSAV2 ;RESTORE AC'S
48 06522 002412         JMP     @RETUR  ;PROCEED TO NEXT TEST
49
50 06523 000012 ITRP1:  12              ;10|1 LOOP CONSTANT
51 06524 000144 ITRP2:  144             ;10|2 LOOP CONSTANT
52 06525 000000 ITRAG:  0               ;# OF LOOP IF ERROR
53 06526 000000 ITR:    0               ;# OF LOOP IN CYCLUS ACTUAL
54 06527 000000 ITRCT:  0               ;LOOP COUNTER
55 06530 000000 EFLAG:  0               ;FIRST ERROR FLAG FOR ALL CYCLES
56 06531 000000 ERRCT:  0               ;ERROR COUNTER
57 06532 000000 ERBCT:  0               ;ERROR BUFFER COUNTER
58 06533 000000 EBFLG:  0               ;FIRST ERROR FLAG BUFFER
59 06534 000000 RETUR:  0
60 06535 000000 CSAV2:  0
61 06536 000000 CSAV1:  0
62 06537 000000 CSAV0:  0
63 06540 000000 LOOPR:  0
64 06541 000010 ERHSW:  10              ;SWITCH 12
65 06542 000437 IBZOT:  BZOUT
66 06543 000574 ISTAC:  SETAC
67 06544 006073 SETSW:  CRESW           ;FOR PRINT INHIBIT ROUTINE
```

```
| 0083  .MAIN
01
02 06545 054432 FRATE:  STA     3,FRATR ;PRINT FAILURE RATE
03 06546 006073         CRESW           ;IF SWITCH 10 = 0
04 06547 024433         LDA     1,INHSW ;AND SWITCH 11 = 1
05 06550 133414         AND#    1,2,SZR
06 06551 002426         JMP     @FRATR  ;INHIBIT PRINTOUT
07 06552 024431         LDA     1,FRASW
08 06553 133415         AND#    1,2,SNR
09 06554 000420         JMP     NORAT   ;PRINT CRLF ONLY
10 06555 020423         LDA     0,CH40
11 06556 006041         CCHAR
12 06557 024752         LDA     1,ERRCT
13 06560 030744         LDA     2,ITRP2
14 06561 006066         MULTI           ;AC1*AC2
15 06562 030744         LDA     2,ITR
16 06563 034744         LDA     3,ITRCT
17 06564 172400         SUB     3,2     ;AC2:=# OF LOOPS TILL NOW
18 06565 006070         DIVID           ;AC0,AC1/AC2
19 06566 006053         CTDEC
20 06567 006057         CDDEC           ;PRINT VALUE
21 06570 020411         LDA     0,PCENT ;EXAMPLE:       89%
22 06571 006045         CDOUT           ;VALUE=ERRCT*100/(ITR-ITRCT)
23 06572 006041         CCHAR
24 06573 006047         CDATT
25 06574 006040 NORAT:  CMESS           ;PRINT CRLF WITH MESS TO WAIT
26 06575 004743         MCRLF           ;FOR DEVICE READY BEFORE NEXT IORST.
27 06576 002401         JMP     @FRATR
28 06577 000000 FRATR:  0
29 06600 000040 CH40:   40
30 06601 000245 PCENT:  245
31 06602 000040 INHSW:  40              ;SWITCH 10
32 06603 000020 FRASW:  20              ;SWITCH 11
33
34              MHEAD:  .TXT |AC0<11>AC1<11>AC2<15><12>PC<40>|
   06604 041501
   06605 004460
   06606 041501
   06607 004461
   06610 041501
   06611 006462
   06612 050012
   06613 020103
   06614 000000
35                      ;"AC0<11>AC1<11>AC2<15><12>PC<40>"
36
37              DHEAD:  .TXT |PC<40>|   ;"PC<40>"
   06615 041520
   06616 000040
38
39 06617 054715 ERROR:  STA     3,RETUR ;ERROR SUBROUTINE
40 06620 034711         LDA     3,ERRCT
41 06621 175005         MOV     3,3,SNR ;FIRST ERROR LOOP ?
42 06622 000404         JMP     ERRO1
43 06623 010707 ERRET:  ISZ     ERBCT   ;COUNT BUFFER ERROR COUNTER
44 06624 002710         JMP     @RETUR  ;EXIT
45 06625 002707         JMP     @RETUR  ;EXIT IF SKIP
```

```
| 0084  .MAIN
01
02 06626 040711 ERR01:  STA     0,CSAV0 ;FIRST ERROR LOOP !
03 06627 044707          STA     1,CSAV1 ;(IN FIRST CYCLUS
04 06630 050705          STA     2,CSAV2 ;OR ERROR CYCLUS)
05 06631 006073          CRESW           ;READS 2 ROUTINE
06 06632 024750          LDA     1,INHSW ;SWITCH 10
07 06633 133414          AND#    1,2,SZR
08 06634 000455          JMP     ERR03   ;INHIBIT PRINTOUT
09 06635 006043          CCRLF
10 06636 024701          LDA     1,CSAV0
11 06637 006052          CTOCT
12 06640 024676          LDA     1,CSAV1
13 06641 006052          CTOCT
14 06642 024673          LDA     1,CSAV2
15 06643 006052          CTOCT           ;PRINT AC'S ONLY AT TTY, LPT
16 06644 006043          CCRLF           ;PRINT CARRIAGE
17 06645 006040          CMESS           ;PRINT HEADER
18 06646 006004          MHEAD
19 06647 006046          CDICL
20 06650 006044          CDISP
21 06651 006015          DHEAD
22 06652 020662          LDA     0,RETUR
23 06653 126000          ADC     1,1
24 06654 107000          ADD     0,1
25 06655 006052          CTOCT
26 06656 006056          CDOCT           ;PRINT PC OF ERROR
27 06657 006063          JSR     @IBZOT  ;WAIT FOR LPT/TTY BEFORE NEXT IORST
28 06660 020650          LDA     0,EFLAG
29 06661 101005          MOV     0,0,SNR
30 06662 000406          JMP     ERR02   ;FIRST ERROR LOOP AT ALL
31 06663 006047          CDATT
32 06664 020653          LDA     0,CSAV0
33 06665 024651          LDA     1,CSAV1
34 06666 030647          LDA     2,CSAV2 ;RESTORE ACS
35 06667 000734          JMP     ERRET
36
37 06670 006050 ERR02:  CHAAT           ;FIRST ERROR LOOP AT ALL: HALT
38 06671 126000          ADC     1,1     ;AC1:=177777
39 06672 044641          STA     1,EBFLG ;SET FIRST ERROR BUFFER FLAG
40 06673 006073          CRESW           ;READS 2 ROUTINE
41 06674 034645          LDA     3,ERHSW
42 06675 157404          AND     2,3,SZR ;SW 12 ?
43 06676 000407          JMP     ERRNH   ;NO HALT
44 06677 034635          LDA     3,RETUR
45 06700 137000          ADD     1,3     ;ERROR. AC3=PC OF ERROR
46 06701 030634          LDA     2,CSAV2
47 06702 024634          LDA     1,CSAV1
48 06703 020634          LDA     0,CSAV0
49 06704 063077          HALT            ;OPERATOR-SET SWITCHES!
50 06705 020632 ERRNH:  LDA     0,CSAV0
51 06706 024630          LDA     1,CSAV1
52 06707 030626          LDA     2,CSAV2
53 06710 000713          JMP     ERRET
54
55 06711 020617 ERR03:  LDA     0,EFLAG ;NO PRINTING
56 06712 101005          MOV     0,0,SNR
57 06713 000755          JMP     ERR02   ;FIRST ERROR LOOP AT ALL
58 06714 020623          LDA     0,CSAV0
59 06715 024621          LDA     1,CSAV1
60 06716 030617          LDA     2,CSAV2
61 06717 000704          JMP     ERRET
```

```
| 0085  .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 06720 054412 XSTAA:  STA     3,RXAST
12 06721 010411         ISZ     RXAST
13 06722 025400         LDA     1,0,3   ;AC1=EXPECTED BITS
14 06723 060417 SDEV1:  DIA     0,DEV
15 06724 040407         STA     0,SXAST ;SAVE AC0
16 06725 123400         AND     1,0
17 06726 122415         SUB#    1,0,SNR
18 06727 010403         ISZ     RXAST   ;OK
19 06730 020403         LDA     0,SXAST ;RESTORE STATUS
20 06731 002401         JMP     @RXAST
21 06732 000000 RXAST:  0
22 06733 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 06734 054407 XSTAN:  STA     3,RXNST
35 06735 010406         ISZ     RXNST
36 06736 025400         LDA     1,0,3   ;AC1=NONEXPECTED STATUS
37 06737 060417 SDEV2:  DIA     0,DEV   ;AC0=ACTUAL STATUS
38 06740 107415         AND#    0,1,SNR
39 06741 010402         ISZ     RXNST   ;OK
40 06742 002401         JMP     @RXNST
41 06743 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 06744 054413 XSTAW:  STA     3,RXWST
52 06745 010412         ISZ     RXWST
53 06746 025400         LDA     1,0,3
54 06747 006073         CRESW           ;READS 2 ROUTINE
55 06750 034410         LDA     3,SMASK ;MASK FOR SWITCH REGISTER
56 06751 157400         AND     2,3
57 06752 167000         ADD     3,1     ;AC1=EXP STATUS
58 06753 060417 SDEV3:  DIA     0,DEV   ;AC0=ACTUAL STATUS
59 06754 106415         SUB#    0,1,SNR
60 06755 010402         ISZ     RXWST   ;OK
61 06756 002401         JMP     @RXWST
62 06757 000000 RXWST:  0
63 06760 000000 SMASK:  000000          ;CHANGE MASK TO ONES FOR
64                                      ;THOSE BITS TO USE FROM SW.
```

```
| 0086  .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 06761 054414 XSTAP:  STA     3,RXPST
13 06762 010413         ISZ     RXPST
14 06763 010412         ISZ     RXPST
15 06764 025400         LDA     1,0,3   ;AC1:=EXPECTED PART
16 06765 031401         LDA     2,1,3   ;AC2:=MASK
17 06766 147400         AND     2,1
18 06767 060417 SDEV4:  DIA     0,DEV   ;AC0:=ACTUAL STATUS
19 06770 115000         MOV     0,3     ;AC3:=ACTUAL STATUS FOR MASK
20 06771 157400         AND     2,3
21 06772 136415         SUB#    1,3,SNR
22 06773 010402         ISZ     RXPST   ;OK
23 06774 002401         JMP     @RXPST
24 06775 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 06776 054407 XSTAS:  STA     3,RXSST
36 06777 010406         ISZ     RXSST
37 07000 025400         LDA     1,0,3   ;AC1:=EXPECTED STATUS
38 07001 060417 SDEV5:  DIA     0,DEV   ;AC0:=ACTUAL STATUS
39 07002 107414         AND#    0,1,SZR
40 07003 010402         ISZ     RXSST   ;OK
41 07004 002401         JMP     @RXSST
42 07005 000000 RXSST:  0
43
44                          ;ROUTINE LOOP REPORT
45
46 07006 054461 XLORE:  STA     3,RPASS
47 07007 006046         CDICL
48 07010 006043         CCRLF
49 07011 024456         LDA     1,RPASS
50 07012 152520         SUBZL   2,2     ;AC2:=1
51 07013 146400         SUB     2,1     ;SUBTRACT 1 FROM JSR ADDR
52 07014 006056         CDOCT
53 07015 006052         CTOCT           ;PRINT ADDR
54 07016 006044         CDISP
55 07017 007077         MLORE
56 07020 006040         CMESS
57 07021 007077         MLORE           ;XXXXXX LOOP-ADDR
58 07022 006047         CDATT
59 07023 006444         JSR     @RPASS  ;RETURN TO START LOOP
```

