

DataGeneral

**DIAGNOSTIC
LISTING**

LISTING

096-000101-09

PROGRAM

MEMORY CHECKERBOARD IV

TAPE

095-000132-09

ABSTRACT

CHECKERBOARD IV IS A MAINTENANCE PROGRAM DESIGNED TO PRODUCE WORST NOISE CONDITIONS ON THE SENSE/INHIBIT WIRES. THE PROGRAM SHOULD BE RUN TO INSURE PROPER OPERATION OF THE SENSE AMPS, INHIBIT DRIVERS, AND MEMORY CURRENTS.

COPYRIGHT © DATA GENERAL CORPORATION, 1973, 1974, 1975, 1976, 1977, 1978.
ALL RIGHTS RESERVED. PRINTED IN U.S.A.


```

0001 .MAIN      MACRO REV 06.30      09:00:50 08/15/78
01
02
03
04
05
06
07
08
09
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
;*****
;
; NAME: CHECKIV.TX                PART NUMBER: 097-000101
;
; DESCRIPTION: CHECKERBOARD IV
;
; REVISION HISTORY:
;
;     REV.      DATE
;
;     00        05/21/73
;     01        01/25/74
;     02        02/22/74
;     03        06/07/74
;     04        04/18/75
;     05        06/11/76
;     06        12/31/76
;     07        03/18/77
;     08        09/02/77
;     09        08/18/78
;
;
; COPYRIGHT © DATA GENERAL CORPORATION, 1973, 1974, 1975, 1976,
; 1977, 1978.
; ALL RIGHTS RESERVED.
; LICENSED MATERIAL-PROPERTY OF DATA GENERAL CORPORATION.
;*****

```

```

10002 .MAIN
01
02
03
04
05
06
07
08
09
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
;: 0.0 REVISION HISTORY
;:
;: REV. 09 WAS CREATED TO
;: IMPLEMENT THE STANDARDS PROVIDED
;: BY DLIS.
;: THIS HAS NOT CHANGED THE PHILOSOPHY
;: OR TEST PROCEDURES IN THIS PROGRAM.
;: ALL UNNECESSARY "IORST" HAVE BEEN
;: DELETED FROM THIS FILE.
;
; CHECKERBOARD IV
;
;1. ABSTRACT
; CHECKERBOARD IV IS A MAINTENANCE PROGRAM DESIGNED
; TO PRODUCE WORST CASE NOISE CONDITIONS ON THE
; SENSE/INHIBIT WIRES. THE PROGRAM SHOULD BE RUN
; TO INSURE PROPER OPERATION OF SENSE AMPS,INHIBIT
; DRIVERS, AND MEMORY CURRENTS.
;
;2. MACHINE REQUIREMENTS
;2.1 NOVA/SUPERNOVA/ECLIPSE PROCESSOR
;2.2 4K READ/WRITE MEMORY
;
;3. SWITCH SETTINGS
;
; LOCATION "SWREG" IS USED TO SELECT THE PROGRAM OPTIONS
; (NOT SYSTEM CONFIGURATION). WHILE RUNNING UNDER DTOS,
; THIS LOCATION WILL BE LOADED BY THE MONITOR.
; HOWEVER UNDER STAND ALONE AND PROGRAM LOAD MODES THIS
; LOCATION WILL BE SET ACCORDING TO THE ANSWERS SUPPLIED
; BY THE OPERATOR. IN ANY CASE THE OPTIONS CAN BE CHANGED
; OR VERIFIED BY USING ONE OF THE COMMANDS GIVEN IN SEC.
; 3.2
;
;3.1 SWITCH OPTIONS
; DIFFERENT BITS AND THEIR INTERPRETATION AT LOCATION
; "SWREG" IS AS FOLLOWS:
;
; BIT      OCTAL      BINARY      INTERPRETATION
; VALUE    VALUE
;
; 1        40000      1          LOOP ON ERROR
;
; 2        20000      1          PRINT TO CONSOLE
;
; 3        10000      1          DO NOT PRINT % FAILURE
;
; 4        04000      1          ALLOW END OF PASS PRINT OUT
;
; 5        0          0          SUPPRESS END OF PASS PRINT OUT
;
; 5        0          0          DO NOT PRINT ON THE LINE PRINTER

```

0003 .MAIN

```

01      ;
02      ;           02000  1      PRINT ON THE LINE PRINTER
03      ;
04      ;           6           0      DO NOT HALT ON ERROR
05      ;           ;           01000  1      HALT ON ERROR
06      ;
07      ;           7           0      DO NOT PRINT SUMMARY AND/OR
08      ;           ;           PASSING OF EACH SUBTEST
09      ;           ;           00400  1      PRINT SUMMARY AND/OR
10      ;           ;           PASSING OF EACH SUBTEST
11      ;           8           0      PRINT ONLY THE FIRST ERROR
12      ;           ;           00200  1      PRINT EVERY ERROR
13      ;
14      ;3.2 SWITCH COMMANDS
15      ; ONCE THE PROGRAM STARTS EXECUTING THE STATE OF ANY OF
16      ; THE BITS CAN BE CHANGED BY HITTING KEYS 1-9, A-F. THE
17      ; PROGRAM WILL CONTINUE RUNNING AFTER UPDATING THE OPTIONS.
18      ; EACH KEY WILL COMPLEMENT THE STATE OF THE BIT AFFILIAT-
19      ; ED WITH IT, THUS BIT 4 CAN BE ALTERED BY HITTING KEY 4.
20      ; SETTING OF ANY BIT OF LOCATION "SWREG" WILL SET BIT 0.
21      ; (DEFAULT MODE IS DEFINED AS ALL BITS OF SWREG SET TO 0)
22      ; THE PROGRAM CAN BE LOCKED INTO SWITCH MODIFICATION MODE
23      ; BY TYPING A 0, IN WHICH CASE MORE THAN ONE BIT CAN BE
24      ; CHANGED BEFORE CONTROL IS ALLOWED TO RETURN TO THE
25      ; MAIN PROGRAM.
26      ;
27      ;3.2.1 OTHER COMMANDS
28      ;
29      ; "CR"  A "RETURN" CAN BE TYPED TO CONTINUE THE PROGRAM
30      ; AFTER ITS LUCKED IN A SWITCH MODIFICATION MODE
31      ;
32      ; *D    THIS COMMAND GIVEN AT ANY TIME WILL RESET "SWREG"
33      ; TO DEFAULT MODE AND RESTART THE PROGRAM.
34      ;
35      ; *R    THIS COMMAND GIVEN AT ANY TIME WILL RESTART THE
36      ; PROGRAM. SWITCHES ARE LEFT WITH THE VALUES THEY
37      ; HAD BEFORE THE COMMAND WAS ISSUED.
38      ;
39      ; *U    THIS COMMAND GIVEN AT ANY TIME WILL CAUSE THE
40      ; PROGRAM CONTROL TO GO TO ODT (NOTE: THIS IS AN
41      ; OPTIONAL COMMAND AND IS AVAILBLE ONLY IF
42      ; ODTPK IS PRESENT)
43      ;
44      ; M    THIS COMMAND GIVEN AT ANY TIME WILL PRINT THE
45      ; CURRENT OPERATING MODES.
46      ;
47      ;
48      ;3.3 SUPPLEMENTAL SWITCHES
49      ;
50      ; BIT  OCTAL  BINARY  INTERPRETATION
51      ;      VALUE  VALUE
52      ;
53      ; C    00010  0      DO NOT DO 1024 DISTURBANCES
54      ;      ;           1      DO 1024 READ/WRITE DISTURBANCES
55      ;
56      ;4. OPERATING PROCEEDURE
57      ;4.1 LOAD THE PROGRAM VIA THE BINARY LOADER
58      ;4.2 SET SWITCHES TO 000002
59      ;4.3 PRESS START
60      ;4.3.1 THE PROGRAM WILL PRINT THE HIGHEST
        ; LOCATION THE PATTERN IS TO USE.

```

0004 .MAIN

```

01      ;4.4 IF THE FAILURES ARE MARGINAL, SETTING SWITCH
02      ; 0 MAY AID IN INDUCING A FAILURE TO OCCURE.
03      ;4.5 WHEN SCOPING OR ADJUSTING CURRENTS, THE
04      ; BELL WILL STILL BE RUNG.
05      ;4.6 PROGRAM MODIFICATIONS
06      ;4.6.1 C(3)=ADR THE STARTING PATTERN ADDRESS
07      ;4.6.2 C(5)=INHIBIT INHIBIT THE CHECKERBOARD
08      ; PATTERN ON CLEARED BITS.
09      ;
10      ;
11      ;
12      ;

```

!0005 .MAIN

```

01
02
03
04
05
06
07
08
09
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33

```

:5. PROGRAM OUTPUT/ERROR DISCRPTION
:5.1 WHEN AN ERROR OCCURS, THE PROGRAM WILL EXIT TO
: THE OOT ROUTINE (OCTAL DEBUG TOOL). EXAMINATION
: OF CELL 4A WILL PROVIDE THE CONTENTS OF THE PC
: AT THE TIME THE ERROR OCCURRED. CHECK LISTING
: FOR CAUSE OF ERROR.
:5.2 SYNC PULSES
: A "P" PULSE (A74) IN STORE CYCLE.
:
:6. PROGRAM DISCRPTION
:6.1 STORE THE CHECKERBOARD PATTERN
:6.2 IF BIT C (12=1) DISTURB THE CONTENTS OF MEMORY BY
: REFERENCING LOCATIONS 0101,0202,0303,ETC. 512
: TIMES. THIS PRODUCES 1024 READ/WRITE DISTURBS.
:6.3 CHECK THE PATTERN WORD
:6.4 COMPLEMENT AND CHECK THE WORD
:6.5 RESTORE THE WORD
:6.6 WHEN THE END OF THE PATTERN IS REACHED THE
: PROGRAM COMPLEMENTS THE PATTERN WORD AND RE-
: TURNS TO STEP 6.1 .
:7. PATTERN DESCRIPTION
: THE PROGRAM SEQUENTIALLY USES:
: 1) A SLIDING FLOATING ONE BIT
: 2) A SLIDING FLOATING ZERO BIT
:8. LIMITATIONS
: NONE
.EOT

0006 .MAIN

```

01
02
03
04
05
06
07
08
09
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36

```

:*****
: NAME: CHECKIV.SR PART NUMBER: 094-000439
: DESCRIPTION: CHECKERBOARD IV
: REVISION HISTORY:
: REV. DATE
: 00 05/21/73
: 01 01/25/74
: 02 02/22/74
: 03 06/07/74
: 04 04/19/75
: 05 06/11/76
: 06 12/31/76
: 07 03/18/77
: 08 09/02/77
: 09 08/18/78
: COPYRIGHT © DATA GENERAL CORPORATION, 1973,1974,1975,1976,1977,
: 1978.
: ALL RIGHTS RESERVED.
: LICENSED MATERIAL-PROPERTY OF DATA GENERAL CORPORATION.
:*****

```

10007 .MAIN
01      000000      .LOC 0
02 00000 002402      DIRT          ;DTOS DIRT BLK PNTR
03 00001 000000      0
04 00002 000307      J4P MSIZ
05 00003 000000 ADR:  0          ;PATTERN STARTING ADDR(LAST 4 HITS MUST BE 0)
06 00004 000000 FINAL: 0        ;PATTERN FINAL ADDRESS
07 00005 177777 INH:  -1        ;MASK FOR INHIBITED BITS
08 00006 000001 PATT:  1        ;PATTERN WORD
09 00007 000000 ERET:  0
10 00010 000000 CATSW: 0
11
12      000040      .LOC 40
13 00040 000101 C101: 101
14 00041 000400 C400: 400
15 00042 000077 C77:  77
16 00043 007777 C7777: 7777
17 00044 000060 C60:  60
18
19      000045      .LOC 45
20 00045 000060      EGGS          ;DTOS EGGS BLK PNTR
21
22      000050      .LOC 50
23 00050 000010      .BLK 10      ;DEBUGGER BREAK POINT LOCATIONS
24
25 00060 000000 EGGS:  0          ;DTOS AUTO RUN SW
26 00061 000000      0          ;DEVICE CODE
27 00062 000000      0          ;CAT SWITCH
28 00063 000000      0          ;# OF PASSES IN AUTO MODE
29 00064 000000      0          ;DTOS RTN ADDRESS
30 00065 000000 SWREG: 0         ;SWITCH REGISTER
31 00066 000000 PAS2S: 0        ;PASS COUNT
32 00067 000000 AC3:  0          ;=AC3
33 00070 177400 C1774: 177400
34 00071 177773 MS:  -5
35 00072 070000 CG70000: 070000
36 00073 000000 MODUAL: 0
37 00074 000000 EDIST: 0
38 00075 001000 K1000: 1000
39 00076 000507 CMA:  MMA2
40 00077 176140 PSIZE: BEGIN-CEND
41 00100 000000 PLOC:  0
42 00101 000000 RPRG:  0
43 00102 000000 EPRG:  0
44 00103 012345 RANDOM: 12345
45 00104 000020 C20:  20
46 00105 177740 M40:  -40
47 00106 000040 C40:  40
48 00107 010000 C10K: 10000
49 00110 177776 KNT:  -2
50 00111 000000 WHICH: 0
51 00112 000020 CX:  20
52 00113 000400 CY:  400
53 00114 177777 CMASK: -1
54 00115 000000 FLAG:  0
55 00116 000000 TMS:  0
56 00117 000000 MKTK:  0
57 00120 002342 TABL:  TABS
58 00121 002341 TABLO: TABS-1
59 00122 000000 BEGL:  0
60 00123 000560 CBEG:  BEGIN

```

```

0008 .MAIN
01 00124 000000 RETURN: 0
02 00125 000467 ICA?T: MMA

```

10009 .MAIN

```
01
02 00126 000000 QDU?K: 0
03 00127 000017 CONO: 17
04 00130 001400 C1?4K: 1400
05 00131 177700 OFF?M: 177700
06 00132 177770 M10: -10
07 00133 000010 C10: 10
08 00134 000000 MTO?P: 0 ;MENTOP
09 00135 000065 ISWR?EG: 0 ;SWREG
10 00136 000000 IOM?OD: 0 ;PNTR TO IOMOD
11 00137 001354 ITP?E: TYP?E
12 00140 002134 IODT?I: ODT?J
13 00141 001671 IINP?I: INP?J
14 00142 000060 IEGG?S: EGG?S
15 00143 001225 ICRL?F: CRL?F
16 00144 001250 IPDC?S: PDC?S
17 00145 001156 IMES?S: MES?S
18 00146 001236 IZOC?T: ZOC?T
19 00147 001242 IPOC?T: POC?T
20 00150 001351 ITPS?P: TPS?P
21 00151 002420 ICEND: CEND
22 00152 001260 IPDE?C: PDE?C
23 00153 000467 IMMA: MMA
```

```
24
25
26
27 ;THIS ROUTINE ASSURES THAT THE CONTENTS OF ADR HAS LAST FOUR
28 ;BITS EQUAL TO 0.
```

```
29
30 00154 026151 CADR: LDA 1, @ICEND ;COMPUTE CONTENTS OF
31 00155 020132 LDA 0, M10
32 00156 123400 AND 1, 0
33 00157 105000 MOV 0, 1 ;LOCATION ADR:
34 00160 030133 LDA 2, C10 ;C(ADR) = CEND
35 00161 101220 MOVZR 0, 0 ;ROUNDED UP TO THE
36 00162 101220 MOVZR 0, 0 ;NEXT HIGHEST
37 00163 101220 MOVZR 0, 0 ;EVEN NUMBER
38 00164 101223 MOVZR 0, 0, SNC ;IN BITS 10, 11, 12
39 00165 147000 ADD 2, 1 ;THIS INSURES THAT
40 00166 147000 ADD 2, 1 ;BITS 12, 13, 14, 15
41 00167 044003 STA 1, ADR ;ALL =0.
42 00170 001400 JMP 0, 3
43
```