```
| 0087  .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 07024 054443 XPASS:  STA     3,RPASS
13 07025 014443         DSZ     PASSB   ;RUN COUNT DOWN
14 07026 002441         JMP     @RPASS  ;NO MESSAGE
15 07027 020442         LDA     0,PASSC
16 07030 040440         STA     0,PASSB ;INITIALIZE RUN COUNT
17 07031 006046         CDICL           ;WRITE PASS MESSAGE
18 07032 006043         CCRLF
19 07033 006040         CMESS
20 07034 005207         MX2SP
21 07035 010435         ISZ     PASSN   ;COUNT PASS #
22 07036 024434         LDA     1,PASSN
23 07037 006053         CTDEC
24 07040 006057         CDDEC
25 07041 006044         CDISP
26 07042 007061         DPASS
27 07043 006040         CMESS
28 07044 007050         MPASS           ; . PASS OF 10 RUNS
29 07045 006043         CCRLF
30 07046 006047         CDATT
31 07047 002420         JMP     @RPASS  ;RETURN
32
33                  MPASS:  .TXT !. PASS OF 5 RUNS! ;". PASS OF 5 RUNS"
   07050 020056
   07051 040520
   07052 051523
   07053 047440
   07054 020106
   07055 020065
   07056 052522
   07057 051516
   07060 000000
34
35                  DPASS:  .TXT !.PASS  5 R!       ;".PASS 5 R"
   07061 050056
   07062 051501
   07063 020123
   07064 032440
   07065 051040
   07066 000000
36
37 07067 000000 RPASS:  0               ;RETURN ADDR
38 07070 000000 PASSB:  0               ;COUNTER FOR # OF RUNS BETWEEN PASS MESS
39 07071 000005 PASSC:  5.              ;# OF RUNS BETWEEN PASS MESS
40 07072 000000 PASSN:  0               ;# OF PASS COUNTER
41 07073 007145 TROTA:  TROTB           ;POINTER TO TABLE FOR BREAKPOINT
42 07074 063077 TROHC:  HALT            ;TROUBLE HALT COMMAND
43 07075 006122 TROLC:  CLORE           ;TROUBLE LOOP REPORT COMMAND
44 07076 000401 TRORC:  JMP     .+1     ;TROUBLE RESET COMMAND
45
46                  MLORE:  .TXT ! LOOP-ADDR!       ;" LOOP-ADDR"
   07077 046040
   07100 047517
   07101 026520
   07102 042101
   07103 051104
   07104 000000
```

```
I 0088  .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 07105 165000 TROHA:  MOV     3,1
27 07106 006072         CSAMS           ;START ADDR MESSAGE
28 07107 036764         LDA     3,@TROTA
29 07110 054023         STA     3,IDX3  ;INIT TABLEPOINTER
30 07111 024763         LDA     1,TROHC ;HALT COMMAND
31 07112 032023 TROHL:  LDA     2,@IDX3
32 07113 151015         MOV#    2,2,SNR ;END OF TABLE ?
33 07114 000427         JMP     TROEN   ;YES, RETURN
34 07115 045000         STA     1,0,2   ;STORE HALT IN LOOP START
35 07116 000774         JMP     TRCHL   ;AGAIN
36
37                          ;TROUBLE LOOP REPORT
38 07117 165000 TROLO:  MOV     3,1
39 07120 006072         CSAMS           ;START ADDR MESSAGE
40 07121 036752         LDA     3,@TROTA
41 07122 054023         STA     3,IDX3  ;INIT TABLEPOINTER
42 07123 024752         LDA     1,TROLC ;LOOP REPORT COMMAND
43 07124 032023 TROLL:  LDA     2,@IDX3
44 07125 151015         MOV#    2,2,SNR ;END OF TABLE ?
45 07126 000415         JMP     TROEN   ;YES, RETURN
46 07127 045000         STA     1,0,2   ;STORE CLORE IN LOOP START
47 07130 000774         JMP     TROLL   ;AGAIN
48
49                          ;TROUBLE RESET
50 07131 165000 TRORE:  MOV     3,1
51 07132 006072         CSAMS           ;START ADDR MESSAGE
52 07133 036740         LDA     3,@TROTA
53 07134 054023         STA     3,IDX3  ;INIT TABLEPOINTER
54 07135 024741         LDA     1,TRORC ;JMP .+1 COMMAND
55 07136 032023 TRORL:  LDA     2,@IDX3
56 07137 151015         MOV#    2,2,SNR ;END OF TABLE ?
57 07140 000403         JMP     TROEN   ;YES, RETURN
58 07141 045000         STA     1,0,2   ;STORE JMP .+1 IN LOOP START
59 07142 000774         JMP     TRCRL   ;AGAIN
60 07143 002401 TROEN:  JMP     @.+1
61 07144 004740         SWISA           ;RESTART MAIN PROGRAM
62
63
64                ;TAPE 4
65
66                .EOT
```

```
  0089  .MAIN
01                ;TAPE 5
02
03 07145 007145 TROTB:  .               ;ADDRES TABLE FOR LOOPS WITH
04                                      ;AUTO BREAKPOINTS.
05 07146 010005         B0
06 07147 010034         B1
07 07150 010052         B3
08 07151 010072         B4
09 07152 010124         B10
10 07153 010201         B30
11 07154 010236         B60
12 07155 010307         B91
13 07156 010402         B210
14 07157 010430         B212
15 07160 011161         DRTES
16 07161 011165         AGAIN
17 07162 011231         DLYAD
18 07163 011474         E10
19 07164 011516         E30
20 07165 011554         E50
21 07166 000000         0               ;END OF TABLE
22
23                PROG:   .TXT !<12>EXT. MEMORY TEST!     ;"EXT. MEMORY TEST"
   07167 042412
   07170 052130
   07171 020056
   07172 042515
   07173 047515
   07174 054522
   07175 052040
   07176 051505
   07177 000124
24
```

```
| 0090  .MAIN
01 07200 007203 RENOQ:  XMTES+2
02
03 07201 165000 XMTES:  MOV     3,1
04 07202 006072         CSAMS           ;PRINT START ADDR.
05 07203 004414         JSR     INIT    ;INITIATE
06 07204 004460         JSR     QUES    ;ANSWERE QUESTIONS
07 07205 004401         JSR     .+1
08 07206 054772         STA     3,RENOQ ;NO QUESTIONS, RESTART ADDR.
09 07207 006423         JSR     @CMTES  ;PERFORM TESTLOOPS AS ONE ROUTINE
10 07210 006123         CPASS           ;CALL PASS ADMINI.
11 07211 000776         JMP     .-2     ;LOOP
12
13 07212 165000 XANSW:  MOV     3,1
14 07213 006072         CSAMS           ;PRINT START ADDR.
15 07214 004403         JSR     INIT    ;INITIA.
16 07215 034763         LDA     3,RENOQ
17 07216 001401         JMP     1,3
18
19
20 07217 054412 INIT:   STA     3,INITR ;INITIATION ROUTINE.
21 07220 026406         LDA     1,@INIPC
22 07221 046406         STA     1,@INIPB;INIT. RUN-COUNTER.
23 07222 102440         SUBO    0,0
24 07223 042405         STA     0,@INIPN;PASSCOUNTER:=0
25 07224 062677         IORST           ;RESET I/O
26 07225 002404         JMP     @INITR
27
28 07226 007071 INIPC:  PASSC
29 07227 007070 INIPB:  PASSB
30 07230 007072 INIPN:  PASSN
31 07231 000000 INITR:  0
32 07232 007606 CMTES:  MTEST
33
34 07233 054424 XDIST:  STA     3,DISRT ;ROUTINE TO DISTURB THE MEMORY
35 07234 030133         LDA     2,FIPOS ;THE MEMORY IS DISTURBED BY
36 07235 024426         LDA     1,C1X   ;REFFERANCING LOCATIONS 101, 202, 303..
37 07236 133400         AND     1,2     ;ETC. 512 TIMES. THIS PRODUCES
38 07237 151112         MOVL#   2,2,SZC ;1024 READ/WRITE DISTURBS.
39 07240 076701         DICP    3,1     ;SET MEM EXT FLAG
40 07241 020127         LDA     0,LPOS
41 07242 024417         LDA     1,C17Z
42 07243 107400         AND     0,1
43 07244 024416         LDA     1,C7777
44 07245 123000         ADD     1,0
45 07246 024412 XDIS1:  LDA     1,C101
46 07247 133000         ADD     1,2
47 07250 176400         SUB     3,3
48 07251 025000         LDA     1,0,2
49 07252 175704         INCS    3,3,SZR
50 07253 000776         JMP     .-2
51 07254 142032         ADCZ#   2,0,SZC
52 07255 000771         JMP     XDIS1
53 07256 002401         JMP     @DISRT
54
55 07257 000000 DISRT:  0
56 07260 000101 C101:   101
57 07261 170000 C17Z:   170000
58 07262 007777 C7777:  7777
59 07263 100000 C1X:    100000
```

```
| 0091  .MAIN
01 07264 054474 QUES:   STA     3,RETQU ;FIRST AND LAST MEMORY LOC. ?
02 07265 006071         CQUES
03 07266 007505         FTMES           ;FIRST MEMORY LOC.
04 07267 007516         FDMES           ;F MEM LOC
05 07270 007357         FANS            ;LASTP
06 07271 006054         CTZOC
07 07272 006060         CDZOC
08 07273 006104         CGTOK           ;READ ANSW.
09 07274 000402         JMP     .+2     ;SUGGESTED ACCEPTED
10 07275 000770         JMP     QUES+1  ;ERROR
11 07276 020075         LDA     0,DIGIN ;ANSWERE INPUT'ED
12 07277 024460         LDA     1,FANS  ;MIN. ANSW
13 07300 122433         SUBZ#   1,0,SNC ;CHECK ANSW
14 07301 000764         JMP     QUES+1
15 07302 040125         STA     0,FIRST
16
17 07303 006071 QUES1:  CQUES
18 07304 007524         LTMES           ;LAST MEMOTY LOC.
19 07305 007534         LDMES           ;L MEM LOC
20 07306 007542         LANS            ;77577
21 07307 006054         CTZOC
22 07310 006060         CDZOC
23 07311 006104         CGTOK           ;READ ANSW.
24 07312 000402         JMP     .+2     ;SUGGESTED ACCEPTED
25 07313 000770         JMP     QUES1   ;ERROR
26 07314 020075         LDA     0,DIGIN ;ANSWERE INPUT'ED
27 07315 024125         LDA     1,FIRST
28 07316 122433         SUBZ#   1,0,SNC ;
29 07317 000432         JMP     LERROR  ;ERROR, FIRST GT. LAST
30 07320 040126         STA     0,LAST
31
32
33 07321 006040 QUES2:  CMESS
34 07322 007365         TYTME           ;"CHOOSE MEMOTY TYPE:
35 07323 006046         CDICL           ; MOVE 2 OR MEM 717/718 : 0
36 07324 006044         CDISP           ; MEM 709/711           : 1
37 07325 007436         TYDM1
38 07326 006044         CDISP
39 07327 007447         TYDM2
40 07330 006044         CDISP
41 07331 007460         TYDM3
42 07332 006071 QUES3:  CQUES           ;MEMORY TYPE ?
43 07333 007471         TTMES           ;DR 103 OR DR 106,
44 07334 007500         TDMES           ;DR 106==NOVA-TYPES.
45 07335 007361         TSANS           ;1
46 07336 006054         CTZOC
47 07337 006060         CDZOC
48 07340 006103         CGTBI           ;READ ANSWERE
49 07341 000402         JMP     .+2     ;SUGGESTED ACCEPTED
50 07342 000770         JMP     QUES3   ;ERROR
51 07343 020075         LDA     0,DIGIN ;ANSWERE INDPUT'ED
52 07344 024416         LDA     1,DR103
53 07345 101005         MOV     0,0,SNR
54 07346 024415         LDA     1,DR106 ;DR106
55 07347 044134         STA     1,BMASK ;DR103
56 07350 002410         JMP     @RETQU  ;RETURN TO MAIN PROGRAM.
57
58 07351 006040 LERROR: CMESS
59 07352 007556         EMESS
60 07353 006046         CDICL           ;CLEAR DISPLAY
61 07354 006044         CDISP
62 07355 007545         EDMES
63 07356 002406         JMP     @CXMTES ;RESTART
64
65 07357 011614 FANS:   LASTP
66 07360 000000 RETQU:  0
```

```
I 0092   .MAIN
01 07361 000001 TSANS:  1
02 07362 010001 DR103:  10001           ;BIT 15 AND 3
03 07363 000420 DR106:  420             ;BIT 11 AND 7
04 07364 007203 CXMTES: XMTES+2
05              TYTME:  .TXT    |<12><15>CHOOSE MEMORY TYPE:
   07365 006412
   07366 044103
   07367 047517
   07370 042523
   07371 046440
   07372 046505
   07373 051117
   07374 020131
   07375 054524
   07376 042520
06 07377 004472                         <12><15>NOVA 2 OR MEM 717/718 : 0
   07400 004411
   07401 006412
   07402 047516
   07403 040526
   07404 031040
   07405 047440
   07406 020122
   07407 042515
   07410 020115
   07411 030467
   07412 027467
   07413 030467
   07414 020070
   07415 020072
07 07416 004460                         <12><15>MEM 709/711           : 1!
   07417 004411
   07420 006412
   07421 042515
   07422 020115
   07423 030067
   07424 027471
   07425 030467
   07426 020061
   07427 020040
   07430 020040
   07431 020040
   07432 020040
   07433 020040
   07434 020072
   07435 000061
08              TYDM1:  .TXT    |<12>CHOOSE MEM TYPE:!
   07436 041412
   07437 047510
   07440 051517
   07441 020105
   07442 042515
   07443 020115
   07444 054524
   07445 042520
   07446 000072
09              TYDM2:  .TXT    |<12>NOVA,717/718: 0|
   07447 047012
   07450 053117
   07451 026101
   07452 030467
   07453 027467
   07454 030467
   07455 035070
   07456 030040
   07457 000000
```

```
| 0093  .MAIN
01
02                      TYDM3:  .TXT    !<12>MEM 709/711 : 1!
   07460 046412
   07461 046505
   07462 033440
   07463 034460
   07464 033457
   07465 030461
   07466 035040
   07467 030440
   07470 000000
03                      TTMES:  .TXT    !MEMORY TYPE:!  ;DR 103
   07471 042515
   07472 047515
   07473 054522
   07474 052040
   07475 050131
   07476 035105
   07477 000000
04                      TDMES:  .TXT    !MEM TYPE:!     ;DR 103
   07500 042515
   07501 020115
   07502 054524
   07503 042520
   07504 000072
05                      FTMES:  .TXT !FIRST MEMORY LOC!
   07505 044506
   07506 051522
   07507 020124
   07510 042515
   07511 047515
   07512 054522
   07513 046040
   07514 041517
   07515 000000
06                      FDMES:  .TXT !<12>F MEM LOC!
   07516 043012
   07517 046440
   07520 046505
   07521 046040
   07522 041517
   07523 000000
07
08                      LTMES:  .TXT !LAST MEMORY LOC!
   07524 040514
   07525 052123
   07526 046440
   07527 046505
   07530 051117
   07531 020131
   07532 047514
   07533 000103
09                      LDMES:  .TXT !<12>L MEM LOC!
   07534 046012
   07535 046440
   07536 046505
   07537 046040
   07540 041517
   07541 000000
10 07542 077577 LANS:   077577  ;32K-1
11 07543 000177 C177:   177
12 07544 000200 C200:   200
```

```
| 0094  .MAIN
01
02                EDMES:  .TXT    !PARM.ERR.SEE.7.1!
   07545 040520
   07546 046522
   07547 042456
   07550 051122
   07551 051456
   07552 042505
   07553 033456
   07554 030456
   07555 000000
03
04                EMESS:  .TXT !<12>PARAM. ERROR: SEE 7.1.1 IN PROGRAM DESCRIPTION
   07556 050012
   07557 051101
   07560 046501
   07561 020056
   07562 051105
   07563 047522
   07564 035122
   07565 051440
   07566 042505
   07567 033440
   07570 030456
   07571 030456
   07572 044440
   07573 020116
   07574 051120
   07575 043517
   07576 040522
   07577 020115
   07600 042504
   07601 041523
   07602 044522
   07603 052120
   07604 047511
   07605 000116
05
06
07
08 07606 054141 MTEST:  STA     3,RTEST ;MEMORY TEST
09 07607 020125         LDA     0,FIRST
10 07610 101132         MOVZL#  0,0,SZC
11 07611 000411         JMP     MEXT    ;ONLY TEST IN MEM. EXTENSION.
12 07612 024126         LDA     1,LAST
13 07613 030415         LDA     2,C32KM
14 07614 132433         SUBZ#   1,2,SNC
15 07615 145000         MOV     2,1     ;AC1:= MIN(LAST,077577)
16 07616 044127         STA     1,LPOS  ;LPOS, LAST TEST LOC.
17 07617 040133         STA     0,FIPOS ;FIPOS, FIRST TEST LOC.
18
19 07620 004560         JSR     TESTP   ;TEST IN FIRST TO MIN(LAST,77577,HMEND)
20 07621 020406         LDA     0,C32K
21 07622 040133 MEXT:   STA     0,FIPOS
22 07623 024126         LDA     1,LAST
23 07624 044127         STA     1,LPOS
24 07625 004553         JSR     TESTP   ;TEST IN 100000/FIRST TO LAST.
25
26 07626 002141         JMP     @RTEST  ;RETURN
27
28 07627 100000 C32K:   100000
29 07630 077577 C32KM:  077577
```

```
I 0095  .MAIN
01
02          010000              .LOC    10000
03  10000   054130  TESTP:      STA     3,RETES
04  10001   030133              LDA     2,FIPOS
05  10002   020127              LDA     0,LPOS
06  10003   142433              SUBZ#   2,0,SNC
07  10004   002130              JMP     @RETES  ;RETURN IF FIPOS GT. LPOS
08
09  10005   000401  B0:         JMP     .+1     ;ADDRES TEST, ACCES MEMORY
10  10006   030133              LDA     2,FIPOS ;BY INDEXING.
11  10007   151132              MOVZL#  2,2,SZC
12  10010   076701              DICP    3,1     ;SET MEM EXT. FLAG
13  10011   151001              MOV     2,2,SKP
14  10012   151400  BM:         INC     2,2
15  10013   006110              SETP0
16  10014   141000              MOV     2,0
17  10015   041000              STA     0,0,2
18  10016   025000              LDA     1,0,2
19  10017   106414              SUB#    0,1,SZR ;CHECK ANSWERE, INDEX 2
20  10020   006114              EHALT           ;MEMORY FAILED
21  10021   006113              LOOP
22  10022   006110              SETP0
23  10023   155000              MOV     2,3
24  10024   041400              STA     0,0,3
25  10025   025400              LDA     1,0,3
26  10026   106414              SUB#    0,1,SZR ;CHECK, INDEX 3
27  10027   006114              EHALT           ;MEMORY FAILED
28  10030   006113              LOOP
29  10031   020127              LDA     0,LPOS
30  10032   142434              SUBZ#   2,0,SZR
31  10033   000757              JMP     BM
32
33
34  10034   000401  B1:         JMP     .+1     ;CHECK ADDRES IN C(ADDRES)
35  10035   030133              LDA     2,FIPOS
36  10036   151400  B2:         INC     2,2
37  10037   151132              MOVZL#  2,2,SZC
38  10040   076701              DICP    3,1     ;SET MEM EXT FLAG
39  10041   141000              MOV     2,0
40  10042   006110              SETP0
41  10043   025000              LDA     1,0,2   ;CHECK READ
42  10044   106414              SUB#    0,1,SZR
43  10045   006114              EHALT           ;MEMORY FAILED
44  10046   006113              LOOP
45  10047   020127              LDA     0,LPOS
46  10050   142434              SUBZ#   2,0,SZR
47  10051   000765              JMP     B2
48
49  10052   000401  B3:         JMP     .+1     ;COMPLEMENT ADDRESS TEST.
50  10053   030133              LDA     2,FIPOS
51  10054   151132              MOVZL#  2,2,SZC
52  10055   076701              DICP    3,1     ;SET MEM EXT FLAG.
53  10056   006112  B3.2:       SETP2
54  10057   140000              COM     2,0     ;AC0:= -,ADDRESS.
55  10060   041000              STA     0,0,2
56  10061   025000              LDA     1,0,2   ;CHECK READ.
57  10062   106414              SUB#    0,1,SZR
58  10063   006114              EHALT           ;MEMORY FAILED.
59  10064   006113              LOOP
60  10065   020127              LDA     0,LPOS
61  10066   112415              SUB#    0,2,SNR
62  10067   000403              JMP     B4      ;FINIS
63  10070   151400              INC     2,2
64  10071   000765              JMP     B3.2
65
```

```
| 0096  .MAIN
01 10072 000401 B4:     JMP     .+1         ;BIT SELECTION TEST,
02                                          ;ALL MEMORY LOCATIONS
03 10073 030133         LDA     2,FIPOS     ;ARE FILLED WITH:
04 10074 151112         MOVL#   2,2,SZC     ;00...01
05 10075 076701         DICP    3,1         ;00...10
06                                          ;.....
07                                          ;.....
08                                          ;10...00
09                                          ;AND THE COMPLIMENT,
10                                          ;(FLOATING ONE AND ZERO,)
11 10076 102520 B5:     SUBZL   0,0         ;AC0:=1
12 10077 006110 B6:     SETP0
13 10100 041000         STA     0,0,2       ;STORE AC0 IN MEM,
14 10101 025000         LDA     1,0,2       ;CHECK READ
15 10102 122414         SUB#    1,0,SZR     ;
16 10103 006114         EHALT               ;MEMORY FAILED
17 10104 006113         LOOP
18 10105 100000         COM     0,0         ;INVERT AC0
19 10106 006110         SETP0
20 10107 041000         STA     0,0,2
21 10110 025000         LDA     1,0,2
22 10111 122434         SUBZ#   1,0,SZR
23 10112 006114         EHALT               ;MEMORY FAILED
24 10113 006113         LOOP
25 10114 100124         COMZL   0,0,SZR
26 10115 000762         JMP     B6
27 10116 020127         LDA     0,LPOS
28 10117 112415         SUB#    0,2,SNR
29 10120 000403         JMP     B8
30 10121 151400         INC     2,2
31 10122 000754         JMP     B5
32
33 10123 000401 B8:     JMP     .+1
34 10124 000401 B10:    JMP     .+1         ;FILL MEMORY WITH ALTANATE
35 10125 102000         ADC     0,0
36 10126 040132         STA     0,FLAG      ;10101010.....10
37 10127 030133         LDA     2,FIPOS     ;01010101.....01
38 10130 020434         LDA     0,CPAT
39 10131 151132         MOVZL#  2,2,SZC
40 10132 076701         DICP    3,1         ;SET MEM EXT FLAG
41 10133 000402         JMP     .+2
42 10134 151400 B11:    INC     2,2
43 10135 041000         STA     0,0,2
44 10136 020127         LDA     1,LPOS
45 10137 146435         SUBZ#   2,1,SNR
46 10140 000407         JMP     B12         ;FINIS
47 10141 151400         INC     2,2
48 10142 104000         COM     0,1
49 10143 045000         STA     1,0,2
50 10144 020127         LDA     1,LPOS
51 10145 146434         SUBZ#   2,1,SZR
52 10146 000766         JMP     B11
53 10147 030133 B12:    LDA     2,FIPOS
54 10150 000402         JMP     .+2
55 10151 151400 B13:    INC     2,2
56 10152 006110         SETP0
57 10153 025000         LDA     1,0,2       ;CHECK READ
58 10154 122434         SUBZ#   1,0,SZR
59 10155 006114         EHALT               ;MEM. CONTAIN NOT THE RIGHT PATTERN.
60 10156 006113         LOOP
61 10157 100000         COM     0,0
62 10160 020127         LDA     1,LPOS
63 10161 146434         SUBZ#   2,1,SZR
64 10162 000767         JMP     B13
65 10163 000402         JMP     B14         ;FINIS
66 10164 125252 CPAT:   125252
```