10010 .MAIN

```
01
02 00171 004261 SEND: JSR .RAND ;GET A RNADUM #
03 00172 000103 RANDOM
04 00173 024004 LDA 1, FINAL
05 00174 030077 LDA 2, PSIZE
06 00175 147000 ADD 2, 1
07 00176 101220 MOVZR 0, 0
08 00177 122422 SUBZ 1, 0, SZC
09 00200 000177 JMP .-1
10 00201 123000 ADD 1, 0 ;AC0=# MODULO C(FINAL).
11 00202 024105 LDA 1, M40
12 00203 123400 AND 1, 0
13 00204 024003 LDA 1, ADR
14 00205 122433 SUBZ# 1, 0, SNC ;IF # TO SMALL USE
15 00206 121000 MOV 1, 0 ;C(ADR) FOR STARTER.
16 00207 040100 STA 0, PLOC
17 00210 040122 STA 0, BEGL
18 00211 010122 ISZ BEGL
19 00212 010122 ISZ BEGL
20 00213 034105 LDA 3, M40
21 00214 117000 ADD 0, 3
22 00215 054101 STA 3, BPROG
23 00216 142400 SUB 2, 0
24 00217 024105 LDA 1, M40
25 00220 122400 SUB 1, 0
26 00221 123400 AND 1, 0
27 00222 040102 STA 0, EPROG
28
29 00223 145000 MOVE: MOV 2, 1
30 00224 030123 LDA 2, CBEG
31 00225 021000 LDA 0, 0, 2 ;MOVE A COPY OF
32 00226 041440 STA 0, 40, 3 ;CHECKERBOARD TO
33 00227 175400 INC 3, 3 ;SELECTED SPOT.
34 00230 151400 INC 2, 2
35 00231 125404 INC 1, 1, SZR ;TEST FOR LAST REG
36 00232 000225 JMP MOVE+2 ;TO BE MOVED.
37 00233 034100 LDA 3, PLOC
38 00234 005401 JSR 1, 3
39 00235 010066 ISZ PAS?S ;INC PASS COUNT
40 00236 000237 JMP .+1
41 00237 006145 JSR @IMES?S ;PRINT PASS
42 00240 000546 PASS
43 00241 024066 LDA 1, PAS?S ;PASS COUNT
44 00242 006152 JSR @IPDE?C ;TYPE IT
45 00243 024060 LDA 1, EGG?S
46 00244 125005 MOV 1, 1, SNR
47 00245 000255 JMP .+8.
48 00246 014063 DSZ EGG?S+3
49 00247 000255 JMP .+6
50 00250 062677 IORST
51 00251 034064 LDA 3, EGG?S+4
52 00252 024063 LDA 1, EGG?S+3
53 00253 045776 STA 1, -2, 3
54 00254 001400 JMP 0, 3
55 00255 010010 ISZ CATSW
56 00256 002153 JMP @IMMA
57 00257 006125 JSR @ICA?T
58 00260 002153 JMP @IMMA
```

10011 .MAIN

```
01
02 00261 054303 .RAND: STA 3,.UD03 ;GENERATE A RANDOM #
03 00262 010303 ISZ .UD03
04 00263 031400 LDA 2,0,3
05 00264 021000 LDA 0,0,2
06 00265 024306 LDA 1,.UD21
07 00266 044304 STA 1,.UD10
08 00267 105120 MOVZL 0,1
09 00270 125120 MOVZL 1,1
10 00271 014304 DSZ .UD10
11 00272 000270 JMP #-2
12 00273 107000 ADD 0,1
13 00274 125120 MOVZL 1,1
14 00275 125120 MOVZL 1,1
15 00276 123000 ADD 1,0
16 00277 034305 LDA 3,.UD20
17 00300 163000 ADD 3,0
18 00301 041000 STA 0,0,2
19 00302 002303 JMP @.UD03 ;RETURN
20
21 00303 000000 .UD03: 0
22 00304 000000 .UD10: 0
23 00305 033031 .UD20: 33031
24 00306 000010 .UD21: 10
25
26
```

10012 .MAIN

```
01
02 RES?T: ;RESTART HERE.
03 00307 102440 MSIZ: SUBO 0,0
04 00310 040066 STA 0,PAS?S ;CLEAR PASS COUNT
05 00311 004154 JSR CADR ;COMPUTE ADR
06 00312 020062 LDA 0,EGGS+2
07 00313 040010 STA 0,CATSW
08 00314 020075 LDA 0,K1000 ;SIZE THE MEMORY
09 00315 115000 MOV 0,3
10 00316 031400 MSIZ1: LDA 2,0,3 ;SAVE C(MEM)
11 00317 055400 STA 3,0,3
12 00320 025400 LDA 1,0,3
13 00321 051400 STA 2,0,3 ;RESTORE MEMORY
14
15 00322 125014 MOV# 1,1,SZR
16 00323 124015 COM# 1,1,SNR
17 00324 000335 JMP MSIZ2 ;END OF MEMORY
18 00325 054067 STA 3,AC3 ;SAVE AC3
19 00326 136414 SUB# 1,3,SZR ;AC1=BAD, AC3=GOOD
20 00327 006140 JSR @I0DT? ;MEMORY FAILED.
21 00330 034067 LDA 3,AC3 ;RESTORE AC3
22 00331 117000 ADD 0,3
23 00332 175113 MOVL# 3,3,SNR ;INCREMENT MEMORY ADDRESS.
24 00333 000316 JMP MSIZ1 ;TEST FOR 32K.
25 00334 000335 JMP MSIZ2 ;
26
27 00335 054134 MSIZ2: STA 3,MTOPP ;STORE TOP OF MEMORY
28 00336 171000 MOV 3,2
29 00337 034142 LDA 3,IEGG?S ;GET OFFSET
30 00340 024131 LDA 1,OFF?M
31 00341 021401 LDA 0,1,3
32 00342 123400 AND 1,0
33 00343 101120 MOVZL 0,0 ;=OFFSET
34 00344 126400 SUB 1,1
35 00345 044136 STA 1,IOM?00 ;CLEAR IOM?00
36 00346 025400 LDA 1,0,3 ;CHECK IF IN AUTO MODE
37 00347 125005 MOV 1,1,SNR ;SKIP = AUTO
38 00350 000356 JMP AUT?E
39 00351 024041 LDA 1,C400 ;COMPUTE PNTR TO IO MODULE
40 00352 101004 MOV 0,0,SZR ;IF IN AUTO MODE
41 00353 112401 SUB 0,2,SKP
42 00354 132401 SUB 1,2,SKP
43 00355 050136 STA 2,IOM?00 ;STORE IO MOD PNTR
44 00356 021402 AUT?E: LDA 0,2,3 ;CHECK IF CAT LOADED
45 00357 101005 MOV 0,0,SNR ;SKIP = CAT
46 00360 000466 JMP AUTND
47 00361 025400 LDA 1,0,3 ;CHECK IF IN AUTO MODF
48 00362 125004 MOV 1,1,SZR ;SKIP = NOT AUTO
49 00363 000455 JMP AUT?N
50 00364 024131 LDA 1,OFF?M ;COMPUTE PNTR TO IO MODULE
51 00365 021401 LDA 0,1,3 ;IF CAT IS LOADED AND
52 00366 123400 AND 1,0 ;NOT IN AUTO MODE
53 00367 024041 LDA 1,C400
54 00370 101124 MOVZL 0,0,SZR
55 00371 112401 SUB 0,2,SKP
56 00372 132401 SUB 1,2,SKP
57 00373 050136 STA 2,IOM?00 ;STORE PNTR TO IO MOD
58 00374 000444 JMP AUT?N ;SKIP NEXT BLOCK
59
```



```

10013 .MAIN
01          :THIS BLOCK RESERVED
02          000400      .LOC   400
03 00400 000040      .BLK   40
04
05
06 00440 141000 AUT?N: MOV     2,0          ;COMPUTE PNTR TO CAT START
07 00441 024075      LDA     1,K1000
08 00442 122400      SUB     1,0
09 00443 040125      STA     0,ICA?T      ;STORE PNTR TO CAT START
10 00444 020130      LDA     0,C1?4K
11 00445 112400      SUB     0,2
12 00446 126040 AUTND: ADCCO  1,1          ;COMPUTE TESTABLE TOP OF MEMORY
13 00447 133000      ADD     1,2
14 00450 050004      STA     2,FINAL      ;STORE TESTABLE TOP OF MEM
15 00451 006143      JSR     @ICRL?F      ;PRINT CR,LF
16 00452 006145      JSR     @IMES?S
17 00453 002402      DIRT
18 00454 006145      JSR     @IMES?S      ;PRINT NAME
19 00455 000552      MPRUN
20 00456 006145      JSR     @IMES?S      ;RUNNING
21 00457 000527      MESIZE
22 00460 024004      LOA     1,FINAL
23 00461 006147      JSR     @IPOC?T
24 00462 006143      JSR     @ICRL?F      ;CR,LF
25
26 00463 126040      ADCCO  1,1
27 00464 065111      DOAS   1,TT0          ;SEND RUBOUT TO TTO
28 00465 063411      SKPBN  TTO          ;CHECK FOR TTO
29 00466 000171      JMP     SEND          ;NO TTO, SKIP THIS PART
30 00467 020064 MMA:   LDA     0,EGGS+4      ;IS THIS DTOS LOAD?
31 00470 101005      MOV     0,0,SNR      ;YES,SKIP
32 00471 000404      JMP     .+4          ;NO,GO TO DO THIS PART
33 00472 020010      LDA     0,CATSW      ;IS CAT ALREADY STARTED?
34 00473 100014      COM#   0,0,SZR      ;NO,DO THIS PART
35 00474 000171      JMP     SEND          ;YES,SKIP THIS PART
36 00475 063611      SKPDW  TTO          ;WAIT FOR TTO DONE
37 00476 000777      JMP     .-1          ;TO GET SET
38 00477 020076      LDA     0,CMA
39 00500 040001      STA     0,1
40 00501 152520      SUBZL  2,2          ;SET INTERRUPT RETURN.
41 00502 102000 MMA1: ADC     0,0          ;THIS PROG TEST FOR INTERRUPT
42 00503 040000      STA     0,0          ;ABILITY TO CLEAR MA.
43 00504 025000      LDA     1,0,2
44 00505 060177      NIOS   CPU
45 00506 005000      JSR     0,2          ;ENABLE INTERRUPT
46                                     ;SET BIT INTO MA
47 00507 045000 MMA2: STA     1,0,2
48 00510 020000      LDA     0,0
49 00511 112414      SUB#   0,2,SZR      ;AC0=PC STORED
50 00512 000405      JMP     .+5          ;AC2=CORRECT. MEMORY FAILED!
51 00513 151120      MOVZL  2,2
52 00514 151113      MOVVL# 2,2,SNCR
53 00515 000765      JMP     MMA1
54 00516 000171      JMP     SEND
55 00517 034060      LDA     3,EGGS
56 00520 175234      MOVZR# 3,3,SZR
57 00521 000403      JMP     .+3
58 00522 006140      JSR     @IODT?      ;EXIT TO ODT
59 00523 000770      JMP     MMA2+4
60 00524 062677      IORST

```

```

0014 .MAIN
01 00525 034064      LDA     3,EGGS+4
02 00526 001400      JMP     0,3

```

10015 .MAIN

```
01
02 00527 005215 MESIZE: .TXTE !<15><12>LAST LOCATION TESTED IS = !
03 040714
04 152123
05 146240
06 141717
07 152101
08 147711
09 120116
10 142724
11 152123
12 042305
13 144640
14 120123
15 120275
16 000000
17 00546 005215 PASS: .TXTE !<15><12>PASS !
18 040520
19 051523
20 000240
21 00552 151240 MPRUN: .TXTE ! RUNNING!
22 047125
23 144516
24 043516
25 000000
```

10016 .MAIN

```
01
02 00557 006140 JSR @I0DT? ;OPERATOR ERROR FIX C(ADR)
03 00560 004401 BEGIN: JSR .+1-.1
04 00561 054124 STA 3,RETURN
05 00562 034003 LDA 3,ADR
06 00563 030072 LDA 2,C070000
07 00564 020004 LDA 0,FINAL
08 00565 143400 AND 2,0
09 00566 040074 STA 0,EDIST
10 00567 173400 AND 3,2
11 00570 050073 STA 2,MODUAL ;THE MEMORY MODUAL
12 00571 034120 MKT: LDA 3,TABL
13 00572 024104 LDA 1,C20
14 00573 044117 STA 1,MKTK
15 00574 020006 LDA 0,PATT
16 00575 054067 STA 3,AC3 ;SAVE AC3
17 MKT0: T?TI1
18 00576 040410 STA 0,+.10 ;SAVE AC0
19 00577 034136 LDA 3,IUM?00 ;
20 00600 175005 MOV 3,3,SNR ;SKP IF I/O MOD LOADED
21 00601 000406 JMP .+6 ;
22 00602 006136 JSR @IOM?00 ;CHK IF KEY HIT AND
23 00603 000000 0 ;INPUT TO ACO IF YES
24 00604 000405 JMP .+5 ;TO SWPAK
25 00605 000411 JMP .+11 ;NO KEY
26 00606 000000 0 ;ACO SAVE LOCATION
27 00607 063610 SKPDN TTI ;
28 00610 000406 JMP .+6 ;NO KEY
29 00611 006141 JSR @IINP? ;TO SWPAK
30 00612 020774 LDA 0,.-4 ;RESTORE ACO
31 00613 024127 LDA 1,C0NO ;WAS IT A CONTROL 0?
32 00614 136415 SUB# 1,3,SNR ;NO CONTINUE
33 00615 006140 JSR @I0DT? ;YES GO TO 0DTPK
34
35 00616 034067 LDA 3,AC3 ;RESTORE AC3
36 00617 024005 LDA 1,INH
37 00620 107400 AND 0,1
38 00621 045400 STA 1,0,3
39 00622 100000 COM 0,0
40 00623 024005 LDA 1,INH
41 00624 107400 AND 0,1
42 00625 045420 STA 1,20,3
43 00626 100000 COM 0,0
44 00627 101122 MOVZL 0,0,SZC
45 00630 101400 INC 0,0
46 00631 175400 INC 3,3
47 00632 054067 STA 3,AC3 ;SAVE AC3
48 00633 014117 DSZ MKTK
49 00634 000742 JMP MKT0
50
51 00635 030003 IPAT: LDA 2,ADR
52 00636 102400 SUB 0,0
53 00637 024113 LDA 1,CY
54 00640 147414 AND# 2,1,SZR
55 00641 100000 IPAT0: COM 0,0
56
57 00642 060300 FILL: NIOP 0 ;SYNC AT A74
58 00643 024114 LDA 1,CMASK
59 00644 044116 STA 1,TMES
60 00645 024101 LDA 1,BPROG
```

0017 .MAIN

01	00646	132033	ADCZ# 1,2,SNC
02	00647	000404	JMP FILL1
03	00650	024102	LDA 1,EPROG
04	00651	132433	SUBZ# 1,2,SNC
05	00652	000452	JMP FILL0
06	00653	024104	FILL1: LDA 1,C20
07	00654	034120	FILL2: LDA 3,TABL
08	00655	101014	MOV# 0,0,SZR
09	00656	137000	ADD 1,3

10018 .MAIN

01			
02	00657	025400	LDA 1,0,3
03	00660	045000	STA 1,0,2
04	00661	025401	LDA 1,1,3
05	00662	045001	STA 1,1,2
06	00663	025402	LDA 1,2,3
07	00664	045002	STA 1,2,2
08	00665	025403	LDA 1,3,3
09	00666	045003	STA 1,3,2
10	00667	025404	LDA 1,4,3
11	00670	045004	STA 1,4,2
12	00671	025405	LDA 1,5,3
13	00672	045005	STA 1,5,2
14	00673	025406	LDA 1,6,3
15	00674	045006	STA 1,6,2
16	00675	025407	LDA 1,7,3
17	00676	045007	STA 1,7,2
18	00677	025410	LDA 1,10,3
19	00700	045010	STA 1,10,2
20	00701	025411	LDA 1,11,3
21	00702	045011	STA 1,11,2
22	00703	025412	LDA 1,12,3
23	00704	045012	STA 1,12,2
24	00705	025413	LDA 1,13,3
25	00706	045013	STA 1,13,2
26	00707	025414	LDA 1,14,3
27	00710	045014	STA 1,14,2
28	00711	025415	LDA 1,15,3
29	00712	045015	STA 1,15,2
30	00713	025416	LDA 1,16,3
31	00714	045016	STA 1,16,2
32	00715	025417	LDA 1,17,3
33	00716	045017	STA 1,17,2
34	00717	024104	LDA 1,C20
35	00720	133000	ADD 1,2
36	00721	010116	ISZ TMES
37	00722	000732	JMP FILL2
38	00723	000403	JMP FILL3
39	00724	024104	FILL0: LDA 1,C20
40	00725	133000	ADD 1,2
41	00726	024042	FILL3: LDA 1,C77
42	00727	133414	AND# 1,2,SZR
43	00730	000711	JMP IPAT0-.,1
44	00731	024004	LDA 1,FINAL
45	00732	146432	SUBZ# 2,1,SZC
46	00733	000703	JMP IPAT+1-.,1

10019 .MAIN

```
01
02 00734 030073 DISTUR: LDA 2,MODHAL ;DISTURB MODULE SELECT
03 00735 020043 LDA 0,C7777 ;DISTURB AT LOCATION
04 00736 024074 LDA 1,EDIST ;0101,0202,0303,ETC.
05 00737 123000 ADD 1,0
06 00740 024040 LDA 1,C101 ;EVERY OTHER CORE IN MEMORY
07 00741 133000 ADD 1,2 ;IS DISTURBED AT LEAST
08 00742 036135 LDA 3,@ISWR?EG ;CHECK HIT 1? (KEY C)
09 00743 024133 LDA 1,C10 ;IF SET, 1024 READ/WRITE
10 00744 167414 AND# 3,1,SZR ;DISTURBS. SKIP = NOT SET
11 00745 142433 SUBZ# 2,0,SNC ;IS SET TO A ONE.
12 00746 000406 JMP ICHK ;END OF DISTURB
13 00747 176400 SUB 3,3
14 00750 025000 LDA 1,0,2 ;REFERANCE MEMORY
15 00751 175704 INCS 3,3,SZR
16 00752 000776 JMP .-2
17 00753 060765 JMP DISTURB+4
18
19 00754 030003 ICHK: LDA 2,ADR
20 00755 102400 SUB 0,0
21 00756 024113 LDA 1,CY
22 00757 147414 AND# 2,1,SZR
23 00760 100000 ICHK0: COM 0,0
24 00761 040115 STA 0,FLAG
25
26 00762 024114 ICHK1: LDA 1,CMASK
27 00763 044116 STA 1,TMES
28 00764 024101 LDA 1,RPR0G
29 00765 132033 ADCZ# 1,2,SNC
30 00766 000404 JMP CHEK
31 00767 024102 LDA 1,EPR0G
32 00770 132433 SUBZ# 1,2,SNC
33 00771 000436 JMP CHEK0
```

10020 .MAIN

```
01
02 00772 024104 CHEK: LDA 1,C20
03 00773 044117 STA 1,MKTK
04 00774 034121 LDA 3,TABLU
05 00775 167000 ADD 3,1
06 00776 054020 STA 3,20
07 00777 044021 STA 1,21
08 01000 020115 LDA 0,FLAG
09 01001 101015 MOV# 0,0,SNK
10 01002 000403 JMP .+3
11 01003 044020 STA 1,20
12 01004 054021 STA 3,21
13 01005 022020 CHEK1: LDA 0,@20
14 01006 025000 LDA 1,0,2
15 01007 106434 SUBZ# 0,1,SZK
16 01010 004467 JSR ERR1
17 01011 022021 LDA 0,@21
18 01012 041000 STA 0,0,2
19 01013 025000 LDA 1,0,2
20 01014 122434 SUBZ# 1,0,SZR
21 01015 004463 JSR ERR2
22 01016 014020 DSZ 20
23 01017 036020 LDA 3,@20
24 01020 055000 STA 3,0,2
25 01021 151400 INC 2,2
26 01022 014117 DSZ MKTK
27 01023 000762 JMP CHEK1
28 01024 010116 ISZ THES
29 01025 000745 JMP CHEK
30 01026 000403 JMP CHEK3
31 01027 024104 CHEK0: LDA 1,C20
32 01030 133000 ADD 1,2
33 01031 020115 CHEK3: LDA 0,FLAG
34 01032 024042 LDA 1,C77
35 01033 133414 AND# 1,2,SZR
36 01034 000724 JMP ICHK0
37 01035 024004 LDA 1,FINAL
38 01036 146432 SUBZ# 2,1,SZC
39 01037 000716 JMP ICHK+1
```

10021 .MAIN

```
01
02 01040 024006 UPDT: LDA 1,PATT
03 01041 130000 COM 1,2
04 01042 125122 MOVZL 1,1,SZC
05 01043 125400 INC 1,1
06 01044 020432 LDA 0,UPDTC
07 01045 101233 MOVZR# 0,0,SNC
08 01046 145000 MOV 2,1
09 01047 044006 STA 1,PATT
10 01050 014426 DSZ UPDTC
11 01051 002122 JMP @BEGL
12 01052 024106 LDA 1,C40
13 01053 044423 STA 1,UPDTC
14 01054 020111 LDA 0,WHICH
15 01055 101400 INC 0,0
16 01056 040111 STA 0,WHICH
17 01057 024104 LDA 1,C20
18 01060 030041 LDA 2,C400
19 01061 176000 ADC 3,3
20 01062 401223 MOVZR 0,0,SNC
21 01063 000404 JMP UPDTC
22 01064 024106 LDA 1,C40
23 01065 030107 LDA 2,C10K
24 01066 176120 ADCZL 3,3
25 01067 044112 UPDTC: STA 1,CX
26 01070 050113 STA 2,CY
27 01071 054114 STA 3,CMASK
28 01072 101223 MOVZR 0,0,SNC
29 01073 002122 JMP @BEGL
30 01074 034124 LDA 3,RETURN
31 01075 001400 JMP 0,3
32
33 01076 000040 UPDTC: 40
```

10022 .MAIN

```
01
02 01077 101020 ERR1: MOVZ 0,0 ;DISTURB ENTRY
03 01100 054007 ERR2: STA 3,ERET ;UNDISTURB ENTRY
04 01101 040452 STA 0,.SV0 ;C(1)=ERROR WORD
05 01102 044452 STA 1,.SV1
06 01103 050452 STA 2,.SV2 ;C(2)=ERROR ADDRESS
07 T?TII ;CHECK IF KEY HIT
08 01104 040410 STA 0,.*10 ;SAVE ACO
09 01105 034136 LDA 3,IOM?00 ;
10 01106 175005 MOV 3,3,SNR ;SKP IF I/O MOD LOADED
11 01107 000406 JMP .+6 ;
12 01110 006136 JSR @IOM?00 ;CHK IF KEY HIT AND
13 01111 000000 0 ;INPUT TO ACO IF YES
14 01112 000405 JMP .+5 ;TO S2PAK
15 01113 000411 JMP .+11 ;NO KEY
16 01114 000000 0 ;ACO SAVE LOCATION
17 01115 063610 SKPDN TTI ;
18 01116 000406 JMP .+6 ;NO KEY
19 01117 006141 JSR @IIMP? ;TO S2PAK
20 01120 020774 LDA 0,.-4 ;RESTORE ACO
21 01121 024127 LDA 1,COND ;WAS IT A CONTROL 0?
22 01122 136415 SUB# 1,3,SNR ;NO CONTINUE
23 01123 006140 JSR @IORT? ;YES GO TO 00TPK
24
25 01124 024427 LDA 1,.SV0
26 01125 006147 JSR @IIOC?T
27 01126 006145 JSR @IMES?S
28 01127 002341 PTAB
29 01130 024424 LDA 1,.SV1
30 01131 006147 JSR @IIOC?T
31 01132 006145 JSR @IMES?S
32 01133 002341 PTAB
33 01134 024421 LDA 1,.SV2
34 01135 006147 JSR @IIOC?T
35 01136 006143 JSR @ICRL?F
36 01137 034060 LDA 3,EGGS ;IS DTOS AUTO MODE?
37 01140 175004 MOV 3,3,SZR ;NO,SKIP
38 01141 000407 JMP HACK ;YES,RETURN TO DTOS
39 01142 034100 LDA 3,PLOC
40 01143 054124 STA 3,RETURN
41 01144 020407 LDA 0,.SV0
42 01145 024407 LDA 1,.SV1
43 01146 030407 LDA 2,.SV2
44 01147 002007 JMP @ERET ;TO NOISFY.
45
46 01150 062677 BACK: ICWST ;
47 01151 034064 LDA 3,EGGS+4 ;RETURN TO DTOS
48 01152 001400 JMP 0,3 ;
49 01153 000000 .SV0: 0
50 01154 000000 .SV1: 0
51 01155 000000 .SV2: 0
52
53
54 0?DT0
55 T?TY0
56
57 ;TELETYPE NON INTERRUPT PACKAGE
58 ;CARRY,ACO,AC1,AC2 SAVED
59
60 ;"MES?S" PRINTS ASCII MESSAGES AS SPECIFIED BY ASSEMBLER?
```

0023 .MAIN

```
01
02 ;"CRL?F" PRINTS A CARRIAGE RETURN/LINE FEED
03
04 ;"POC?T" PRINTS C(1) IN OCTAL
05 ;"ZOC?T" PRINTS C(1) IN OCTAL, LEADING ZEROS SUPPRESSED
06 ;"PDC?C" PRINTS C(1) IN DECIMAL, LEADING ZEROS SUPPRESSED,
07 ;THE ABOVE THREE ARE FOLLOWED BY A TAB
08 ;"PDC?S" PRINTS C(1) IN DECIMAL, LEADING ZEROS SUPPRESSED,
09 ;FOLLOWED BY THE CHARACTER STORED AT CALLING LOCATION +1.
10 ;PROGRAM RETURNS TO CALLING LOCATION +2.
11
12 ;"TIN?A" ACCEPTS ASCII CHARACTERS INTO AC1 FROM THE TTY AND
13 ;CLEARS THE TTI.
14 ;"TIN?O" ACCEPTS OCTAL, AND
15 ;"TIN?D" ACCEPTS DECIMAL SINGLE PRECISION SIGNED INTEGERS
16 ;INTO AC1 FROM THE TTY. LEADING NULLS, TABS,
17 ;AND SPACES ARE IGNORED. A 16 BIT UNSIGNED INTEGER IS
18 ;FORMED, THEN NEGATED IF A MINUS SIGN IS TYPED.
19 ;EXIT AT CALL+1 IF INPUT ERROR WITH ACO=BAD CHARACTER.
20 ;(NOT A LEGAL DIGIT OR TERMINATING CHARACTER)
21 ;EXIT AT CALL+2 UPON TERMINATING CHARACTER
22 ;WITH ACO=0, 0, 40, 12, 15, 55
23 ;FOR NULL, SPACE, LINE-FEED, CARRIAGE RETURN, COMMA
24 ;NOTE: A "CR" TERMINATOR RETURNS AN ASCII "LF" (12),
25 ;AND A "LF" TERMINATOR RETURNS AN ASCII "CR" (15).
26
27 ;THE ABOVE WAIT FOR TTY DONE, THEN TTY IS CLEARED.
28 ;RUBOUT WILL DELETE THE LAST DIGIT TYPED IN "TIN?O" AND
29 ;"TIN?D"
30
31 ;"CHC?T" PRINTS ASCII CHARACTER IN C(0)R; C(0)L MUST BE 0.
32 ;EXITS CALL +2 IF C(0)R=0 CALL +1 OTHERWISE; SIMULATES TAB
33
34 ;"TYP?E" PRINTS C(0)R TO THE TTY OR LPT OR BOTH AS PER THE
35 ;SWITCH SELECTION REGISTER 'SAWEG'.
36 ;EXITS AT CALL+1. REPLACE "TYP?E" WITH
37 ;"INTERRUPT 'TYP?E' IF DESIRED.
38
39 ;"TPS?P" PRINTS A SPACE AND EXITS AT CALL+1
40
41 ;NOTE: THE TTYIO PACKAGE DOES NOT TURN OFF THE
42 ;INTERRUPTS OR MASK OUT THE TTY/LPT INTERRUPTS
43 ;WHEN IT PRINTS A CHARACTER. SO IF THE USER IS
44 ;RUNNING WITH INTERRUPTS ON AND WITHOUT TTY/LPT
45 ;MASKED OUT HE WILL RECEIVE INTERRUPTS FROM THESE DEVICES
46 ;IN WHICH CASE HE SHOULD CLEAR THE DEVICE, RE=FNABLE
47 ;THE INTERRUPTS AND RETURN.
48
49 ;NOTE: THE USER MAY VARY THE TAB SIZE AT RUN TIME IN
50 ;HIS PROGRAM BY CHANGING THE VALUE OF LOC "PC?T".
51 ;"PC?T" = NUMBER OF SPACES PER TAB (OCTAL)
52
53 ;
54 ;
55 ;MES?S ROUTINE
56 ;
57 ;THE CALLING SEQUENCE IS:
58 ;
59 ; JSR @MES?S
60 ; POINTER TO MESSAGE TO BE PRINTED
```

0024 .MAIN

```
01 01156 054546 MES?S: STA 3,RTN?A ;SAVE THE RTN ADDRESS
02 01157 004556 JSR SAV?E ;SAVE THE STATE OF MACHINE
03 01160 034544 LDA 3,RTN?A ;
04 01161 010543 ISZ RTN?A ;
05 01162 031400 LDA 2,0,3 ;C(2) POINTS TO MESSAGE
06 01163 024414 LDA 1,P37?T ;AN 8 BIT MASK
07 01164 021000 MES?M: LDA 0,0,2 ;C(2)=DATA WORD
08 01165 125112 MOV# 1,1,SZC ;
09 01166 123701 ANDS 1,0,SKP ;
10 01167 123401 AND 1,0,SKP ;C(0)=DATA CHARACTER (RIGHT BYTE)
11 01170 151400 INC 2,2 ;INC TO NEXT WORD
12 01171 124000 COM 1,1 ;FLIP MASK
13 01172 004411 JSR CHC?T ;PRINT
14 01173 000771 JMP MES?M ;ANOTHER
15 01174 000547 JMP RST?R ;AND EXIT
16
17 ;
18 ;CHC?T ROUTINE
19 ;
20 ;THE CALLING SEQUENCE IS:
21 ;
22 ; LDA 0,CHARACTER TO BE PRINTED (RIGHT BYTE)
23 ; JSR @CHC?T
24 ;
25
26 01175 000000 PSP?: 0
27 01176 000000 SPI?G: 0
28 01177 000377 P37?T: 377
29 01200 000011 PC1?I: 11
30 01201 000000 CHR?E: 0
31 01202 000000 CAC?O: 0
32
33 01203 040777 CHC?T: STA 0,CAC?O ;SAVE ACO
34 01204 101315 MOV# 0,0,SNR ;RETURN +2 IF NULL
35 01205 001401 JMP 1,3 ;
36 01206 175100 MOV# 3,3 ;FOR CARRY SAVE
37 01207 054772 STA 3,CHR?E ;PRINT C(0) RIGHT BYTE
38 01210 034770 LDA 3,PC1?I ;AC3 = 11
39 01211 116415 SUB# 0,3,SNR ;SKIP IF A TAB IS NOT TO
;BE SIMULATED
40
41 01212 000403 JMP CHA?S ;
42 01213 006137 JSR @ITYP?E ;PRINT IT
43 01214 000405 JMP CHE?X ;
44 01215 004534 CHA?S: JSR TPS?P ;PRINT A SPACE
45 01216 020512 LDA 0,CHR?T ;
46 01217 101004 MOV 0,0,SZM ;
47 01220 000775 JMP CHA?S ;SIMULATE A TAB
48 01221 020761 CHE?X: LDA 0,CAC?O ;
49 01222 034757 LDA 3,CHR?E ;
50 01223 175220 MOV# 3,3 ;
51 01224 001400 JMP 0,3 ;
52
53 ;
54 ;
55 ;
56 ;CRL?F ROUTINE
57 ;
58 ;THE CALLING SEQUENCE IS:
59 ;
60 ; JSR @CRL?F
```

```

0025 .MAIN
01
02
03
04 01225 054477 CRL?F: STA 3,RTN?A ;SAVE RETURN
05 01226 004507 JSR SAV?E ;SAVE THE WORLD
06 01227 020406 LDA 0,K15?
07 01230 006137 JSW @ITYP?E ;PRINT CARRIAGE AND LF
08 01231 020403 LDA 0,K12?
09 01232 006137 JSR @ITYP?E
10 01233 000510 JMP RST?R ;EXIT
11 01234 000012 K12?: 12
12 01235 000015 K15?: 15
13
14
15
16 ;
17 ;ZOC?T, POC?T, PDC?S AND PDE?C ROUTINES.
18 ;THE CALLING SEQUENCE IS:
19 ;
20 ; LDA 1,OCTAL NUMBER TO BE PRINTED
21 ; (LEADING ZEROES SUPPRESSED)
22 ; JSR @IZOC?T
23 ;
24 ;THE CALLING SEQUENCE IS:
25 ;
26 ; LDA 1,OCTAL NUMBER TO BE PRINTED
27 ; (LEADING ZEROES NOT SUPPRESSED)
28 ; JSR @IPOC?T
29 ;
30 ;THE CALLING SEQUENCE IS:
31 ;
32 ; LDA 1,DECIMAL NUMBER TO BE PRINTED
33 ; (LEADING ZEROES SUPPRESSED)
34 ; JSR @IPDE?C
35 ;
36 ;THE CALLING SEQUENCE IS:
37 ;
38 ; LDA 1,DECIMAL NUMBER TO BE PRINTED
39 ; (LEADING ZEROES SUPPRESSED)
40 ; JSR @IPDC?S
41 ; ALPHA WHERE ALPHA IS THE CHARACTER PRINTED
42 ; AFTER THE DECIMAL NUMBER
43
44
45 01236 054466 ZOC?T: STA 3,RTN?A ;SAVE THE RTN ADDRESS
46 01237 004476 JSR SAV?E ;SAVE THE WORLD
47 01240 102400 SUB 0,0
48 01241 000404 JMP ZPO?T
49 01242 054462 POC?T: STA 3,RTN?A ;SAVE THE RTN ADDRESS
50 01243 004472 JSR SAV?E ;SAVE THE WORLD
51 01244 020465 LDA 0,PC6?0
52 01245 152620 ZPO?T: SUBZR 2,2 ;PRINT C(1) IN OCTAL
53 01246 034464 LDA 3,PC1?0 ;C(2)=100000, C(3)=10
54 01247 000416 JMP PDC?1
55 01250 175400 PDC?S: INC 3,3 ;UPDATE THE RTN ADDR PNTR
56 01251 054453 STA 3,RTN?A
57 01252 004463 JSR SAV?E ;SAVE THE WORLD
58 01253 034451 LDA 3,RTN?A
59 01254 021777 LDA 0,-1.3 ;READ THE CHARACTER TO BE
60 ;PRINTED AFTER THE DECIMAL