```
| 0097  .MAIN
01
02 10165 020132 B14:    LDA     0,FLAG
03 10166 101015         MOV#    0,0,SNR ;IF FIRST TIME, (FLAG <> 0 )
04 10167 000407         JMP     B16     ;PERFORM THE ALTERNATE
05 10170 102400         SUB     0,0     ;TEST (B10) AGAIN, WITH THE
06 10171 040132         STA     0,FLAG  ;COMP. PATTERN.
07 10172 024772         LDA     1,CPAT
08 10173 124000         COM     1,1
09 10174 044770         STA     1,CPAT
10 10175 000732         JMP     B10+3
11 10176 024766 B16:    LDA     1,CPAT
12 10177 124000         COM     1,1
13 10200 044764         STA     1,CPAT
14
15 10201 000401 B30:    JMP     .+1     ;WORST CASE TEST. THE BIT POSITIONS
16 10202 102000         ADC     0,0     ;MENTIONED ARE FOR DR103. THE CORRESPOND
17 10203 040132         STA     0,FLAG  ;BIT POS. FOR DR106 ARE 7 AND 11.
18                                      ;LOCATIONS WITH ADDRESS
19                                      ;BIT 3 AND 15 EQUAL ARE
20 10204 020135 B33:    LDA     0,C20   ;SET TO ONES (ZEROS).
21 10205 040131         STA     0,CTR   ;THE REST OF THE LOCATIONS ARE
22 10206 030133 B35:    LDA     2,FIPOS ;FILLED WITH ZEROS (ONES).
23 10207 151112         MOVL#   2,2,SZC
24 10210 076701         DICP    3,1     ;SET MEM EXT FLAG
25 10211 020137 B40:    LDA     0,XPPBA
26 10212 040136         STA     0,XPBAS
27 10213 010136 B45:    ISZ     XPBAS
28 10214 034136         LDA     3,XPBAS
29 10215 025400         LDA     1,0,3   ;GET PATTERN.
30 10216 034134         LDA     3,BMASK
31 10217 157405         AND     2,3,SNR ;MASK OUT BIT 3 AND 15
32 10220 000403         JMP     B50     ;ADR(3)=ADR(15)=0
33 10221 020134         LDA     0,BMASK
34 10222 162435         SUBZ#   3,0,SNR
35 10223 124000 B50:    COM     1,1     ;ADR(3)=ADR(15)=1
36 10224 045000         STA     1,0,2
37 10225 020127         LDA     0,LPOS
38 10226 142435         SUBZ#   2,0,SNR
39 10227 000407         JMP     B60     ;FINIS
40 10230 151400         INC     2,2
41 10231 014131         DSZ     CTR
42 10232 000761         JMP     B45
43 10233 020135         LDA     0,C20
44 10234 040131         STA     0,CTR
45 10235 000754         JMP     B40
46 10236 000401 B60:    JMP     .+1     ;CHECK READ 10 TIMES
47 10237 006111         SETP1
48 10240 020137         LDA     0,XPPBA
49 10241 040136         STA     0,XPBAS ;RESTORE CONT. OF BASE REG.
50 10242 020135         LDA     0,C20
51 10243 040131         STA     0,CTR
52 10244 006124 B65:    DISTB           ;DISTURB THE MEMORY.
53 10245 030133         LDA     2,FIPOS
54 10246 151112         MOVL#   2,2,SZC
55 10247 076701         DICP    3,1     ;SET MEM EXT FLAG
56 10250 010136 B70:    ISZ     XPBAS
57 10251 034136         LDA     3,XPBAS
58 10252 021400         LDA     0,0,3   ;GET PATTERN.
59 10253 034134         LDA     3,BMASK
60 10254 157405         AND     2,3,SNR ;MASK OUT BIT 3 AND 15
61 10255 000403         JMP     B80     ;ADR(3)=ADR(15)=0
62 10256 024134         LDA     1,BMASK
63 10257 166435         SUBZ#   3,1,SNR
64 10260 100000 B80:    COM     0,0     ;ADR(3)=ADR(15)=1
65 10261 025000         LDA     1,0,2
66 10262 106434         SUBZ#   0,1,SZR
67 10263 006114         EHALT           ;MEMOTY FAILED.
68
```

```
I 0098  .MAIN
01
02 10264 100000          COM     0,0
03 10265 041000          STA     0,0,2   ;STORE ORG. PATTERN.
04 10266 025000          LDA     1,0,2   ;CHACK READ.
05 10267 106434          SUBZ#   0,1,SZR
06 10270 006114          EHALT           ;MEMORY FAILED.
07 10271 100000          COM     0,0
08 10272 041000          STA     0,0,2   ;STORE THE ORG. PATT.
09 10273 024127          LDA     1,LPOS
10 10274 146435          SUBZ#   2,1,SNR
11 10275 000411          JMP     B90     ;FINIS
12 10276 151400          INC     2,2
13 10277 014131          DSZ     CTR
14 10300 000750          JMP     B70
15 10301 020135          LDA     0,C20
16 10302 040131          STA     0,CTR
17 10303 020137          LDA     0,XPPBA
18 10304 040136          STA     0,XPBAS
19 10305 000743          JMP     B70
20 10306 006113 B90:     LOOP
21 10307 000401 B91:     JMP     .+1     ;FAST FLOATING ZERO AND ONE TEST.
22 10310 006110          SETP0           ;IN THIS TEST YOU CAN NOT MAKE
23 10311 030133          LDA     2,FIPOS ;ERROR LOOPS.
24 10312 151132          MOVZL#  2,2,SZC
25 10313 076701          DICP    3,1     ;SET MEM. EXT. FLAG
26 10314 025000 B93:     LDA     1,0,2
27 10315 044464          STA     1,XSAVE ;SAVE THE CONT. OF TEST LOC
28 10316 102520          SUBZL   0,0     ;AC0:=1
29 10317 041000 B92:     STA     0,0,2
30 10320 025000          LDA     1,0,2
31 10321 106414          SUB#    0,1,SZR
32 10322 006114          EHALT           ;MEMORY FAILED
33 10323 101124          MOVZL   0,0,SZR
34 10324 000773          JMP     B92
35                                       ;FINIS
36 10325 102120          ADCZL   0,0     ;AC0:=-2
37 10326 041000 B94:     STA     0,0,2
38 10327 025000          LDA     1,0,2   ;CHECK READ
39 10330 106414          SUB#    0,1,SZR
40 10331 006114          EHALT           ;MEMORY FAILED
41 10332 101142          MOVCL   0,0,SZC
42 10333 000773          JMP     B94
43                                       ;FINIS
44 10334 020445          LDA     0,XSAVE
45 10335 041000          STA     0,0,2   ;RESTORE CONT. OF TEST LOC.
46 10336 024127          LDA     1,LPOS
47 10337 132415          SUB#    1,2,SNR
48 10340 000403          JMP     B98     ;LOOP FINIS
49 10341 151400          INC     2,2
50 10342 000752          JMP     B93
51 10343 006113 B98:     LOOP
52
```

```
| 0099  ,MAIN
01
02 10344 010132          ISZ     FLAG
03 10345 000407          JMP     B200
04 10346 020140          LDA     0,XPCBASE
05 10347 024137          LDA     1,XPPBAS
06 10350 040137          STA     0,XPPBAS
07 10351 044140          STA     1,XPCBAS
08 10352 002401          JMP     @XB33   ;PERFORM THE TEST AGAIN
09 10353 010204 XB33:    B33             ;WHIT THE COMP.PATTERN.
10 10354 020137 B200:    LDA     0,XPPBA ;AFTER A LOOP THE PATTERN IS SHIFTED
11 10355 024140          LDA     1,XPCBA ;ONE LEFT.
12 10356 040140          STA     0,XPCBA
13 10357 044137          STA     1,XPPBA
14 10360 030137          LDA     2,XPPBAS
15 10361 034140          LDA     3,XPCBASE
16 10362 126520          SUBZL   1,1     ;
17 10363 151400 B205:    INC     2,2
18 10364 175400          INC     3,3
19 10365 021000          LDA     0,0,2
20 10366 101122          MOVZL   0,0,SZC
21 10367 101400          INC     0,0
22 10370 041000          STA     0,0,2
23 10371 021400          LDA     0,0,3
24 10372 101143          MOVOL   0,0,SNC
25 10373 101120          MOVZL   0,0
26 10374 041400          STA     0,0,3
27 10375 125124          MOVZL   1,1,SZR
28 10376 000765          JMP     B205
29 10377 000403          JMP     B210
30
31 10400 000017 X17:     17
32 10401 000000 XSAVE:   0
33
34 10402 000401 B210:    JMP     .+1     ;ROUTINE TO TEST
35 10403 020506          LDA     0,K10   ;INDRECET ACCES TO MEM <= 32K WORDS, VIA
36 10404 040131          STA     0,CTR   ;AUTI INCREMENT/DECREMENT LOCATIONS
37 10405 020503          LDA     0,CIDD  ;2X/3X. THE CPU'S ABILLITY TO INC/DEC
38 10406 024427          LDA     1,ID1   ;THE LOCATIONS ARE ALSO TESTED.
39 10407 107400          AND     0,1     ;THE TEST IS PERFORMED IN 8 TURNS.
40 10410 044425          STA     1,ID1   ;SET INSTRUKTION TO IDX0(LOC,20)/DDX0(LO
41 10411 024433          LDA     1,ID2
42 10412 107400          AND     0,1
43 10413 044431          STA     1,ID2
44 10414 024434          LDA     1,ID3
45 10415 107400          AND     0,1
46 10416 044432          STA     1,ID3
47 10417 024423          LDA     1,DD1
48 10420 107400          AND     0,1
49 10421 044421          STA     1,DD1
50 10422 024423          LDA     1,DD2
51 10423 107400          AND     0,1
52 10424 044421          STA     1,DD2
53 10425 024426          LDA     1,DD3
54 10426 107400          AND     0,1
55 10427 044424          STA     1,DD3
56 10430 000401 B212:    JMP     .+1
57 10431 030133          LDA     2,FIPOS
58 10432 151132          MOVZL#  2,2,SZC
59 10433 000424          JMP     B220
```

```
| 0100  .MAIN
01
02 10434 020127 B215:   LDA     0,LPOS
03 10435 050020 ID1:    STA     2,IDX0  ;AUTO INC. LOC (2X).
04 10436 151400         INC     2,2
05 10437 112415         SUB#    0,2,SNR
06 10440 000417         JMP     B220    ;FINIS
07 10441 155400         INC     2,3
08 10442 054030 DD1:    STA     3,DDX0  ;AUTO DEC. LOC. (3X)
09 10443 140000         COM     2,0
10 10444 042020 ID2:    STA     0,@IDX0 ;STORE COMP. ADDRESS.
11 10445 026030 DD2:    LDA     1,@DDX0 ;CHECK READ.
12 10446 106414         SUB#    0,1,SZR ;MEMORY FAILED.
13 10447 006114         EHALT           ;MEMORY FAILED.
14 10450 024020 ID3:    LDA     1,IDX0
15 10451 132414         SUB#    1,2,SZR
16 10452 004417         JSR     CPUERR  ;CPU FAILED TO INC. LOC 20.
17 10453 024030 DD3:    LDA     1,DDX0
18 10454 146414         SUB#    2,1,SZR
19 10455 004414         JSR     CPUERR  ;CPU FAILED TO DEC. LOC 30.
20 10456 000756         JMP     B215
21
22 10457 014131 B220:   DSZ     CTR
23 10460 000402         JMP     .+2
24 10461 002130         JMP     @RETES  ;RETURN TO MAIN ROUTINE.
25 10462 010753         ISZ     ID1     ;COUNT AUTO LOC ONE UP.
26 10463 010761         ISZ     ID2
27 10464 010764         ISZ     ID3
28 10465 010755         ISZ     DD1
29 10466 010757         ISZ     DD2
30 10467 010764         ISZ     DD3
31 10470 000740         JMP     B212
32
33 10471 000401 CPUERR: JMP     .+1     ;ROUTINE TO TYPE MESS. AND CALL
34 10472 040466         STA     0,FSAV0 ;MAIN ERROR ROUTINE
35 10473 044466         STA     1,FSAV1
36 10474 050466         STA     2,FSAV2
37 10475 054466         STA     3,FSAV3
38 10476 006040         CMESS
39 10477 010512         CPUME           ;"CPU FAILED TO INC/DEC AUTO-LOC,
40                                      ;SEE PROG. LISTNING."
41 10500 006046         CDICL           ;CLEAR DISPLAY
42 10501 006044         CDISP           ;"SEE PROG.LIST"
43 10502 010550         CPUMD
44 10503 020455         LDA     0,FSAV0
45 10504 024455         LDA     1,FSAV1
46 10505 030455         LDA     2,FSAV2
47 10506 034455         LDA     3,FSAV3
48 10507 002455         JMP     @CFERR  ;RETURN TO MAIN ERROR ROUTINE
49
50
51 10510 177770 CIDD:   177770
52 10511 000010 K10:    10
```