```

```

0026 .MAIN
01
02 01255 040720 STA 0,PSP? ;NUMBER
03 01256 102000 ADC 0,0 ;SAVE THE SPECIAL CHAR.
04 01257 000404 JMP PDC?2 ;AC0= -1
05 01260 054444 PDE?C: STA 3,RTN?A ;
06 01261 004454 JSR SAV?E ;SAVE THE RTN ADDRESS
07 01262 102400 SUB 0,0 ;SAVE THE WORLD
08 01263 034751 PDC?2: LDA 3,K12? ;
09 01264 030447 LDA 2,DET?H ;C(3)=12
10 01265 040711 PDC?1: STA 0,SPT?G ;PRINT C(1) IN DECIMAL
11 ;ACTIVATE/DEACTIVATE THE TAG FOR
12 ;SPECIAL CHARACTER
13 01266 101415 INC# 0,0,SNR ;BOTH ENTRIES PRINT NUMBER
14 01267 101400 INC 0,0 ;SKIP IF AC0 IS NOT -1
15 01270 040562 STA 0,ZSU?P ;THEN TAR TO NEXT POSITION
16 01271 054443 STA 3,TMP? ;SAVE AC3
17 01272 034560 DCU?T: LDA 3,ZSU?P ;ZEROS SUPPRESS STUF
18 01273 102001 DEC?T: ADC 0,0,SKP ;SKIP FIRST TIME HERE PER DIGIT
19 01274 146400 SUB 2,1 ;DIVIDE C(AC1) BY C(AC2)
20 01275 101405 INC 0,0,SNR ;
21 01276 151235 MOVZR# 2,2,SNR ;FOR ZERO SUPPRESS
22 01277 034432 LDA 3,PC6?0
23 01300 146453 SUBO# 2,1,SNR ;SUBTRACT MORE?
24 01301 000773 JMP -.5 ;YES,GO BACK
25 01302 054550 STA 3,ZSU?P ;NO,SAVE ZERO SUPPRESS FLAG
26 ;C(0)=DIGIT
27 01303 163004 ADD 3,0,SZR ;MAKE ASCII
28 01304 004677 JSR CHC?T ;PRINT
29 01305 034427 LDA 3,TMP? ;RESTORE AC3
30 01306 102400 SUB 0,0
31 01307 172423 SUBZ 3,2,SNR ;
32 ;DIVIDE C(AC2) BY C(AC3)
33 01310 000403 JMP .+3 ;SKIP IF AC3 > AC2
34 01311 101400 INC 0,0 ;AC3 < AC2
35 01312 000775 JMP .-3 ;SUBTRACT MORE
36 01313 111004 MOV 0,2,SZR ;WAS IT LAST DIGIT?
37 01314 000756 JMP DCU?T ;NO,GET NEXT DIGIT
38 01315 034661 LDA 3,SPT?G ;YES,CHECK THE SPECIAL CHAR FLAG
39 01316 020662 LDA 0,PC1?1 ;FOLLOW THE PRINTOUT WITH
40 01317 175405 INC 3,3,SNR ;TAB IF NOT SPCL CHAR FLAG
41 01320 020655 LDA 0,PSP? ;OTHERWISE FOLLOW WITH THE CHAR
42 01321 004662 JSR CHC?T
43 01322 034402 LDA 3,RTN?A
44 01323 000420 JMP RST?R ;EXIT
45 01324 000000 RTN?A: 0
46 01325 000000 PCR?Y: 0 ;CRY SAVE LOCATION
47 01326 000000 PAC?0: 0 ;AC0 SAVE LOCATION
48 01327 000000 PAC?2: 0 ;AC2 SAVE LOCATION
49 01330 000000 CHR?Z: 0
50 01331 000060 PC6?0: 60
51 01332 000010 PC1?0: 10
52 01333 023420 DET?B: 10000.
53 01334 000000 TMP??: 0
54
55 ;
56 ; SAV?E , SAVE THE WORLD ROUTINE
57 ;
58 ; THIS ROUTINE SAVES AC0,AC1,AC2, AND CRY
59 ;
60

```

```

0027 .MAIN
01 01335 040771 SAV?E: STA 0,PAC?0 ;
02 01336 044513 STA 1,PAC?1 ;
03 01337 050770 STA 2,PAC?2 ;
04 01340 102560 SUBCL 0,0 ;
05 01341 040764 STA 0,PCR?Y ;
06 01342 001400 JMP 0,3 ;
07
08
09 ; RST?R , RESTORE THE WORLD ROUTINE
10 ;
11 ; THIS ROUTINE RESTORES THE ACO,AC1,AC2, AND CRY
12 ;
13
14 01343 034762 RST?R: LDA 3,PCR?Y ;
15 01344 175200 MOVR 3,3 ;
16 01345 020761 LDA 0,PAC?0 ;
17 01346 024503 LDA 1,PAC?1 ;
18 01347 030760 LDA 2,PAC?2 ;
19 01350 002754 JMP @RTN?A ;
20
21 ;
22 ;TYP?E AND TPS?P ROUTINES.
23 ;
24 ;THE CALLING SEQUENCE IS:
25 ;
26 ; LDA 0,CHARACTER TO BE PRINTED (RIGHT BYTE)
27 ; JSR @ITYP?E
28 ;
29 ;THE CALLING SEQUENCE IS:
30 ;
31 ; JSR @ITPS?P
32 ; NORMAL RETURN WITH ACU = 40
33 ;
34
35
36 01351 040502 TPS?P: STA 0,TAC?0 ;SAVE ACO
37 01352 020505 LDA 0,PC4?0 ;
38 01353 101001 MOV 0,0,SKP ;SKIP OVER ACO SAVE
39 01354 040477 TYP?E: STA 0,TAC?0 ;SAVE ACO
40 01355 044477 STA 1,TAC?1 ;SAVE AC1
41 01356 050477 STA 2,TAC?2 ;SAVE AC2
42 01357 175100 MOVL 3,3 ;SAVE CRY AND RTN ADDR
43 01360 054476 STA 3,TYP?R ;TYPE THE RIGHT BYTE OF ACO
44 01361 034525 LDA 3,P17?7 ;STRIP THE PARITY BIT
45 01362 163400 AND 3,0 ;
46 01363 024573 LDA 1,INT? ;IF IT IS HERE DUE TO SWITCH
47 ;
48 ;SETTING ROUTINE THEN THE TYPE
49 01364 125404 INC 1,1,SZR ;OUTS TO THE TTY WILL BE ENABLED
50 01365 026135 LDA 1,@ISWR?EG ;SKIP IF INT? IS -1
51 01366 127100 ADDL 1,1 ;READ THE SWITCHES
52 01367 125112 MOVL# 1,1,SZC ;SHIFT AC1 BY 2 PLACES
53 ;SKIP IF TYPEOUTS ARE NOT
54 ;SUPPRESSED
55 01370 000417 JMP PLP?T ;
56 01371 030136 TTY?: LDA 2,IOM?D0 ;CHK IF I/O MOD LOED
57 01372 151005 MOV 2,2,SNR ;SKP IF LOADED
58 01373 000404 JMP +4 ;
59 01374 006136 JSR @IOM?D0 ;OUTPUT CHAR IN ACO
60 01376 000411 JMP PLP?T ;

```

```

0028 .MAIN
01 01377 063511 SKPBZ TTO
02 01400 000777 JMP -1
03 01401 061111 DOAS 0,TTO
04 01402 063511 SKPBZ TTO
05 01403 000777 JMP -1
06 01404 063511 SKPBZ TTO
07 01405 000775 JMP -3
08 01406 060211 NIOC TTO ;CLEAR TTO DONE FLAG
09 01407 127100 PLP?T: ADDL 1,1 ;SHIFT AC3 BY 2 PLACES
10 01410 127103 ADDL 1,1,SNC ;SKIP IF THE OUTPUT IS
;REQUIRED ON THE LPT
11
12 01411 000416 JMP TPR?T
13 01412 151005 MOV 2,2,SNR ;CHK IF I/O MOD LOADED
14 01413 000404 JMP +4 ;SKP IF LOADED
15 01414 006136 JSR @IOM?D0 ;OUPUT CHAR TO LPT
16 01415 000003 S ;
17 01416 000411 JMP TPR?T ;
18 01417 063517 SKPBZ LPT
19 01420 000777 JMP -1
20 01421 061117 DOAS 0,LPT ;OUTPUT THE CHARACTER TO LPT
21 01422 063517 SKPBZ LPT ;WAIT FOR LPT
22 01423 000777 JMP -1
23 01424 063517 SKPBZ LPT
24 01425 000775 JMP -3
25 01426 060217 NIOC LPT ;CLEAR THE DONE FLAG FOR LPT
26 01427 034457 TPR?T: LDA 3,P17?7 ;
27 01430 116043 ADCO 0,3,SNR ;SKIP IF IT WAS RUBOUT
28 01431 034426 LDA 3,PC4?0 ;AC3 = 40
29 01432 162432 SUBZ# 3,0,SZC ;SKIP FOR NON PRINTING CHR.
30 01433 010675 ISZ CHR?2
31 01434 034554 LDA 3,PC1?5 ;AC3 = 15
32 01435 116445 SUBO 0,3,SNR ;SKIP IF IT WAS NOT A "CR"
33 01436 054672 STA 3,CHR?Z ;CLEAR THE HORZ POS
34 01437 020671 LDA 0,CHR?Z ;
35 01440 034420 LDA 3,PC?2 ;
36 01441 116405 SUB 0,3,SNR ;
37 01442 054666 STA 3,CHR?Z ;
38 01443 020410 LDA 0,TAC?0 ;RESTORE ACO
39 01444 024410 LDA 1,TAC?1 ;RESTORE AC1
40 01445 030410 LDA 2,TAC?2 ;RESTORE AC2
41 01446 034410 LDA 3,TYP?K ;RESTORE CRY AND RTN ADDR
42 01447 175200 MOVR 3,3 ;
43 01450 001400 JMP 0,3 ;RETURN
44 01451 000000 PAC?1: 0
45 01452 000000 ZSU?P: 0
46 01453 000000 TAC?0: 0
47 01454 000000 TAC?1: 0
48 01455 000000 TAC?2: 0
49 01456 000000 TYP?R: 0
50 01457 000040 PC4?0: 40
51 01460 000010 PC??: 10 ;TAB VALUE
52
53
54 ;TIN?0, TIN?D AND TIN?A ROUTINES.
55 ;
56 ;THE CALLING SEQUENCE IS:
57 ;
58 ;
59 ; JSR @ITI?0 ;ACCEPT IN OCTAL
60 ; ERROR RETURN WITH BAD CHARACTER IN ACO
; NORMAL RETURN WITH TERMINATING

```


0029 .MAIN

```

01      ; CHARACTER IN AC0
02      ; 0,40,12,15,55 FOR
03      ; NULL,SPACE,C/R,L/F AND
04      ; COMMA RESPECTIVELY.
05      ;
06      ; NOTE:
07      ; THE NUMBER IS ACCEPTED IN AC1 FROM TTI BY TIN?0 AND TIN?0
08      ; BAD CHARACTER IS ANY CHARACTER THAT IS NOT A LEGAL
09      ; DIGIT OR A TERMINATING CHARACTER
10      ;
11      ; THE CALLING SEQUENCE IS:
12      ;
13      ; JSR @TITI?0 ;ACCEPT IN DECIMAL
14      ; ERROR RETURN ;SAME FORMAT AS TIN?0
15      ; NORMAL RETURN
16      ;
17      ; THE CALLING SEQUENCE IS:
18      ;
19      ; JSR @TITI?A ;ACCEPT ALL ASCII CHARACTERS
20      ; NORMAL RETURN ;CHAR IN AC0
21      ;
22      ; NOTE: THE NUMBER IS ACCEPTED IN AC0 FROM TTI BY TIN?A
23      ;
24 01461 020526 TIN?C: LDA 0,PC1?2
25 01462 004672 JSR TYP?E
26 01463 010641 TIN?X: LSZ RTN?A
27 01464 040421 TIN?R: STA 0,FST?0 ;"FST?0" IS NON -1
28 01465 152000 ADC 2,2 ;AC2 = -1
29 01466 020764 TSI??: LDA 0,ZSU?P
30 01467 175629 INCR# 3,3 ;AC3 IS 1 IF THE CHARACTER
; TYPED WAS A "+" AND A 100000
; IF IT WAS A "-".
31
32
33 01470 054762 STA 3,ZSU?P
34 01471 101112 MOV# 0,0,SZC ;SKIP IF THE PREVIOUS SIGN
; WAS A PLUS
35 ;TAKE TWO'S COMPLEMENT IF
; THE PREVIOUS SIGN WAS "-"
36 01472 124400 NEG 1,1
37
38 01473 034756 LDA 3,PAC?1
39 01474 167000 ADD 3,1
40 01475 044754 STA 1,PAC?1 ;PAC?1 HAS THE INTERMEDIATE
; RESULT
41
42 01476 126400 SUB 1,1
43 01477 151113 MOV# 2,2,SNC ;SKIP IF EXIT IS REQUIRED
44 01500 000435 JMP TIN?W
45 01501 020404 LDA 0,FST?0 ;LEAVE CHAR IN AC0
46 01502 040624 STA 0,PAC?0
47 01503 034621 LDA 3,RTN?A ;RESTORE THE WORLD AND EXIT
48 01504 000637 JMP RST?R ;RESTORE THE WORLD
49 01505 000000 FST?0: 0
50 01506 000177 P17??: 177
51 01507 054615 TOD?T: STA 3,RTN?A ;SAVE THE RTN ADDR
52 01510 004625 JSR SAV?E ;SAVE THE WORLD
53 01511 102000 ADC 0,0 ;AC1 = -1 (ENTRY FOR ODT)
54 01512 040773 STA 0,FST?0 ;LOOK FOR FIRST DIGIT
55 01513 101120 MOVZL 0,0 ;AC1 = -2
56 01514 000411 JMP TIN?Z
57 01515 054607 TIN?0: STA 3,RTN?A ;OCTAL ENTRY,SAVE RTN ADDR
58 01516 004617 JSR SAV?E ;SAVE THE WORLD
59 01517 102120 ADCZL 0,0 ;OCTAL ENTRY SWITCH
60 01520 000404 JMP TIN?Q ;

```

0030 .MAIN

```

01 01521 054603 TIN?0: STA 3,RTN?A ;DECIMAL ENTRY,RTN SAVED
02 01522 004613 JSR SAV?E ;SAVE THE WORLD
03 01523 102440 SUB# 0,0 ;DECIMAL ENTRY SWITCH
04 01524 126400 TIN?0: SUB 1,1
05 01525 030462 TIN?Z: LDA 2,PC1?2
06 01526 113000 ADD 0,2 ;AC2 IS 10 FOR OCTAL AND 12
; FOR DECIMAL NUMBERS
07
08 01527 102440 SUB# 0,0
09 01530 040722 STA 0,ZSU?P ;SIGN AND LEADING SPACES FLAG
10 01531 034721 TIN?S: LDA 3,ZSU?P
11 01532 175014 MOV# 3,3,SZR ;SKIP FOR LEADING SPACES
12 01533 000730 JMP TIN?X
13 01534 054715 STA 3,PAC?1
14 01535 004460 TIN?W: JSR TIN?A
15 01536 004616 JSR TYP?E
16 01537 034747 LDA 3,P17?2
17 01540 163400 AND 3,0 ;STRIP THE PARITY BIT
18 01541 116415 SUM# 0,3,SNR ;SKIP IF NOT A RRR-ONE
19 01542 000466 JMP RUR?
20 01543 034714 LDA 3,PC4?0
21 01544 116414 SUB# 0,3,SZR
22 01545 101015 MOV# 0,0,SNR
23 01546 000763 JMP TIN?S ;SPACE, OR NULL
24 01547 034443 LDA 3,TIN?2
25 01550 116405 SUB 0,3,SNW ;COMMA
26 01551 000712 JMP TIN?X
27 01552 175414 INC# 3,3,SZR ;MINUS
28 01553 175235 MOVZR# 3,3,SNR ;FOR PLUS ?
29 01554 000712 JMP TSI? ;MODIFY THE SIGN
30 01555 000402 JMP TIN?M ;SKP OVER INT? LOC
31
32 01556 000000 INT?: 0 ;TYPE OUTS CAN BE FORCED TO
;TTY BY PLACING -1 IN THIS LOC
33
34
35 01557 034431 TIN?M: LDA 3,PC1?5
36 01560 116415 SUB# 0,3,SNR ;AC3 = 15
37 01561 000700 JMP TIN?C ;IS IT A CARRIAGE RETURN?
38 01562 044425 LDA 3,PC1?2 ;IF CR THEN GO TO TIN?C
39 01563 116404 SUB 0,3,SZR ;AC3 = 12
40 01564 000404 JMP TIN?N ;SKIP FOR LINE FEED
41 01565 020423 LDA 0,PC1?5
42 01566 006137 JSR @ITYP?E ;AC0 = 15
43 01567 000673 JMP TIN?C+1
44 01570 034421 TIN?N: LDA 3,TIN?1
45 01571 117022 ADDZ 0,3,SZC ;SKIP IF NOT A DIGIT
46 01572 156513 SUB# 2,3,SNC ;SKIP IF DIGIT
47 01573 000671 JMP TIN?R
48 01574 010656 ISZ ZSU?P ;OUT OF LEADING SPACES
49 01575 102400 SUB 0,0 ;AC0 = 0
50 01576 010707 ISZ FST?0 ;SKIP IF IT WAS FIRST DIGIT
; FOR ODT
51
52 01577 121120 MOVZL 1,0
53 01600 105120 MOVZL 0,1
54 01601 125120 MOVZL 1,1 ;AC1 IS SHIFTED BY 3 PLACES
55 01602 167000 ADD 3,1 ;8 OLD PAC?1'S + NEW DIGIT
56 01603 155220 MOVZR 2,3
57 01604 175232 MOVZR# 3,3,SZC ;SKIP IF OCTAL MODE
58 01605 107000 ADD 0,1 ;ADD 2 OLD PAC?1'S
59 01606 000727 JMP TIN?W
60

```

```

0031 .MAIN
01 01607 000012 PC1??: 12
02 01610 000015 PC1??: 15
03 01611 177720 TIN??: -60
04 01612 000054 TIN??: 54
05 01613 000100 C10??: 100
06 01614 000000 PA?C3: 0
07
08 01615 054777 TIN?A: STA 3,PA?C3 ;SAVE AC3
09 01616 020136 LDA 0,IOM?OD ;CHK FOR I/O MOD
10 01617 101005 MOV 0,0,SNR ;SKP IF I/O MOD LOADED
11 01620 000404 JMP ;
12 01621 006136 JSR @IOM?OD ;INPUT CHAR TO AC0
13 01622 000001 1 ;
14 01623 002771 JMP @PA?C3 ;RTN
15 01624 063610 SKPDN TTI ;WAIT FOR KEY
16 01625 000777 JMP .-1 ;
17 01626 060610 DIAC 0,TTI ;
18 01627 002765 JMP @PA?C3 ;RTN
19
20 01630 135005 RUB?: MOV 1,3,SNR ;CAN'T RUB-OUT IF AC1 = 0
21 01631 000633 JMP TIN?R ;RETURN WITH ILLEGAL CHARACTER
22 01632 126400 SUH 1,1 ;
23 01633 156422 SUH2 2,3,SZC ;SKIP IF AC3 IS LESS THAN AC2
24 01634 125401 INC 1,1,SKP ;
25 01635 157001 ADD 2,3,SKP ;
26 01636 000775 JMP .-3 ;
27 01637 054646 STA 3,FST?D ;"FST?D" IS NON -1
28 01640 020404 LDA 0,RB6?D ;AC0 = 60
29 01641 163000 ADU 3,0 ;
30 01642 006137 JSR @ITYP?E ;ECHO AND DELETE THE DIGIT
31 01643 000672 JMP TIN?W ;
32 01644 000060 RB6?D: 60 ;
33 01645 000642 TOD?X: JMP TOD?T ;
34 S?WPX ;
35 S?WPD 1 ;
36
37
38 ;1. SWITCH SETTINGS
39 ;
40 ; LOCATION "SWREG" IS USED TO SELECT THE PROGRAM OPTIONS
41 ; (NOT SYSTEM CONFIGURATION). WHILE RUNNING UNDER DTOS,
42 ; THIS LOCATION WILL BE LOADED BY THE MONITOR.
43 ; HOWEVER UNDER STAND ALONE AND PROGRAM LOAD MODES THIS
44 ; LOCATION WILL BE SET ACCORDING TO THE ANSWERS SUPPLIED
45 ; BY THE OPERATOR. IN ANY CASE THE OPTIONS CAN BE CHANGED
46 ; OR VERIFIED BY USING ONE OF THE COMMANDS GIVEN IN SEC.
47 ; 1.2
48 ;
49 ;
50 ;1.1 SWITCH OPTIONS
51 ; DIFFERENT BITS AND THEIR INTERPRETATION AT LOCATION
52 ; "SWREG" IS AS FOLLOWS:
53 ;
54 ; BIT OCTAL BINARY INTERPRETATION
55 ; VALUE VALUE
56 ;
57 ; 1 40000 0 LOOP ON ERROR
58 ; 2 00000 1 SKIP LOOPING ON ERROR
59 ;
60 ; 2 00000 0 PRINT TO CONSOLE

```

```

0032 .MAIN
01 ; 20000 1 ABORT PRINT OUT TO CONSOLE
02 ;
03 ; 3 0 DO NOT PRINT % FAILURE
04 ; 10000 1 PRINT % FAILURE
05 ;
06 ; 4 0 ALLOW END OF PASS PRINT OUT
07 ; 04000 1 SUPPRESS END OF PASS PRINT OUT
08 ;
09 ; 5 0 DO NOT PRINT ON THE LINE PRINTER
10 ; 02000 1 PRINT ON THE LINE PRINTER
11 ;
12 ; 6 0 DO NOT HALT ON ERROR
13 ; 01000 1 HALT ON ERROR
14 ;
15 ; 7 0 DO NOT PRINT SUMMARY AND/OR
16 ; PASSING OF EACH SUBTEST
17 ; 00400 1 PRINT SUMMARY AND/OR
18 ; PASSING OF EACH SUBTEST
19 ; 0 PRINT ONLY THE FIRST ERROR
20 ; 00200 1 PRINT EVERY ERROR
21
22 ;1.2 SWITCH COMMANDS
23 ; ONCE THE PROGRAM STARTS EXECUTING THE STATE OF ANY OF
24 ; THE BITS CAN BE CHANGED BY HITTING KEYS 1-9, A-F. THE
25 ; PROGRAM WILL CONTINUE RUNNING AFTER UPDATING THE OPTIONS.
26 ; EACH KEY WILL COMPLEMENT THE STATE OF THE BIT AFFILIAT-
27 ; ED WITH IT, THUS BIT 4 CAN BE ALTERED BY HITTING KEY 4.
28 ; SETTING OF ANY BIT OF LOCATION "SWREG" WILL SET BIT 0.
29 ; (DEFAULT MODE IS DEFINED AS ALL BITS OF SWREG SET TO 0)
30 ; THE PROGRAM CAN BE LOCKED INTO SWITCH MODIFICATION MODE
31 ; BY TYPING A 0, IN WHICH CASE MORE THAN ONE BIT CAN BE
32 ; CHANGED BEFORE CONTROL IS ALLOWED TO RETURN TO THE
33 ; MAIN PROGRAM.
34
35 ;1.2.1 OTHER COMMANDS
36 ;
37 ; "CR" A "RETURN" CAN BE TYPED TO CONTINUE THE PROGRAM
38 ; AFTER ITS LOCKED IN A SWITCH MODIFICATION MODE
39 ;
40 ; *D THIS COMMAND GIVEN AT ANY TIME WILL RESET "SWREG"
41 ; TO DEFAULT MODE AND RESTART THE PROGRAM.
42 ;
43 ; *R THIS COMMAND GIVEN AT ANY TIME WILL RESTART THE
44 ; PROGRAM. SWITCHES ARE LEFT WITH THE VALUES THEY
45 ; HAD BEFORE THE COMMAND WAS ISSUED.
46 ;
47 ; *O THIS COMMAND GIVEN AT ANY TIME WILL CAUSE THE
48 ; PROGRAM CONTROL TO GO TO 00T (NOTE: THIS IS AN
49 ; OPTIONAL COMMAND AND IS AVAILBLE ONLY IF
50 ; OUTPK IS PRESENT)
51 ;
52 ; *M THIS COMMAND GIVEN AT ANY TIME WILL PRINT THE
53 ; CURRENT OPERATING MODES.
54 ;
55 ;
56 ; THIS PACKAGE IS USED TO CHANGE THE SETTINGS OF LOCATION
57 ; "SWREG" OF PAGE 0. THE PROGRAM CONTROL SHOULD ENTER
58 ; INP?J OR INP?K WITH AC3 HAVING THE RTN ADDR.
59 ; THE PROGRAM SHOULD ENTER AT INP?J IF OUT IS NOT PRESENT
60

```

0033 .MAIN

```

01 ;OR AT INP?K IF ODT IS PRESENT.
02 ;THE INPUT IS ECHOED AFTER A "CR". IF THE COMMAND IS
03 ;NOT A LEGAL ONE THEN THE CONTROL IS RETURNED WITHOUT DOING
04 ;ANYTHING, OTHERWISE ONE OF THE FOLLOWING COMMANDS IS
05 ;EXECUTED:
06 ;KEYS 1-9 AND A-F ARE USED TO COMPLEMENT THE CURRENT VALUE
07 ;OF BITS 1-15 OF "SWREG". IF ONE OF THESE KEYS IS HIT THE
08 ;CORRESPONDING BIT OF "SWREG" IS COMPLEMENTED AND THE CONTROL
09 ;IS RETURNED TO THE STATE PROGRAM HAD BEFORE HITTING THE KEY
10 ;TYPING OF A "0" WILL LOCK THE PROGRAM IN A SWITCH MODIFICATION
11 ;MODE IN WHICH CASE MORE THAN ONE BIT CAN BE CHANGED BEFORE
12 ;THE CONTROL IS ALLOWED TO RETURN TO THE MAIN PROGRAM. HITTING
13 ;THE "CR" KEY WILL UNLOCK THE PROGRAM FROM THIS MODE.
14 ;"D" THIS COMMAND GIVEN AT ANY TIME WILL RESET THE "SWREG"
15 ;TO DEFAULT MODE (ALL ZEROS) AND RESTART THE PROGRAM AT ADD.
16 ;STORED IN LOCATION "INS?"
17 ;"R" THIS COMMAND GIVEN AT ANY TIME WILL RESTART THE PROG.
18 ;AT ADDRESS STORED IN LOCATION "INS?"
19 ;"M" THIS COMMAND GIVEN AT ANY TIME WILL PRINT THE CURRENT
20 ;OPERATING MODES.
21 ;BEFORE THE CONTROL IS RETURNED TO THE MAIN PROGRAM BIT 0 WILL
22 ;BE SET IF ANY OF THE OTHER BITS OF "SWREG" ARE SET, OTHERWISE
23 ;IT WILL BE CLEARED
24 ;THE LABEL RES?T MUST BE A PROGRAM RESTART LOCATION FOR
25 ;THE CONTROL R AND CONTROL D INPUT KEY FUNCTIONS
26 ;
27 ;   EXAMPLE:
28 ;           RES?T:
29 ;           BEGIN: XXX
30 ;
31 ;WHERE RES?T AND BEGIN ARE AT THE FIRST INSTRUCTION OF THE PROGRAM.
32 ;
33 ;NOTE: THIS PACKAGE REQUIRES THE TTYIO PACKAGE IN
34 ;
35 ;   ORDER TO RUN
36 ;
37 ;   THE LOCATION "SWREG" MUST BE POINTED TO BY THE PAGE
38 ;
39 ;   ZERO POINTER ISWR?EG AND SHOULD BE THE SIXTH WORD OF
40 ;
41 ;   THE EGGS BLOCK.
42 ;
43 ;   A USER WHO WANTS TO DEFINE HIS OWN TTI COMMANDS SHOULD
44 ;
45 ;   BUILD A ROUTINE WITH THE ENTRY INP?X WHICH CALLS INP?K
46 ;
47 ;   OR INP?J AS DESIRED AND THEN CHECKS FOR HIS INPUT KEY
48 ;
49 ;   WHICH WILL BE PLACED IN AC3 ON RETURN FROM THE SWPAK ROUTINE
50 ;
51 ;THE CALLING SEQUENCE IS:
52 ;
53 ;           JSR   @IINP? ,WHERE IINP? POINTS TO
54 ;
55 ;           INP?J OR INP?K
56 ;
57 ;NOTE: THE INP?K ROUTINE IS IN THE O?DTP MACRO.
58
59 01646 176000 IN3?:  ADC   3,3           ;AC3 = -1
60 01647 054413   STA   3,INL?K        ;LOCK IN SWITCH INPUT MODE
61 01650 026135   LDA   1,@ISWR?EG      ;READ THE CURRENT VALUE OF
62                               ;"SWREG"
63 01651 133414   AND#  1,2,SZR         ;TAKE XOR OF AC1 AND AC2
64 01652 146401   SUB   2,1,SKP
65 01653 147000   ADD   2,1
66 01654 046135   STA   1,@ISWR?EG      ;SAVE THE NEW VALUE OF "SWREG"
67 01655 000554   JMP   IN4?
68 01656 000033 IN3?3:  33
69 01657 000136 IN1?36: 136
70 01660 000104 IN1?04: 104
71 01661 000000 INT?E:  0
72 01662 000000 INL?K:  0