```
! 0101   .MAIN
01                  CPUME:  .TXT    !<12><15>CPU FAILED TO INC/DEC AUTO-LOC,
   10512 006412
   10513 050103
   10514 020125
   10515 040506
   10516 046111
   10517 042105
   10520 052040
   10521 020117
   10522 047111
   10523 027503
   10524 042504
   10525 020103
   10526 052501
   10527 047524
   10530 046055
   10531 041517
02 10532 004454                     <12><15>SEE PROGRAM LISTNING.!
   10533 005011
   10534 051415
   10535 042505
   10536 050040
   10537 047522
   10540 051107
   10541 046501
   10542 046040
   10543 051511
   10544 047124
   10545 047111
   10546 027107
   10547 000000
03
04                  CPUMD:  .TXT    !SEE PROG. LIST.!
   10550 042523
   10551 020105
   10552 051120
   10553 043517
   10554 020056
   10555 044514
   10556 052123
   10557 000056
05
06 10560 000000 FSAV0:  0
07 10561 000000 FSAV1:  0
08 10562 000000 FSAV2:  0
09 10563 000000 FSAV3:  0
10 10564 006617 CFERR:  ERROR
```

```
| 0102  .MAIN
01
02
03 10565 010565 PBASE:  .
04 10566 000001         1B15
05 10567 000002         1B14
06 10570 000004         1B13
07 10571 000010         1B12
08 10572 000020         1B11
09 10573 000040         1B10
10 10574 000100         1B9
11 10575 000200         1B8
12 10576 000400         1B7
13 10577 001000         1B6
14 10600 002000         1B5
15 10601 004000         1B4
16 10602 010000         1B3
17 10603 020000         1B2
18 10604 040000         1B1
19 10605 100000         1B0
20
21 10606 010606 PCBASE: .
22 10607 177776         -1-1B15
23 10610 177775         -1-1B14
24 10611 177773         -1-1B13
25 10612 177767         -1-1B12
26 10613 177757         -1-1B11
27 10614 177737         -1-1B10
28 10615 177677         -1-1B9
29 10616 177577         -1-1B8
30 10617 177377         -1-1B7
31 10620 176777         -1-1B6
32 10621 175777         -1-1B5
33 10622 173777         -1-1B4
34 10623 167777         -1-1B3
35 10624 157777         -1-1B2
36 10625 137777         -1-1B1
37 10626 077777         -1-1B0
```

```
| 0103  .MAIN
01
02
03 10627 165000 XDRIV:  MOV     3,1       ;X-Y DRIVER LOAD TEST, NOT FOR NOVE TYPE
04 10630 006072         CSAMS             ;PRINT START ADDRESS.
05 10631 004410         JSR     TQUES     ;ANSWERE QUESTIONS
06 10632 004427         JSR     UQUES     ;ANSW. QUES.
07 10633 006424         JSR     @CDINI    ;INITIATE
08 10634 004401         JSR     .+1
09 10635 056423         STA     3,@CRENO;NO QUESTIONS, RESTART ADD.
10 10636 006503         JSR     @CDRTE    ;PERFORM THE TEST AS ONE ROUTINE.
11 10637 006503         JSR     @CDLYA    ;CALL DELAY ADMINI.
12 10640 000776         JMP     .-2       ;LOOP
13
14
15 10641 054412 TQUES:  STA     3,TQSRT ;ANSWERE MEMORY TYPE
16 10642 004401         JSR     .+1
17 10643 020405         LDA     0,C4
18 10644 117000         ADD     0,3
19 10645 056404         STA     3,@PON
20 10646 002404         JMP     @PONP
21 10647 002404         JMP     @TQSRT    ;RETURN
22 10650 000004 C4:     4
23 10651 007360 PON:    RETQU
24 10652 007321 PONP:   QUES2
25
26 10653 000000 TQSRT:  0
27 10654 010001 C10001: 10001
28 10655 000420 C420:   420
29 10656 170000 M10K:   -10000
30 10657 011041 CDINI:  DINIT
31 10660 007200 CRENO:  RENOQ
32
33 10661 054772 UQUES:  STA     3,TQSRT ;ANSWERE MEMORY MODUL SIZE, ETC.
34 10662 006071         CQUES
35 10663 010745         TMES2             ;32KW MODUL
36 10664 010763         DMES2             ;32KW MODUL
37 10665 010770         SANS2             ;1
38 10666 006054         CTZOC
39 10667 006060         CDZOC
40 10670 006103         CGTBI             ;READ ANSWERE
41 10671 000402         JMP     .+2       ;SUGGESTED ACCEPTED
42 10672 000770         JMP     UQUES+1 ;ERROR
43 10673 020075         LDA     0,DIGIN ;ANSWERE INPUT'ED
44 10674 024541         LDA     1,C10K  ;16K MODUL
45 10675 101014         MOV#    0,0,SZR
46 10676 024540         LDA     1,C30K  ;32K MODUL
47 10677 034541         LDA     3,ABASE
48 10700 167000         ADD     3,1
49 10701 044537         STA     1,ABASE ;STORE IN DRIVER ADDRESS BASE.
50 10702 101014         MOV#    0,0,SZR
51 10703 000435         JMP     UQUE3   ;32KW
52
53
54 10704 006071 UQUE1:  CQUES
55 10705 010771         TMES3             ;16KW MODUL
56 10706 011007         DMES3             ;16KW MODUL
57 10707 011014         SANS3             ;1
58 10710 006054         CTZOC
59 10711 006060         CDZOC
60 10712 006103         CGTBI             ;READ ANSW.
61 10713 000402         JMP     .+2       ;SUGGESTED ACCEPTED
62 10714 000770         JMP     UQUE1   ;ERROR
63 10715 020075         LDA     0,DIGIN ;ANSWERE INPUT'ED
64 10716 101015         MOV#    0,0,SNR
65 10717 000743         JMP     UQUES+1 ;ONLY 16 OR 32 KWORDS MODULS
```

```
I 0104   .MAIN
01 10720 006071 UQUE2:  CQUES
02 10721 011015         TMES4           ;LOWER MODUL
03 10722 011027         DMES4
04 10723 011034         SANS4           ;1
05 10724 006054         CTZOC
06 10725 006060         CDZOC
07 10726 006103         CGTBI           ;READ ANSWERE
08 10727 000402         JMP     .+2     ;SUGGESTED ACCEPTED
09 10730 000770         JMP     UQUE2   ;ERROR
10 10731 020075         LDA     0,DIGIN ;ANSWERE INDPUT'ED
11 10732 101014         MOV#    0,0,SZR
12 10733 000405         JMP     UQUE3   ;LOWER MODUL
13 10734 024503         LDA     1,C40K  ;16KW MODUL
14 10735 030503         LDA     2,ABASE ;DRIVER ADDRESS BASE
15 10736 133000         ADD     1,2
16 10737 050501         STA     2,ABASE ;STORE IN DRIVER ADDR. BASE
17 10740 002713 UQUE3:  JMP     @TGSRT  ;RETURN
18
19
20 10741 011161 CDRTE:  DRTES
21 10742 011231 CDLYA:  DLYAD
22 10743 011120 CTABB:  TABB
23 10744 011040 CABAS:  ABASE
24
25              TMES2:  .TXT !32KW MEMORY MODUL (709/717)!
   10745 031063
   10746 053513
   10747 046440
   10750 046505
   10751 051117
   10752 020131
   10753 047515
   10754 052504
   10755 020114
   10756 033450
   10757 034460
   10760 033457
   10761 033461
   10762 000051
26              DMES2:  .TXT    !32KW MOD!              ;MEM 709 OR 717
   10763 031063
   10764 053513
   10765 046440
   10766 042117
   10767 000000
27 10770 000001 SANS2:  1
28              TMES3:  .TXT    !16KW MEMORY MODUL (711/718)!
   10771 033061
   10772 053513
   10773 046440
   10774 046505
   10775 051117
   10776 020131
   10777 047515
   11000 052504
   11001 020114
   11002 033450
   11003 030461
   11004 033457
   11005 034061
   11006 000051
29              DMES3:  .TXT    !16KW MOD.!             ;MEM 711 OR 718
   11007 033061
   11010 053513
   11011 046440
   11012 042117
   11013 000056
```

```
| 0105   .MAIN
01 11014 000001 SANS3:  1
02              TMES4:  .TXT    !LOWER MEMORY MODUL!
   11015 047514
   11016 042527
   11017 020122
   11020 042515
   11021 047515
   11022 054522
   11023 046440
   11024 042117
   11025 046125
   11026 000000
03              DMES4:  .TXT    !LOWER MOD!
   11027 047514
   11030 042527
   11031 020122
   11032 047515
   11033 000104
04 11034 000001 SANS4:  1
05 11035 010000 C10K:   10000
06 11036 030000 C30K:   30000
07 11037 040000 C40K:   40000
08 11040 000000 ABASE:  0
09
10 11041 000401 DINIT:  JMP     .+1     ;INITIATE ROUTINE
11 11042 054407         STA     3,DINRT
12 11043 020413         LDA     0,DELST ;
13 11044 040406         STA     0,DELAY ;INIT DELAY, THE DELAY IN 10'S MSEC,
14 11045 020410         LDA     0,CINTPT
15 11046 040001         STA     0,1     ;STORE INTERRUPT ROUTINE ADDR,
16 11047 004410         JSR     DMESS   ;PRINT DELAY
17 11050 002401         JMP     @DINRT  ;RETURN
18
19 11051 000000 DINRT:  0
20 11052 000000 DELAY:  0
21 11053 000000 DELCTR: 0
22 11054 177773 IMASK:  177773
23 11055 011343 CINTPT: INTPT
24 11056 000004 DELST:  4
25 11057 000401 DMESS:  JMP     .+1     ;PRINT DELAY AT TTY/LPT/OCP
26 11060 054416         STA     3,CRET
27 11061 006046         CDICL           ;CLEAR DISPLAY
28 11062 006040         CMESS
29 11063 011077         TMES5           ;<12><15>DELAY IN MSEC:
30 11064 006044         CDISP
31 11065 011110         DMES5           ;DELAY IN MSEC:
32 11066 024764         LDA     1,DELAY
33 11067 131120         MOVZL   1,2
34 11070 125120         MOVZL   1,1
35 11071 127120         ADDZL   1,1
36 11072 147000         ADD     2,1     ;AC1:=10*AC1
37 11073 006053         CTDEC           ;PRINT DELAY, AC1
38 11074 006057         CDDEC           ;DISP AC1
39 11075 002401         JMP     @DRET   ;RETURN
40
41 11076 000000 DRET:   0
42              TMES5:  .TXT    !<12><15>DELAY IN MSEC: !
   11077 006412
   11100 042504
   11101 040514
   11102 020131
   11103 047111
   11104 046440
   11105 042523
   11106 035103
   11107 000040
```