```

0034 .MAIN

```

01 01663 152000 INP?1:  ADC   2,2           ;FORCE THE TYPE OUTS ON
02 01664 050672   STA   2,INT?         ;THE TELETYPE
03 01665 002143   JMP   @ICRL?F
04
05 01666 054576 INP?1:  STA   3,INP?R
06 01667 176000   ADC   3,3
07 01670 177061   ADCC  3,3,SKP
08 01671 054573 INP?J:  STA   3,INP?R        ;SAVE THE RETURN ADDRESS
09 01672 054767   STA   3,INT?E
10 01673 004553   JSR   INS?V        ;SAVE THE STATUS
11 01674 040766   STA   0,INL?K      ;"INL?K" IS NOT -1
12 01675 020572   LDA   0,INS?0      ;GET CHAR IF USING I/O MOD
13 01676 034136 IN0?:  LDA   3,IOM?00      ;CHK IF I/O MOD LOADED
14 01677 175005   MOV   3,3,SNR      ;SKP IF NO I/O MOD
15 01700 060610   DIAC  0,TTI        ;INPUT CHAR
16 01701 034605   LDA   3,P1?77      ;AC3 = 177
17 01702 163400   AND   3,0           ;GET RID OF THE PARITY BIT
18 01703 040567   STA   0,INS?3      ;SAVE CHARACTER
19 01704 034704   LDA   3,PC1?5      ;AC1 = 15
20 01705 116415   SUB#  0,3,SNR      ;SKIP IF THE CHARACTER TYPED
21                               ;WAS NOT "CR"
22 01706 000425   JMP   INR?
23 01707 024704   LDA   1,C10?0
24 01710 034677   LDA   3,PC1?2      ;AC1 = 100
25 01711 116414   SUB#  0,3,SZR
26 01712 034744   LDA   3,IN3?3      ;SKIP IF IT IS A LINE FEED
27 01713 162453   SUB#  0,3,SNC      ;DON'T SKIP IF AC0 IS 33
28                               ;FOR MORE
29 01714 000447   JMP   IN1?
30 01715 107000 IN5?:  ADD   0,1           ;AC1 = 100+ ASCII VALUE OF
31                               ;CONTROL CHARACTER
32                               ;AC0 = 136
32 01716 020741   LDA   0,IN1?36
33 01717 006137   JSR   @ITYP?E      ;TYPE "
34 01720 121000   MOV   1,0
35 01721 006137   JSR   @ITYP?E
36 01722 034555   LDA   3,I12?1      ;AC3 = 121
37 01723 162015   ADC#  3,0,SNR      ;SKIP IF IT IS NOT "R"
38 01724 000405   JMP   IN6?
39 01725 034733   LDA   3,IN1?04      ;AC3 = 104
40 01726 116404   SUB   0,3,SZR      ;SKIP IF IT WAS A "D"
41 01727 000502   JMP   IN4?
42 01730 056135   STA   3,@ISWR?EG   ;LOAD "SWREG" WITH 0
43 01731 034545 IN6?:  LDA   3,INS?        ;AC3 = ADDRESS OF THE LOCATION
44                               ;WHERE THE PROGRAM WILL START
45 01732 054532   STA   3,INP?R
46 01733 176400 INR?:  SUB   3,3
47 01734 165000   MOV   3,1
48 01735 006143   JSR   @ICRL?F
49 01736 054620   STA   3,INT?
50 01737 032135   LDA   2,@ISWR?EG
51 01740 176220   ADCZR 3,3
52 01741 173404   AND   3,2,SZR      ;AC3 = 77777
53                               ;SKIP IF THE SWITCHES ARE SET
54 01742 172000   ADC   3,2          ;TO ALL ZERO'S
55 01743 052135   STA   2,@ISWR?EG
56 01744 034527   LDA   3,STA?1
57 01745 175223   MOVZR 3,3,SNC
58
59 01746 060211   NIOC  TTO
60 01747 175220   MOVZR 3,3

```

```

0035 .MAIN
01 01750 175204      MOVR    3,3,SZR      ;LOAD THE CARRY BIT AND SKIP
02                                     ;IF THE INTERRUPTS ARE NOT TO
03                                     ;BE ENARLED
04 01751 125004      MOV     1,1,SZR      ;SKIP IF THE INSTRUCTION BEING
05                                     ;EXECUTED IS A "P"
06 01752 176000      AOC     3,3         ;INTERRUPTS ARE TO BE LEFT DIS-
07                                     ;ABLED
08 01753 054520      STA     3,STA?T     ;"STA?T" IS 0 IF INTERRUPTS ARE
09                                     ;TO BE ENARLED AND -1 OTHERWISE
10 01754 020513      LDA     0,INS?0     ;RESTORE THE ACCUMULATORS
11 01755 024513      LDA     1,INS?1
12 01756 030513      LDA     2,INS?2
13 01757 034513      LDA     3,INS?3
14 01760 010513      ISZ    STA?T
15 01761 060177      INTEN
16 01762 002502      JMP     @INP?R      ;START EXECUTING THE USEW'S
17                                     ;PROGRAM
18
19 01763 006137 IN1?: JSR     @ITYP?E     ;ECHO THE CHARACTER
20 01764 034514      LDA     3,IN6?0     ;AC3 = 60
21 01765 152620      SUBZR  2,2         ;AC2 = 100000
22 01766 116405      SUB    0,3,SNR      ;SKIP IF THE DIGIT TYPED WAS
23                                     ;NOT 0
24 01767 000657      JMP     IN3?
25
26 01770 151221 IN2?: MOVZR  2,2,SKP     ;SHIFT AC2 TO RIGHT
27 01771 126520      SUBZL  1,1         ;AC1 = 1
28 01772 175405      INC    3,3,SNR
29 01773 000655      JMP    IN3?+2
30 01774 147415      AND#   2,1,SNR
31                                     ;STAY IN LOOP UNTIL ALL BITS
32                                     ;OF SWREG ARE CHECKED
33 01775 000773      JMP    IN2?
34 01776 106400      SUB    0,1         ;WHEN THE CONTROL COMES HERE
35                                     ;FOR THE FIRST TIME AC1 = 100
36
37 01777 135000      MOV    1,3
38 02000 151225      MOVZR  2,2,SNR
39 02001 000430      JMP    IN4?
40 02002 024500      LDA    1,IN1?5
41 02003 167004      ADD    3,1,SZR
42 02004 000765      JMP    IN2?+1
43
44 02005 006143 INM?: JSR     @ICRL?F     ;TYPE A "CR" AND "LF"
45 02006 152520      SUBZL  2,2         ;SET AC?# 1
46 02007 006144      JSR    @IPDC?S     ;PRINT THE CONTENTS OF AC1
47 02010 000040      40
48 02011 125400      INC    1,1
49 02012 034467      LDA    3,IN1?2
50 02013 166452      SUBO#  3,1,SZC
51
52 02014 006150      JSR    @IIPS?P     ;PRINT A SPACE
53 02015 151124      MOVZL  2,2,SZR     ;SKIP AFTER TYPING # 15
54 02016 000771      JMP    INM?+2
55 02017 006143      JSR    @ICRL?F
56 02020 032135      LDA    2,@ISW?REG ;AC2 HAD SWITCH SETTINGS
57 02021 151140      MOVOL  2,2         ;BRING THE CARRY BIT IN AC1
58 02022 126560      SUBCL  1,1         ;TYPE THE CONTENTS OF AC1
59 02023 006144      JSR    @IPDC?S
60 02024 000040      40
61 02025 006150      JSR    @IIPS?P
62 02026 151124      MOVZL  2,2,SZR

```

```

0036 .MAIN
01                                     ;BITS
02 02027 000773      JMP     .-5
03 02030 006143      JSR    @ICRL?F
04 02031 010631 IN4?: ISZ    INL?K      ;SKIP IF THE PROGRAM IS LOCKED
05                                     ;IN SWITCH INPUT MODE
06 02032 000701      JMP    INW?
07 02033 014627      DSZ   INL?K      ;NEVER SKIP
08 02034 034136      LDA    3,IOM?0D   ;CHK FOR I/O MOD
09 02035 175005      MOV    3,3,SNR    ;SKP IF I/O MOD LOADED
10 02036 000404      JMP    .+4
11 02037 006136      JSR    @IOM?0D    ;INPUT CHAR TO AC0
12 02040 000001      1
13 02041 000635      JMP    IN0?
14 02042 063610      SKPDN  TTI        ;WAIT FOR OPERATOR INPUT
15 02043 000777      JMP    .-1
16 02044 000632      JMP    IN0?
17
18 02045 000600 TOD?Y: JMP    TOD?X      ;TO REACH ADDRESS
19
20 02046 040421 INS?V: STA    0,INS?0     ;SAVE THE ACC.
21 02047 044421      STA    1,INS?1
22 02050 050421      STA    2,INS?2
23 02051 102560      SUBCL  0,0         ;SAVE THE CARRY
24 02052 010607      ISZ   INT?E     ;SKIP IF ODT IS ENTERED THRU
25                                     ;INTERRUPT HANDLER
26 02053 063577      SKPHZ  CPU        ;SKIP IF INTERRUPTS ARE NOT
27                                     ;ENARLED
28 02054 101141      MOVOL  0,0,SKP
29 02055 101120      MOVZL  0,0
30 02056 060277      INTOS
31 02057 063611      SKPDN  TTO        ;SAVE THE TIO STATUS
32 02060 101121      MOVZL  0,0,SKP
33 02061 101140      MOVOL  0,0
34 02062 040411      STA    0,STA?T
35 02063 000600      JMP    INP?1
36
37 02064 000000 INP?R: 0
38 02065 000604 INP?Q: JMP    INP?J
39 02066 005400 INS?A: JSR    0,3
40 02067 000000 INS?0: 0
41 02070 000000 INS?1: 0
42 02071 000000 INS?2: 0
43 02072 000000 INS?3: 0
44 02073 000000 STA?T: 0
45 02074 177777 INB?A: -1
46 02075 000000 INB?I: 0
47 02076 000307 INS??: RES?T
48 02077 000121 I12?1: 121
49 02100 000060 IN6?0: 60
50 02101 000012 IN1?2: 12
51 02102 000015 IN1?5: 15
52
53 0?DTP
54 0?DIO 2
55
56                                     ;2. OCTAL DERIBG TOOL (ODT)
57                                     ;
58                                     ; THE DIAGNOSTIC IS EQUIPED WITH A BUILT IN ODT WHICH CAN
59                                     ; BE ACCESSED BY HITTING CONTROL 0 (O) AT ANY TIME DURING
60                                     ; THE EXECUTION OF THE PROGRAM (AFTER SETTING THE PARA-

```

0037 .MAIN

```
01 : METERS).
02 : ON ENTERING ODT THE ADDRESS OF THE LOCATION HAVING THE
03 : NEXT INSTRUCTION TO BE EXECUTED WILL BE TYPED-OUT.
04 :
05 :2.1 CONVENTIONS AND SYMBOLS
06 : THE FOLLOWING CONVENTIONS ARE USED BY THE ODT:
07 : ? PRESSING ANY ILLEGAL KEY CAUSES THE ODT TO RES-
08 : POND WITH A "?".
09 : @ ODT IS READY AND AT YOUR SERVICE.
10 :
11 :2.2 COMMAND STRUCTURE
12 : AN ODT COMMAND HAS THE FOLLOWING FORMAT:
13 : [ARGUMENT] [COMMAND]
14 : AN ARGUMENT MAY BE ONE OF THE FOLLOWING:
15 : "EXP" AN OCTAL EXPRESSION CONSISTING OF OCTAL NUMBERS
16 : SEPARATED BY PLUS (+) OR MINUS (-) SIGNS. LEAD-
17 : ING ZEROS NEED NOT BE TYPED.
18 : "ADR" AN ADDRESS IS THE SAME AS AN EXPRESSION EXCEPT
19 : THAT BIT 0 IS NEGLECTED.
20 : A COMMAND IS A SINGLE TELETYPE CHARACTER
21 :
22 :2.3 ODT COMMANDS
23 : THE LOCATIONS THAT CAN BE EXAMINED AND MODIFIED BY THE
24 : USER ARE CALLED CELLS. THESE CELLS ARE OF TWO TYPES:
25 : INTERNAL CPU CELLS AND MEMORY LOCATIONS.
26 :
27 :2.3.1 OPENING INTERNAL CELLS
28 : THE COMMAND TO OPEN ONE OF THE INTERNAL REGISTERS IS OF
29 : THE FORM "NA" WHERE N IS ANY OCTAL EXPRESSION BETWEEN
30 : 0 AND 7
31 : 0-3 FOR ACCUMULATORS 0-3
32 : 4 FOR PC OF THE NEXT INSTRUCTION TO BE EXECUTED IN
33 : THE EVENT OF A "P" COMMAND.
34 : 5 CPU AND TIO STATUS
35 : BIT INTERPRETATION
36 : 15 STATUS OF TIO DONE FLAG
37 : 14 STATUS OF INTERRUPTS (ION FLAG)
38 : 13 STATUS OF CARRY BIT
39 : 6 ADDRESS OF THE LOCATION HAVING THE BREAK POINT (IF
40 : ANY)
41 : 7 INSTRUCTION AT THE BREAK POINT LOCATION
42 :
43 : OTHER COMMANDS TO OPEN CELLS ARE:
44 :
45 : "ADR"/ OPEN THE CELL AND PRINT ITS CONTENTS
46 : ./ OPEN THE CELL CURRENTLY POINTED TO BY THE POINTER
47 : AND PRINT ITS CONTENTS.
48 : .+"ADR"/ ADD "ADR" TO THE POINTER, OPEN THE CELL
49 : AND PRINT ITS CONTENTS.
50 : .-"ADR"/ SUBTRACT "ADR" FROM THE POINTER, OPEN
51 : THE CELL AND PRINT ITS CONTENTS.
52 : "CR" THE RETURN KEY IS USED TO CLOSE THE OPEN CELL
53 : WITH OR WITHOUT MODIFICATION.
54 : "LF" LINE FEED IS USED TO CLOSE THE OPEN CELL WITH OR
55 : WITHOUT MODIFICATION AND TO OPEN THE SUCCEEDING
56 : CELL.
57 : ^ CLOSE THE OPEN CELL WITH OR WITHOUT MODIFICATION
58 : AND OPEN THE PRECEDING CELL
59 : / CLOSE THE OPEN CELL WITHOUT MODIFICATION, AND
60 : OPEN THE CELL POINTED TO BY ITS CONTENTS.
```

0038 .MAIN

```
01 : +"ADR"/ CLOSE THE OPEN CELL WITHOUT MODIFICATION, AND
02 : OPEN THE CELL POINTED TO BY ITS CONTENTS + "ADR".
03 : -"ADR"/ CLOSE THE OPEN CELL WITHOUT MODIFICATION, AND
04 : OPEN THE CELL POINTED TO BY ITS CONTENTS - "ADR".
05 :
06 :2.3.2 MODIFICATION OF A CELL
07 : ONCE A CELL HAS BEEN OPENED ITS CONTENTS CAN BE MODIFIED
08 : BY TYPING THE NEW VALUE THE CELL IS TO CONTAIN IN THE
09 : FORM OF AN OCTAL EXPRESSION FOLLOWED BY "CR" OR "LF".
10 : IF A + OR - IS TYPED AS THE FIRST CHARACTER OF THE EX-
11 : PRESSION THEN THE VALUE OF THE EXPRESSION IS ADDED TO OR
12 : SUBTRACTED FROM THE OLD CONTENTS OF THE CELL. THE
13 : ADDRESS ITSELF OR AN EXPRESSION RELATIVE TO THE ADDRESS
14 : CAN BE DEPOSITED BY TYPING A "." OR ".+/-OCTAL EXPRESS-
15 : ION". A RUBOUT COMMAND GIVEN RIGHT AFTER OPENING A CELL
16 : ALLOWS THE MODIFICATION OF ITS CONTENTS AS IF THEY WERE
17 : TYPED IN JUST BEFORE THE COMMAND WAS ISSUED.
18 :
19 :2.3.3 OTHER ODT COMMANDS
20 : RUBOUT THIS KEY IS USED TO DELETE ERRONEOUSLY TYPED
21 : DIGITS. EACH TIME THE KEY IS PRESSED THE RIGHT MOST
22 : DIGIT IS DELETED AND ECHOED ON THE TERMINAL. IF
23 : THE RUBOUT KEY IS PRESSED RIGHT AFTER OPENING A
24 : CELL THEN IT DELETES THE RIGHT MOST DIGIT OF THE CELL'S
25 : CONTENTS. THIS ALLOWS THE MODIFICATION OF THE CELL
26 : AS IF ITS CONTENTS WERE TYPED IN JUST BEFORE THE
27 : KEY WAS PRESSED.
28 : "ADR"b INSERT A BREAK POINT AT LOCATION "ADR".
29 : ONLY ONE BREAK POINT CAN BE INSERTED AND ANY
30 : ENTRY TO ODT AFTER EXECUTING A BREAK POINT WILL
31 : CAUSE IT TO BE DELETED.
32 : D DELETE THE BREAK POINT IF ANY.
33 : P RESTART THE EXECUTION OF THE PROGRAM AT LOCATION
34 : POINTED BY 4A.
35 : "ADR"R START EXECUTING THE PROGRAM AT "ADR" AFTER AN
36 : IO-RESET.
37 : K KILL THE STRING TYPED SO FAR. THE ODT RESPONDS
38 : WITH A "?" AND THE OPEN CELL IS CLOSED WITHOUT
39 : MODIFICATION.
40 : = PRINT THE OCTAL VALUE OF THE INPUT ONLY.
41 : THIS WILL CLOSE ANY OPEN CELLS WITHOUT
42 : MODIFICATION AND WILL NOT OPEN A CELL
43 :
44 :
45 : NOTE: IN PROGRAMS WHICH RELOCATE THEMSELVES THE
46 : THE USER SHOULD PLACE BREAK POINTS ONLY IN THE
47 : ORIGINAL PROGRAM AREA. IF A BREAK POINT IS
48 : PLACED OUTSIDE THIS AREA THE RESULTS WILL
49 : BE UNPREDICTABLE.
50 :
51 :
52 : THIS PACKAGE IS USED TO PROVIDE AN INTERNAL
53 : OCTAL EDITOR/DEBUGGER TO A PROGRAM.
54 : ODTPK REQUIRES A SINGLE-PAGE ZERO LOCATION FOR
55 : A BREAKPOINT WHICH MUST HAVE THE LABEL ODO?K
56 : IT ALSO REQUIRES THE TTYIO AND SWPAK PACKAGES
57 : THE ENTRY TO THE ODTPK IS ODT?J WHICH SHOULD BE CALLED
58 : ONLY IF A CONTROL 0 (^O) WAS INPUT FROM THE TTY.
59 : THIS MAY BE DONE USING THE INP?K ENTRY OF THIS PACKAGE
60 :
```