```
| 0106  .MAIN
01                DMESS:  .TXT    !DELAY IN MSEC:!
   11110 042504
   11111 040514
   11112 020131
   11113 047111
   11114 046440
   11115 042523
   11116 035103
   11117 000000
02
03 11120 011121 TABB:   TAB     ;       DRIVER ADDRESSES.
04                              ;
05                              ;                 DR 103
06                              ;NO  XCA/XCC   XC     YCA/YCC      YC
07                              ;                                32K|16K
08 11121 000000 TAB:    0000    ;1       0       8       0       8    0
09 11122 001111         1111    ;2       1       9       1       9    1
10 11123 002222         2222    ;3       2       10      2       10   2
11 11124 003333         3333    ;4       3       11      3       11   3
12 11125 004444         4444    ;5       4       12      4       12   4
13 11126 005555         5555    ;6       5       13      4       13   5
14 11127 006666         6666    ;7       6       14      6       14   6
15 11130 007777         7777    ;8       7       15      7       15   7
16 11131 010000         10000   ;9       8       0       0       0    8
17 11132 011111         11111   ;10      9       1       1       1    9
18 11133 012222         12222   ;11      10      2       2       2    10
19 11134 013333         13333   ;12      11      3       3       3    11
20 11135 014444         14444   ;13      12      4       4       4    12
21 11136 015555         15555   ;14      13      5       5       5    13
22 11137 016666         16666   ;15      14      6       6       6    14
23 11140 017777         17777   ;16      15      7       7       7    15
24
25                              ;
26                              ;                 DR 106
27 11141 000400         0400    ;1
28 11142 001530         1530    ;2
29 11143 002603         2603    ;3
30 11144 003733         3733    ;4
31 11145 014444         14444   ;5
32 11146 015574         15574   ;6
33 11147 016647         16647   ;7
34 11150 017777         17777   ;8      X-Y DRIVERS: SEE ABOVE.
35 11151 020000         20000   ;9
36 11152 021130         21130   ;10
37 11153 022203         22203   ;11
38 11154 023333         23333   ;12
39 11155 034044         34044   ;13
40 11156 035174         35174   ;14
41 11157 036247         36247   ;15
42 11160 037377         37377   ;16
43                      ;XXXXX
44
45
46
47                              ;DRIVER ADDRESSES:
48
49                              ;       DR 103.         DR 106.
50
51                              ;16K WORD MODUL (711)       (718)
52
53                              ;LOWER: 10000 + XXXXX        XXXXX
54                              ;UPPER: 50000 + XXXXX   40000+XXXXX
55
56                              ;32K WORD MODUL (709)       (717)
57
58                              ;LOWER: 30000 + XXXXX   20000+XXXXX
```

```
| 0107  .MAIN
01
02 11161 000401 DRTES:  JMP     .+1     ;PERFORM THE LOAD OF THE DRIVERS.
03 11162 054141         STA     3,RTEST
04 11163 020456         LDA     0,K20
05 11164 040503         STA     0,ADRCTR;ADDRESS COUNTER:=16.
06
07 11165 000401 AGAIN:  JMP     .+1
08 11166 004502         JSR     AMESS   ;PRINT DRIVER ADDRESS.
09 11167 024452         LDA     1,K20
10 11170 020477         LDA     0,ADRCTR;GET ADDR. COUNTER
11 11171 106400         SUB     0,1
12 11172 102400         SUB     0,0
13 11173 042462         STA     0,@CDELC;DELAY COUNTER := 0
14 11174 034724         LDA     3,TABB
15 11175 137000         ADD     1,3
16 11176 021400         LDA     0,0,3
17 11177 030641         LDA     2,ABASE ;AC2:=DRIVER ADDR. BASE
18 11200 113000         ADD     2,2
19 11201 021000         LDA     0,0,2
20 11202 040417         STA     0,CONT  ;SAVE TEST LOCATION.
21 11203 020417         LDA     0,CJSR. ;JSR    .
22 11204 041000         STA     0,0,2   ;STORE JSR  . IN DRIVER ADDR.
23 11205 026452         LDA     1,@CIMAS
24 11206 062677         ICRST           ;RESET I/O
25 11207 102520         SUBZL   0,0     ;THE FREQUENCE IS DETERM. BY:
26 11210 101120         MOVZL   0,0     ;AC0:=2 ;AC=0  50HZ, LINE FREQ.
27 11211 061014         DOA     0,RTC   ;       AC=1  10 "
28                                      ;       AC=2  100"
29                                      ;       AC=3  1000 HZ
30 11212 060114         NIOS    RTC
31 11213 063514         SKPBZ   RTC
32 11214 000777         JMP     .-1     ;SYNC. THE CLOCK, FREQ=100HZ
33 11215 066077         MSKO    1       ;MASK OUT DEVICES EXCL. RTC
34 11216 060114         NIOS    RTC     ;START THE CLOCK, DELAY 10 MSEC.
35 11217 060177         INTEN           ;ENABLE INTERRUPT
36 11220 001000         JMP     0,2     ;JMP TO TEST LOCATION
37
38 11221 000000 CONT:   0
39 11222 004400 CJSR.:  JSR     .
40
41 11223 000401 BACK:   JMP     .+1     ;RETURN FROM THE LOAD TEST
42 11224 020775         LDA     0,CONT
43 11225 041000         STA     0,0,2   ;RESTORE CONT. OF TEST LOC
44 11226 014441         DSZ     ADRCTR  ;DECREMENT ADDRESS COUNTER
45 11227 000736         JMP     AGAIN
46 11230 002141         JMP     @RTEST  ;RETURN
47
48
49 11231 000401 DLYAD:  JMP     .+1     ;DELAY ADMINISTRATION
50 11232 054406         STA     3,DLYRT
51 11233 022423         LDA     0,@CDELA;GET DELAY
52 11234 101120         MOVZL   0,0     ;DELAY:=2*DELAY
53 11235 042421         STA     0,@CDELA;RESTORE DELAY
54 11236 004621         JSR     DMESS   ;PRINT DELAY
55 11237 002401         JMP     @DLYRT  ;RETURN
56
57
```

```
| 0108  .MAIN
01
02 11240 000000 DLYRT:  0
03 11241 000020 K20:    20
04              TMES6:  .TXT    !<12><15>DRIVER ADDRESS NO: !
   11242 006412
   11243 051104
   11244 053111
   11245 051105
   11246 040440
   11247 042104
   11250 042522
   11251 051523
   11252 047040
   11253 035117
   11254 000040
05 11255 011053 CDELC:  DELCTR
06 11256 011052 CDELA:  DELAY
07 11257 011054 CIMAS:  IMASK
08              DMES6:  .TXT    !DRI,ADDR,NO:!
   11260 051104
   11261 027111
   11262 042101
   11263 051104
   11264 047056
   11265 035117
   11266 000000
09 11267 000000 ADRCTR: 0
10
11
12 11270 000401 AMESS:  JMP     .+1         ;ROUTINE TO PRINT DRIVER
13 11271 054441         STA     3,ARET      ;ADDRESS NUMBER,
14 11272 022437         LDA     0,@CADRC    ;GET ADDRESS COUNTER
15 11273 024440         LDA     1,C21
16 11274 106400         SUB     0,1
17 11275 044441         STA     1,ADRSV     ;SAVE ADDRESS NUMBER
18 11276 102000         ADC     0,0         ;AC0:=-1
19 11277 107014         ADD#    0,1,SZR     ;
20 11300 000403         JMP     AMES1       ;ADDR<>1
21 11301 006040         CMESS
22 11302 011242         TMES6               ;DRIVER ADDRESS NO:
23 11303 006046 AMES1:  CDICL               ;CLEAR DISPLAY
24 11304 006044         CDISP
25 11305 011260         DMES6               ;DRI,ADDR,NO:
26 11306 024430         LDA     1,ADRSV     ;GET ADDRESS NUMBER
27 11307 006057         CDDEC               ;DISPLAY AC1
28 11310 020427         LDA     0,CSP       ;AC0:=" "
29 11311 006041         CCHAR               ;PRINT AC0-R
30 11312 020426         LDA     0,CONE      ;AC0:="1"
31 11313 034427         LDA     3,D10       ;AC3:=10.
32 11314 136033         ADCZ#   1,3,SNC
33 11315 006041         CCHAR               ;PRINT "1"
34 11316 024420         LDA     1,ADRSV     ;AC1<AC3
35 11317 034423         LDA     3,D10       ;AC3:=10.
36 11320 136032         ADCZ#   1,3,SZC
37 11321 000402         JMP     .+2
38 11322 166400         SUB     3,1         ;AC1>=AC3
39 11323 020416         LDA     0,C60
40 11324 123000         ADD     1,0
41 11325 006041         CCHAR               ;PRINT AC0-R
42 11326 020407         LDA     0,CKOM      ;
43 11327 006041         CCHAR               ;PRINT ","
44 11330 002402         JMP     @ARET       ;RETURN
45
46
```

```
| 0109  .MAIN
01
02 11331 011267 CADRC:  ADRCTR
03 11332 000000 ARET:   0
04 11333 000021 C21:    21
05 11334 000010 C10:    10       ;8.
06 11335 000054 CKOM:   ",
07 11336 000000 ADRSV:  0
08 11337 000040 CSP:    "
09 11340 000061 CONE:   "1
10 11341 000060 C60:    60
11 11342 000012 D10:    10.      ;
12
13
14 11343 000401 INTPT:  JMP      .+1      ;INTERRUPT ROUTINE FOR X-Y LOAD TEST
15 11344 063777         SKPDZ    CPU
16 11345 000420         JMP      POWIN    ;POWER INTERRUPT
17 11346 024000         LDA      1,0      ;GET INTERRUPTED PROG.ADDR.
18 11347 020076         LDA      0,POWZE  ;GET POWER INSTR.
19 11350 040000         STA      0,0      ;STORE POWER INSTR. IN LOC 0
20 11351 132414         SUB#     1,2,SZR
21 11352 006114         EHALT             ;MEMORY FAILED, THE DREVER ADDRESS
22                                        ;IS CHANGED. THE RIGHT ADDR. IN AC2.
23 11353 012702         ISZ      @CDELC   ;INCREMENT DELAY COUNTER
24 11354 026702         LDA      1,@CDELA;AC1:=DELAY
25 11355 022700         LDA      0,@CDELC;GET DELAY COUNTER
26 11356 106415         SUB#     0,1,SNR
27 11357 000644         JMP      BACK     ;DELAY FINIS
28 11360 022677         LDA      0,@CIMAS;GET MASK
29 11361 062077         MSKO     0        ;MASK OUT DEV. EXCL. RTC
30 11362 060114         NIOS     RTC      ;START THE CLOCK, DELAY 10 MSEC.
31 11363 060177         INTEN             ;ENABLE INTERRUPT.
32 11364 001000         JMP      0,2      ;RETURN TO INTERRUPTED PROG.
33
34 11365 024076 POWIN:  LDA      1,POWZE
35 11366 044000         STA      1,0      ;RESTORE POWER INSTRUKTION
36 11367 063077         HALT
37 11370 000777         JMP      .-1
38
39
```

```
| 0110  .MAIN
01
02 11371 165000 XSTES:  MOV     3,1       ;PAGE ZERO AND DOMAIN OF BIN.LOADR. TEST
03 11372 006072         CSAMS             ;PRINT START ADDRESS
04 11373 006432         JSR     @CINIT    ;INITIATE PASS COUNTER
05 11374 004401         JSR     .+1
06 11375 056461         STA     3,@KRENO;NO QUESTIONS, RESTART ADDRESS.
07 11376 004433         JSR     SINIT     ;INITIATE SUBPASS COUNTER
08 11377 062677 XSTE1:  ICRST             ;RESET I/O
09                                        ;SETUP FOR TEST IN PAGE ZERO
10 11400 102400         SUB     0,0
11 11401 040133         STA     0,FIPOS ;FIRST TEST LOC :=0
12 11402 020425         LDA     0,D255  ;AC0:=377
13 11403 040127         STA     0,LPOS  ;LAST TEST LOC :=255.
14 11404 004466         JSR     STEST   ;PERFORM THE TEST AS A ROUTINE
15
16                                        ;SETUP FOR TEST IN THE DOMAIN
17                                        ;CONTAINING THE BINARY LOADER,
18                                        ;THE LOADER IS NOT DESTROYED.
19 11405 020074         LDA     0,HMEND
20 11406 040133         STA     0,FIPOS ;FIRST TEST LOC:=LAST MEM LOC-200
21 11407 024421         LDA     1,K177
22 11410 123000         ADD     1,0
23 11411 040127         STA     0,LPOS  ;LAST TEST LOC:= LAST MEM LOC (=MEM EXT.
24 11412 004460         JSR     STEST   ;PER FORM THE TEST
25
26                                        ;TEST FOR INTERRUPT FROM
27                                        ;TTY0, TTY1 OR OCP.
28 11413 020413         LDA     0,SMSK
29 11414 062077         MSKO    0         ;MASK OUT
30 11415 060177         INTEN             ;ENABLE INTERRUPT
31 11416 101000         MOV     0,0
32 11417 063477         SKPBN   CPU       ;TEST FOR INTERRUPT
33 11420 002077         JMP     @IRESA  ;INTERRUPT OFF, INTERRUPT HAS OCCURED,
34                                        ;RETURN TO MAIN PROGRAM
35 11421 062677         ICRST             ;INTERRUPT ON, RESET I/O, SET INTER. OFF
36 11422 004417         JSR     SUBPS   ;CALL SUBPASS ADMINISTRA.
37 11423 006123         CPASS             ;CALL PASS ADMINI.
38 11424 000753         JMP     XSTE1   ;PERFORM THE TEST AGAIN
39
40 11425 007217 CINIT:  INIT
41 11426 177375 SMSK:   177375            ;BIT 14 = BIT 7 = 0
42 11427 000377 D255:   255.
43 11430 000177 K177:   177
44
45
46 11431 000401 SINIT:  JMP     .+1       ;ROUTINE TO INITIATE
47 11432 020425         LDA     0,D20
48 11433 040422         STA     0,SUBMX ;SUBMX:=100.
49 11434 102400         SUB     0,0
50 11435 040417         STA     0,SUBCT ;SUBPASS COUNTER:=0
51 11436 020415         LDA     0,CSINPT
52 11437 040001         STA     0,1       ;STORE INTERRUPT ROUTINE ADDR. IN LOC 1
53 11440 001400         JMP     0,3       ;RETURN
54
```

```
| 0111  .MAIN
01
02 11441 000401 SUBPS:  JMP     .+1     ;SUB-PASS ADMINISTRATION
03 11442 054416         STA     3,SUBRT
04 11443 020412         LDA     0,SUBMX
05 11444 010410         ISZ     SUBCT   ;INCREMENT SUBPASS COUNTER
06 11445 024407         LDA     1,SUBCT ;GET SUBPASS COUNTER
07 11446 106414         SUB#    0,1,SZR
08 11447 001401         JMP     1,3     ;RETURN+1, SKIP PASS ADMINI.
09 11450 102400         SUB     0,0
10 11451 040403         STA     0,SUBCT ;SUB-COUNTER:=0
11 11452 001400         JMP     0,3     ;RETURN +0, CALL PASS ADMINI.
12
13 11453 011461 CSINPT: SINTPT
14 11454 000000 SUBCT:  0
15 11455 000000 SUBMX:  0
16 11456 007200 KRENO:  RENCQ
17 11457 000024 D20:    20.
18 11460 000000 SUBRT:  0
19
20 11461 000401 SINTPT: JMP     .+1     ;INTERRUPT ROUTINE ,PAGE ZERO TEST.
21 11462 063777         SKPDZ   CPU
22 11463 002406         JMP     @CPOWI;POWER INTERRUPT
23 11464 034000         LDA     3,0
24 11465 020076         LDA     0,POWZE ;GET POWER INSTR.
25 11466 040000         STA     0,0     ;STORE POWER INSTRUCTION IN LOC 0.
26 11467 062677         IORST           ;RESET I/C
27 11470 001400         JMP     0,3     ;RETURN TO INTERUPTED PROG.
28
29 11471 011365 CPOWI:  POWIN
30
```

```
| 0112  ,MAIN
01
02 11472 000401 STEST:  JMP     .+1         ;ADDRESS AND BIT SELECTION TEST.
03                                          ;FROM FIPOS TO LPOS
04 11473 054464         STA     3,RETST
05
06 11474 000401 E10:    JMP     .+1         ;ADDRESS SELECTION TEST:C(I):=I
07 11475 030133         LDA     2,FIPOS
08 11476 151001         MOV     2,2,SKP
09 11477 151400 E20:    INC     2,2
10 11500 021000         LDA     0,0,2
11 11501 040455         STA     0,CONTX     ;SAVE CONT. OF TEST LOC
12 11502 141000         MOV     2,0
13 11503 006110         SETP0
14 11504 051000         STA     2,0,2       ;STORE ADDRESS
15 11505 025000         LDA     1,0,2       ;CHECK READ
16 11506 132414         SUB#    1,2,SZR
17 11507 004451         JSR     EERRO       ;MEMORY FAILED
18 11510 020446         LDA     0,CONTX
19 11511 041000         STA     0,0,2       ;RESTORE CONT. IN TEST LOC
20 11512 006113         LOOP
21 11513 020127         LDA     0,LPOS
22 11514 112434         SUBZ#   0,2,SZR
23 11515 000762         JMP     E20
24
25 11516 000401 E30:    JMP     .+1         ;BIT SELECTION TEST
26 11517 030133         LDA     2,FIPOS
27 11520 102520 E35:    SUBZL   0,0         ;ALL MEMORY LOC IS FILLED WITH:
28 11521 025000         LDA     1,0,2       ;00.....01
29 11522 044434         STA     1,CONTX     ;00.....10
30 11523 006110 E40:    SETP0
31 11524 041000         STA     0,0,2       ;..........
32 11525 025000         LDA     1,0,2       ;..........
33 11526 106414         SUB#    0,1,SZR     ;01......00
34 11527 004431         JSR     EERRO       ;MEMORY FAILED
35 11530 024426         LDA     1,CONTX
36 11531 045000         STA     1,0,2       ;RESTORE CONT. OF TEST LOC.
37 11532 006113         LOOP
38 11533 100000         COM     0,0         ;10......00
39 11534 006110         SETP0
40 11535 041000         STA     0,0,2       ;00......00
41 11536 025000         LDA     1,0,2
42 11537 106414         SUB#    0,1,SZR
43 11540 004420         JSR     EERRO       ;MEMORY FAILED
44 11541 024415         LDA     1,CONTX
45 11542 045000         STA     1,0,2       ;RESTORE CONT. OF TEST LOC.
46 11543 006113         LOOP
47 11544 100000         COM     0,0
48 11545 101124         MOVZL   0,0,SZR
49 11546 000755         JMP     E40
50 11547 020127         LDA     0,LPOS
51 11550 112415         SUB#    0,2,SNR
52 11551 000403         JMP     E50         ;FINIS
53 11552 151400         INC     2,2
54 11553 000745         JMP     E35
55
56 11554 000401 E50:    JMP     .+1
57
58 11555 002402         JMP     @RETST      ;RETURN TO MAIN PROG.
59
60 11556 000000 CONTX:  0
61 11557 000000 RETST:  0
62
```

```
| 0113  .MAIN
01
02
03 11560 000401 EERRO:  JMP     .+1       ;ERROR ROUTINE
04 11561 040406         STA     0,CONT0   ;SAVE AC0
05 11562 020774         LDA     0,CONTX   ;GET ORG. CONT. OF TEST LOC
06 11563 041000         STA     0,0,2     ;RESTORE CONT.
07 11564 020403         LDA     0,CONT0
08 11565 002401         JMP     @CEERR    ;GOTO MAIN ERROR ROUTINE
09
10 11566 006617 CEERR:  ERROR
11 11567 000000 CONT0:  0
12
13
14 11570 000401 XMOVE:  JMP     .+1       ;MOVE THE PROGRAM TO THE OTHER
15                                        ;MEMORY MODUL:
16                                        ;IF MEM > 32 KWORD THE TO
17                                        ;32 - 36 K ELSE TO 16 - 20 K.
18 11571 165000         MOV     3,1
19 11572 006072         CSAMS             ;PRINT START ADDRESS.
20 11573 076701         DICP    3,1       ;SET MEM EXT FLAG
21 11574 020074         LDA     0,HMEND   ;GET LAST MEM LOC
22 11575 034413         LDA     3,C40,K   ;GET NEW PROG BASE.
23 11576 101132         MOVZL#  0,0,SZC
24 11577 034412         LDA     3,C100K   ;MEM EXTEND.
25 11600 152400         SUB     2,2       ;AC2:=0, MEM < 32K.
26 11601 020411         LDA     0,C14K    ;AC0:=6K
27 11602 040411         STA     0,MCTR    ;INITIALIZE COUNTER.
28
29 11603 021000 MC10:   LDA     0,0,2
30 11604 041400         STA     0,0,3
31 11605 014406         DSZ     MCTR
32 11606 000775         JMP     MC10      ;
33
34 11607 002077         JMP     @IRESA    ;FINIS - RETURN TO MAIN PROG.
35
36 11610 040000 C40,K:  40000
37 11611 100000 C100K:  100000
38 11612 014000 C14K:   14000     ;6K
39 11613 000000 MCTR:   0
40
```