0039 .MAIN

```
01 ;THE ROUTINE TO CALL ODTPK COULD LOOK LIKE THIS:
02 ;
03 ;
04 ;     EITHER
05 ;         SKPDN  TTI    ;KEY HIT?
06 ;         JMP     .+5    ;NO
07 ;         JSR    @IINP? ;TO SWPAK
08 ;         LDA    1,CONO ;CHECK FOR
09 ;         SUB#   1,3,SNR ;CONTROL 0
10 ;         JSR    @IODT? ;TO ODTPK
11 ;
12 ;
13 ;
14 ;         CONO:  17      ;CONTROL 0
15 ;
16 ;WHERE IINP? POINTS TO INP?J AND IODT? POINTS TO ODT?J
17 ;
18 ;     OR
19 ;         SKPDZ  TTI    ;KEY HIT?
20 ;         JSR    @IINP? ;TO SWPAK
21 ;
22 ;
23 ;
24 ;
25 ;WHERE IINP? POINTS TO INP?K
26 ;
27 ;NOTE: INP?J IS AN ENTRY IN SWPAK
28 ;THE ENTRY ODT?K IS USED BY PROGRAMS WHICH RELOCATE THEMSELVES
29 ;AND SHOULD BE CALLED ONCE IN THE PROGRAM. THE CALL MUST BE
30 ;FOLLOWED BY THE ADDRESS OF THE LOCATION USED BY THE PROGRAM
31 ;TO STORE THE RELOCATION CONSTANT.
32 ;
33 ;THE CAL TO ODT?K IS:
34 ;
35 ;     JSR    @IOD?K
36 ;     RELOC
37 ;
38 ;WHERE IOD?K POINTS TO ODT?K AND RELOC IS THE LOCATION OF THE
39 ;RELOCATION CONSTANT.
40 ;
41 02103 006144 ODA?C: JSR    @IPDC?S    ;TYPE THE ACCUMULATOR NUMBER TO
42 ;BE OPENED
43 02104 000101 010?1: JSR    @ITPS?P    ;TYPE A SPACE
44 02105 006150 ODA??: JSR    @INS?A     ;LOAD ACC WITH THE ADDRESS OF
45 02106 004760 JSR    INS?A     ;INS?0
46 ;
47 02107 171000 MOV    3,2
48 02110 133000 ADD    1,2
49 02111 034575 LDA    3,OD?7? ;AC3 = 7
50 02112 136453 SUB#   1,3,SNR  ;SKIP IF ACC. NUMBER IS NOT OK
51 02113 000515 JMP    ODA?L
52 ;
53 02114 116005 ODP?C: ADC    0,3,SNR
54 02115 000617 ODR?T: JMP    INR?+1 ;IF IT IS A "P" THEN PROCEED
55 02116 034762 LDA    3,OD6?0 ;AC3 = 60
56 02117 116005 ADC    0,3,SNR  ;SKIP IF ACC IS NOT "/"
57 02120 000476 JMP    ODL?C
58 02121 145000 MOV    2,1
59 02122 175225 MOVZR  3,3,SNR  ;SKIP IF IT IS NOT A "."
60 02123 000561 JMP    ODE?4
```

0040 .MAIN

```
01 02124 006145 JSR    @IMES?S    ;TYPE A "?"
02 02125 002316 TP??
03 02126 000441 JMP    ODW?T
04 ;
05 02127 000717 INS?X: JMP    INS?V
06 02130 000735 IINR?: JMP    INP?G
07 02131 054741 ODT?I: STA    3,INS?3
08 ;
09 ;
10 02132 034000 LDA    3,0
11 02133 00040? JMP    .+2
12 02134 054736 ODT?J: STA    3,INS?3
13 ;
14 02135 054727 STA    3,INP?R
15 02136 004713 JSR    INS?V+3
16 02137 152400 SUB    2,2
17 02140 024724 LDA    1,INP?R
18 02141 044723 ODT?1: STA    1,INP?R
19 02142 006146 JSR    @ITOC?T
20 02143 151404 INC    2,2,SZR
21 ;
22 02144 000422 JMP    ODW?T-1
23 02145 020730 ODD?B: LDA    0,INR?I
24 ;
25 02146 034726 LDA    3,INR?A
26 02147 126001 ADC    1,1,SKP
27 02150 055000 ODB?P: STA    3,0,2
28 02151 175415 INC#   3,3,SNR
29 ;
30 02152 000412 JMP    ODT?2
31 02153 044557 STA    1,INR?1
32 02154 042720 STA    0,@INR?A
33 02155 024543 LDA    1,ODT?3
34 02156 125415 INC#   1,1,SNR
35 02157 000405 JMP    ODT?2
36 02160 034714 LDA    3,INR?A
37 02161 026537 LDA    1,@ODT?3
38 02162 137000 ADD    1,3
39 02163 041400 STA    0,0,3
40 02164 024546 ODT?2: LDA    1,INR?1
41 02165 044707 STA    1,INR?A
42 02166 060210 NIOC   TTI
43 02167 006143 ODW?T: JSR    @ICRL?F
44 02170 054524 STA    3,ODL?T
45 ;
46 02171 006145 JSR    @IMES?S
47 02172 002317 ODT?P
48 02173 004652 JSR    TOD?Y
49 ;
50 02174 060077 NIO    CPU
51 02175 034513 LDA    3,ODE?0
52 02176 116415 SUB#   0,3,SNR
53 02177 000476 JMP    ODE?1
54 02200 034511 LDA    3,ODU?A
55 02201 116415 SUB#   0,3,SNR
56 02202 000475 JMP    ODE?2
57 02203 034676 LDA    3,OD1?2
58 02204 116414 SUB#   0,3,SZR
59 02205 034675 LDA    3,OD1?5
60 02206 116414 SUB#   0,3,SZR

;NORMALLY THE PROGRAM SHOULD
;ENTER ODT AT THIS LOCATION
;THROUGH AN INTERRUPT HANDLER
;AC3 = RETURN ADDRESS

;ODT SHOULD BE ENTERED AT THIS
;POINT THRU A JSR
;"INP?R" HAS THE RETURN ADD.
;SAVE THE STATUS

;TYPE THE HALT LOCATION+1
;SKIP IF THE BREAK POINT HAS
;TO BE DELETED

;LOAD ACC IN CASE THE BREAK
;POINT IS TO BE REMOVED

;SKIP IF THERE IS A BREAK
;POINT

;CHECK FOR PROGRAM RELOCATION
;SKIP IF YES

;ENABLE BREAK POINTS
;CLEAR THE TTY INPUT DOME SIGNAL

;MAKE THE TAG FOR OPEN LOCATION
;NON =1 (NO LOCATION IS OPEN)

;ACCEPT OCTAL NUMBERS SEPARATED
;BY + OR - SIGNS
;NO=0P

;SKIP IF NOT "="

;SKIP IF NOT "*"

;CHECK FOR A LINE FEED OR RETURN
```

```

0041 .MAIN
01 02207 000443      JMP      ODD?R      ;DECODE REST OF THE COMMANDS
02 02210 010504      ISZ     ODL?T      ;SKIP IF A LOCATION IS OPEN
03 02211 000402      JMP      .+2
04 02212 045000      STA     1,0,2      ;RESTORE THE OPEN LOCATION
05 02213 101233      MOVZR# 0,0,SNC     ;SKIP IF IT WAS NOT A "CQ"
06 02214 000753      JMP     ODW?T
07 02215 145400      INC     2,1
08
09 02216 152220 ODL?C: ADCZR  2,2      ;IF IT IS A LINE FEED THEN ADD
10 02217 147400      AND     2,1        ;I TO AC2
11 02220 020466      LDA     0,OD7?     ;AC2 = 77777
12 02221 004645      JSR     INS?A      ;NEGLECT ADDRESS BIT 0
13
14 02222 166422      SUBZ   3,1,SZC     ;ACO = 7
15
16 02223 122433      SUBZ#  1,0,SNC     ;LOAD AC3 WITH THE ADDRESS
17 02224 167001      ADD     3,1,SKP    ;OF INS?0
18 02225 000656      JMP     ODA?C      ;SKIP IF THE ADDRESS IN AC1 IS
19 02226 006146      JSR     @IZOC?T    ;LESS THAN THE ADDRESS OF INS?0
20 02227 131000      MOV     1,2        ;SKIP IF AC1 IS 7 OR LESS
21 02230 126000 ODA?L: ADC     1,1
22 02231 044463      STA     1,ODL?T    ;OTHERWISE IT MUST BE AN ACC.
23 02232 050460      STA     2,ODD?T    ;TYPE AC1
24 02233 025000      LDA     1,0,2
25 02234 006147      JSR     @IPOC?T    ;OPEN A LOCATION
26 02235 000736      JMP     ODW?T+4    ;TYPE THE OPEN LOCATION
27
28 02236 175665 ODD?C: INCCR  3,3,SNR    ;SKIP IF THE COMMAND IS NOT
29
30 02237 000706      JMP     ODD?B      ;"D" TO DELETE HREAK POINT
31 02240 034637      LDA     3,112?1    ;AC3=121
32 02241 162014      ADC#   3,0,SZR     ;IF IT WAS A "R" THEN START THE
33
34 02242 000652      JMP     ODP?C      ;USERS PROGRAM
35 02243 062677      IORST  ;NOVA/ECLIPSE IORST
36 02244 061277      DOAC   0,CPU      ;MICRO NOVA IORST
37 02245 044617      STA     1,INP?R
38 02246 026441      LDA     1,@ODI?T  ;
39 02247 046440      STA     1,@ODI?T  ;I.T. IORST
40 02250 024614      LDA     1,INP?R ;RESTORE AC1
41 02251 000644      JMP     ODR?T
42
43 02252 034632 ODD?R: LDA     3,010?1    ;AC3=101
44 02253 116405      SUB     0,3,SNR    ;IF AN "A" HAS BEEN TYPED
45 02254 000631      JMP     ODA?       ;THEN GO TO SERVICE ACCUMULATOR
46
47 02255 175655      INCOR# 3,3,SNR    ;ROUTINE
48 02256 034616      LDA     3,INB?A    ;SKIP IF IT WAS NOT A "B"
49 02257 175645      INCOR  3,3,SNR    ;MAKE SURE THAT THIS IS THE
50 02260 125112      MOVL#  1,1,SZC     ;FIRST BREAK POINT
51
52
53 02261 000755      JMP     ODD?C      ;SKIP IF BIT 0 OF THE ADDRESS
54 02262 044612      STA     1,INB?A    ;WHERE THE BREAK POINT WILL
55 02263 036611      LDA     3,@INR?A   ;BE PLACED IS 0
56 02264 054611      STA     3,INB?I   ;DECODE OTHER COMMANDS
57 02265 030426      LDA     2,ODD?F    ;SAVE THE BREAK POINT ADD.
58
59
60 02266 020427      LDA     0,ODI?N    ;SAVE THE BREAK POINT INS?
;RRING THE ADDRESS OF LOC.
;IN PAGE 0 TO BE USED BY THE
;BREAK POINT INSTRUCTION

```

```

0042 .MAIN
01 02267 143000      ADD     2,0        ;ACO = BREAK POINT INST.
02 02270 004660      JSR     ODH?P
03 02271 054601 ODH?E: STA     3,INS?3    ;THIS IS THE BREAK POINT ENTRY
04 02272 004635      JSR     INS?X      ;SAVE EVERYTHING
05 02273 024601      LDA     1,INB?A
06 02274 000645      JMP     ODT?1
07
08 02275 006147 ODE?1: JSR     @IPOC?T
09 02276 000671      JMP     ODW?T
10
11 02277 126520 ODE?2: SUBZL  1,1
12 02300 132400      SUB     1,2
13 02301 145000      MOV     2,1
14 02302 006143      JSR     @ICRL?F
15 02303 000713      JMP     ODL?C
16
17 02304 024406 ODE?4: LDA     1,ODD?T
18 02305 000666      JMP     ODW?T+4
19
20 02306 000007 ODT?1: 7
21 02307 077370 ODI?T: 77370      ;I.T. IORST ADDRESS
22 02310 000075 ODE?0: 75
23 02311 000136 ODU?A: 136
24 02312 000000 ODD?T: 0
25 002101 001?2= IN1?2
26 002102 001?5= IN1?5
27 002100 006?0= IN6?0
28 02313 000126 ODD?F: ODD?K      ;ADDRESS IN PAGE 0 TO BE USED
29
30 02314 000000 ODL?T: 0          ;BY THE BREAK POINT
31 02315 002000 ODI?N: JMP     @0
32 02316 000077 TP??: 77
33 02317 000100 ODT?P: "@      ;PROMPT CHAR
34
35 02320 177777 ODT?3: -1
36
37
38 ; TTY INPUT SERVICE ROUTINE
39 ; THIS ROUTINE IS USED TO HANDLE TTY INPUTS ON A
40 ; NONINTERRUPT BASIS WHEN SWPAK AND ODT ARE
41 ; PRESENT IN A PROGRAM AND SHOULD BE CALLED ONLY
42 ; IF A TTY KEY HAS BEEN HIT.
43 ;
44 ; THE CALL TO INP?K IS:
45 ;
46 ; SKPDZ TTI ;SKIP IF NO TTY DONE
47 ; JSR @INP?
48 ;
49 ; WHERE IINP? IS A PAGE ZERO POINTER TO INP?K
50 ;
51 02321 054410 INP?K: STA     3,INR?K    ;SAVE RETURN ADDRESS
52 02322 044410      STA     1,INR?1    ;SAVE AC1
53 02323 004605      JSR     IINR?      ;CALL SWPAK
54 02324 024407      LDA     1,INR?0    ;CHECK FOR CONTROL 0
55 02325 136415      SUB#   1,3,SNR     ;
56 02326 006140      JSR     @IODT?     ;YES GO TO ODTPK
57 02327 024403      LDA     1,INR?1    ;RESTORE AC1
58 02330 002401      JMP     @INR?K     ;RETURN
59
60 02331 000000 INR?K: 0          ;RETURN

```

```

0043 .MAIN
01 02332 000000 INR?1: 0          ;SAVE AC1
02 02333 000017 INR?0: 17        ;CONTROL 0
03
04 ;THIS ENTRY IS USED BY PROGRAMS WHICH RELOCATE THEMSELVES
05 ;AND SHOULD BE CALLED ONCE IN THE PROGRAM. THE CALL MUST BE
06 ;FOLLOWED BY THE ADDRESS OF THE LOCATION USED BY THE PROGRAM
07 ;TO STORE THE RELOCATION CONSTANT.
08 ;
09 ;THE CAL TO ODT?K IS:
10 ;
11 ;     JSR   @IODTK
12 ;     RELOC
13 ;
14 ;WHERE IODTK POINTS TO ODT?K AND RELOC IS THE LOCATION OF THE
15 ;RELOCATION CONSTANT.
16 ;
17 02334 050776 ODT?K: STA 2,INR?1 ;SAVE AC2
18 02335 031400 LDA 2,0,3 ;GET RELOCATION POINTER
19 02336 050762 STA 2,ODT?3 ;STORE IN ODT?3
20 02337 030773 LDA 2,INR?1 ;RESTORE AC2
21 02340 001401 JMP 1,3 ;RETURN
22
23
24

```

```

!0044 .MAIN
01
02 02341 000040 PTAB: 40
03 02342 000040 TABS: .RLK 40
04
05
06
07 02402 044303 DIRT: .TXTE ICHECKER 409!
08 141705
09 142513
10 120322
11 030264
12 000071
13 02410 000000 000000
14 02411 000002 000002
15 02412 000002 000002
16 02413 000000 000000
17 02414 000000 000000
18 02415 000000 000000
19 02416 000000 000000
20 02417 000000 000000
21 02420 002420 CEWD: .
22
23 02421 047503 .TXT /COPYRIGHT © DATA GENERAL CORPORATION,
24 054520
25 044522
26 044107
27 020124
28 041450
29 020051
30 040504
31 040524
32 043440
33 047105
34 051105
35 046101
36 041440
37 051117
38 047520
39 040522
40 044524
41 047117
42 020054
43 02445 030411 1973, 1974, 1975, 1976, 1977, 1978.
44 033471
45 026063
46 030440
47 033471
48 026064
49 030440
50 033471
51 026065
52 030440
53 033471
54 026066
55 030440
56 033471
57 026067
58 030440
59 033471
60 027070

```


0045 .MAIN

01 02467 004440
 02 046101
 03 020114
 04 044522
 05 044107
 06 051524
 07 051040
 08 051505
 09 051105
 10 042526
 11 027104
 12 000000
 13 02503 044514
 14 042503
 15 051516
 16 042105
 17 046440
 18 052101
 19 051105
 20 040511
 21 026514
 22 051120
 23 050117
 24 051105
 25 054524
 26 047440
 27 020106
 28 040504
 29 040524
 30 043440
 31 047105
 32 051105
 33 046101
 34 02530 041411
 35 051117
 36 047520
 37 040522
 38 044524
 39 047117
 40 000056
 41
 42
 43

ALL RIGHTS RESERVED./

.TXT /LICENSED MATERIAL-PROPERTY OF DATA GENERAL

CORPORATION./

.END

**00000 TOTAL ERRORS, 00000 PASS 1 ERRORS

0046 .MAIN

AC3 000067 7/32 12/18 12/21 16/16 16/35 16/47
 ADR 000003 7/05 9/41 10/13 16/05 16/51 19/19
 AUTND 000446 12/46 13/12
 AUT?E 000356 12/38 12/44
 AUT?N 000440 12/49 12/58 13/06
 BACK 001150 22/38 22/46
 BEGIN 000560 7/40 7/60 16/03
 BEGL 000122 7/59 10/17 10/18 10/19 21/11 21/29
 BPROG 000101 7/42 10/22 16/60 19/28
 C0700 000072 7/35 16/06
 C10 000133 9/07 9/34 19/09
 C101 000040 7/13 19/06
 C10K 000107 7/48 21/23
 C10?0 001613 31/05 34/23
 C1774 000070 7/33
 C174K 000130 9/04 13/10
 C20 000104 7/45 16/13 17/06 18/34 18/39 20/02 20/31
 21/17
 C40 000106 7/47 21/12 21/22
 C400 000041 7/14 12/39 12/53 21/18
 C60 000044 7/17
 C77 000042 7/15 18/41 20/34
 C7777 000043 7/16 19/03
 CAC?0 001202 24/31 24/33 24/48
 CANDR 000154 9/30 12/05
 CATSW 000010 7/10 10/55 12/07 13/33
 CBEG 000123 7/60 10/30
 CEND 002420 7/40 9/21 44/21
 CHA?3 001215 24/41 24/44 24/47
 CHC?T 001203 24/13 24/33 26/28 26/42
 CHEK 000772 19/30 20/02 20/29
 CHEK0 001027 19/33 20/31
 CHEK1 001005 20/13 20/27
 CHEK3 001031 20/30 20/33
 CHE?X 001221 24/43 24/48
 CHR?E 001201 24/30 24/37 24/49
 CHR?Z 001330 24/45 26/49 28/30 28/33 28/34 28/37
 CMA 000076 7/39 13/38
 CMASK 000114 7/53 16/58 19/26 21/27
 CONO 000127 9/03 16/31 22/21
 CRL?F 001225 9/15 25/04
 CX 000112 7/51 21/25
 CY 000113 7/52 16/53 19/21 21/26
 DC0?T 001272 26/17 26/37
 DEC?T 001273 26/18
 DET?B 001333 26/09 26/52
 DIRT 002402 7/02 13/17 44/07
 DISTU 000734 19/02 19/17
 EDIST 000074 7/37 16/09 19/04
 EGGS 000060 7/20 7/25 9/14 10/45 10/48 10/51 10/52
 12/06 13/30 13/55 14/01 22/36 22/47
 7/43 10/27 17/03 19/31
 EPROG 000102 7/09 22/03 22/44
 ERET 000007
 ERR1 001077 20/16 22/02
 ERR2 001100 20/21 22/03
 FILL 000642 16/57
 FILL0 000724 17/05 18/39
 FILL1 000653 17/02 17/06
 FILL2 000654 17/07 18/37

0047 .MAIN

FILL3	000726	18/38	18/41						
FINAL	000004	7/06	10/04	13/14	13/22	16/07	18/44	20/37	
FLAG	000115	7/54	19/24	20/08	20/33				
FST2D	001505	29/27	29/45	29/49	29/54	30/50	31/27		
I12?1	002077	34/36	36/48	41/31					
ICA?T	000125	8/02	10/57	13/09					
ICEND	000151	9/21	9/30						
ICHK	000754	19/12	19/19	20/39					
ICHK0	000760	19/23	20/36						
ICHK1	000762	19/26							
ICRL?	000143	9/15	13/15	13/24	22/35	34/03	34/48	35/42	
IEGG?	000142	35/53	36/03	40/43	42/14				
IINP?	000141	9/14	12/29						
IINP?	000141	9/13	16/29	22/19					
IINR?	002130	40/06	42/53						
IMES?	000145	9/17	10/41	13/16	13/18	13/20	22/27	22/31	
		40/01	40/46						
IMMA	000153	9/23	10/56	10/58					
IN0?	001676	34/13	36/13	36/16					
IN1?	001763	34/29	35/19						
IN1?0	001660	33/58	34/39						
IN1?2	002101	35/47	36/50	42/25					
IN1?3	001657	33/57	34/32						
IN1?5	002102	35/38	36/51	42/26					
IN2?	001770	35/26	35/32	35/40					
IN3?	001646	33/47	35/24	35/29					
IN3?3	001656	33/56	34/26						
IN4?	002031	33/55	34/41	35/37	36/04				
IN5?	001715	34/30							
IN6?	001731	34/38	34/43						
IN6?0	002100	35/20	36/49	42/27					
INB?A	002074	36/45	40/25	40/32	40/36	40/41	41/48	41/54	
		41/55	42/05						
INB?I	002075	36/46	40/23	41/56					
INH	000005	7/07	16/36	16/40					
INL?K	001662	33/48	33/60	34/11	36/04	36/07			
INM?	002005	35/42	35/52						
INP?1	001663	34/01	36/35						
INP?I	001666	34/05							
INP?J	001671	9/13	34/08	36/38					
INP?K	002321	42/51							
INP?Q	002065	36/38	40/06						
INP?R	002064	34/05	34/08	34/45	35/16	36/37	40/14	40/17	
		40/18	41/37	41/40					
INR?	001733	34/22	34/46	36/06	39/54				
INR?1	002332	40/31	40/40	42/52	42/57	43/01	43/17	43/20	
INR?K	002331	42/51	42/58	42/60					
INR?0	002333	42/54	43/02						
INS?	002076	34/43	36/47						
INS?0	002067	34/12	35/10	36/20	36/40				
INS?1	002070	35/11	36/21	36/41					
INS?2	002071	35/12	36/22	36/42					
INS?3	002072	34/18	35/13	36/43	40/07	40/12	42/03		
INS?A	002066	36/39	39/45	41/12					
INS?V	002046	34/10	36/20	40/05	40/15				
INS?X	002127	40/05	42/04						
INT?	001556	27/46	30/32	34/02	34/49				
INT?E	001661	33/59	34/09	36/24					

0048 .MAIN

I0DT?	000140	9/12	12/20	13/58	16/02	16/33	22/23	42/56	
I0M?0	000136	9/10	12/35	12/43	12/57	16/19	16/22	22/09	
		22/12	27/55	27/58	28/15	31/09	31/12	34/13	
		36/08	36/11						
IPAT	000635	16/51	18/46						
IPAT0	000641	16/55	18/43						
IPDC?	000144	9/16	35/44	35/57	39/41				
IPDE?	000152	9/22	10/44						
IPOC?	000147	9/19	13/23	22/26	22/30	22/34	41/25	42/08	
ISWR?	000135	9/09	19/08	27/50	33/49	33/54	34/42	34/50	
		34/55	35/54						
ITPS?	000150	9/20	35/50	35/59	39/44				
ITYP?	000137	9/11	24/42	25/07	25/09	30/42	31/30	34/33	
		34/35	35/19						
IZOC?	000146	9/18	40/19	41/19					
K1000	000075	7/38	12/08	13/07					
K12?	001234	25/08	25/11	26/08					
K15?	001235	25/06	25/12						
KNT	000110	7/49							
M10	000132	9/06	9/31						
M40	000105	7/46	10/11	10/20	10/24				
MS	000071	7/34							
MESIZ	000527	13/21	15/02						
MES?M	001164	24/07	24/14						
MES?S	001156	9/17	24/01						
MKT	000571	16/12							
MKT0	000576	16/17	16/49						
MKTK	000117	7/56	16/14	16/48	20/03	20/26			
MMA	000467	8/02	9/23	13/30					
MMA1	000502	13/41	13/53						
MMA2	000507	7/39	13/47	13/59					
MDDUA	000073	7/36	16/11	19/02					
MOVE	000223	10/29	10/36						
MPRUN	000552	13/19	15/21						
MSIZ	000307	7/04	12/03						
MSIZ1	000316	12/10	12/24						
MSIZ2	000335	12/17	12/25	12/27					
MT0?P	000134	9/08	12/27						
O10?1	002104	39/43	41/43						
OD1?2	002101	40/57	42/25						
OD1?5	002102	40/59	42/26						
OD6?0	002100	39/55	42/27						
OD7?	002306	39/49	41/11	42/20					
ODA?	002105	39/44	41/45						
ODA?C	002103	39/41	41/18						
ODA?L	002230	39/51	41/21						
ODB?E	002271	42/03							
ODB?P	002150	40/27	42/02						
ODC?R	002145	40/23	41/30						
ODD?R	002252	41/01	41/43						
ODE?1	002312	41/23	42/17	42/24					
ODE?2	002275	40/53	42/08						
ODE?2	002277	40/56	42/11						
ODE?4	002304	39/60	42/17						
ODE?Q	002310	40/51	42/22						
ODI?N	002315	41/60	42/31						
ODI?T	002307	41/38	41/39	42/21					
ODL?C	002216	39/57	41/09	42/15					

0049 .MAIN

ODL?T	002314	40/44	41/02	41/22	42/30				
ODD?C	002236	41/28	41/53						
ODD?F	002313	41/57	42/28						
ODD?K	000126	9/02	42/28						
ODP?C	002114	39/53	41/34						
ODR?T	002115	39/54	41/41						
ODT?1	002141	40/18	42/06						
ODT?2	002164	40/30	40/35	40/40					
ODT?3	002320	40/33	40/37	42/35	43/19				
ODT?I	002131	40/07							
ODT?J	002134	9/12	40/12						
ODT?K	002334	43/17							
ODT?P	002317	40/47	42/33						
ODU?A	002311	40/54	42/23						
ODW?T	002167	40/03	40/22	40/43	41/06	41/26	42/09	42/18	
OFF?M	000131	9/05	12/30	12/50					
O?D?T	004502	MC 22/54							
O?D?D	000454	MC 36/54							
O?D?T	011220	MC 36/53							
P1?T?	001506	27/44	28/26	29/50	30/16	34/16			
P3?T?	001177	24/06	24/28						
PAC?0	001326	26/47	27/01	27/16	29/46				
PAC?1	001451	27/02	27/17	28/44	29/38	29/40	30/13		
PAC?2	001327	26/48	27/03	27/18					
PAS?	000546	10/42	15/17						
PAS?S	000066	7/31	10/39	10/43	12/04				
PAT?	000006	7/08	15/15	21/02	21/09				
PA?C?	001614	31/06	31/08	31/14	31/18				
PC1?0	001332	25/53	26/51						
PC1?1	001200	24/29	24/38	26/39					
PC1?2	001607	29/24	30/05	30/38	31/01	34/24			
PC1?5	001610	28/31	30/35	30/41	31/02	34/19			
PC4?0	001457	27/37	28/28	28/50	30/20				
PC6?0	001331	25/51	26/22	26/50					
PC??	001460	28/35	28/51						
PCR?Y	001325	26/46	27/05	27/14					
PDC?1	001265	25/54	26/10						
PDC?2	001263	26/04	26/08						
PDC?S	001250	9/16	25/55						
PDE?C	001260	9/22	26/05						
PLOC	000100	7/41	10/16	10/37	22/39				
PLP?T	001407	27/54	27/60	28/09					
POC?T	001242	9/19	25/49						
PSIZE	000077	7/40	10/05						
PSP?	001175	24/26	26/02	26/41					
PTAB	002341	22/28	22/32	44/02					
RANDO	000103	7/44	10/03						
RB6?0	001644	31/28	31/32						
RES?T	000307	12/02	36/47						
RETUR	000124	8/01	16/04	21/30	22/40				
RST?R	001343	24/15	25/10	26/44	27/14	29/48			
RTN?A	001324	24/01	24/03	24/04	25/04	25/45	25/49	25/56	
		25/58	26/05	26/43	26/45	27/19	29/26	29/47	
		29/51	29/57	30/01					
RUB?	001630	30/19	31/20						
SAV?E	001335	24/02	25/05	25/46	25/50	25/57	26/06	27/01	
		29/52	29/58	30/02					
SEND	000171	10/02	13/29	13/35	13/54				

0050 .MAIN

SPT?G	001176	24/27	26/10	26/38					
STA?T	002073	34/56	35/08	35/14	36/34	36/44			
SWREG	000065	7/30	9/09						
S?W?P	000000	MC 2/27	31/35						
S?W?P	007611	MC 51/34							
TABL	000120	7/57	16/12	17/07					
TABL0	000121	7/58	20/04						
TABS	002342	7/57	7/58	44/03					
TAC?0	001453	27/36	27/39	28/38	28/46				
TAC?1	001454	27/40	28/39	28/38					
TAC?2	001455	27/41	28/40	28/48					
TIN?1	001611	30/44	31/03						
TIN?2	001612	30/24	31/04						
TIN?A	001615	30/14	31/08						
TIN?C	001461	29/24	30/37	30/43					
TIN?D	001521	30/01							
TIN?M	001557	30/30	30/35						
TIN?N	001570	30/40	30/44						
TIN?O	001515	29/57							
TIN?Q	001524	29/60	30/04						
TIN?R	001464	29/27	30/47	31/21					
TIN?S	001531	30/10	30/23						
TIN?W	001535	29/44	30/14	30/59	31/31				
TIN?X	001463	29/26	30/12	30/25					
TIN?Z	001525	29/56	30/05						
TMES	000116	7/55	16/59	18/36	19/27	20/28			
TMP?	001334	26/16	26/29	26/53					
TOD?T	001507	29/51	31/33						
TOD?X	001645	31/33	36/18						
TOD?Y	002045	36/18	40/48						
TPR?T	001427	28/12	28/17	28/26					
TPS?P	001351	9/20	24/44	27/36					
TP??	002316	40/02	42/32						
TSI?	001466	29/29	30/29						
TTY?	001371	27/55							
TYP?E	001354	9/11	27/39	29/25	30/15				
TYP?R	001456	27/43	28/41	28/49					
T?T?I	004630	MC 16/17	22/07						
T?T?Y	004711	MC 22/55							
UPDT	001040	21/02							
UPD?T	001067	21/21	21/25						
UPD?C	001076	21/06	21/10	21/13	21/33				
WHICH	000111	7/50	21/14	21/16					
ZOC?T	001236	9/18	25/45						
ZPO?T	001245	25/48	25/52						
ZSU?P	001452	26/15	26/17	26/25	28/45	29/29	29/33	30/09	
		30/10	30/48						
.RAND	000261	10/02	11/02						
.SV0	001153	22/04	22/25	22/41	22/49				
.SV1	001154	22/05	22/29	22/42	22/50				
.SV2	001155	22/06	22/33	22/43	22/51				
.UD03	000303	11/02	11/03	11/19	11/21				
.UD10	000304	11/07	11/10	11/22					
.UD20	000305	11/16	11/23						
.UD21	000306	11/06	11/24						