```
| 0114   .MAIN
01
02 11614 000000 LASTP:  0
03       001400          .END    REBIN
  0115   .MAIN

AANUK    005313      68/45    68/47
ABASE    011040     103/47   103/49   104/14   104/16   104/23   105/08   107/17
ACT0     001071      24/03    24/16
ACT6     001072      24/09    24/17
ACTN     001054      23/29    24/02
ADDRS    001702      31/46    32/18    32/28    32/38    32/39    33/02
ADRCT    011267     107/05   107/10   107/44   108/09   109/02
ADRSV    011336     108/17   108/26   108/34   109/07
AGAIN    011165      89/16   107/07   107/45
AGTYP    003013      44/11    44/18    44/37
ALPTT    002255      36/59    38/48
AMEND    001712      33/15    33/23
AMES1    011303     108/20   108/23
AMESS    011270     107/08   108/12
ARESW    001322      28/18    28/24
ARET     011332     108/13   108/44   109/03
B0       010005      89/05    95/09
B1       010034      89/06    95/34
B10      010124      89/09    96/34    97/10
B11      010134      96/42    96/52
B12      010147      96/46    96/53
B13      010151      96/55    97/00
B14      010165      97/01    97/02
B16      010176      97/04    97/11
B2       010036      95/36    95/47
B200     010354      99/03    99/10
B205     010363      99/17    99/28
B210     010402      89/13    99/29    99/34
B212     010430      89/14    99/56   100/31
B215     010434     100/02   100/20
B220     010457      99/59   100/06   100/22
B3       010052      89/07    95/49
B30      010201      89/10    97/15
B33      010204      97/20    99/09
B35      010206      97/22
B3.2     010056      95/53    96/00
B4       010072      89/08    95/62    96/01
B40      010211      97/25    97/45
B45      010213      97/27    97/42
B5       010076      96/11    96/31
B50      010223      97/32    97/35
B6       010077      96/12    96/26
B60      010236      89/11    97/39    97/46
B65      010244      97/52
B70      010250      97/56    98/14    98/19
B8       010123      96/29    96/33
B80      010260      97/61    98/00
B90      010306      98/11    98/20
B91      010307      89/12    98/21
B92      010317      98/29    98/34
B93      010314      98/26    98/50
B94      010326      98/37    98/42
B98      010343      98/48    98/51
BACK     011223     107/41   109/27
BBNUK    005320      68/51
BC20     001660      32/32    32/46
BC377    001610      31/55    31/59
BEND     001703      33/03
BIDIG    006115      76/20    76/43    76/49
```

LSTSW	001366	28/14	28/18	28/21	28/25	28/33	28/44	28/56	29/28
LTMES	007524	91/18	93/08						
M10K	010656	103/29							
MBILO	001501	30/10							
MCMEX	002264	37/16	38/31						
MCMNM	002273	37/22	37/26	37/50	38/24				
MCPUT	005002	44/43	44/45	61/11					
MCRLF	004743	33/55	61/02	69/44	69/46	83/26			
MCTR	011613	113/27	113/31	113/39					
MDCTR	002700	42/31	42/55	43/00					
MDMMC	005113	63/08	64/07	64/08					
MDMMF	005120	63/10	64/18	64/19					
MELOC	001461	13/06	30/04						
MESCH	000434	19/16	19/21						
MESSA	000422	19/10	19/17						
MEXT	007622	94/11	94/21						
MHEAD	006604	83/34	84/18						
MIMEX	002313	37/23	37/43	37/51	38/25	38/32			
MLLOC	001472	29/33	29/35	30/08	37/30	37/32			
MLOOP	002656	42/32	42/36						
MLORE	007077	86/55	86/57	87/46					
MLPTT	002223	36/31	38/46	38/47					
MO10	011603	113/29	113/32						
MODAT	001414	29/20	29/26						
MOFID	002246	36/42	36/51						
MOGTT	002250	36/38	36/53						
MOLAD	002247	36/41	36/52						
MOPTB	002230	36/37	38/60						
MOPTR	002245	36/37	36/46	36/50					
MOREP	002236	36/43	36/49						
MPASS	007050	87/28	87/33						
MPOWO	004745	60/27	61/04						
MSAMS	001521	30/15	33/50	33/52					
MSAQU	004755	60/32	61/08						
MSAV	002701	42/30	42/38	42/54	42/63	43/01			
MST10	002623	41/59	42/11						
MSTIA	002557	41/19	41/23						
MSTIB	002562	41/17	41/22						
MSTIC	002627	42/02	42/04	42/16					
MSTID	002630	41/27	42/00	42/03					
MSTIM	002553	14/24	41/15						
MSTIO	002624	41/21	41/62						
MSTIR	002652	41/15	42/17	42/19	42/20	42/21			
MSWRG	001513	28/32	30/13						
MTEST	007606	90/32	94/08						
MULSA	005564	71/39	71/43	71/46					
MULTE	005556	71/26	71/39						
MULTI	006066	15/40	42/12	83/14					
MX2SP	005207	65/07	65/45	87/20					
MXMMF	005070	62/07	63/02						
MXMMT	005106	62/25	62/26	63/06	64/29	64/30			
MXQUE	005206	65/05	65/41	65/43					
NBEVE	006366	79/41	79/57						
NC125	003635	51/13	51/45						
NC8	003634	50/25	51/44						
NCTYP	003540	50/34	50/37	50/39					
NEXIS	003553	50/27	50/43						
NFTYP	003531	50/21	50/32						
NINHI	000577	21/03	21/06						

NIRET	003541	50/04	50/38	50/40			
NITYP	003532	50/33	51/53				
NN10	000772	23/13	23/48				
NN500	003632	50/17	51/42				
NN9	003633	50/18	51/43				
NOEX	006517	82/16	82/26	82/35	82/45		
NORAT	006574	83/09	83/25				
NOTRE	003575	51/02	51/10	51/11			
NOTYM	003542	50/41	51/05	51/07			
NOTYP	003564	50/33	51/02				
NRESW	001352	28/17	28/42				
NRTYP	003533	50/31	50/34				
NTDEC	003630	51/14	51/34	51/39			
NTREP	003602	51/17	51/35				
NTRES	003631	51/15	51/22	51/29	51/36	51/40	
NTRET	003627	51/16	51/37	51/38			
NTTYP	003576	50/16	51/13				
NUK	000034	15/10	68/45	68/51			
NUMB2	006042	74/08	75/03	75/18	75/23	75/28	75/34
NUMB3	006151	76/08	77/03	77/18	77/21	77/26	77/32
NUMB4	005567	70/09	71/06	71/25	71/28	71/49	72/19
NUMSC	005704	73/08	73/41	73/49	73/55	73/62	
NUTYP	003014	44/10	44/38				
NWAIT	002437	39/21	39/25				
NWTYP	003475	44/36	50/04				
NXDIS	000570	19/50	20/50	20/60	20/63		
NXTYP	003515	50/20	50/24				
NYTYP	003636	51/33	51/47				
NZTYP	003645	51/49	51/52	51/55			
OCTAB	000651	21/18	21/54				
OF2CO	006040	75/10	75/26				
OF3CO	006147	77/10	77/24				
OFTDC	005572	70/17	71/02	72/02			
OFTE2	006012	74/16	75/02	75/30			
OFTE3	006123	76/16	77/02	77/28			
OFTRE	005640	72/30	72/36	72/41			
OFTSC	005706	73/15	73/44	73/58			
OFTSI	005635	72/11	72/14	72/38			
OFTTX	006217	78/15	78/39	79/35			
OKDIG	005774	74/20	74/43	74/57			
OMEND	002376	38/08	38/35				
ONTER	005402	69/39	69/43	71/03	74/52	79/33	
ORDIN	003224	46/23	46/58	46/60	47/10		
OUTS	006363	79/53	79/58				
PASSB	007070	87/13	87/16	87/38	90/29		
PASSC	007071	87/15	87/39	90/28			
PASSN	007072	87/21	87/22	87/40	90/30		
PBASE	010565	16/11	16/12	102/03			
PBINC	001033	23/50	23/55				
PBINN	001037	23/36	23/41	23/54			
PBINR	001074	23/44	23/53	24/20	26/14	26/17	26/33
PCBAS	010606	16/13	102/21				
PCENT	006601	83/21	83/30				
PCH14	004732	60/14	60/45				
PCH35	004733	60/18	60/46				
PCH37	004734	60/22	60/47				
PCCTT	004731	60/09	60/44				
PCCUN	004730	60/10	60/12	60/43			
PDEC1	000613	20/21	21/20				

TCP10	002605	41/44						
TCP11	002606	41/45						
TCP12	002607	41/46						
TCP13	002610	41/47						
TCP14	002611	41/48						
TCP15	002612	41/49						
TCP16	002613	41/50						
TCP17	002614	41/51						
TCP2	002577	41/38						
TCP20	002615	41/52						
TCP21	002616	41/53						
TCP22	002617	41/54						
TCP23	002620	41/55						
TCP24	002621	41/56						
TCP25	002622	41/57						
TCP3	002600	41/39						
TCP4	002601	41/40						
TCP5	002602	41/41						
TCP6	002603	41/42						
TCP7	002604	41/43						
TDMES	007500	91/44	93/04					
TEMP1	001566	31/12	31/18	31/37				
TEMP2	001567	31/22	31/30	31/36	31/38	32/25		
TERBI	006156	77/30	77/32					
TERDC	005520	71/04	71/06					
TERMI	005526	71/09	71/12					
TERMB	006152	76/11	77/28					
TERMD	005514	70/12	71/02					
TERMO	006043	74/11	75/30					
TERMS	005722	73/11	73/58					
TERMT	005407	69/23	69/49	70/11	74/47	79/27		
TERMX	006341	78/11	79/35					
TEROK	006047	75/32	75/34					
TERSC	005726	73/60	73/62					
TERTX	006345	79/37	79/39					
TESTP	010000	94/19	94/24	95/03				
TEXCN	006303	78/52						
TEXIA	006231	78/50	79/16	79/53	80/07			
TEXIN	006232	78/51						
TIMCT	002571	41/30	42/09	42/13				
TIMEM	002572	41/31	42/15	47/04				
TIMEA	0025[illegible]	41/26	41/29	41/63	42/10			
TIM[illegible]	[illegible]	15/30	[illegible]					
T[illegible]	[illegible]	15/32						
T[illegible]	[illegible]	15/37	19/26	19/30	24/45	25/13		
TINIT	[illegible]	23/46	23/61	24/32	25/05	25/28	25/32	26/16	26/37
TMES[illegible]	[illegible]	[illegible]	[illegible]	33/43				
TMES2	010745	103/35	104/25					
TMES3	010771	103/55	104/23					
TMES4	011015	104/02	105/02					
TMES5	011077	105/29	105/42					
TMES6	011242	108/04	108/22					
TOMA	003314	47/37	47/56					
TOMB	003337	47/47	47/57					
TOMC	003335	47/55	48/00					
TOMD	003327	47/45	47/48					
TOMER	003275	44/35	47/22					
TOMF	003331	47/50	47/53					
TOMG	003341	47/59	47/63					

TQMH	003342	47/57	47/60							
TQMJ	003332	47/48	47/51							
TQSRT	010653	103/15	103/21	103/26	103/33	104/17				
TQUES	010641	103/05	103/15							
TRCEN	007143	88/33	88/45	88/57	88/60					
TRCHA	007105	36/22	88/26							
TRCHC	007074	87/42	88/30							
TRCHL	007112	88/31	88/35							
TRCLC	007075	87/43	88/42							
TRCLL	007124	88/43	88/47							
TRCLO	007117	36/24	88/38							
TRCRC	007076	87/44	88/54							
TRCRE	007131	36/26	88/50							
TRCRL	007136	88/55	88/59							
TRCTA	007073	87/41	88/28	88/40	88/52					
TRCTB	007145	87/41	89/03							
TRTYP	002762	44/12	44/19							
TSANS	007361	91/45	92/01							
TTMES	007471	91/43	93/03							
TXCOU	006330	69/61	78/08	78/44	79/05	79/06	79/14	79/24	79/39	
		79/46	80/02							
TXEND	006401	69/62	79/57	80/13						
TXNDR	006412	80/13	80/21	80/22						
TYDM1	007436	91/37	92/08							
TYDM2	007447	91/39	92/09							
TYDM3	007460	91/41	93/02							
TYLIM	003217	45/20	47/05							
TYMA	003065	45/30	45/49							
TYMB	003111	45/41	45/50							
TYMC	003107	45/48	45/57							
TYMD	003101	45/39	45/42							
TYME	003233	47/14	47/17							
TYMEM	003216	45/19	47/04							
TYMEN	003214	45/24	46/48	47/02	47/24					
TYMER	003031	44/12	45/02							
TYMF	003103	45/44	45/47							
TYMG	003113	45/52	45/56							
TYMH	003114	45/50	45/53							
TYMJ	003104	45/42	45/45							
TYPE1	000742	22/52	24/16							
TYPE2	000744	22/54	24/17							
TYPE3	000753	22/57	22/61							
TYPE4	000756	22/60	23/00							
TYPE5	000760	23/02								
TYPIN	005357	69/19	69/22	69/31	69/33	70/23	74/49	79/29		
TYPNX	000474	19/55	19/58	20/45	20/55					
TYPRE	005365	69/22	69/24	69/27	69/28					
TYRTC	003215	47/03	47/23							
TYTME	007365	91/34	92/05							
TYTTF	003221	45/23	47/07							
TYTTS	003220	45/21	47/06							
ULPTT	002256	36/60	38/55							
UMEND	002366	38/18	38/21	38/27						
UNTIM	001527	30/17	44/23	44/25						
UNTST	002733	43/37	44/27							
UQUE1	010704	103/54	103/62							
UQUE2	010720	104/01	104/09							
UQUE3	010740	103/51	104/12	104/17						
UQUES	010661	103/06	103/33	103/42	104/01					

0134 .MAIN

YPDEC	000502	19/49	19/54	20/02	
YPOCT	000603	20/54	20/59	21/12	
YTAB3	000712	19/43	22/26		
YTRMT	006332	78/10	79/27		
YTYPE	000732	22/22	22/34	22/44	
YTYPN	006334	78/20	79/29		
YZOCT	000601	20/44	20/49	21/09	
ZCHAR	000456	19/23	19/41		
ZDOUT	000457	19/42	21/00		
ZSUPP	000771	21/20	21/26	21/36	23/12
ZTAB3	000460	19/43	19/62		